PlantPulse® **Platform**

INDUSTRIAL DATA INFRASTRUCTURE

제품 소개 및 도입 제안서

흩어진 공장 데이터를, 하나의 인프라로.

수집 · 메시징 · 저장 · 처리 · 분석 · 시각화까지 데이터 전 주기를 **하나의 설치 패키지**로 제공하는 엔터프라이즈 IIoT 데이터 플랫폼입니다. SCADA · Historian · 게이트웨이 · DB 로 나뉘어 있던 인프라를 대체합니다.



40,000/s
메시지 수집

500,000/s
실시간 이벤트 평가

24+
사전 통합 요소

2026년 9월

엔터프라이즈 산업 데이터 인프라 · 온프레미스 / 폐쇄망

kopens®

CONTENTS

목차

01	왜 지금인가 — 인프라의 문제	3
02	PlantPulse Platform 이란	4
03	7개 계층, 하나의 패키지	5
04	특장점 ①②③	6–8
05	인더스트리얼 데이터레이크 · **자체 엔진 둘**	9–15
06	특장점 ④⑤⑥	16–18
07	특장점 ⑦⑧	19–20
08	아키텍처 — 데이터 흐름	21
09	사용 흐름 (실제 화면)	22–23
10	보안 · 운영	24
11	설치 · 납품 모델	25
12	레퍼런스	26
13	도입 절차 및 일정	27

한 장 요약

> 공장 데이터 인프라는 대개 **제품 다섯 개를 이어 붙인 결과물**입니다. PlantPulse Platform 은 그 자리를 **설치 한 번**으로 바꿉니다.

40,000/s	500,000/s	24+
메시지 수집 처리량	실시간 이벤트 평가(CEP)	사전 통합 마이크로서비스

핵심 차이
도구 모음이 아니라 하나의 제품입니다. 수집기 · 브로커 · 시계열 DB · 룰 엔진 · 대시보드를 각각 도입해 붙이는 대신, 7개 계층이 **사전 통합된 상태**로 설치됩니다. 온톨로지 · Unified Namespace · Flow · 복합 이벤트 처리가 같은 데이터 기반 위에서 함께 동작합니다.

이 문서가 다루는 것

무엇을 대체하는가

3–5쪽. 흩어진 인프라의 문제, 제품 정의, 계층 구성.

무엇이 다른가

6–20쪽. 성능, 표준화, **인더스트리얼 데이터레이크(9–15쪽)** · **자체 엔진 둘(10쪽)**, ISO 지표, 온톨로지, UNS, Flow, MCP.

실제로 어떻게 쓰는가

21–23쪽. 데이터 흐름과 구동 중인 시스템에서 캡처한 화면.

도입하면 어떻게 되는가

24–27쪽. 보안·운영, 설치, 레퍼런스, 도입 일정.

01 · 배경

왜 지금인가 — 인프라의 문제

공장에 데이터가 없어서 문제인 경우는 드뭅니다. 문제는 그 데이터가 **다섯 군데에 다른 모양으로** 들어 있다는 것입니다.

문제 1

제품이 다섯, 계약이 다섯

SCADA · Historian · 게이트웨이 · DB · BI 를 각각 도입하면 벤더도 다섯, 라이선스도 다섯, 장애 시 책임 소재도 다섯입니다.

문제 2

이어 붙이는 일이 프로젝트가 된다

제품 사이를 잇는 연동이 매번 개발 항목이 됩니다. 하나를 바꾸면 나머지 넷의 연동을 다시 손봐야 합니다.

문제 3

같은 설비가 사이트마다 다른 모양

매핑을 사람이 하니 라인마다·공장마다 구조가 달라집니다. 비교도, 집계도, AI 학습도 그 앞에서 멈춥니다.

문제 4

지표를 만드는 것부터 일이다

OEE 하나를 보려 해도 정의부터 계산 로직까지 새로 만듭니다. 사이트마다 정의가 달라 합산이 안 됩니다.

그래서 필요한 것

> 설치하면 **바로 도는** 하나의 데이터 기반. 그 위에서 표준화 · 지표 · 자동화 · AI 가 **같은 데이터를 보고** 움직이는 구조.

PlantPulse Platform 은 그 기반입니다. 24개 이상의 구성요소가 사전 통합되어 하나의 패키지로 설치되고, 설치 직후부터 수집 → 저장 → 처리 → 분석 → 시각화 전 파이프라인이 가동됩니다.

02 · 소개

PlantPulse Platform 이란

설치 즉시 가동되는 산업 데이터 인프라입니다. 엣지·설비에서 올라온 데이터를 받아 저장·표준화·분석하고, 현장과 AI 가 같은 맥락으로 쓸 수 있게 만드는 것이 이 제품의 일입니다.

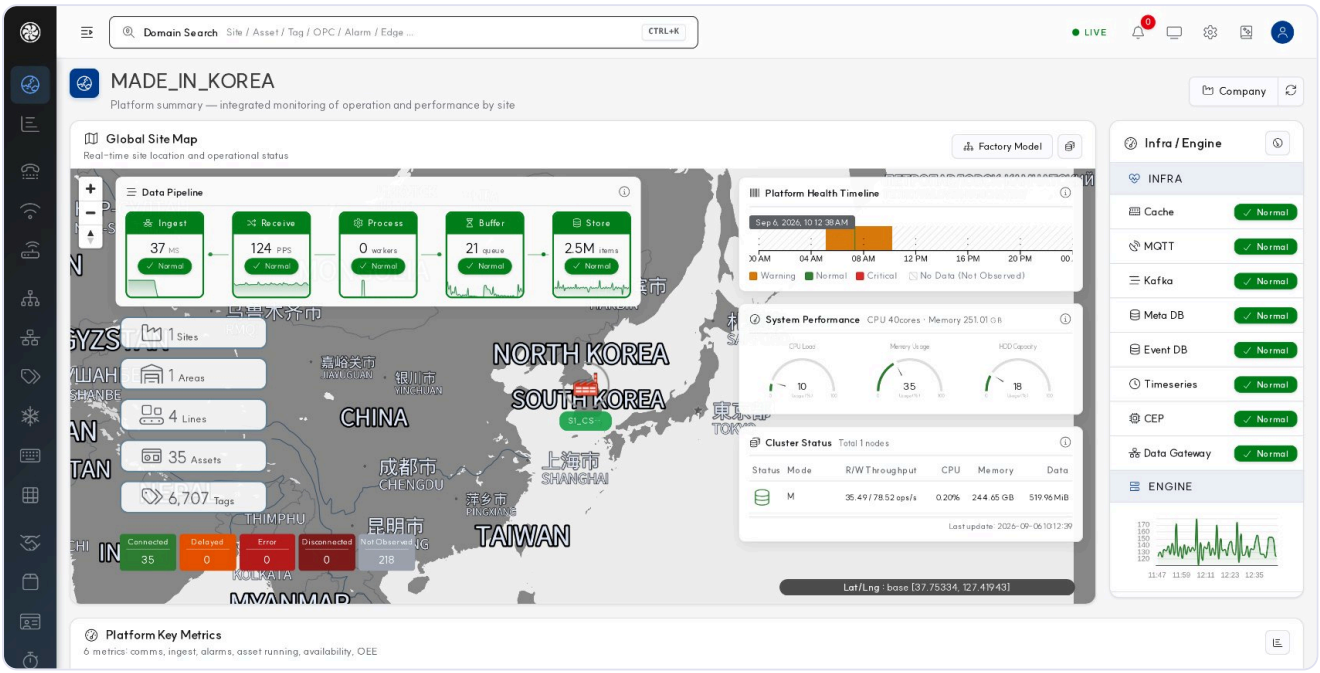


그림 1. 통합 모니터링 화면 — 사이트 지도, 데이터 파이프라인 각 단계의 처리량과 상태, 시스템 성능, 클러스터 상태, 인프라 구성요소 건강도, 주요 지표를 한 화면에서 봅니다. *실제 구동 인스턴스 캡처입니다.*

받는다
수집 · 메시징
엣지·설비·외부 시스템의 데이터를 받아 큐에 태웁니다.

담는다
저장 · 표준화
분산 시계열·이벤트 저장소에 담고 설비 표준 모델로 정리합니다.

쓴다
분석 · 자동화 · 개방
지표를 만들고 규칙으로 감지하고 AI 와 외부 시스템에 열어줍니다.

무엇을 대체하나

기존에 따로 도입하던 것	플랫폼이 담고 있는 것
SCADA · HMI 서버	실시간 모니터링 화면과 캔버스 대시보드
Historian	분산 시계열·이벤트 저장소 (수평 확장)
메시지 브로커	MQTT · 메시징 브로커 · Sparkplug B 내장
룰 엔진 · 알람 서버	복합 이벤트 처리(CEP)와 고급 알람
ETL · 연동 개발	노코드 Flow 자동화
BI · 리포팅	ISO 표준 지표 자동 생성과 분석 화면

03 · 구성

7개 계층, 하나의 패키지

아래 계층은 각각 따로 도입하는 제품이 아니라 **한 번에 설치되는 구성요소**입니다. 설치 후 몇 분이면 전 파이프라인이 가동됩니다.



그림 2. 7개 계층. 아래에서 위로 데이터가 흐르고, 각 계층은 따로 고를 수 있는 선택지가 아니라 **같이 설치되고 같이 검증되는 한 벌**입니다.

> **사전 통합의 의미.** 각 계층을 따로 도입하면 버전 조합·연동·성능 튜닝이 매번 프로젝트가 됩니다. 사전 통합은 그 작업을 제품 측이 이미 끝내 두었다는 뜻이며, 설치 후 몇 분 안에 데이터가 실제로 흐르는 상태가 됩니다.

설치 직후부터 도는 것

데이터가 흐릅니다

엣지를 붙이면 수집 → 버퍼 → 저장 → 조회가 즉시 동작합니다. 저장소 튜닝이나 스키마 설계를 먼저 하지 않아도 됩니다.

화면이 있습니다

통합 모니터링·자산 탐색·알람·클러스터 콘솔이 기본 제공됩니다. 첫 대시보드를 만드느라 며칠 쓰지 않습니다.

지표가 나옵니다

자산 모델을 구성하는 즉시 OEE·RAM·EMS 가 표준 정의로 계산됩니다.

열려 있습니다

UNS 토픽과 AI 도구가 처음부터 켜져 있어, 외부 연동을 나중 단계로 미루지 않아도 됩니다.

남는 일은 「우리 공장에 맞추는 것」뿐입니다 — 자산 계층 구성, 설비 유형별 규칙, 알람 기준, 자동화 흐름. 인프라를 세우는 일이 아니라 **업무를 반영하는 일**부터 시작합니다.

04 · 특장점

① 들어오는 양을 감당합니다

설비를 늘릴수록 데이터는 선형이 아니라 계단식으로 늘어납니다. PlantPulse Platform 은 **수집과 실시간 판정 모두**에서 산업 현장 규모를 전제로 설계됐습니다.

40,000/s	500,000/s	선형 확장
서버 1대 기준 메시지 수집	인메모리 CEP 이벤트 평가	노드 추가 = 용량 증가

실시간 판정 — 복합 이벤트 처리(CEP)

단일 태그의 임계 초과만 잡는 것이 아니라, **시간 윈도우 안의 패턴**을 봅니다. "5분 안에 3회 이상 진동 급등이 있고 그 뒤 온도가 상승하면" 같은 조건을 전용 질의 언어로 정의해 스트림에서 즉시 판정합니다.

- **시간 윈도우 기반 규칙** — 순간 값이 아니라 흐름으로 판단합니다.
- **자산 도메인 룰** — 설비 유형별 표준 판정 규칙을 재사용합니다.
- **집계·스냅샷·예측까지 롤업** — 원시 측정값이 곧바로 의미 있는 단위로 정리됩니다.

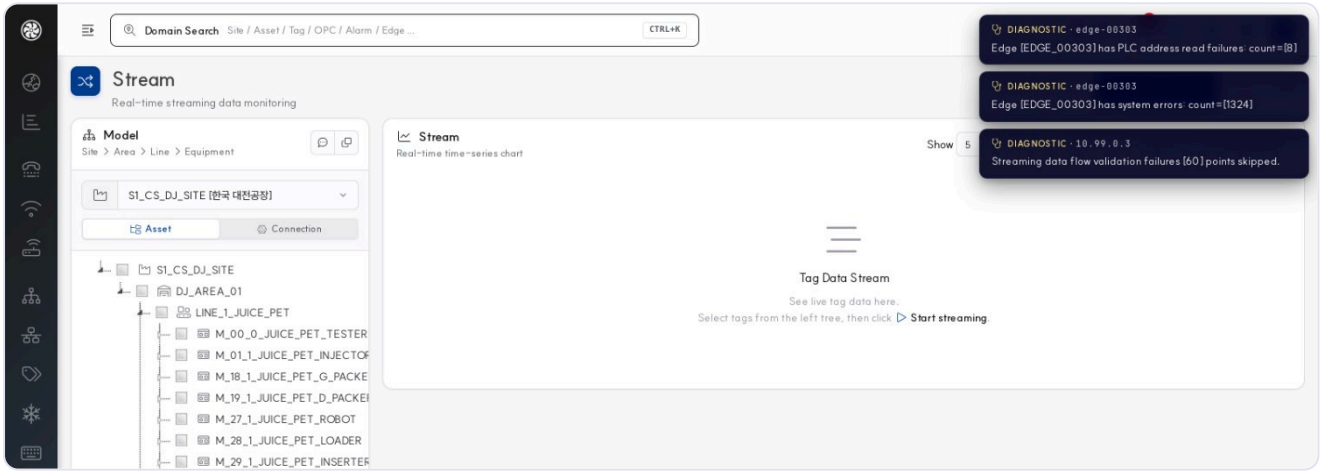


그림 3. 실시간 스트리밍 모니터링 — 자산 계층을 따라 흐르는 데이터를 그 자리에서 확인합니다.

왜 중요한가
수집만 빠르면 저장소가 밀리고, 저장만 크면 판정이 늦습니다. **수집·저장·판정이 같은 제품 안에서 함께 설계**되어야 병목이 한쪽으로 몰리지 않습니다.

04 · 특장점

② 늘리는 방법이 노드를 꽂는 것입니다

산업 데이터는 줄지 않습니다. 그래서 저장소는 **늘릴 수 있어야** 하고, 늘리는 일이 이관 프로젝트가 되어서는 안 됩니다.

분산 데이터레이크

글로벌 하이퍼스케일러가 쓰는 것과 같은 계열의 **분산 이벤트·시계열 저장소**를 산업 데이터에 맞게 내장했습니다. 데이터가 노드에 자동 분산되고, 복제 기반이라 노드 하나가 죽어도 데이터가 유실되지 않습니다.

- **노드 추가만으로 선형 확장** — 저장 용량과 처리량이 함께 올라갑니다.
- **시계열 · 이벤트 · BLOB 을 한 클러스터에** — 도면·이미지까지 같은 곳에 담습니다.
- **상태를 눈으로 봅니다** — 링 토폴로지, 노드 부하, 저장 분포를 콘솔에서 확인합니다.

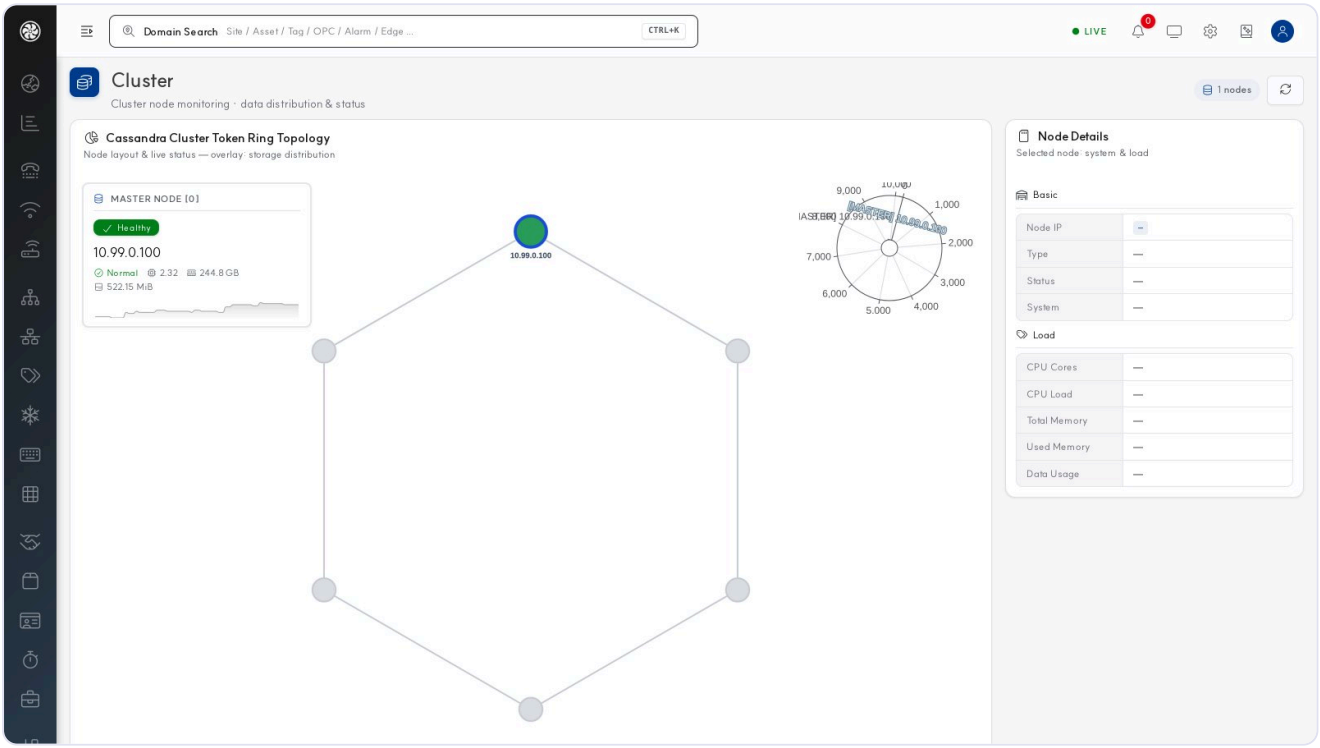


그림 4. 클러스터 콘솔 — 노드별 상태·처리량·자원·데이터 분포와 링 토폴로지를 실시간으로 확인합니다.

> **운영자가 감당할 수 있는 확장.** 분산 저장소는 대개 전문 인력이 필요합니다. 이 제품은 그 저장소를 제품 안에 넣고 **콘솔에서 보이도록** 만들어, 별도 전담 조직 없이 운영할 수 있게 했습니다. 구조·계층·운영은 **9–15쪽 «인더스트리얼 데이터레이크»**에서 자세히 다룹니다.

04 · 특장점

③ 설비 100대 추가가 매핑 100번이 아닙니다

데이터 프로젝트가 늘어지는 진짜 이유는 **매핑**입니다. 설비마다 사람이 구조를 정하면, 설비가 늘어난 만큼 일이 늘어납니다.

여섯 가지 데이터 클래스

설비가 쏟아내는 데이터를 여섯 종류로 나누고, **설비 유형별로 규칙을 한 번** 정의합니다. 같은 유형의 새 설비는 그 규칙을 그대로 재사용합니다.

스냅샷 현재 값	**알람** 이상 발생	**이벤트** 상태 변화
맥락 생산 지시·조건	**집계** 기간 요약	**명령** 제어 이력

- **코딩·스키마 설계 없이** 화면 설정만으로 매핑합니다.
- **표준 자산 계층**(사이트 → 구역 → 라인 → 설비 → 태그)에 같은 구조로 저장됩니다.
- 같은 유형이면 **라인·사이트가 달라도 구조가 같아** 즉시 비교·집계됩니다.
- 구조가 같으므로 **AI 학습에 바로 쓸 수 있는** 형태가 됩니다.

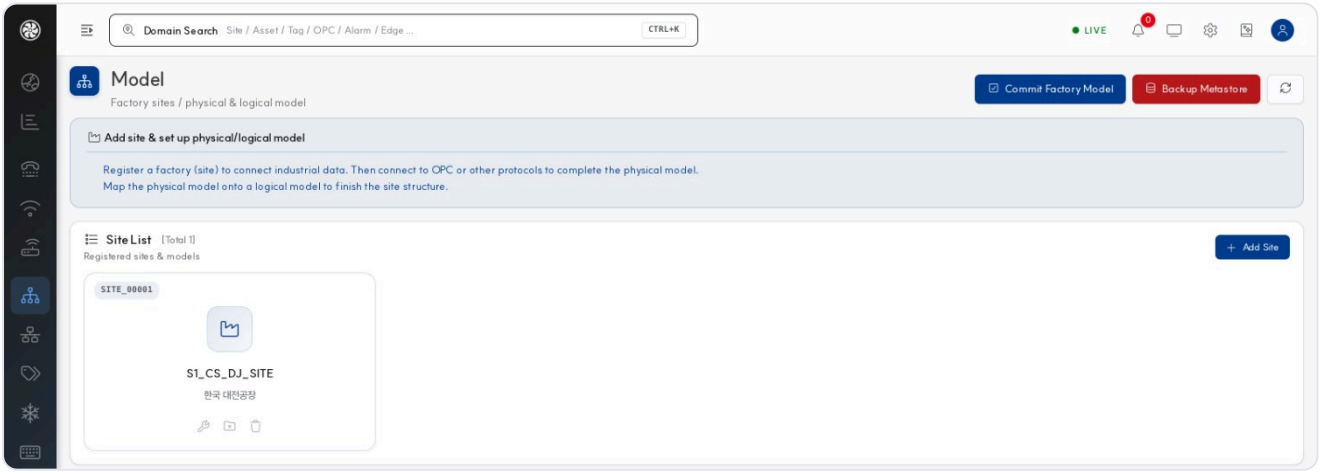


그림 5. 공장 모델 — 사이트의 물리·논리 구조를 화면에서 정의하고 변경 사항을 커밋합니다.

05 · 데이터레이크

인더스트리얼 데이터레이크 — 한 이미지에 담긴 16개 컴포넌트

플랫폼의 바닥은 **PlantPulse DataLake** 입니다. 저장 · 메시징 · 시계열 · 실시간 처리 · 분석 · 워크플로 · 조회 게이트웨이 · 관리 콘솔이 **하나의 컨테이너 이미지**로 나오고, `pd start` 한 번에 정해진 순서로 올라옵니다. 산업 데이터를 위해 조립된 «데이터 인프라 한 벌»입니다.



그림 6. 데이터레이크 관리 콘솔 첫 화면 — «지금 문제가 있는가»에 한 줄로 답합니다. 컴포넌트 16개 모두 정상 · 서비스 19/19 · 알람 없음, 아래는 컴포넌트별 24시간 타임라인. *실제 구동 인스턴스 캡처입니다.*

안에 무엇이 있나

계층	구성요소	역할
저장	Aster TSDB (Cassandra 6.0 기반 자체 빌드) · PostgreSQL 18 · Valkey · RustFS	시계열·이벤트 링 / 마스터·메타 / 인메모리 캐시·스트림 / S3 오브젝트
메시징	Apache Kafka 4.3 (KRaft) · **Aster MQTT**(자체 브로커)	수집 버퍼 · 발행/구독 · Sparkplug B · 브로커 인증·ACL
시계열	PlantPulse TSE (시계열 엔진) · Grafana	롤업 · 30여 종 집계 함수 · TQL 질의 · 대시보드
실시간 처리	Aster CEP (EQL) — 자체 엔진	시간 윈도우 규칙 · 패턴 · 고급 알람
분석	Apache Spark 4.1 · Kyuubi · Hive · Hadoop · Gravitino	Iceberg 레이크하우스 · SQL 분석 · JDBC 서빙 · 카탈로그
워크플로	Temporal · Kestra · Warehouse 워커	일일 아카이브 · 주간 최적화 · 선언형 파이프라인
개방	Data Gateway · SQL 콘솔	PQL · CQL · SQL · TQL 을 단일 REST 로
운영	관리 콘솔 · `pd` CLI · 백업 툴킷	상태 · 토폴로지 · 스토리지 · 백업 · 업그레이드

왜 «하나의 이미지»인가. 위 구성요소를 따로 설치하면 버전 조합·설정·기동 순서가 사이트마다 달라집니다. 데이터레이크는 조합이 검증된 상태로 이미지에 구워지고, 이미지에는 **캘린더 버전과 빌드 커밋**이 박히며, 배포 전에 **서명과 체크섬**이 검증됩니다. 같은 이미지는 어느 사이트에서도 같은 데이터레이크입니다.

05 · 데이터레이크

자체 엔진 셋 — Aster TSDB · Aster CEP · Aster MQTT

데이터레이크의 심장부는 **쌓는 곳 · 판정하는 곳 · 들어오는 문**입니다. 이 세 자리는 남의 제품을 가져다 붙인 것이 아니라 **KOPENS 가 소스째 개발하고 유지하는 엔진**입니다. 그래서 현장에서 필요해진 기능이 다음 판에 들어가고, 결함은 벤더의 로드맵을 기다리지 않고 고칩니다.

3	7.1×	0
자체 엔진 — 저장 · 판정 · 수집	시계열 저장 축소(실측)	별도 스트리밍 클러스터

저장

Aster TSDB — 산업 시계열 데이터베이스

Apache Cassandra 6.0 기반의 자체 빌드입니다. **CQL 문법과 디스크 포맷은 그대로**라 쓰던 데이터를 그대로 읽고, 시계열 기능은 테이블마다 켜서 씁니다.

- **저장 7.1× 축소** — 11.9 → 1.7 B/row (운영 스키마 2천만 행 실측)
- **서버측 CQL 시계열 함수 21종** — 버킷 · 갭필 · 회귀 · 시간가중평균
- **계층형 저장** — 오래된 구간을 자동 압축하고 `SELECT` 는 그대로. 검증은 릴리즈 게이트 단언 93개 · 3노드 49개

판정

Aster CEP — 복합 이벤트 처리 엔진

규칙은 **EQL**(SQL-92 코어 + 스트리밍 확장)로 쓰고, 판정은 CEP 서버 **안에서** 이루어집니다 — 스트리밍 클러스터를 따로 세우고 운영할 필요가 없습니다.

- **SQL:2016 행 패턴 인식** — 「구간 사이의 «모든» 행」을 요구하는 설비 판정을 위해 엔진 안에 직접 구현
- **답을 대조해 맞춥니다** — 같은 규칙을 Apache Flink 2.3.0 에 넣어 결과를 맞대 봅니다
- **골든 규칙 152건**을 매 빌드에 대조하고, **못 돌리는 규칙은 이름을 대고 거부**합니다 — 조용히 틀린 답을 내지 않습니다

수집

Aster MQTT — 설비가 들어오는 문

공장의 센서·PLC 가 붙는 MQTT 브로커입니다. 표준 MQTT 를 그대로 쓰면서, 운영에서 문제가 되는 자리를 우리가 고쳤습니다.

- 🔴 **인증 장치가 없으면 기동을 거부합니다** — 보안 확장이 안 뜨면 브로커가 «열린 채로 정상처럼» 보이던 것을 막았습니다
- **버려진 메시지의 «이유» 를 셉니다** — 용량 · 백프레셔 · 정책을 구분해 일곱 가지로. 숫자 하나로는 왜 버려졌는지 알 수 없습니다
- **공개 취약점 15건**을 닫은 네트워크 라이브러리로 올려 두었습니다

왜 «자체 엔진»이 중요한가

이럴 때	가져다 쓴 제품이라면	자체 엔진이면
현장이 없는 기능을 요구할 때	기능 요청 → 상류 로드맵 대기	문법과 실행기를 직접 고쳐 다음 판에 넣습니다
결함이 나올 때	벤더 릴리즈까지 우회로 버팁니다	원인을 소스에서 찾아 고치고 시험으로 못 박습니다
폐쇄망에 넣을 때	라이선스 · 공급 경로를 따로 확인	Apache-2.0 기반이고 공급 경로가 우리 손에 있습니다

가져다 쓸 것은 가져다 씁니다. Kafka · Spark · Iceberg 처럼 표준이 된 것은 그대로 씁니다. 직접 만든 것은 **산업 시계열 · 실시간 판정 · 설비 수집** 세 자리뿐입니다.

05 · 데이터레이크

핫 · 웜 · 콜드 — 데이터가 나이를 먹는 길

산업 데이터는 시간이 지나며 쓰임이 바뀝니다. 방금 온 값은 화면과 규칙이, 지난달 값은 리포트가, 지난해 값은 분석과 AI 학습이 씁니다. 데이터레이크는 세 층을 두고 데이터를 **자동으로** 옮깁니다 — 사람이 옮기지 않습니다.



웜 계층 — 테이블마다 다른 보존 기간 (기본값)

데이터	TTL	왜 이 길이인가
실시간 설비 데이터 (자산 단위 스냅샷)	10일	화면과 규칙이 쓰는 값 — 짧게 두고 빠르게
태그 포인트 (원시 측정값)	62일	두 달 치 원본을 즉시 조회 가능하게
집계 · 샘플링 · 스냅샷	93일	분기 리포트와 추세 비교
아카이브 테이블	1년	콜드로 넘어간 뒤에도 링에서 바로 읽는 완충
시스템 로그 · 모니터링 · 배치/감사 이력	1 ~ 31일	운영 진단용 — 오래 둘 이유가 없는 것

콜드 계층 — 자동으로 옮기고, 자동으로 다듬는다

- **매일 02:20** Temporal 워크플로가 Spark 잡을 띄워 Aster TSDB 의 **8개 테이블**을 Iceberg 로 넘깁니다. 어느 테이블을 넘기는지는 플러그인(SPI)으로 선언됩니다.
- **매주 일요일 03:00** 최적화 — 작은 파일 병합(128 MB 목표) · 오래된 스냅샷 만료 · 고아 파일 제거. 콜드 계층이 스스로 가벼워집니다.
- 파티션은 **사이트 / 연 / 월 / 일** — 사이트별 격리와 기간 조회가 파일 단위에서 바로 잘립니다. 파일 256 MB · zstd 압축.

05 · 데이터레이크

토큰 링 — 노드를 꽂으면 용량이 늘어납니다

웜 계층의 저장소는 글로벌 하이퍼스케일러가 쓰는 것과 같은 계열의 **토큰 링 분산 데이터베이스**입니다. 데이터가 노드에 자동 분산되고 복제되어, **노드를 더하면 용량과 처리량이 함께 오르고** 노드 하나가 죽어도 데이터가 남습니다.



그림 7. 토폴로지 화면 — 마스터 1 + 워커 1, 빈 슬롯 4. 노드 상세에 링 상태 · DC/랙 · 부하(8.37 GiB) · 토큰 · 키스페이스 유효 소유율(48.8%)이 보입니다. *실제 구동 인스턴스 캡처입니다.*

늘리는 절차 — 명령 셋

1. **워커 컨테이너를 띄운다.** 같은 이미지, 모드만 WORKER. 저장 · 분석 · 관리 API 만 오릅니다.
2. `pd node status` — 새 노드가 링에 `UN` (Up · Normal)으로 보이는지 확인합니다.
3. `pd node add` — 살아 있는 노드 수에 맞춰 **복제 계수를 자동으로 올리고**(2대 → 2, 3대 이상 → 3) 리페어까지 돌립니다. 데이터센터 이름이나 노드 수를 읽지 못하면 **추측하지 않고 멈춥니다** — 잘못된 복제 설정은 조용히 데이터를 잃기 때문입니다.

무엇이 복제되나

구성요소	방식	장애 시
Aster TSDB 링	토큰 링 자동 분산 · 복제 계수 최대 3 · 데이터센터 인지 배치	노드 1대 손실에도 읽기·쓰기 지속
PostgreSQL	마스터 → 워커 스트리밍 복제 · 복제 슬롯으로 WAL 보전	워커가 최신 사본을 보유
Valkey	프라이머리 → 레플리카	캐시 사본 즉시 승격 가능
헬스 에이전트	워커마다 자기 상태를 마스터에 보고	콘솔 «워커» 화면에 뒤처짐(랙)이 수치로

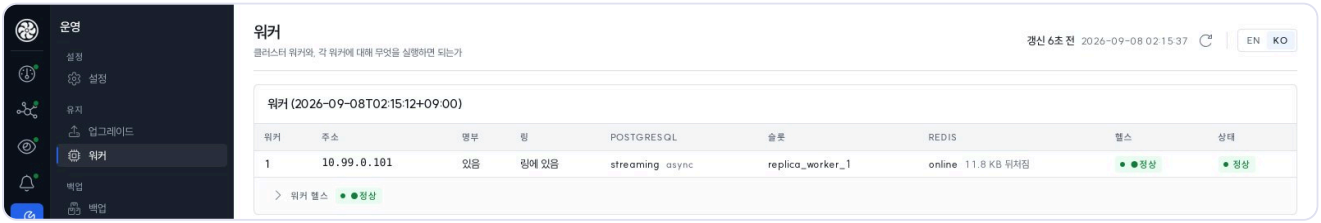


그림 8. 워커 화면 — 명부 · 링 합류 · PostgreSQL 스트리밍 · 복제 슬롯 · Valkey 온라인(11.8 KB 뒤처짐) · 헬스를 한 줄로 판정합니다.

> **마스터 1 + 워커 최대 5.** 대부분의 공장은 마스터 1대로 시작해 태그와 보존 기간이 늘 때 워커를 붙입니다. 이관 프로젝트 없이, 서비스를 세우지 않고 늘리는 것이 이 구조의 목적입니다.

05 · 데이터레이크

레이크하우스 — 오픈 포맷, 표준 SQL, 잠기지 않는 데이터

데이터레이크의 값은 «담을 수 있는가»가 아니라 **«꺼내 쓸 수 있는가»**에서 결정됩니다. 콜드 계층은 **Apache Iceberg v2 + Parquet** — 특정 벤더가 아니라 어떤 분석 엔진도 읽는 열린 포맷이고, 모든 계층이 **표준 JDBC** 로 열려 있습니다.

ICEBERG 가 주는 것

- **ACID 트랜잭션** — 아카이브 도중 읽어도 반쯤 쓰인 데이터가 보이지 않습니다.
- **스키마 진화** — 태그·컬럼이 늘어도 지난 파일을 다시 쓰지 않습니다.
- **타임 트래블** — «지난주 이 시점의 테이블»을 그대로 다시 봅니다.
- **히든 파티셔닝** — 질의에 파티션 컬럼을 몰라도 파일이 잘립니다.

위에 얹힌 엔진

- **Spark 4.1** — 아카이브·최적화·대용량 분석 실행.
- **Kyuubi** — 다중 사용자 SQL 서빙 · JDBC 엔드포인트.
- **Gravitino · Hive 메타스토어** — Iceberg REST 카탈로그 · 테이블 메타.
- **RustFS** — S3 호환 오브젝트 스토리지, 온프레미스에서.

어떻게 묻나 — 네 언어, 한 게이트웨이

언어	대상	접근
PQL	PostgreSQL — 자산 모델 · 설정 · 마스터	표준 JDBC
CQL	Aster TSDB — 원 시계열 · 이벤트 · 알람	자체 **JDBC 드라이버**
SQL	Spark / Iceberg — 콜드 계층 · 대용량 분석	Kyuubi Thrift · 자체 **Spark SQL JDBC 드라이버**
TQL	시계열 엔진 — 롤업 · 30여 집계 함수 · 시간 윈도우	`jdbc:ts://` — SQL 을 아는 사람이 바로 씀

네 언어 모두 **Data Gateway** 한 곳(`POST /api/v1/query`)으로도 갑니다 — 대용량은 `/query/stream` 이 NDJSON 으로 흘려보내고, `/cache` 는 인메모리 현재값을 돌려줍니다. 클라이언트 SDK 는 Java 로 제공됩니다.



그림 9. 콘솔 «도구» — CEP · Grafana · 시계열 엔진 · Spark · Temporal · Kestra · RustFS · Data Gateway 가 한 노드 안에서 각자 화면을 갖고, 상태가 함께 표시됩니다.

열되, 게이트웨이가 지킨다
DDL(DROP · TRUNCATE · ALTER …) 차단 · 다중 구문 차단 · 결과 **100,000 행** 상한 · IP 당 **1,000 req/s** 제한 · 요청 크기 상한 · 모든 질의를 **쿼리 로그**에 기록. 분석가에게 열어 주되 운영 저장소가 다치지 않게 하는 장치입니다.

05 · 데이터레이크

운영이 보이는 데이터레이크 — 콘솔과 `pd`

분산 저장소는 보통 전담 인력을 요구합니다. 이 제품은 그 저장소를 제품 안에 넣고 **«지금 문제가 있는가»를 첫 화면에서 답하도록** 만들었습니다 — 16개 컴포넌트를 각각의 임계값으로 재고, 24시간을 띠로 보여줍니다.

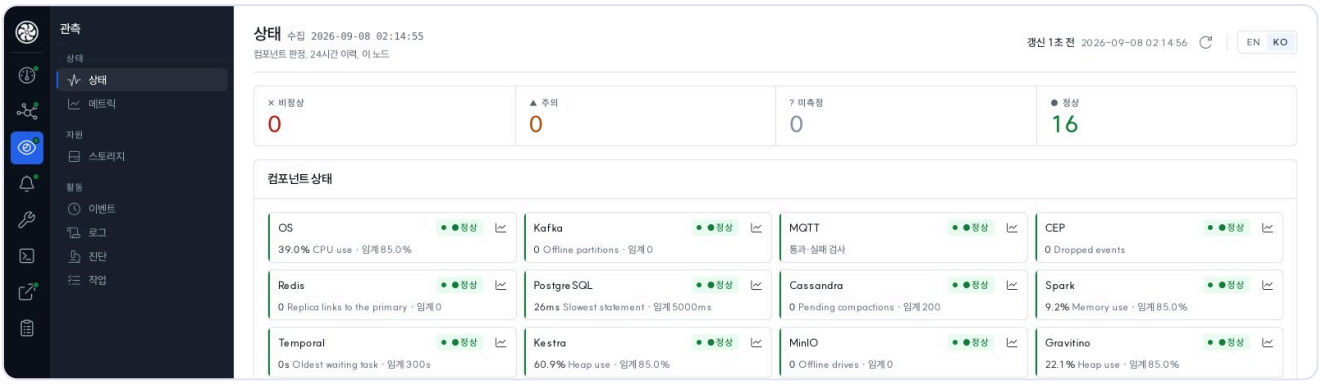


그림 10. 상태 화면 — 컴포넌트마다 «무엇을 쟀고 임계가 얼마인지»가 같이 적힙니다 (Kafka 오프라인 파티션 0, PostgreSQL 최장 문장 26 ms, Aster TSDB 대기 컴팩션 0, 시계열 엔진 인제스트 큐 626 …).

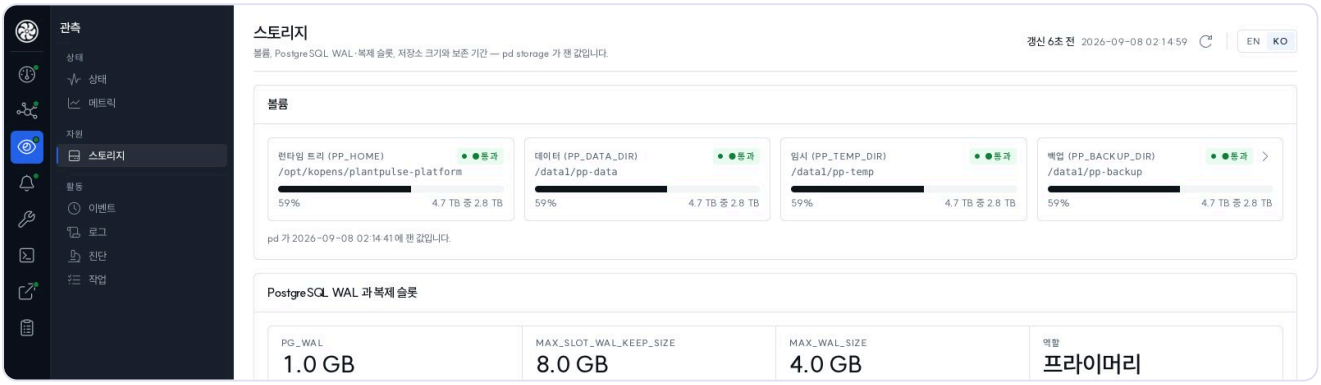


그림 11. 스토리지 화면 — 볼륨 4종의 사용률(4.7 TB 중 2.8 TB), PostgreSQL WAL 과 복제 슬롯, Aster TSDB · Kafka 디스크와 보존 기간. 판정은 `pd storage` 가 잰 값을 그대로 씁니다 — 화면이 임계값을 따로 만들지 않습니다.

콘솔 열두 화면 · 다섯 묶음

묶음	화면
개요	대시보드 — 한 줄 판정 · 활성 알람 · 컴포넌트 헬스 · 타임라인
관측	상태 · 메트릭(1시간 ~ 7일) · 스토리지 · 이벤트 · 로그 · 진단(`pd doctor`) · 작업
운영	설정(템플릿 렌더 · 차이) · 업그레이드 · 워커 · 백업
도구 · 기록	다른 웹 화면 바로가기 8종 · 감사 로그

`pd` — 데이터레이크의 단일 진입점

`pd start | stop | restart | status` · `pd config render | diff` · `pd secret rotate` · `pd node add | status | repair` · `pd backup run | restore | schedule` · `pd doctor` · `pd storage` · `pd logs` . 흩어져 있던 기동·설정·복구 스크립트 수십 개가 동사 하나로 모였고, 모든 명령은 `--json` 으로 같은 값을 냅니다 — 콘솔이 보는 것과 터미널이 보는 것이 같습니다.

> **«미측정»은 정상이 아닙니다.** 못 잰 값은 회색으로 남기고 0 이나 초록으로 채우지 않습니다. 초록 점은 실제로 쟀다는 뜻일 때만 켜집니다 — 운영자가 화면을 믿을 수 있게 하는 규칙입니다.

05 · 데이터레이크

지키는 장치 — 백업 · 복구 · 업그레이드

데이터레이크는 공장의 기억입니다. 그래서 **백업은 기본 켜져 있고**, 복원은 버튼이며, 업그레이드는 이미지 교체 한 번입니다. 세 가지 모두 콘솔에서 보이고 `pd` 로 되풀이할 수 있습니다.

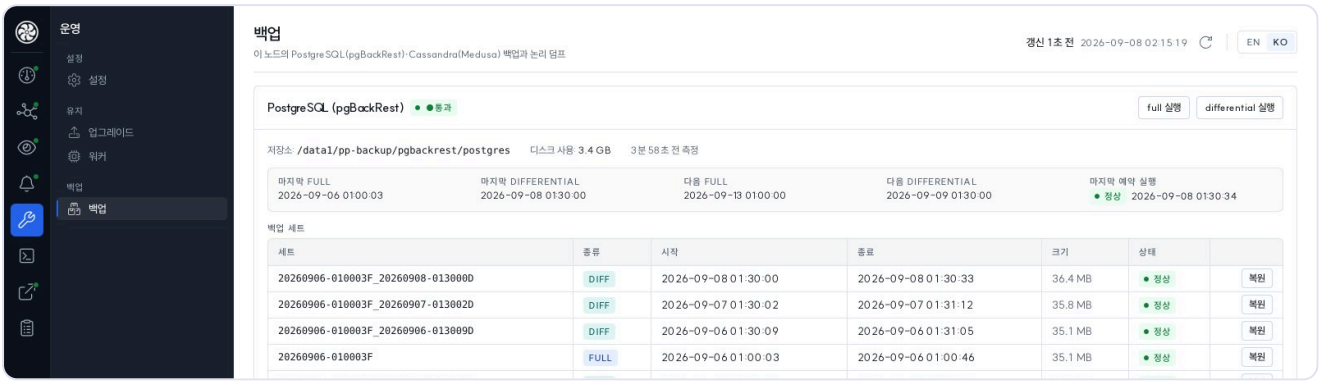


그림 12. 백업 화면 — PostgreSQL(pgBackRest) 의 마지막 full · differential, 다음 예정, 백업 세트 목록과 **복원 버튼**. Aster TSDB(Medusa) 도 같은 모양으로 아래에 이어집니다. *실제 구동 인스턴스 캡처입니다.*

기본 스케줄 — 설치하면 이미 돌고 있습니다

대상	도구	주기
PostgreSQL	pgBackRest	differential 매일 01:30 · full 매주 일요일 01:00
Aster TSDB	Medusa	differential 매일 01:20 · full 매주 일요일 00:10
보존 정리	purge	매주 일요일 03:00 — 보존 기간 지난 세트 삭제
콜드 아카이브	Temporal + Spark	매일 02:20 (앞 장) — 백업과 별개의 두 번째 사본

- 스케줄은 컨테이너 안 **systemd 타이머**가 실행하고, 끈 상태면 «설정에 의해 건너뜀»을 **기록으로 남깁니다** — 조용히 안 돈 백업은 없습니다.
- 백업 루트(`/data1/pp-backup`)는 **별도 디스크나 NAS**로 두기를 권장합니다. 외부 S3 동기화는 구성 항목입니다.
- 부분 유실에는 **데이터 복구 도구**가 있습니다 — 원시 태그 포인트에서 설비 단위 데이터를 기간·설비 지정으로 재구성합니다.

업그레이드 — 이미지 하나 바꾸는 일

무엇이 도는지 보인다

콘솔 «업그레이드» 가 이미지 태그 · 빌드 커밋 · 굽기 시각을 보여주고, 바꾸는 명령을 적어 줍니다.

설정 드리프트가 없다

모든 런타임 설정은 **템플릿에서 매 기동 때 렌더**됩니다. 컨테이너 안에서 손으로 고친 것은 남지 않습니다.

되돌릴 수 있다

이전 이미지 태그로 돌리면 끝. 데이터는 볼륨에, 설정은 템플릿에 있어 이미지와 분리돼 있습니다.

> **이 장의 화면은 모두 상시 가동 중인 개발 인스턴스**에서 캡처했습니다 — 백업 세트 · 볼륨 사용률 · 노드 부하가 실제 값입니다. 고객사 환경에서는 보존 기간 · 백업 주기 · 노드 수를 도입 단계에서 함께 정합니다.

06 · 특장점

④ 지표를 만들지 않아도 나옵니다

OEE 를 보려고 정의부터 계산 로직까지 새로 만드는 일이 흔합니다. PlantPulse Platform 은 수집된 데이터에서 **국제 표준 지표를 자동 생성**합니다.

지표	기준	내용
OEE	ISO 22400	가용성 × 성능 × 품질을 수집 즉시 계산하고 6대 손실로 분해합니다
RAM	신뢰성 공학	MTBF · MTTR · 가용도 · 신뢰도 R(t) 를 운영 데이터에서 자동 산출합니다
EMS	ISO 50001	에너지 원단위 · 피크 · CO_2 산정을 자동 집계합니다

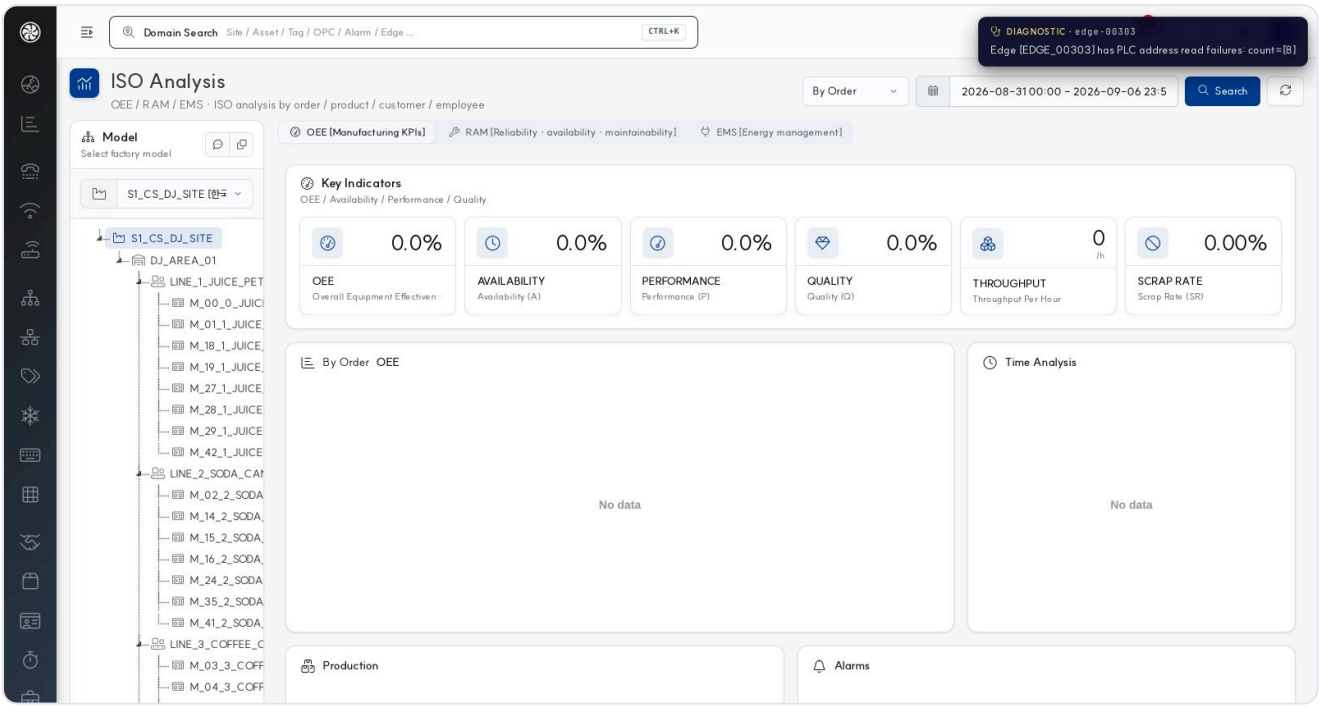


그림 13. ISO 분석 화면 — 오더 · 제품 · 고객 · 작업자 기준으로 OEE · RAM · EMS 를 나눠 봅니다.

왜 "표준 정의 그대로"가 중요한가

- **사이트 간 비교가 됩니다.** 공장마다 정의가 다르면 숫자를 합칠 수 없습니다.
- **감사·보고에 그대로 씁니다.** 표준 정의라 근거를 따로 설명할 필요가 없습니다.
- **지표 정의를 사람이 관리하지 않습니다.** 담당자가 바뀌어도 계산이 흔들리지 않습니다.

06 · 특장점

⑤ 공장을 지식그래프로 만듭니다

자산을 트리로 분류하는 것까지는 흔합니다. PlantPulse Platform 은 그 위에 **설비 사이의 관계**를 얹어, 공장 전체를 질의 가능한 지식그래프로 만듭니다.

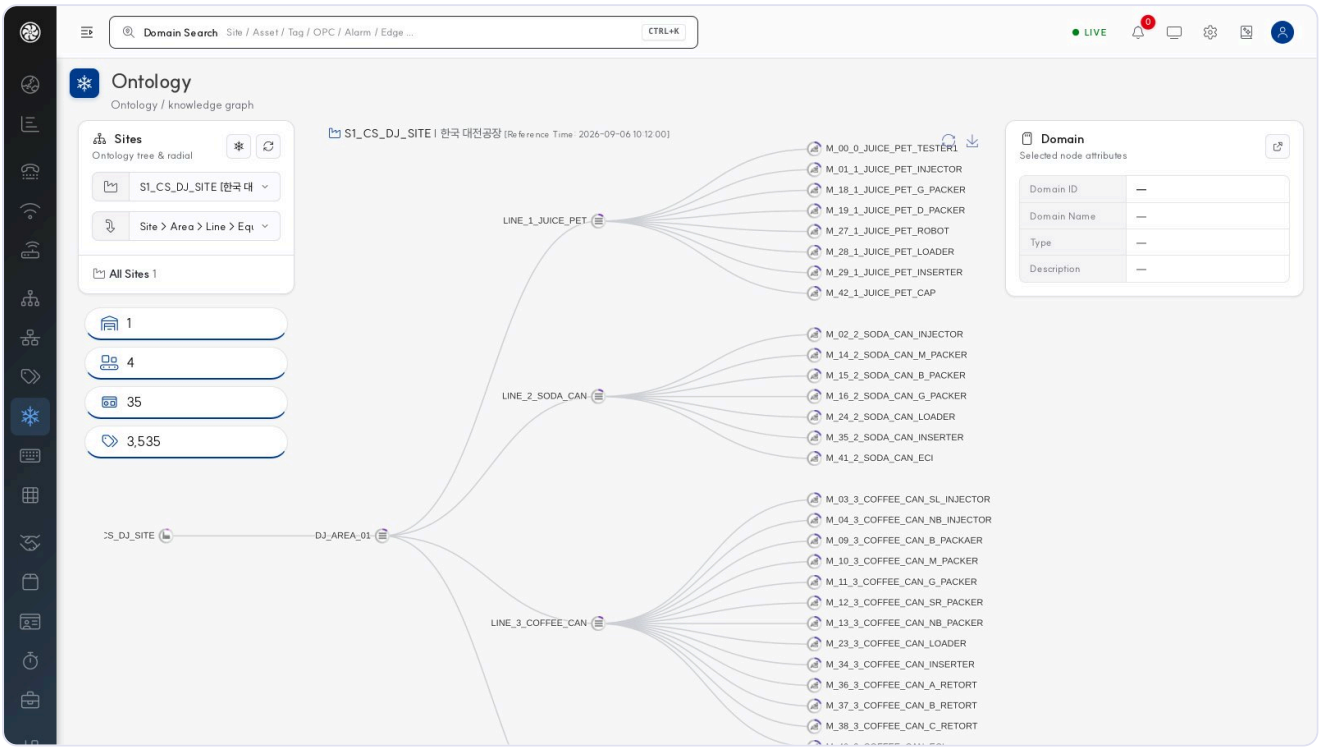


그림 14. 온톨로지 화면 — 사이트·자산·태그와 그 사이의 관계를 그래프로 탐색합니다.

트리로는 답할 수 없는 질문

"이 설비가 멈추면 어디까지 영향인가"

공급·의존·생산 방향 관계를 따라 하류 영향 범위를 그래프 탐색으로 답합니다.

"알람 설정이 빠진 태그가 있나"

설정값 태그와 알람 설정 레지스트리를 관계로 엮어 누락을 찾아냅니다.

사람과 AI 가 같은 맥락을 봅니다

온톨로지는 표준 지식 표현 형식으로 저장되고 읽기 전용 질의로 열려 있습니다. 내장 AI 는 이 그래프를 **근거**로 삼아 진단하므로, "왜 그렇게 판단했는가"를 설비 관계로 설명할 수 있습니다.

> **표준 계층을 넘어서는 이유.** 표준 자산 계층은 "어디에 있는가"를 말해 주지만 "무엇에 영향을 주는가"는 담지 못합니다. 고장 파급·원인 추적처럼 실제로 급한 질문은 대부분 후자입니다.

06 · 특장점

⑥ 자산 계층이 곧 실시간 주소 체계입니다

벤더마다 다른 통신 방식을 외부 시스템이 하나씩 배워야 한다면 연동은 끝나지 않습니다. **Unified Namespace**는 자산 트리 자체를 실시간 데이터의 **단일 주소 체계**로 만듭니다.

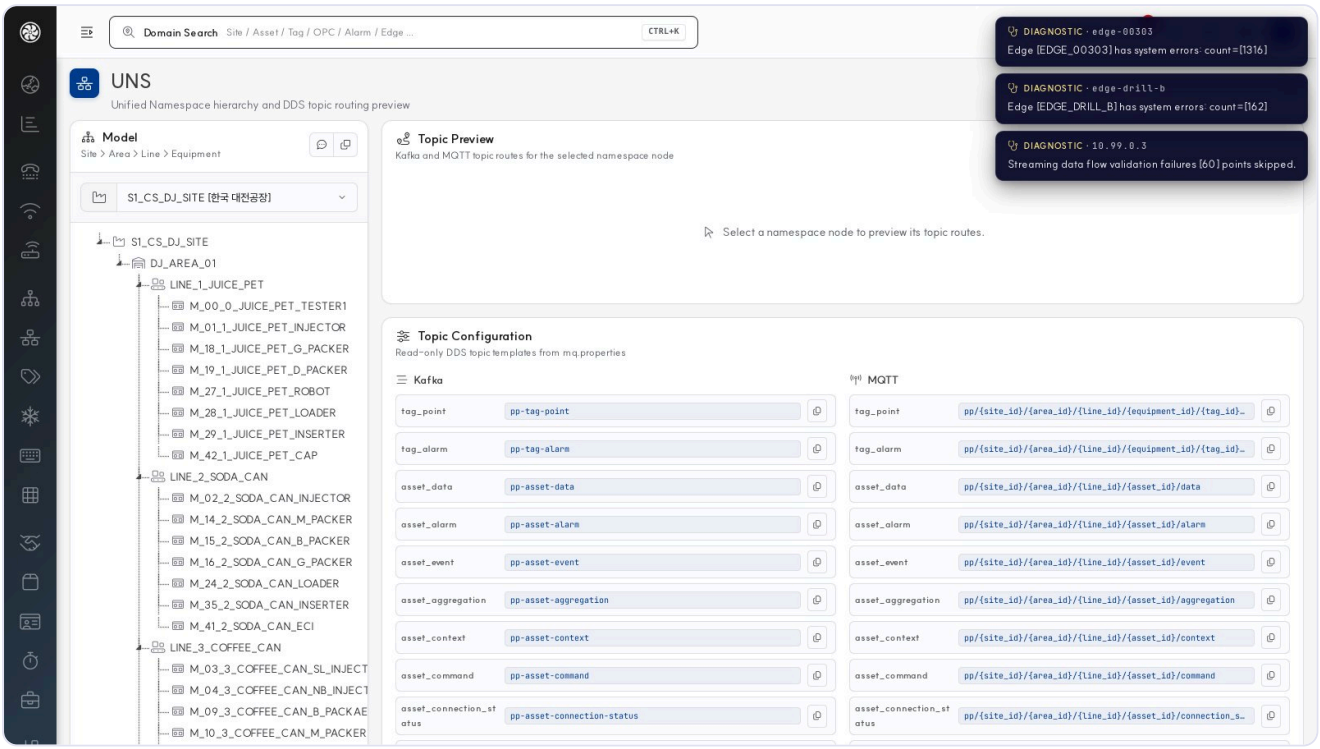


그림 15. UNS 화면 — 사이트 → 구역 → 라인 → 설비 계층이 그대로 토픽 경로가 되고, 콘솔에서 경로를 미리 보고 복사해 외부 시스템에 바로 붙입니다.

무엇이 미러링되나

데이터
측정값 · 집계 · 예측

사건
알람 · 이벤트 · 명령

상태
연결 상태 · 건강 상태

세 가지 산업 표준 토픽 체계로 자동 매핑되므로, 외부 시스템은 **하나의 규칙**만 알면 공장 전체를 구독할 수 있습니다.

연동이 개발에서 설정으로

- 새 설비가 들어와도 **주소 체계가 자동으로 확장**됩니다 — 연동 대상 목록을 다시 관리하지 않습니다.
- 연결·건강 상태까지 같은 체계로 나오므로 **죽은 설비를 구독자가 스스로 압니다.**
- 벤더별로 흩어진 통신이 **하나의 표준 네임스페이스**로 모입니다.

07 · 특장점

⑦ 자동화를 운영자가 직접 만듭니다

"알람이 뜨면 작업지시를 만들고 담당자에게 알리고 MES 에 넘긴다" — 이런 흐름은 늘 개발 항목이었습니다. **Flow** 는 그것을 화면에서 그리게 합니다.

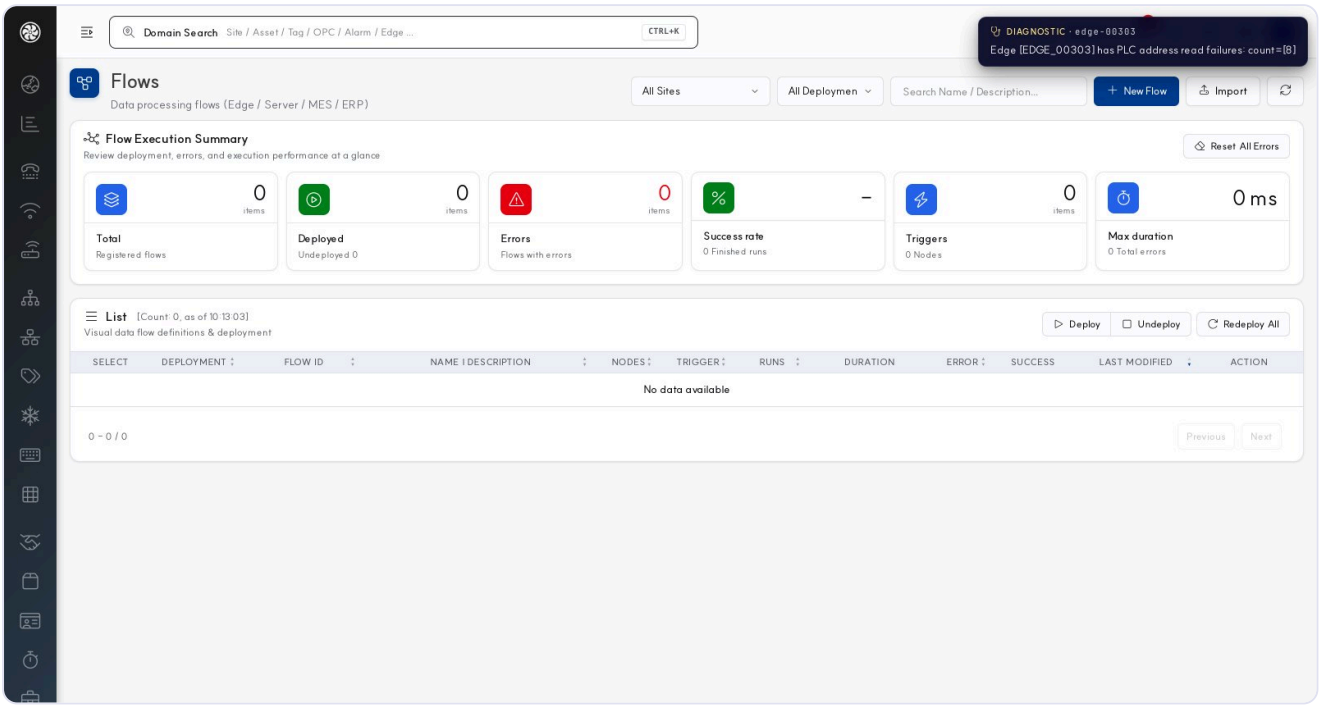


그림 16. Flow 목록 — 엣지 · 서버 · MES · ERP 대상 데이터 처리 흐름을 만들고 실행 이력을 관리합니다.

무엇을 이어붙일 수 있나

구분	예
트리거	태그 값 · 알람 · 자산 이벤트 · OEE 변화 · 엣지 상태
가공	필터 · 스크립트 · 집계 · 재시도 · 흐름 제어
액션	작업지시 발행 · 이메일/SMS/푸시 알림 · 외부 시스템 호출 · 엣지 제어

100여 종의 노드를 드래그앤드롭으로 연결해 한 화면에서 구성합니다.

운영자가 만들어도 안전하게

배포 · 롤백

문제가 생기면 이전 버전으로 되돌립니다.

자동 버전관리

변경 시점마다 스냅샷이 남습니다.

라이브 디버그

실제 데이터가 흐르는 것을 보며 고칩니다.

> **개발 대기열이 사라집니다.** "이 알람도 메신저로 받게 해 주세요" 같은 요청이 개발 티켓이 아니라 현장 담당자의 30분짜리 작업이 됩니다.

07 · 특장점

⑧ AI 가 쓸 수 있는 형태로 열려 있습니다

AI 에게 데이터베이스 접근 권한을 주는 것과, **운영 개념으로 물어볼 수 있게** 하는 것은 전혀 다른 일입니다. PlantPulse Platform 은 후자를 표준 방식으로 제공합니다.

테이블이 아니라 운영 개념으로

플랫폼의 데이터를 **산업 운영형 고수준 도구**로 개방합니다. AI 에이전트는 SQL 을 쓰는 대신 "3라인 이번 교대 OEE" 처럼 묻고 답을 받습니다.

제공되는 도구의 종류

설비 상태 조회 · 시계열 조회 · 알람 조회 · OEE 등 지표 · 온톨로지 관계 탐색 · 문서 검색 등 100여 종의 고수준 도구.

누가 붙을 수 있나

PlantPulse AI 뿐 아니라 **사내에서 만든 에이전트**와 표준을 지원하는 어떤 LLM 도구도 같은 방식으로 연결됩니다.

열되, 통제 안에서

- 모든 도구는 **플랫폼의 권한 체계** 안에서 실행됩니다 — AI 라고 더 볼 수 있지 않습니다.
- 호출은 **감사 로그**에 남습니다 — 누가 무엇을 물었는지 추적됩니다.
- **온프레미스에서 실행**됩니다 — 폐쇄망·에어갭 환경에서도 완결됩니다.

왜 이것이 자산인가

AI 도구는 계속 바뀝니다. 데이터를 **표준 방식으로 열어 두면** 도구가 바뀌어도 연동을 다시 만들지 않습니다. 특정 AI 제품에 종속되지 않는 구조입니다.

08 · 아키텍처

데이터가 지나가는 길

엣지에서 올라온 한 건의 측정값이 어떤 경로로 저장되고, 지표가 되고, 화면과 AI 에 닿는지를 한 장으로 봅니다.

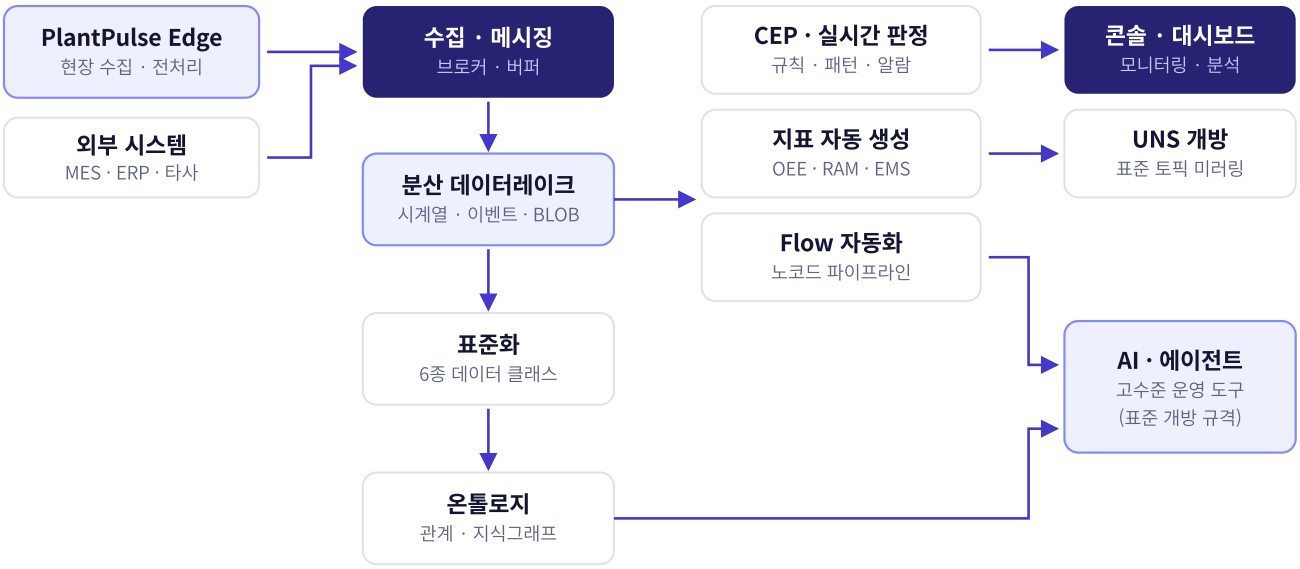


그림 17. 수집에서 개방까지. 표준화와 온톨로지를 거친 데이터가 화면·외부 시스템·AI 에 **같은 형태**로 도달하는 것이 이 구조의 요점입니다.

09 · 사용 흐름

운영자가 매일 보는 화면

아래 화면은 모두 **실제로 가동 중인 시스템에서 캡처**한 것입니다.

통합 모니터링 — 한 화면에서 시작합니다

사이트 지도, 데이터 파이프라인의 각 단계, 인프라 구성요소의 건강도, 주요 지표가 한 화면에 있습니다. **"지금 정상인가"**를 스크롤 없이 판단하도록 배치했습니다.

데이터 파이프라인

수신 → 처리 → 버퍼 → 저장 각 단계의 처리량과 상태를 개별로 표시합니다. 어디서 밀리는지 한눈에 드러납니다.

인프라 건강도

캐시 · 메시징 · 저장소 · 처리 엔진 등 구성요소별 상태를 나열합니다. 하나가 흔들리면 그 줄만 색이 바뀝니다.

플랫폼 건강 타임라인

하루 동안의 정상·경고·심각 구간을 띠로 보여줍니다. "언제부터 이상했나"를 되짚습니다.

주요 지표

통신 연결률 · 오늘 수집량 · 알람 · 설비 가동률 · 가용도 · 종합 OEE.

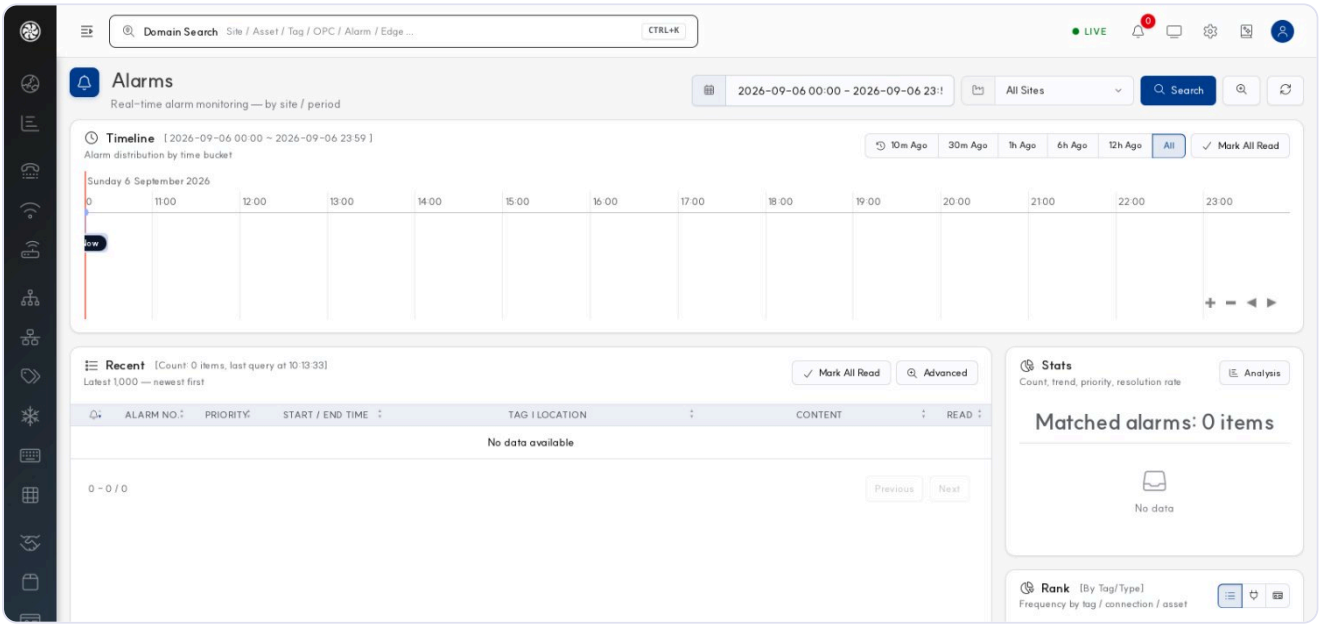


그림 18. 알람 화면 — 사이트·기간별 실시간 알람을 조회합니다. 단순 임계 알람과 CEP 가 만든 고급 알람이 같은 화면에 모입니다.

09 · 사용 흐름

엣지와 자산을 관리한다

엣지 연결 상태

현장에 깔린 엣지 게이트웨이의 연결과 수집 상태를 중앙에서 봅니다. 설비가 아니라 **수집기 자체가 살아 있는지**를 먼저 확인하는 화면입니다.

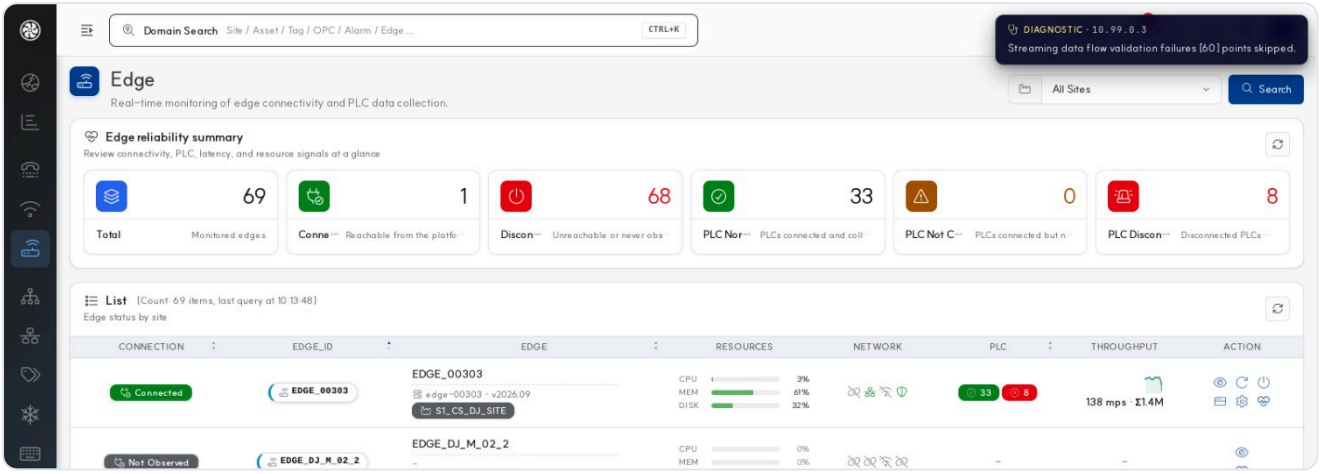


그림 19. 엣지 화면 — 사이트별 엣지의 연결 상태와 PLC 수집 현황을 실시간으로 확인합니다.

대시보드는 직접 만듭니다

표준 화면 외에 필요한 대시보드는 **캔버스**로 직접 구성합니다. 자산·태그를 끌어다 놓아 화면을 만들고 공유합니다.

자산 탐색

트리와 관계 그래프로 설비를 찾아 들어갑니다.

도메인 검색

사이트·자산·태그·알람·엣지를 한 검색창에서 찾습니다.

감사 로그

누가 무엇을 바꿨는지 기록으로 남습니다.

> **화면이 많은 것이 목적이 아닙니다.** 운영자가 하루에 여는 화면은 서너 개입니다. 나머지는 문제가 생겼을 때 파고드는 경로이고, 그 경로가 **같은 데이터 위에** 놓여 있다는 것이 이 제품의 값입니다.

10 · 보안

보안 · 운영

공장 데이터 전체가 모이는 시스템입니다. 그래서 **누가 무엇을 볼 수 있는가**와 **무엇이 기록되는가**가 기능만큼 중요합니다.

접근 통제

- **사용자 · 역할 기반 권한** — 사이트·자산 단위로 열람 범위를 나눕니다.
- **API 토큰** — 외부 시스템 연동은 발급된 토큰으로만. 발급·회수가 콘솔에서 관리됩니다.
- **AI 도구도 같은 권한 체계** — 에이전트라고 더 볼 수 있지 않습니다.
- **감사 로그** — 조회·변경·연동 호출이 기록됩니다.

망 구성

플랫폼은 **고객 인프라 안에 설치**됩니다. 외부 연결 없이 폐쇄망에서 완결되며, AI 기능까지 온프레미스로 구성할 수 있습니다 — 데이터가 사이트를 벗어나지 않습니다.

고가용성

항목	내용
저장소 이중화	복제 기반 분산 저장 — 노드 장애 시에도 데이터 유실 없음
노드 확장	운영 중 노드 추가로 용량·처리량 확장
구성요소 감시	인프라 구성요소별 헬스체크와 상태 타임라인
엣지 단절 대응	엣지 측 버퍼링으로 상위 단절 구간의 데이터 보전

> **도입 전 보안 검토를 권장합니다.** 위 항목은 제품이 기본 제공하는 것이며, 고객사 보안 정책(망 분리 수준, 인증 연동, 로그 수집 대상)에 맞춘 구성은 도입 단계에서 함께 확정합니다.

11 · 납품

설치 · 납품 모델

7개 계층 24개 이상의 구성요소가 **하나의 설치 패키지**로 제공됩니다. 구성요소를 따로 설치하고 버전을 맞추는 작업이 없습니다.

설치
패키지 하나로 전 계층 배포. 설치 후 수 분 내 전 파이프라인 가동.

폐쇄망
인터넷 없는 환경을 위한 오프라인 번들 제공.

확장
운영 중 노드 추가로 저장·처리 용량 확대.

규모 산정의 기준

기준	영향
태그 수 · 수집 주기	초당 수집 건수를 결정합니다 — 노드 수 산정의 1차 기준
보존 기간	저장 용량을 결정합니다. 원본·집계 보존 기간을 나눠 잡습니다
동시 사용자 · 대시보드	조회 부하를 결정합니다
CEP 규칙 수	실시간 판정 부하를 결정합니다

위 네 가지를 알려주시면 **권장 노드 구성과 사양**을 산정해 드립니다.

업그레이드

정기 릴리즈로 제공하며, 검증을 마친 특정 시점 버전에 고정해 운영할 수 있습니다. 업그레이드는 계층별로 흩어지지 않고 **패키지 단위**로 이뤄집니다.

엣지와의 관계
플랫폼은 **PlantPulse Edge** 와 함께 쓸 때 가장 자연스럽습니다. 다만 필수는 아닙니다 — 표준 프로토콜로 올라오는 데이터라면 타사 게이트웨이도 받습니다.

12 · 레퍼런스

레퍼런스

15년간 제조 현장에서 검증된 기술입니다.

162 수행 프로젝트	**34** 운영 중 사이트	**642,021** 연결된 센서
100TB+ 관리 중 데이터	**15년** 제조 현장 경험	**6** 수상 실적

대표 구축 사례

발전 · 중공업
두산
ICT 융합 풍력 운영 시스템 — 디지털 트윈 프레임워크와 결합해 터빈 건전성 추적과 예지보전으로 가동률을 개선.

식음료
롯데칠성
생산 라인 대상 스마트팩토리 플랫폼과 실시간 OEE 체계 구축.

조선 · 해양
한화오션
조선소 대규모 OT 데이터의 실시간 파이프라인 구축 — 건조·운영 데이터의 디지털화.

물류 · 전자
삼성전자
차세대 물류센터 OT 데이터 플랫폼 — 물류 설비·공정 데이터의 실시간 통합 운영 기반.

공공 · 실증
AX 실증단지
산업 디지털 전환(AX) 실증단지 구축 사업자로 선정.

연구 · 실증
KERI
한국전기연구원 스마트팩토리 실증 플랜트의 산업 데이터 플랫폼 구축 — 설비 데이터 수집부터 실시간 모니터링까지.

함께해 온 고객

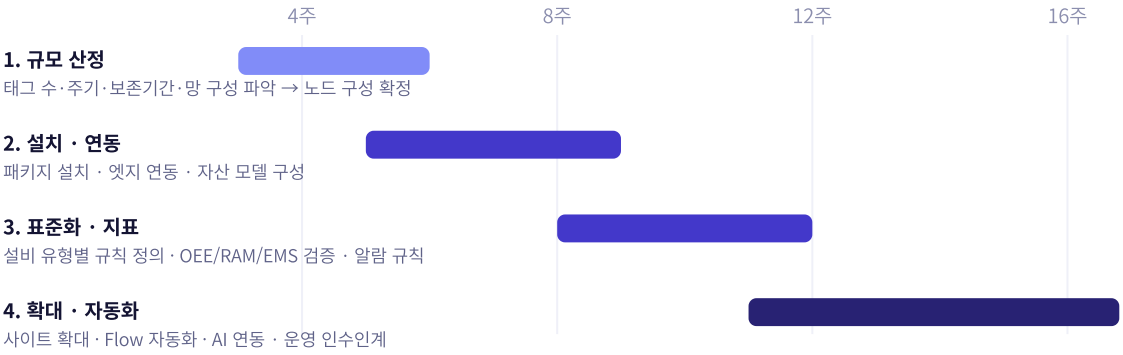
삼성전자 · 삼성물산 · LG · 두산 · 두산모트롤 · 한화 · 한화오션 · 한화에어로스페이스 · 한화엔진 · 롯데칠성 · 롯데케미칼 · 롯데정밀화학 · 롯데웰푸드 · 아모레퍼시픽 · 현대로템 · 현대모비스 · HD현대중공업 · 효성 · 한솔 · SK케미칼 · CJ대한통운 · KAI · KEPCO · KT · 덴소 · 캐리어 · 씨젠 · 신라 · 위니아 · STX · 일진 · KERI · KEIT · 산업통상자원부 외.

> **산업별 상세 구축 범위와 도입 효과**는 담당 영업 채널을 통해 개별 안내드립니다.

13 · 도입

도입 절차 및 일정

한 사이트에서 시작해 전사로 확장하는 **4단계**를 권장합니다.



PoC 에서 확인하시길 권합니다

- 우리 설비의 태그 규모에서 **수집이 밀리지 않는가**
- 설비 유형 규칙 하나로 **같은 유형 설비가 그대로 붙는가**
- 표준 정의로 계산된 OEE 가 **현장 인식과 맞는가**
- 폐쇄망에서 **AI 기능까지 완결되는가**

도입 단계에서 함께 정할 것

보존 정책

원본과 집계의 보존 기간을 나눠 저장 비용을 설계합니다.

기존 시스템 관계

대체할 것과 병행할 것을 구분합니다.

권한 체계

사이트·자산 단위 열람 범위와 사내 계정 연동 방식.

AI 실행 위치

온프레미스 모델과 외부 API 중 망 정책에 맞는 쪽.

PlantPulse® Platform

INDUSTRIAL DATA INFRASTRUCTURE

도입 문의

산업 AX 전환, 지금 시작하세요.

도입 상담 · 데모 요청 · 견적 문의 — 무엇이든 남겨주세요. 담당 전문가가 영업일 기준 1일 이내에 답변드리고, 원하시면 실제 제품 화면으로 데모 시연도 바로 잡아드립니다.

전화
1588-5268

이메일
webmaster@kopens.com

도입 문의 · 데모 요청
kopens.com/ko/contact

회사 홈페이지
kopens.com

제품 홈페이지
plantpulse.io

제품 문서
docs.plantpulse.io

PlantPulse® 제품군 — 함께 도입하면 현장부터 AI 까지 한 벌로 이어집니다

Edge
산업 프로토콜 엣지 게이트웨이

Platform
엔터프라이즈 IIoT / DataOps 플랫폼

AI
플랜트 운영 AI 코파일럿

Studio
AI 애플리케이션 빌더

㈜한국오픈솔루션 · KOPENS, Inc.

© 2026 KOPENS. All rights reserved. 플랜트펄스®는 ㈜한국오픈솔루션의 등록상표입니다.
본 문서 Rev. 2026.09 · 기재된 수치·화면은 작성 시점 기준이며 사전 통지 없이 변경될 수 있습니다.

kopens®