

# Machine Learning operations maturity model

The purpose of this maturity model is to help clarify the Machine Learning Operations (MLOps) principles and practices. The maturity model shows the continuous improvement in the creation and operation of a production level machine learning application environment. You can use it as a metric for establishing the progressive requirements needed to measure the maturity of a machine learning production environment and its associated processes.

## Maturity model

The MLOps maturity model helps clarify the Development Operations (DevOps) principles and practices necessary to run a successful MLOps environment. It's intended to identify gaps in an existing organization's attempt to implement such an environment. It's also a way to show you how to grow your MLOps capability in increments rather than overwhelm you with the requirements of a fully mature environment. Use it as a guide to:

- Estimate the scope of the work for new engagements.
- Establish realistic success criteria.
- Identify deliverables you'll hand over at the conclusion of the engagement.

As with most maturity models, the MLOps maturity model qualitatively assesses people/culture, processes/structures, and objects/technology. As the maturity level increases, the probability increases that incidents or errors will lead to improvements in the quality of the development and production processes.

The MLOps maturity model encompasses five levels of technical capability:

 Expand table

Level	Description	Highlights	Technology
0	No MLOps	<ul style="list-style-type: none"><li>• Difficult to manage full machine learning model lifecycle</li><li>• The teams are disparate and releases are painful</li><li>• Most systems exist as "black boxes," little feedback during/post deployment</li></ul>	<ul style="list-style-type: none"><li>• Manual builds and deployments</li><li>• Manual testing of model and application</li><li>• No centralized tracking of model performance</li><li>• Training of model is manual</li></ul>
1	DevOps but no MLOps	<ul style="list-style-type: none"><li>• Releases are less painful than No MLOps, but rely on Data Team for every new model</li><li>• Still limited feedback on how well a model performs in production</li><li>• Difficult to trace/reproduce results</li></ul>	<ul style="list-style-type: none"><li>• Automated builds</li><li>• Automated tests for application code</li></ul>
2	Automated Training	<ul style="list-style-type: none"><li>• Training environment is fully managed and traceable</li><li>• Easy to reproduce model</li><li>• Releases are manual, but low friction</li></ul>	<ul style="list-style-type: none"><li>• Automated model training</li><li>• Centralized tracking of model training performance</li><li>• Model management</li></ul>

Level	Description	Highlights	Technology
3	<a href="#">Automated Model Deployment</a>	<ul style="list-style-type: none"> <li>Releases are low friction and automatic</li> <li>Full traceability from deployment back to original data</li> <li>Entire environment managed: train &gt; test &gt; production</li> </ul>	<ul style="list-style-type: none"> <li>Integrated A/B testing of model performance for deployment</li> <li>Automated tests for all code</li> <li>Centralized tracking of model training performance</li> </ul>
4	<a href="#">Full MLOps Automated Operations</a>	<ul style="list-style-type: none"> <li>Full system automated and easily monitored</li> <li>Production systems are providing information on how to improve and, in some cases, automatically improve with new models</li> <li>Approaching a zero-downtime system</li> </ul>	<ul style="list-style-type: none"> <li>Automated model training and testing</li> <li>Verbose, centralized metrics from deployed model</li> </ul>

The tables that follow identify the detailed characteristics for that level of process maturity. The model will continue to evolve.

## Level 0: No MLOps

 Expand table

People	Model Creation	Model Release	Application Integration
<ul style="list-style-type: none"> <li>Data scientists: siloed, not in regular communications with the larger team</li> <li>Data engineers (<i>if exists</i>): siloed, not in regular communications with the larger team</li> <li>Software engineers: siloed, receive model remotely from the other team members</li> </ul>	<ul style="list-style-type: none"> <li>Data gathered manually</li> <li>Compute is likely not managed</li> <li>Experiments aren't predictably tracked</li> <li>End result might be a single model file manually handed off with inputs/outputs</li> </ul>	<ul style="list-style-type: none"> <li>Manual process</li> <li>Scoring script might be manually created well after experiments, not version controlled</li> <li>Release handled by data scientist or data engineer alone</li> </ul>	<ul style="list-style-type: none"> <li>Heavily reliant on data scientist expertise to implement</li> <li>Manual releases each time</li> </ul>


## Level 1: DevOps no MLOps

 Expand table

People	Model Creation	Model Release	Application Integration
<ul style="list-style-type: none"> <li>Data scientists: siloed, not in regular</li> </ul>	<ul style="list-style-type: none"> <li>Data pipeline gathers data automatically</li> </ul>	<ul style="list-style-type: none"> <li>Manual process</li> </ul>	<ul style="list-style-type: none"> <li>Basic integration tests exist for the</li> </ul>


People	Model Creation	Model Release	Application Integration
<p>communications with the larger team</p> <ul style="list-style-type: none"><li>• Data engineers (if exists): siloed, not in regular communication with the larger team</li><li>• Software engineers: siloed, receive model remotely from the other team members</li></ul>	<ul style="list-style-type: none"><li>• Compute is or isn't managed</li><li>• Experiments aren't predictably tracked</li><li>• End result might be a single model file manually handed off with inputs/outputs</li></ul>	<ul style="list-style-type: none"><li>• Scoring script might be manually created well after experiments, likely version controlled</li><li>• Is handed off to software engineers</li></ul>	<p>model</p> <ul style="list-style-type: none"><li>• Heavily reliant on data scientist expertise to implement model</li><li>• Releases automated</li><li>• Application code has unit tests</li></ul>

## Level 2: Automated Training

 Expand table


People	Model Creation	Model Release	Application Integration
<ul style="list-style-type: none"><li>• Data scientists: Working directly with data engineers to convert experimentation code into repeatable scripts/jobs</li><li>• Data engineers: Working with data scientists</li><li>• Software engineers: siloed, receive model remotely from the other team members</li></ul>	<ul style="list-style-type: none"><li>• Data pipeline gathers data automatically</li><li>• Compute managed</li><li>• Experiment results tracked</li><li>• Both training code and resulting models are version controlled</li></ul>	<ul style="list-style-type: none"><li>• Manual release</li><li>• Scoring script is version controlled with tests</li><li>• Release managed by Software engineering team</li></ul>	<ul style="list-style-type: none"><li>• Basic integration tests exist for the model</li><li>• Heavily reliant on data scientist expertise to implement model</li><li>• Application code has unit tests</li></ul>

## Level 3: Automated Model Deployment

 Expand table

People	Model Creation	Model Release	Application Integration
<ul style="list-style-type: none"><li>• Data scientists: Working directly with data engineers to convert experimentation code into repeatable scripts/jobs</li><li>• Data engineers: Working with data scientists and software engineers to manage inputs/outputs</li><li>• Software engineers: Working with data engineers to automate model integration into application code</li></ul>	<ul style="list-style-type: none"><li>• Data pipeline gathers data automatically</li><li>• Compute managed</li><li>• Experiment results tracked</li><li>• Both training code and resulting models are version controlled</li></ul>	<ul style="list-style-type: none"><li>• Automatic release</li><li>• Scoring script is version controlled with tests</li><li>• Release managed by continuous delivery (CI/CD) pipeline</li></ul>	<ul style="list-style-type: none"><li>• Unit and integration tests for each model release</li><li>• Less reliant on data scientist expertise to implement model</li><li>• Application code has unit/integration tests</li></ul>

## Level 4: Full MLOps Automated Retraining

 Expand table

People	Model Creation	Model Release	Application Integration
<ul style="list-style-type: none"><li>• Data scientists: Working directly with data engineers to convert experimentation code into repeatable scripts/jobs. Working with software engineers to identify markers for data engineers</li><li>• Data engineers: Working with data scientists and software engineers to manage inputs/outputs</li><li>• Software engineers: Working with data engineers to automate model integration into application code. Implementing post-deployment metrics gathering</li></ul>	<ul style="list-style-type: none"><li>• Data pipeline gathers data automatically</li><li>• Retraining triggered automatically based on production metrics</li><li>• Compute managed</li><li>• Experiment results tracked</li><li>• Both training code and resulting models are version controlled</li></ul>	<ul style="list-style-type: none"><li>• Automatic Release</li><li>• Scoring Script is version controlled with tests</li><li>• Release managed by continuous integration and CI/CD pipeline</li></ul>	<ul style="list-style-type: none"><li>• Unit and Integration tests for each model release</li><li>• Less reliant on data scientist expertise to implement model</li><li>• Application code has unit/integration tests</li></ul>

## Next step

MLOps and GenAIOps for AI workloads

## Related resources

- [Learning path: Introduction to machine learning operations \(MLOps\)](#)
- [MLOps: Model management, deployment, and monitoring with Azure Machine Learning](#)
- [Machine Learning registries for MLOps](#)

- [Orchestrate MLOps by using Azure Databricks](#)
- [Machine learning operations](#)