Journal of Frontiers in Multidisciplinary Research

Cost-Reduction Models in Cloud-Native Applications: Strategies for Billing Optimization in AWS Environments

Samuel Owoade [1*], Bolaji Iyanu Adekunle [2], Ejielo Ogbuefi [3], Oyejide Timothy Odofin [4], Oluwademilade Aderemi Agboola [5], Oluwasanmi Segun Adanigbo [6]

[1] Kennesaw State University, USA
[2] Data Scientist, GSFEN Limited, Nigeria
[3] University of Massachusetts amherst And Soodle Technology
[4] DXC Technology, Poland
[5] Data Culture, New York, USA
[6] Remis Limited, Lagos, Nigeria

* Corresponding Author: **Samuel Owoade**

Article Info

E-ISSN: 3050-9726
P-ISSN: 3050-9718
Volume: 04
Issue: 01
January-June 2023
Received: 10-03-2023
Accepted: 14-04-2023
Published: 20-05-2023
Page No: 398-405

Abstract

This paper explores cost-reduction models and strategies for optimizing billing in AWS environments, particularly focusing on cloud-native applications. The growing adoption of cloud technologies has made cost optimization an essential factor in achieving sustainable and efficient cloud infrastructure. AWS offers a wide array of pricing models and services, but effective cost management requires businesses to understand and apply strategic measures such as right-sizing instances, leveraging Reserved Instances (RIs), and utilizing Savings Plans. Additionally, advanced techniques like auto-scaling, Spot Instances, and serverless computing present substantial opportunities for reducing cloud expenses. The paper also highlights the role of third-party cost management tools and AWS's native cost monitoring solutions in providing businesses with the insights needed to optimize resource utilization and minimize waste. The findings show that through continuous monitoring and dynamic resource allocation, organizations can optimize costs while maintaining performance. The paper concludes with insights on future directions for cost reduction in AWS environments, including the use of AI-powered optimization tools and multi-cloud management strategies, ensuring that cloud-native applications can scale efficiently without incurring unnecessary expenses.

DOI: https://doi.org/10.54660/.JFMR.2023.4.1.398-405

Keywords: Cloud-Native Applications, AWS Cost Optimization, Billing Models, Auto-Scaling

1. Introduction

1.1 Overview of Cloud-Native Applications

Cloud-native applications represent a modern approach to software design that leverages the flexibility and scalability of cloud computing [1, 2]. These applications are built to fully utilize the cloud environment fully, often adopting microservices architecture, containerization, and continuous integration/continuous deployment (CI/CD) practices [3, 4]. Unlike traditional on-premise systems, cloud-native applications are designed to scale dynamically, be fault-tolerant, and optimize resources efficiently. They are highly portable across different cloud platforms, which enhances their operational flexibility and reduces dependency on specific hardware or infrastructure [5, 6].

The architecture of cloud-native applications allows for easier maintenance, faster development cycles, and greater resilience. By utilizing cloud services such as storage, databases, and compute power on-demand, cloud-native apps offer significant

advantages in terms of cost-efficiency, performance, and flexibility. These advantages, however, come with the challenge of managing complex billing structures associated with cloud services like AWS [7-9].

For organizations transitioning to cloud-native environments, the transition often involves rethinking traditional processes of managing and optimizing costs [10, 11]. Given the rapidly changing nature of cloud services, ensuring that costs are kept under control while maintaining high performance is a critical concern. Cloud-native applications enable organizations to take full advantage of cloud elasticity but also require diligent oversight of resource consumption and cost management [4, 12, 13].

1.2 Importance of Cost Optimization in AWS Environments

AWS provides a vast array of cloud services, each with its own pricing structure, making it essential for organizations to manage their spending effectively. Without a clear cost optimization strategy, it is easy for cloud resources to spiral out of control, especially in a cloud-native environment where scaling and resource usage can fluctuate frequently. Cost optimization is critical not only for reducing waste but also for ensuring that organizations derive the most value from their cloud investments [14, 15].

In AWS, a variety of cost-related factors contribute to overall spending, such as compute power, storage, data transfer, and API requests. For businesses running cloud-native applications, understanding and managing these variables is key to staying within budget and preventing unnecessary expenditures [16, 17]. Effective cost optimization in AWS can lead to significant savings, particularly through strategies such as right-sizing resources, using reserved instances, and taking advantage of volume discounts [18-20].

The AWS ecosystem is designed to support a wide range of use cases, and its flexibility offers organizations the ability to choose from numerous services tailored to their specific needs. However, with this flexibility comes the challenge of managing costs in a transparent and predictable manner. As organizations move towards larger, more complex cloud-native applications, the need for a systematic approach to cost optimization becomes even more pressing [21-23].

1.3 Purpose and Scope of the Paper

This paper seeks to provide a comprehensive understanding of cost-reduction models within cloud-native applications hosted on AWS. It explores the various billing models offered by AWS and identifies key strategies for optimizing spending without compromising the performance and scalability of cloud-native applications. The paper also aims to present a balanced view, weighing both the technical complexities and the financial benefits of cloud optimization strategies in AWS environments.

The scope of the paper is centered on AWS, as it remains one of the most widely used cloud platforms globally. By focusing on AWS, the paper can provide in-depth insights into cost-reduction strategies specifically tailored to this cloud environment. However, the principles discussed can be applied more broadly to other cloud platforms, offering valuable lessons for organizations seeking to optimize costs in any cloud-native setting.

By examining advanced techniques and tools for cost management, such as AWS cost explorer, rightsizing, and using serverless computing, this paper will offer practical guidance for IT professionals and decision-makers. Ultimately, the goal is to help organizations improve their understanding of cloud-native cost optimization, leading to more efficient and cost-effective cloud strategies.

2. Understanding Billing in AWS

2.1 AWS Pricing Models

AWS provides a variety of pricing models to accommodate different organizational needs and usage patterns. The most common pricing model is the pay-as-you-go model, where users are billed based on actual resource consumption. This model allows organizations to pay only for what they use, which is ideal for variable workloads or when scaling resources up and down. It offers flexibility but can lead to unpredictable costs if resource consumption is not carefully monitored [24-26].

Another key pricing model is Reserved Instances (RIs), which offer significant cost savings in exchange for committing to a specific instance type and term length, typically one or three years. This model is particularly beneficial for workloads with steady, predictable traffic, as it can provide up to a 75% discount compared to on-demand pricing [18, 27, 28]. AWS also offers Savings Plans, which provide cost reductions in exchange for a commitment to a specific amount of compute usage over a one- or three-year period. Unlike RIs, Savings Plans offer more flexibility in instance family and size [22, 29].

Finally, AWS has a spot pricing model for instances, where unused capacity is offered at a discounted rate. This model is suitable for flexible workloads that can tolerate interruptions but can lead to substantial cost savings. While this pricing model can be highly beneficial for non-critical applications, it requires careful management to ensure that workloads can be interrupted and restarted without causing service disruptions [17, 18, 30].

2.2 Common Cost Drivers in Cloud-Native Applications

Cloud-native applications, by nature, are designed to scale and utilize a variety of cloud services. As such, several factors contribute to their overall cost. One significant driver is compute resources—the virtual machines, containers, and serverless functions that run the application [31, 32]. The number and size of instances, along with the frequency and duration of their usage, directly impact the total cost. While containerized applications might optimize resource usage, mismanagement in instance sizing can lead to over-provisioning, which increases costs [33-35].

Another significant cost driver is storage. Cloud-native applications often generate large volumes of data, which need to be stored and managed efficiently. AWS offers multiple storage options, including Amazon S3 for object storage and Elastic Block Store (EBS) for block storage, each with different pricing models based on usage and data transfer. Inefficient data storage practices—such as keeping unused data or failing to optimize data lifecycle management—can drive up costs [36, 37].

Additionally, data transfer between AWS services and to external locations can also become a major cost driver. AWS charges for data transferred between regions or out of the AWS network, which can add up quickly if cloud-native applications are not optimized for minimal data movement. Efficiently managing data transfer and minimizing unnecessary inter-region communication or egress is crucial to reducing overall costs [38-40].

2.3 Challenges in Managing AWS Billing

Managing AWS billing can be complex due to the sheer number of services and pricing models available. The first challenge is the lack of transparency in understanding exactly where and how costs are incurred. AWS provides detailed billing reports, but these can be overwhelming, especially when organizations use multiple services across various accounts. Identifying cost anomalies, such as unexpected spikes in resource usage or underutilized services, can be difficult without the right tools and processes in place [41-43].

Another challenge is the dynamic nature of cloud costs. As cloud-native applications scale and evolve, so too do their resource requirements, and managing these changes in real time can be tricky. While cloud-native architectures offer immense scalability, they also increase the potential for cost inefficiencies. For example, automatic scaling features might inadvertently scale up resources during periods of high traffic, leading to higher-than-expected costs. Without monitoring tools, such spikes can go unnoticed until the monthly billing cycle [18, 44].

Finally, managing different AWS accounts can be an issue, particularly for organizations with multiple teams or departments using AWS independently. While AWS Organizations can help centralize billing, each account may have different spending patterns, and cross-account visibility can be limited [18, 45]. This decentralized approach can make it challenging to ensure that costs are appropriately allocated and optimized across the organization, especially when budgets are not aligned with actual usage patterns. Effective multi-account management and cost allocation strategies are essential for ensuring that AWS billing remains under control [27, 46].

3. Cost-Reduction Strategies

3.1 Instance Sizing and Right-Sizing

One of the most effective strategies for cost reduction in AWS environments is instance sizing and right-sizing. This involves selecting the appropriate type and size of compute resources to match the specific requirements of the workloads running on AWS. Over-provisioning resources can lead to significant cost overruns, as organizations may be paying for unused computing capacity. Conversely, under-provisioning can result in performance issues or downtime, which could negatively impact the user experience [47, 48].

Right-sizing involves continuously monitoring workloads to adjust resources based on usage patterns. AWS offers tools like AWS Trusted Advisor and AWS Compute Optimizer, which provide recommendations for right-sizing instances based on their actual performance and utilization [25, 49]. By identifying underutilized instances and recommending smaller, more efficient alternatives, these tools can help organizations minimize unnecessary costs. The process of right-sizing should be ongoing, as workloads change over time, and resource requirements fluctuate [18, 50, 51].

To effectively implement right-sizing, organizations must establish a clear understanding of the performance characteristics of their workloads. This includes analyzing metrics such as CPU utilization, memory usage, and network throughput to ensure that the selected instance types are neither oversized nor undersized. By optimizing instance sizes for workloads, businesses can achieve significant cost reductions without compromising performance [15, 52, 53].

3.2 Reserved Instances and Savings Plans

Another critical strategy for reducing AWS costs is leveraging Reserved Instances (RIs) and Savings Plans. These models allow organizations to commit to using certain AWS services over a long-term period, typically one or three years, in exchange for substantial discounts compared to on-demand pricing [54-56]. Reserved Instances are ideal for predictable, steady-state workloads that require continuous compute capacity. AWS offers RIs for services like EC2 and RDS, with discounts ranging from 30% to 75% depending on the instance type and commitment length [57-59].

Savings Plans, introduced by AWS as a more flexible alternative to RIs, provide savings across a wider range of services, including EC2, Lambda, and Fargate [60, 61]. Unlike RIs, Savings Plans do not tie users to specific instance types or regions, offering greater flexibility while still achieving cost savings. The two types of Savings Plans—Compute Savings Plans and EC2 Instance Savings Plans—allow organizations to optimize costs across a broader spectrum of cloud services, accommodating changes in workload demands [62, 63].

These models are particularly useful for organizations with predictable and stable resource usage patterns, such as those running enterprise applications, websites, or databases with consistent traffic [64]. However, the key to realizing the full benefits of Reserved Instances and Savings Plans is understanding the anticipated usage over the term and committing accordingly. Businesses that anticipate fluctuating workloads or those with dynamic scaling requirements should consider carefully whether these models are the right fit for their needs [65-67].

3.3 Efficient Resource Utilization through Auto-Scaling

Auto-scaling is a powerful AWS feature that enables organizations to adjust their compute resources based on the demand automatically. By using auto-scaling, organizations can scale up or down their instances or services in response to changes in traffic, ensuring that resources are allocated dynamically. This can significantly reduce costs, as organizations only pay for the compute power they need at any given time, rather than maintaining over-provisioned resources [68, 69].

AWS offers several auto-scaling solutions, such as EC2 Auto Scaling and Elastic Load Balancing (ELB), which work together to adjust the number of instances based on pre-defined policies automatically [70]. For example, during periods of high traffic, auto-scaling will increase the number of running instances to handle the load, and during off-peak hours, it will scale down resources to minimize costs. This allows organizations to maintain optimal performance during peak periods while avoiding the costs associated with idle resources [71, 72].

In addition to EC2, auto-scaling can also be applied to serverless services like AWS Lambda and containerized applications using AWS Fargate. For businesses running cloud-native applications, auto-scaling helps ensure that resources are efficiently utilized, reducing the need for manual intervention and minimizing waste [73]. To maximize the benefits of auto-scaling, businesses should regularly review scaling policies and thresholds to ensure they align with real-time usage patterns, thus preventing over-provisioning or under-provisioning that could lead to additional costs [74-76].

4. Advanced Optimization Techniques

4.1 Cost Monitoring and Cost Explorer Tools

Effective cost monitoring is a fundamental aspect of managing cloud expenses in AWS. AWS provides several tools to help organizations track and manage their spending. One of the most powerful tools is AWS Cost Explorer, which allows users to visualize, analyze, and forecast their AWS costs over time [67]. With Cost Explorer, businesses can break down costs by service, account, region, and linked resources, helping to identify areas where spending can be optimized. It offers detailed reports and insights into how resources are being used, allowing users to detect cost anomalies, track usage patterns, and forecast future expenditures based on historical data [77, 78].

Another essential tool is AWS Budgets, which enables organizations to set custom cost and usage budgets and receive alerts when costs exceed the set thresholds. This proactive approach ensures that businesses are notified in real-time if their AWS spending is spiraling out of control, allowing for swift corrective actions [58]. AWS also provides Cost Anomaly Detection, which uses machine learning to identify unusual spending patterns and potential cost overruns. This tool is particularly helpful for detecting unexpected spikes in resource usage or misconfigured services that might lead to unnecessary expenses [79].

By leveraging these cost monitoring tools, businesses can gain greater visibility into their AWS spending and make informed decisions about resource allocation. Continuous monitoring allows organizations to adjust usage patterns, optimize resource utilization, and ultimately reduce costs. With accurate and real-time cost data, companies can ensure they remain within budget while maximizing the value derived from their cloud resources [80, 81].

4.2 Spot Instances and Serverless Computing

Spot Instances are a powerful way to reduce AWS costs for flexible workloads. AWS Spot Instances allow users to bid on unused EC2 capacity at a significant discount compared to on-demand prices—up to 90% less [82, 83]. These instances are ideal for non-critical or stateless applications that can tolerate interruptions. Since Spot Instances can be terminated by AWS with little notice, they are best suited for applications that can quickly resume or restart, such as batch processing jobs, data analysis, or background processing tasks [84, 85].

Spot Instances can be used in conjunction with EC2 Auto Scaling to automatically replace terminated instances and ensure that workloads continue running smoothly. This hybrid approach of combining Spot Instances with On-Demand or Reserved Instances enables businesses to achieve optimal cost savings while maintaining the necessary resources for their applications [86, 87]. However, organizations need to build resilient architectures that can handle the potential volatility of Spot Instances, including implementing fault-tolerant designs like stateful storage and distributed processing [88, 89].

Serverless computing is another key optimization technique that can help reduce costs in cloud-native applications. AWS services like AWS Lambda, Fargate, and API Gateway provide serverless computing capabilities, where users are charged based on actual usage rather than maintaining dedicated servers [90, 91]. Serverless computing eliminates the need to provision and manage infrastructure, which results in a more cost-effective solution for workloads with variable or unpredictable traffic. Since users are billed based on the number of function invocations or compute time, serverless models help minimize idle resource costs, especially for event-driven or microservices-based applications [59, 92, 93].

4.3 Third-Party Cost Management Tools

While AWS provides a range of native cost management tools, many organizations find value in integrating third-party cost management tools to gain more granular insights into their cloud spending. These tools typically offer enhanced analytics, better cost allocation, and easier integration with multi-cloud environments. Leading third-party tools include CloudHealth by VMware, CloudCheckr, and Spot by NetApp, which provide advanced cost optimization features and real-time cost monitoring across multiple AWS accounts and services [94, 95].

Third-party tools often come with more customizable reporting and alerting features compared to AWS native tools. For example, CloudHealth offers an extensive suite of features that allow organizations to monitor and optimize their cloud costs by providing in-depth analysis and recommendations on reserved instance usage, resource utilization, and budget tracking [62]. These platforms also integrate with various cloud providers, enabling businesses to manage their cloud costs across a multi-cloud or hybrid cloud environment, ensuring a more comprehensive approach to cost management [18, 96, 97].

Another advantage of third-party cost management tools is their ability to provide predictive analytics, helping organizations forecast future costs based on historical data. This can be particularly valuable when planning for future cloud spending and aligning budgets with anticipated usage patterns [98]. Moreover, third-party solutions often offer more intuitive dashboards, allowing non-technical stakeholders to understand cloud spending trends and identify opportunities for savings without needing to navigate the complex AWS billing interface [99]. By leveraging these tools, organizations can gain a more complete view of their cloud expenses and make data-driven decisions to optimize their AWS billing [100].

5. Conclusion

In this paper, we have explored various strategies for optimizing billing and reducing costs in AWS environments, particularly for cloud-native applications. One of the key findings is the importance of selecting the right instance sizing and continuously applying right-sizing principles to ensure resources are aligned with actual workload requirements. This approach helps organizations avoid over-provisioning and reduce waste. Additionally, leveraging AWS's Reserved Instances (RIs) and Savings Plans provides substantial cost savings, especially for predictable workloads, by offering discounts in exchange for long-term commitments.

We also identified auto-scaling as an essential technique for optimizing resource utilization. This strategy allows for dynamic adjustments to resource allocation based on demand, ensuring that cloud-native applications can scale efficiently while minimizing costs. Furthermore, advanced optimization techniques such as Spot Instances and serverless computing were highlighted as effective means to reduce costs for non-critical or flexible workloads. Tools like AWS Cost Explorer and third-party cost management solutions offer valuable insights and enable continuous monitoring to

help organizations stay within budget while maximizing cloud usage.

Billing optimization has a profound impact on cloud-native applications, as it directly affects the cost-efficiency and scalability of the architecture. By adopting cost-optimization strategies, organizations can significantly reduce their cloud expenditure while maintaining or even improving application performance. For cloud-native applications, which are designed to scale dynamically, efficient resource allocation is essential to avoid unnecessary costs during both peak and off-peak periods. Right-sizing, auto-scaling, and using spot instances allow companies to handle varying traffic loads without over-provisioning, ensuring cost efficiency across the application lifecycle.

The impact of billing optimization extends beyond just financial savings; it also enhances operational agility. Cloud-native applications are typically deployed in environments that require rapid scaling and constant adjustments to meet evolving demands. Billing optimization ensures that these applications remain responsive to changes in workload without overspending. As a result, organizations can invest savings into other critical areas, such as innovation, feature development, or improving customer experience, thus driving business growth while managing costs effectively. Moreover, optimized billing practices promote a culture of financial accountability within organizations, as teams become more aware of resource utilization and cost management. This cultural shift towards cost-consciousness can improve decision-making across departments and help businesses maintain a sustainable and efficient use of cloud resources.

As AWS continues to evolve and introduce new services and pricing models, there are significant opportunities for future advancements in cost-reduction strategies. One key direction is the growing use of machine learning (ML) and AI-powered cost optimization tools. AWS is already leveraging AI in its cost monitoring tools, and as these technologies mature, they will enable even more precise cost predictions and automated optimization actions. Businesses can look forward to more sophisticated systems that analyze usage patterns and suggest proactive cost-saving measures based on historical data and predictive analytics.

Another emerging area is the integration of multi-cloud environments. As organizations increasingly adopt hybrid or multi-cloud strategies, there will be a need for tools and approaches that optimize costs across various platforms. Third-party cost management solutions are expected to play a critical role in helping businesses manage expenses across not only AWS but also other cloud providers like Azure and Google Cloud, ensuring that resources are allocated optimally across diverse infrastructures. Lastly, serverless computing and edge computing will likely become more prominent in the future, offering new avenues for cost reduction. As cloud-native applications become even more distributed, the ability to allocate resources at the edge dynamically could lead to further reductions in bandwidth and compute costs. As these technologies mature, organizations will be able to adopt more flexible, scalable, and cost-effective models for running their cloud-native applications.

References

1. Debski A, Szczepanik B, Malawski M, Spahr S, Muthig D. In search for a scalable & reactive architecture of a cloud application: CQRS and event sourcing case study. IEEE Softw. 2017;99.
2. Srirama SN, Adhikari M, Paul S. Application deployment using containers with auto-scaling for microservices in cloud environment. J Netw Comput Appl. 2020;160:102629.
3. López MR, Spillner J. Towards quantifiable boundaries for elastic horizontal scaling of microservices. In: Companion Proceedings of the 10th International Conference on Utility and Cloud Computing. 2017. p. 35–40.
4. Laisi A. A reference architecture for event-driven microservice systems in the public cloud. 2019.
5. Kumar TV. Cloud-Based Core Banking Systems Using Microservices Architecture. 2019.
6. Hatami-Alamdari E, Etzioni Z. Monolithic architecture vs. multi-layered cloud-based architecture in the CRM application domain. 2019.
7. Bolscher R. Leveraging serverless cloud computing architectures: developing a serverless architecture design framework based on best practices utilizing the potential benefits of serverless computing [dissertation]. University of Twente; 2019.
8. Taibi D, Systä K. From monolithic systems to microservices: A decomposition framework based on process mining. In: International Conference on Cloud Computing and Services Science. SciTePress; 2019. p. 153–64.
9. Debski A, Szczepanik B, Malawski M, Spahr S, Muthig D. A scalable, reactive architecture for cloud applications. IEEE Softw. 2017;35(2):62–71.
10. Grohmann J, Nicholson PK, Iglesias JO, Kounev S, Lugones D. Monitorless: Predicting performance degradation in cloud applications with machine learning. In: Proceedings of the 20th International Middleware Conference. 2019. p. 149–62.
11. Raghunathan S. Elevating System Reliability through Observability in Cloud Native Applications. J Technol Innov. 2020;1(4).
12. Khatri A, Khatri V. Mastering Service Mesh: Enhance, secure, and observe cloud-native applications with Istio, Linkerd, and Consul. Packt Publishing Ltd; 2020.
13. Thumala S. Building Highly Resilient Architectures in the Cloud. Nanotechnol Percept. 2020;16(2).
14. Raj P, Raman A, Raj P, Raman A. Automated multi-cloud operations and container orchestration. In: Software-Defined Cloud Centers: Operational and Management Technologies and Tools. 2018. p. 185–218.
15. Zhong Z, Buyya R. A cost-efficient container orchestration strategy in kubernetes-based cloud computing infrastructures with heterogeneous resources. ACM Trans Internet Technol. 2020;20(2):1–24.
16. Garrison J, Nova K. Cloud native infrastructure: patterns for scalable infrastructure and applications in a dynamic environment. O'Reilly Media, Inc.; 2017.
17. Gilbert J. Cloud Native Development Patterns and Best Practices: Practical architectural patterns for building modern, distributed cloud-native systems. Packt Publishing Ltd; 2018.
18. Laszewski T, Arora K, Farr E, Zonooz P. Cloud Native Architectures: Design high-availability and cost-effective applications for the cloud. Packt Publishing Ltd; 2018.
19. Zhang Y, Chen M. Cloud based 5G wireless networks.

Springer; 2016.
20. Nagaraju S, Parthiban L. Trusted framework for online banking in public cloud using multi-factor authentication and privacy protection gateway. J Cloud Comput. 2015;4(1):22.
21. Winn DC. Cloud Foundry: the cloud-native platform. O'Reilly Media, Inc.; 2016.
22. Reznik P, Dobson J, Gienow M. Cloud native transformation: practical patterns for innovation. O'Reilly Media, Inc.; 2019.
23. Toffetti G, Brunner S, Blöchlinger M, Spillner J, Bohnert TM. Self-managing cloud-native applications: Design, implementation, and experience. Future Gener Comput Syst. 2017;72:165–79.
24. Rose R. Hands-on serverless computing with Google Cloud: build, deploy, and containerize apps using Cloud Functions, Cloud Run, and cloud-native technologies. Packt Publishing Ltd; 2020.
25. Al Kiswani JHA. Smart-Cloud: A Framework for Cloud Native Applications Development [dissertation]. University of Nevada, Reno; 2019.
26. Jakóbczyk MT. Cloud-Native Architecture. In: Practical Oracle Cloud Infrastructure: Infrastructure as a Service, Autonomous Database, Managed Kubernetes, and Serverless. Springer; 2020. p. 487–551.
27. Chinamanagonda S. Cloud Migration Strategies and Best Practices. SSRN. 2019.
28. Gholami MF, Daneshgar F, Beydoun G, Rabhi F. Challenges in migrating legacy software systems to the cloud—an empirical study. Inf Syst. 2017;67:100–13.
29. Kratzke N. A brief history of cloud application architectures. Appl Sci. 2018;8(8):1368.
30. Davis C. Cloud Native Patterns: Designing Change-Tolerant Software. Simon and Schuster; 2019.
31. Adane M. Cloud computing adoption: Strategies for Sub-Saharan Africa SMEs for enhancing competitiveness. Afr J Sci Technol Innov Dev. 2018;10(2):197–207.
32. Dogo EM, Salami AF, Aigbavboa CO, Nkonyana T. Taking cloud computing to the extreme edge: A review of mist computing for smart cities and industry 4.0 in Africa. In: Edge computing: from hype to reality. 2019. p. 107–32.
33. Gillwald A, Moyo M, Odufuwa F, Frempong G, Kamoun F. The cloud over Africa. Res ICT Afr. 2014;1–33.
34. Yeboah-Boateng EO, Essandoh KA. Factors influencing the adoption of cloud computing by small and medium enterprises in developing economies. Int J Emerg Sci Eng. 2014;2(4):13–20.
35. Thota RC. Enhancing Resilience in Cloud-Native Architectures Using Well-Architected Principles. Int J Innov Res Eng Multidiscip Phys Sci. 2020;8:1–10.
36. Mosweu T, Mosweu O, Luthuli L. Implications of cloud-computing services in records management in Africa: Achilles heels of the digital era? S Afr J Inf Manag. 2019;21(1):1–12.
37. Alshamaila Y, Papagiannidis S, Li F. Cloud computing adoption by SMEs in the north east of England: A multi-perspective framework. J Enterp Inf Manag. 2013;26(3):250–75.
38. Williams B. The economics of cloud computing. Cisco Press; 2012.
39. Wakunuma K, Masika R. Cloud computing, capabilities and intercultural ethics: Implications for Africa. Telecommun Policy. 2017;41(7–8):695–707.
40. Clohessy T, Acton T, Morgan L. The impact of cloud-based digital transformation on IT service providers: evidence from focus groups. Int J Cloud Appl Comput. 2017;7(4):1–19.
41. Sabiri K, Benabbou F. Methods migration from on-premise to cloud. IOSR J Comput Eng. 2015;17(2):58–65.
42. Kshetri N. Cloud Computing in the Global South: drivers, effects and policy measures. Third World Q. 2011;32(6):997–1014.
43. Li M, Zhao D, Yu Y. TOE drivers for cloud transformation: direct or trust-mediated? Asia Pac J Mark Logist. 2015;27(2):226–48.
44. Bond J. The enterprise cloud: Best practices for transforming legacy IT. O'Reilly Media, Inc.; 2015.
45. Vainikka M. Migrating legacy applications to cloud: case TOAS. 2014.
46. Gade KR. Data Center Modernization: Strategies for transitioning from traditional data centers to hybrid or multi-cloud environments. Adv Comput Sci. 2019;2(1).
47. Marchioni F. Hands-on Cloud-native Applications with Java and Quarkus: Build High Performance, Kubernetes-native Java Serverless Applications. Packt Publishing Ltd; 2019.
48. Nosyk Y. Migration of a legacy web application to the cloud. 2018.
49. Indrasiri K, Suhothayan S. Design Patterns for Cloud Native Applications. O'Reilly Media, Inc.; 2021.
50. Scholl B, Swanson T, Jausovec P. Cloud native: using containers, functions, and data to build next-generation applications. O'Reilly Media; 2019.
51. El Khatib MM, Al-Nakeeb A, Ahmed G. Integration of cloud computing with artificial intelligence and Its impact on telecom sector—A case study. iBusiness. 2019;11(01):1.
52. Coutinho EF, de Carvalho Sousa FR, Rego PAL, Gomes DG, de Souza JN. Elasticity in cloud computing: a survey. Ann Telecommun. 2015;70:289–309.
53. Barnawi A, Sakr S, Xiao W, Al-Barakati A. The views, measurements and challenges of elasticity in the cloud: A review. Comput Commun. 2020;154:111–7.
54. Teagan R, Caleb L. Cost Optimization Strategies for Navigating the Economics of AWS Cloud Services. 2021.
55. Fowler B. AWS for Public and Private Sectors. Springer; 2023.
56. Kristiansson C, Lundström F. Cloud Computing Pricing and Deployment Efforts: Navigating Cloud Computing Pricing and Deployment Efforts: Exploring the Public-Private Landscape. 2023.
57. Ferrua S. The "Delta" Case: New AWS Data Platform Implementation [dissertation]. Politecnico di Torino; 2023.
58. Tatineni S. Cost Optimization Strategies for Navigating the Economics of AWS Cloud Services. Int J Adv Res Eng Technol. 2019;10(6):827–42.
59. Gade KR. Migrations: AWS Cloud Optimization Strategies to Reduce Costs and Improve Performance. 2022.
60. Deochake S. Cloud cost optimization: A comprehensive review of strategies and case studies. arXiv preprint arXiv:2307.12479. 2023.

61. Patel S. Migrating To the Cloud: A Step-By-Step Guide for Enterprise. 2023.
62. Armstrong J. Migrating to AWS: A Manager's Guide: how to Foster Agility, Reduce Costs, and Bring a Competitive Edge to Your Business. O'Reilly Media; 2020.
63. Whitehouse L, Buffington J. Amazon Web Services: Enabling Cost-Efficient Disaster Recovery Leveraging Cloud Infrastructure. Enterprise Strategy Group, White Paper. 2012.
64. Nama P. Cost management and optimization in automation infrastructure. Iconic Res Eng J. 2022;5(12):276–85.
65. Kambala G. Designing resilient enterprise applications in the cloud: Strategies and best practices. World J Adv Res Rev. 2023;17:1078–94.
66. Gade KR. Cost Optimization Strategies for Cloud Migrations. 2021.
67. Chinamanagonda S. Cost Optimization in Cloud Computing-Businesses focusing on optimizing cloud spend. J Innov Technol. 2020;3(1).
68. Jannapureddy R, Vien QT, Shah P, Trestian R. An auto-scaling framework for analyzing big data in the cloud environment. Appl Sci. 2019;9(7):1417.
69. Thota RC. Intelligent auto-scaling in AWS: Machine learning approaches for predictive resource allocation. Int J Sci Res Manag. 2022;10(10):1–8.
70. Sartoni M. AWS Services for Cloud Robotics Applications [dissertation]. Politecnico di Torino; 2022.
71. Qu C. Auto-scaling and deployment of web applications in distributed computing clouds [dissertation]. University of Melbourne, Australia; 2016.
72. Kampars J, Pinka K. Auto-scaling and adjustment platform for cloud-based systems. In: Environment. Technologies. Resources. Proceedings of the International Scientific and Practical Conference. 2017;2:52–7.
73. Bodhanya TA. Comparing cloud orchestrated container platforms: under the lenses of performance, cost, ease-of-use, and reliability. 2022.
74. Aziz WA, Ammar AA, Soliman JN. Auto Scaling Solutions for Cloud Applications. Int J Simul Syst Sci Technol. 2023;24(3).
75. Fernandez IG, Renjith JA. A novel approach on auto-scaling for resource scheduling using AWS. In: International Virtual Conference on Industry 4.0: Select Proceedings of IVCI4.0 2020. Springer; 2021. p. 99–109.
76. Singh P, Gupta P, Jyoti K, Nayyar A. Research on auto-scaling of web applications in cloud: survey, trends and future directions. Scalable Comput Pract Exp. 2019;20(2):399–432.
77. Thota RC. Cost optimization strategies for micro services in AWS: Managing resource consumption and scaling efficiently. Int J Sci Res Arch. 2023;10(2):1–12.
78. Bhagavathiperumal S. Auto scaling of cloud resources using time series and machine learning prediction [dissertation]. University of Technology Sydney (Australia); 2020.
79. Cristofaro T. Kube: a cloud ERP system based on microservices and serverless architecture [dissertation]. Politecnico di Torino; 2023.
80. Prasad VK, Dansana D, Bhavsar MD, Acharya B, Gerogiannis VC, Kanavos A. Efficient resource utilization in IoT and cloud computing. Information. 2023;14(11):619.
81. Pasham SD. AI-Driven Cloud Cost Optimization for Small and Medium Enterprises (SMEs). The Computertech. 2017;1–24.
82. Wong W, Zavodovski A, Corneo L, Mohan N, Kangasharju J. SPA: harnessing availability in the AWS spot market. In: IEEE INFOCOM 2021-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE; 2021. p. 1–6.
83. Pothecary R. Running Microsoft Workloads on AWS. Springer; 2021.
84. Purohit N, Srivastava P, Tripathi V, Mohd N. Spot Pricing in Cloud Computing: A Comprehensive Survey of Mechanisms, Strategies, and Future Directions. In: International Conference on Data Science and Network Engineering. Springer; 2023. p. 331–46.
85. Fabra J, Ezpeleta J, Álvarez P. Reducing the price of resource provisioning using EC2 spot instances with prediction models. Future Gener Comput Syst. 2019;96:348–67.
86. Radhika E, Sadasivam GS. A review on prediction based autoscaling techniques for heterogeneous applications in cloud environment. Mater Today Proc. 2021;45:2793–800.
87. Dogani J, Namvar R, Khunjush F. Auto-scaling techniques in container-based cloud and edge/fog computing: Taxonomy and survey. Comput Commun. 2023;209:120–50.
88. Chaudhari BY. A Cost-Effective And Practical Solution For AWS Resources Management With Usage Visualization [dissertation]. Dublin, National College of Ireland; 2023.
89. Jaishankar RK. Forecasting the price of AWS On-spot instances using Deep Neural Network Architectures [dissertation]. Dublin Business School; 2022.
90. Patterson S. Learn AWS Serverless Computing: A Beginner's Guide to Using AWS Lambda, Amazon API Gateway, and Services from Amazon Web Services. Packt Publishing Ltd; 2019.
91. Kodakandla N. Serverless Architectures: A Comparative Study of Performance, Scalability, and Cost in Cloud-native Applications. Iconic Res Eng J. 2021;5(2):136–50.
92. Sisák M. Cost-optimal AWS Deployment Configuration for Containerized Event-driven Systems [dissertation]. 2021.
93. Lin L, Pan L, Liu S. Methods for improving the availability of spot instances: A survey. Comput Ind. 2022;141:103718.
94. Miryala NK, Gupta D. Big Data Analytics in Cloud–Comparative Study. Int J Comput Trends Technol. 2023;71(12):30–4.
95. Thallam NST. Comparative Analysis of Public Cloud Providers for Big Data Analytics: AWS, Azure, and Google Cloud. Int J AI BigData Comput Manag Stud. 2023;4(3):18–29.
96. Sharma H. Effectiveness of CSPM in Multi-Cloud Environments: A study on the challenges and strategies for implementing CSPM across multiple cloud service providers (AWS, Azure, Google Cloud), focusing on interoperability and comprehensive visibility. Int J Comput Sci Eng Res Dev. 2020;10(1):1–18.
97. Raj P, Raman A, Raj P, Raman A. Multi-cloud

management: Technologies, tools, and techniques. In: Software-defined cloud centers: Operational and management technologies and tools. 2018. p. 219–40.
98. Ittmann HW. The impact of big data and business analytics on supply chain management. J Transp Supply Chain Manag. 2015;9(1):1–9.
99. Ghosh P, Biswas A, Ghosh S. Fundamentals and technicalities of big data and analytics. In: Intelligent systems in healthcare and disease identification using data science. Chapman and Hall/CRC; 2023. p. 51–106.
100. Olayinka OH. Leveraging Predictive Analytics and Machine Learning for Strategic Business Decision-Making and Competitive Advantage. Int J Comput Appl Technol Res. 2019;8(12):473–86.