Stout in Computer Science and Technology Studies (SCSTS)

DOI: https://doi.org/10.61424/9hpkf784
Journal Homepage: https://stoutjournals.org/index.php/SCSTS

Generative AI Application Deployment on Microsoft Azure: Performance, Security, and Cost Considerations

Shekar Rao Lakavath
Independent Researcher, Bonney Lake, Washington, USA
✉ **Corresponding Author**: Shekar Rao Lakavath, **E-mail**: Shekar.lakavath@gmail.com

ARTICLE INFORMATION

Submitted: April 22, 2025
Accepted: June 15, 2025
Published: August 29, 2025
Volume: 1
Issue: 1

KEYWORDS

Generative artificial intelligence, Microsoft Azure, cloud computing, large language models, cloud security, cloud cost optimisation, MLOps

ABSTRACT

The rapid adoption of generative artificial intelligence (AI) has pushed organisations to look beyond experimentation and toward production-grade deployment on public cloud infrastructure. Microsoft Azure has emerged as one of the leading platforms for hosting generative AI applications through services such as Azure OpenAI Service, Azure Machine Learning, Azure Kubernetes Service, and Azure Container Apps. However, moving a generative AI workload from prototype to production surfaces three interconnected challenges: sustaining acceptable inference performance under variable load, protecting sensitive data and model assets against an expanding attack surface, and controlling the often unpredictable cost of large-model inference. This paper synthesises the current literature on cloud computing, large language model (LLM) serving, cloud security, and cloud financial management to propose an integrated framework for evaluating generative AI deployments on Azure. We describe a reference architecture spanning identity, orchestration, model serving, data grounding, and cost-and-operations layers; compare the performance and scalability profiles of five common Azure deployment options; and summarise seven cost-optimisation levers available to architects. The analysis shows that performance, security, and cost decisions are not independent: provisioning choices made to reduce latency (for example, reserved throughput) directly increase fixed cost, while security controls such as private networking and content filtering introduce measurable latency overhead. We conclude that a mature Azure-based generative AI deployment requires continuous, cross-functional governance—combining MLOps, FinOps, and security operations—rather than a one-time architectural decision, and we outline directions for future empirical research.

1. Introduction

Generative AI systems, including large language models (LLMs) and diffusion-based image generators, have moved from research curiosities to core components of enterprise software within a few years of the release of transformer-based architectures (Vaswani et al., 2017) and large-scale pretrained models such as GPT-3 (Brown et al., 2020). Enterprises increasingly consume these capabilities not by training models from scratch but by deploying and orchestrating them on public cloud platforms, which provide the elastic compute, storage, and managed AI services required to operationalise generative AI at scale (Patel et al., 2024).

Microsoft Azure is one of a small number of hyperscale providers—alongside Amazon Web Services and Google Cloud—capable of supporting the full generative AI lifecycle, from data preparation and model fine-tuning to inference serving and monitoring (Mell & Grance, 2011; Patel et al., 2024). Azure's position is reinforced by its native integration of the Azure OpenAI Service, which exposes state-of-the-art foundation models through the same governance, identity, and networking controls used across the broader Azure ecosystem. This integration is attractive to regulated industries but also concentrates architectural decisions that were previously distributed across separate systems into a single deployment surface, making trade-offs between performance, security, and cost more visible and more consequential.

Copyright: © 2025 The Author(s). This open access article is distributed under the terms and conditions of the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC- BY-NC-SA) License (https://creativecommons.org/licenses/by-nc-sa/4.0/), published by Stout Journals, London, United Kingdom

Despite widespread industry interest, much of the available guidance on deploying generative AI on Azure is vendor-authored documentation or informal practitioner commentary rather than peer-reviewed research. At the same time, the academic literature on cloud computing, LLM inference serving, cloud security, and cloud financial operations (FinOps) has matured substantially, offering transferable principles even where Azure-specific empirical studies remain scarce. This paper bridges that gap by synthesising findings from cloud architecture, MLOps, cloud cost optimisation, and AI security research into a coherent framework specifically oriented toward Azure-based generative AI deployment.

The remainder of the paper is organised as follows. Section 2 reviews related literature across four themes: cloud computing foundations, generative AI and LLM serving, cloud and AI security, and cloud cost management. Section 3 presents a reference architecture for generative AI deployment on Azure. Section 4 examines performance considerations across common Azure deployment options. Section 5 discusses security considerations specific to generative AI workloads. Section 6 analyses cost considerations and optimisation levers. Section 7 discusses the interdependencies among the three dimensions, Section 8 identifies limitations and future research directions, and Section 9 concludes.

2. Literature Review

2.1 Cloud Computing Foundations

The formal definition of cloud computing provided by the U.S. National Institute of Standards and Technology characterises it as a model for enabling on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned with minimal management effort (Mell & Grance, 2011). This elasticity is what makes cloud platforms attractive for generative AI, whose compute demands can vary by orders of magnitude between the training, fine-tuning, and inference phases. Container orchestration technologies, particularly Kubernetes, have become the de facto standard for packaging and scaling such workloads (Bernstein, 2014; Carrión, 2022), and Azure Kubernetes Service extends this standard with Azure-managed control planes and GPU-enabled node pools. Complementary serverless computing models reduce operational overhead further by abstracting away infrastructure management entirely, at the cost of reduced control over performance predictability (Li, Guo, et al., 2022).

2.2 Generative AI and LLM Serving

The transformer architecture (Vaswani et al., 2017) and the subsequent demonstration that scaling transformer-based language models yields strong few-shot performance (Brown et al., 2020) established the technical foundation for the current generation of generative AI services, including Azure OpenAI Service. Serving these models efficiently in production, however, is a distinct engineering problem from training them. Recent surveys document a wide range of system-level techniques—key-value caching, dynamic batching, quantisation, and speculative decoding—that reduce inference latency and cost without altering model outputs (Li, Jiang, et al., 2024; Miao et al., 2023). These findings are directly relevant to Azure deployments, since the choice between shared and dedicated (provisioned) capacity on Azure OpenAI Service is, in effect, a choice about which of these system-level optimisations the platform applies on the customer's behalf versus what the customer must manage independently through Azure Machine Learning or Azure Kubernetes Service.

2.3 Cloud and AI Security

Security research on cloud systems has historically focused on reference architectures that separate concerns of identity, network isolation, and data protection across infrastructure, platform, and software service layers (Fernandez & Monge, 2014). These principles remain applicable to generative AI workloads, but generative AI introduces additional, model-specific risks. Large language models are documented to be vulnerable to prompt injection, jailbreaking, data poisoning, and unintended data leakage through model outputs (Liu & Hu, 2024). Because generative AI applications frequently process personal or proprietary data as part of retrieval-augmented generation pipelines, they also inherit the compliance obligations associated with regulations such as the General Data Protection Regulation (GDPR); recent work proposes architectural patterns, including sticky data-usage policies and continuous cloud auditing, to make GDPR compliance verifiable rather than merely declarative (Barati et al., 2021; Cambronero et al., 2024). Privacy-preserving training techniques such as differentially private stochastic gradient descent further illustrate that security and privacy protections can be embedded directly into model training rather than added only at the infrastructure boundary (Abadi et al., 2016).

2.4 Cloud Cost Management

The pay-as-you-go pricing model that underpins public cloud computing is a double-edged sword: it removes upfront capital expenditure but can produce highly variable operating costs, a problem that is amplified for generative AI because GPU-backed inference capacity is substantially more expensive than general-purpose compute (Deochake, 2023). The FinOps discipline has emerged to address this by embedding cost accountability into engineering workflows rather than treating it as a purely financial afterthought. Formal cost-optimisation models for cloud service deployment demonstrate that operational cost can be reduced significantly through better matching of workload characteristics to deployment configuration (Khan et al., 2024; Stupar

& Huljenic, 2023), while stochastic optimisation approaches show that resource-type selection under uncertain demand is a solvable, quantifiable problem rather than a matter of guesswork (Qu et al., 2024). For machine learning workloads specifically, MLOps practices—covering automated pipelines, model versioning, and continuous monitoring—have been shown to reduce both the operational burden and the cost of maintaining production models over their lifecycle (Hewage & Meedeniya, 2022; Kreuzberger et al., 2023).

Taken together, the literature suggests that performance, security, and cost in cloud-hosted generative AI systems are governed by overlapping architectural decisions rather than independent levers, motivating the integrated treatment adopted in the remainder of this paper.

3. A Reference Architecture for Generative AI Deployment on Azure

Figure 1 presents a reference architecture synthesised from the cloud architecture and MLOps literature reviewed above (Bernstein, 2014; Fernandez & Monge, 2014; Kreuzberger et al., 2023) and mapped onto specific Azure services. Client requests enter through Azure Front Door or Azure API Management, which provide global routing, throttling, and web application firewall protection. Authentication and authorisation are handled by Microsoft Entra ID before requests reach an orchestration layer implemented in Azure App Service, Azure Container Apps, or Azure Kubernetes Service, depending on control and scaling requirements. The orchestration layer coordinates two downstream concerns: a model-serving layer (Azure OpenAI Service or Azure Machine Learning managed online endpoints) and a data-grounding layer (Azure AI Search, Cosmos DB, and Blob Storage) that supports retrieval-augmented generation. Three cross-cutting layers—security and governance, cost and operations, and MLOps/LLMOps—apply uniformly across the model-serving and data layers rather than sitting at a single point in the request path.

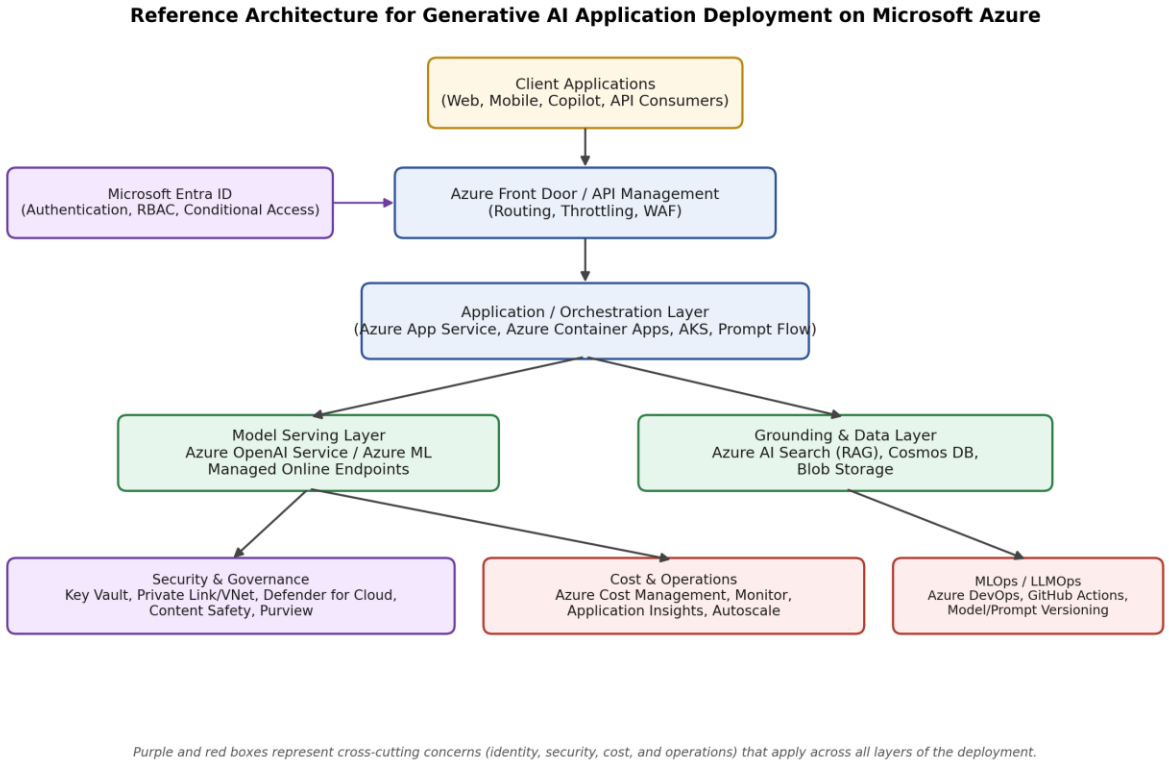


Figure 1. Reference architecture for generative AI application deployment on Microsoft Azure, showing the request path (client → gateway → orchestration → model/data layers) and the cross-cutting identity, security, cost, and MLOps concerns.

This layered separation is important for the remainder of the analysis because it clarifies that performance, security, and cost decisions are made at different, but interacting, layers: performance decisions are concentrated in the model-serving layer, security decisions span the gateway and cross-cutting governance layer, and cost decisions are distributed across nearly every layer through compute selection and autoscaling policy.

4. Performance Considerations

Performance in a generative AI application is typically evaluated along three axes: latency (time to first token and time to complete response), throughput (requests or tokens processed per unit time), and consistency (variance in latency under load). Azure offers several distinct deployment options for hosting generative AI models, each with a different performance profile, summarised in Table 1.

Deployment Option	Description	Typical Latency Profile	Scalability Model	Best-Fit Workload
Azure OpenAI Service – Pay-as-you-go	Shared multi-tenant capacity billed per token	Variable; subject to shared-capacity throttling	Elastic, provider-managed	Prototyping, low/variable traffic
Azure OpenAI Service – Provisioned Throughput Units (PTU)	Reserved, dedicated model capacity	Stable, predictable low latency	Fixed capacity; scales via additional PTUs	High-volume, latency-sensitive production traffic
Azure Machine Learning Managed Online Endpoints	Custom or open-source model hosting on managed compute	Depends on instance SKU and batching	Autoscaling instance pools	Fine-tuned or custom generative models
Azure Kubernetes Service (AKS) with GPU node pools	Self-managed containerized inference on GPU VMs	Low, tunable via batching and caching	Horizontal pod/node autoscaling	Complex multi-model pipelines, fine-grained control
Azure Container Apps (serverless GPU workload profiles)	Serverless containers with scale-to-zero	Cold-start latency on scale-from-zero	Scale-to-zero, event-driven	Bursty or intermittent inference workloads

Table 1. Comparison of common Azure deployment options for generative AI inference.

The pay-as-you-go tier of Azure OpenAI Service offers the fastest path to production but shares model capacity across tenants, which can introduce variable latency and rate-limiting under load. Provisioned Throughput Units address this by reserving dedicated model capacity, producing more predictable latency at the cost of reduced elasticity—an early illustration of the performance–cost trade-off examined further in Section 6. For organisations deploying fine-tuned or open-source models rather than first-party Azure OpenAI models, Azure Machine Learning managed online endpoints and Azure Kubernetes Service provide more granular control over batching, quantisation, and hardware selection, consistent with system-level optimisation techniques documented in the LLM-serving literature (Li, Jiang, et al., 2024; Miao et al., 2023). Serverless options such as Azure Container Apps with GPU workload profiles trade a degree of latency—primarily through cold-start delay when scaling from zero—for substantially reduced idle cost, making them well suited to bursty or intermittent workloads rather than latency-critical, always-on services.

A recurring theme across the LLM-serving literature is that many of the most effective performance interventions—key-value cache management, dynamic request batching, and speculative decoding—operate below the level of the deployment option itself (Li, Jiang, et al., 2024). On Azure OpenAI Service, these optimisations are largely applied by the platform and are not directly configurable by the customer, whereas on Azure Machine Learning or AKS-hosted deployments, the responsibility for implementing them shifts to the deploying organisation. This distinction should inform the choice of deployment option as much as raw price or nominal throughput figures.

5. Security Considerations

Security considerations for generative AI deployments on Azure can be grouped into three categories: infrastructure security, data protection, and model-specific security.

5.1 Infrastructure Security

Classical cloud security reference architectures emphasise identity and access management, network segmentation, and encryption of data at rest and in transit as foundational controls (Fernandez & Monge, 2014). On Azure, these controls map to Microsoft Entra ID for identity, Azure Private Link and virtual network integration to keep traffic to Azure OpenAI Service and Azure Machine Learning endpoints off the public internet, and Azure Key Vault for managing encryption keys and API secrets. Azure Defender for Cloud provides continuous posture assessment across these controls, consistent with the audit-oriented security patterns recommended for regulated cloud deployments (Barati et al., 2021).

5.2 Data Protection and Regulatory Compliance

Generative AI applications, particularly those using retrieval-augmented generation, often process proprietary or personal data drawn from enterprise data stores. This raises direct compliance obligations under data protection regulations such as the GDPR. Architectural patterns that attach usage policies directly to data as it moves through a cloud system—so-called sticky policies—offer a mechanism for making compliance verifiable at the data level rather than relying solely on perimeter controls (Cambronero et al., 2024). Continuous cloud auditing techniques extend this by providing an immutable record of how personal data is accessed and processed across a multi-tenant cloud environment (Barati et al., 2021), a capability that is increasingly available on Azure through the combination of Microsoft Purview and activity logging.

5.3 Model-Specific Security Risks

Beyond infrastructure and data protection, generative AI introduces security risks that do not have direct analogues in traditional cloud applications. These include prompt injection, in which crafted input causes a model to disregard its original instructions; jailbreaking, in which safety behaviours are circumvented through adversarial prompting; and training-data extraction, in which a model inadvertently reproduces sensitive information seen during training or fine-tuning (Liu & Hu, 2024). Azure AI Content Safety provides a managed mitigation for some of these risks by filtering both prompts and model outputs, but the underlying survey literature emphasises that no single control is sufficient and that defence should be layered across input validation, output filtering, and monitoring (Liu & Hu, 2024). Differentially private training techniques offer an additional layer of protection specifically against training-data extraction when organisations fine-tune models on sensitive datasets (Abadi et al., 2016), although such techniques introduce their own performance and accuracy costs that must be weighed against the sensitivity of the underlying data.

A practical implication of this layered threat model is that security controls are not cost-free from a performance perspective: content-safety filtering, private-network routing, and additional authentication steps each add measurable latency to the request path shown in Figure 1, reinforcing the need to treat performance and security as jointly optimised rather than sequentially addressed concerns.

6. Cost Considerations

Generative AI inference is materially more expensive per request than traditional web application workloads because it depends on specialised, GPU-backed compute whose supply is constrained relative to demand (Deochake, 2023). Table 2 summarises seven cost levers available to architects deploying generative AI applications on Azure, drawn from the cloud cost-optimisation and FinOps literature (Deochake, 2023; Khan et al., 2024; Qu et al., 2024; Stupar & Huljenic, 2023).

Cost Lever	Mechanism on Azure	Primary Trade-off
Consumption-based pricing	Pay-as-you-go billing per input/output token on Azure OpenAI Service	Low commitment but unpredictable spend under variable load
Provisioned Throughput Units (PTUs)	Pre-purchased, reserved model capacity for a fixed term	Predictable performance and cost at high volume, but reduced flexibility and risk of under-utilisation
Reserved Instances / Savings Plans	One- or three-year commitments on GPU/CPU virtual machines	Significant discounts in exchange for reduced elasticity
Autoscaling and scale-to-zero	Azure Container Apps, AKS Horizontal Pod Autoscaler, Azure ML endpoint autoscale rules	Reduces idle spend but introduces cold-start latency

Cost Lever	Mechanism on Azure	Primary Trade-off
Caching and prompt optimisation	Semantic caching, response reuse, and prompt-length reduction	Lowers token consumption but requires engineering investment and cache-invalidation logic
Model right-sizing	Choosing smaller or distilled models for narrow tasks instead of a single large model	Cuts inference cost but may reduce output quality for complex tasks
Spot virtual machines	Discounted, interruptible VM capacity for batch fine-tuning or evaluation jobs	Substantial savings offset by eviction risk, unsuitable for real-time inference

Table 2. Cost-optimisation levers for generative AI deployment on Azure.

Several patterns emerge from this comparison. First, the levers that most directly reduce unit cost—reserved capacity and spot virtual machines—do so by exchanging flexibility for a lower price, meaning they are appropriate only where workload demand is either highly predictable (favouring reservations) or fully tolerant of interruption (favouring spot capacity for batch fine-tuning or evaluation jobs). Second, application-level levers such as semantic caching, prompt-length reduction, and model right-sizing reduce cost without requiring a long-term commitment, but they require engineering investment and ongoing monitoring to remain effective as usage patterns evolve. Third, autoscaling and scale-to-zero configurations reduce idle spend but reintroduce the cold-start latency trade-off discussed in Section 4, illustrating again that cost decisions cannot be evaluated independently of their performance consequences.

Formal cost-optimisation research supports treating these levers as a joint optimisation problem rather than applying them independently: graph-based cost models can identify the combination of deployment configuration and resource type that minimises operational cost subject to service-level constraints (Khan et al., 2024), while stochastic models of cloud resource selection under uncertain demand provide a principled basis for choosing between reserved and on-demand capacity (Qu et al., 2024). Embedding this analysis into a FinOps practice—supported on Azure by Azure Cost Management and budget alerts—allows organisations to revisit these trade-offs continuously as usage patterns and Azure pricing evolve, rather than fixing them at initial deployment.

7. Discussion: Balancing Performance, Security, and Cost

The preceding sections show that performance, security, and cost decisions in Azure-based generative AI deployments are coupled rather than independent. Provisioned Throughput Units improve latency predictability but increase fixed cost regardless of actual utilisation. Private networking and content-safety filtering improve the security posture but add latency to every request. Autoscaling and scale-to-zero configurations reduce cost but reintroduce the latency variability that provisioned capacity is designed to eliminate. These interactions mean that an architectural decision justified purely on performance or purely on cost grounds is likely to be incomplete.

A practical implication is that generative AI deployment on Azure benefits from being treated as a continuous, cross-functional governance process—drawing on MLOps practices for model and pipeline management (Kreuzberger et al., 2023), FinOps practices for cost accountability (Deochake, 2023), and security operations for ongoing threat monitoring—rather than a one-time architectural decision made at project inception. Organisations with predictable, high-volume traffic and strict latency requirements are likely to favour Provisioned Throughput Units combined with private networking, accepting higher fixed cost in exchange for predictability and control. Organisations with variable or exploratory workloads may instead favour pay-as-you-go pricing combined with aggressive caching and autoscaling, accepting greater latency variability in exchange for cost efficiency. Neither configuration is universally superior; the appropriate balance depends on the specific latency, compliance, and budget constraints of the application in question.

8. Limitations and Future Research Directions

This paper is a literature-based synthesis rather than an empirical study, and several limitations follow from that scope. First, because peer-reviewed, Azure-specific empirical studies of generative AI deployment remain limited, much of the mapping between general cloud-computing and LLM-serving findings and specific Azure services is necessarily interpretive rather than directly validated. Second, Azure's generative AI service offerings and pricing are evolving rapidly, and quantitative comparisons

(for example, exact latency or cost figures) would require empirical benchmarking that falls outside the scope of this review. Third, the analysis focuses primarily on inference-time deployment and does not address the performance, security, and cost considerations specific to model fine-tuning and training pipelines in depth.

Future research would benefit from controlled empirical benchmarking of Azure OpenAI Service, Azure Machine Learning, and Azure Kubernetes Service deployment options under realistic production traffic patterns, reporting latency, throughput, and cost together rather than in isolation. Further work is also needed on quantifying the latency and accuracy cost of layered security controls, such as content filtering and differentially private fine-tuning, for specific classes of generative AI application, and on developing decision frameworks that allow architects to select among the deployment options summarised in Tables 1 and 2 based on explicit service-level and budget constraints rather than general heuristics.

9. Conclusion

Deploying generative AI applications on Microsoft Azure requires architects to reconcile performance, security, and cost objectives that are frequently in tension with one another. Drawing on the cloud computing, LLM-serving, cloud security, and cloud cost-management literature, this paper has proposed a reference architecture for generative AI deployment on Azure and examined how common deployment options and cost-optimisation levers affect each of these three dimensions. The evidence reviewed indicates that no single deployment configuration is optimal across all workloads: the appropriate combination of model-serving option, security control, and cost lever depends on the latency sensitivity, regulatory context, and traffic predictability of the specific application. Treating performance, security, and cost as a single, continuously governed decision space—rather than three separate concerns addressed by different teams—is likely to be a defining characteristic of mature, production-grade generative AI deployments on Azure.

Funding: This research received no external funding

Conflicts of Interest: The authors declare no conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers

Artificial Intelligence (AI) Use Disclosure: The authors declare that no artificial intelligence tools were used in the preparation of this manuscript

References

[1] Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L. (2016). Deep learning with differential privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (pp. 308–318). Association for Computing Machinery. https://doi.org/10.1145/2976749.2978318

[2] Barati, M., Aujla, G. S., Llanos, J. T., Duodu, K. A., Rana, O., Carr, M., & Ranjan, R. (2021). Privacy-aware cloud auditing for GDPR compliance verification in online healthcare. IEEE Transactions on Industrial Informatics, 18(7), 4808–4819. https://doi.org/10.1109/TII.2021.3100152

[3] Bernstein, D. (2014). Containers and cloud: From LXC to Docker to Kubernetes. IEEE Cloud Computing, 1(3), 81–84. https://doi.org/10.1109/MCC.2014.51

[4] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Amodei, D. (2020). Language models are few-shot learners (arXiv:2005.14165). arXiv. https://doi.org/10.48550/arXiv.2005.14165

[5] Cambronero, M. E., Martínez, M. A., Llana, L., Rodríguez, R. J., & Russo, A. (2024). Towards a GDPR-compliant cloud architecture with data privacy controlled through sticky policies. PeerJ Computer Science, 10, Article e1898. https://doi.org/10.7717/peerj-cs.1898

[6] Carrión, C. (2022). Kubernetes as a standard container orchestrator: A bibliometric analysis. Journal of Grid Computing, 20(4), Article 42. https://doi.org/10.1007/s10723-022-09629-8

[7] Deochake, S. (2023). Cloud and AI infrastructure cost optimization: A comprehensive review of strategies and case studies (arXiv:2307.12479). arXiv. https://doi.org/10.48550/arXiv.2307.12479

[8] Fernandez, E. B., & Monge, R. (2014). A security reference architecture for cloud systems. In Proceedings of the 2014 Conference on Research in Adaptive and Convergent Systems (RACS '14) (pp. 1–5). Association for Computing Machinery. https://doi.org/10.1145/2578128.2578229

[9] Hewage, N., & Meedeniya, D. (2022). Machine learning operations: A survey on MLOps tool support (arXiv:2202.10169). arXiv. https://doi.org/10.48550/arXiv.2202.10169

[10] Khan, A. Q., Matskin, M., Prodan, R., Bussler, C., Roman, D., & Soylu, A. (2024). Cost modelling and optimisation for cloud: A graph-based approach. Journal of Cloud Computing, 13, Article 118. https://doi.org/10.1186/s13677-024-00709-6

[11] Kreuzberger, D., Kühl, N., & Hirschl, S. (2023). Machine learning operations (MLOps): Overview, definition, and architecture. IEEE Access, 11, 31866–31879. https://doi.org/10.1109/ACCESS.2023.3262138

[12] Li, B., Guo, L., Cheng, J., Chen, Q., He, B., & Guo, M. (2022). The serverless computing survey: A technical primer for design architecture. ACM Computing Surveys, 54(10s), 1–34. https://doi.org/10.1145/3508360

[13] Li, B., Jiang, Y., Gadepally, V., & Tiwari, D. (2024). LLM inference serving: Survey of recent advances and opportunities (arXiv:2407.12391). arXiv. https://doi.org/10.48550/arXiv.2407.12391

[14] Mell, P., & Grance, T. (2011). The NIST definition of cloud computing (NIST Special Publication 800-145). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-145

[15] Miao, X., Oliaro, G., Zhang, Z., Cheng, X., Jin, H., Chen, T., & Jia, Z. (2023). Towards efficient generative large language model serving: A survey from algorithms to systems (arXiv:2312.15234). arXiv. https://doi.org/10.48550/arXiv.2312.15234

[16] Patel, D., Raut, G., Cheetirala, S. N., Nadkarni, G. N., Freeman, R., Glicksberg, B. S., Klang, E., & Timsina, P. (2024). Cloud platforms for developing generative AI solutions: A scoping review of tools and services (arXiv:2412.06044). arXiv. https://doi.org/10.48550/arXiv.2412.06044

[17] Qu, Z., Dawande, M., & Janakiraman, G. (2024). Cloud cost optimization: Model, bounds, and asymptotics. Operations Research. Advance online publication. https://doi.org/10.1287/opre.2022.0362

[18] Stupar, I., & Huljenic, D. (2023). Model-based cloud service deployment optimisation method for minimisation of application service operational cost. Journal of Cloud Computing, 12, Article 24. https://doi.org/10.1186/s13677-023-00389-8

[19] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need (arXiv:1706.03762). arXiv. https://doi.org/10.48550/arXiv.1706.03762