

# MLOps 개요



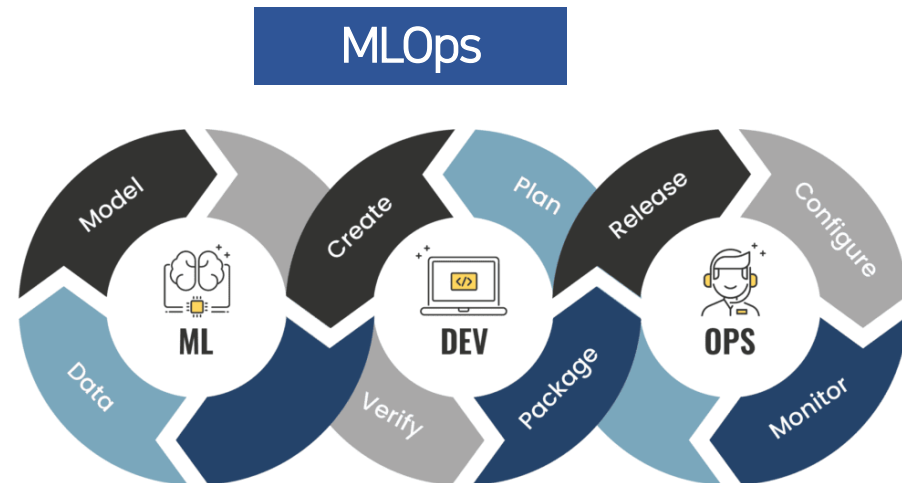
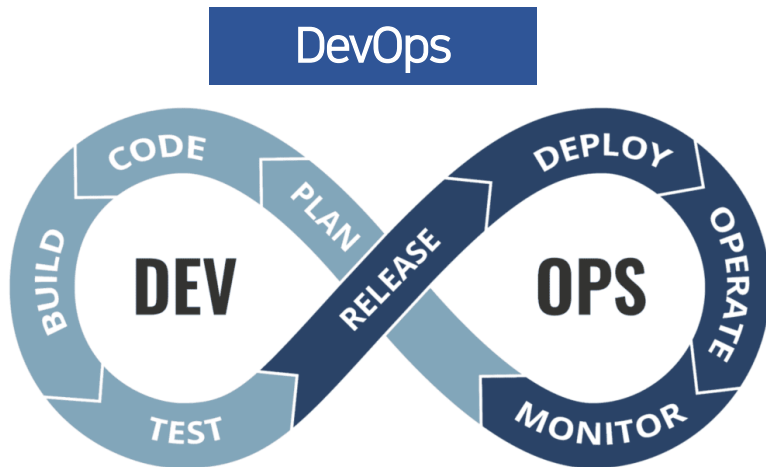
# MLOps

- MLOps는 운영 환경에서 머신러닝 모델을 안정적이고 효율적으로 배포하고 유지 관리하는 것을 목표로 하는 패러다임입니다. Machine Learning과 소프트웨어 개발분야의 지속적인 개발 관행인 DevOps의 합성어입니다.

<https://en.wikipedia.org/wiki/MLOps>

- MLOps는 머신러닝 엔지니어링의 핵심 기능으로, 머신러닝 모델을 프로덕션에 적용하고 유지 관리 및 모니터링하는 프로세스를 간소화하는 데 중점을 둡니다. MLOps는 협업 기능으로, 데이터 과학자, 개발 엔지니어, IT 부서로 구성되는 경우가 많습니다.

<https://www.databricks.com/glossary/mlops>

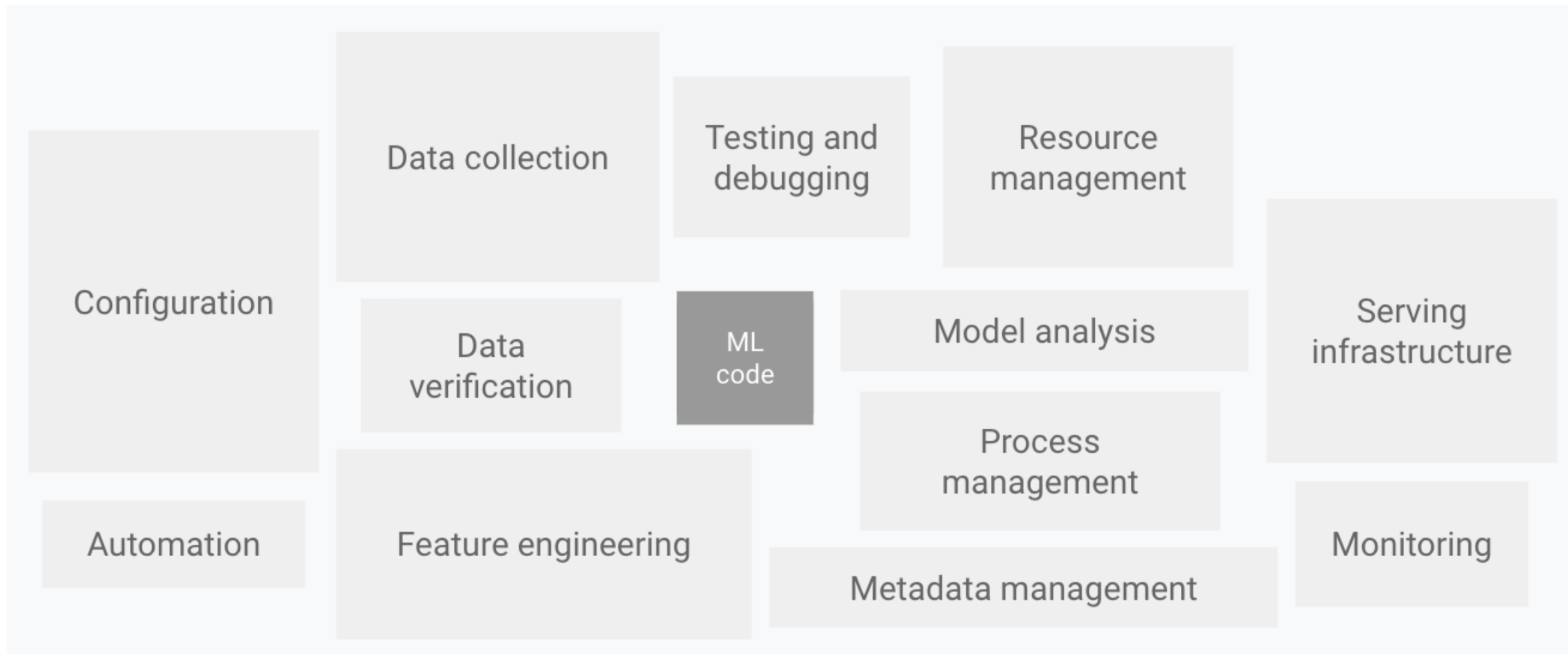


# DevOps vs MLOps

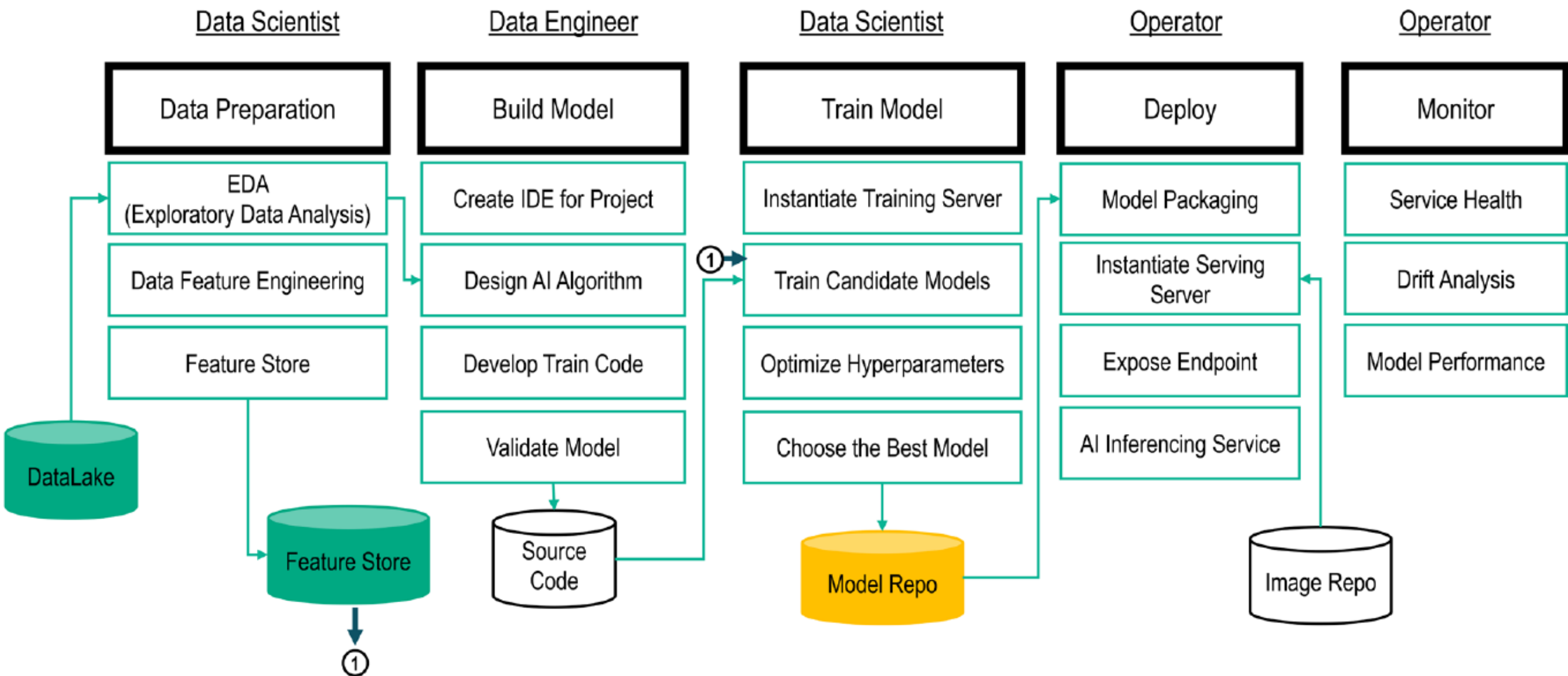
구분	DevOps	MLOps
정의	<ul style="list-style-type: none"><li>▪ 소프트웨어 개발자 간의 소통과 협업을 강조하는 개발 문화</li><li>▪ Dev + Ops</li></ul>	<ul style="list-style-type: none"><li>▪ 데이터사이언티스트와 머신러닝 엔지니어간 소통과 협업을 강조하는 ML 엔지니어링 문화</li><li>▪ ML + Ops</li></ul>
CI (Continuous Integration)	<ul style="list-style-type: none"><li>▪ 개발자를 위한 자동화 프로세스인 지속적인 통합</li></ul>	<ul style="list-style-type: none"><li>▪ 데이터, 데이터 스키마, 모델에 대해 테스트하고 검증</li></ul>
CD (Continuous Development)	<ul style="list-style-type: none"><li>▪ 지속적인 서비스 제공 및 지속적인 배포</li></ul>	<ul style="list-style-type: none"><li>▪ 모델 학습 및 예측 서비스를 자동으로 배포해야 하는 시스템</li></ul>
CT (Continuous Training)		<ul style="list-style-type: none"><li>▪ 머신러닝 시스템 고유 속성</li><li>▪ 모델을 자동으로 재학습시키고 서비스 제공</li></ul>

# ML시스템 구성요소

실제 ML 시스템의 일부만 ML 코드로 구성됩니다. 필수 주변 요소는 광범위하고 복잡합니다.

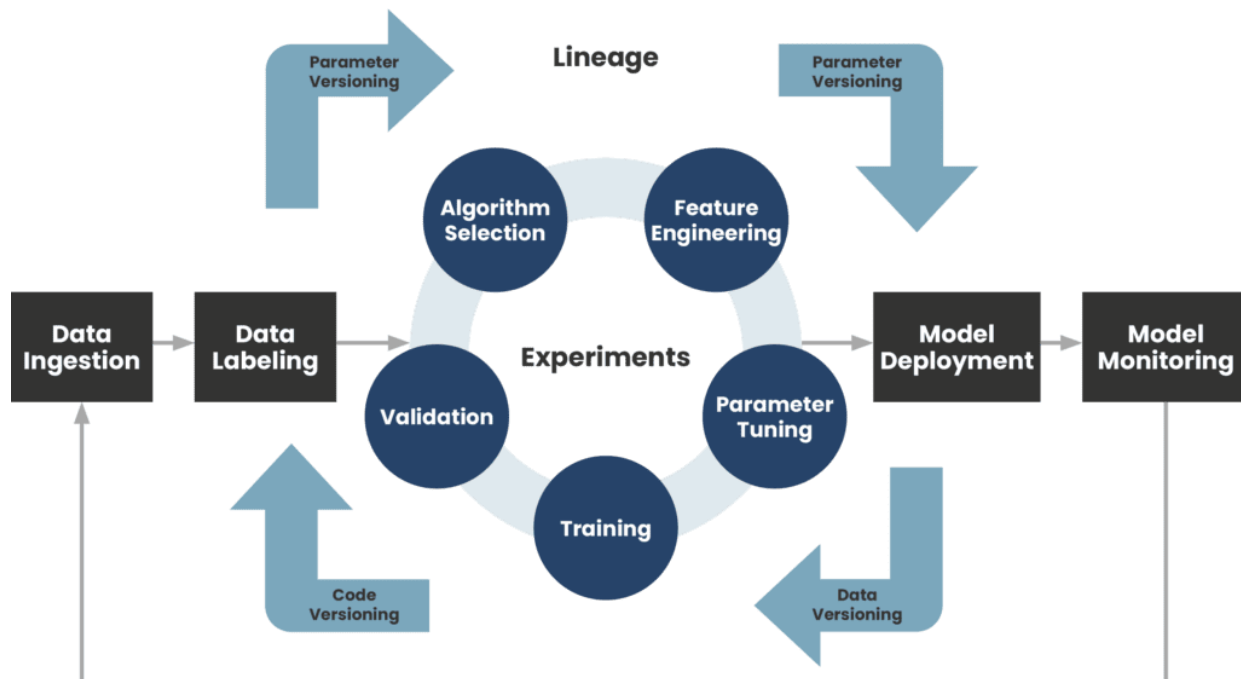


# ML 시스템 구축 프로세스



# ML시스템 특성

- 팀 기술
  - 탐색적 데이터 분석, 모델 개발, 실험에 중점을 두는 데이터 과학자 또는 머신러닝 연구원을 포함
  - 프로덕션 수준의 서비스를 빌드 가능한 소프트웨어 엔지니어가 없을 수도 있음
- 개발
  - 머신러닝은 기본적으로 실험적임
  - 피처, 알고리즘, 모델링 기법, 매개변수 성을 다양하게 시도
  - 효과가 있었던 것을 추적하고, 코드 재사용성을 극대화하면서 재현성 유지가 필요
- 테스트
  - 단위테스트, 통합 테스트 외에도 데이터 검증, 학습된 모델 품질 평가, 모델 검증이 필요
- 배포 : 모델을 자동으로 재학습시키고 배포하기 위해 다단계 파이프라인 구성과 자동화가 필요
- 프로덕션 : ML 모델 성능은 최적화되지 않은 코딩, 지속적으로 변하는 데이터로 인해 성능이 저하될 수 있음



# AI/ML 도입하는 기업의 현실적인 문제들

AI 도입 시, 실제 비즈니스 효과 달성을 위해 많은 시간 및 비용 소요

## 제한된 ML 학습 환경

( 더 많은 혁신적 아이디어 검증 한계 )

- 적시에 고품질 데이터 제공 어려움
- 많은 모델의 동시학습 한계 (수 천개 이상의 ML 모델)
- 거대 언어모델 학습 비용 이슈

## ML 공유/협업 이슈

( ML 중복투자 및 재사용 불가 )

- ML모델의 효율적 관리 체계 부재
- Feature 재사용 어려움
- ML 모델의 재현 어려움  
(code , data, parameter, lib)

## Biz. 성과 연계 어려움

( 사업의 적시성 확보 이슈 )

- ML 모델의 배포 복잡성/제약
- 불규칙한 모델의 예측 정확도
- 실제 운영 환경의 다양한 제약  
(예측 불가능한 자원, 규제, Biz Rule등)

# 조직 확장에 따른 ML 모델 관리의 한계

수 많은 ML 모델을 관리하는 것은 기업의 생산성 및 비용 관점에서 매우 중요

## ML 학습 이력 확인 어려움

- 개인 별로 자체적인 ML 학습 이력 관리 (File, Excel 등)
- 수 많은 ML실험 중 최고의 정확도를 가진 모델은 ?
- 현재 배포된 ML 모델의 학습 이력을 확인하려면?

## ML 모델의 공유,재현 어려움

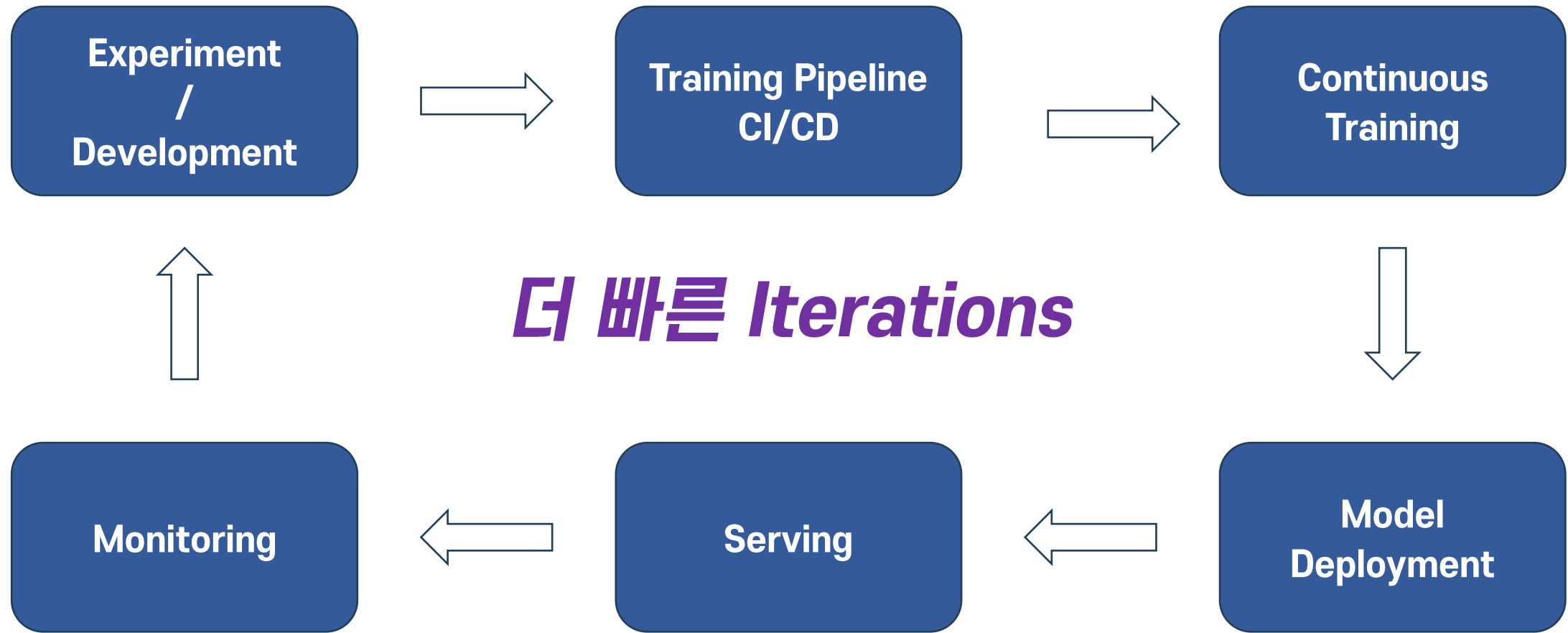
- 실행 가능한 Code 및 Library version 불일치
- 학습에 활용된 Feature의 불일치 (당시 데이터 부재)
- 모델 학습에 사용된 Hyperparameter 미확인

## 운영 중인 ML 모델 관리 필요

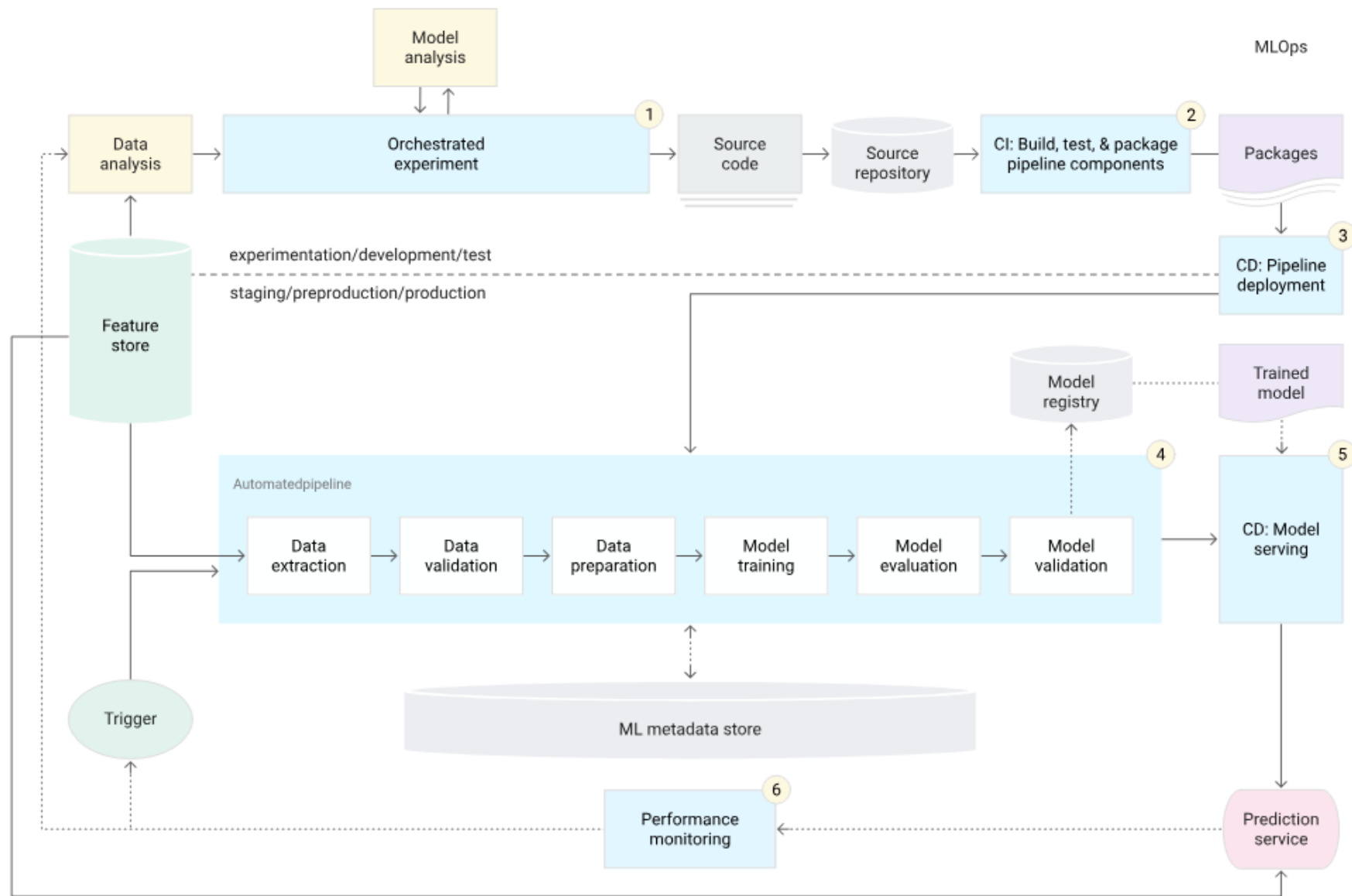
- 비즈니스 적용을 위한 ML 버전 관리 어려움
- 배포된 ML 모델이 어떤 Code 및 Data로 학습되었는지 확인불가 (Black-Box)



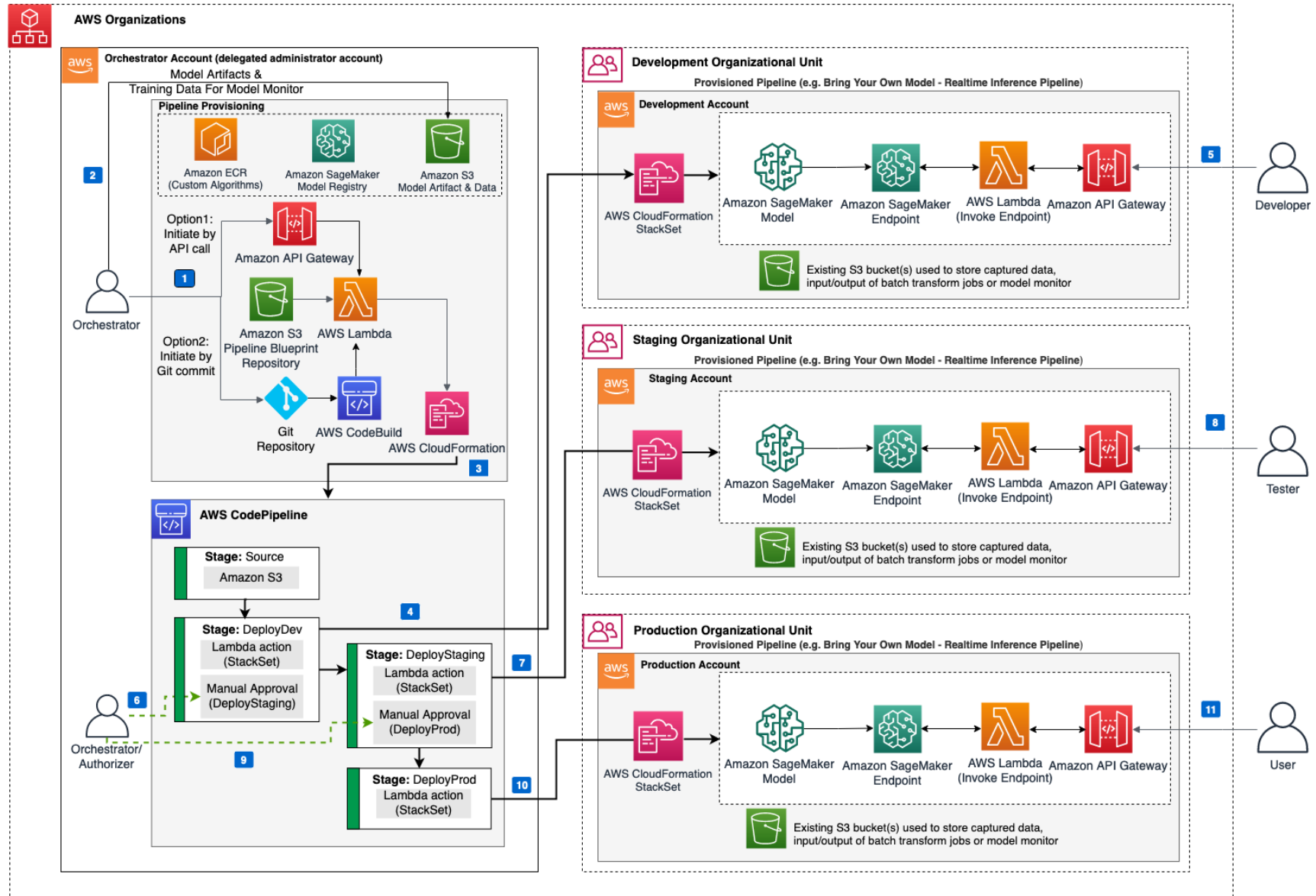
# MLOps 목표



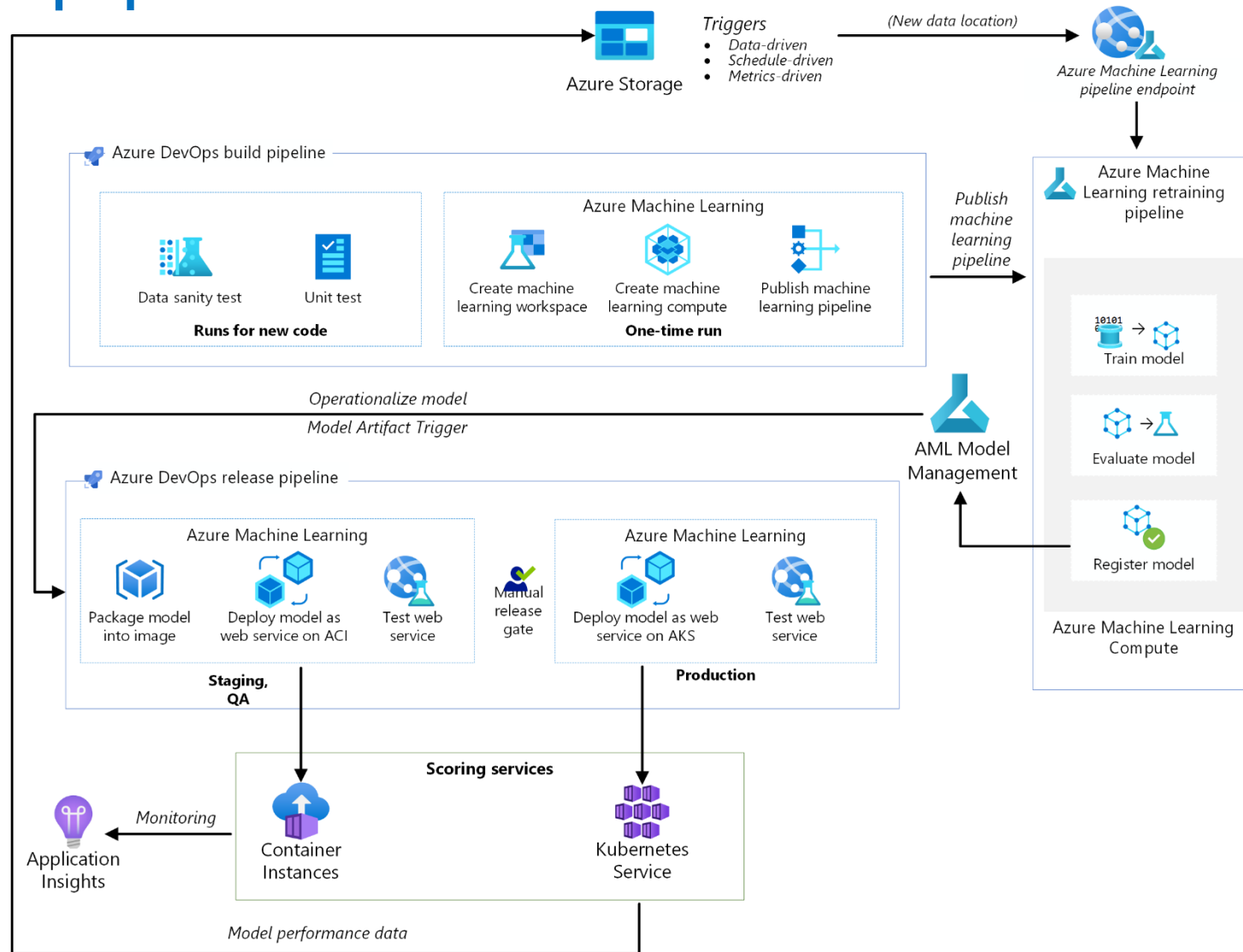
# MLOps 아키텍처



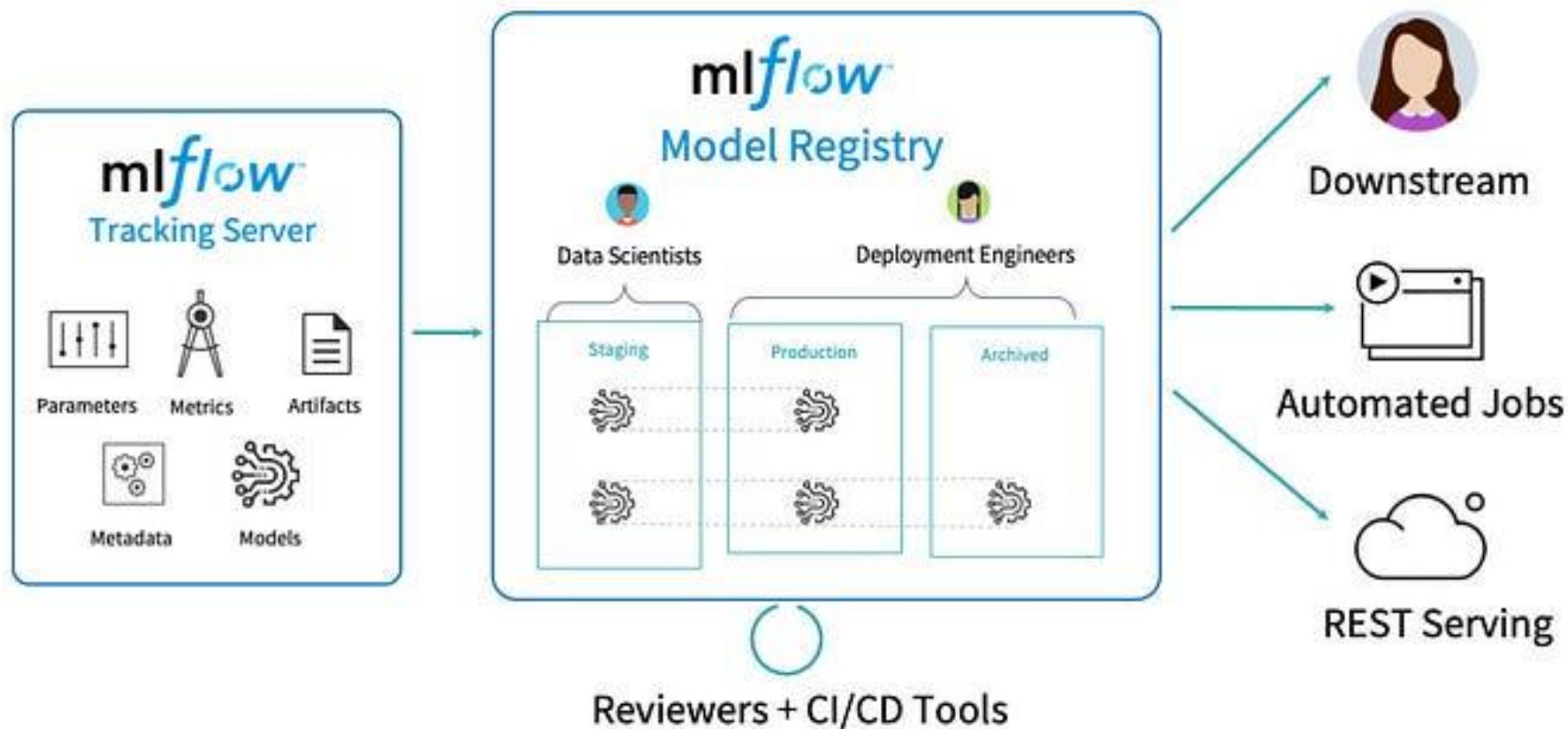
# MLOps 아키텍처



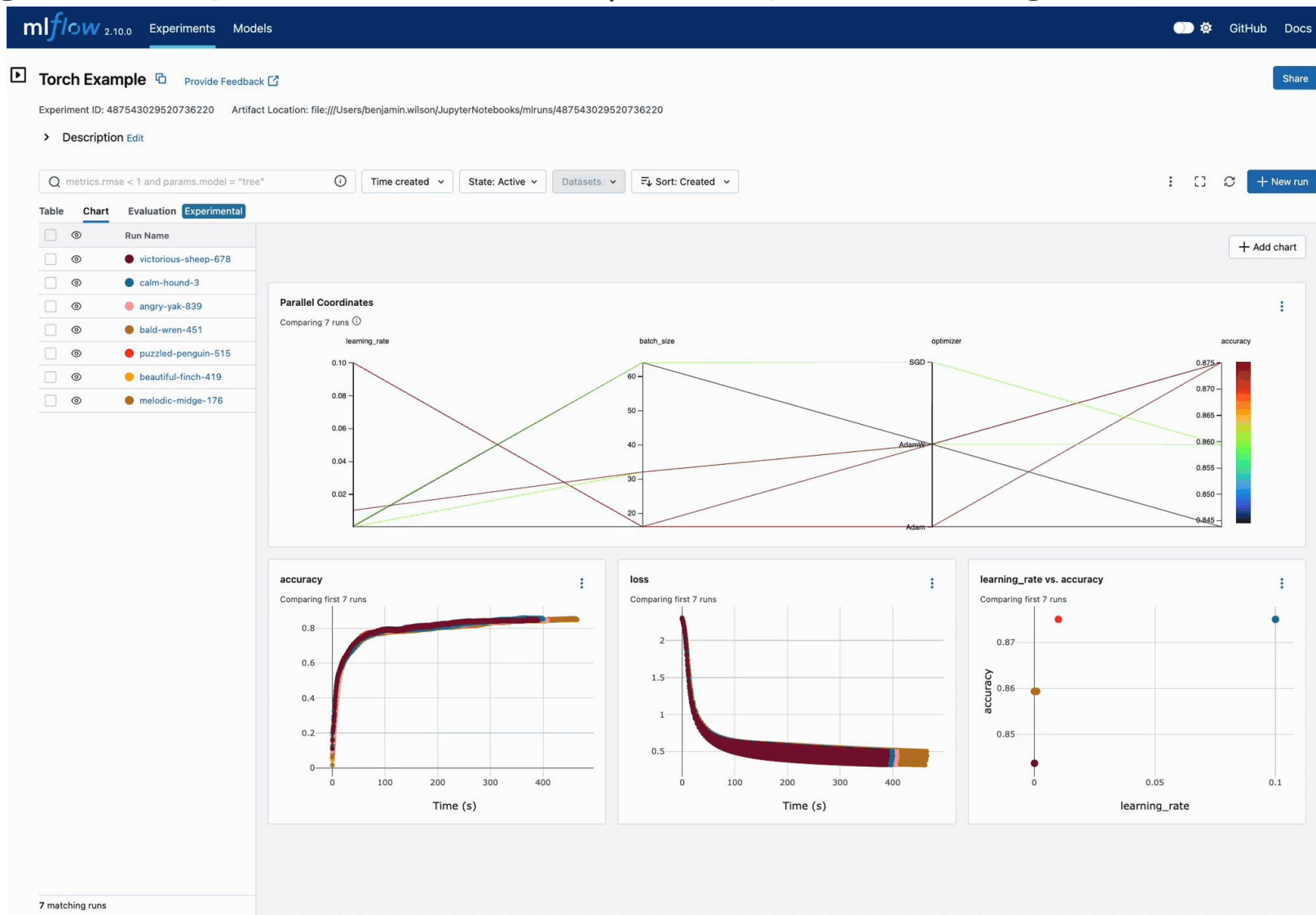
# MLOps 아키텍처



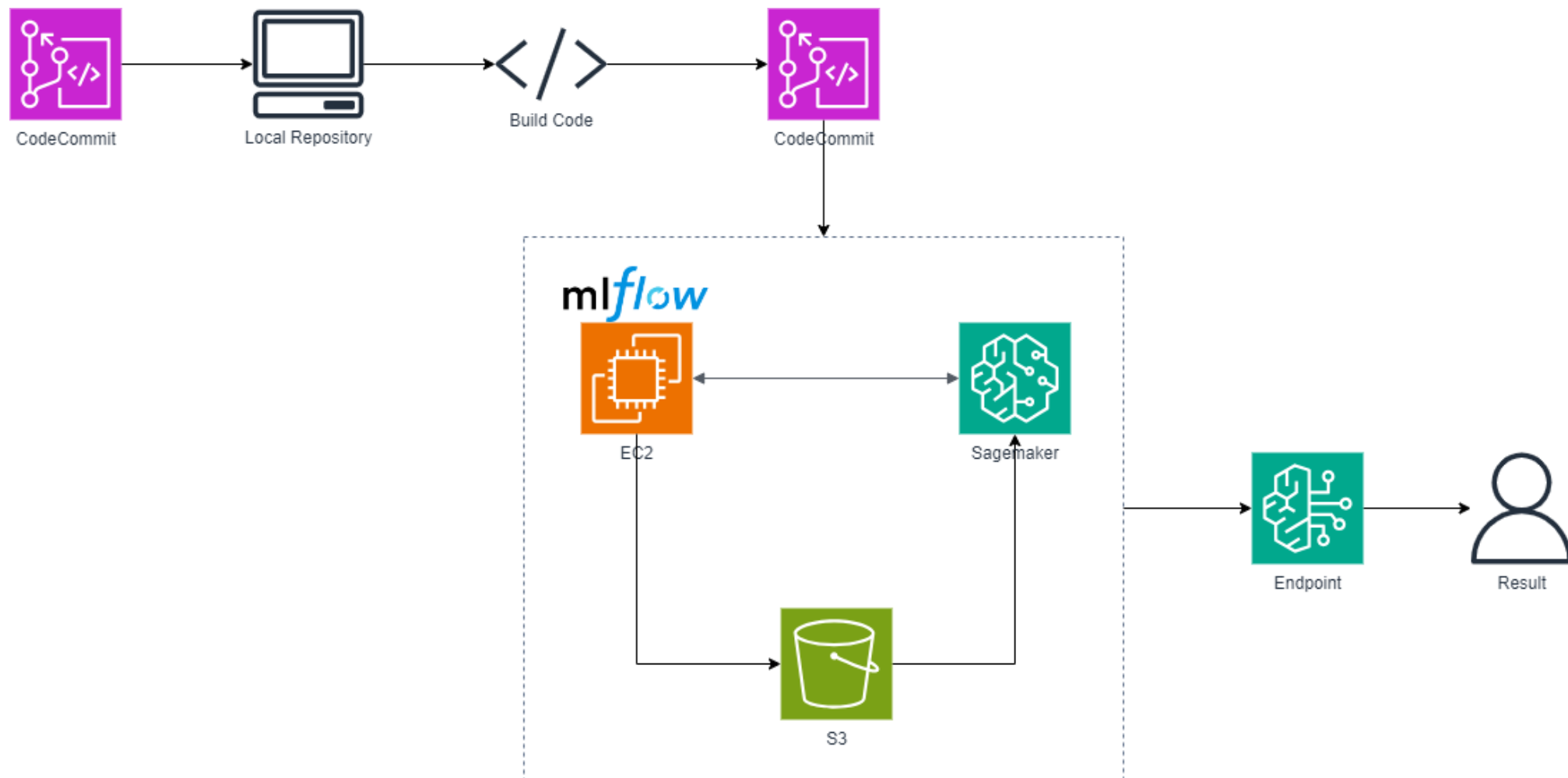
MLflow는 머신러닝 프로세스의 복잡성을 처리하는 데 도움을 주는 오픈 소스 플랫폼입니다.  
MLflow는 머신러닝 프로젝트의 전체 수명 주기에 중점을 두어 각 단계를 관리, 추적 및 재현할 수 있도록 합니다.



차트를 사용하여 모델 학습 수렴을 쉽게 비교하고, 훈련 반복에서 우수한 구성 세트를 빠르게 식별할 수 있습니다



# MLOps 실습 아키텍처



Thank you 😊