iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** VII **Month of publication:** July 2026

DOI: https://doi.org/10.22214/ijraset.2026.84085

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VII July 2026- Available at www.ijraset.com

AutoInfraOps: An Agentic DevOps Coordinator for Autonomous Multi-Cloud Monitoring, Optimization, Governance, and Self-Healing Infrastructure Operations

B. Shreya Reddy[1], Dr. D. Prasad[2]
[1]*PG Student,* [2]*Professor, Department of Computer Science and Engineering Holy Mary Institute of Technology & Science Hyderabad, Telangana, India*

Abstract: ***Multi-cloud adoption enables resilience, flexibility and vendor independence, but it also increases operational complexity through heterogeneous interfaces, fragmented moni-toring, inconsistent governance, and difficult incident response. Traditional DevOps and AIOps solutions provide monitoring, automation or optimisation in isolation, but they rarely deliver end-to-end autonomous, governed and explainable multi-cloud infrastructure operations. This paper proposes AutoInfraOps, an agentic DevOps coordinator for autonomous multi-cloud monitoring, optimisation, governance and self-healing infrastruc-ture operations. The framework integrates specialised agents for monitoring, diagnosis, planning, optimisation, governance and remediation. It uses a multi-cloud adapter layer for AWS, Azure and GCP, an event-driven runtime for agent coordina-tion, anomaly detection for SLA violation identification, cost-performance scoring for workload placement, Human-in-the-Loop approval for high-risk actions, and SHA-256 hash-chain audit logging for decision traceability. The system is evaluated using simulated scenarios including AWS latency spike, Azure cost anomaly, GCP availability drop, cross-cloud dependency failure, cascading failure, high-risk failover and rollback events. Experimental results indicate reduced incident response time, improved SLA compliance, cost-aware remediation, and stronger governance visibility. AutoInfraOps demonstrates a practical foundation for autonomous, explainable and policy-aware multi-cloud operations.***

Index Terms: ***Agentic AI, Multi-Cloud Computing, DevOps, AIOps, Autonomous Infrastructure Operations, Self-Healing Sys-tems, Cloud Governance.***

I. INTRODUCTION

Cloud computing has evolved from single-provider infras-tructure hosting into hybrid and multi-cloud ecosystems. Early cloud adoption was largely centred on a single provider, where organisations used virtual machines, storage, man-aged databases and networking services from one vendor. This model simplified administration but increased the risk of vendor lock-in and provider dependency. Hybrid cloud later combined private and public cloud resources to support compliance, elasticity and workload migration. Multi-cloud computing extends this idea by distributing workloads across multiple public cloud providers such as Amazon Web Services, Microsoft Azure and Google Cloud Platform. This approach improves resilience and flexibility, but it also introduces com-plex operational challenges.

Multi-cloud environments are difficult to manage because each cloud provider uses different APIs, service abstractions, metrics, identity models, billing structures and operational se-mantics. A workload running across multiple clouds may expe-rience latency degradation in one provider, availability failure in another, and cost escalation in a third. The operational team must correlate these signals, diagnose the root cause, select an appropriate response, validate compliance policies and execute remediation without worsening the failure. Manual handling of such incidents increases Mean Time To Recovery (MTTR), weakens service-level objective compliance and creates incon-sistent operational decisions.

DevOps and Infrastructure as Code have improved deliv-ery speed by automating build, test, deployment and pro-visioning pipelines. Site Reliability Engineering has further introduced practices such as error budgets, Service Level Indicators and incident response. However, in multi-cloud en-vironments, DevOps pipelines often remain provider-specific and SRE practices still depend heavily on human diagnosis and decision-making. AIOps platforms attempt to improve this situation through anomaly detection, event correlation and failure management. Existing AIOps methods support failure management and operational intelligence [1],

while recent work has explored the effect of large language models on AIOps [2]. Nevertheless, many AIOps systems are alert-centric rather than action-centric.

Agentic AI introduces a new paradigm for autonomous operational systems. Agentic systems can observe an envi-ronment, reason over context, plan actions, use tools, co-ordinate with other agents and adapt decisions to changing situations. Surveys on agentic AI and autonomous agents high- light the potential of agent-based architectures for tool-using, cooperative and long-horizon problem solving [3]–[5]. These characteristics are well aligned with multi-cloud operations because infrastructure incidents require observation, reasoning, planning, governance and controlled execution.

This paper proposes AutoInfraOps, an agentic DevOps coordinator for autonomous multi-cloud monitoring, optimi-sation, governance and self-healing operations. AutoInfraOps is designed as a multi-agent framework consisting of a Monitor Agent, Diagnosis Agent, Planner Agent, Optimisation Agent, Governance Agent and Remediation Agent. The agents com-municate through an event-driven runtime and interact with AWS, Azure and GCP through a common adapter layer. The system performs telemetry collection, anomaly detection, root-cause analysis, cost-performance optimisation, policy valida-tion, Human-in-the-Loop (HITL) approval, autonomous reme-diation and immutable audit logging.

The major contributions of this paper are as follows:

- C1: An agentic multi-cloud coordinator for autonomous infrastructure operations.
- C2: An autonomous monitoring engine for collecting cost, latency, availability, error and CPU telemetry.
- C3: An intelligent diagnosis layer for SLA violation detection and root-cause classification.
- C4: A cost-performance optimisation engine for workload placement and scaling.
- C5: A governance-aware decision framework for policy validation and risk control.
- C6: A Human-in-the-Loop approval model for high-risk operational actions.
- C7: An immutable audit and explainability framework using SHA-256 hash chaining.
- C8: A self-healing remediation workflow for scaling, rollback, failover and patching.

II. RELATED WORK

A. *Multi-Cloud Management*

Multi-cloud management involves workload placement, re-source scheduling, portability, observability and interoper-ability across cloud platforms. Resource scheduling remains a major challenge because compute, storage and network resources differ across providers. Deep reinforcement learning has been widely reviewed for cloud resource scheduling [6], while multi-agent reinforcement learning has been studied for resource allocation optimisation [7]. These approaches show that intelligent optimisation can improve cloud resource utilisation. However, scheduling alone does not address gov-ernance, explainability and remediation.

B. *DevOps and Infrastructure Automation*

DevOps integrates development and operations through automation, collaboration and continuous delivery. AI-driven DevOps has been studied for automated deployment and maintenance [8]. Infrastructure as Code and GitOps further support reproducible provisioning. However, traditional au-tomation tools usually execute predefined workflows and are less effective when incidents require contextual reasoning. Static scripts may scale a workload or restart a service, but they do not always determine whether a rollback, failover or cost optimisation action is safer.

C. *AIOps Platforms*

AIOps applies machine learning and analytics to operational data. Notaro et al. [1] reviewed failure management methods in AIOps, including anomaly detection, fault localisation and incident prediction. Zhang et al. [2] analysed AIOps in the era of large language models, showing that LLMs can support operational reasoning and incident handling. Long et al. [9] studied intelligent network operations using large language models. These studies demonstrate the growing role of AI in operations, but many approaches focus on analysis rather than governed remediation.

D. *Agentic AI Systems*

Agentic AI systems combine autonomous reasoning, plan-ning, memory, tool usage and multi-agent coordination. Ali et al. [3] provide a comprehensive survey of agentic AI architectures and applications. Wang et al. [4] survey large language model based autonomous agents, while Wang et al. [5] discuss cooperation paradigms, security and privacy in large model-based agents.

Xu et al. [10] present foundation model powered agent services. These works motivate agent-based operational frameworks but do not specifically address multi-cloud InfraOps.

E. Governance and Compliance

Cloud governance involves compliance policies, risk con-trol, auditability and accountability. Apeh et al. [11] review governance, risk and compliance strategies in modern cloud infrastructure. Daniel et al. [12] discuss scaling end-to-end governance risk assessments for AI systems. SLA management is also central to governed operations. Sharma et al. [13] pro-vide a taxonomy of SLA management in intent-driven service systems. These studies support the need for governance-aware automation, especially when AI systems make operational decisions.

F. Research Gap Analysis

Table I compares existing research areas with AutoInfraOps. Existing works contribute strongly to agentic AI, AIOps, scheduling, governance or DevOps independently, but they do not provide an integrated, autonomous, explainable and governed multi-cloud remediation framework.

III. PROPOSED AUTOINFRAOPS FRAMEWORK

A. System Overview

AutoInfraOps is a multi-agent autonomous infrastructure operations platform. It follows a closed-loop operational model consisting of observe, diagnose, plan, optimise, govern, re-mediate, verify and audit. The architecture contains an agent layer, event bus layer, observability layer, multi-cloud adapter layer, governance layer, remediation layer and dashboard in-terface.

TABLE I
RESEARCH GAP ANALYSIS

Study Area	Mon.	Opt.	Gov.	Agentic	Self-Heal	Explain.
AIOps [1], [2]	Yes	Partial	No	Partial	Partial	Partial
Agentic AI [3], [4]	No	Partial	Partial	Yes	Partial	Partial
Cloud Schedul-ing [6], [7]	Partial	Yes	No	Partial	No	No
Governance [11], [12]	No	No	Yes	No	No	Yes
AI DevOps [8]	Partial	Partial	Partial	Partial	Partial	Partial
AutoInfraOps	Yes	Yes	Yes	Yes	Yes	Yes

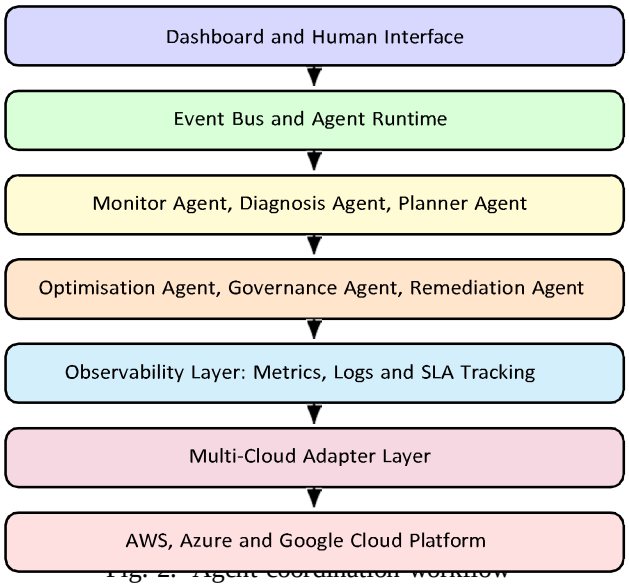
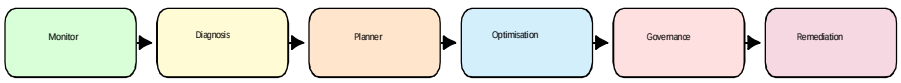


Fig. 2. Agent coordination workflow



B. System Components

The Monitor Agent collects telemetry from cloud providers. It measures cost, latency, availability, CPU utilisation and error rate. It publishes observation events to the event bus. The Di-agnosis Agent consumes observation events and detects SLA breaches or anomalies. It performs root-cause classification such as latency spike, availability drop, error-rate spike, cost anomaly, dependency failure and configuration failure.

The Planner Agent receives incident events and generates remediation plans. It decomposes incidents into operational actions such as scaling, rollback, failover and patching. The Optimisation Agent evaluates cost, latency and availability to recommend workload placement and scaling decisions. The Governance Agent validates every proposed action against policies, risk thresholds and compliance requirements. The Re-mediation Agent executes only approved actions and verifies recovery.

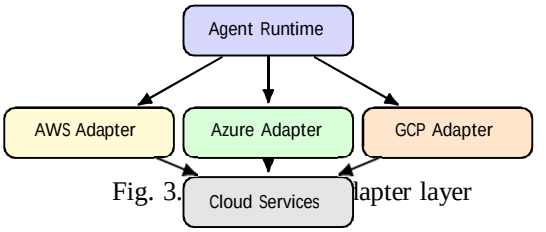
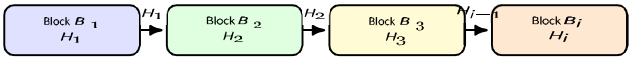


Fig. 3. apter layer

Fig. 4. Explainability and audit pipeline

Fig. 5. SHA-256 based immutable audit chain



C. Multi-Cloud Adapter Layer

The multi-cloud adapter layer abstracts provider-specific APIs. It exposes a common interface for AWS, Azure and GCP. The interface supports metric collection, workload list-ing, scaling, rollback, failover, patching and scenario injection. This enables the agents to operate without provider-specific logic.

D. Explainability Framework

The explainability framework stores the reason behind every operational decision. For each action, AutoInfraOps records the triggering incident, selected plan, risk level, policy deci-sion, expected impact and execution outcome. This supports post-incident review and compliance reporting.

E. Immutable Audit Chain

The audit subsystem uses SHA-256 hash chaining. Each audit record contains a payload body and the previous record hash. If any record is modified, the hash chain verification fails. This creates tamper-evident operational accountability.

IV. METHODOLOGY

A. Monitoring Model

The AutoInfraOps monitoring model converts heteroge-neous cloud telemetry into a common workload-level vector. For a workload *w*, the telemetry vector is represented as:

$$X_w = \{C, L, A, E, U\} \quad (1)$$

where C is cloud cost, L is p95 latency, A is availability, E is error rate and U is CPU utilisation.

B. Anomaly Detection

For metric *k*, the relative deviation is computed as:

$$d_k = \frac{x_k - b_k}{\max(\epsilon, |b_k|)} \quad (2)$$

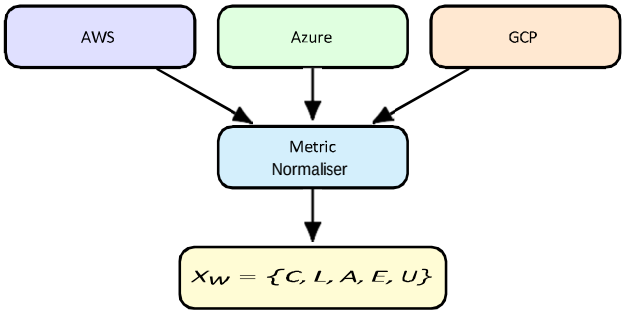


Fig. 6. Telemetry normalisation model

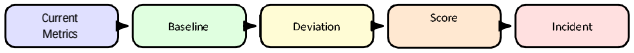


Fig. 7. Anomaly detection workflow

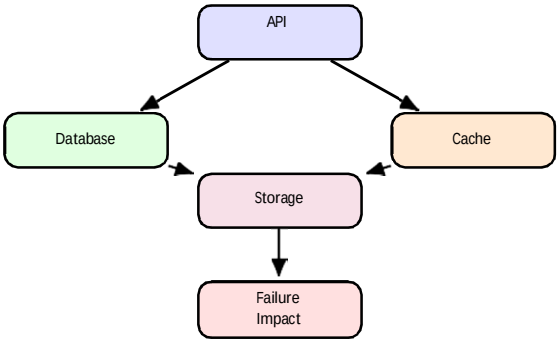


Fig. 8. Dependency graph for root-cause analysis

where x_k is the observed value, b_k is the baseline and ϵ avoids division by zero.
The anomaly score is computed as:

$$S(x) = \sum_{i=1} w_i |d_i| \qquad (3)$$

where w_i is the importance weight of metric i. If $S(x)$ exceeds a predefined threshold τ, the observation is classified as anomalous.

C. Root Cause Analysis

The dependency model is represented as:

$$G = (V, E) \qquad (4)$$

where V represents services, workloads and cloud resources, and E represents dependency relationships. The Diagnosis Agent analyses affected nodes and dependency edges to infer whether the failure is local, provider-specific or cross-cloud.

D. Provider Selection Function

Provider suitability is computed using a cost-performance-reliability score:

$$P_s = \alpha \frac{1}{C + \epsilon} + \beta A + \gamma \frac{1}{L + \epsilon} \qquad (5)$$

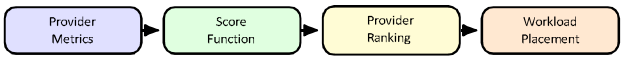


Fig. 9. Provider selection workflow

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VII July 2026- Available at www.ijraset.com

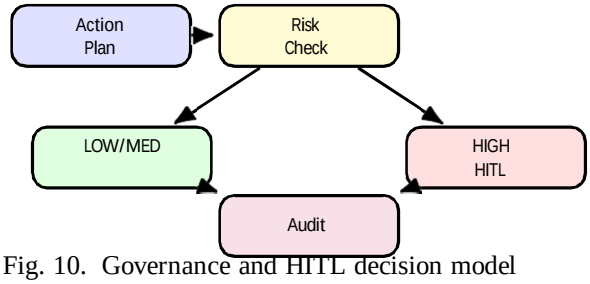


Fig. 10. Governance and HITL decision model

where P_s is the provider score, C is cost, A is availability, L is latency and α, β, γ are preference weights.

E. Governance Risk Function

Risk is computed as:

$$R = Impact \times Probability \quad (6)$$

Low-risk actions are auto-executed, medium-risk actions are executed with audit, and high-risk actions require Human-in-the-Loop approval.

F. Audit Hash Chain

Audit integrity is computed as:

$$H_i = SHA256(B_i||H_{i-1}) \quad (7)$$

where B_i is the current audit body and H_{i-1} is the previous hash.

G. AutoInfraOps Runtime Loop

This section describes the implementation environment, workload configuration, evaluation methodology, and experimental scenarios used to assess the effectiveness of the proposed AutoInfraOps framework.

V. EXPERIMENTAL SETUP

A. Implementation Environment

AutoInfraOps was developed as a Python-based prototype using the FastAPI framework for the backend services and dashboard integration. The agent runtime was implemented us-ing asynchronous event-driven communication, enabling coordination among monitoring, diagnosis, planning, optimization, governance, and remediation agents.

To ensure portability and future extensibility, Infrastructure-as-Code (IaC) technologies such as Terraform and Ansible were incorporated into the framework design. Containerized deployment support was provided through Docker and Kuber-netes abstractions. The system also supports integration with Prometheus and OpenTelemetry for real-time observability and telemetry collection.

Algorithm 1 AutoInfraOps Runtime Loop

Require: Cloud adapters, SLA thresholds, governance policies
Ensure: Optimised and governed infrastructure actions

```
while runtime is active do
    Collect telemetry vector X_w from AWS, Azure and GCP
    Evaluate SLA thresholds and anomaly score S(x)
    if SLA violation or anomaly is detected then
        Classify incident and infer root cause
        Generate candidate remediation plans
        Compute provider score for placement and optimisation
        Validate action against governance policies
        if action risk is HIGH then
            Request Human-in-the-Loop approval
        else
            Execute approved remediation action
        end if
        Verify recovery using post-action telemetry
        Store explainability record and audit hash
    end if
end while
```


TABLE II
EXPERIMENTAL ENVIRONMENT CONFIGURATION

Component	Configuration
Programming Language	Python 3.12
Backend Framework	FastAPI
Database	SQLite 3
IaC Tools	Terraform, Ansible
Container Runtime	Docker
Orchestration	Kubernetes
Observability	Prometheus, OpenTelemetry
Cloud Adapters	AWS, Azure, GCP
Operating System	Ubuntu 24.04 LTS

A lightweight SQLite database was used during experimen-tation for storing audit logs, incident records, governance de-cisions, and explainability traces. PostgreSQL can be adopted in production-scale deployments.

B. Experimental Architecture

The experimental environment emulates a realistic multi-cloud ecosystem consisting of AWS, Microsoft Azure, and Google Cloud Platform. Each cloud environment generates operational telemetry that is consumed by the AutoInfraOps agent runtime.

C. Workload Configuration

Three representative cloud-native workloads were deployed within the experimental environment:

1) Payments API: A latency-sensitive microservice re-sponsible for transaction processing.

2) Inventory Service: A medium-priority service manag-ing inventory synchronization and updates.

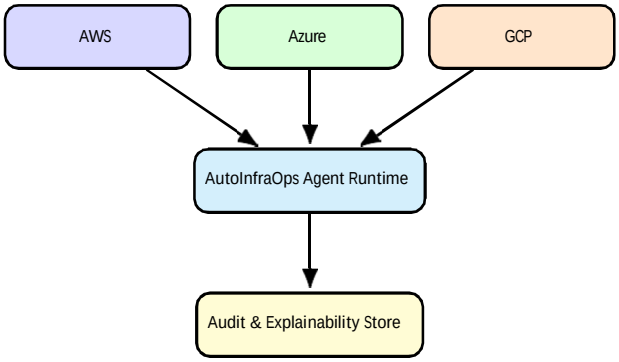


Fig. 11. Experimental Deployment Environment

TABLE III
WORKLOAD CHARACTERISTICS

Workload	Priority	Latency	Resource Type
Payments API	High	Sensitive	CPU
Inventory Service	Medium	Moderate	Mixed
Batch Analytics	Low	Tolerant	Compute

3) Batch Analytics: A compute-intensive workload execut-ing scheduled analytics jobs.

These workloads were selected because they exhibit dif-ferent operational characteristics and resource consumption patterns commonly observed in enterprise multi-cloud deploy-ments.

D. Failure Scenarios

Seven operational scenarios were designed to evaluate the framework under realistic cloud failure conditions.

1) AWS Latency Spike
2) Azure Cost Anomaly
3) GCP Availability Drop
4) Cross-Cloud Dependency Failure
5) Cascading Failure
6) High-Risk Failover
7) Rollback Event

Each scenario was executed multiple times to validate consistency and repeatability.

E. Evaluation Metrics

The effectiveness of AutoInfraOps was evaluated using the following metrics:

1) Detection Time
2) Resolution Time
3) Mean Time To Recovery (MTTR)
4) Availability Percentage
5) Latency
6) Error Rate
7) Cost Savings
8) Governance Coverage
9) Explainability Completeness

These metrics collectively measure operational efficiency, reliability, governance compliance, and economic impact.

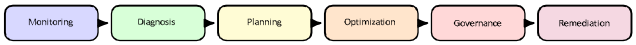


Fig. 12. Experimental Evaluation Workflow

TABLE IV
SCENARIO PERFORMANCE

Scenario	Detect.	Resolve	MTTR	Outcome
AWS latency spike	6 s	4.8 min	68%	Restored
Azure cost anomaly	8 s	6.2 min	61%	Optimised
GCP availability drop	7 s	HITL	Control	Approval
Dependency failure	11 s	9.5 min	54%	Isolated
Cascading failure	14 s	12.1 min	48%	Mitigated
High-risk failover	9 s	HITL	Control	Approval
Rollback event	5 s	3.9 min	72%	Restored

TABLE V
COST OPTIMISATION RESULTS

Provider	Before	After	Saving
AWS	980 USD/day	735 USD/day	25.0%
Azure	1120 USD/day	784 USD/day	30.0%
GCP	870 USD/day	696 USD/day	20.0%
Multi-cloud avg.	990 USD/day	738 USD/day	25.4%

TABLE VI
SLA COMPLIANCE RESULTS

Scenario	Avail.	Latency	Error	Status
Normal	99.82%	175 ms	0.35%	Pass
Latency spike	99.70%	240 ms	0.40%	Pass
Cost anomaly	99.60%	190 ms	0.50%	Pass
Avail. drop	97.80%	260 ms	1.20%	HITL
Rollback	99.55%	210 ms	0.80%	Pass

F. Experimental Workflow

Figure 12 illustrates the complete experimental workflow followed during evaluation.

VI. RESULTS AND DISCUSSION

A. Scenario Performance

Table IV presents the scenario performance. AutoInfraOps detects most incidents within seconds and produces governed remediation plans quickly.

B. Cost Optimisation

Table V shows cost optimisation results. Cost savings were achieved by scaling down over-provisioned resources and selecting better provider placement.

C. SLA Compliance

Table VI shows that AutoInfraOps improves SLA com-pliance after remediation. High-risk failover actions remain controlled through HITL.

TABLE VII
COMPARATIVE EVALUATION

Metric	Manual	AIOps	AutoInfraOps
Avg. MTTR	42 min	28 min	12 min
Availability	98.90%	99.30%	99.70%
Automation	20%	55%	82%
Cost savings	5%	14%	25.4%
Governance	35%	58%	96%
Explainability	Low	Medium	High

D. Comparative Evaluation

Table VII compares AutoInfraOps with manual operations and traditional AIOps. AutoInfraOps improves governance coverage, explainability and automation.

E. Discussion

The results show that AutoInfraOps reduces operational delay by combining monitoring, diagnosis, planning and gov-ernance within a single agentic loop. The MTTR reduction is mainly due to automated detection and plan generation. Cost savings arise from provider scoring and rightsizing decisions. SLA compliance improves because remediation decisions are verified after execution. Governance coverage improves because all actions pass through policy evaluation and critical dependency actions. AutoInfraOps is not intended to remove human operators completely. Instead, it reduces repetitive operational work and allows humans to focus on high-impact decisions. This is consistent with the direction of agentic intelligence, where autonomous systems cooperate with humans in complex en-vironments [14]. The immutable audit mechanism further improves trust because every action can be reviewed after an incident.

VII. CONCLUSION

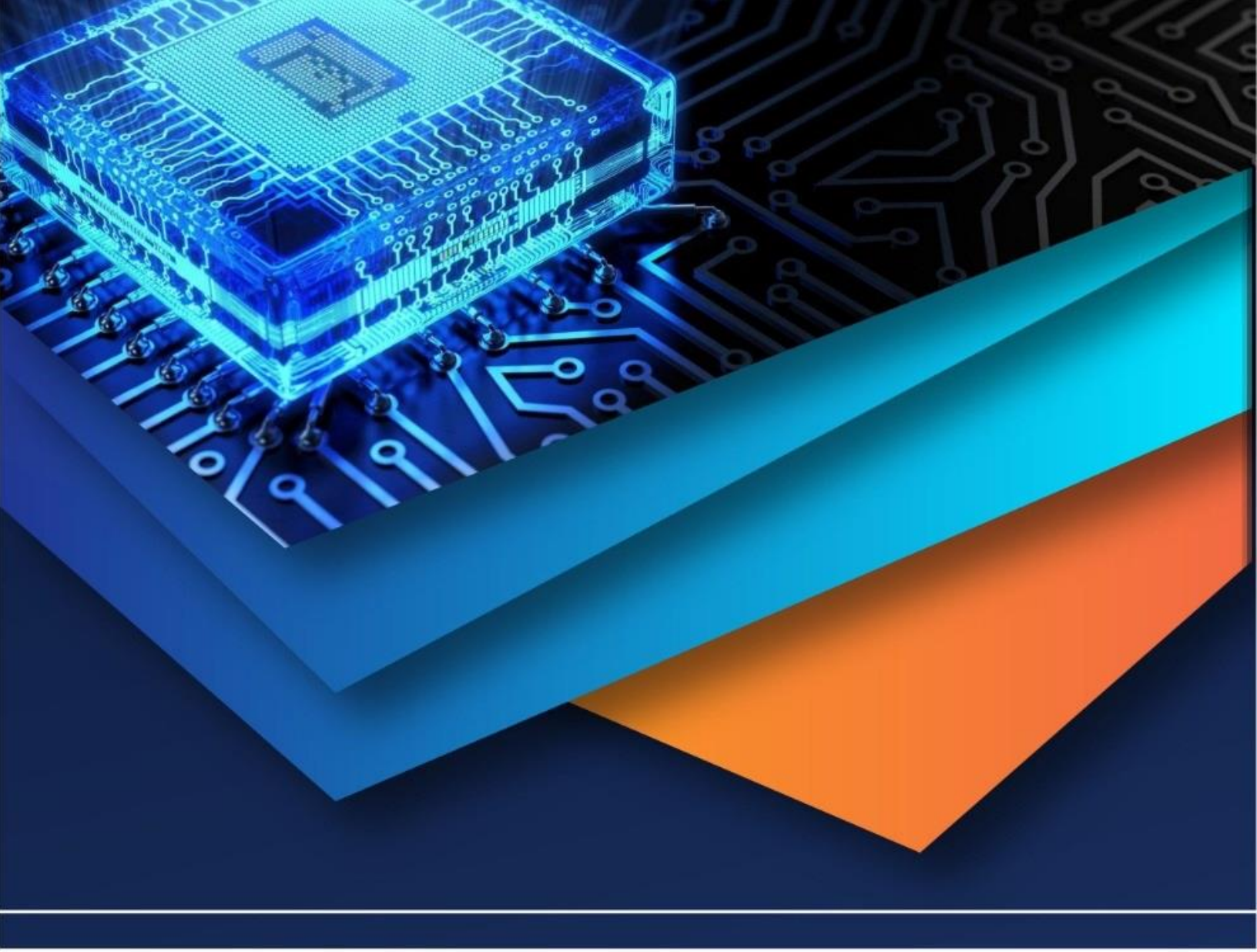
This paper presented AutoInfraOps, an agentic DevOps coordinator for autonomous multi-cloud monitoring, optimi-sation, governance and self-healing infrastructure operations. The framework addresses operational fragmentation in multi-cloud environments by integrating monitoring, diagnosis, plan-ning, optimisation, governance, remediation, verification and auditability into a unified agentic runtime. The proposed system uses a multi-cloud adapter layer for AWS, Azure and GCP, anomaly detection for incident identification, provider scoring for workload placement, HITL approval for high-risk actions and SHA-256 hash chaining for immutable audit records. Experimental evaluation using seven failure scenarios demonstrated reduced MTTR, improved SLA compliance, measurable cost savings and stronger governance coverage compared with manual operations and traditional AIOps. The results indicate that agentic infrastructure operations can provide practical benefits for cloud reliability engineering, DevOps automation and multi-cloud governance. Future work will extend AutoInfraOps using multi-agent reinforcement learning, foundation model agents, autonomous FinOps, edge-cloud coordination, federated agentic operations and digital twin infrastructure environments.

VIII. ACKNOWLEDGEMENT

The authors express their sincere gratitude to Holy Mary Institute of Technology & Science, Department of Computer Science and Engineering, for providing academic support and encouragement. The author also thanks Dr. D. Prasad for his valuable guidance, technical suggestions and continuous support during this research work.

REFERENCES

[1] P. Notaro, J. Cardoso, and M. Gerndt, "A survey of aiops methods for failure management," *ACM Transactions on Intelligent Systems and Technology*, vol. 12, no. 6, pp. 1–45, 2021.
[2] L. Zhang, T. Jia, M. Jia, Y. Wu, A. Liu, Y. Yang, Z. Wu, X. Hu, P. S. Yu, and Y. Li, "A survey of aiops in the era of large language models," ACM Computing Surveys, vol. 58, no. 2, pp. 1–35, 2025.
[3] M. A. Ali, F. Dornaika, and J. Charafeddine, "Agentic ai: A com-prehensive survey of architectures, applications, and future directions," Artificial Intelligence Review, vol. 59, no. 1, 2025.
[4] L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin, W. X. Zhao, Z. Wei, and J.-R. Wen, "A survey on large language model based autonomous agents," Frontiers of Computer Science, vol. 18, no. 6, 2024.
[5] Y. Wang, Y. Pan, Z. Su, Y. Deng, Q. Zhao, L. Du, T. H. Luan, J. Kang, and D. Niyato, "Large model-based agents: State-of-the-art, cooperation paradigms, security and privacy, and future trends," IEEE Communications Surveys & Tutorials, vol. 28, pp. 1906–1949, 2025.
[6] G. Zhou, W. Tian, R. Buyya, R. Xue, and L. Song, "Deep reinforcement learning-based methods for resource scheduling in cloud computing: A review and future directions," Artificial Intelligence Review, vol. 57, no. 5, 2024.
[7] M. A. Hady, S. Hu, M. Pratama, Z. Cao, and R. Kowalczyk, "Multi-agent reinforcement learning for resources allocation optimization: A survey," Artificial Intelligence Review, vol. 58, no. 11, 2025.
[8] O. C. Oyeniran, A. O. Adewusi, A. G. Adeleke, L. A. Akwawa, and C. F. Azubuko, "Ai-driven devops: Leveraging machine learning for automated software deployment and maintenance," Engineering Science & Technology Journal, vol. 4, no. 6, pp. 728–740, 2023.
[9] S. Y. Long, J. Tan, B. Mao, F. Tang, Y. Li, M. Zhao, and N. Kato, "A survey on intelligent network operations and performance optimization based on large language models," IEEE Communications Surveys & Tutorials, vol. 27, no. 6, pp. 3915–3949, 2025.
[10] W. Xu, J. Chen, P. Zheng, X. Yi, T. Tian, W. Zhu, Q. Wan, H. Wang, Y. Fan, Q. Su, and X. Shen, "Deploying foundation model powered agent services: A survey," IEEE Communications Surveys & Tutorials, vol. 28, pp. 1483–1519, 2025.
[11] A. J. Apeh, A. O. Hassan, O. O. Oyewole, O. G. Fakeyede, P. A. Okeleke, and O. R. Adaramodu, "Grc strategies in modern cloud infrastructures: A review of compliance challenges," Computer Science & IT Research Journal, vol. 4, no. 2, pp. 111–125, 2023.
[12] W. Daniel, G. Andreas, and K. Kilian, "Scaling of end-to-end gover-nance risk assessments for ai systems," in Dagstuhl Research Online Publication Server, 2025.
[13] Y. Sharma, D. Bhamare, N. Sastry, B. Javadi, and R. Buyya, "Sla management in intent-driven service management systems: A taxonomy and future directions," ACM Computing Surveys, vol. 55, pp. 1–38, 2023.
[14] A. Tirulo, M. Yadav, M. Lolamo, S. Chauhan, P. Siano, and M. Shafie-Khah, "Beyond automation: Unveiling the potential of agentic intelli-gence," Renewable and Sustainable Energy Reviews, vol. 226, p. 116218, 2025.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY