

논문 2026-3-15 <http://dx.doi.org/10.29056/jsf.2026.09.15>

Towards an Autonomous Agent-based Cloud Service Architecture

Jaehun Song*, Jinhong Kim**†

Towards an Autonomous Agent-based Cloud Service Architecture

Jaehun Song*, Jinhong Kim**†

Abstract

The rapid advancement of large language models (LLMs) and multimodal artificial intelligence (AI) is transforming cloud computing and web services from traditional, manually controlled environments into proactive, collaborative ecosystems driven by autonomous agents. This paper introduces an Agentic Cloud Architecture—an autonomous agent-based framework tailored for web-based cloud service environments. The proposed architecture empowers agents to interpret complex user intents in natural language and autonomously execute end-to-end cloud operations, including resource management, web application deployment, and system recovery. To achieve full operational autonomy, the architecture implements a four-stage cyclic framework comprising perception, reasoning and planning, execution, and validation. Through continuous iteration across these stages, this architecture significantly enhances the autonomy, adaptability, and operational efficiency of cloud services. Finally, we present the comprehensive architectural design, identify key technical challenges in agentic cloud computing, and outline potential directions for future research.

keywords : Autonomous Agents, Agentic Cloud Computing, Large Language Models (LLMs), Cloud Service Automation, Multimodal AI

1. Introduction

The rapid evolution of cloud computing [1] has established it as the foundational backbone of modern digital infrastructure, delivering scalable compute, storage, and networking

resources across diverse web services. Traditionally, managing these complex cloud environments has heavily relied on manual human intervention, rigid scripting, and predefined rules. DevOps [2] engineers and system administrators must constantly monitor

* 경기과학기술대학교 컴퓨터모바일융합학과

** 배재대학교 소프트웨어학과

† 교신저자: 김진홍(email: jinhkm@pcu.ac.kr)

접수일자: 2026.09.02. 심사완료: 2026.09.09.

게재확정: 2026.09.20.

infrastructure health, configure deployment pipelines, provision resources, and manually resolve system failures. However, as modern cloud architectures transition toward microservices [3], multi-cloud deployments, and highly distributed systems, their operational complexity has increased exponentially. Human-centric operational paradigms are increasingly hitting structural limits, leading to operational bottlenecks, extended mean time to recovery [4], higher error rates, and escalating infrastructure management costs. Concurrently, the emergence of Large Language Models [5] and multimodal Artificial Intelligence technologies has fundamentally redefined human-computer interaction and task automation. Moving beyond simple text generation or pattern recognition, recent advances in generative AI have enabled the development of autonomous agents—systems capable of high-level reasoning, decision-making, and multi-step tool execution. When integrated into software environments, these AI agents can break down complex objectives, formulate strategic plans, call external APIs, and dynamically adjust their behavior based on real-time feedback. Despite the immense capabilities of LLM-based agents, bridging the gap between high-level natural language intent and low-level, mission-critical cloud management remains a formidable challenge. Conventional web and cloud service architectures are not natively built to accommodate real-time, autonomous agent interactions [6]. Designing a novel architecture that seamlessly integrates autonomous AI

agents into web-based cloud operations is essential for unlocking the full potential of next-generation cloud infrastructure, ensuring high operational efficiency, robust adaptability, and automated lifecycle management.

The remainder of this paper is organized as follows. Section 2 reviews related work on Large Language Models (LLM) and Autonomous AI Agents, Autonomous Cloud Operations and AIOps, and Agentic Architecture for Cloud Services and Infrastructure Management. Section 3 presents the proposed system in detail. Section 4 describes the experimental and evaluation. Section 5 concludes the paper.

2. Related Works

2.1 Large Language Models (LLM) and Autonomous AI Agents

The paradigm of Artificial Intelligence has fundamentally shifted from traditional task-specific models toward Large Language Models (LLMs) and multimodal AI architectures. Advanced foundation models demonstrate emergent reasoning, deep context processing, and complex instruction-following capabilities across diverse domains. Building upon these foundational advances, modern AI research has converged on the concept of autonomous agents—systems that utilize LLMs as central cognitive engines to perceive environments, formulate sequential strategies, invoke external APIs, and iteratively resolve non-trivial objectives. Architecturally, an LLM-based autonomous

agent integrates four core functional modules: Profiling, Memory, Planning, and Action.

1)Profiling: Defines the agent's persona, domain expertise, and operational constraints.

2)Memory: Combines short-term contextual scratchpads with long-term vector databases to maintain operational history and persistent knowledge across multi-turn workflows.

3)Planning: Leverages advanced reasoning paradigms—such as Chain-of-Thought (CoT), ReAct, and Reflexion—to decompose high-level natural language goals into executable sub-tasks while performing real-time self-critique.

4)Action: Enables interaction with external digital ecosystems via function calling, web APIs, command-line interfaces, and software tool execution.

Agentic software frameworks (e.g., LangChain, AutoGPT) and multi-agent systems have demonstrated remarkable efficacy in deterministic software environments, code generation, and diagnostic triage [7]. However, deploying autonomous agents within mission-critical, dynamic cloud computing environments introduces distinct technical challenges. Conventional agents operate in stateless or loosely coupled environments, whereas enterprise cloud infrastructures demand state persistence, transactional consistency, real-time safety validation, and deterministic

rollback capabilities. Bridging this architectural gap requires transitioning from isolated execution scripts to continuous, closed-loop agent frameworks tailored specifically for cloud runtime governance.

2.2 Autonomous Cloud Operations and AIOps

The continuous operational management of dynamic, large-scale cloud infrastructure has long been a core focus of system engineering. Early foundations were established by IBM's Autonomic Computing vision, which conceptualized self-managing systems through the MAPE-K (Monitor-Analyze-Plan-Execute over Knowledge) [8] feedback loop. In modern enterprise environments, this paradigm evolved into Artificial Intelligence for IT Operations (AIOps) [9] an operational methodology that applies machine learning and data analytics to automate telemetry parsing, anomaly detection, event correlation, and root cause analysis (RCA) [10] across complex microservice architectures. Despite their success in reducing operational noise, conventional AIOps frameworks exhibit fundamental structural limitations:

1)Predictive ML over Generative Execution: Standard AIOps platforms rely primarily on supervised time-series models and static classification heuristics. While highly effective at surfacing alerts, they lack the cognitive capacity to synthesize contextual operational strategies or dynamically formulate multi-step recovery workflows for unscripted failure modes.

2)Passive Telemetry vs. Proactive Orchestration: Traditional implementations operate predominantly as diagnostic monitoring layers that escalate actionable insights to human engineers. They lack an end-to-end execution framework capable of taking stateful, low-level infrastructural actions safely without human intervention.

3)Rigid Operational Boundaries: Existing toolkits depend on predefined rules and inflexible threshold configurations, making them brittle when deployed across modern multi-cloud or hybrid-cloud platforms with dynamic topological changes.

Recent research has attempted to bridge these gaps by incorporating LLM-driven interfaces into AIOps pipelines. These hybrid systems utilize natural language querying to inspect log streams and generate terminal commands or Infrastructure-as-Code (IaC) scripts. However, most current solutions treat LLMs as isolated assistant tools rather than embedded autonomous operators. They frequently fail to establish a closed-loop framework that continuously monitors environmental state changes, validates execution outcomes, and performs dynamic rollbacks. Our proposed Agentic Cloud Architecture advances beyond static AIOps by unifying the classic MAPE-K feedback principle with LLM-backed cognitive planning. This creates a proactive, self-healing operational substrate capable of autonomous perception, reasoning, execution, and continuous

validation within web-based cloud environments.

2.3 Agentic Architecture for Cloud Services and Infrastructure Management

The integration of autonomous AI agents into cloud services represents a fundamental shift in infrastructure governance. Traditional infrastructure management relies heavily on Infrastructure as Code frameworks and CI/CD automation pipelines [11]. While IaC standardizes resource provisioning through declarative configurations, it remains static and rigid: it cannot dynamically adapt to runtime environmental changes, interpret high-level operational intent expressed in natural language, or perform real-time, self-healing remediation without manual code updates. Recent literature explores agentic cloud management to overcome these constraints by connecting Large Language Models directly with cloud orchestration APIs [12]. Advanced agent architectures interpret complex goals and translate them into actionable operational workflows [13]. When system anomalies occur, agents inspect container logs, analyze runtime metrics, isolate failure roots, and autonomously trigger corrective measures like pod redeployment or load balancer reconfiguration.

Despite these promising capabilities, current agentic cloud architectures [14] face several critical technical hurdles:

1)Fragmented Execution Loops: Existing approaches often rely on single-turn command

generation without a tight coupling between reasoning and validation, leading to unverified infrastructure states and drift.

2)Non-Determinism and Safety Risks: The probabilistic nature of LLMs introduces non-deterministic outputs, creating severe risks of hallucinated API calls, unintended resource deletions, or security policy violations in production environments.

3)State Synchronization: Maintaining an accurate, real-time internal state model of complex, highly dynamic cloud environments remains challenging for LLM reasoning engines.

3. Proposed System

3.1 Overview System Architecture

The proposed four-stage closed-loop architecture addresses the inherent limitations of the classic MAPE-K framework by shifting from static, rule-based control to an AI agent-driven, highly dynamic paradigm. *First)* while MAPE-K's Monitor stage focuses on collecting raw metrics against predefined thresholds, the proposed Perception module interprets complex, multi-modal system data to extract semantic context. *Second)* unlike the static Analyze/Plan phase that relies on hardcoded policies, the Reasoning & Planning module leverages Large Language Models (LLMs) to dynamically reason through unprecedented failure scenarios and decompose

abstract goals into actionable execution steps. *Third)* whereas MAPE-K lacks an explicit verification step—relying instead on passive re-monitoring—the proposed model introduces a dedicated Validation module. This ensures real-time evaluation of action correctness and safety, creating an immediate feedback loop to trigger corrective re-planning when necessary.

The proposed Agentic Cloud Architecture is an autonomous framework designed to automate end-to-end cloud operations in web-based cloud environments. Moving beyond traditional declarative automation, the system translates high-level natural language user intents into safe, verified operational actions. The overall system topology is illustrated in the architectural diagram below.

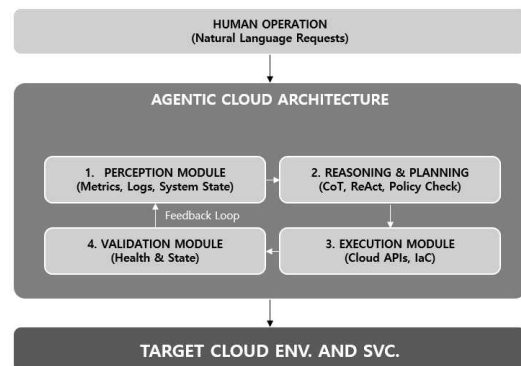


Fig. 1. Proposed Architectural Diagram

3.2 Four-Stage Cyclic Framework

The proposed architecture operates through a continuous, closed-loop four-stage framework designed to guarantee operational safety, transactional reliability, and context-aware

adaptability:

1) *Perception Module:* Collects multi-source telemetry data from the target cloud environment, including real-time performance metrics, system log streams, container runtime states, and user-provided natural language inputs. It cleans, structures, and converts raw infrastructure states into context-rich prompts for the LLM core.

2) *Reasoning and Planning Module:* Acts as the cognitive engine of the architecture. Utilizing advanced reasoning techniques it decomposes broad objectives into an ordered, dependency-aware plan.

3) *Execution Module:* Translates abstract plan steps into concrete, low-level cloud commands.

Code (IaC) execution drivers, Kubernetes API controllers, and CLI shell abstractions.

4) *Validation Module:* Monitors system state immediately following execution. It verifies whether the executed action successfully achieved the target sub-goal without introducing system instability or regression. If a validation failure or unexpected error occurs, a structured feedback signal is generated and routed back to the Perception and Planning modules to trigger automated dynamic replanning or rollback.

The cornerstone of the proposed system is its continuous iterative feedback loop. When executing complex operational workflows—such as rolling out microservice updates or auto-scaling multi-region clusters—unforeseen runtime exceptions (e.g., resource quotas, network timeouts, image pull errors) frequently occur. Instead of terminating upon failure, the Validation Module captures post-execution metric anomalies and error outputs, packaging them into a feedback prompt. The Reasoning and Planning Module reevaluates the failed step in real time, adjusts its internal state representation, and dynamically generates an alternative remediation path. This closed-loop cycle guarantees high autonomy, dynamic adaptability, and resilient self-healing capabilities across heterogeneous web-based cloud service platforms.

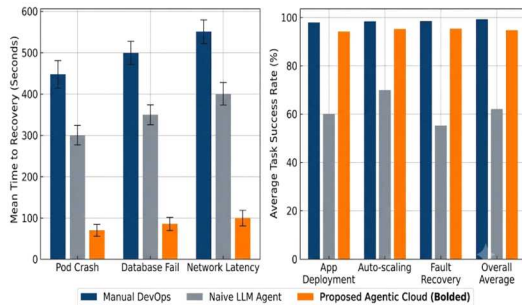


Fig. 2. Experimental Performance Evaluation. (Left) Mean Time to Recovery (MTTR) Comparison by Failure Scenario. (Right) Task Success Rate(%) across Cloud Operational Tasks

It interfaces directly with infrastructure layers using cloud provider SDKs, Infrastructure as

4. Experiments and Evaluation

Our research presents the experimental design and evaluation results conducted to empirically validate the autonomous execution performance, operational efficiency, and fault recovery capabilities of the proposed Agentic Cloud Architecture. The evaluation was conducted on a Kubernetes-based microservices web application benchmark environment (Online Boutique MSA Benchmark). The Cognitive Engine of the proposed architecture utilized an LLM-backed Reasoning Core. For performance baseline comparisons, two control groups were established: traditional Manual DevOps Operations and a Naive LLM Agent (executing single-turn command-line calls without closed-loop feedback). The evaluation focused on three primary operational scenarios:

1) Autonomous Web Application Deployment: End-to-end automated deployment and pod configuration of MSA services based on high-level natural language requests.

2) Traffic Burst Autoscaling: Dynamic HPA (Horizontal Pod Autoscaler) adjustment and optimal resource allocation under sudden traffic spikes.

3) Autonomous Fault Recovery: Root cause diagnosis and self-healing execution during database connection timeouts, pod CrashLoopBackOff errors, and severe network latency.

The experimental results demonstrate that the proposed architecture—driven by its cyclic

four-stage framework (Perception-Reasoning-Execution-Validation) significantly reduces overall execution time (Time-to-Completion) while maintaining a high task success rate compared to conventional approaches by Fig. 2.

5. Conclusion

Our research paper establishes a pivotal shift in cloud computing paradigm, transitioning from traditional, static API-driven automation toward fully autonomous, agentic web-based cloud services. By integrating Large Language Models (LLMs) as core cognitive engines with dynamic environment perception, multi-step planning, and autonomous tool manipulation, the proposed framework addresses the long-standing challenges of handling non-deterministic, complex, and unformatted web environments. Rather than relying on rigid, pre-programmed scripts that break under minor structural UI or system updates, the agentic architecture demonstrates remarkable resilience through real-time feedback loops, self-reflection, and adaptive error recovery. The primary contribution of this research lies in validating that web-based cloud operations can be reliably executed with minimal human intervention. However, the realization of fully production-grade agentic cloud services still faces critical bottlenecks. The study highlights non-trivial trade-offs between cognitive flexibility and system stability. High inference latency, token context limitations, non-deterministic decision-making paths, and cumulative execution errors present continuous

operational risks. Furthermore, granting web-browsing autonomous agents permission to interact with cloud environments introduces unique security vulnerabilities, including prompt injection attacks, unauthorized credential handling, and unexpected resource allocation cost overruns. Looking forward, this research lays a robust foundational blueprint for next-generation self-operating digital infrastructure. Overcoming the highlighted limitations requires future research focusing on lightweight specialized control models, rigorous guardrail alignment, dual-verification execution frameworks, and standardized safety protocols. As foundation models mature in reliability and computational efficiency, the integration of autonomous agentic workflows will redefine cloud orchestration, transforming web-based systems into truly self-configuring, self-healing, and intelligent computational environments.

본 논문은 2026학년도 배재대학교
교내학술연구비 지원에 의하여 수행됨

References

- [1] G. Amudha, P. Gopika, S. G. V. R. M. G. Dinesh and S. S. R., "Cloud Computing: Transformations, Opportunities, and Challenges," 2025 3rd International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS), Erode, India, 2025, pp. 897-903, doi: 10.1109/ICSSAS66150.2025.11080697.
- [2] K. R. K M, K. K. Poojary, Abhay, A. R S and L. K. S R, "AI-Driven DevOps for Intelligent Automation in Continuous Software Delivery Pipelines," 2025 5th International Conference on Evolutionary Computing and Mobile Sustainable Networks (ICECMSN), Coimbatore, India, 2025, pp. 433-440, doi: 10.1109/ICECMSN68058.2025.11382867.
- [3] R. Spista, B. Crispo, P. Giorgini, A. Marchetto and G. Riccardi, "LLMs for Microservice Generation: Capabilities, Challenges, and Advancements," 2025 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT), Bali, Indonesia, 2025, pp. 25-32, doi: 10.1109/IAICT65714.2025.11100313.
- [4] F. Peci, E. Hamiti and I. Khan, "Agentic AI with Chatops for Large Scale Network Operations," 2025 IEEE Conference on Artificial Intelligence (CAI), Santa Clara, CA, USA, 2025, pp. 1617-1626, doi: 10.1109/CAI64502.2025.00242.
- [5] V. Gogineni, "LLM-Powered Multi-Agent Systems: A Technical Framework for Collaborative Intelligence Through Optimized Knowledge Retrieval and Communication," 2025 6th International Conference on Artificial Intelligence, Robotics and Control (AIRC), Savannah, GA, USA, 2025, pp. 452-456, doi: 10.1109/AIRC64931.2025.11077480.
- [6] Y. Zheng, W. Tan, Q. Li, L. Liu, Z. Yan and H. Dai, "A Hybrid Pipeline and Large Language Model System for Task-Oriented Dialogue," 2025 IEEE International Conference on High Performance Computing and Communications (HPCC), Exeter, United Kingdom, 2025, pp. 391-398, doi: 10.1109/HPCC67675.2025.00069.
- [7] S. Singireddy, A. M. Sivagnanam and G. Surianarayanan, "Agentic Test Automation Framework for Complex Software and Database Systems," 2026 International Conference on Smart Futuristic

- Technology, Bangalore, India, 2026, pp. 1-6, doi: 10.1109/ICSFT66733.2026.11507055.
- [8] Y. Xing et al., "SABER: A MAPE-K-based Self-Adaptive Framework for Microservice Bad Smell Refactoring," 2025 IEEE International Conference on Web Services (ICWS), Helsinki, Finland, 2025, pp. 673-684, doi: 10.1109/ICWS67624.2025.00090.
- [9] S. Ahmed, M. Singh, B. Doherty, E. Ramlan, K. Harkin and D. Coyle, "AI for Information Technology Operation (AIOps): A Review of IT Incident Risk Prediction," 2022 9th International Conference on Soft Computing & Machine Intelligence (ISCMI), Toronto, ON, Canada, 2022, pp. 253-257, doi: 10.1109/ISCMI56532.2022.10068482.
- [10] A. Thayumanavan, "Root Cause Analysis of Telecom Network Failures Using Explainable AI Models," 2026 7th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI), Goathgaun, Nepal, 2026, pp. 998-1005, doi: 10.1109/ICMCSI67283.2026.11412649.
- [11] V. S. Oleynikov, E. V. Zhdamirova and O. S. Aleksandrova, "Implementation of CI/CD in Industrial Automation," 2026 ElCon Conference of Young Researchers in Electrical Engineering, Automation & Control Systems (ElCon-EE), Saint Petersburg, Russian Federation, 2026, pp. 359-361, doi: 10.1109/ElCon-EE69794.2026.11453271.
- [12] S. Jindal, P. Rani, V. K. Garg and G. Gupta, "Next-Generation Cloud-Edge Orchestration: Agentic AI with Post-Quantum Security," 2026 International Conference on Computing, Sciences and Communications (ICCSC), Ghaziabad, India, 2026, pp. 1-6, doi: 10.1109/ICCSC67078.2026.11468381.
- [13] M. Dashtbani and L. Tahvildari, "Key Considerations for Auto-Scaling: Lessons from Benchmark Microservices," 2025 IEEE International Conference on Collaborative Advances in Software and COmputiNg (CASCON), Toronto, ON, Canada, 2025, pp. 118-123, doi: 10.1109/CASCON66301.2025.00034.
- [14] U. Agarwal, "Agentic AI Decentralized Edge-Cloud Thermal Orchestration System," 2025 IEEE 11th International Conference on Edge Computing and Scalable Cloud (EdgeCom), New York City, NY, USA, 2025, pp. 68-72, doi: 10.1109/EdgeCom66327.2025.00018.

Authors



Jaehun Song

2017.8 Ph.D. in Electrical, Electrical and
Computer Engineering from
Sungkyunkwan University
2019.3-2025.2 : IT Convergence Research
Researcher, Sungkyunkwan University
2025.3-Present : Assistant Professor in
Gyeonggi University of
Science and Technology
<Research Interests> Information
Security, Image Processing, Holographic
Optical Encryption, Artificial Intelligent



Jinhong Kim

2006.2 Ph.D. in Electrical, Electrical and
Computer Engineering from
Sungkyunkwan University
2017.3-2020.2 : Assistant Professor,
Department of Information
Communication Engineering, Seoul
University
2020.3-Present : Associate Professor,
Department of Software, Paichai
University

<Research Interests> Big Data, Artificial
Intelligent, Intelligent Software, Smart
Vehicular Network, Smart Platform