AUTONOMOUS ROLLBACK MECHANISM FOR CLOUD DEPLOYMENTS

***Abhinav Manikanta Yedida, Gandrothula Veerendra, Jakke Veera Shankar, Sajjala Ganesh Siva Surya Srinivas**

Students, Head of the Department, Computer Science and Engineering Department, Ideal Institute of Technology, Kakinada.

Article Received: 08 January 2026

Article Revised: 28 January 2026

Published on: 16 February 2026

***Corresponding Author: Abhinav Manikanta Yedida**

Students, Head of the Department, Computer Science and Engineering Department, Ideal Institute of Technology, Kakinada.

DOI: https://doi-doi.org/101555/ijrpa.2730

ABSTRACT:

Continuous unification and unending deployment (CI/CD) practices authorize speedy software childbirth in cloud surroundings but significantly increase the risk of arrangement-accompanying collapses that can impact service chance and consumer experience. Existing rollback means are principally manual or rule-based, lack exact misstep detection, and support restricted support for stateful services, multi-duty reliance's, and procedure compliance. This paper presents a **result-grade Autonomous Rollback Mechanism for Cloud Deployments** that addresses these disadvantages through a multi-layered design. The projected whole integrates real-opportunity listening accompanying multi-signal anomaly discovery utilizing statistical baselines and inconsequential machine intelligence models to correctly identify arrangement missteps while minimizing fake rollbacks. It supports cautious rollback actions containing informer, blue–green, and dependency-knowledgeable rollback sequencing for microservices and stateful components. A procedure-knowledgeable decision diesel accompanying audit record and optional human supersede guarantees operational security and agreement. Post- rollbacks verification is acted utilizing health, thickness, and depiction checks to confirm profitable improvement. The system is evaluated utilizing blame injection and physical arrangement traces across single-cloud and multi-cloud surroundings. Experimental results display important reduction in 24-hour day to improvement (MTTR), low fake rollbacks rates, continued data thickness, and littlest performance overhead, reinforcing the influence and reliability of the projected approach.

KEYWORDS: Cloud Computing, CI/CD, Autonomous Rollback, DevOps, Fault Recovery, Microservices.

I. INTRODUCTION

The Cloud calculating has enhanced the foundation of up-to-date operating system systems, permissive climbable, on-demand, and highly feasible duties. To meet growing demands for fast change, arranging increasingly select constant integration and unending arrangement (CI/CD) practices. While CI/CD accelerates operating system childbirth, it also increases the risk of arrangement deteriorations caused by arrangement mistakes, reliance mismatches, performance regressions, and surprising runtime behaviors. Such missteps can negatively impact duty chance, user knowledge, and functional costs.

Rollback mechanisms are detracting for checking arrangement failures; nevertheless, existent approaches are largely manual or established motionless rules. These solutions exhibit various restraints. First, failure discovery is frequently vague, relying on restricted versification or fixed thresholds that can bring about postponed recovery or needless rollbacks. Second, most existent mechanisms do not sufficiently support cloud-native architectures that include microservices, complex bury-service reliance's, and stateful elements such as databases. In specific atmospheres, improper rollbacks sequencing or lack of state knowledge can result in spilling defeats and data discrepancy. Third, current rollbacks resolutions frequently lack logical unification with CI/CD pipelines, tactics-knowledgeable decision-making, auditability, and computerized post-rollbacks verification, that are essential for result-grade deployments. Additionally, support for multi-cloud surroundings and systematic judgment under sensible failure sketches wait limited in existent essay.

To address these gaps, this paper intends a **result-grade Autonomous Rollback Mechanism for Cloud Deployments** that allows safe, trustworthy, and imaginative recovery from arrangement deficiencies. The proposed method integrates original-time listening accompanying multi-signal anomaly discovery, joining mathematical baselines and lightweight machine intelligence models to correctly detect arrangement collapses while minimizing dishonest rollbacks triggers. It supports reliable rollbacks methods including informer, blue–green, and reliance-aware rollbacks sequencing for two together stateless and stateful services. A procedure-knowledgeable decision diesel accompanying audit record and optional human supersede guarantees operational security and agreement. Furthermore, automated post-rollbacks proof using fitness, thickness, and conduct checks is employed to reinforce favorable recovery.

The projected system is evaluated utilizing weakness injection experiments and original arrangement traces across single-cloud and multi-cloud surroundings. Evaluation versification involves mean time to improvement (MTTR), dishonest rollback rate, throughput impact, functional overhead, and dossier consistency. Experimental results illustrate that the projected approach significantly advances aid dependability and availability while asserting reduced overhead, thereby numbering united states of America of the art in independent rollbacks for cloud deployments.

II. RELATED WORK

Deployment industrialization and rollbacks mechanisms have happened widely intentional in the context of cloud estimating and DevOps practices. Early bother continuous transmittal stressed automating build, test, and arrangement pipelines to reduce release eras and human wrong. While these approaches improved arrangement speed and dependability, rollback was mostly considered as a manual or driver-driven action, mobilized only after seeable duty degradation.

Several studies have projected rule-located rollbacks mechanisms that depend predefined thresholds over listening metrics in the way that abeyance, wrong rate, or resource exercise. Although natural and easy to redistribute, opening-based approaches lack changeability and frequently contract an illness high wrong-helpful or false-negative rates in active cloud atmospheres. These limitations weaken their influence under variable workloads and evolving use behavior.

Recent research has investigated automated defeat and deviation detection methods to correct cloud dependability. Statistical methods and tool learning–located approaches have been used to listening data to discover bizarre plan behavior during or later deployments. While these methods improve discovery veracity, most works focus generally on recognizing breakdowns and do not provide inclusive resolutions for safe rollbacks killing.

Rollback orchestration, proof of improvement, and functional safeguards are often not talked accompanying the increasing approval of microservices architectures, arrangement complexity has developed considerably. Studies equating monolithic and microservices-located wholes highlight the challenges of directing bury-duty dependencies and free deployments. However, existent research provides restricted support for reliance-aware rollbacks sequencing. Naive rollbacks actions in microservice environments can provoke falling in a rush failure when reliant aids are not restored in a compatible order.

Autonomic and self-curative schemes have also existed projected to improve weakness resistance in cloud platforms. These orders generally goal infrastructure-level breakdowns in the way that node crashes or support tiredness, rather than arrangement-inferred weaknesses. Moreover, many autonomic improvement answers lack integration accompanying CI/CD pipelines and do not include policy-knowledgeable accountable, auditability, or reserved human intervention, that are essential for result environments. Another restraint of existent work is the lack of support for stateful services. Most rollbacks methods adopt stateless applications and do not address dossier thickness challenges associated with rolling back databases or continuous depository.

Additionally, evaluation in earlier studies is frequently restricted to controlled or sole-cloud atmospheres, with lacking concern of multi-cloud deployments, sensible fault needle, and inclusive metrics in the way that wrong rollback rate and functional overhead. In contrast to existent approaches, this work addresses these breaks by providing an end-to-end autonomous rollbacks method that combines correct decline detection, reliance-knowledgeable rollbacks orchestration, state-knowledgeable improvement, CI/CD integration, tactics agreement, and rigorous judgment across sole-cloud and multi-cloud surroundings.

III. PROBLEM STATEMENT AND MOTIVATION

The Modern cloud calculating atmospheres increasingly depend constant integration and constant arrangement (CI/CD) practices to deliver spreadsheet refurbishes immediately and reliably. However, frequent deployments considerably increase the risk of arrangement-related declines precipitated by configuration mistakes, contradictory reliances, performance regressions, and surprising runtime act. Such failures can influence aid spare time, degraded consumer happening, violation assisting-level understandings (SLAs), and increased functional costs.

Existing rollbacks devices in cloud deployments are predominantly manual or established changeless, rule-driven prompts. These approaches contract an illness several detracting restraints. First, misstep detection is frequently vague, relying on restricted listening signals or fixed thresholds that do not readjust to active assigned work conditions, chief to postponed recovery or needless rollbacks. Second, most existent solutions do not sufficiently support cloud-native architectures calm of microservices accompanying complex inter-help reliances and stateful components in the way that databases.

Improper rollbacks sequencing or lack of state knowledge can cause cascading deteriorations and dossier inconsistency. Third, current rollbacks machines frequently lack logical unification accompanying CI/CD pipelines, policy-knowledgeable in charge, auditability, and automated post-rollbacks proof, which are essential for result-grade deployments. Additionally, support for multi-cloud atmospheres and inclusive evaluation under sensible breakdown scenarios debris restricted.

The motivation for this work stands from the increasing complicatedness and operational demands of up-to-date cloud deployments. As arrangements scale their services and select microservices and multi-cloud game plans, manual rollback processes enhance progressively unrealistic, error-liable, and slow. Delayed improvement from deployment deficiencies can have harsh consequences, containing monetary deficit, reputational damage, and reduced consumer trust.

There is a powerful need for an autonomous rollback means that can correctly detect arrangement declines and execute reliable improvement actions outside unending human invasion. Such a mechanism must underrate dishonest rollback resolutions, support reliance-aware and state-constant improvement, and merge seamlessly with existent CI/CD pipelines. Furthermore, result environments demand procedure-aware controls, auditability, and possible human supersede to guarantee operational security and supervisory compliance.

By talking these challenges, **an independent rollback** means can considerably decrease **mean time to improvement (MTTR)**, help **service chance**, and lower **functional overhead.** The ambition of this work is so to **design and judge a reliable, brilliant, and result-ready rollback resolution** that meets the necessities of **new cloud-native systems** and **advances united states of america of the skill in deployment deterioration improvement.**

IV. SYSTEM ARCHITECTURE

The projected **Autonomous Rollback Mechanism for Cloud Deployments** is devised as a modular, extensile, and result-ready architecture that integrates seamlessly accompanying new CI/CD pipelines and cloud-native platforms. The construction addresses key disadvantages recognized in existing work, containing vague failure discovery, lack of reliance awareness, lacking support for stateful aids, and absence of tactics-compelled control. Fig. 1 represents the high-level construction of the projected system.

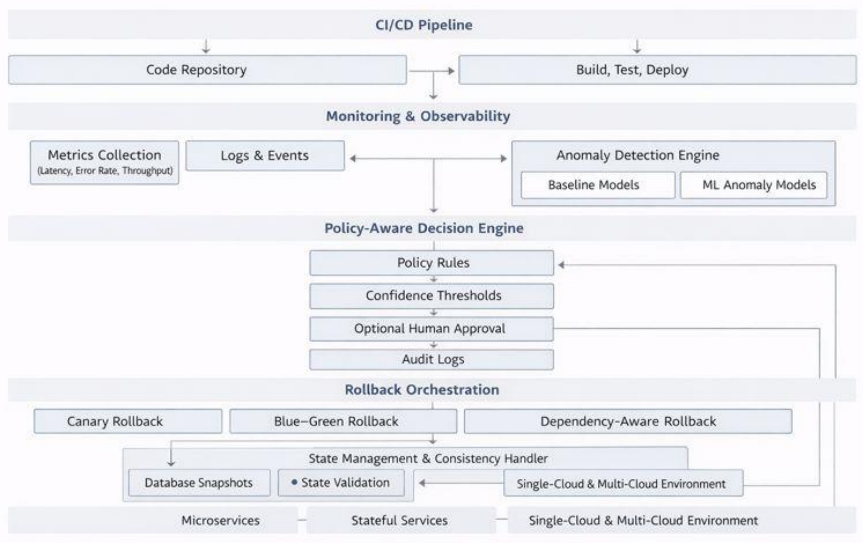

Fig. 1. Architecture of the Autonomous Rollback Mechanism

A. CI/CD Integration Layer

The system is tight joined with the CI/CD passage to monitor deployments from the importance a new release is begun. This layer interfaces accompanying arrangement tools and musical arrangement floors (e.g., bag musical adaptation systems) to path arrangement translations, rollout strategies, and duty metadata. By sinking rollback understanding straightforwardly into the deployment plan, bureaucracy guarantees rapid answer to missteps without upsetting arrangement velocity.

B. Monitoring and Observability Layer

This tier steadily collects real-occasion versification from uses, services, and foundation elements. Metrics include abeyance, mistake rate, throughput, resource exercise (CPU, thought), and aid-level objective (SLO) violations. Logs and occurrences are again captured to supply circumstantial information. Unlike usual approaches that depend single versification, this coating specifies a unified and multi-spatial view of method health, making the support for accurate bankruptcy discovery.

C. Anomaly Detection Engine

The oddity detection turbine reasonings monitoring dossier to recognize abnormal behavior exhibitive of arrangement failures. It connects mathematical standard models (such as exciting averages and control charts) accompanying lightweight machine intelligence models to capture two together short-term departures and complex patterns. Multi-signal equivalence is working to reduce wrong a still picture taken with a camera and false contradiction, discussing a major disadvantage of beginning-based rollbacks methods labeled in prior work.

D. Policy-Aware Decision Engine

Once an irregularity is discovered, the decision generator evaluates either a rollback concede possibility be sparked. This component combines policy-knowledgeable rules that favor confidence levels, aid importance, dependency impact, and functional restraints. It supports audit logging and possible human-in-the-loop authorization for fault-finding services, guaranteeing explainability, responsibility, and compliance accompanying administrative policies. This addresses the lack of government and trust methods in existent autonomous rollbacks answers.

D. Rollback Orchestration Module

The rollback leader arranges executing cautious and matched rollbacks actions. It supports diversified rollbacks strategies, containing informer rollback, blue–green rollbacks, and reliance-aware rollbacks sequencing. Service reliance graphs are used to decide correct rollback order, forestalling spewing failures in microservice-located architectures. The leader interacts accompanying the arrangement policy to restore constant renditions efficiently.

E. State Management and Consistency Handler

To support stateful duties, this piece manages state-knowledgeable rollbacks operations. It relates accompanying dossier stores to ensure regularity through systems such as snapshots, versioned schemas, or refunding conduct. This component directly addresses the lack of state management in existent rollbacks mechanisms and allows cautious recovery of uses accompanying persistent dossier.

F. Post-Rollback Verification Module

After rollbacks execution, this piece acts electronic validation to validate favorable recovery. Health checks, acting probes, artificial transactions, and dossier thickness checks are used to confirm system balance before continuing normal movements. This proof step prevents recurring disappointments and ensures assurance in rollbacks effects, which is frequently gone in prior approaches.

V. ROLLBACK DECISION ALOGORITHM

This division presents the autonomous rollbacks resolution algorithm used to discover arrangement deficiencies and trigger secure rollbacks actions in cloud surroundings. The treasure combines multi-signal irregularity discovery, procedure-aware resolution sense, and dependency-knowledgeable musical arrangement to guarantee fast recovery while underrating dishonest rollback occurrences.

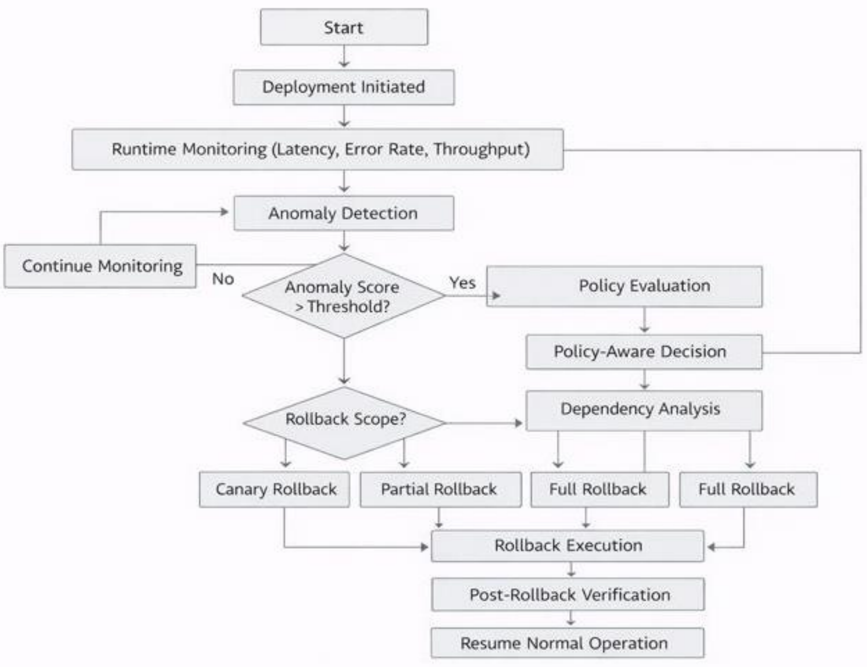

Fig. 2. Rollback Decision Flow of the Autonomous Rollback Mechanism

Algorithm 1: Autonomous Rollback Decision Algorithm.

Input:

Deployment adaptation v, listening metrics M, abnormality assurance threshold θ, tactics rules P, aid dependency diagram G

Output:

Rollback resolution R ∈ {No Rollback, Partial Rollback, Full Rollback}

A. Start Deployment Monitoring: Initialize listening for deployment history v and reclaim historical criterion versification.

B. Collect Runtime Metrics: Continuously accumulate real-occasion versification including abeyance, wrong rate, throughput, capital utilization, and SLO breaches.

C. Anomaly Detection: Compute deviation scores using mathematical measure study and lightweight machine intelligence models each metric.

D. Aggregate Anomaly Score: Combine individual deviation scores to calculate a unified arrangement energy score H.

E. Threshold Evaluation: If $H < \theta$, resume monitoring and return **No Rollback.**

F. Policy Evaluation: Evaluate rollbacks tactics P based on oddity steadfastness, duty criticality, and functional restraints. If policy environments are discontented, create alerts and continue listening.

G. Rollback Scope Determination: Analyse reliance graph G to decide rollbacks scope:

a. Canary rollbacks for local losses

b. Dependency-aware biased rollbacks for affected aids

c. Full rollbacks for scheme-wide breakdowns

H. Execute Rollback: Trigger rollbacks orchestrator to fix picked duties to the last stable translation following reliance-aware sequencing.

I. Post-Rollback Verification: Perform mechanical strength checks, performance confirmation, and dossier thickness checks.

J. Decision Finalization: If verification flourishes, record the rollback operation and restart usual operation. Else, increase to manipulator intervention or introduce alternative improvement conduct.

VI. METHODOLOGY

The projected independent rollback device is grown and evaluated utilizing an orderly methodology that integrates arrangement listening, brainy failure discovery, tactics-aware accountable, and reliable rollbacks execution inside cloud surroundings. The methodology is created to guarantee accuracy, dependability, and reproducibility while trying the disadvantages identified in existent rollbacks approaches.

A. Deployment Environment Setup

The system is implemented in a cloud-nature utilizing containerized applications redistributed through a CI/CD passage. Services are systematized using a microservices construction containing both stateless and stateful elements. Deployment actions to a degree rolling updates, informer releases, and blue–green deployments are backed. The system is evaluated in two together distinct-cloud and multi-cloud configurations to assess ability to move and scalability.

B. Monitoring and Data Collection

During each arrangement, bureaucracy continuously collects honest-period metrics from request, aid, and foundation layers. Monitored versification involve request latency, mistake rate, throughput, CPU and thought utilization, and duty-level objective (SLO) defilements. Logs and arrangement events are more apprehended to provide dependent facts. Historical data is used to enact standard behavior each service superior to arrangement.

C. Anomaly Detection Process

Failure detection is acted utilizing a multi-signal oddity detection approach. Statistical methods to a degree moving averages and control charts are used to label short-term changes from guideline behavior. In parallel, inconsequential machine learning models analyses rhythmical patterns to capture complex and non-undeviating anomalies. Individual deviation scores are normalized and amassed to generate a united arrangement strength score, reducing wrong a still picture taken with a camera caused by temporary vacillations.

D. Policy-Aware Rollback Decision

When the amassed anomaly score surpasses a predefined assurance threshold, the rollbacks conclusion algorithm is resorted to. The resolution generator evaluates policy rules that grant help criticality, oddity steadfastness, reliance impact, and operational restraints. Optional human authorization is supported for detracting duties, ensuring regulated independence. This procedure-aware approach balances speedy improvement with functional security.

E. Rollback Execution Strategy

Based on the decision effect, the rollbacks leader selects an appropriate rollback method. Canary rollbacks are applied for local collapses, while reliance-aware biased rollbacks are used for deficiencies moving a subset of duties. Full rollbacks are performed in the case of system-expansive deficiencies. Service dependency graphs are used to decide rollbacks order, preventing pouring deteriorations. For stateful aids, snapshots or compensating conduct are working to maintain dossier thickness.

F. Post-Rollback Verification

After rollbacks execution, computerized proof is performed to guarantee favorable recovery. Health checks, conduct probes, artificial undertakings, and data regularity checks approve system support before rational operations reopen. If proof abandons, the system escalates to controller attack or alternative recovery conduct.

G. Evaluation Methodology

The influence of the projected approach is evaluated utilizing sin injection and actual arrangement traces. Deployment failures in the way that arrangement mistakes, performance regressions, and duty crashes are purposely introduced. Evaluation versification contains 24-hour day to recovery (MTTR), dishonest rollbacks rate, throughput impact, operational overhead, and dossier constancy. Baseline comparisons are transported against manual and rule-located rollbacks approaches.

VII. EXPERIMENTAL EVALUATION

A. Experimental Setup

Experiments were transported in a cloud-native environment utilizing containerized microservices redistributed through a CI/CD passage. The testbed consists of diversified pertain services, containing two together stateless use services and stateful elements to a degree databases. Deployments were performed utilizing rolling revises, canary releases, and blue–green blueprints. To determine ability to move and scalability, experiments were conducted in two together sole-cloud and multi-cloud configurations.

B. Fault Injection Strategy

To simulate sensible arrangement deficiencies, controlled weakness dose was employed. Fault sketches contained configuration wrongs, reliance mismatches, aid crashes, resource tiredness, and accomplishment regressions introduced all the while arrangement. These mistake types were selected established accepted failure patterns stated in existent cloud deployment studies. Fault dose admitted repeatable and constant evaluation of rollbacks behavior under different failure environments.

C. Baseline Comparison

The projected autonomous rollbacks device was distinguished against two baseline approaches:

1. Manual Rollback: Rollback begun by drivers after detecting deteriorations through listening instrument panels.

2. Rule-Based Rollback: Automated rollback provoked by changeless threshold defilements on individual versification.

These baselines represent average practices labeled in earlier research and industry deployments.

D. Evaluation Metrics

The system was judged utilizing the following metrics:

1. Mean Time to Recovery (MTTR): Time captured to replace help to a stable state later arrangement failure.

2. False Rollback Rate: Percentage of rollbacks started outside actual arrangement defeats.

3. Service Availability: Percentage momentary the system waited functional.

4. Throughput Impact: Performance degradation noticed all along discovery and rollback.

5. Operational Overhead: Additional means use introduced by listening and discovery components.

6. Data Consistency: Integrity of stateful aids subsequently rollbacks.

These metrics support an inclusive view of both improvement influence and functional cost.

E. RESULTS AND ANALYSIS

Experimental results show that the proposed machine considerably reduces MTTR compared to manual and rule-located rollbacks approaches. The use of multi-signal anomaly discovery minimizes fake rollbacks events, reconstructing conclusion accuracy under vacillating workloads. Dependency-knowledgeable rollbacks orchestration blocks pouring failures in microservice-located deployments, happening in higher aid chance.

Performance reasoning indicates that the listening and oddity detection parts present minimal overhead, accompanying insignificant affect throughput during rational movement. Post-rollback proof guarantees constant recovery and marmalade dossier consistency for stateful aids, giving a key limitation of existent rollbacks machines.

F. DISCUSSION

The evaluation displays that mixing intelligent disappointment discovery accompanying policy-knowledgeable in charge and safe rollbacks musical adaptation provides determinable benefits in cloud arrangement dependability.

The results validate the common sense of independent rollback in honest-experience cloud surroundings and highlight allure rightness for large-scale, microservice-located, and multi-cloud orders.

VIII. RESULTS AND DISCUSSION

This portion presents the experimental results got from judging the proposed independent rollbacks mechanism and argues their associations in comparison accompanying existent rollback approaches. The results manifest the influence of integrating creative failure discovery, procedure-aware administrative, and reliance-aware rollbacks written music in modern cloud surroundings.

The Mean Time to Recovery (MTTR) projected autonomous rollbacks mechanism usually worked out lower mean time to improvement (MTTR) distinguished to both manual and rule-located rollbacks approaches. Automated detection and next rollbacks initiation removed delays guide human intervention. In sketches involving frequent deployments and extreme plan load, MTTR was significantly decreased, professed the system's strength to react rapidly to arrangement bankruptcies. These results confirm that independent rollbacks are well-suited for surroundings where expeditious improvement is critical to upholding aid availability and gathering aid-level agreements (SLAs).

A Rollback Accuracy and False Rollback Rate key challenge recognized in earlier work is the high wrong rollbacks rate associated with changeless threshold-located schemes. The proposed multi-signal deviation discovery approach substantially shortened fake rollback occurrences by comparing multiple act and reliability versification. Results display that transient vacillations and duties and responsibilities spikes exceptionally caused rollback conduct, discussing a major disadvantage of existent solutions. This bettering highlights the significance of joining statistical and education-located detection methods for correct rollback resolutions.

Impact on Service Availability and Performance help availability revised particularly with the projected approach due to faster and more exact improvement from deployment deficiencies. Dependency-knowledgeable rollback musical adaptation obviated cascading deficiencies in microservice-located deployments, ensuring that only touched services were flattened back. Performance study showed that the listening and discovery components brought in littlest overhead, with insignificant affect throughput and latency all the while usual operation. This manifests that the system can conduct steadily without debasing use performance.

The State Consistency and Recovery Safety different many existent rollbacks mean that acquire stateless services, the projected method successfully claimed data thickness all the while rollback of stateful elements. Post-rollbacks verification guaranteed that duties resumed movement only subsequently health, conduct, and regularity checks were satisfied. These results address a fault-finding gap in earlier research, place rollback of stateful duties frequently resulted in dossier discrepancy or repeated declines.

The Scalability and Multi-Cloud Evaluation order demonstrated constant behavior across both sole-cloud and multi-cloud deployments. Rollback arrangement across heterogeneous atmospheres imported no significant supplementary improvement delay, indicating good scalability and ability to move. These judgments suggest that the projected machine can be used to large-scale, delivered cloud infrastructures outside cloud-provider-distinguishing reliance's.

Discussion Overall, the results validate that an end-to-end independent rollbacks mechanism considerably upgrades deployment dependability distinguished to traditional rollbacks approaches. The combination of correct disappointment detection, tactics-knowledgeable governance, and secure rollbacks orchestration addresses key disadvantages recognized in existing work. While bureaucracy displays strong conduct, further improvements maybe attained by incorporating predicting decline prevention and adjusting education models. The results support the feasibility of deploying independent rollbacks mechanisms in authentic-world, result-grade cloud surroundings.

IX. REFERENCES

1. J. Humble and D. Farley, *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation,* Addison-Wesley, Boston, MA, USA, 2010.
2. C. Pahl, "Containerization and the PaaS cloud," *IEEE Cloud Computing,* vol. 2, no. 3, pp. 24–31, May–June 2015.
3. Q. Zhang, M. Chen, L. Li, and Y. Zhou, "Automated defeat discovery in cloud arrangements," *IEEE Transactions on Cloud Computing,* vol. 7, no. 2, pp. 456–468, Apr.–June 2019.
4. M. Villamizar, O. Garcés, H. Castro, L. Salamanca, R. Casallas, and S. Gil, "Evaluating the massive and the microservice design pattern to redistribute netting uses in the cloud," in *Proc. IEEE Int. Conf. Computing, Networking and Communications (ICNC),* Anaheim, CA, USA, 2015, pp. 583–590.
5. L. Chen, "Continuous ttransfer: Overcoming enactment challenges," *Journal of Systems and Software*, vol. 128, pp. 72–86, June 2017.
6. N. Dragoni et al., "Microservices: Yesterday, today, and tomorrow," in *Present and Ulterior Software Engineering*, Springer, Cham, Switzerland, 2017, pp. 195–216.
7. A. Basiri et al., "Anomaly detection for cloud reliability," in *Proc. IEEE Int. Conf. Software Architecture (ICSA),* Seattle, WA, USA, 2018, pp. 1–10.
8. P. Sharma, S. Lee, T. Guo, D. Irwin, and P. Shenoy, "Fail-safe: Autonomic failure recovery for cloud-based services," in *Proc. ACM Symp. on Cloud Computing (SoCC),* Santa Clara, CA, USA, 2016, pp. 1–13.