

MLOps: A process or a platform?

The most important criteria for efficient and effective MLOps

MLOps defined

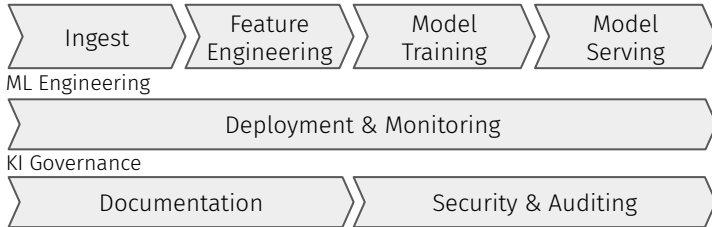
From one perspective, MLOps covers all processes in the context of AI governance. On the other hand, MLOps serves as a tool for data scientists, software developers, DevOps and platform engineers to efficiently create and sustainably operate AI solutions.

MLOps as a process defines the steps, results and quality points such as traceability, transparency, repeatability and documentation.

MLOps as a platform provides efficient technical means and all functions to optimally support each role in its individual competence and responsibility as well as in collaboration.

MLOps as a Process

Data Engineering, Data Science, Application Development



MLOps defines standardized, efficient processes over the entire life cycle of an ML system, from data procurement to the construction of models and their operation (serving). Supporting processes for deployment, security and monitoring must be fully integrated for MLOps to be effective. This requires a wide range of skills and IT expertise - a key reason the implementation of MLOps from scratch, lacking a specialized platform, is considered very complex.

MLOps as a Platform

omega-ml meets all the requirements of a modern, efficient and future-proof MLOps implementation. This includes all the functions that powerful data science and DevOps teams need in practice:

- **Repository** as a central database for all models, data, scripts and pipelines
- **Metadata** form the backbone of the repository and enable control of all data sources, artifacts and documentation
- **Tracking** Automatically records model training (experiments) and productive execution (predictions), enabling monitoring and assessing data and model drift
- **Runtime** executes pipelines, models and customized services in a scalable, controlled and secure manner, online or in batch processing.

MLOps = DevOps for AI?

MLOps is often characterized as "DevOps for AI", but there are significant differences. This is why a typical DevOps approach is considered inadequate for MLOps purpose.

Aspect	DevOps	MLOps
Deployment artifacts	Executable code, infrastructure (IaC)	ML models, data, pipelines (Code/conf.)
Basis for quality assurance	Functional and integration tests, anonymized test data, test system	Historic and current productive data, live system
Compute Capacity	Mainly in production	Mainly during training & development
Deployment Type	Self-contained (singular unit)	Additive (existing platform)
Monitoring	System Performance	Data Quality Model Performance

Checklist

Functional Requirements

- Process coverage as a self-service**
Data ingestion, feature Engineering, model training, model serving (no DevOps skills required) ✓
- MLOps infrastructure**
Repository for models, data, pipelines & scripts,, tracking & drift-monitoring. Runtime for online and batch services, model serving, plugin extensibility ✓
- Data access and storage**
SQL DB, NoSQL, Streaming (e.g. Kafka), REST API ✓
- Model Serving, App Serving**
Standard Model REST API, Customized Service Endpoints (OpenAPI/Swagger), Apps & Dashboards ✓
- Experiments, Monitoring and Drift Detection**
Automated and explicit recording of metrics, parameters, versioning of models and data, in development and production ✓

Non-Functional Requirements

- Security**
Integration with Identity Providers (IDP), data and platform security, opt. secure computing nodes ✓
- IT Integration**
Windows (Client), Linux (Client/Server), Bare Metal, Kubernetes, Cloud, On-/Off-Premises ✓
- Scalability**
Dynamic scaling of the runtime and extensibility by data scientist-provided packages ✓
- Deployment and Maintainability**
Deployment of models, data and scripts
Is fully automated and integrated with CI/CD ✓