



Author: Adam Graves, a seasoned CITO professional with over 20 years of experience in IT services, DevOps architecture, and a Master's degree in Applied AI at USD.

From DevOps to MLOps: A Comprehensive Guide to Migrating and Adding Services

The evolution of [DevOps](#) to include machine learning operations ([MLOps](#)) represents a significant shift in how organizations approach the development, deployment, and maintenance of machine learning models. Integrating MLOps into an existing DevOps framework involves several crucial steps to ensure seamless integration and operational efficiency. This article outlines these steps, providing a roadmap for organizations looking to embark on this journey.

Understanding DevOps and MLOps

Before diving into the migration process, it's essential to understand the fundamental differences between DevOps and MLOps.

DevOps focuses on automating and streamlining the software development lifecycle ([SDLC](#)) to improve the speed, quality, and reliability of software delivery. For those that were managing application development back in the early 2000's it had become apparent that many development failures were due to the lack of knowledge and integration of the actual operational environment in the development cycle. As more and more development communities began to merge the operations into their development cycle the official word of DevOps had surfaced, with new conferences like Devopsdays (<https://devopsdays.org>) bringing these new communities together.

Today we find different flavors to DevOps meaning within businesses, some have an advanced infrastructure that includes Continuous Integration and Continuous Deployment ([CI/CD](#)), and some have the entire structure under third party vendors.

Over the years DevOps has integrated the Best Practices methodologies along with Ethical Coding methodologies.

MLOps, on the other hand, extends DevOps principles to machine learning and data science workflows. It encompasses the practices, tools, and cultural changes required to manage the end-to-end lifecycle of machine learning models, from data preparation and model training to deployment and monitoring.

In most cases, the main change with MLOps is the increase in the volume and complexity of data that the application must handle. A dedicated team is required for analyzing and managing this data, which is often referred to as a DataOps team.

The Eckerson Group defines DataOps as: “DataOps is a data engineering approach that is designed for rapid, reliable, and repeatable delivery of production-ready data for analytics and data science. Beyond speed and reliability, DataOps enhances and advances data governance through engineering disciplines that support versioning of data, data transformations, and data lineage. DataOps supports operational agility for business operations, with the ability to meet new and changing data needs quickly. It also supports portability and technical operations agility with the ability to rapidly redeploy data pipelines across multiple platforms in on-premises, cloud, multi-cloud, and hybrid data ecosystems.” (D.Wells, 2019)

In MLOps, there is a need for AI model professionals who can develop and manage machine learning (ML) models and determine the best fit for development. The MLOps structure can be viewed as comprising three key components: 1. DevOps, 2. ModelOps, and 3. DataOps.

Steps to Migrate or Add Services from DevOps to MLOps

1. Assess Current DevOps Maturity

The first step is to evaluate your current DevOps maturity level. This involves assessing your existing CI/CD pipelines, infrastructure management practices, and the overall DevOps culture within your organization. Understanding your starting point helps identify the gaps that need to be addressed to integrate MLOps successfully.

2. Define MLOps Objectives and Requirements

Clearly define what you aim to achieve with MLOps. Have a clear understanding of the data, its sources, the potential bias, and ethics, and regulations if applicable.

Set specific goals such as improving model deployment frequency, reducing the time from model training to production, and ensuring robust monitoring and management of models. Outline the requirements, including data management needs, computational resources, and security considerations.

3. Build a Cross-Functional MLOps Team

Transitioning to MLOps requires collaboration across multiple disciplines. Form a cross-functional team comprising data scientists, machine learning engineers, software developers, and operations professionals. This team will be responsible for designing, implementing, and maintaining MLOps processes and tools.

4. Implement Version Control for Code and Data

Version control is a cornerstone of both DevOps and MLOps. Use version control systems (e.g., Git) not only for code but also for data, models, and configuration files. Tools like [DVC](#) (Data Version Control) can help manage datasets and model versions, ensuring reproducibility and traceability.

5. Develop Continuous Integration and Continuous Deployment (CI/CD) Pipelines for ML

Adapt your existing CI/CD pipelines to accommodate machine learning workflows. This involves setting up automated testing for data and models, integrating model training processes, and

deploying models to production environments. Tools like Jenkins, GitLab CI, and CircleCI can be extended with ML-specific plugins and scripts.

6. Automate Data Preparation and Feature Engineering

Data preparation and feature engineering are critical steps in the ML lifecycle. Automate these processes using pipelines that can handle data extraction, transformation, and loading ([ETL](#)). Platforms like Apache Airflow, Kubeflow Pipelines, and various Cloud Services Step Functions can orchestrate these workflows, ensuring data consistency and reducing manual effort.

7. Establish Model Training and Experimentation Frameworks

Implement frameworks for model training and experimentation to streamline the development process. Tools like [MLflow](#), [TensorBoard](#), and Weights & Biases facilitate experiment tracking, hyperparameter tuning, and model versioning. These frameworks enable data scientists to iterate quickly and manage their experiments effectively.

8. Integrate Model Validation and Testing

Ensure rigorous validation and testing of machine learning models before deployment. This includes unit tests for individual components, integration tests for end-to-end workflows, and performance tests to assess model accuracy and robustness. Implement automated testing in your CI/CD pipelines to catch issues early and maintain model quality.

9. Deploy Models with Continuous Deployment Practices

Deploying machine learning models requires careful consideration of serving infrastructure, scalability, and latency requirements. Use platforms like [Kubernetes](#), [Docker](#), and cloud-based services (e.g., AWS SageMaker, Google AI Platform, Azure AKS) to deploy models as scalable microservices. Implement continuous deployment practices to roll out updates seamlessly.

10. Implement Monitoring and Management for ML Models

As part of the ModelOps services, monitoring ML models in production is crucial to ensure their performance remains consistent over time. Set up monitoring systems to track model accuracy, drift, and resource usage. Tools like [Prometheus](#), [Grafana](#), and specialized ML monitoring platforms (e.g., Fiddler, Seldon) can provide insights and alerts to manage model performance proactively.

11. Ensure Security and Compliance

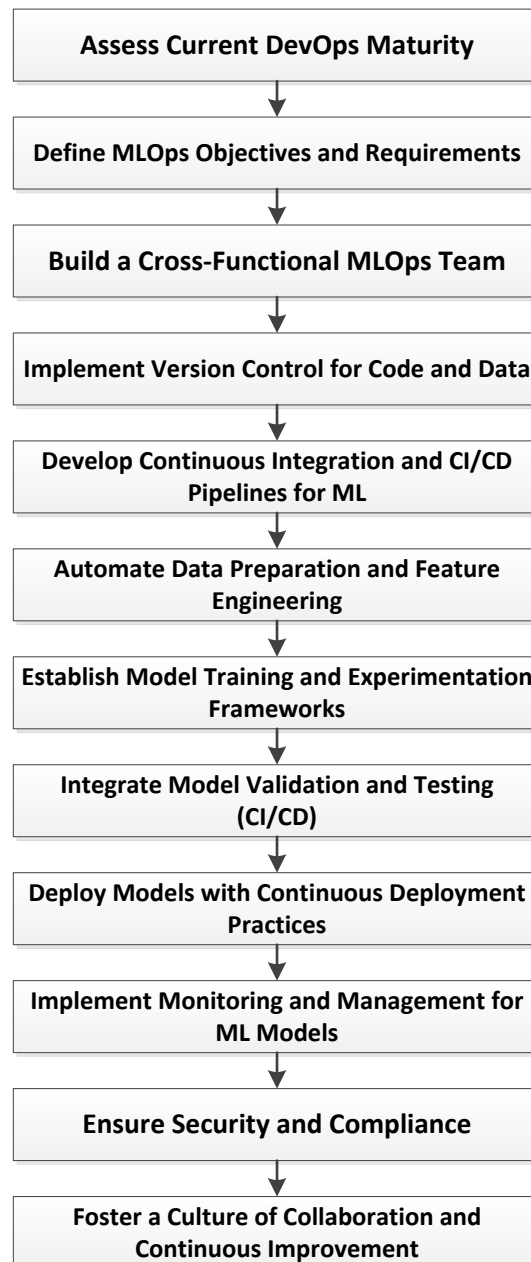
As part of an Access Management Services, security and compliance are vital considerations when dealing with sensitive data and models. Implement security best practices such as data encryption, access controls, and regular audits. If applicable ensure compliance with relevant regulations (e.g., [GDPR](#), [HIPAA](#)) to protect user data and maintain trust.

While these considerations are also crucial for DevOps, the increased data within MLOps makes this a more challenging task, necessitating properly allocated services and resources.

12. Foster a Culture of Collaboration and Continuous Improvement

Transitioning to MLOps is not just about technology; it's also about culture. Promote a culture of collaboration between data scientists, engineers, and operations teams. Encourage continuous learning and improvement through regular training, knowledge sharing, and retrospectives.

Steps to Migrate or Add Services from DevOps to MLOps



Conclusion

Migrating from DevOps to MLOps involves integrating machine learning workflows into your existing DevOps practices, ensuring seamless collaboration between teams, and leveraging automation to manage the ML lifecycle. By following these steps, organizations can harness the power of MLOps to deliver machine learning models more efficiently, reliably, and securely, ultimately driving greater business value from their AI initiatives.

References

1. DataOps: More Than DevOps for Data Pipelines. Wells, D. 2019. Eckerson Group
2. The Big Book of MLOps. eBook. Databricks.
3. DevOps Reading: <https://thenewstack.io/devops/>
4. Alexandra. February 28th, 2024. What Is SDLC? Understand the Software Development Life Cycle. Retrieved from: <https://stackify.com/what-is-sdlc/>
5. Isaac, Sacolick. April 1, 2024. What is CI/CD? Continuous integration and continuous delivery explained. InfoWorld.com
6. Data Version Control (DVC) Overview. Retrieved from: <https://dvc.org/doc/use-cases/versioning-data-and-models>
7. Kubernetes Overview. Retrieved from: <https://kubernetes.io/docs/concepts/overview>
8. Docker Overview. Retrieved from: <https://www.docker.com/>
9. Prometheus Overview. Retrieved from: <https://prometheus.io/>
10. Grafana Overview. Retrieved from: <https://grafana.com/>
11. The Future of Extract, Transform & Load tools (ETL). Retrieved from: <https://www2.deloitte.com/nl/nl/pages/technology/articles/future-etl-extract-transform-load-big-data.html>
12. General Data Protection Regulation (EU-GDPR) Overview. Retrieved from: <https://gdpr.eu/tag/gdpr/>
13. HIPAA Overview. Retrieved from: <https://www.hhs.gov/hipaa/index.html>
14. MLFlow Overview. Retrieved from: <https://www.run.ai/guides/machine-learning-operations/mlflow>
15. Derrick Mwit. 30th August, 2023. Deep Dive Into TensorBoard: Tutorial With Examples. Retrieved from: <https://neptune.ai/blog/tensorboard-tutorial>