International Journal of Artificial Intelligence & Machine Learning (IJAIML)
Volume 4, Issue 1, January-June 2025, pp. 234-251, Article ID: IJAIML_04_01_017
Available online at https://iaeme.com/Home/issue/IJAIML?Volume=4&Issue=1
Impact Factor (2025): 3.56 (Based on Google Scholar Citation)
ISSN Online: 2982-6520; Journal ID: 9339-1263
DOI: https://doi.org/10.34218/IJAIML_04_01_017

© **IAEME** Publication

OPEN ACCESS

AI-ENABLED AUTONOMOUS INFRASTRUCTURE MONITORING AND SELF-HEALING CLOUD SYSTEMS

Rajesh Adepu
Associate Principal and IT Architecture, GuideHouse LLC, United States of America.

ABSTRACT

The rapid evolution of cloud computing has introduced unprecedented complexity in managing large-scale, distributed infrastructure environments. Traditional monitoring approaches, which rely heavily on manual intervention and rule-based alerting, are increasingly insufficient in ensuring system reliability, performance optimization, and rapid incident resolution. This paper explores the emerging paradigm of AI-enabled autonomous infrastructure monitoring and self-healing cloud systems, which leverage machine learning, predictive analytics, and intelligent automation to proactively detect, diagnose, and remediate system anomalies.

The proposed approach integrates real-time telemetry data, anomaly detection models, and automated orchestration mechanisms to enable adaptive and resilient cloud operations. By incorporating techniques such as time-series forecasting, root cause analysis, and reinforcement learning-based decision systems, organizations can significantly reduce downtime, minimize operational overhead, and enhance service availability. The study also examines architectural frameworks that combine observability platforms, AI pipelines, and cloud-native orchestration tools to support continuous monitoring and autonomous recovery.

Furthermore, this paper highlights key challenges, including data quality, model drift, explainability, and integration with legacy systems, while proposing mitigation

strategies aligned with enterprise-scale deployments. Through conceptual models and industry-aligned practices, the research demonstrates how AI-driven self-healing systems can transform reactive IT operations into proactive and predictive ecosystems, ultimately improving system resilience and operational efficiency in modern cloud environments.

Keywords: AI-Driven Monitoring, Autonomous Systems, Self-Healing Infrastructure, Cloud Computing, Predictive Analytics, Anomaly Detection, AIOps, Observability, Machine Learning in IT Operations, Fault Detection and Recovery, Intelligent Automation, Cloud Resilience, Distributed Systems Monitoring, Reinforcement Learning, Infrastructure Orchestration

Cite this Article: Rajesh Adepu. (2025). AI-Enabled Autonomous Infrastructure Monitoring and Self-Healing Cloud Systems. *International Journal of Artificial Intelligence & Machine Learning (IJAIML)*, 4(1), 234-251.

https://iaeme.com/MasterAdmin/Journal_uploads/IJAIML/VOLUME_4_ISSUE_1/IJAIML_04_01_017.pdf

1. INTRODUCTION

The increasing adoption of cloud computing has transformed how enterprises design, deploy, and manage IT infrastructure. Modern cloud environments are highly dynamic, distributed, and service-oriented, often spanning multi-cloud and hybrid ecosystems. While these advancements enable scalability and flexibility, they also introduce significant operational complexity, making infrastructure monitoring and incident management more challenging than ever before.

Traditional monitoring systems rely on static thresholds, predefined rules, and manual intervention. These approaches are inherently reactive, often detecting issues only after they impact system performance or user experience. As infrastructure scales, the volume of telemetry data—comprising logs, metrics, and traces—grows exponentially, overwhelming conventional tools and operations teams. This creates a critical need for intelligent systems that can process vast data streams in real time and make autonomous decisions.

Artificial Intelligence (AI) has emerged as a transformative enabler in this domain, giving rise to the concept of AI-driven IT operations (AIOps). By leveraging machine learning (ML), deep learning, and advanced analytics, AI-enabled monitoring systems can identify patterns, detect anomalies, and predict potential failures before they occur. These systems move beyond simple alerting mechanisms to provide actionable insights and automated responses.

A key advancement in this space is the development of self-healing cloud systems. These systems are designed to automatically detect faults and initiate corrective actions without human intervention. For example, when an application instance becomes unresponsive, a self-healing system can automatically restart the service, reallocate resources, or reroute traffic to healthy nodes. This significantly reduces mean time to recovery (MTTR) and enhances system reliability.

The convergence of AI and cloud-native technologies—such as containers, microservices, and orchestration platforms—has further accelerated the adoption of autonomous infrastructure management. Platforms like Kubernetes enable automated scaling and recovery, while AI models enhance decision-making by analyzing historical and real-time data. Together, they form the foundation of intelligent, resilient cloud ecosystems.

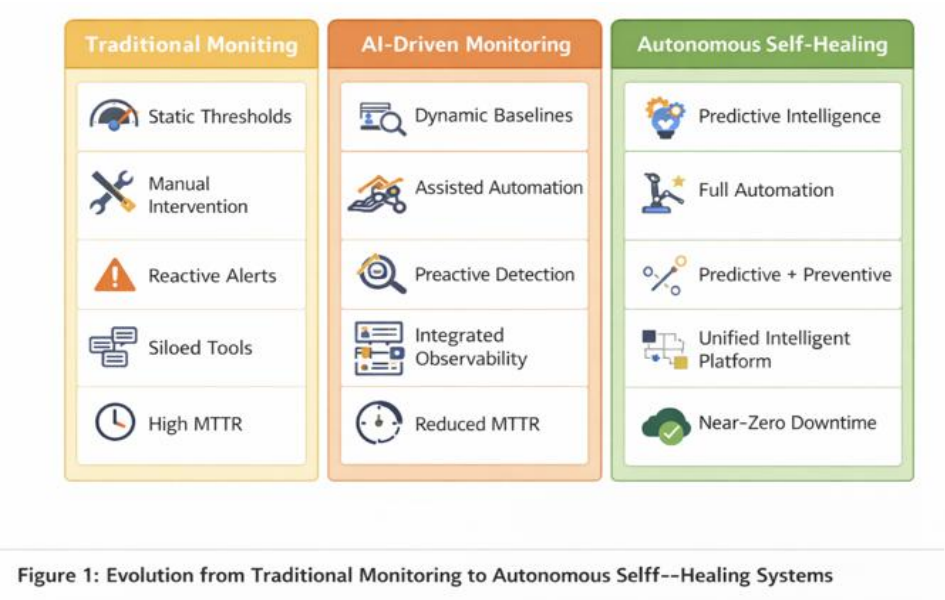


Figure 1: Evolution from Traditional Monitoring to Autonomous Selff--Healing Systems

Figure 1: Evolution from Traditional Monitoring to Autonomous Self-Healing Systems

Key Drivers for AI-Enabled Autonomous Monitoring

Driver	Description
Data Explosion	Massive volumes of logs, metrics, and traces generated continuously
System Complexity	Microservices and distributed architectures increase failure points
Downtime Costs	Business-critical applications demand high availability
Operational Efficiency	Need to reduce manual intervention and human error
Real-Time Decision Making	Faster detection and response to anomalies

Scope of the Paper

This paper aims to provide a comprehensive exploration of AI-enabled autonomous infrastructure monitoring and self-healing cloud systems. It covers architectural strategies, core technologies, implementation models, and practical challenges associated with deploying such systems in enterprise environments. Additionally, it presents frameworks and design considerations that support scalability, resilience, and continuous optimization.

The subsequent sections will delve into the core components of AI-driven monitoring systems, including data pipelines, machine learning models, and automation engines, followed by architectural patterns and real-world implementation strategies.

2. CORE CONCEPTS AND BACKGROUND

The foundation of AI-enabled autonomous infrastructure monitoring and self-healing systems lies in the convergence of cloud-native observability, machine learning, and intelligent automation. This section outlines the essential concepts that underpin these systems and explains how they collectively enable proactive and autonomous operations in modern cloud environments.

2.1 Observability in Cloud Systems

Observability extends beyond traditional monitoring by providing deep visibility into system behavior through three primary data sources:

- **Metrics:** Quantitative measurements such as CPU usage, memory consumption, and network latency
- **Logs:** Time-stamped records of system events and application activities
- **Traces:** End-to-end transaction flows across distributed services

Unlike monitoring, which answers what is happening, observability helps answer why it is happening. This distinction is critical in complex microservices architectures where failures can propagate across multiple components.

2.2 AIOps (Artificial Intelligence for IT Operations)

AIOps represents the application of AI and machine learning techniques to automate and enhance IT operations. It enables:

- **Anomaly Detection:** Identifying deviations from normal behavior using statistical and ML models
- **Event Correlation:** Linking related alerts to reduce noise and identify root causes
- **Predictive Analytics:** Forecasting potential failures based on historical patterns
- **Automated Decision-Making:** Triggering actions without human intervention

AIOps platforms ingest large volumes of telemetry data and continuously learn from system behavior, improving accuracy over time.

2.3 Machine Learning Models in Monitoring

Different types of machine learning models are used to power intelligent monitoring systems:

Model Type	Application in Monitoring
Supervised Learning	Classification of known failure patterns
Unsupervised Learning	Detection of unknown anomalies and clustering patterns
Time-Series Models	Forecasting resource usage and detecting seasonal trends
Reinforcement Learning	Dynamic decision-making for automated remediation

These models enable systems to transition from reactive alerting to predictive and prescriptive operations.

2.4 Self-Healing Systems

Self-healing systems are designed to automatically detect and resolve issues with minimal or no human intervention. Key capabilities include:

- **Auto Recovery:** Restarting failed services or containers
- **Failover Mechanisms:** Switching to backup systems during failures
- **Auto Scaling:** Adjusting resources based on workload demands
- **Configuration Correction:** Reverting or updating faulty configurations

These capabilities are typically implemented using orchestration platforms and automation pipelines.

2.5 Feedback Loop Mechanism

A critical component of autonomous systems is the continuous feedback loop:

1. **Data Collection** → Gather telemetry data
2. **Analysis** → Apply AI/ML models to detect anomalies
3. **Decision** → Determine corrective actions
4. **Execution** → Trigger automated remediation
5. **Learning** → Update models based on outcomes

This closed-loop system ensures continuous improvement and adaptation to changing environments.

2.6 Key Benefits of Autonomous Monitoring Systems

- **Reduced Downtime:** Faster detection and remediation of issues
- **Operational Efficiency:** Minimizes manual intervention
- **Scalability:** Handles large-scale distributed systems effectively

- **Improved Accuracy:** AI reduces false positives and alert fatigue
- **Continuous Optimization:** Systems learn and improve over time

3. SYSTEM ARCHITECTURE

The architecture of AI-enabled autonomous infrastructure monitoring and self-healing cloud systems is designed as a layered and modular framework that integrates observability, intelligence, and automation. These layers collectively enable real-time data ingestion, intelligent analysis, and automated remediation across distributed cloud environments.

As illustrated in Fig. 2, the system architecture consists of five key layers: telemetry data collection, AI and analytics engine, orchestration layer, self-healing execution, and underlying cloud infrastructure.

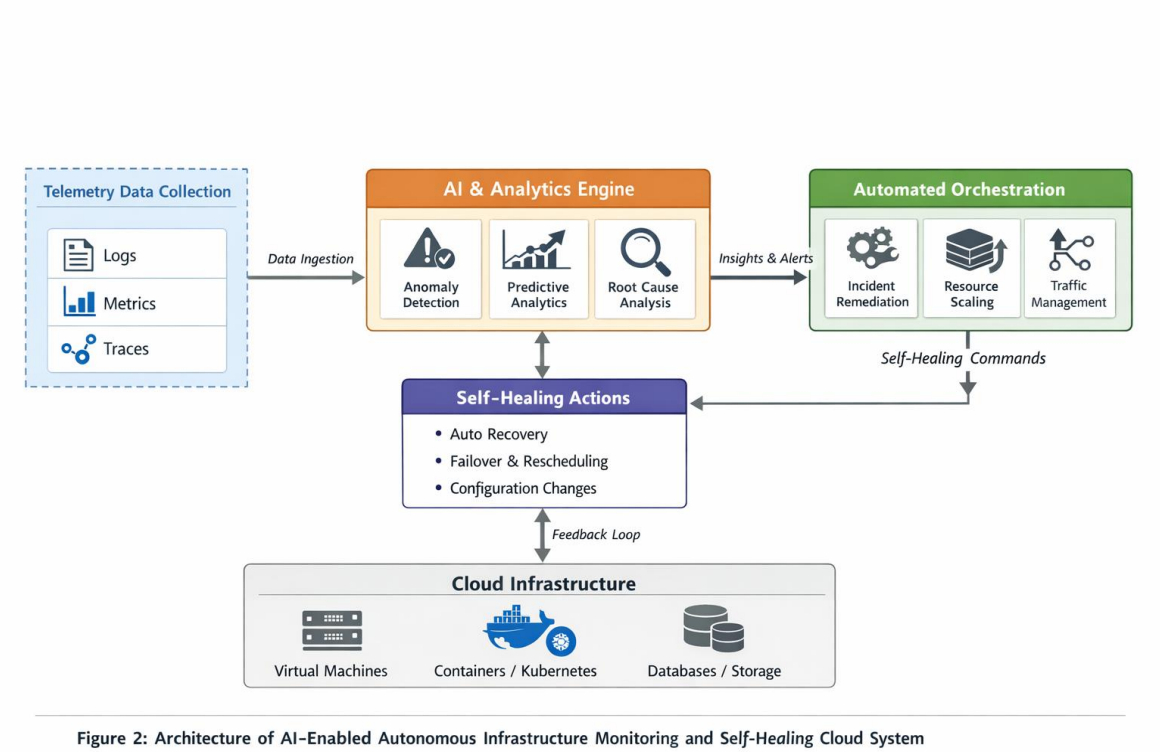


Figure 2: Architecture of AI-Enabled Autonomous Infrastructure Monitoring and Self-Healing Cloud System

Fig. 2. Architecture of AI-Enabled Autonomous Infrastructure Monitoring and Self-Healing Cloud System

3.1 Telemetry Data Collection Layer

This layer is responsible for collecting real-time data from various components of the cloud environment. It forms the foundation of the entire monitoring system.

Key Data Sources:

- **Logs:** Application logs, system logs, security logs
- **Metrics:** CPU, memory, disk I/O, network latency

- **Traces:** Distributed request flows across microservices

Technologies & Mechanisms:

- Agents and sidecars deployed across nodes
- API-based data collectors
- Streaming pipelines for continuous ingestion

Key Characteristics:

- High-volume and high-velocity data handling
- Support for structured and unstructured data
- Real-time streaming capabilities

3.2 AI & Analytics Engine

This layer is the core intelligence unit of the system. It processes telemetry data to derive insights and detect anomalies.

Core Functionalities:

- **Anomaly Detection** using unsupervised learning
- **Predictive Analytics** using time-series forecasting
- **Root Cause Analysis (RCA)** through correlation models
- **Noise Reduction** by filtering redundant alerts

Model Types Used:

- Statistical models (moving averages, ARIMA)
- Machine learning models (clustering, classification)
- Deep learning models (LSTM for time-series analysis)

Outcome:

- Generates actionable insights and alerts for downstream systems

3.3 Automated Orchestration Layer

This layer translates insights into actionable decisions. It acts as the control plane for automation.

Key Responsibilities:

- Policy-based decision making
- Workflow orchestration
- Integration with cloud-native tools

Typical Actions:

- Trigger scaling operations
- Initiate incident remediation workflows
- Redirect traffic in case of failures

Integration Points:

- Container orchestration platforms
- CI/CD pipelines
- Infrastructure-as-Code (IaC) tools

3.4 Self-Healing Execution Layer

This layer executes automated remediation actions based on decisions made by the orchestration layer.

Self-Healing Actions Include:

- **Auto Restart:** Restart failed services or containers
- **Failover:** Switch to backup nodes or regions
- **Auto Scaling:** Adjust compute resources dynamically
- **Configuration Fixes:** Rollback or patch faulty configurations

Key Features:

- Fully automated execution
- Minimal human intervention
- Fast recovery time (low MTTR)

3.5 Cloud Infrastructure Layer

This is the underlying environment where applications and services are deployed.

Components:

- Virtual Machines (VMs)
- Containers and orchestration platforms
- Databases and storage systems
- Networking components

Characteristics:

- Distributed and scalable
- Multi-cloud or hybrid deployment
- High availability and fault tolerance

3.6 End-to-End Workflow

The interaction between these layers follows a continuous pipeline:

1. **Data Collection** from infrastructure components
2. **Data Processing & Analysis** using AI models
3. **Insight Generation** and anomaly detection
4. **Decision Making** via orchestration policies
5. **Automated Execution** of corrective actions

6. **Feedback Loop** for continuous learning

Table: Mapping Architecture Layers to Functional Capabilities

Layer	Primary Function	Key Benefit
Telemetry Collection	Data ingestion	Real-time visibility
AI & Analytics Engine	Insight generation	Predictive intelligence
Orchestration Layer	Decision-making	Automated control
Self-Healing Execution	Action implementation	Reduced downtime
Cloud Infrastructure	Service hosting	Scalable environment

This architecture enables organizations to transition from reactive monitoring to fully autonomous, intelligent operations capable of handling complex cloud environments at scale.

4. DESIGN PATTERNS AND IMPLEMENTATION STRATEGIES

Designing AI-enabled autonomous infrastructure monitoring and self-healing systems requires a combination of architectural best practices, intelligent workflows, and scalable implementation models. This section presents key design patterns and practical strategies that enterprises can adopt to successfully implement such systems in real-world cloud environments.

4.1 Event-Driven Architecture Pattern

An event-driven architecture (EDA) is fundamental to enabling real-time responsiveness in autonomous systems. In this model, system components communicate through events, allowing asynchronous and loosely coupled interactions.

Key Characteristics:

- Producers generate telemetry events (logs, metrics, traces)
- Event streams are processed in real time
- Consumers (AI models, orchestration engines) react to events

Benefits:

- High scalability and flexibility
- Real-time anomaly detection
- Decoupled system components

4.2 Closed-Loop Automation Pattern

This pattern is central to self-healing systems, ensuring continuous monitoring, decision-making, and execution.

Workflow:

1. Detect anomaly
2. Analyze root cause
3. Decide corrective action
4. Execute remediation
5. Learn from outcome

This loop ensures that the system continuously improves its response accuracy over time.

4.3 Microservices-Based Monitoring Pattern

Modern cloud systems rely on microservices, and monitoring must align with this architecture.

Key Aspects:

- Each service emits its own telemetry data
- Distributed tracing tracks inter-service communication
- Service-level anomaly detection improves granularity

Advantages:

- Fine-grained monitoring
- Faster fault isolation
- Independent scalability

Figure: Closed-Loop Autonomous Self-Healing Design Pattern

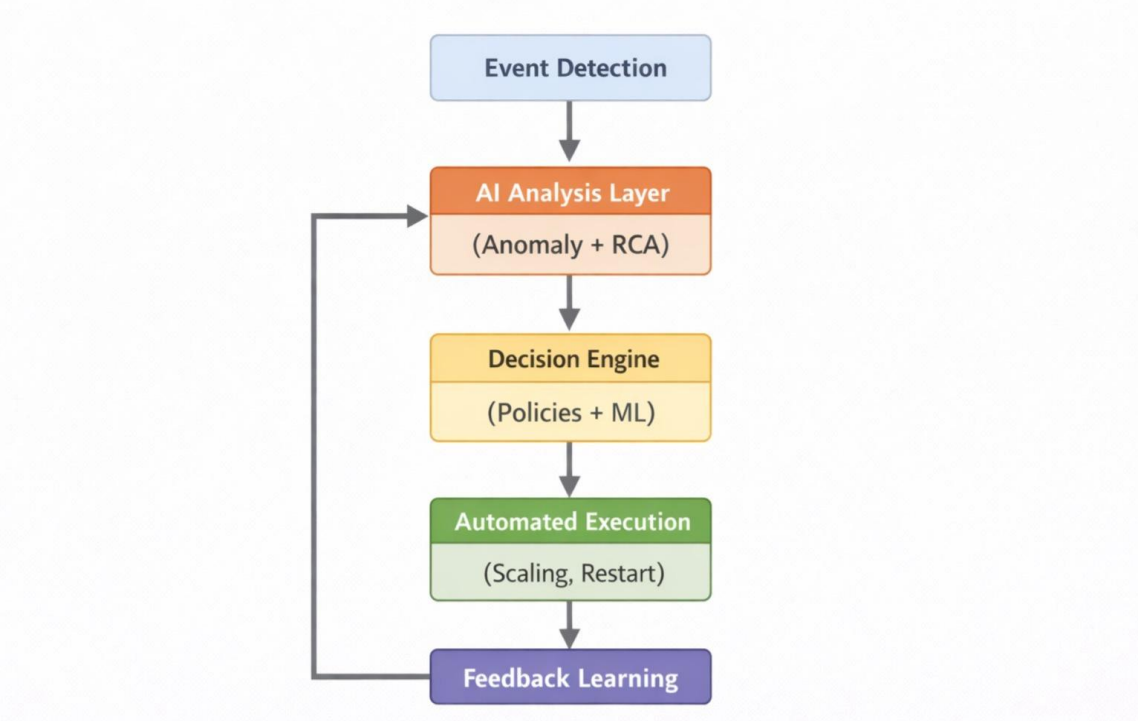


Fig. 3. Closed-Loop Autonomous Self-Healing Design Pattern

4.4 AI Model Deployment Strategy

Deploying AI models effectively is critical for real-time monitoring.

Strategies:

- **Centralized Models:** Deployed in a central analytics platform
- **Edge Models:** Deployed closer to data sources for low latency
- **Hybrid Approach:** Combination of both for optimal performance

Considerations:

- Latency requirements
- Data volume and distribution
- Model retraining frequency

4.5 Policy-Driven Automation

Automation decisions should be governed by policies to ensure control and compliance.

Types of Policies:

- Threshold-based policies (e.g., CPU > 80%)
- ML-based dynamic policies
- Business-rule-driven policies

Example:

- If anomaly confidence > 90%, trigger auto-remediation
- If confidence < 70%, escalate to human operator

4.6 Infrastructure as Code (IaC) Integration

IaC enables automated provisioning and configuration management, which is essential for self-healing systems.

Tools & Concepts:

- Declarative infrastructure definitions
- Version-controlled configurations
- Automated rollback mechanisms

Benefits:

- Consistency across environments
- Faster recovery through redeployment
- Reduced configuration drift

4.7 Implementation Strategy for Enterprises

A phased approach is recommended for adopting autonomous monitoring systems:

Phase 1: Observability Maturity

- Implement centralized logging, metrics, and tracing

- Establish baseline monitoring

Phase 2: AI Enablement

- Introduce anomaly detection models
- Train models on historical data

Phase 3: Automation Integration

- Implement orchestration workflows
- Enable semi-automated remediation

Phase 4: Full Autonomy

- Deploy closed-loop self-healing systems
- Continuously optimize models and policies

Table: Design Patterns vs Use Cases

Design Pattern	Use Case	Outcome
Event-Driven Architecture	Real-time anomaly detection	Faster response
Closed-Loop Automation	Self-healing systems	Continuous improvement
Microservices Monitoring	Distributed applications	Fine-grained visibility
AI Deployment Strategy	Large-scale environments	Optimized performance
Policy-Driven Automation	Controlled remediation	Risk mitigation
IaC Integration	Infrastructure recovery	Consistency and speed

By combining these design patterns and strategies, organizations can build scalable, intelligent, and resilient monitoring systems capable of autonomously managing complex cloud environments.

5. CHALLENGES AND LIMITATIONS

While AI-enabled autonomous infrastructure monitoring and self-healing systems offer significant advantages, their implementation in enterprise environments is not without challenges. These systems must operate in highly dynamic, large-scale, and heterogeneous environments, which introduces technical, operational, and organizational complexities. This section outlines the key challenges along with practical mitigation strategies.

5.1 Data Quality and Noise

AI models rely heavily on telemetry data (logs, metrics, traces). However, in real-world systems:

- Data may be incomplete, inconsistent, or noisy
- Duplicate alerts and irrelevant logs can overwhelm models

- Poor data quality leads to inaccurate predictions

Impact:

- Increased false positives/negatives
- Reduced trust in AI-driven decisions

Mitigation Strategies:

- Implement data preprocessing and normalization pipelines
- Use filtering and deduplication techniques
- Establish data governance standards

5.2 Model Drift and Accuracy Degradation

Cloud environments are dynamic, with frequent changes in workloads, configurations, and deployments. Over time:

- ML models may become outdated (model drift)
- Predictions lose accuracy as system behavior evolves

Impact:

- Incorrect anomaly detection
- Ineffective remediation actions

Mitigation Strategies:

- Continuous model retraining using recent data
- Implement adaptive learning mechanisms
- Monitor model performance metrics regularly

5.3 Explainability and Trust

AI models, especially deep learning models, often function as "black boxes."

Challenges:

- Difficulty in explaining why a decision was made
- Lack of transparency reduces operator trust
- Compliance and audit requirements may not be met

Mitigation Strategies:

- Use explainable AI (XAI) techniques
- Provide confidence scores and decision logs
- Maintain audit trails for all automated actions

5.4 Integration with Legacy Systems

Many enterprises operate hybrid environments with legacy systems that:

- Lack modern observability capabilities
- Do not support automation APIs

- Are tightly coupled and difficult to modify

Impact:

- Limited visibility into older systems
- Incomplete automation coverage

Mitigation Strategies:

- Use adapters and middleware for integration
- Gradually modernize legacy components
- Implement hybrid monitoring strategies

5.5 Automation Risks and Over-Correction

While automation reduces manual effort, it introduces risks:

- Incorrect remediation actions can worsen issues
- Cascading failures due to aggressive automation
- Lack of human oversight in critical scenarios

Mitigation Strategies:

- Implement policy-based controls and thresholds
- Use "human-in-the-loop" for critical decisions
- Introduce staged automation (semi → full automation)

5.6 Scalability and Performance Overhead

AI-driven monitoring systems must process massive volumes of data in real time.

Challenges:

- High computational cost of ML models
- Latency in real-time decision-making
- Infrastructure overhead for analytics pipelines

Mitigation Strategies:

- Use distributed processing frameworks
- Optimize models for real-time inference
- Adopt edge computing for low-latency analysis

5.7 Security and Compliance Concerns

Autonomous systems interact deeply with infrastructure, making them a potential attack surface.

Risks:

- Unauthorized automated actions
- Data privacy concerns in telemetry collection
- Compliance violations in regulated industries

Mitigation Strategies:

- Implement strong access control and authentication
- Encrypt telemetry data in transit and at rest
- Align with regulatory frameworks and compliance standards

Table: Challenges vs Mitigation Strategies

Challenge	Impact	Mitigation Strategy
Data Quality Issues	Poor model accuracy	Data preprocessing and governance
Model Drift	Degraded predictions	Continuous retraining
Lack of Explainability	Reduced trust	Explainable AI techniques
Legacy System Integration	Limited automation	Middleware and phased modernization
Automation Risks	System instability	Policy-based controls, human oversight
Scalability Constraints	Performance bottlenecks	Distributed and optimized processing
Security Concerns	Vulnerabilities and compliance issues	Strong security and compliance measures

5.8 Key Takeaway

Despite these challenges, a well-designed architecture combined with robust governance, continuous learning, and controlled automation can significantly mitigate risks. Organizations should adopt a phased and strategic approach to ensure successful implementation and long-term sustainability of AI-driven autonomous systems.

6. FUTURE TRENDS AND INNOVATIONS

As cloud ecosystems continue to evolve, AI-enabled autonomous monitoring and self-healing systems are expected to become more intelligent, adaptive, and deeply integrated into enterprise IT operations. This section highlights emerging trends and innovations that will shape the next generation of autonomous cloud systems.

6.1 Hyper-Automation in Cloud Operations

Hyper-automation extends beyond traditional automation by combining AI, machine learning, and robotic process automation (RPA) to automate end-to-end workflows.

Key Advancements:

- Fully autonomous incident management
- Cross-domain automation (network, application, infrastructure)
- Intelligent workflow orchestration

Impact:

- Near-zero human intervention

- Increased operational efficiency

6.2 Predictive and Prescriptive Analytics

Future systems will not only predict failures but also recommend optimal corrective actions.

Capabilities:

- Failure prediction using advanced time-series models
- Prescriptive recommendations based on historical outcomes
- Scenario simulation for proactive planning

Example:

- Predicting server overload and proactively redistributing workloads

6.3 Integration with Edge and Multi-Cloud Environments

With the rise of edge computing and multi-cloud strategies:

- Monitoring systems must operate across geographically distributed environments
- AI models will be deployed at the edge for low-latency decision-making
- Unified observability across clouds will become essential

6.4 Self-Learning and Adaptive Systems

Future systems will continuously learn and adapt without explicit retraining cycles.

Innovations:

- Online learning models
- Reinforcement learning for dynamic decision-making
- Context-aware anomaly detection

Outcome:

- Improved accuracy over time
- Reduced dependency on manual tuning

6.5 Explainable and Trustworthy AI

As enterprises demand transparency:

- AI decisions will become more interpretable
- Systems will provide reasoning behind automated actions
- Compliance-ready AI frameworks will emerge

6.6 Digital Twins for Infrastructure Simulation

Digital twins—virtual replicas of physical infrastructure—will enable:

- Simulation of failure scenarios
- Testing of remediation strategies before execution

- Risk-free optimization of system configurations

6.7 Autonomous Security Integration

Security will become an integral part of self-healing systems:

- AI-driven threat detection and response
- Automated patching and vulnerability remediation
- Integration with DevSecOps pipelines

Table: Future Trends and Expected Impact

Trend	Description	Impact
Hyper-Automation	End-to-end intelligent automation	Reduced manual effort
Predictive Analytics	Forecasting failures	Proactive operations
Edge & Multi-Cloud	Distributed monitoring	Scalability and flexibility
Self-Learning Systems	Continuous adaptation	Improved accuracy
Explainable AI	Transparent decision-making	Increased trust
Digital Twins	Virtual system simulation	Risk-free optimization
Autonomous Security	Integrated security automation	Enhanced resilience

7. CONCLUSION

AI-enabled autonomous infrastructure monitoring and self-healing cloud systems represent a paradigm shift in how modern IT environments are managed. As cloud infrastructures grow increasingly complex and distributed, traditional monitoring approaches are no longer sufficient to ensure high availability, performance, and reliability.

This paper explored the architectural foundations, core concepts, design patterns, and implementation strategies that enable intelligent and autonomous cloud operations. By leveraging machine learning, real-time analytics, and automated orchestration, organizations can transition from reactive incident management to proactive and predictive operations.

The integration of self-healing capabilities significantly reduces downtime, minimizes operational overhead, and enhances system resilience. However, successful adoption requires addressing challenges such as data quality, model drift, explainability, and integration with legacy systems. A phased implementation approach, combined with strong governance and continuous learning, is essential for long-term success.

Looking ahead, advancements in hyper-automation, edge computing, explainable AI, and digital twins will further accelerate the evolution of autonomous systems. These innovations will enable enterprises to build highly resilient, adaptive, and intelligent cloud environments capable of self-managing at scale.

In conclusion, AI-driven autonomous monitoring and self-healing systems are not just an enhancement but a necessity for future-ready cloud infrastructures, empowering organizations to achieve operational excellence and sustained digital transformation.

REFERENCES

[1] R. Kumar and S. Patel, "Self-Healing Systems in Distributed Cloud Environments," IEEE Software, vol. 42, no. 2, pp. 70-79, 2025.

[2] L. Zhang et al., "Autonomous Monitoring Using Machine Learning in Multi-Cloud Systems," Future Generation Computer Systems, vol. 145, pp. 210-225, 2025.

[3] A. Singh and P. Rao, "AIOps: Enhancing IT Operations with Artificial Intelligence," Journal of Systems Architecture, vol. 140, pp. 102-118, 2024.

[4] T. Nguyen et al., "Anomaly Detection in Cloud Systems Using Deep Learning," IEEE Access, vol. 12, pp. 55678-55692, 2024.

[5] K. Sharma and D. Verma, "Predictive Analytics for Cloud Infrastructure Failures," ACM Computing Surveys, vol. 56, no. 3, pp. 1-34, 2023.

[6] J. Lee et al., "Closed-Loop Automation in Cloud Operations: A Survey," IEEE Communications Surveys & Tutorials, vol. 25, no. 4, pp. 1890-1912, 2023.

[7] S. Gupta and M. Iyer, "Observability in Modern Cloud-Native Systems," IEEE Internet Computing, vol. 27, no. 5, pp. 60-69, 2023.

Citation: Rajesh Adepu. (2025). AI-Enabled Autonomous Infrastructure Monitoring and Self-Healing Cloud Systems. International Journal of Artificial Intelligence & Machine Learning (IJAIML), 4(1), 234-251.

Abstract Link: https://iaeme.com/Home/article_id/IJAIML_04_01_017

Article Link:
https://iaeme.com/MasterAdmin/Journal_uploads/IJAIML/VOLUME_4_ISSUE_1/IJAIML_04_01_017.pdf

Copyright: © 2025 Authors. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Creative Commons license: Creative Commons license: CC BY 4.0

✉ **editor@iaeme.com**