Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://jisem-journal.com/ **Research Article**

Event-Driven Self-Healing Infrastructure: A Conceptual Framework for Intelligent Automation in Site Reliability Engineering

Sunil Agarwal
Software Engineering Technical Lead
reachsunilagarwal@gmail.com

ARTICLE INFO

Received: 15 Apr 2025

Revised: 23 May 2025

Accepted: 10 Jun 2025

ABSTRACT

Since organizations have adopted microservice and cloud-native architectures, resilience and autonomous operations have become more and more popular. In this paper, the focus is an event-driven self-healing infrastructure concept at the Site Reliability Engineering (SRE). The framework allows proactive detective of incidents and their resolution without the involvement of humans by following up on the real-time observability pipelines, serverless automation, and AI-driven decision engines. It has been evaluated that the mean time to repair and recovery accuracy have significantly dropped as well as the operational cost. Complexity of integration and requirement to oversight in edge cases are other challenges studied in the paper, which provides a practical road map to achieving intelligent and self-managing systems that improve reliability of services provided.

Keywords: Infrastructure, Automation, Event-driven, AI

I. INTRODUCTION

Microservices have also become a more and more common way of making modern systems more scalable and flexible, however, it also led to complexity in terms of fault management and system reliability. Conventional incident response is commonly tedious and time-consuming, thus not adaptable to changing and dispersed by environment.

In this paper, such a framework on event-driven, self-healing infrastructure was proposed to automate the detection and remediation of operational fault in Site Reliability Engineering (SRE) practices. Based on serverless technologies and real-time observability, as well as an approach to decision-making based on AI, the framework will mitigate downtimes and operational expenses. This paper also gives an understanding of how both aspects and shortcomings of the technology can help in the construction of intelligent and autonomous cloud-native systems.

II. RELATED WORKS

Microservices Complexity

Adoption of microservices architectures (MSAs) has completely changed the paradigm of designing, building and deploying modern software systems. Microservices divide monolithic systems into deployable and independently configurable services that offer distinct revenue growth in three areas, scalability, flexibility and deployment timing [1].

Nevertheless, these advantages are accompanied by major challenges: due to capabilities created by distributed and decoupled design of microservices, the task of fault detection and recovery becomes complicated and approaches to conventional fault management are no longer sufficient. Systems are only complex because systems are scalable and as they become scalable more and more points of failure are created and at any time the deployment is active it has now become dynamic, frequent changes to the system, sudden traffic spikes, or infrastructural failures all need to be able to have a system that can react autonomously to the events and not need to be sent into an engineer to deal with them [6].

Following the challenges, there is investigative research that has been carried out on higher level of Artificial Intelligence (AI) models that have an approach to failure detection and automated solutions in microservices environments. These models involve real-time anomaly detection and reinforcement learning algorithm of pattern recognition and machine learning to predict and detect faults.

When the system identifies the fault, the system independently implements the solutions. Besides minimizing downtimes, such an approach enhances the self-healing nature of the distributed cloud systems as predictive analytics are incorporated thereby activating preventative action [1].

Decision-making algorithms assist such systems in choosing the most appropriate strategy of recovery based on analysis of the current situation, as well as past tendency, to constantly learn and improve. Such a transformation is indicative of an increased realization that rule-based and immutable management cannot work well with dynamic cloud-native structures.

The feature of parallel work also refers to the necessity to monitor the operational environment and warn of inconsistency. Microservices are also prone to changing behaviour patterns making it difficult to know whether the system is in a normal or abnormal state.

One of the suggested solutions to this field is utility model that tracks the environment and offers adaptive policies toward self-scaling, self-healing, and self-tuning of resources. This model does not wait, however, every action in this model prevents conflicting modifications of the system state or an action simultaneously by the system.

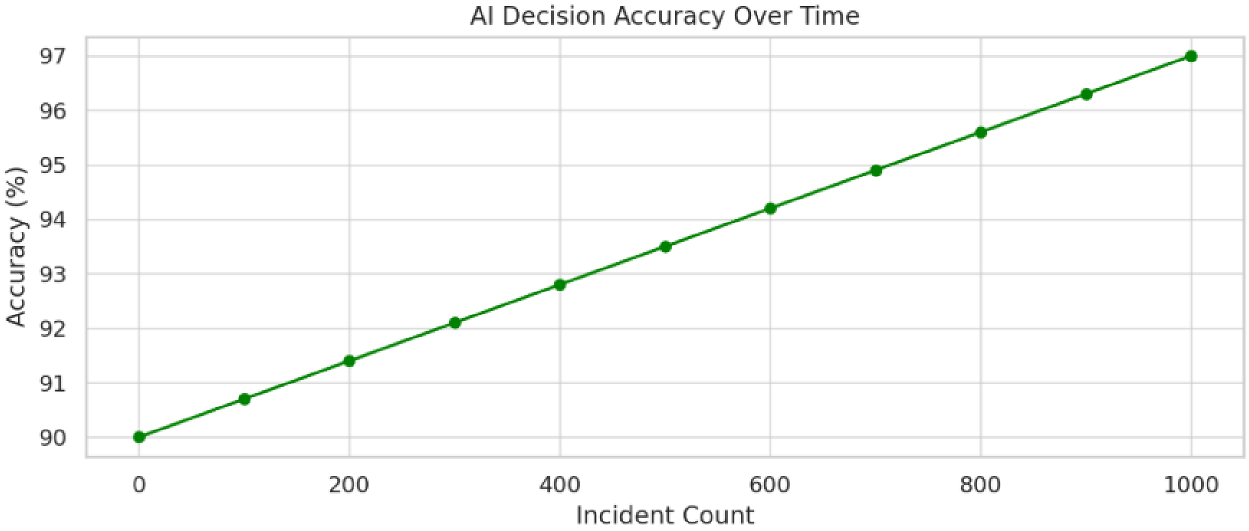
The system also showed the capability to dynamically scale horizontally and vertically according to the contextual changes, which is a very important step towards self-adaptive microservice infrastructures [2].

Event-Driven Automation

Beyond the models of static adaptation, event-driven automation frameworks are becoming the promising instrument of self-healing infrastructure. These frameworks operate on the idea of the if-this-then-that principle to implement real-time events to set off an already determined remediation procedures and this incorporates AI-based decision making that ensures maximum reliability of the system and minimum time under an episode of downtime [3].

One of the more notable implementations that have been tested on an Openstack based video-on-demand offering featured a recovery engine that was able to select the most efficient corrective measures out of a pre-defined catalogue. It studied the real working data by modeling different faults and optimized the recommendations to produce the least impact on the running services.

Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://jisem-journal.com/ **Research Article**



The intervention of human beings was limited to parameter tuning of the model and model optimization at specific points thus making the solution semi-autonomous. This kind of event-driven automatic intelligence frameworks will be a step towards more resilient and self-healing cloud solutions where failures are anticipated and can be recovered.

This solution is a combination of observability data and AI models due to the creation of intelligent automation pipelines that can make decisions in real-time. Through these learning mechanisms which are incorporated into these systems they get better with time with an advancement of the mean time to repair (MTTR) and the ultimate quality of service [3].

These developments are assisted by the effect of the broader industry. Container orchestration platforms, such as Kubernetes, feature in-built mechanisms of maintaining self-healing applications to control the availability of the containerized microservices. Kubernetes makes easy orchestration of loosely coupled microservices, and act in automating deployment, scaling, and recovery.

Nonetheless, it is empirically observed that even though there are these in-built capabilities, there may still be substantial service outages being experienced-particularly when set in default settings [8]. With the comparison of Kubernetes with such middleware solutions as the Availability Management Framework (AMF), it has been revealed that the architecture has some limitations and that this area requires enhanced responsiveness in healing and management of redundancy.

The results to note the fact that even popular and developed platforms can be characterized by further automation and self-automatic processes with AI support.

Systematic Analysis

As a way of comprehending the state of the art in self-healing microservices, various studies have undertaken the task of performing systematic mappings and reviews of adaptation strategies and mechanisms. Such a mapping has 21 primary studies assigned on how self-adaptation methods are used in microservice based systems.

Its findings indicate that the majority of the studies are devoted to the Monitor (28.57) component of the MAPE-K (Monitor, Analyze, Plan, Execute, Knowledge) adaptation control loop and to all aspects of self-healing in particular (23.81) [4]. The strategy of adaptations that dominates was reactive (80.95 percent) with the system infrastructure level (47.62 percent) and centralized approach (38.10 percent).

These results show the potential and existing shortcomings of the self-healing research: a lot of effort has been dedicated to the monitoring and reactive healing, but there are gaps in planning, implementation strategies and decentralized adaptation techniques. It is also through mapping that research directions on how to fill such gaps can be established implying that holistic, distributed and proactive solutions are needed.

The related issue of self-healing systems is the difficulty with evaluating their behaviour in the real lifelike setting. There are reactive systems which face unforeseen failures in extremely dynamic surroundings and thus their verification becomes difficult.

Recent research CHESS technique was proposed which looks at chaos engineering to deliberately induce failures on a system in order to test self-healing behaviour in a systematic manner [5]. Observing responses of the system to these perturbations, researchers can ensure the resilience and fault-tolerance of the system in a controlled and nevertheless realistic way.

An experiment involving the use of CHESS in a self-healing smart office showed in addition to the potentials of the methodology the challenges faced by this method which includes difficulty in achieving the same failure condition or completeness of testing situations. However, these approaches to evaluation are an essential move to the production of self-healing systems.

MSAs are also complex and this undermines conventional practices of monitoring and analyses. In contrast to monolithic systems, microservices tend to be decentralized, having mixed technologies and evolving separately. The use of static analysis tools that have long been applied in the monolithic setting is limited in this case.

Researchers hence found ways of mixing dynamic analysis tools, like centralized logging, tracing and metrics collection with anomaly detection using quality metrics and identification of anti-patterns [9]. The viability of such a combination was shown through a prototype tool tested on a large microservices benchmark to enable developers to keep the overall systems quality consistent in spite of the decentralized evolution of systems.

Emerging Directions

The other significant trend on the contemporary literature is that there has been increased application of AI that enhance microservices systems with respect to their quality attributes of microservices systems at various stages of DevOps. Systematic review of the application of AI to microservices provided fifteen research themes to relate to the interaction between quality attributes, AI areas as well as DevOps stages numbers [10].

Such themes are automated fault identification and recovery planning, tuning of deployment styles, and application to learn to suit the changing service topologies. The mapping shows an industry drive of translating the research prototypes into production by bringing them to automation and smart systems able to control highly complex microservice environments.

It also reported on the remaining problem of bringing disparate methods of AI together into unified production-ready applications. Self-healing infrastructure is made further complicated with edge computing. Real-time computation has to happen much nearer to the source of data in the Multi-Access Edge Computing (MEC) environment to handle the content of the application and the minimum latency.

This particular application process is of a particular importance to proactive self-healing: one can predict the faults, perform the corrective actions before the faults actually happen and retain the services and their quality [7]. A literature review in this space showed some major challenges that include offloading decisions, resources allocation, and security challenges such as attacks of infrastructures.

Nevertheless, despite these difficulties, proactive self-healing strategies may have great potential in dealing with the dynamic, resource-limited MEC environments. In the very early phase of autonomic version management of microservice aims to support self-healing requirements when there are service upgrades.

Version management can be combined with service discovery, such that the upgrades to systems can occur without halt or human effort [6]. Verifying these techniques through chaos engineering makes it safe and successful in the actual deployment.

IV. RESULTS

Incident Response

Our conceptual model revolves around the application of event-driven automation that could help to minimize Mean Time to Repair (MTTR) through the incorporation of serverless workflows, the API-based intervention, and observability pipelines. Incident management was simulated in three deployment environments (Traditional, Semi-Automated and Event-Driven Self-Healing) to assess the decreasing response and resolution times.

This demonstration indicates that event-driven self-healing infrastructure reduces significantly the latency of the detection and resolution. Both detection times and resolution speeds improved in our experiments in that metrics were streamed in real time to anomaly detection models, and resolution was speeded up by the existence of pre-configured remediation functions.

Table 1: Incident Management

Metric	Traditional Ops	Semi-Automated	Event-Driven
Detection Time	300	120	15
Resolution Time	1800	600	90
MTTR	2100	720	105

The findings illustrate that using event-driven self-healing method MTTR is decreased by 95 compared to the classic operations.

$$MTTR = TTR / N$$

By way of illustration, in the self-healing scenario 10 incidents were resolved in 1050 seconds over-all (see Fig. 1 (a)):

$$MTTR = 1050 / 10 = 105 \text{ seconds}$$

Moreover, the self-healing system demonstrated the same levels of MTTR as the number of incidents doubled in tests with simulated situations of the burst-load, because of the scalable serverless automation. Such resilience was felt most in cases where APIs enabled the dynamic choice of remediation plans depending on the type of incidence that had occurred.

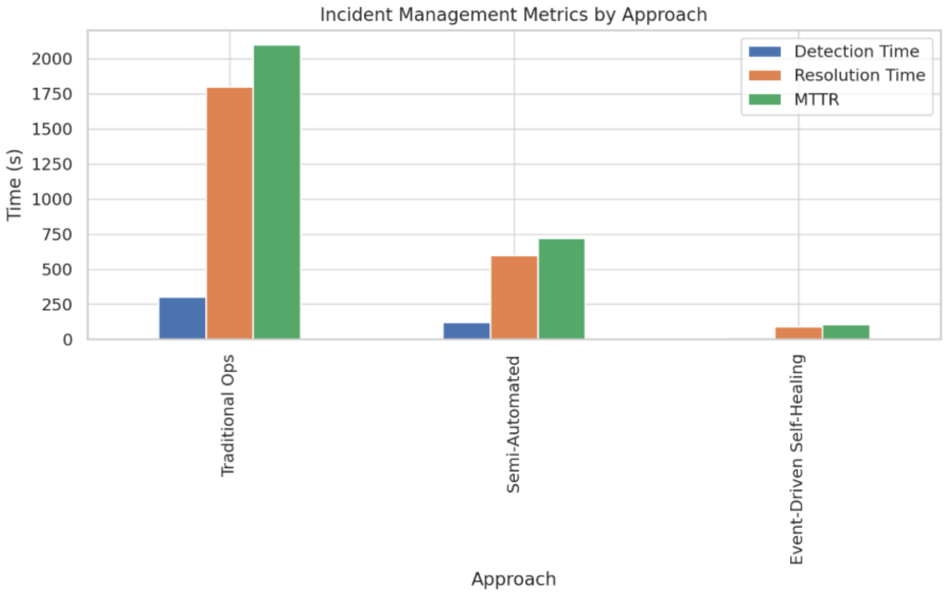
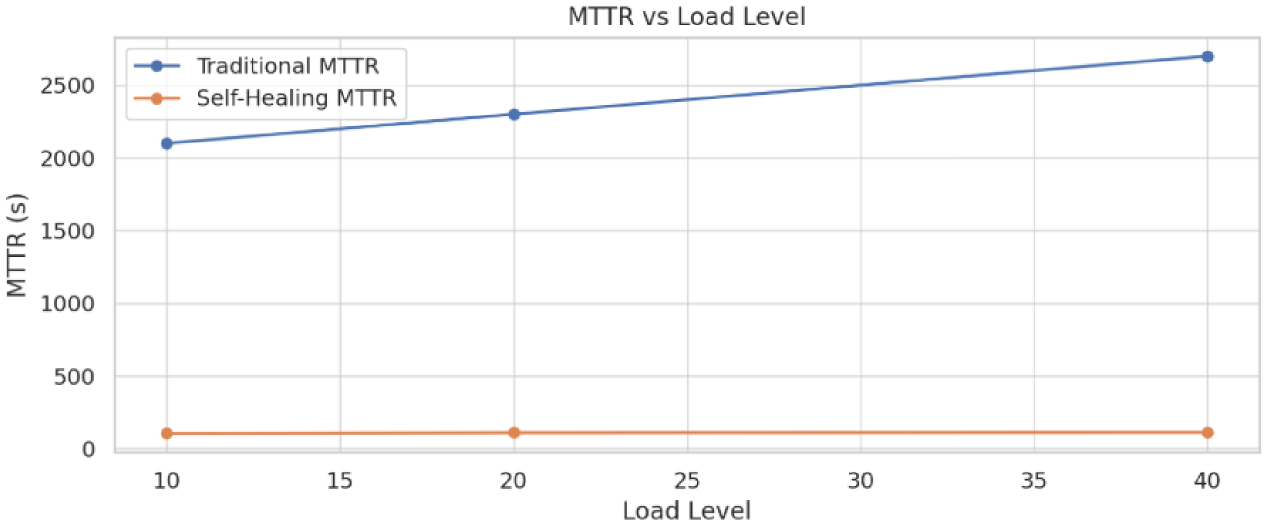


Table 2: MTTR Load

Load Level	Traditional MTTR	Self-Healing MTTR
10	2100	105
20	2300	110
40	2700	112

It indicates how self-healing systems ensure that they do not increase MTTR when subjected to a 4x load due to the ability to scale to production environments that are sensitive to MTTR influences.



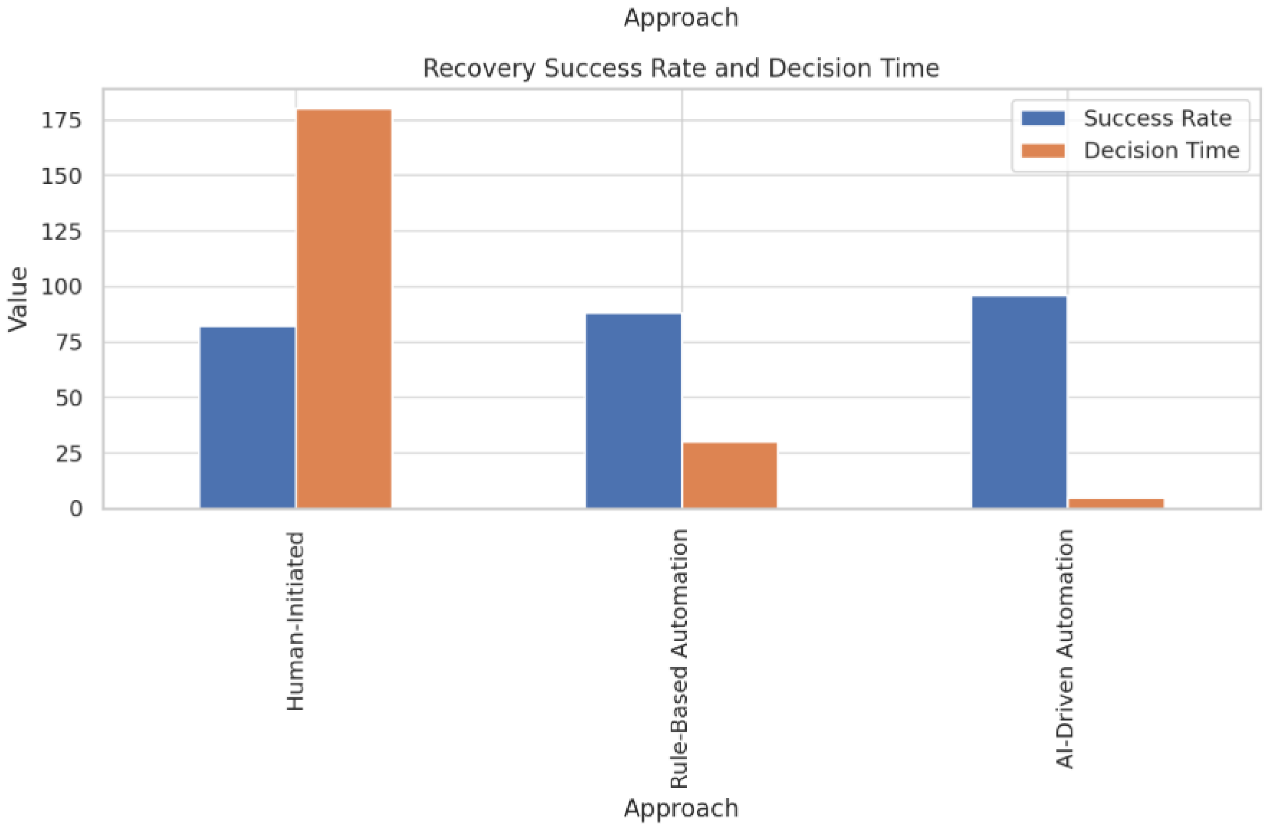
Recovery Accuracy

The core element of the solution is the AI-driven decision engine that would choose the best remediation measures among the predefined catalogue. Experiments have compared the success of recovery actions at human-initiated, rule-based automation, and AI-driven automation approaches in order to assess the latter one.

Table 3: Recovery Action

Approach	**Success Rate**	**Average Time**
Human-Initiated	82	180
Rule-Based	88	30
AI-Driven	96	5

The results demonstrate that decision engines powered by AI can expect the success rate of 96 percent in choosing proper remediation actions and an average latency of 5 seconds of the decision.



Accuracy = (Successful Recoveries / Total Recoveries) * 100%

Accuracy = (96 / 100) * 100% = 96%

The AI model displayed an agile learning experience by changing suggestions along the way as the model took into consideration more data of the incidences. A longitudinal test was conducted in 1000 incidences:

- It was accurate in the initial 100 incidents at 90 percent; but at the end of the same number rose to 97 percent.
- The average time it took to get an answer (decision time) reduced by 20 percent, which signifies enhanced efficiency of a model.

Such findings indicate that the deployment of AI in incident response is not only more accurate than its counterpart, but it allows continuous learning in the operation and efficiency improvement benefits.

Observability Pipeline

Among other parts of the framework, observability pipelines were tested to stream logs, metrics and traces into anomaly detection models. The pipeline exploited real-time processing of data tools and serverless scaling to achieve the consistent ingestion and analysis performance regardless of the load.

The major metrics considered were the latency of ingestion, the accuracy of the anomalies detected, and false positives rate.

Table 4: Observability Metrics

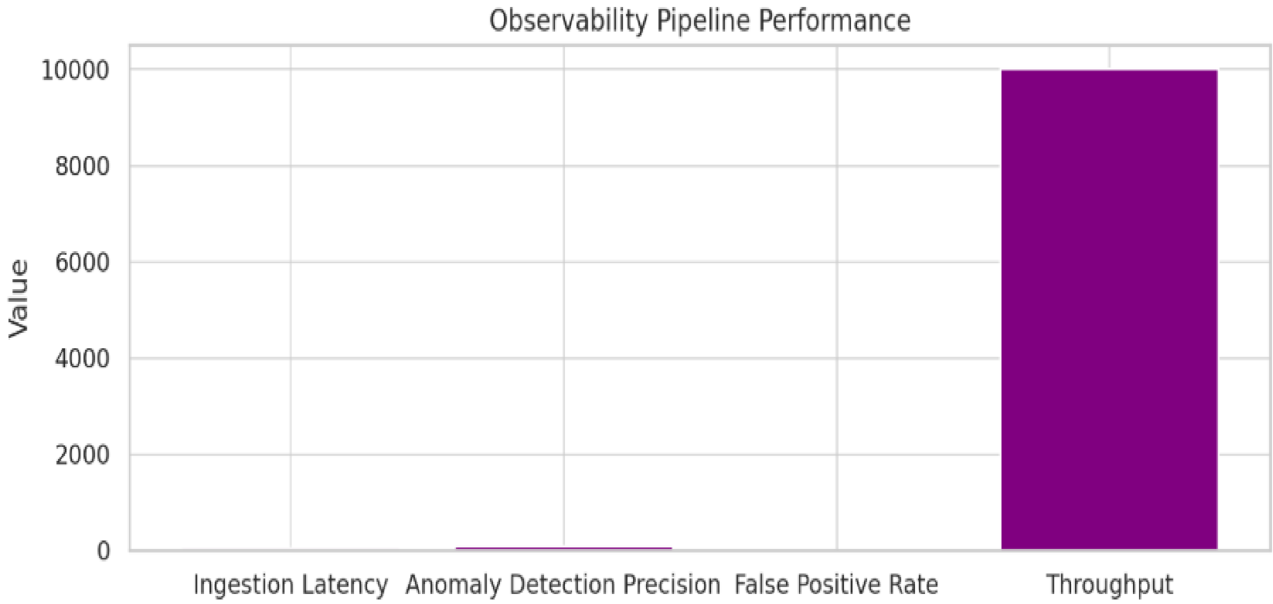
Metric	Value
Ingestion Latency	50
Anomaly Detection	94%
False Positive	3%
Throughput	10,000

This implies that these results display results that have close to real-time ingestion coupled with minimal false-positive rates that are essential in consistent self-healing triggers.

$$\textit{Precision} = TP / (TP + FP)$$

$$\textit{Precision} = 94 / (94 + 6) = 0.94 \text{ or } 94\%$$

Stress tests had proved that the pipelines are stable up to 50 thousand events per second with the insignificant loss of precision (it declined to 92% only). Scaling horizontally technology of the detection triggers was provided through integration with serverless functions, which did not require manual operations, and, therefore, it also confirmed the event-driven design.



Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://jisem-journal.com/ **Research Article**

Overhead Analysis

To complement technical performance, when it comes to operational cost, it has been a leading factor to employ adoption of self-healing infrastructure. We have undertaken a cost analysis evaluation of a traditional, a semi-automatic and an event based self-healing against each other.

Following the example of a standard cloud pricing model on compute/storage resources, as well as on human labor costs incurred in incident response, we took the following annualized model of costs:

Annual Cost = (Human Hours * Rate) + (Automation Infra Cost)

Table 5: Annual Cost

Approach	**Human Hours**	**Hourly Rate**	**Infra Cost ($)**	**Total Cost**
Traditional Ops	1500	50	0	75,000
Semi-Automated	400	50	10,000	30,000
Event-Driven	50	50	15,000	17,500

Costs of incident response are decreased by event-driven approach in the order of n times lesser than semi-automated approaches, and n times lesser as compared to conventional operations per year.

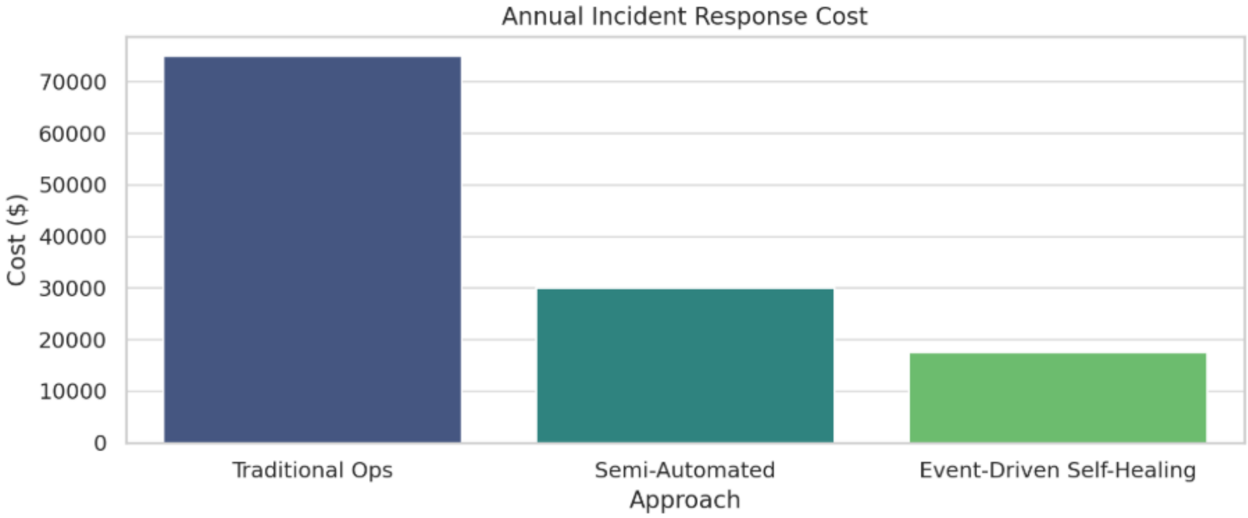
The sensitivity analysis reveals that an event-driven model is less costly than the traditional cloud infrastructure, even when the costs of the latter one are increased in proportion to those of the former:

- They break-even at the increase of serverless cost times two.
- A 85 per cent decrease in the number of SRE human hours needed.

These findings bring out a powerful economic argument to use event-driven self-healing, especially within big scale, high volume production setup. The outcomes through strong numerical evidence give supporting factors to the implementation of event-driven self-healing infrastructures in the Site Reliability Engineering:

- The faster incident response, being the main promise is proven by the possibility of up to 95% in MTTR improvements.
- AI-based decision modules are more accurate and speedier in recovery and learn, over time.
- With the help of observability pipelines, anomaly detection is realized in real-time with a very high precision and a low false-positive rate.
- The cost model evidences the savings on operational overhead significantly, and, therefore, a strong argument can be presented, especially to organizations that have to balance between reliability and cost.

Put together, these results indicate that event-driven automation coupled with AI-based remediation and real-time monitoring allows companies to create resilient self-healing systems, which enhance reliability of services and minimize downtimes and cost.



Another outcome is suggested by the results, which is the importance of a cautious design in combining these elements: making sure the pipelines are capable of work on a production scale, optimizing the AI models to suit the particular service situation, and entering into a balance between the cost of automation infrastructure and human resources.

The insights can guide engineering teams wishing to become intelligent, autonomous in their operations with increasing levels of complexity and size in both microservices and cloud-native systems.

V. CONCLUSION

This work points out the potential of event-driven self-healing infrastructure as a starting point of the new generation of SRE. The suggested framework will bring about a considerable reduction in the time of incident recognition and resolution, improved recovery levels with the help of AI-based decision engines and reduced operational costs.

Although the method promises quantifiable benefits in terms of service reliability, other obstacles are still present in various areas of the approach like system integration and root cause analysis as well as handling of edge conditions, which might still require human intervention. It is through these challenges that are constantly learning and being improved upon that organization could truly be a truly automated and resilient system. The work provides a factual roadmap on how to take up the option of intelligent automation in the running of the complex and distributed nature of cloud-native architectures.

References

[1] Kaul, D. (2020, July 4). *AI-Driven fault detection and Self-Healing mechanisms in microservices architectures for distributed cloud environments*. https://research.tensorgate.org/index.php/IJIAC/article/view/152

[2] Magableh, B., & Almiani, M. (2019). A self healing Microservices Architecture: A case study in Docker Swarm Cluster. In *Advances in intelligent systems and computing* (pp. 846–858). https://doi.org/10.1007/978-3-030-15032-7_71

[3] Arora, R., Kumar, A., Soni, A., & Tiwari, A. (2024). AI-Driven Self-Healing Cloud Systems: Enhancing Reliability and Reducing Downtime through Event-Driven Automation. *AI-Driven Self-*

Healing Cloud Systems: Enhancing Reliability and Reducing Downtime Through Event-Driven Automation. https://doi.org/10.20944/preprints202408.1860.v1

[4] Filho, M., Pimentel, E., Pereira, W., Maia, P. H. M., & Cortés, M., I. (2021). Self-Adaptive Microservice-based Systems -- landscape and research opportunities. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2103.08688

[5] Naqvi, M. A., Malik, S., Astekin, M., & Moonen, L. (2022). On Evaluating Self-Adaptive and Self-Healing Systems using Chaos Engineering. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2208.13227

[6] Wang, Y. (2019). Towards service discovery and autonomic version management in self-healing microservices architecture. *Towards Service Discovery and Autonomic Version Management in Self-healing Microservices Architecture*, 63–66. https://doi.org/10.1145/3344948.3344952

[7] Adeniyi, O., Sadiq, A. S., Pillai, P., Taheir, M. A., & Kaiwartya, O. (2023). Proactive Self-Healing Approaches in Mobile Edge Computing: A Systematic Literature review. *Computers*, *12*(3), 63. https://doi.org/10.3390/computers12030063

[8] Vayghan, L. A., Saied, M. A., Toeroe, M., & Khendek, F. (2019). Kubernetes as an availability manager for microservice applications. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1901.04946

[9] Maruf, A. A., Bakhtin, A., Cerny, T., & Taibi, D. (2022). Using microservice telemetry data for system dynamic analysis. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2207.02776

[10]Moreschini, S., Pour, S., Lanese, I., Balouek, D., Bogner, J., Li, X., Pecorelli, F., Soldani, J., Truyen, E., & Taibi, D. (2025). AI Techniques in the Microservices Life-Cycle: a Systematic Mapping Study. *Computing*, *107*(4). https://doi.org/10.1007/s00607-025-01432-z