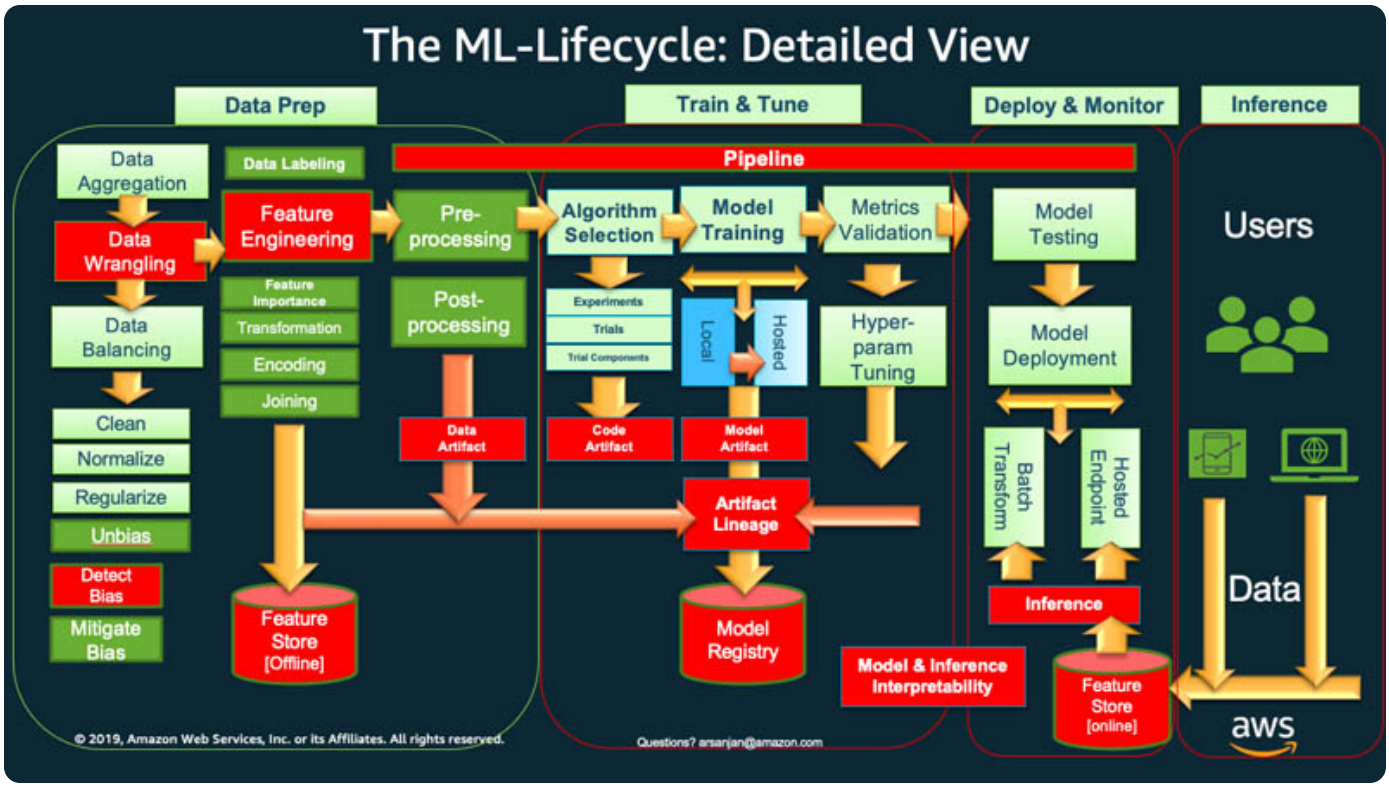
What is MLOps?

What is MLOps?

Machine learning operations (MLOps) are a set of practices that automate and simplify machine learning (ML) workflows and deployments. Machine learning and artificial intelligence (AI) are core capabilities that you can implement to solve complex real-world problems and deliver value to your customers. MLOps is an ML culture and practice that unifies ML application development (Dev) with ML system deployment and operations (Ops). Your organization can use MLOps to automate and standardize processes across the ML lifecycle. These processes include model development, testing, integration, release, and infrastructure management.

Why is MLOps required?



At a high level, to begin the machine learning lifecycle, your organization typically has to start with data preparation. You fetch data of different types from various sources, and perform activities like aggregation, duplicate cleaning, and feature engineering.

After that, you use the data to train and validate the ML model. You can then deploy the trained and validated model as a prediction service that other applications can access through APIs.

Exploratory data analysis often requires you to experiment with different models until the best model version is ready for deployment. It leads to frequent model version deployments and data versioning. Experiment tracking and ML training pipeline management are essential before your applications can integrate or consume the model in their code.

MLOps is critical to systematically and simultaneously manage the release of new ML models with application code and data changes. An optimal MLOps implementation treats the ML assets similarly to other continuous integration and delivery (CI/CD) environment software assets. You deploy ML models alongside the applications and services they use and those that consume them as part of a unified release process.

What are the principles of MLOps?

Next, we explain four key principles of MLOps.

Version control

This process involves tracking changes in the machine learning assets so you can reproduce results and roll back to previous versions if necessary. Every ML training code or model specification goes through a code review phase. Each is versioned to make the training of ML models reproducible and auditable.

Reproducibility in an ML workflow is important at every phase, from data processing to ML model deployment. It means that each phase should produce identical results given the same input.

Automation

Automate various stages in the machine learning pipeline to ensure repeatability, consistency, and scalability. This includes stages from data ingestion, preprocessing, model training, and validation to deployment.

These are some factors that can trigger automated model training and deployment:

- Messaging
- Monitoring or calendar events
- Data changes
- Model training code changes
- Application code changes.

Automated testing helps you discover problems early for fast error fixes and learnings. Automation is more efficient with infrastructure as code (IaC). You can use tools to define and manage infrastructure. This helps ensure it's reproducible and can be consistently deployed across various environments.

Read about IaC »

Continuous X

Through automation, you can continuously run tests and deploy code across your ML pipeline.

In MLOps, *continuous* refers to four activities that happen continuously if any change is made anywhere in the system:

- *Continuous integration* extends the validation and testing of code to data and models in the pipeline
- *Continuous delivery* automatically deploys the newly trained model or model prediction service
- *Continuous training* automatically retrains ML models for redeployment
- *Continuous monitoring* concerns data monitoring and model monitoring using metrics related to business

Model governance

Governance involves managing all aspects of ML systems for efficiency. You should do many activities for governance:

- Foster close collaboration between data scientists, engineers, and business stakeholders
- Use clear documentation and effective communication channels to ensure everyone is aligned
- Establish mechanisms to collect feedback about model predictions and retrain models further
- Ensure that sensitive data is protected, access to models and infrastructure is secure, and compliance requirements are met

It's also essential to have a structured process to review, validate, and approve models before they go live. This can involve checking for fairness, bias, and ethical considerations.

What are the benefits of MLOps?

Machine learning helps organizations analyze data and derive insights for decision-making. However, it's an innovative and experimental field that comes with its own set of challenges. Sensitive data protection, small budgets, skills shortages, and continuously evolving technology limit a project's success. Without control and guidance, costs may spiral, and data science teams may not achieve their desired outcomes.

MLOps provides a map to guide ML projects toward success, no matter the constraints. Here are some key benefits of MLOps.

Faster time to market

MLOps provides your organization with a framework to achieve your data science goals more quickly and efficiently. Your developers and managers can become more strategic and agile in model management. ML engineers can provision infrastructure through declarative configuration files to get projects started more smoothly.

Automating model creation and deployment results in faster go-to-market times with lower operational costs. Data scientists can rapidly explore an organization's data to deliver more business value to all.

Improved productivity

MLOps practices boost productivity and accelerate the development of ML models. For instance, you can standardize the development or experiment environment. Then, your ML engineers can launch new projects, rotate between projects, and reuse ML models across applications. They can create repeatable processes for rapid experimentation and model training. Software engineering teams can collaborate and coordinate through the ML software development lifecycle for greater efficiency.

Efficient model deployment

MLOps improves troubleshooting and model management in production. For instance, software engineers can monitor model performance and reproduce behavior for troubleshooting. They can track and centrally manage model versions and pick and choose the right one for different business use cases.

When you integrate model workflows with continuous integration and continuous delivery (CI/CD) pipelines, you limit performance degradation and maintain quality for your model. This is true even after upgrades and model tuning.

How to implement MLOps in the organization

There are three levels of MLOps implementation, depending upon the automation maturity within your organization.

MLOps level 0

Manual ML workflows and a data-scientist-driven process characterize *level 0* for organizations just starting with machine learning systems.

Every step is manual, including data preparation, ML training, and model performance and validation. It requires a manual transition between steps, and each step is interactively run and managed. The data scientists typically hand over trained models as artifacts that the engineering team deploys on API infrastructure.

The process separates data scientists who create the model and engineers who deploy it. Infrequent releases mean the data science teams may retrain models only a few times a year. There are no CI/CD considerations for ML models with the rest of the application code. Similarly, active performance monitoring is nonexistent.

MLOps level 1

Organizations that want to train the same models with new data frequently require *level 1* maturity implementation. MLOps level 1 aims to train the model continuously by automating the ML pipeline.

In level 0, you deploy a trained model to production. In contrast, for level 1, you deploy a training pipeline that runs recurrently to serve the trained model to your other apps. At a minimum, you achieve continuous delivery of the model prediction service.

Level 1 maturity has these characteristics:

- Rapid ML experiment steps that involve significant automation
- Continuous training of the model in production with fresh data as live pipeline triggers
- Same pipeline implementation across development, preproduction, and production environments

Your engineering teams work with data scientists to create modularized code components that are reusable, composable, and potentially shareable across ML pipelines. You also create a centralized feature store that standardizes the storage, access, and definition of features for ML training and serving. In addition, you can manage metadata—like information about each run of the pipeline and reproducibility data.

MLOps level 2

MLOps *level 2* is for organizations that want to experiment more and frequently create new models that require continuous training. It's suitable for tech-driven companies that update their models in minutes, retrain them hourly or daily, and simultaneously redeploy them on thousands of servers.

As there are several ML pipelines in play, a MLOps level 2 setup requires all of the MLOps level 1 setup. It also requires these:

- An ML pipeline orchestrator
- A model registry for tracking multiple models

The following three stages repeat at scale for several ML pipelines to ensure model continuous delivery.

Build the pipeline

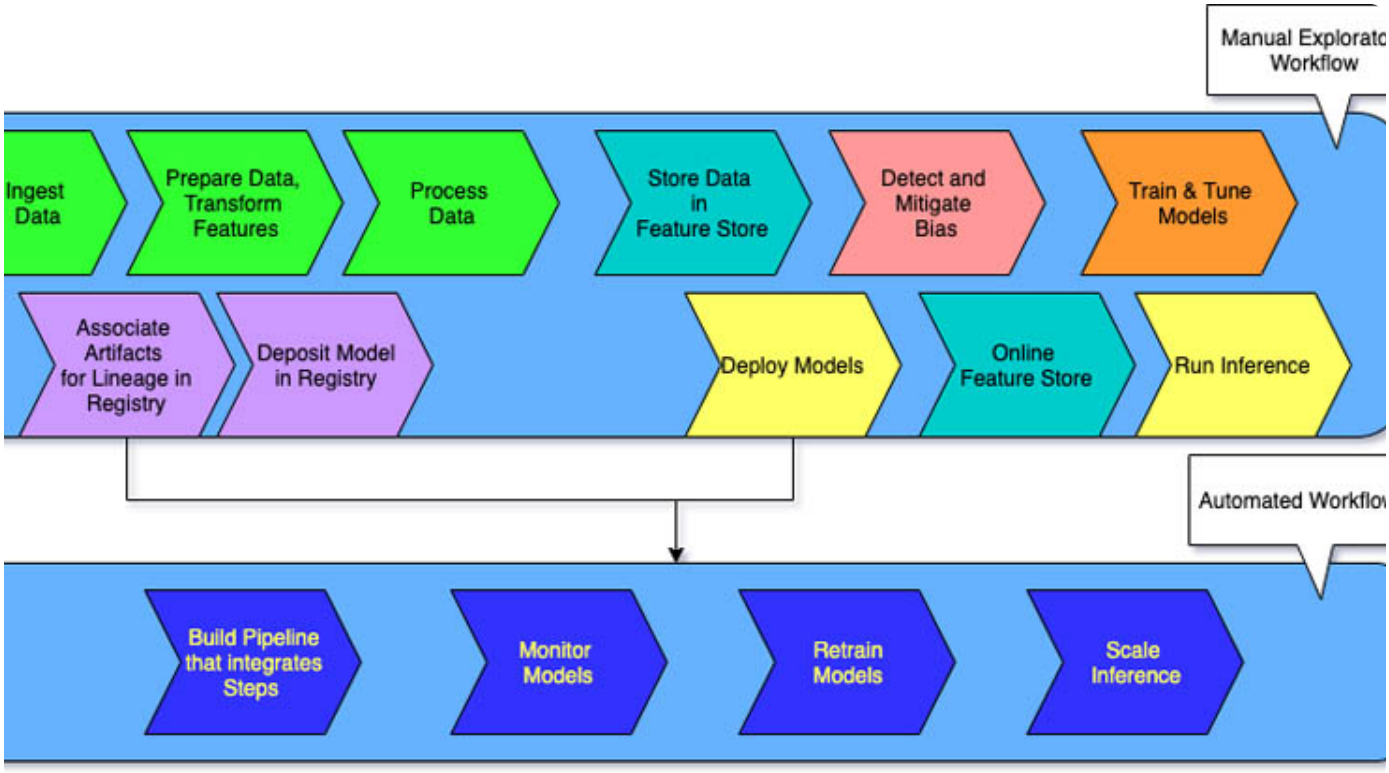
You iteratively try out new modeling and new ML algorithms while ensuring experiment steps are orchestrated. This stage outputs the source code for your ML pipelines. You store the code in a source repository.

Deploy the pipeline

Next, you build the source code and run tests to obtain pipeline components for deployment. The output is a deployed pipeline with the new model implementation.

Serve the pipeline

Finally, you serve the pipeline as a prediction service for your applications. You collect statistics on the deployed model prediction service from live data. This stage output is a trigger to run the pipeline or a new experiment cycle.



What is the difference between MLOps and DevOps?

MLOps and DevOps are both practices that aim to improve processes where you develop, deploy, and monitor software applications.

DevOps aims to bridge the gap between development and operations teams. DevOps helps ensure that code changes are automatically tested, integrated, and deployed to production efficiently and reliably. It promotes a culture of collaboration to achieve faster release cycles, improved application quality, and more efficient use of resources.

MLOps, on the other hand, is a set of best practices specifically designed for machine learning projects. While it can be relatively straightforward to deploy and integrate traditional software, ML models present unique challenges. They involve data collection, model training, validation, deployment, and continuous monitoring and retraining.

MLOps focuses on automating the ML lifecycle. It helps ensure that models are not just developed but also deployed, monitored, and retrained systematically and repeatedly. It brings DevOps principles to ML. MLOps results in faster deployment of ML models, better accuracy over time, and stronger assurance that they provide real business value.

How can AWS support your MLOps requirements?

Amazon SageMaker is a fully managed service that you can use to prepare data and build, train, and deploy ML models. It's suitable for any use case with fully managed infrastructure, tools, and workflows.

SageMaker provides purpose-built tools for MLOps to automate processes across the ML lifecycle. By using Sagemaker for MLOps tools, you can quickly achieve level 2 MLOps maturity at scale.

Here are key features of SageMaker you can use:

- Use SageMaker Experiments to track artifacts related to your model training jobs, like parameters, metrics, and datasets.
- Configure SageMaker Pipelines to run automatically at regular intervals or when certain events are triggered.
- Use SageMaker Model Registry to track model versions. You can also track their metadata, such as use case grouping, and model performance metrics baselines in a central repository. You can use this information to choose the best model based on your business requirements.

Get started with MLOps on Amazon Web Services (AWS) by creating an account today.

Next Steps on AWS

Learn

Resources

Developers

Help

Back to top