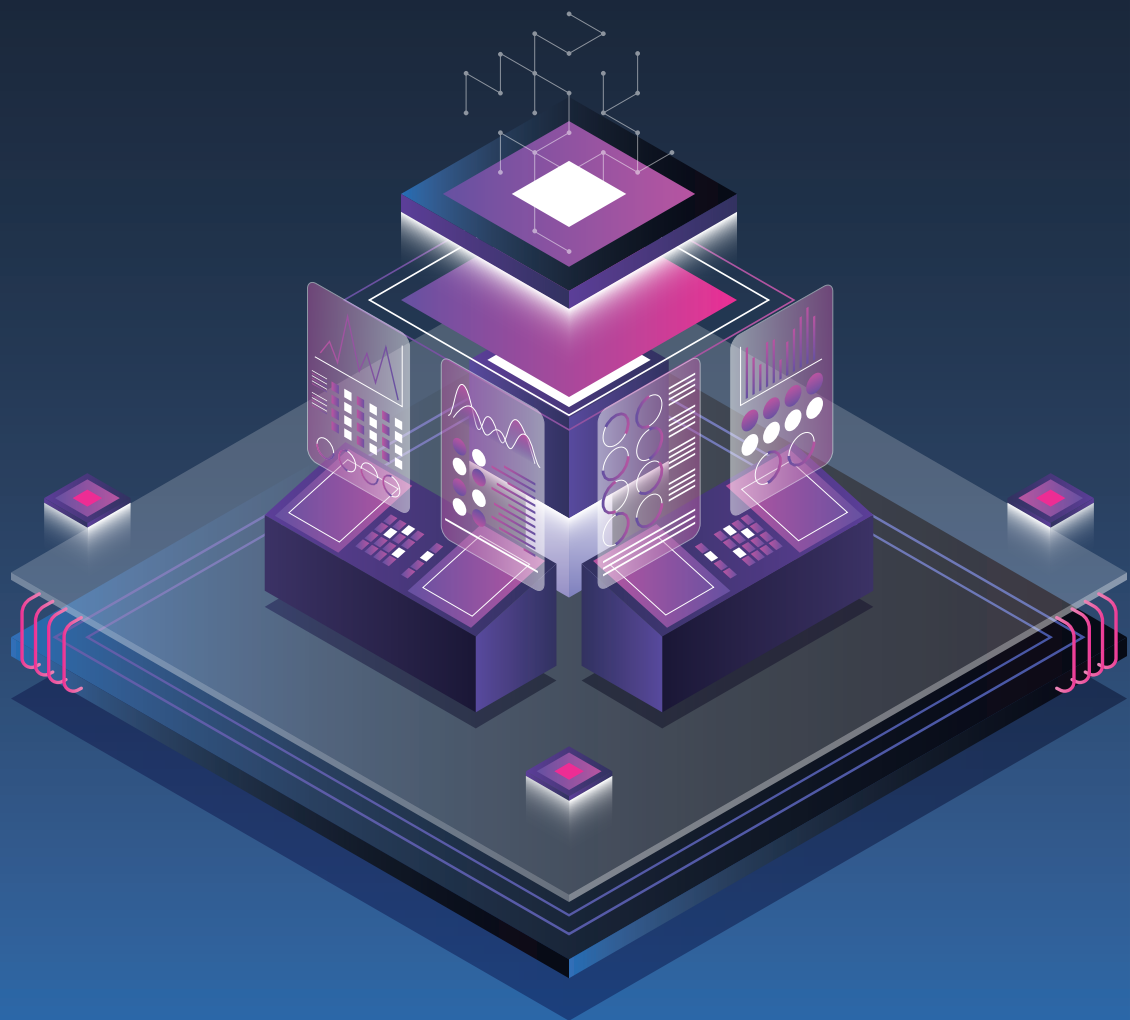


AI 모델의 전 생애주기를 통합 관리합니다

The MLOps Platform for Managing the Full AI Lifecycle

# NUBISON MLPlatform





# AI 모델 개발 이후 운영이 더 중요

개발된 AI 모델을 운영 환경에 안정적으로 배포하고 성능을 유지하는 것은 별도의 운영 역량을 필요로 합니다. MLPlatform은 개발과 운영 사이의 간극을 해소합니다.

01

## 코드 기반 운영의 한계

개인별 코드와 환경 차이로  
재현성과 추적성이 떨어집니다.

02

## 재학습 없는 모델 노화

데이터 변화로 성능이 저하되어도  
감지·재학습 체계가 부족합니다.

03

## 운영 환경 단절

데이터·설비·AI 시스템 간 연계가  
단발성으로 운영됩니다.

## MLPlatform으로 AI 모델 운영을 표준화합니다.

- 모든 모델·실험·배포 이력 통합 관리
- 개발부터 운영·재학습까지 단일 플랫폼 제공
- Data Lake·IoT·Agent와 표준 API 기반 연계

AI의 가치는 학습이 아니라 운영에서 완성됩니다.

# ML Platform

## 5가지 핵심 역량

NUBISON MLPlatform은 거버넌스, 라이프사이클, 운영 절차, 시스템 연동 사용 편의성의 다섯 가지 관점에서 AI 모델 운영 문제를 해결합니다.

### 거버넌스 GOVERNANCE

모든 모델, 실험, 배포 이력을 중앙에서 추적하고 관리합니다.

[버전 관리](#)[실험 추적](#)[감사 로그](#)[롤백](#)

### 전 생애주기 FULL LIFECYCLE

모델 개발부터 운영 배포까지 7단계 전 과정을 단일 플랫폼에서 수행합니다.

[초기 개발](#)[등록](#)[변환](#)[추론](#)[배포](#)

### 운영 절차 OPERATIONS

단계적 승인 · 테스트 · 배포의 표준화된 운영 프로세스를 제공합니다.

[검증](#)[승인](#)[Canary 배포](#)[모니터링](#)

### 시스템 연동 INTEGRATION

Datalake 및 Agent와 표준 API 기반으로 연동됩니다.

[NUBISON Datalake](#)[IoT](#)[Agent Orchestration](#)

### 비개발자 친화 ACCESSIBILITY

GUI 기반 환경을 통해 비개발자도 재학습과 배포 작업을 수행할 수 있습니다.

[No-Code 재학습](#)[GUI 워크플로우](#)[5분 재배포](#)

# 모든 모델·실험·배포 이력을 추적합니다

누가·언제·어떤 데이터로 모델을 개발하고 배포했는지  
모든 이력이 자동으로 기록됩니다.

## Model Versioning

### 모델 버전 관리

- 모델 형상 관리 (Asset Registry)
- 버전별 학습 데이터 · 파라미터 추적
- 이전 버전으로 신속하게 롤백 가능

## Experiment Tracking

### 실험 이력 추적

- 모든 실험의 자동 기록
- 메트릭 · 하이퍼파라미터 비교
- 실험 재현이 가능한 운영 환경 제공

## Access Control

### 권한 통제

- 역할 기반의 모델 및 데이터 접근 제어
- 환경별 (개발 · 스테이징 · 운영) 격리
- 민감 모델의 엄격한 통제

## Audit Trail

### 감사 로그

- 배포, 재학습, 접근 등 주요 행위 기록
- 규제 대응 · 내부 감사 즉시 활용
- 불변성 로그 저장

**거버넌스가 없는 AI 운영은 블랙박스가 될 수 있습니다.**

NUBISON MLPlatform은 모델의 생성, 학습, 배포 과정을  
추적 가능한 형태로 기록하여 규제 대응, 내부 감사, 문제 원인 분석을 지원합니다.

# 모델 생성부터 운영까지 하나의 플랫폼에서

AI 모델의 개발부터 테스트·배포·운영까지 전 과정을 7단계 표준 프로세스로 제공합니다.  
각 단계는 유기적으로 연결되어 효율적이고 일관된 AI 운영을 지원합니다.

## 7단계 표준 프로세스 | 모델 생성부터 배포까지 7단계가 한 플랫폼으로 운영 지원

|                                                                                                 |                                                                                                 |                                                                                                 |                                                                                                 |                                                                                                   |                                                                                                   |                                                                                                   |
|-------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|
| 1<br><br>초기 개발 | 2<br><br>코드 등록 | 3<br><br>코드 변환 | 4<br><br>지원·관리 | 5<br><br>추론 테스트 | 6<br><br>모델 시험 | 7<br><br>운영 배포 |
| Jupyter 기반<br>AI 모델<br>프로토타이핑                                                                   | AI 코드<br>및 모델<br>형상 등록                                                                          | 운영 가능한<br>추론 서비스로<br>자동 변환                                                                      | 모델·컨테이너<br>자원 및<br>버전 관리                                                                        | 추론 서버<br>배포 및<br>기능 검증                                                                            | 운영 환경<br>연계 및<br>성능 검증                                                                            | 검증 완료 후<br>운영 환경<br>자동 배포                                                                         |

## 4단계 표준 운영 절차 | 안정적인 운영 적응을 위한 4단계 배포 절차

### STAGE 01

#### 검증

- 단위 테스트
- 성능 기준 검증
- 데이터 호환성 검증
- 보안 스캔

### STAGE 02

#### 승인

- 변경 사유 기록
- 관리자 결재 워크플로
- 위험도 평가
- 거버넌스 체크리스트

### STAGE 03

#### 단계적 배포

- Canary 배포
- Blue-Green 전환
- 트래픽 단계적 전환
- 즉시 롤백 옵션

### STAGE 04

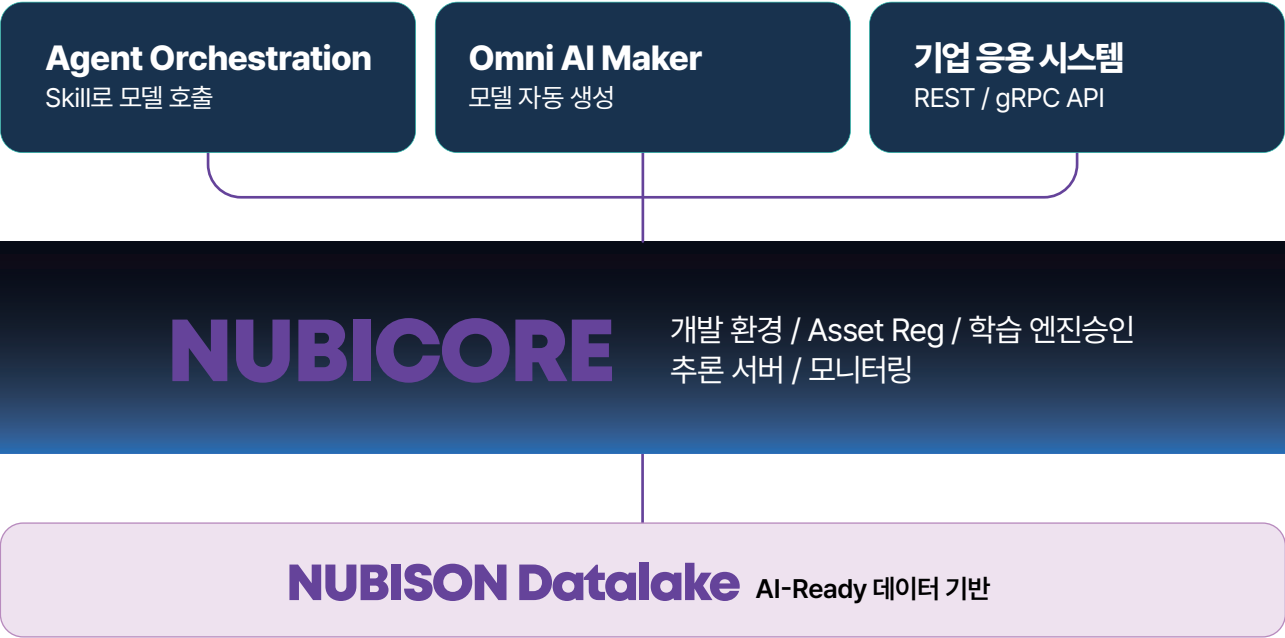
#### 모니터링

- 성능 메트릭 실시간
- Data Drift 감지
- 장애 자동 알림
- 재학습 트리거

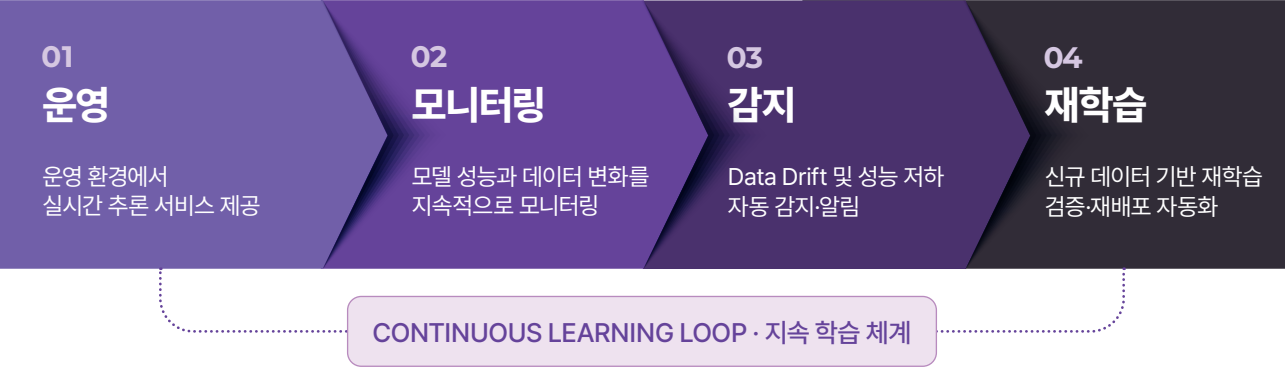
# 시스템 연동과 지속적 학습 루프

Datalake가 데이터를 공급하고, MLPlatform이 모델 라이프사이클을 책임지며 Agent가 모델을 소비하는 3층 구조입니다.

## 통합 데이터 기반 아키텍처 | 3층 구조 아키텍처



## 지속적 성능 개선 체계 | 최신 데이터를 기반으로 재학습하여 성능을 지속적으로 유지·개선합니다.



# ML 도구를 넘어 운영 중심 MLOps

모델 개발부터 운영까지 전 과정을 통합 관리하며  
No-Code 기반의 재학습과 표준화된 운영 체계를 제공합니다.

## 기존 제품과 차이점 | NUBISON MLPlatform의 4가지 핵심 차별점

| 구분     | 기존 ML 도구                     | NUBISON MLPlatform               |
|--------|------------------------------|----------------------------------|
| 사용자 편의 | 대부분 코드 기반으로 기능 구현            | 반복 재학습 및 운영 작업을 No-Code로 지원      |
| 인프라 관리 | CPU·GPU·NPU 자원 관리 기능이 제한적    | 기능별 CPU·GPU·NPU 자원 할당 및 관리 지원    |
| 기능 완결성 | 모델 개발·추론·배포 중심, 운영 기능 일부 미흡  | 등록·형상관리·테스트·추론·배포·인프라 관리 통합 제공   |
| 데이터 연계 | 데이터 연계 기능이 제한적이며 변경 시 재개발 필요 | Datalake와 유기적 연계, 소스 변경에 유연하게 대응 |

## 도입 효과 | 개발 생산성·운영 안정성·사용자 접근성·모델 신뢰성까지 4가지 핵심 영역에서 실질적인 변화를 제공

**70%↓**

**모델 배포 시간 단축**

등록부터 배포까지 자동화  
배포 시간 최대 70% 단축

**100%**

**행위 추적성**

모든 모델·실험·배포 이력 기록  
완전한 감사 추적 제공

**No-Code**

**재학습 자동화**

GUI 기반 재학습 워크플로우  
코딩 없이 재학습·배포 수행

**Auto**

**Drift 대응 자동화**

Data Drift 자동 감지  
재학습·재배포 자동 수행

**AI 도입 가속**

PoC부터 운영  
전환까지의 기간 단축

**운영 안정성**

지속적 재학습으로  
모델 성능 유지

**조직 확산**

현업 주도의  
AI 운영 환경 구현



**SimPlatform**  
www.simplatform.com

**(주)심플랫폼**

(본사) 서울시 금천구 가산디지털1로 226 에이스하이엔드타워5차 2003호

(AX본부) 서울시 강남구 삼성로 555 알앤텍빌딩 4층

Tel. **02-6405-4338** | email **ax@simplatform.com**