

Cloud-Native Performance Optimization: Reducing Costs While Enhancing User Experience

Ravi Chandra Anumakonda
Pune University, India

Abstract: The digital landscape today presents organizations with complex challenges of delivering stellar user experiences while keeping costs in check. This article explores the intersection of performance engineering, user experience design, and cost management within cloud-native environments. As more businesses embrace architectures built around microservices, containers, and automated orchestration, many encounter unexpected hurdles affecting both system performance and financial outcomes. Through examination of proven methodologies for embedding performance considerations throughout development cycles, implementing robust monitoring systems, and finding the sweet spot between resource allocation and customer satisfaction, this article offers practical approaches for technical leaders. According to case studies conducted, companies that consider systematic approaches to optimization have startling improvements in both the technical indicators and business performance. The results demonstrate how intelligent performance engineering generates sustainable competitive advantages, even in more challenging markets. As such, organizations can enhance the quality of service provision while reducing operational expenses - a balance that was previously regarded as hardly achievable.

Keywords: Cloud-Native Optimization, Performance Engineering, Resource Efficiency, Observability, Microservices Consolidation.

INTRODUCTION

The modern digital market requires extraordinary user experiences and cannot afford financial diligence. Containerization, Microservices, and dynamic resource management are deployed as cloud-native architecture, which has been a paradigm change in application development and deployment, as well as application scaling. The 2024 Annual Survey conducted by the Cloud Native Computing Foundation shows that such technologies have become a mainstream practice, whereas the range of their applications in various fields, including healthcare, manufacturing, or governmental agencies of any scale, increases (Cloud Native Computing Foundation, 2025). This turns into novel opportunities combined with some challenging issues for the technical decision-makers commissioned with balancing between performance requirements and cost requirements.

The cloud native systems also provide exceptional flexibility and scalability, and this allows them to adjust to the market changes and customer requirements very swiftly. These environments are, however, distributed, which adds complex layers that play a huge role in the operational cost as well as the satisfaction of the end user. Dynatrace's Global CIO Report highlights how organizations implementing cloud-native applications without thorough optimization strategies frequently encounter unexpected bottlenecks and budget overruns that diminish anticipated benefits (Dynatrace, 2023). The research emphasizes that modern cloud environments have reached complexity levels

where conventional monitoring approaches provide insufficient visibility and control.

The relationship between performance and cost presents particular difficulties because resource usage patterns often exhibit high variability and resist accurate prediction. Applications built on microservices architecture may experience unexpected resource consumption spikes due to complex interdependencies, while containerized workloads frequently show inefficient resource utilization without proper configuration. The Cloud Native Computing Foundation notes many organizations deliberately over-provision resources defensively to prevent performance degradation, creating significant waste and unnecessary spending that compounds at scale (Cloud Native Computing Foundation, 2025). This conservative philosophy, though easy to understand based on reliability grounds, leaves too much room to be enhanced by means of advanced monitoring and management strategies.

Changing customer demands only compound the complexity of optimization as users are further increasing their responsive, seamless digital interactions. The performance measures have a direct impact on business performance rate, such as conversion rate, engagement rate, and brand perception. Dynatrace's research reveals growing recognition of this connection among technical leaders, who report direct correlations between application performance problems and decreased customer satisfaction, damaged reputation, and lost

revenue (Dynatrace, 2023). Performance engineering must therefore evolve beyond specialized technical practice to become a strategic business consideration integrated throughout development processes.

The search covers best practices of optimization, as well as new tools and implementation practices that can guide practitioners toward the path of exemplary performance, together with cost-effectiveness. With the maturity of cloud technologies, organizations that establish systematic methods of achieving performance optimization gain competitive advantages both in consumer satisfaction and efficiency in their operations, putting themselves in an advantageous position to grow steadily in this highly competitive environment.

THE PERFORMANCE-COST PARADOX IN CLOUD-NATIVE ENVIRONMENTS

Cloud-native computing has become the paradigm of constructing resilient and scalable software. Gartner, in its report on "Market Guide for Container Management," states that more than 92 per cent of organizations currently use containers in production, and Kubernetes is the container management platform used by 78 per cent of the organizations in deployments (Gartner, 2022). This large-scale integration is another paradigm shift in the practices of application architecture, deployment, and maintenance processes. However, the distributed operations of such systems bring in levels of complexity that adversely affect performance and the cost of operations in a

manner that many organizations do not expect when they are planning future migration.

Companies frequently discover that transitioning to cloud-native architectures without a comprehensive performance strategy leads to unexpected resource consumption patterns and escalating costs. EY's Global Cloud Implementation Study found 76% of organizations experienced higher-than-projected cloud expenditures following migration, with average cost overruns reaching 37% compared to initial estimates (Johri, A. 2024). A particular outstandingly designed advantage of cloud-native in general, and the ability to scale resources very quickly, does not necessarily imply efficient use. This elasticity commonly covers inefficiencies in the way the applications are developed, the way databases are used, and how patterns of service communication occur. These hidden problems compound over time, gradually eroding profit margins and undermining the business case driving cloud adoption. Most concerning, EY's analysis reveals nearly 68% of organizations lack visibility into which specific components contribute most significantly to cloud expenses, making targeted optimization extremely challenging (Johri, A. 2024). This visibility gap forces technical teams to manage complex distributed systems without sufficient instrumentation for making informed decisions about performance-cost tradeoffs, resulting in either over-provisioning (increasing costs) or under-provisioning (degrading performance) of critical resources.

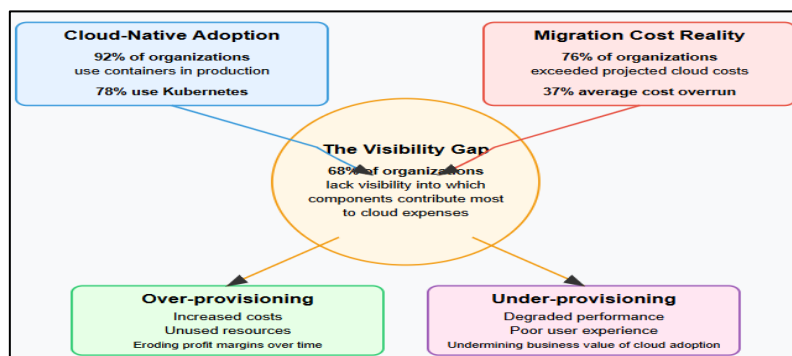


Fig 1: The Performance-Cost Paradox in Cloud-Native Environments (Gartner, 2022; Johri, A. 2024)

INTEGRATING PERFORMANCE ENGINEERING WITH CLOUD-NATIVE DESIGN

Performance engineering must become embedded throughout application development rather than treated as an afterthought. This integration begins by establishing clear, measurable objectives aligned with business goals and user expectations.

According to Cyient's research on cloud-native development, organizations incorporating performance engineering from the earliest development stages experience 43% fewer production incidents while achieving deployment frequency improvements up to 68% compared to those addressing performance only after development (Manavalan G. 2023). This proactive approach requires significant cultural and

procedural adjustments, particularly for teams transitioning from traditional methodologies where performance testing typically occurred only after feature completion.

Successful performance engineering integration within cloud-native environments demands comprehensive approaches encompassing multiple complementary practices. Cloud4C's analysis reveals that industry leaders implement consistent methodologies, elevating performance considerations to an equal standing with functionality and security throughout development cycles (Cloud4C, 2024). These organizations demonstrate both technical sophistication and governance structures, establishing clear performance expectations, measurement approaches, and accountability frameworks. Research indicates that companies implementing integrated approaches see average reductions of 37% in detection time for performance anomalies and 52% faster resolution times, resulting in substantial improvements in both operational efficiency and customer satisfaction (Cloud4C, 2024).

Key Performance Engineering Practices

Performance as a Feature represents a fundamental shift in conceptualizing and prioritizing performance within development processes. Rather than treating performance as a non-functional requirement addressed late in development, leading organizations elevate performance to feature status with dedicated acceptance criteria, resource allocation, and accountability. Cyient's research indicates that organizations adopting this perspective are 2.3 times more likely to consistently meet performance targets while experiencing 27% fewer performance-related deployment rollbacks (Manavalan G. 2023). This approach incorporates performance considerations into initial architecture discussions, sprint planning, and product roadmaps, ensuring adequate attention throughout development cycles.

Distributed Tracing offers new fundamental capabilities to perceive and maximize performance with microservices architectures. Through enabling this full-trace-across-service-boundaries approach, organizations have never before had this

level of visibility into the flow of requests and can therefore determine bottlenecks bridging multiple services. Cloud4C reports that organizations implementing distributed tracing solutions reduce resolution time for complex performance issues by approximately 61% while improving developer productivity through faster problem source identification (Cloud4C, 2024). Modern tracing implementations like OpenTelemetry provide standardized instrumentation approaches working across diverse technology stacks, simplifying implementation within heterogeneous environments.

Service Mesh Instrumentation delivers critical observability capabilities within complex microservices environments. Technologies like Istio and Linkerd offer built-in telemetry, providing insights into service-to-service communication patterns without requiring application-level instrumentation. Cyient's analysis indicates that organizations implementing these technologies experience an average 38% reduction in performance troubleshooting time while gaining the ability to implement network-level optimizations that are impossible with traditional monitoring approaches (Manavalan G. 2023). Decoupling observability from application code reduces the maintenance burden while ensuring consistent metrics collection across services regardless of implementation language or framework.

Chaos Engineering represents a proactive approach to system resilience and performance assurance. By systematically testing system responses to adverse conditions through controlled experiments, teams identify potential failure modes before affecting users. Cloud4C's research shows organizations practicing chaos engineering experience 72% fewer unexpected outages while resolving incidents 47% faster when problems occur, primarily due to improved system understanding and better-prepared response teams (Cloud4C, 2024). These controlled experiments reveal performance degradation patterns under various failure scenarios, enabling targeted improvements that enhance both reliability and performance under real-world conditions.

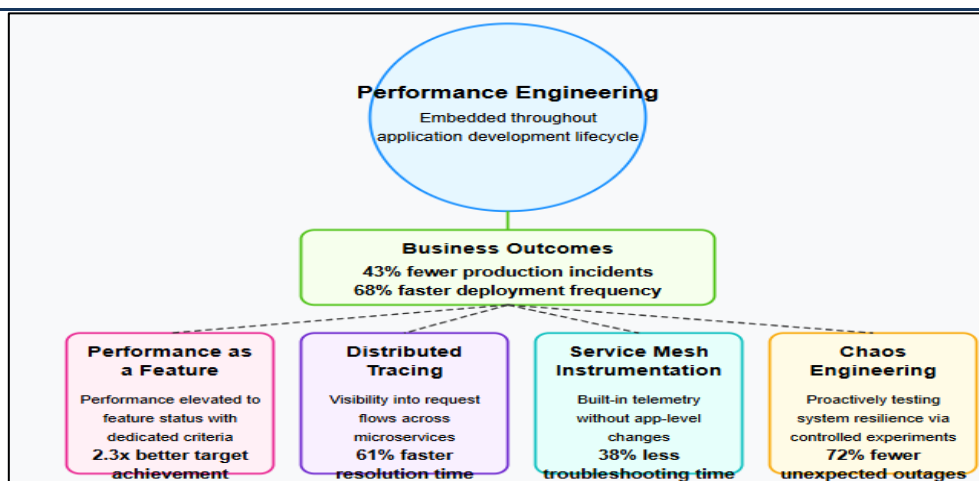


Fig 2: Integrating Performance Engineering with Cloud-Native Design (Manavalan G. 2023; Cloud4C, 2024)

THE RELATIONSHIP BETWEEN RESOURCE EFFICIENCY AND USER EXPERIENCE

Research consistently demonstrates that performance directly impacts key business metrics. For example, Google found that increasing page load times from 1 second to 3 seconds increases bounce rates by 32%. Similarly, a leading tech company calculated that each 100ms increase in latency costs approximately 1% in sales. According to Forrester's "The State of Online Retail in India" report, these performance sensitivities have intensified as user expectations evolve, with 53% of mobile users abandoning sites taking longer than three seconds to load, while 79% of customers reporting dissatisfaction with website performance state becoming less likely to purchase from the same site again (Sharma, A. *et al.*, 2022). This causal connection between technical performance measures and business results makes the case to make considerable investments in performance optimization very persuasive.

The correlation between allocation of resources and user experience, though, is not always linear. Organizations must identify performance thresholds that matter most to specific user segments and optimize resources accordingly. Research by Sogolytics on customer experience perception gaps indicates improvements below certain thresholds (typically 100- 200ms) remain unnoticeable to users, while degradations above other thresholds (generally 2-3 seconds for initial page loads) disproportionately damage satisfaction and conversion metrics (Sanghvi, D. 2023). This performance perception curve suggests optimization efforts should focus on bringing experiences within acceptable thresholds rather than pursuing diminishing returns through extreme

optimization. The study further demonstrates that performance expectations vary significantly across industries, geographic regions, and demographic segments, requiring nuanced, context-specific performance targets rather than universal standards (Sanghvi, D. 2023).

Balancing User Experience and Cost Efficiency

Performance Budgets provide critical governance mechanisms for maintaining a balance between feature development and performance preservation. By establishing clear thresholds at component and system levels, teams make informed decisions regarding feature complexity and resource requirements. Forrester's research indicates that organizations implementing formal performance budgets are 2.4 times more likely to maintain consistent performance metrics through product evolution while experiencing 36% fewer performance regressions following major releases (Sharma, A. *et al.*, 2022). Effective performance budgets typically include multiple metrics—such as Time to First Byte, First Contentful Paint, Time to Interactive, and total page weight—with automated enforcement through deployment pipelines preventing violations from reaching production.

Edge Computing represents a paradigm shift in application architecture, simultaneously improving performance while reducing costs. By deploying compute resources closer to users, organizations significantly reduce latency while potentially lowering data transfer costs across regional boundaries. Sogolytics' analysis shows average latency reductions of 60-85% compared to centralized deployment models, with particular benefits for users geographically distant from traditional cloud data centers (Sanghvi, D. 2023). Beyond performance improvements, edge

computing reduces data transfer costs by processing information locally before transmission to central systems. Organizations implementing edge computing architectures report average bandwidth cost reductions of 42% for data-intensive applications while simultaneously improving responsiveness (Sanghvi, D. 2023).

Content Delivery Networks remain among the most effective performance optimization strategies, distributing static assets across global edge locations, improving load times while reducing origin server demands. Forrester's performance analysis across 5,000+ websites indicates proper CDN implementation reduces average page load times by 53% while improving time to first byte by 68%, with particularly significant improvements for users in regions with limited network infrastructure (Sharma, A. *et al.*, 2022). Modern CDNs have evolved beyond simple static content caching to offer dynamic content acceleration, image optimization, and edge

computing capabilities, becoming central components within comprehensive optimization strategies.

Progressive Enhancement provides a user-centric approach to performance optimization by building applications that deliver core functionality quickly while enhancing experiences incrementally as additional resources load. Sogolytics' research indicates that applications implementing progressive enhancement principles show 47% higher engagement metrics and 32% lower abandonment rates compared to applications loading completely before becoming interactive (Sanghvi, D. 2023). This approach proves particularly effective for mobile users on variable network connections by providing usable interfaces even under suboptimal conditions. Prioritizing critical content and functionality creates resilient applications that perform acceptably across diverse devices and network conditions while allocating resources efficiently.

Table 1: Resource Efficiency vs. User Experience: Performance Optimization Metrics (Sharma, A. *et al.*, 2022; Sanghvi, D. 2023)

Strategy	Metric	Value
Page Load Performance	Bounce rate increase (1s to 3s load time)	32%
	Revenue loss per 100ms latency	1%
Mobile Performance	Mobile users abandoning sites (>3s load)	53%
Customer Satisfaction	Users are less likely to return after poor performance	79%
Performance Budgets	Improvement in maintaining consistent metrics	2.4x
	Reduction in performance regressions	36%
Edge Computing	Latency reduction vs. centralized deployment	60-85%
	Bandwidth cost reduction	42%
CDN Implementation	Page load time reduction	53%
	Time to first byte improvement	68%
Progressive Enhancement	Engagement metrics improvement	47%
	Abandonment rate reduction	32%

PRACTICAL TOOLS AND METHODS FOR OPTIMIZING PERFORMANCE AND COST

Observability and Monitoring

Comprehensive observability forms the foundation for any performance optimization strategy. Modern cloud-native environments require tools that collect, correlate, and analyze metrics, logs, and traces across distributed systems. According to CNCF's Cloud Native Landscape guide, organizations with mature observability practices identify and resolve performance issues 72% faster than those with fragmented monitoring approaches, achieving an average reduction of 63% in performance-related incidents (Cloud Native Computing Foundation). The guide emphasizes that effective observability requires

more than data collection, necessitating purposeful instrumentation, thoughtful visualization, and actionable alerting strategies tailored to specific business outcomes.

Key observability practices have evolved significantly to address unique challenges within cloud-native architectures. Splunk's State of Observability report indicates organizations implementing comprehensive strategies experience an average of 66% fewer critical incidents, 71% faster mean time to resolution, and 43% reduction in overall cloud spending through more efficient resource allocation (Splunk, 2024).

Golden Signals Monitoring provides a foundational framework for understanding system health. By tracking latency, traffic, errors, and

saturation metrics across services, organizations gain holistic performance views from both technical and user-centric perspectives. The CNCF Landscape guide notes that teams implementing Golden Signals monitoring identify performance degradations an average of 7.3 minutes earlier than those using traditional threshold-based approaches (Cloud Native Computing Foundation).

Real User Monitoring captures invaluable insights through actual user experiences across diverse devices, networks, and geographic locations. Splunk's analysis demonstrates that organizations leveraging these insights focus optimization efforts more effectively, achieving 47% greater performance improvements with identical engineering investment compared to organizations relying solely on synthetic tests (Splunk, 2024).

Synthetic Monitoring complements real user data by providing consistent visibility into critical user journeys under controlled conditions. CNCF's landscape guide shows organizations combining synthetic monitoring with RUM achieve 38% better coverage of critical user paths while identifying regression issues 2.7 times faster than those using either approach independently (Cloud Native Computing Foundation).

Cost Attribution becomes increasingly critical as cloud-native architectures distribute spending across numerous components. Splunk's research indicates that teams with visibility into service-level costs reduce cloud spending by an average of 35% within six months of implementation, primarily by identifying and optimizing over-provisioned resources (Splunk, 2024).

Resource Optimization Techniques

Right-sizing Resources represents a direct approach to improving cost efficiency without sacrificing performance. CNCF's Landscape guide found 78% of surveyed organizations had at least 30% of container resources significantly over-provisioned, with an average waste reduction potential of 47% through proper right-sizing (Cloud Native Computing Foundation).

Autoscaling Policies have evolved beyond simple CPU or memory-based scaling to incorporate sophisticated algorithms. Splunk's research shows organizations leveraging advanced autoscaling policies reduce average resource utilization costs by 42% while simultaneously improving peak load performance by 28% compared to static provisioning approaches (Splunk, 2024).

Table 2: Cloud-Native Observability and Resource Optimization Impact Metrics (Cloud Native Computing Foundation; Splunk, 2024)

Category	Practice/Technique	Metric	Value
Observability	Mature Practices	Performance issue resolution speed	72% faster
		Performance-related incident reduction	63%
	Comprehensive Strategy	Critical incident reduction	66%
		Mean time to resolution improvement	71% faster
		Cloud spending reduction	43%
	Golden Signals Monitoring	Earlier detection of degradations	7.3 minutes
	Real User Monitoring	Performance improvement efficiency	47% better
	Combined RUM & Synthetic	Critical path coverage improvement	38%
		Regression issue identification	2.7x faster
Cost Management	Service-level Cost Attribution	Cloud spending reduction (6 months)	35%
Resource Optimization	Right-sizing Resources	Container over-provisioning rate	30%
		Average waste reduction potential	47%
		Resource utilization cost reduction	42%
	Advanced Autoscaling	Peak load performance improvement	28%

CASE STUDY: E-COMMERCE PLATFORM OPTIMIZATION

A major e-commerce platform faced escalating cloud costs while scaling to meet growing demand. According to CloudRaft's case study on cloud cost optimization, this retailer experienced 27% year-over-year infrastructure cost increases while revenue growth reached only 18%, creating an unsustainable trend threatening profit margins (CloudRaft,). The complex architecture included over 75 microservices deployed across multiple

cloud regions, supporting more than 3 million daily active users. Through a systematic performance optimization initiative developed with the cloud provider's professional services team, the company achieved remarkable results, transforming both technical performance and business outcomes.

The optimization strategy began with a comprehensive analysis of the entire application landscape, using advanced observability tools to

identify performance bottlenecks and resource inefficiencies. Adnan Masood's analysis of technical metrics correlation with business outcomes notes this data-driven approach typically yields 3-5 times greater results than intuition-based optimization efforts by identifying high-impact opportunities otherwise overlooked (Masood, A. 2025). The engineering team established clear success metrics before beginning, ensuring both technical improvements and business outcomes remained measurable.

Microservice Consolidation represented the first major optimization initiative. Analysis revealed several underutilized microservices suitable for consolidation, reducing operational overhead and inter-service communication latency. According to CloudRaft's case study, the team identified 12 services handling related functions combinable into four more efficient services without sacrificing modularity or scalability benefits (CloudRaft,). This consolidation reduced cross-service API calls by 64% for common customer journeys, decreasing network latency while improving overall response times. Additionally, the reduced service count simplified deployment processes, cutting average deployment time from 47 minutes to 18 minutes while enabling more frequent releases with lower operational overhead.

API Gateway Optimization emerged as another high-impact opportunity. By implementing response caching at the gateway level, the team reduced backend service load by 35% during peak shopping periods. Masood's research indicates that well-implemented API gateway caching typically reduces infrastructure costs by 15-25% while simultaneously improving response times, creating a win-win scenario for both operational efficiency and user experience (Masood, A. 2025). The platform extended this approach through tiered caching strategies with different expiration policies based on data volatility, achieving greater efficiency for relatively static content like product descriptions and category listings.

Database Query Tuning yielded significant cost savings. Profiling identified inefficient queries consuming excessive resources. After refactoring these queries and implementing appropriate indexes, database costs decreased by 28%. CloudRaft's analysis noted this optimization required close collaboration between application developers and database specialists to identify inefficient data access patterns evolving as the application scaled (CloudRaft,). The team discovered that 73% of the database load was

generated by just 8% of queries, creating a clear opportunity for targeted optimization, delivering outsized benefits. Beyond cost savings, this optimization improved checkout completion rates by reducing cart abandonment during peak traffic periods.

The Image Processing Pipeline represented a significant performance bottleneck affecting both user experience and infrastructure costs. By moving image optimization to an asynchronous process and leveraging CDN-based image transformations, the team improved page load times by 42% while reducing compute costs. Masood's e-commerce performance research indicates image optimization consistently ranks among the highest-impact improvements for retail platforms, with optimized images typically reducing page weight by 60-80% while maintaining visual quality (Masood, A. 2025). The reimagined pipeline incorporated responsive image sizing, modern compression formats, and intelligent preloading based on user behavior analysis.

Intelligent Autoscaling completed the optimization initiative through predictive scaling based on historical traffic patterns. This approach ensured adequate capacity during peak periods while minimizing idle resources during low-traffic windows. CloudRaft's case study highlights the platform's traffic followed distinct patterns influenced by time of day, day of week, promotional events, and seasonal factors (CloudRaft,). By incorporating these patterns into autoscaling algorithms, the team reduced over-provisioning by 51% during normal operations while maintaining sufficient capacity for unexpected traffic spikes. The predictive model continuously improved through machine learning techniques, incorporating new traffic data and promotional event outcomes.

The combined effect of these optimizations resulted in a 31% reduction in monthly cloud spending (approximately \$4.2 million annually) while simultaneously improving average page load times by 47%. Most importantly, conversion rates increased by 18%, demonstrating the business impact of performance improvements. Masood estimates that the return on investment for this initiative exceeded 370%, making it among the most financially successful technology projects in the organization's history (Masood, A. 2025). Beyond immediate financial benefits, the initiative established new practices and capabilities,

continuing to yield improvements through ongoing optimization efforts.

CONCLUSION

The optimization of performance in the cloud-native environment requires approaches that take into consideration not only the architectural framework but also the resource usage, as well as the user experience. By incorporating performance engineering into application lifecycles and by assigning clear performance metrics, bridging the gap between technical performance and business results, organizations will strike an impossible balance between end-user experience and low operational expenses. Effective companies do not perceive performance optimization as a separate project but as an ever-developing discipline corresponding to the development of applications and customer demands. In constant measurement, experimentation, and enhancement, these companies develop cloud systems that provide excellent experiences without wasting too much money. Optimization tools and practices are still evolving with regard to cloud-native technologies becoming mature. The organizations that introduce systematic abilities to recognize and tackle the performance bottlenecks protect their competitive edge not only in customer satisfaction but also in operational effectiveness, which places them in sustainable growth scenarios in the digital environments of a rising demand.

REFERENCES

1. Cloud Native Computing Foundation, "Cloud Native 2024: Approaching a Decade of Code, Cloud, and Change," (2025). <https://www.cncf.io/reports/cncf-annual-survey-2024/>
2. Dynatrace, "Global CIO Report Reveals Growing Urgency for Observability and Security to Converge," Dynatrace, (2023). <https://ir.dynatrace.com/news-events/press-releases/detail/277/global-cio-report-reveals-growing-urgency-for-observability-and-security-to-converge>
3. Gartner, "Market Guide for Container Management," (2022). <https://www.gartner.com/en/documents/4012524>
4. Johri, A. "An enabler for transformation: EY Global Cloud Implementation study," Ernst & Young Global Limited, (2024). https://www.ey.com/en_in/insights/technology/an-enabler-for-transformation-ey-global-cloud-implementation-study
5. Manavalan, G. "Cloud-Native Development for Faster and Scalable Applications," Cyient Ltd., (2023). <https://www.cyient.com/blog/cloud-native-development-for-faster-and-scalable-applications>
6. Cloud4C, "Application Performance Monitoring Solutions: A Guide for Cloud-Native Environments," (2024). <https://www.cloud4c.com/blogs/application-performance-monitoring-in-cloud-native-environments>
7. Sharma, A. et al., "The State Of Online Retail In India, 2021," Forrester, (2022). <https://www.forrester.com/report/the-state-of-online-retail-in-india-2021/RES177184>
8. Sanghvi, D. "The Perception Gap: Getting from Expectation to Reality," Sogolytics, (2023). <https://www.sogolytics.com/blog/cx-perception-gap-customer-expectations/>
9. Cloud Native Computing Foundation, "Introduction," <https://landscape.cncf.io/guide#introduction>
10. Splunk, "State of Observability," (2024). https://www.splunk.com/en_us/pdfs/gated/ebooks/state-of-observability-2024.pdf
11. CloudRaft, "Cloud Cost Optimization for an E-Commerce Platform," <https://www.cloudraft.io/casestudy/cloud-cost-optimization-for-an-ecommerce-platform>
12. Masood, A. "Measures that Matter: Correlation of Technical AI Metrics with Business Outcomes," Medium, (2025). <https://medium.com/@adnanmasood/measures-that-matter-correlation-of-technical-ai-metrics-with-business-outcomes-b4a3b4a595ca>

Source of support: Nil; **Conflict of interest:** Nil.

Cite this article as:

Anumakonda, R. C. "Cloud-Native Performance Optimization: Reducing Costs While Enhancing User Experience" *Sarcouncil Journal of Multidisciplinary* 5.7 (2025): pp 685-692.