International Journal of Computer (IJC)

ISSN 2307-4523 (Print & Online)

https://ijcjournal.org/index.php/InternationalJournalOfComputer/index

Scaling MLOps in Pharma: Automating Model Deployment and Monitoring with DataRobot on AWS

Nivedha Sampath*

Platform Engineer at Takeda Pharmaceuticals,Boston, USA

Email:nivedha.sampath@takeda.com

Abstract

This article discusses the MLOps scaling challenge amidst an increase in medical data and growing regulatory demands, which pharmaceutical companies are facing. On the AWS infrastructure, implementing DataRobot will make it possible to automate model deployment as well as monitoring and maintenance of models — without manual oversight; this traditional bottleneck usually creates another risk avenue for non-compliance with good manufacturing practice requirements. Such solution relevance is driven by an acute need for algorithm reproducibility and traceability under circumstances where pharmaceutical companies concurrently operate dozens of models. A lapse in validation or documentation can lead to clinical program delays and increased costs. The novelty here is a unified automated model registry plus deployment pipelines integrated with built-in drift/accuracy tracking and regulatorily significant audit systems. The major findings are that by automating the lifecycle, artificial intelligence stops being an artisanal collection of disconnected experiments and becomes a manageable production process. It shows how running DataRobot on AWS not only speeds up getting algorithms to the clinic but also makes sure strong FDA and GxP rules are followed with built-in version control, legally applicable report making, and data encryption. Such a strong setup where scientific change and rule order do not fight but instead help each other. The article will be of great use to drug makers, data engineers, MLOps workers, and rule managers who want to see the real use of auto ways for model work.

Keywords: MLOps; pharmaceuticals; DataRobot; AWS; automation; model monitoring; regulatory requirements.

Received: 10/16/2025
Accepted: 12/16/2025
Published: 12/26/2025

----

* *Corresponding author.*

1. Introduction

Over the past five years, the pharmaceutical industry has been plunging into an ocean of heterogeneous data—from RNA-sequencing outputs and real-world clinical records to streaming signals from wearables. Analysts estimate that the cumulative volume of medical and pharmaceutical information is growing by more than 35% annually and by 2025 will approach ten thousand exabytes, effectively tripling relative to 2020 levels [1]. This avalanche of information elevates machine learning from an experimental instrument to a central mechanism for accelerating preclinical research, optimizing clinical programs, and strengthening post-marketing surveillance.

Yet as companies deploy dozens—and in some cases hundreds—of models, the flip side of progress becomes apparent. A recent survey of 100 leaders in the life sciences sector said that while 75% of organizations have implemented artificial intelligence within the last two years, only half have instituted any formal process for ongoing algorithm stewardship [2]. This means that with every new model, manual version control, metric reconciliation, infrastructure reconfiguration, and inspection-grade documentation must be worked out for months to come, long before it adds regulatory risk. Thus does the DataRobot-AWS combo step into the foreground with a unified tech fabric wherein automated registration, deployment, and continuous oversight of models are delivered as platform standards. The AWS Marketplace highlights that DataRobot Enterprise AI Suite covers the full lifecycle from data prep and training up to real-time monitoring and version management that can be deployed in a public cloud or an isolated segment to fulfill GxP requirements [3]. Added to this are built-in mechanisms for tracking feature drift, accuracy, and resilience available through the MLOps module, which automatically flags degradation or deviations from permitted thresholds [4]. Thus, the DataRobot ecosystem on AWS turns the landscape of disparate models into a governed data-science conveyor, freeing specialists from routine toil and allowing them to focus on scientific novelty and clinical value.

2. Materials and Methodology

This paper is built around a general review of academic, industry, and regulatory sources that describe changes in the pharmaceutical sector under twin forces: the exponential growth of data and platform solutions for operating machine-learning models. The theoretical base draws on works about the dynamics of pharmaceutical data and their value for research and executive surveys, revealing a gap between the scale of algorithm adoption and the maturity of their operational stewardship. Major technological components include the DataRobot Enterprise AI Suite available through AWS Marketplace, and its included MLOps module, which offers versioning, accuracy monitoring, and automated response to feature drift [1-4].

Methodologically, this work conducts a systematic review of regulations and design guidance from authorities, drawing on the FDA’s recommendations on the algorithm lifecycle and requirements under 21 CFR Part 11 concerning traceability of electronic records [15, 18]. It uses these documents to map both DataRobot and AWS capabilities in the context of pharmaceutical practice to assess their applicability under GxP. In parallel with the legal analysis, a comparative examination of AWS architectural options has been carried out: scalability by Lambda and EKS clusters [7, 8], data management by S3 and Glue Data Catalog [10, 11], and private access plus infrastructure protection mechanisms by VPC Endpoints and AWS Config [13, 14].

The methods include content analysis of DataRobot and AWS technical documentation, a juxtaposition of DevOps and MLOps automation mechanisms drawing on examples with GitHub Actions and AWS CodePipeline [5, 6], and an analysis of practical model-deployment scenarios using SageMaker Endpoints and blue/green and canary strategies [9, 17, 19, 20]. In addition, market data on the growth of AI in healthcare [21] are used to interpret these processes not only through technological feasibility but also through the lens of economic pressures on pharmaceutical organizations.

3. Results and Discussion

The regulatory environment lags the exponential rise of algorithms; hence, new requirements are coalescing around pharmaceutical model developers to convert one-off experiments into a predictable, documented, and reproducible process. In January 2025, the U.S. Food and Drug Administration released a draft guidance on the lifecycle of AI-enabled device software functions, which for the first time proposes that each model include a change dossier delineating the boundaries of permissible self-learning and the criteria for re-verification before an updated version is placed on the market [15]. In the same paradigm, 21 CFR Part 11 obliges pharma companies to ensure the traceability of electronic records and the legal force of electronic signatures; the document explicitly states that every step in processing data used for regulatory purposes must be validated and recoverable even years later.

Taken together, these provisions define a new reality: every pharmaceutical model must operate in a context where version control, audit logging, drift assessment, and rollback are technically embedded, and where inspector-ready documents are generated with keystrokes rather than handcrafted reports. In other words, the norms effectively nudge companies to adopt platform solutions such as DataRobot on AWS, not as an option but as a necessary infrastructural layer that simultaneously accelerates scientific cycles and proves to regulators that algorithms remain reliable, transparent, and governable at each turn of their evolution.

The skeleton of the target architecture that binds scientific ambition to regulatory discipline begins at continuous integration: the moment a researcher commits code changes, GitHub Actions automatically forms a model artifact and, without leaving the protected repository, hands it to AWS CodePipeline. The size of the community working behind this chain is stunning. More than 150 million developers already publish and maintain projects, according to the Octoverse report, and initiatives related to generative algorithms increased by 98% year-over-year. In the production segment of the conveyor, CodePipeline guarantees 99.9% availability every month, which meets the strict internal SLAs of pharma companies and enables models to be promoted continuously from test to production environments [6].

The next layer is where the compute placement management happens. For workloads that require bursting—such as high-speed signal detection in pharmacovigilance—workload runs on AWS Lambda, which by default provides up to 1000 concurrent executions per region, scaling within fractions of a second without any administrator intervention needed [7]. When there is a prolonged resource-hungry three-dimensional protein structuring calculation needed, an Amazon EKS Cluster with support automatically adds or removes nodes based on GPU instances as indicated by the presence of unscheduled Pods [8]. For latency-sensitive scenarios characteristic of personalized dosing, the model is deployed as a SageMaker Real-Time Endpoint, where the service wrapper automatically tunes throughput to the actual request stream [9].

Above the compute resources lies the DataRobot MLOps control plane. Whether an organization chooses a SaaS option or a hardened self-managed cluster in a private VPC, the platform records every model version, monitors feature drift, and when needed, launches challenger algorithms without interrupting primary inference [4]. In regulator alignment, a critical point is that model state reports are generated from the same event log and can be affixed with an electronic signature, satisfying the Part 11 requirements cited earlier.

The data foundation of this ecosystem stands on three pillars. Amazon S3—where, as of January 2023, server-side encryption is enabled by default for all new objects—provides inexpensive yet protected storage for laboratory matrices and clinical observations [10]. Metadata about datasets, their schemas, and lineage is accumulated in the AWS Glue Data Catalog; catalog crawlers automatically scan sources both inside and outside the cloud, creating a unified map of organizational data and allowing DataRobot to reference input-sample versions down to the partition [11]. For analytics workloads—from simulating population pharmacokinetics to assessing cold-chain inventories—Amazon Redshift is used, where encryption at the cluster and snapshot levels is activated with a single setting and governed via AWS KMS [12].

The defensive perimeter completes the picture. All services are invoked through private VPC endpoints based on AWS PrivateLink, ensuring production data never leaves the company's address space [13]. The audit log is formed continuously: an AWS Config conformance pack compares every infrastructure change against the checkpoints of 21 CFR Part 11, enabling real-time detection of deviations from regulated procedures and automatically initiating remediation [14]. In sum, these layers build not just a technical chart but a breathing ecology where model growth, rollout, and use join into one ongoing loop while staying clear to reviewers and strong against work hitches. The next right move, after the support frame is set up, is to turn the flow of trials into a strict, nearly clockwork custom where each code version takes an equally marked route. The starting point is the DataRobot Model Registry, where a container-folder is created for each scientific concept; within it, the system automatically numbers releases, records artifact hashes, and binds them to specific datasets—eliminating the traditional confusion between latest_final_v2 and the actual production build [16]. Because the registry page itself preserves a full chronology of changes, an auditor can retrospectively reconstruct which training parameters and sample were used a year—or even a decade—ago, which is especially important for subsequent registration dossiers.

As soon as a new release lands on the registry shelf, the build–test–deploy conveyor kicks in: GitHub Actions, upon receiving the merge event, compiles auxiliary dependencies, runs unit tests, and passes the baton to AWS CodePipeline. The latter not only rebuilds the image but also pushes it into a private registry, guaranteeing 99.9% availability under the service-level agreement specifically documented with Amazon [6]. Thus, even rare outages of the centralized build line fit within the allowable-faults regimen for clinical and manufacturing information systems.

At the packaging stage, DataRobot provides a self-contained execution image—already including compiled dependent libraries, the prediction code proper, and an auxiliary agent for subsequent monitoring. The image is exported to Amazon Elastic Container Registry or directly to a managed SageMaker environment, where it can run as a continuous service or in a serverless mode if the request flow is bursty [17]. Described in Fig. 1 is MLOps Agent Monitoring. This unbinds the crew from hand chasing after fit framework forms and allows for the same load to be set in both cloud and on-prem parts without needing to rework.

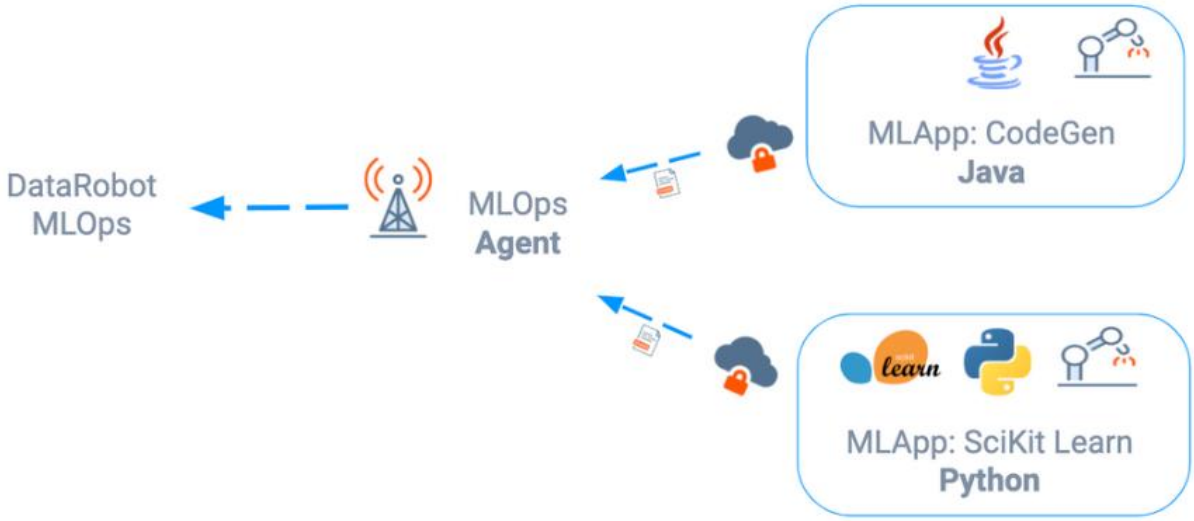

Figure 1: MLOps Agent Monitoring [17]

Once integration testing has been completed successfully, the conveyor will automatically promote deployment from staging into production, and every action taken will be wrapped inside an electronic signature that is 21 CFR Part 11 recognized: storing the signature along with test results. It serves as legal confirmation that the algorithm has passed all agreed checks [18]. If metrics fall outside pre-agreed bounds, the version is automatically sent back for correction, and the registry is marked with a declined release—creating an unbroken, tamper-evident chain of trust.

To minimize technology risk when rolling out an update, the platform applies a blue/green scheme: old and new versions continue to operate in parallel, with traffic shifting either gradually or all at once depending on process criticality [19]. Where a case is extra-sensitive, a canary policy is implemented, in which only a minor percentage of requests are first sent to the new model, depicted in Figure 2. The system contrasts precision, delay, and resource-use measures on-the-fly and just after getting a positive result increases the load share—while maintaining the capacity to roll back immediately with no data loss [20].

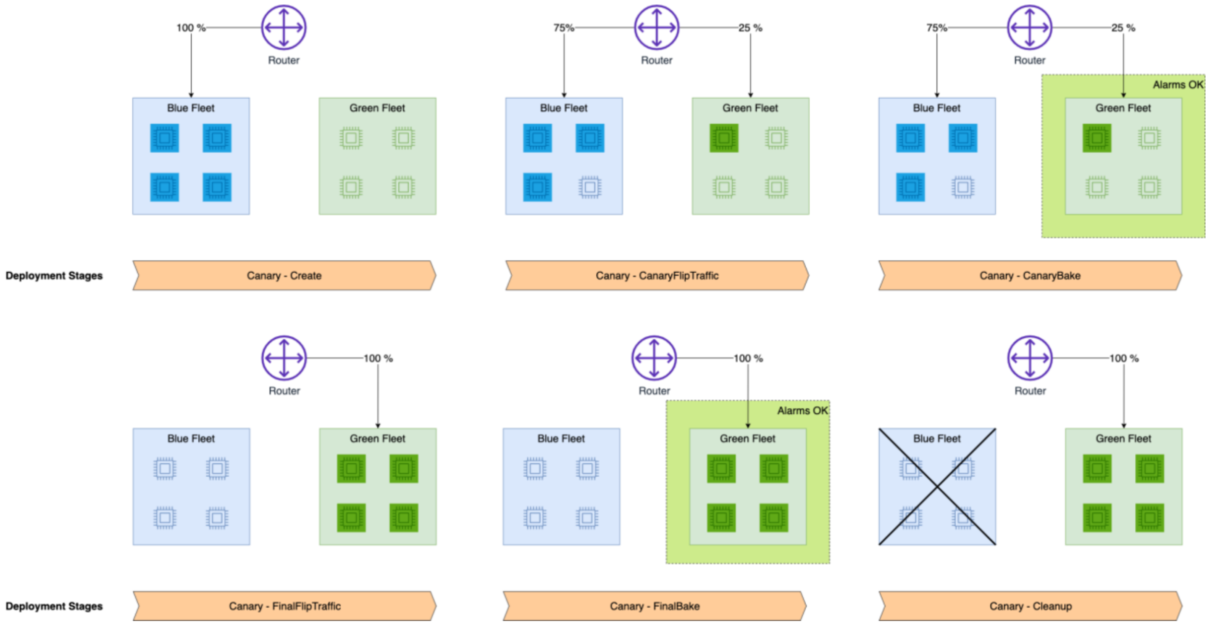

Figure 2: A two-step canary traffic shift from the old fleet to the new fleet [20]

As a result, the registry → pipeline → signature → two- or three-stage placement chain becomes a self-tuning regulatory corridor that accelerates updates without relaxing the control demanded by the pharmaceutical domain. At the same time, economic pressure is palpable. The global AI in healthcare market size was estimated at USD 26.57 billion in 2024 and is projected to reach USD 187.69 billion by 2030, growing at a CAGR of 38.62% [21].

Once a model intercepts its first live data stream, a quiet yet unceasing chronicle of its behavior begins: for every input feature, the system preserves distributions, comparing them against those seen during training; at the slightest divergence—drift—it raises an alert via the built-in notification center. Record in parallel any swing of the reference labels, since it is not always the patient profile that changes, which causes the labels to swing. In these cases, answer back by reviewing the clinical paradigm, not by retraining. A graphical console brings together all these statistical tests into one unified heatmap sketch so that a validation engineer can see at a glance where and how much fidelity is leaking.

A champion-challenger duel feeds these visualizations. The incumbent continues to serve production requests, while one more alternative invites a tournament running in parallel and being scored with the use of delayed labels on the same inputs it receives. If the challenger maintains better accuracy over a period defined as control time, it will automatically earn its place to take over from the incumbent; demotion is just as clinical, thereby avoiding any emotive discussion within the team and lowering subjective decision-making regulatory risk.

Cost accounting is another sentinel standing watch above computing steadiness. EKS clusters grow and shrink, counting every core; serverless Lambda functions instantly scale out concurrency under request surges and go away when there's no traffic too. Standing up monitoring correlates call frequency, instance duration, and actual GPU utilization, showing on the dashboard not just proof of overspend but a forecast of when the current configuration is no longer economically justified.

All events—from drift to budget oscillations—are journaled in an end-to-end audit stream. With a single click, a chronological report is produced in a format suitable for an inspector; control signatures, version hash sums, and links to data sources are inserted automatically so any auditor can reconstruct the path of each predicted quantity from sensor to final metric.

The protected contour encloses these processes: computation happens within a separated virtual network, and access to object storage is organized by private endpoints with no exposure, naturally, to the public internet. All data will be encrypted with symmetric keys generated and rotated by the key-management service; unauthorized copying is forbidden by policy, ensuring that no record will leave the trust zone unencrypted.

A role model divides user rights by the principle of least privilege: the researcher sees only those datasets with which she works, the operations lead gets metrics but not confidential patient fields, and login is attached to a single sign-on service, making revocation simple when teams change. Every administrative action is logged just as carefully as new model rollouts so that the chain of trust is never interrupted.

A traceability matrix with horizontal rows of regulatory requirements and vertical columns of specific lifecycle artifacts. The table fills as soon as the model has passed each applicable control gate, thereby eliminating the onerous set of screenshots and oral attestations, replacing it with a digital passport ready for presentation without pre-assembled slide decks. The overall architecture is shown in Figure 3.

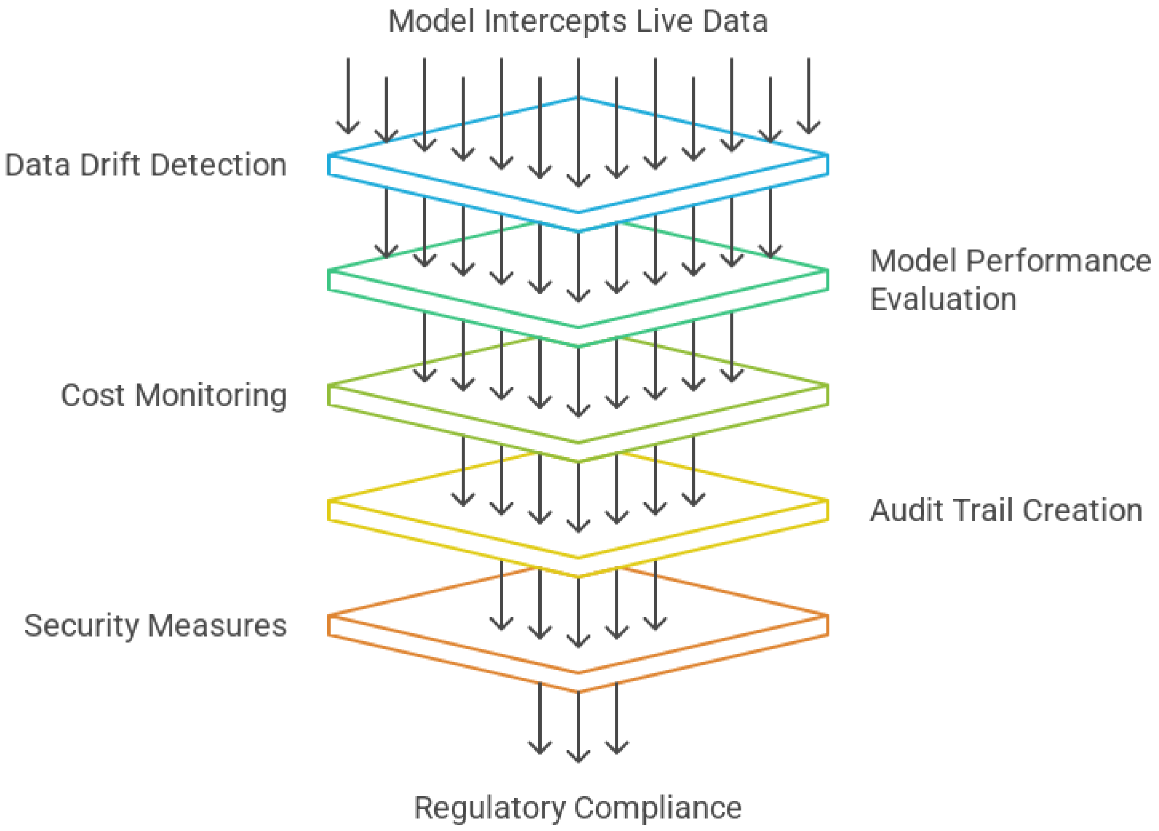

Figure 3: Model Validation and Security Process

Thus, surveillance, security, and documentation become one bloodstream enabling algorithms to leap forward fast but still within control—and enabling organizations to stay sure that every byte is computed, stored, and verified fully in line with good manufacturing practice.

4.Conclusion

Thus, under conditions of surging pharmaceutical data volumes and rising regulatory demands, platform solutions based on DataRobot and AWS become not merely instruments of optimization but, in effect, a necessary infrastructural standard. This combination guarantees continuity and openness throughout the model lifecycle, such that all steps— from data preparation to production deployment— are captured in a repeatable, legally meaningful, and regulatorily verifiable way. It demonstrates that by making the model operations process automated, it eliminates manual documentation—a key bottleneck—unmanaged drift risk, as well as subjective managerial decisions. There shall be accuracy monitoring, distribution comparison of features, cost tracking, as well as auditing infrastructure changes, which will enable a closed loop of trust for organizations to accelerate scientific progress without weakening oversight. Equally critical is security: network segmentation, data encryption, access control, and a defensive barrier aligned with stringent GxP and Part 11 requirements.

So the suggested setup changes machine learning in drugs from being just separate tests to a controlled and manageable process. This also cuts down time-to-clinic for steps, lowers work dangers, and shows rule keepers the rightness of every guess. In the long run, such integrated MLOps approaches are pivotal to enabling the industry to cope with the avalanche of data while preserving scientific soundness, economic efficiency, and social responsibility.

References

[1] "Tapping Into New Potential: Realising the Value of Data in the Healthcare Sector," *L.E.K. Consulting*, Dec. 04, 2023. https://www.lek.com/insights/hea/eu/ei/tapping-new-potential-realising-value-data-healthcare-sector (accessed Aug. 01, 2025).

[2] M. Goldman, "Life science firms move ahead on AI, with concerns," *Axios*, Nov. 14, 2024. https://www.axios.com/2024/11/14/life-sciences-ai-concerns (accessed Aug. 01, 2025).

[3] AWS, "DataRobot Enterprise AI Suite for AWS," *AWS*. https://aws.amazon.com/marketplace/pp/prodview-fuf2kssofoydm (accessed Aug. 02, 2025).

[4] Datarobot, "MLOps: DataRobot docs," *Datarobot*. https://docs.datarobot.com/en/docs/mlops/index.html (accessed Aug. 03, 2025).

[5] "Engineering leadership in the age of AI: Insights from GitHub." Accessed: Aug. 04, 2025. [Online]. Available: https://assets.ctfassets.net/wfutmusr1t3h/71Tv1g9g7em6GNdeZZIOQO/5083290f232ba3f2d68864b92f8654ae/GitHub-Engineering-Leadership.pdf

[6] AWS, "AWS CodePipeline Service Level Agreement," *AWS*, 2025. https://aws.amazon.com/ru/codepipeline/sla/ (accessed Aug. 05, 2025).

[7] AWS, "Lambda function scaling," *AWS*. https://docs.aws.amazon.com/lambda/latest/dg/lambda-concurrency.html (accessed Aug. 06, 2025).

[8] AWS, "Scale cluster compute with Karpenter and Cluster Autoscaler," *AWS*. https://docs.aws.amazon.com/eks/latest/userguide/autoscaling.html (accessed Aug. 07, 2025).

[9] AWS, "Real-time inference," *AWS*. https://docs.aws.amazon.com/sagemaker/latest/dg/realtime-endpoints.html (accessed Aug. 08, 2025).

[10] AWS, "Protecting data using encryption," *AWS*, 2023. https://docs.aws.amazon.com/AmazonS3/latest/userguide/UsingEncryption.html (accessed Aug. 09, 2025).

[11] AWS, "Data Catalog and crawlers in AWS Glue," *AWS*. https://docs.aws.amazon.com/glue/latest/dg/catalog-and-crawler.html (accessed Aug. 10, 2025).

[12] AWS, "Encryption at rest," *AWS*. https://docs.aws.amazon.com/redshift/latest/mgmt/security-server-side-encryption.html (accessed Aug. 11, 2025).

[13] AWS, "Access an AWS service using an interface VPC endpoint," *AWS*. https://docs.aws.amazon.com/vpc/latest/privatelink/create-interface-endpoint.html (accessed Aug. 12, 2025).

[14] AWS, "Operational Best Practices for FDA Title 21 CFR Part 11," *AWS*. https://docs.aws.amazon.com/config/latest/developerguide/operational-best-practices-for-FDA-21CFR-Part-11.html (accessed Aug. 13, 2025).

[15] "AI-Enabled Device Software Functions," *U.S. Food and Drug Administration*. https://www.fda.gov/regulatory-information/search-fda-guidance-documents/artificial-intelligence-enabled-device-software-functions-lifecycle-management-and-marketing (accessed Aug. 04, 2025).

[16] "Model Registry: DataRobot docs," *Datarobot*. https://docs.datarobot.com/en/docs/mlops/deployment/registry/reg-create.html (accessed Aug. 05, 2025).

[17] "Deploy models on SageMaker," *Datarobot*, 2025. https://docs.datarobot.com/en/docs/integrations/aws/sagemaker/sagemaker-deploy.html (accessed Aug. 06, 2025).

[18] FDA, "Part 11 Electronic Records Electronic Signatures Scope and Application," *U.S. Food and Drug*

Administration. https://www.fda.gov/regulatory-information/search-fda-guidance-documents/part-11-electronic-records-electronic-signatures-scope-and-application (accessed Aug. 07, 2025).

[19] AWS, "Blue/Green Deployments," *AWS*. https://docs.aws.amazon.com/sagemaker/latest/dg/deployment-guardrails-blue-green.html (accessed Aug. 08, 2025).

[20] AWS, "Use canary traffic shifting," *AWS*. https://docs.aws.amazon.com/sagemaker/latest/dg/deployment-guardrails-blue-green-canary.html (accessed Aug. 09, 2025).

[21] Grand View Research, "Artificial Intelligence In Healthcare Market Size Report, 2019-2025," *Grand View Research*. https://www.grandviewresearch.com/industry-analysis/artificial-intelligence-ai-healthcare-market (accessed Aug. 20, 2025).