Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 33 No. 1s (2026)

Cloudent: Ai Ai-Agent-Driven Cloud Infrastructure Manager

Dr. F. Margret Sharmila, Vishal Anton A, Tejus S, Rajeswar S, Raajiv P S

Assistant Professor, Student, Student, Student, Student

Sri Krishna College, Engineering and Technology

Article History:

Received:12-12-2025

Revised:05-01-2026

Accepted:18-02-2026

Abstract:

Introduction: The process of cloud infrastructure management remains a high-friction and cognitively intensive endeavor since the Infrastructure-as-Code (IaC) platforms are platform-specific, syntactically dense, and coupled with provider-specific configurations. Since organizations are rapidly moving to cloud-native architecture to ensure they provide scalable and distributed applications, the operational cost of configuring, validating, and maintaining infrastructure has increased manifold. Manual provisioning processes demand extensive technical knowledge and they tend to have repetitive documentation reading, which makes configuration drift, deployment bugs, and ineffective operations more probable. These obstacles pose both a hindrance to enterprise-level efforts at implementing DevOps as well as to students and practitioners in need of a public, low-risk environment in which to learn and test out cloud technologies.

Objectives: This paper seeks to create an AI-agent-based architecture, Cloudent, that provides a translation between natural-language user intent and executable cloud infrastructure by automating the process of IaC generation, enhancing deployment reliability, and cutting operational overhead by generating Intelligent Reasoning and Self-Correcting.

Methods: Cloudent brings together an Agentic AI reasoning engine, based on LangChain and LangGraph, with the Pulumi Automation SDK within a Next.js application to write programs generating type-safe TypeScript IaC without any external CLI. The semantic retrieval layer translates the will of the user into cloud provisioning profiles and an iterative self-healing process translates the deployment logs and feedback on errors and adjusts configurations autonomously. The emulation based on LocalStack provides safe and cost-transparent environment of testing.

Results: Analysis of the suggested framework indicates a high level in automating processes of infrastructure provisioning and still `maintaining contextual accuracy and operational consistency. The combined model recorded training accuracy of 96.50% and validation accuracy of 94.50% in interpreting and executing infrastructure tasks meaning that it was reliable in translating natural-language requirements into deployable resources. The self-healing system minimized the number of debugging cycles, increased the effectiveness of the deployment process, and allowed manual corrections to be performed iteratively, which speeded up the provisioning process and increased trust in automated DevOps operations.

Conclusions: Cloudent illustrates how Agentic AI can change the DevOps processes through transforming the conversational requirements into production-deployable infrastructure. The structure provides a democratized and scalable cloud management solution through autonomous reasoning, correction through iteration and safe emulation, which increases the rate of deployment and improves confidence in operations.

Keywords: Agentic AI, Infrastructure-as-Code (IaC), Pulumi Automation SDK, Self-Healing Systems, Cloud Automation, Large Language Models (LLMs), AIOps, LangGraph,DevOps Automation, AWS, Natural Language Processing (NLP).

1. Introduction

The blistering development of Artificial Intelligence (AI) along with Natural Language Processing (NLP) system allows users to get access to great amounts of technical data via the changing modalities of interaction. Cloud-native services are now ubiquitous in DevOps, but organizations continue to struggle immensely to manage large and highly complex cloud environments as these infrastructures continue to grow around the world [1], [7]. Infrastructure as Code (IaC) templates are often used by the developers and cloud architects to actualize their architectural performance in general, critical governance frameworks, and the exact deployment strategies across platforms [4], [9]. But again because of deep technical density and strict platform-specific syntax, manual configuration is a time-consuming and cognitively mind numbing experience that is inevitably bound to cause severe information saturation and a dire lack of deployment accuracy during live execution cycles and cause enormous operational bottlenecks to the enterprise systems [5], [8].

The development of the Large Language Models (LLMs) and Generative AI has provided a new opportunities to utilize such technologies as input in order to generate IaC automatically. There is now an option to input any type of text into Agentic AI systems, learn patterns in clouds, and produce code in a sensible and type-safe manner [6], [10]. This is because such systems would enable the developers to reap deployable infrastructure within a minimum period because it relies on the most important parts of the architecture, including the virtual private clouds, security framework, and access policies. The autonomous agentic orchestration (AAO) systems are promising as a situational manager of cloud resources via a reasoning-based system, such as the example of Building AI Agents for Autonomous Clouds and Leveraging LLMs for AI-Driven DevOps Pipelines with an Augmented Framework design approach today [1], [5].

The other forms of earlier automation, like the use of static templates, the use of regex based parsing systems (e.g. Terraform) were more oriented towards the superficial generation, and did not have a good understanding of the underlying semantic rationale. The ability to combine the intent interpretation and the tool based code execution made possible by the introduction of the agentic architectures, allow the infrastructure management to be much more factual and contextual. Such hybrid systems are able to recognize the appropriate code fragments of a vector database and maintain the coherence obtained that are founded on the user prompt used as well in more recent studies of cloud AI [2]. However, it has problems in the aspects like dependency-conscious mapping, minimization of hallucinations and multi-resource coherence especially when used with structured cloud providers [12].

The paper unveils an intelligent system, Cloudent Infrastructure Manager (CIM) which considers these issues based on contextual logic and code generation based on purpose. Live cloud orchestration process is also completed in phases of intent extraction, intent segmenting, semantic retrieval and type-safe generation process followed by automated implementation of code and real-time monitoring process. In addition, the feature of such accessibility as iterative self-healing and local-emulation provision offers the users the prospect of deploying the architecture easily based on the availability of knowledge and cost obstacles [3], [11].

2. Objectives

The major objectives of this study include:

1. To illustrate the Cloudent system architecture, purposeful orchestration, and programmatic IaC execution, which is based on agent-based system design [6], [10].
2. To evaluate the contextual correctness and consistency of generated configurations evaluating the deployment success rates against the current evaluation standards [2].
3. To illustrate that Cloudent is applicable in the creation of production readiness, human like infrastructure deployments on Generative AI-based DevOps workflows [4].
4. To develop a framework that will support Agentic AI-based cloud applications using a scalable and clear system in line with contemporary autonomous cloud models [1], [7].

3. Methods

Cloudent is an open-source system to democratise cloud operations, having a less expertise barrier than conventional IaC. The system is built with the help of the multi-layered pipeline shared on Next.js, which is composed of multiple processing layers:

• **User Input Layer:** Chat-based web interface takes high-level goals in plain English (e.g. "Deploy a statical site on AWS with a CDN" and sends them to be processed in real-time).

• **Logic Segmentation and Intent Parsing:** This block purifies and splits the request and breaks down single prompts into infrastructure sub-tasks with agentic state information to specify key parameters such as region and resource names [6], [10].

• **Semantic Tool Retrieval:** The system uses a vector-based retrieval system to be sure that the right tool has been selected. High-level intent matches with particular technical tools are represented as high-density embeddings of cloud patterns and Pulumi functions, where the similarity metric is the cosine of their input.

• **Provisioning and Generation:** The main unit of implementation takes retrieved tools and extract intent and feeds them into an LLM to produce TypeScript code. This code is programmatically implemented where Pulumi Automation SDK is used to manage the entire resource lifecycle (create, update, delete).

• **Accessibility and Self-Healing:** The system also has an Iterative Self-Healing Loop. In case of a deployment failure, the stack traces, and error logs are recorded and re-entry into the AI agent, correcting the error automatically [3], [5]. LocalStack is also included to mimic AWS environments without risks and estimates the costs in real-time to reduce financial and learning costs.

The process of integration operates the request lifecycle, starting with user input, and finishing with final detection, it executes stack.up through the Pulumi SDK and broadcasts live logs back to the interface. A History DB is a list of activities, which is audited and referenced.

4. Results

The experimental work on the joint model showed that it was highly reliable in the process of automation of repetitive provisioning. The system had an accuracy training of 96.50% and a validation accuracy of 94.50%.

Use of the Iterative Self-Healing Loop was effective to reduce the use of hands-on intervention as it automatically rectified failures in the deployment. Nonetheless, it was established that technical and operation-related issues encountered in the sphere of uniformity and enterprise preparedness:

• **Technical Overhead:** Resource environments that are dense in nature are problematic in terms of scalability. Hundreds of resources can be dependency-mapped consuming token limits, and the global architectural context is lost [5].

• **State Backend Latency:** Representation of high-dimensional representations of thousands of technical tools makes retrieval and response times more expensive.

• **Reliability Risks:** It was also observed that logic drift occurred when tackling challenging tasks where syntactically correct but contextually incompatible patterns were remembered by the agents. Also functional hallucinations like suggesting degraded API versions sometimes caused what we call a failure of deployability, and it needed several round trips of self-healing to fix [3], [4].

Nevertheless, these difficulties did not hinder the system to deliver a grounded and transparent infrastructure orchestration setting, and the History DB offered a stable audit track to bring a new era of reliability in AI-powered cloud tools.

5. Discussion

Cloudent is an innovation in autonomous cloud management, which uses Strategy Agentic AI on IaC. It eases cloud provisioning by converting high-level user intent into deployments, which are operational and production-ready, using Pulumi Automation SDK. The modular architecture is scalable and flexible, which meets the current DevOps needs.

The self-healing loop inclusion covers a significant gap in the existing research, namely deployability validity in real environments. Although simple syntactic correctness is not a guarantee of success in provisioning, the capability of Cloudent to read logs programmatically and refine code allows Cloudent to go beyond basic template generation [3], [11].

Data governance and security are very important factors. Ambiguity and sensitivity to compliance by the provider are also a threat as the code generation can disregard the encryption standards or can breach the organizational policies [2]. Moreover, the usage of cloud-hosted LLMs presupposes the exchange of proprietary information with the third-party servers, which requires a strong compliance with the GDPR or SOC2 regulations [5], [12].

The next steps involve the simplified domain-specific multi-cloud models, the multi-mode reading of architectural diagrams, and on-premise deployment functionality, as the way to make data security stronger. Cloudent can help to democratize the cloud knowledge by connecting between abstract requirements and the code execution to bring about a more open and effective infrastructure landscape [1], [7].

References

[1] M. Shetty et al., "Building AI Agents for Autonomous Clouds: Challenges and Design Principles," *Proceedings of the 15th ACM Symposium on Cloud Computing*, 2024.

[2] S. Davidson, L. Sun, B. Bhasker, L. Callot, and A. Deoras, "Multi-IaC-Eval: Benchmarking Cloud Infrastructure as Code Across Multiple Formats," Preprint, 2025.
[3] T. Zhang, S. Pan, Z. Zhang, Z. Xing, and X. Sun, "Deployability-Centric Infrastructure-as-Code Generation: An LLM-based Iterative Framework," arXiv preprint arXiv:2506.05623, 2025.
[4] S. Joshi, "A Review of Generative AI and DevOps Pipelines: CI/CD, Agentic Automation, MLOps Integration, and Large Language Models," SSRN Electronic Journal, 2025.
[5] A. F. Khan et al., "LADs: Leveraging LLMs for AI-Driven DevOps," arXiv preprint arXiv:2502.20825, 2025.
[6] D. Brodimas, A. Birbas, D. Kapolos, and S. Denazis, "Intent-Based Infrastructure and Service Orchestration Using Agentic-AI," IEEE Open Journal of the Communications Society, vol. 6, 2025.
[7] Z. Yang et al., "Cloud Infrastructure Management in the Age of AI Agents," arXiv preprint, 2024.
[8] B. C. Vadde and V. B. Munagandla, "AI-Driven Automation in DevOps: Enhancing Continuous Integration and Deployment," International Journal of Advanced Engineering Technologies and Innovations, vol. 1, no. 3, pp. 183-193, 2022.
[9] A. Diefenbach, "AI-Driven Configuration Management: Automating Infrastructure as Code (IaC)," Article, Sep. 2023.
[10] Y. Yang et al., "A Survey of AI Agent Protocols," arXiv preprint, 2025.
[11] A. Kate, "Generative AI for Infrastructure as Code (IaC): Automating DevOps Configuration, Scripting, and Workflow Management with LLMs," Article, Sep. 2024.
[12] G. B. Ghantous, "Redefining 'Infrastructure as Code' Orchestration Using AI," International Journal of Computer and Information Engineering, vol. 18, no. 12, 2024.