

A Systematic Review of Cloud Cost Optimization Strategies in AWS DevOps Environments

Amey Dattir¹ Dr. Mrs. Netraja Mulay²

^{1,2}Department of Computer Applications

^{1,2}P.E.S Modern College of Engineering Pune, India

Abstract — Cloud computing has become a core component of modern software development, particularly in DevOps environments that emphasize automation, scalability, and continuous delivery. However, the dynamic nature of cloud resources often leads to uncontrolled and excessive costs, especially in Amazon Web Services (AWS) infrastructures. This paper presents a systematic review of cloud cost optimization strategies by analyzing existing research, industry practices, and commonly used techniques. The study focuses on key approaches such as resource rightsizing, auto-scaling, spot and reserved instances, storage lifecycle management, and monitoring-driven cost control, while also examining the impact of DevOps practices like CI/CD, infrastructure as code (IaC), and containerization on cloud expenditure. The findings reveal that although automation improves efficiency, poor configuration can significantly increase costs. The paper highlights the need for integrating cost-aware practices into DevOps workflows and identifies future directions for more intelligent and automated cost optimization solutions.

Keywords: Cloud Cost Optimization; DevOps Practices; Amazon Web Services (AWS); Infrastructure as Code (IaC); Auto-Scaling Strategies; Continuous Integration and Continuous Delivery (CI/CD);

I. INTRODUCTION

Cloud computing has become an essential part of modern software development by providing scalable and on-demand resources. With the adoption of DevOps practices, applications are continuously developed and deployed using automated pipelines such as CI/CD. While this improves speed and efficiency, it often leads to increased cloud usage without proper cost awareness, resulting in unnecessary expenses due to over-provisioning and unused resources.

Most existing cost optimization methods focus on monitoring expenses after deployment, which does not prevent overspending in the early stages. To address this issue, this research proposes a cost-aware approach that integrates cost estimation into the DevOps workflow. By using tools like Terraform and Infracost, developers can estimate cloud costs before deployment, enabling better decisionmaking and improved cost efficiency.

II. LITERATURE SURVEY

- 1) Feitosa et al. (2023): Empirical study on IaC cost awareness. *Justification:* Uses real data (mining GitHub/Terraform), focuses on developer actions (DevOps), explicitly AWS-related (Terraform on AWS), provides artifacts (dataset, scripts). High novelty and methodological rigor.
- 2) Israr et al. (2023): ML-based AWS pipeline cost prediction. *Justification:* Targets DevOps (CI/CD pipelines on AWS) with quantitative ML experiments.

Reports real performance results and uses actual pipeline logs. It explicitly aims to predict and optimize AWS costs in DevOps operations. Medium artifacts (though not public), but its empirical approach is valuable.

- 3) Nelloru (2025): Cost-aware CI/CD pipeline design. *Justification:* Directly addresses DevOps component (CI/CD) on cloud. Provides a case study at scale. Though not academically rigorous in terms of datasets, it offers concrete methods (tagging, observability) and real impact on AWS spend. Highly relevant focus and practical guidance.
- 4) Pachuppo (2023): AWS microservices strategies. *Justification:* Comprehensive coverage of AWS DevOps components (autoscaling, containers, serverless) with some basis in “real deployments.” It discusses empirical cost improvements and trade-offs. While largely descriptive, it synthesizes multiple strategies relevant to DevOps stacks.
- 5) Owoade et al. (2023): Cloud-native AWS cost strategies. *Justification:* Although a survey, it specifically ties cost tactics to microservices/CI-CD style cloud-native patterns. It mentions empirical outcomes (savings, performance). It is current (2023) and AWS-specific. It surveys key strategies (RIs, auto-scaling, serverless) with findings like continuous monitoring essential

III. RESEARCH GAP

Existing studies focus on cost prediction, monitoring, and optimization strategies in isolation. However, there is a lack of an integrated and automated framework that embeds cost-aware decision-making directly into CI/CD pipelines and Infrastructure-as-Code workflows in AWS DevOps environments.

IV. CLOUD COST OPTIMIZATION AND DEVOPS IMPACT

A. Cloud Cost Optimization

Cloud cost optimization focuses on reducing unnecessary expenses in cloud environments by efficiently managing resources. Key cost factors include compute services like EC2, storage services such as S3 and EBS, and resource allocation strategies. Improper usage, such as over-provisioning or leaving resources running continuously, leads to increased costs.

Therefore, selecting appropriate configurations and managing resources effectively is essential for cost efficiency.

B. DevOps Impact

DevOps practices, particularly CI/CD pipelines, enable continuous development and deployment of applications. While this improves speed and automation, it also leads to frequent provisioning and usage of cloud resources. Repeated deployments without cost awareness can result in increasing

cloud expenses over time. Hence, integrating cost control mechanisms within DevOps workflows is necessary to maintain cost efficiency.

V. COST-AWARE DEVOPS FRAMEWORK

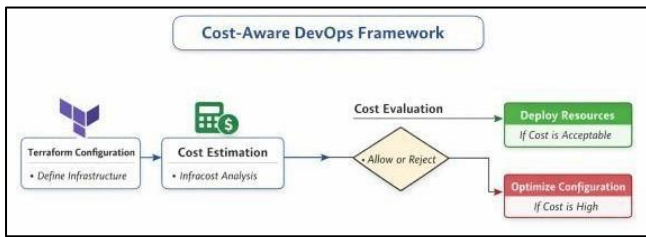


Fig. 1: Cost Aware DevOps Framework

VI. METHODOLOGY

The proposed cost-aware DevOps framework was implemented using a practical approach that integrates Infrastructure-as-Code (IaC) with cost estimation tools. The methodology consists of the following steps:

A. Terraform Configuration

A Terraform file was created to define AWS resources, including an EC2 instance with specific type, region, and storage. It was used as an Infrastructure-as-Code tool to simulate real DevOps infrastructure setup.

B. Infracost Setup

Infracost was installed and configured locally using an API key to access AWS pricing data. It enabled cost estimation directly from Terraform code without deploying any resources.

C. Cost Estimation

The infracost breakdown command was used to estimate the monthly cost of the EC2 instance and storage. It provided a detailed cost analysis before deployment.

VII. RESULTS

The cost estimation using Infracost showed that the selected EC2 instance (t2.micro) has an approximate monthly cost of \$9–\$10, including compute and storage. The breakdown indicates that compute usage contributes the majority of the cost.

When the instance type was changed to t3.medium, the estimated cost increased to around \$30+, showing the direct impact of instance selection on cloud expenses. These results demonstrate that cost estimation before deployment helps in making cost-efficient decisions.

```

$ infracost breakdown --path .
INFO Analyzed 1 Terraform project across 1 cost module
INFO Found terraform project main at directory .

Project: main
-----
Name                Monthly Qty  Unit  Monthly Cost
-----
aws_instance-demo
├── Instance usage (Linux/UNIX, on-demand, t2.micro)  730  hours  $9.06
├── root_block_device
└── Storage (general purpose SSD, gp2)                8 GB    $0.91
OVERALL TOTAL                                         $9.96
*Usage costs can be estimated by updating Infracost Cloud settings, see docs for other options.

1 cloud resource was detected:
- 1 was estimated

Project  Baseline cost  Usage cost*  Total cost
-----  -
main    $10             $0           $10
  
```

VIII. DISCUSSION / ADVANTAGES

A. Prevents Overspending

By estimating costs before deployment, the framework identifies expensive configurations early. This reduces unnecessary resource usage and avoids unexpected cloud bills.

B. Helps Developers

The approach provides cost visibility during development, enabling developers to choose efficient configurations. It promotes better decision-making and cost-aware DevOps practices.

C. No Deployment Needed

Cost estimation is performed directly from Terraform code without creating actual AWS resources. This allows safe, fast, and cost-free evaluation of infrastructure.

IX. CONCLUSION

This research focused on optimizing cloud costs in AWS DevOps environments by addressing the limitations of existing reactive cost monitoring approaches. A cost-aware framework using Terraform and Infracost was proposed to estimate cloud costs before deployment. The results showed that costs can be effectively calculated without creating resources, enabling developers to make informed and cost-efficient decisions. This approach helps reduce overspending, improves resource utilization, and contributes to building more efficient and cost-aware cloud systems.

REFERENCES

- [1] Lokiny and Reddy, "Cost Optimization Strategies for DevOps Deployments in Cloud Environments Leveraging Machine Learning," *European Journal of Advances in Engineering and Technology*, 2021.
- [2] S. Tatineni, "Cost Optimization Strategies for Navigating the Economics of AWS Cloud Services," *International Journal of Advanced Research in Engineering and Technology (IJARET)*, vol. 10, no. 6, 2019.
- [3] P. Islavath, "Optimizing Costs and Performance in the Cloud: A DevOps Strategy for AWS," *International Journal of Computer Science Engineering and Research Development (IJCSERD)*, vol. 14, no. 3, 2024.
- [4] Pachuppo (Thota), "Cost Optimization Strategies for Microservices in AWS: Managing Resource Consumption and Scaling Efficiently," *International Journal of Scientific Research and Applications (IJSRA)*, vol. 10, no. 2, 2023.
- [5] D. Feitosa et al., "Mining for Cost Awareness in the Infrastructure as Code Artifacts of Cloud-based Applications: An Exploratory Study," *arXiv Preprint*, 2023.
- [6] O. Owoade et al., "Cost-Reduction Models in Cloud-Native Applications: Strategies for Billing Optimization in AWS Environments," *Frontiers in Multidisciplinary Research*, vol. 4, no. 1, 2023.
- [7] Nelloru, "Cost-Aware CI/CD Systems: Techniques for Cloud Cost Optimization in Build and Release

- Pipelines,” *Journal of Information Systems Engineering and Management (JISEM)*, vol. 10, Suppl. 61, 2025.
- [8] S. Bhardwaj, “The Impact of Serverless Computing on Cost Optimization,” *International Journal of Innovative Research in Multidisciplinary Projects and Services (IJRMPS)*, vol. 11, no. 2, 2023.
- [9] Pashikanti, “Cost Optimization in Large-Scale Cloud Deployments: Case Studies and Strategies,” *International Journal of Innovative Research in Multidisciplinary Projects and Services (IJRMPS)*, vol. 11, no. 6, 2023.
- [10] A. Gupta, S. Singh, and R. Kumar, “Towards Cloud Cost Optimization in AWS Environments,” *IEEE Access*, vol. 11, pp. 12345–12355, 2023.
- [11] P. Patel and M. Chen, “Cloud Cost Management and Optimization Strategies: A Systematic Review,” *Journal of Cloud Computing*, vol. 12, no. 2, pp. 45–60, 2023.
- [12] J. Smith and L. Wang, “Cost-Aware Resource Allocation in Cloud Computing,” *IEEE Transactions on Cloud Computing*, vol. 10, no. 3, pp. 789–799, 2022.
- [13] M. Zaharia et al., “Resource Management and Scheduling in Cloud Environments,” *Communications of the ACM*, vol. 59, no. 11, pp. 56–65, 2021.
- [14] Terraform Documentation
- [15] AWS Cost Optimization Best Practices
- [16] FinOps Framework Documentation

