iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** XI **Month of publication:** November 2025

DOI: https://doi.org/10.22214/ijraset.2025.75440

www.ijraset.com

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

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue XI Nov 2025- Available at www.ijraset.com

An Automated System for Deploying and Monitoring Cloud Instances Across Cloud Platform

Prof. Kalyani Sonawane[1], Pawar Rohan[2], Sathe Vedant[3], Waje Mohan[4]
Information Technology, Pune, India

Abstract: ***Cloud resource provisioning and monitoring are critical activities for organizations using cloud-based infrastructures. Manual management often leads to configuration errors, higher deployment time, and limited tracking. This work presents an automated system designed to deploy and monitor instances on a single cloud platform. The system uses a command-line interface (CLI) and integrated backend services to automate provisioning, configuration, performance monitoring, and fault reporting. Results demonstrate reduced deployment time, improved resource visibility, and lower manual dependency. Results demonstrate ~80% reduction in deployment time and higher reliability through automated monitoring***
Keywords: cloud, automation, deployment, monitoring, CLI, instances

I. INTRODUCTION

Cloud computing has become a major component of modern software infrastructure. Organizations rely on cloud virtual machines for application hosting, experimentation, and data processing. Provisioning and monitoring cloud instances manually are time-consuming and prone to errors. This project proposes an automated system that allows users to deploy cloud instances and monitor their performance efficiently. It simplifies instance management through a unified CLI and dashboard service. The system also monitors CPU load, memory consumption, and status logs, providing real-time insights to users. The remainder of the paper is organized as follows: Section II describes the system design… etc.

A. Problem Statement

Modern organizations rely heavily on cloud infrastructure to deploy applications and services. However, manual provisioning of cloud instances is slow, error-prone, and requires skilled administrators. Moreover, monitoring cloud resources becomes difficult due to distributed systems, leading to inefficient performance, poor visibility, and higher cost. To overcome these issues, an automated cloud instance deployment and monitoring system is essential to reduce human intervention, improve reliability, and optimize performance.

B. System Model

The architecture consists of four major components:

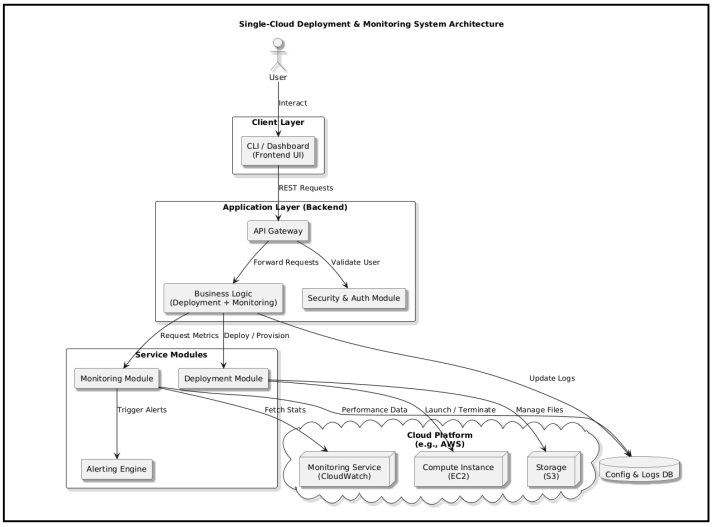


Fig.1-System Architecture

- User Interface (CLI): Issues deployment and monitoring commands.
- Backend Controller: Acts as a bridge between CLI and cloud services.
- Cloud Platform Services: Responsible for provisioning VM resources.
- Monitoring Engine: Captures metrics and reports performance status.

The user interacts with the CLI, triggering calls to the backend, which then communicates with cloud APIs to launch or stop VMs. Resource metrics are periodically collected and displayed to users.

a) Instance Deployment

Users specify instance requirements (image type, size, region). The backend forwards these parameters to cloud APIs, creating and configuring virtual machines.

b) Instance Monitoring

The monitoring engine periodically fetches health statistics like:
CPU utilization
Memory utilization
Running status
Error logs

c) Logging & Alerts

If usage exceeds predefined thresholds, alerts are generated. Logs are stored for debugging.

II. IMPLEMENTATION

1) Frontend: Command-line interface developed using Python.
2) Backend: REST-based middleware for managing cloud API calls
3) Cloud: Single cloud platform used for VM provisioning
4) Monitoring Stack: Uses cloud monitoring APIs for metric extraction.

III. FLOWCHART

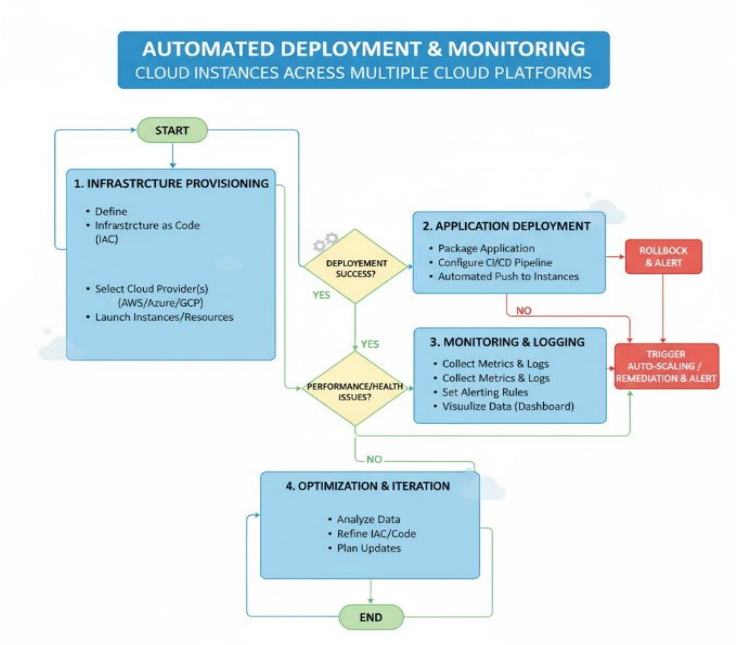


Fig.2-Workflow of automated cloud deployment and monitoring

IV. RESULTS AND DISCUSSION

The following were observed after testing:

1) Deployment time reduced significantly compared to manual approach.
2) Resource usage statistics updated consistently.
3) System successfully detected abnormal CPU and memory levels.
4) Logs helped in debugging instance-level failures.

A user without advanced cloud knowledge can reliably deploy and monitor instances.
A sample metric snapshot:
Instance status → Running
CPU → 23%
Memory → 58%

V. CONCLUSION

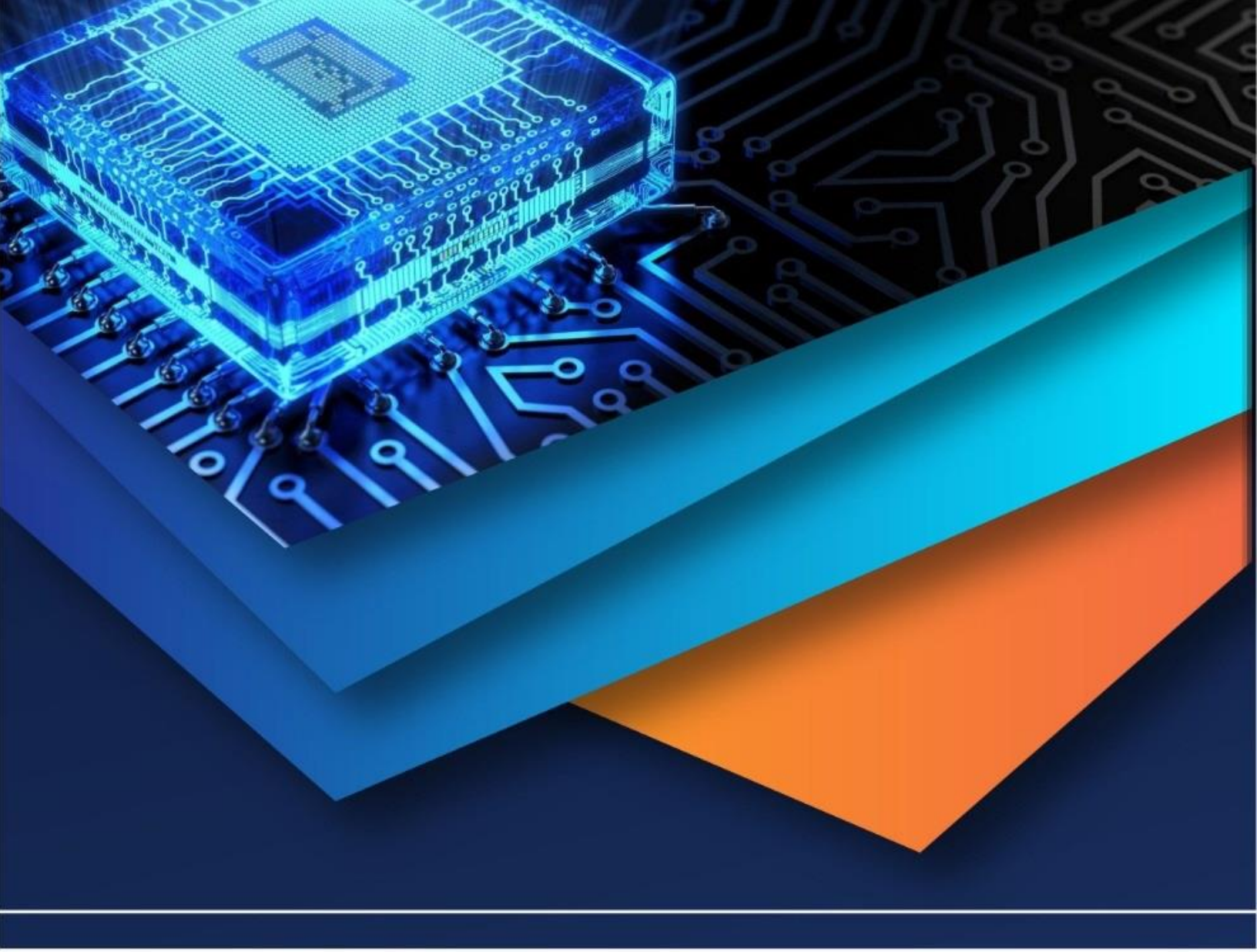
This system allows users to deploy and monitor cloud instances without manual control. With automation, efficiency improved and operational burden reduced. The system can be enhanced by adding multi-cloud support and predictive resource analytics. Future work includes multi-cloud integration and predictive analytics.

VI. ACKNOWLEDGMENTS

We (authors) would like to thank faculty staff and mentors for guidance during the project.

REFERENCES

[1] S. Chinamanagonda, "Automating Infrastructure with Infrastructure as Code (IaaC)," SSRN Tech. Rep., 2019. SSRN
[2] P. Ravindran, "Automating Multi-Cloud Infrastructure: Leveraging Terraform and IaaC for Scalable, Secure, and Efficient Cloud Management," *Int. J. Sci. Res. Compute. Sci. Eng. Inf. Technol.*, vol. 11, no. 2, pp. 2240-2247, Mar. 2025, doi: 10.32628/
[3] P. Mell and T. Grance, "The NIST definition of cloud computing," NIST SP 800-145, 2011.
[4] Amazon Web Services, "Amazon EC2 Documentation."
[5] Google Cloud Platform, "Compute Engine Documentation."



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY