목차

- Monithub 소개
  - 1. Monithub가 해결하려는 문제
  - 2. 관측에서 대응까지 이어지는 하나의 흐름
  - 3. 누구를 위한 제품인가요?
  - 4. 먼저 알아두면 좋은 핵심 개념
  - 5. 주요 기능을 사용자 관점에서 살펴보기
  - 6. 실제 상황에서는 어떻게 사용하나요?
  - 7. 사용할 때 기억할 점
  - 8. 역할에 맞게 시작하기
- 제품 사용 가이드
  - 기본 사용 흐름
  - 메뉴 한눈에 보기
  - 먼저 알아둘 것
- 가입과 조직 참여
  - 새 조직을 만들면서 가입하기
  - 기존 조직에 참여 요청하기
  - 초대 메일로 참여하기
  - 초대가 진행되지 않을 때
- 멤버 초대와 참여 요청
  - 조직의 멤버 화면 열기
  - 이메일로 멤버 초대하기
  - 참여 요청 확인하기
  - 요청이 보이지 않거나 처리할 수 없을 때
- 홈과 기본 탐색
  - 상단 바
  - 인시던트 활동
  - 시스템 상태
  - 좌측 메뉴
  - 홈에서 바로 판단할 것
- 대시보드

    - 대시보드 선택
    - 필터와 시간 범위
    - 패널 읽기
    - 이상탐지
    - 편집 기능
- 패널 편집
    - 진입 방법
    - 저장과 취소
    - 타입 선택
    - Data 탭
    - Fields 탭
    - Axes & Legend
    - 타입별 문서
    - Panel & Layout
    - JSON
    - Row와 Text 패널
    - 추천 편집 순서
- Time Series 패널
    - Data
    - Fields
    - Visualization
    - 확인 순서
- Bar 패널
    - Data
    - Fields
    - Visualization
    - 확인 순서
- Stat 패널
    - Data
    - Fields
    - Visualization

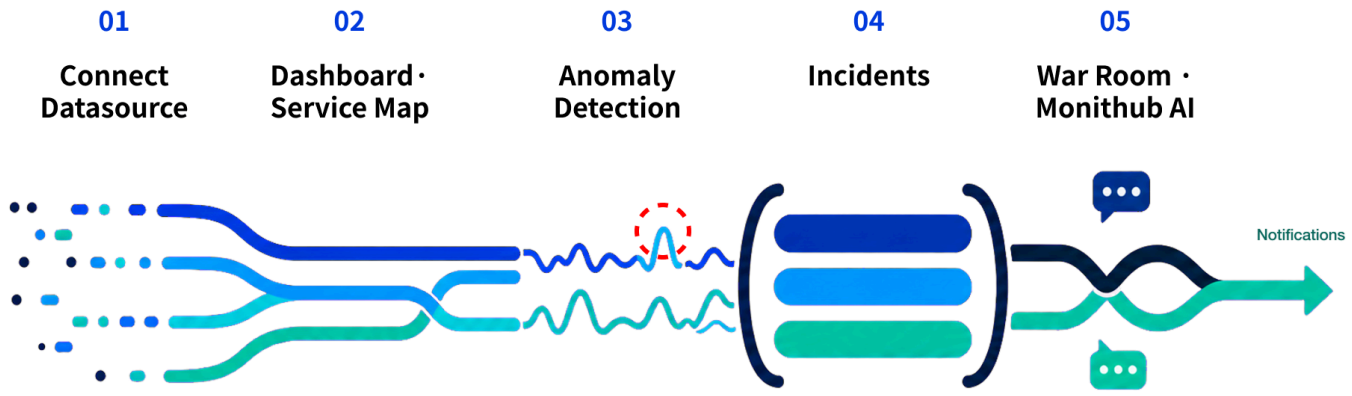
    - 확인 순서
- Gauge 패널
    - Data
    - Fields
    - Visualization
    - 확인 순서
- Table 패널
    - Data
    - Fields
    - Visualization
    - 확인 순서
- Heatmap 패널
    - Data
    - Fields
    - Visualization
    - 확인 순서
- Scatter 패널
    - Data
    - Fields
    - Visualization
    - 확인 순서
- Status Grid 패널
    - Data
    - Fields
    - Visualization
    - 확인 순서
- Trace 패널
    - Data
    - Fields
    - Visualization
    - 확인 순서

- Gantt 패널
  - Data
  - Fields
  - Visualization
  - 확인 순서
- Pie 패널
  - Data
  - Fields
  - Visualization
  - 확인 순서
- Logs 패널
  - Data
  - Fields
  - Visualization
  - 확인 순서
- 인시던트 및 워룸 개요
  - 권장 대응 흐름
  - 먼저 구분해야 할 세 가지 동작
  - 문서 묶음
- 인시던트 확인
  - 목록에서 대응할 문제 찾기
  - 상세에서 1차 판단하기
  - 담당자와 상태 관리하기
  - 필요한 신호 직접 추가하기
  - 정탐·오탐·무시로 근거 분류하기
  - 인시던트 전체 삭제하기
  - 검토를 워룸 대응으로 잇기
- 워룸 협업
  - 채널을 찾고 현재 상황 파악하기
  - 대응 기록 남기기
  - AI 요약과 명령어 사용하기

    - 오른쪽 작업 메뉴 활용하기
    - 채널을 닫기 전 확인하기
- 종료와 정리
    - 상태별 의미
    - 해결 처리 전 확인할 것
    - 종료 뒤 남길 기록
    - 삭제와 종료를 혼동하지 않기
    - 권장 마무리 순서
- 알림 확인과 설정
    - 최근 알림 빠르게 확인하기
    - 필요한 알림 찾기
    - 어떤 알림을 받을지 정하기
    - 알림을 받을 위치 정하기
    - Monithub를 보고 있을 때 알림 받기
    - 알림이 보이거나 들리지 않을 때
- Telegram 연결과 워룸 전송
    - Telegram 계정 연결하기
    - Telegram에서 알림 확인하기
    - Telegram 메시지를 워룸으로 보내기
    - 연결된 조직과 도움말 확인하기
    - 계정을 바꾸거나 연결 해제하기
    - 연결이나 전송이 되지 않을 때
- Connect 시작하기
    - 시작하기 전에
    - 첫 연결 만들기
    - 설정 진행 단계
    - 생성된 설정 적용하기
    - 카탈로그 선택하기
    - 데이터 수집 확인하기
- 인프라 연결
    - Linux 호스트

- Docker
- NGINX
- 연결 후 확인

Monithub 소개



Monithub는 대시보드, 회의 기록, 문서와 대화방에 흩어진 운영 맥락을 하나로 연결하는 관측 및 인시던트 대응 플랫폼입니다. 문제가 발생할 때마다 여러 화면에서 데이터를 찾고, 흩어진 기록을 모아 상황을 다시 설명하는 반복을 줄이기 위해 만들었습니다. 시스템에서 발생하는 신호와 대응 기록을 연결해, **"지금 어디에 문제가 있고, 무엇이 영향을 받았으며, 우리 팀은 다음에 무엇을 해야 하는가?"**라는 질문에 팀이 함께 답하도록 돕습니다. 대응 과정에서 남은 근거와 판단, 해결 결과는 다음 대응과 탐지 기준 개선에 다시 쓰여 운영 지식이 담당자의 기억에만 머물지 않게 합니다.

메트릭, 로그, 트레이스와 네트워크 연결 정보는 시스템 상태를 이해하는 핵심 자료입니다. 그러나 데이터가 서로 다른 화면에 있고 판단과 논의가 별도의 회의록이나 대화방에 남으면, 문제를 발견할 때마다 맥락을 처음부터 다시 구성해야 합니다. 이상 징후가 나타난 시점과 영향 대상을 찾고, 근거를 모아 담당자와 대응 과정을 공유하는 일까지 하나의 흐름으로 이어져야 문제를 빠르게 해결할 수 있습니다. Monithub는 관측, 판단, 협업과 기록이 도구 사이에서 끊기지 않도록 연결합니다.

> **이 문서에서 알 수 있는 것**
>
> 이 문서는 화면 조작 순서보다 Monithub가 해결하는 문제, 기능 간 연결과 팀의 시작 지점을 설명합니다. 실제 화면 사용 방법은 제품 사용 가이드에서 확인할 수 있습니다.

1. Monithub가 해결하려는 문제

운영 중인 시스템에서는 하나의 문제가 여러 신호로 나타납니다. 응답 시간이 느려지고 오류 로그가 늘어나며, 특정 서비스의 호출이 실패하거나 Kubernetes 리소스 상태가 달라질 수 있습니다. 신호를 따로 보면 중요한 변화인지 일시적인 흔들림인지 판단하기 어렵습니다.

문제를 발견한 뒤에도 담당자는 여러 화면에서 근거를 모으고, 메신저에서 상황을 다시 설명하며, 누가 무엇을 확인했는지 기록해야 합니다. 대응이 끝나면 흩어진 기록을 다시 정리하고, 같은 유형의 신호가 반복될 때는 탐지 기준도 조정해야 합니다.

이 과정이 개인의 경험과 기억에만 남으면 담당자가 바뀔 때마다 같은 조사와 설명을 반복하게 됩니다. 대응 중 확인한 근거와 판단, 조치 결과가 다음 문제 해결과 탐지 개선에 다시 쓰일 수 있어야 장애 대응과 운영 노하우가 조직 전체의 자산이 됩니다.

Monithub는 이 과정에서 반복되는 네 가지 어려움을 줄이는 데 초점을 둡니다.

운영 중 겪는 어려움	Monithub가 돕는 방식
데이터는 많지만 현재 상태를 한눈에 파악하기 어렵습니다.	홈, 대시보드와 서비스 맵에서 조직의 상태와 변화 흐름을 여러 관점으로 확인합니다.
여러 이상 신호 중 무엇부터 확인해야 할지 판단하기 어렵습니다.	Sentinel의 이상탐지 결과와 인시던트의 심각도, 영향 범위, 관련 근거를 함께 살펴봅니다.
분석 화면과 대응 대화가 분리되어 맥락이 끊깁니다.	인시던트에서 확인한 문제를 워룸으로 이어 담당자, 대화, 요약과 대응 기록을 같은 맥락에서 관리합니다.
대응 경험이 다음 탐지와 운영 개선으로 이어지지 않습니다.	인시던트 해결 기록과 탐지 결과 라벨을 함께 남기고, 이를 모델 학습·검토와 후속 개선에 연결해 운영 경험을 조직의 자산으로 축적합니다.

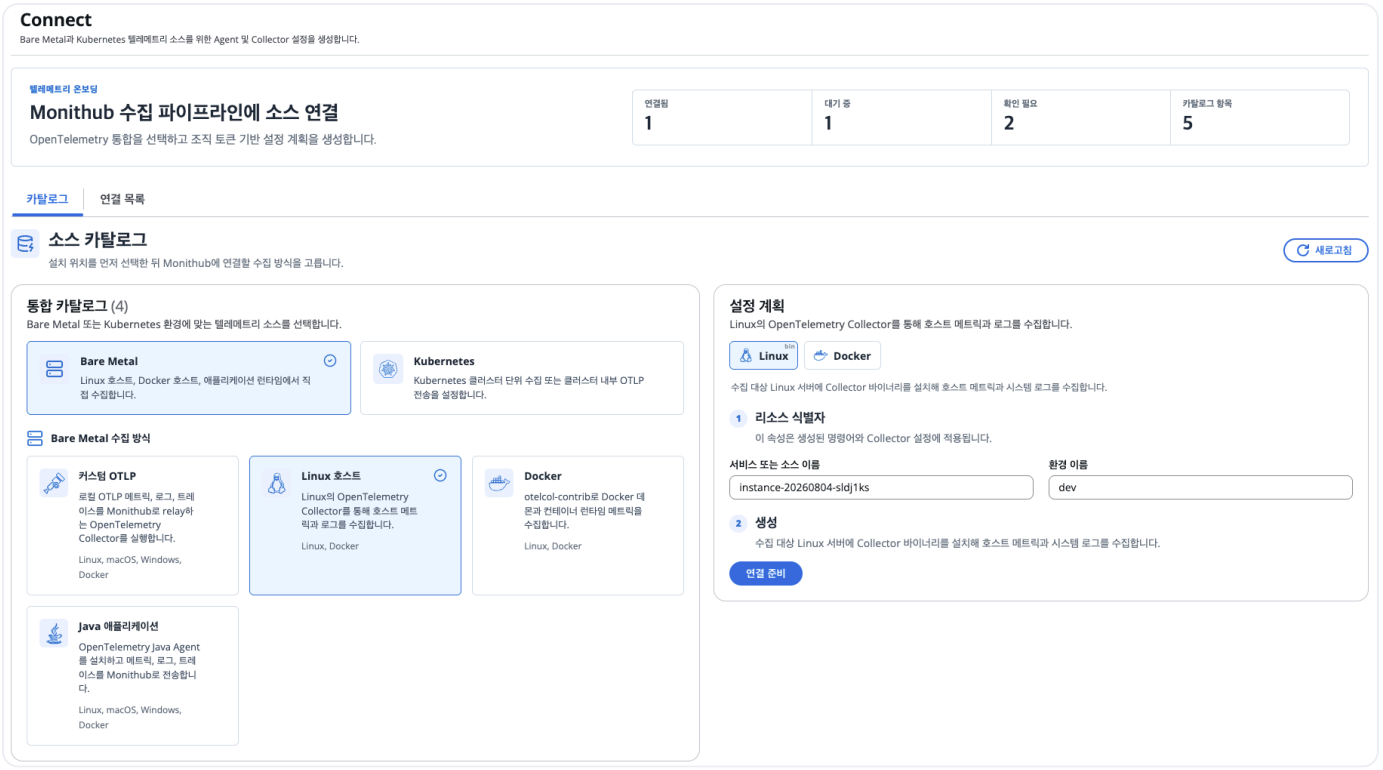
2. 관측에서 대응까지 이어지는 하나의 흐름

Monithub는 개별 기능의 모음이 아니라, 데이터를 연결하고 상태를 관측한 뒤 문제에 대응하며 그 경험을 다음 운영에 활용하는 하나의 흐름으로 이해할 수 있습니다.

2.1. 운영 데이터를 연결합니다

먼저 조직의 시스템에서 Monithub로 관측 데이터가 들어오도록 연결합니다. Connect에서는 OpenTelemetry Collector, Linux 호스트, Docker, Java 애플리케이션, Kubernetes와 Custom OTLP 같은 수집 대상을 선택하고 연결 계획을 세울 수 있습니다. Kubernetes 환경에서는 Monithub K8S Operator가 수집 컴포넌트를 설치하고 관리하도록 구성합니다.

연결한 뒤에는 설치 여부뿐 아니라 실제 데이터가 수집되는지, 어떤 서비스가 관찰되는지 확인해야 합니다. 연결 목록, 대시보드와 서비스 맵을 함께 살펴보면 수집 상태를 더 정확하게 검증할 수 있습니다.



2.2. 현재 상태와 변화 흐름을 관측합니다

홈은 로그인 직후 현재 조직의 열린 인시던트, 최근 활동과 시스템 상태를 요약해 보여 줍니다. 더 자세한 분석이 필요하면 대시보드에서 시간 범위와 변수를 조정하고, 메트릭·로그·트레이스·이벤트를 목적에 맞는 패널로 살펴봅니다.

서비스 맵은 서비스 간 호출 관계를 토폴로지로 보여 줍니다. 느리거나 실패한 연결을 중심으로 upstream과 downstream을 좁혀 보면 문제가 시작된 지점과 영향이 퍼지는 방향을 파악할 수 있습니다.

2.3. 평소와 다른 신호를 찾고 판단합니다

Monithub Sentinel은 관측 데이터에서 평소 패턴과 다른 구간을 찾습니다. 원본 데이터와 탐지 결과를 같은 시간축에 표시하므로 값의 변화와 이상 구간을 함께 비교할 수 있습니다.

이상 신호가 곧 확정된 장애를 의미하지는 않습니다. 운영자는 해당 시점의 시스템 상태와 관련 근거를 확인하고 정탐·오탐·미탐 같은 라벨을 남길 수 있습니다. 운영팀은 이 판단 기록과 실제 해결 결과를 바탕으로 새 후보 모델을 학습하고 품질을 검토한 뒤, 운영에 적합한 버전만 활성화합니다. 탐지 성향과 학습 조건을 조정하고 모델을 승격하는 일은 운영 근거에 따라 사람이 결정합니다.

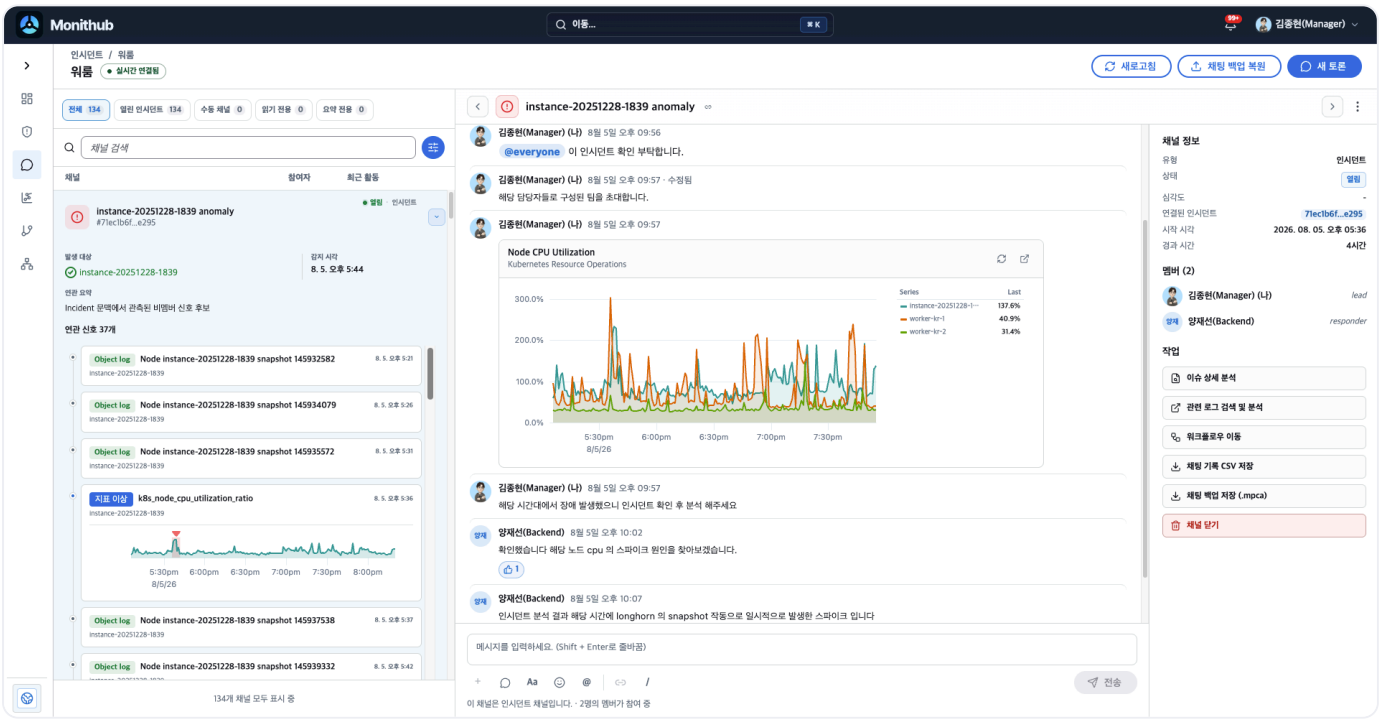
2.4. 관련 신호를 인시던트의 맥락으로 확인합니다

인시던트는 문제 대응의 시작점입니다. 개별 경고를 나열하는 대신, 함께 살펴봐야 할 이상 신호와 영향 대상, 심각도, 상태, 담당자와 시간 흐름을 한 공간에 모아 보여 줍니다.

운영자는 인시던트가 실제로 진행 중인지, 어떤 서비스나 리소스가 영향을 받았는지, 무엇을 먼저 확인해야 하는지 판단합니다. 메트릭, 로그, 트레이스와 Kubernetes 맥락을 함께 살펴보면 화면을 옮길 때마다 문제를 다시 설명하는 일을 줄일 수 있습니다. 발견 당시의 근거부터 최종 해결 결과까지 같은 인시던트에 남아 이후 유사한 문제를 검토할 때 다시 활용할 수 있습니다.

2.5. 워룸에서 함께 대응하고 기록합니다

워룸은 인시던트별 대화와 대응 기록을 관리하는 협업 공간입니다. 담당자를 부르고 확인한 사실과 가설을 공유하며, 누가 무엇을 확인했는지, 어떤 근거로 다음 행동을 정했는지, 조치 결과가 어땠는지를 한 흐름으로 남길 수 있습니다.



Monithub AI는 워룸 대화와 인시던트 맥락을 바탕으로 요약과 브리프 작성을 돕습니다. 긴 대화에서 확인된 내용과 남은 질문을 빠르게 파악하는 데 유용하지만, 최종 판단과 실행은 실제 시스템 상태를 확인한 담당자가 결정합니다.

이렇게 남은 기록은 종료 후 회고를 위한 자료에 그치지 않습니다. 새 담당자가 과거의 판단 근거를 파악하고, 같은 유형의 문제가 반복될 때 이전 조치와 결과를 재사용하는 조직의 운영 지식이 됩니다.

2.6. 대응 결과를 다음 운영에 반영합니다

문제를 해결한 뒤에는 원인, 영향 범위, 확인한 근거, 실행한 조치와 결과를 인시던트와 워룸의 맥락 안에서 정리합니다. 탐지 결과가 실제 상황과 달랐다면 운영팀은 해결 기록과 정탐·오탐·미탐 라벨을 바탕으로 새 모델 후보를 학습하고, 기존 모델과 비교·검토해 탐지 기준을 조정합니다.

관측, 판단, 대응과 피드백이 이렇게 연결되면 유사한 문제가 생겼을 때 과거의 근거와 해결 방법을 바로 찾아 더 일관된 기준으로 대응할 수 있습니다. 장애 대응과 운영 노하우가 조직 전체의 운영 자산으로 축적되어 특정 사람의 기억에 의존하지 않는 운영 관리를 돕는 것이 이 흐름의 목적입니다.

3. 누구를 위한 제품인가요?

Monithub는 운영 데이터를 직접 분석하는 사람뿐 아니라, 문제 상황에서 같은 맥락을 공유해야 하는 팀 전체를 위한 제품입니다.

사용자	Monithub에서 주로 하는 일
SRE·DevOps·운영 담당자	홈과 대시보드에서 상태를 확인하고, 서비스 맵과 인시던트 근거를 바탕으로 문제 범위를 좁힙니다.
개발자	담당 서비스의 메트릭, 로그와 트레이스를 확인하고, 워룸에서 원인 가설과 수정 결과를 공유합니다.
인시던트 리드	심각도와 영향 범위를 판단하고, 담당자와 다음 행동을 정하며, 대응 상황을 요약하고 종료 조건을 확인합니다.
플랫폼·조직 관리자	조직과 멤버, 데이터 연결, 수집 상태, 알림 채널, 사용량과 요금제를 관리합니다.
기술 책임자·팀 리더	현재 위험과 반복되는 문제, 대응 진행 상황을 파악하고 우선순위와 후속 개선 작업을 결정합니다.

한 사람이 여러 역할을 맡을 수도 있습니다. 소규모 팀에서는 관리자가 데이터 연결과 인시던트 대응을 함께 담당하고, 규모가 커지면 관측, 분석, 대응과 조직 관리 역할을 나눌 수 있습니다.

역할이나 담당자가 바뀌더라도 같은 인시던트 맥락과 해결 기록을 이어서 볼 수 있으므로, 팀은 개인의 경험을 조직의 공통된 운영 기준으로 발전시킬 수 있습니다.

4. 먼저 알아두면 좋은 핵심 개념

처음에는 메뉴 이름보다 아래 개념이 어떻게 이어지는지 이해하는 것이 중요합니다.

개념	쉽게 말하면
조직	사용자, 권한, 연결과 운영 데이터를 구분하는 작업 공간입니다. 상단에서 조직을 바꾸면 화면에 보이는 데이터 범위도 달라집니다.
관측 데이터	시스템 상태를 이해하기 위해 수집하는 메트릭, 로그, 트레이스, 이벤트와 네트워크 관계 정보입니다.
Connect	어떤 환경과 수집 대상을 Monithub에 연결할지 선택하고 설정 계획을 만드는 기능입니다.
대시보드	여러 관측 데이터를 목적에 맞는 패널로 배치해 함께 보는 화면입니다.
패널	숫자, 시계열, 표, 로그, 트레이스 등 하나의 질문에 답하도록 데이터를 표현하는 대시보드 구성 단위입니다.
서비스 맵	서비스 간 호출 방향과 연결 상태를 그래프로 보여 주는 화면입니다.
이상 신호	평소 패턴과 달라 추가 확인이 필요한 데이터 구간입니다. 이상 신호가 항상 실제 장애를 뜻하는 것은 아닙니다.
Sentinel 모델	이상 신호를 찾는 기준입니다. 운영자가 해결 기록과 탐지 결과 라벨을 바탕으로 새 후보를 학습·검토한 뒤 활성 버전을 운영에 사용합니다.
인시던트	관련 신호와 영향 대상을 묶어 문제의 근거, 판단, 조치와 해결 결과를 관리하는 작업 단위입니다.
워룸	인시던트 참여자가 누가 무엇을 확인하고 결정했는지 대화, 행동과 결과를 함께 남기는 협업 공간입니다.
알림	새 인시던트, Connect 수집 상태와 워룸 메시지처럼 사용자가 확인해야 할 변화를 전달하는 기능입니다.
Gateway	AI Gateway의 상태, 등록된 서비스와 모델, 라우팅 및 요청 테스트를 확인하는 운영 화면입니다.

5. 주요 기능을 사용자 관점에서 살펴보기

5.1. 홈: 어디에서부터 확인할지 정합니다

홈은 여러 운영 화면 중 첫 번째 확인 지점입니다. 열린 인시던트 수, 최근 활동, 영향 대상과 시스템 상태를 함께 보며 지금 확인해야 할 문제가 있는지 판단할 수 있습니다.

정상 여부를 단정하기보다 **오늘의 운영 상태에서 어디를 먼저 볼지 정하는 화면**입니다. 위험 또는 주의 상태가 보이거나 특정 대상이 반복해서 영향을 받았다면 대시보드나 인시던트로 이동해 근거를 확인합니다.

홈과 기본 탐색 방법 알아보기 →

5.2. 대시보드와 패널: 질문에 맞게 데이터를 구성합니다

대시보드는 Kubernetes 리소스, 애플리케이션, 로그, 트레이스, Gateway, 비용과 수집량 등 서로 다른 운영 주제를 패널 단위로 구성합니다. 시간 범위와 Namespace, Node, Pod 같은 변수를 사용하면 같은 대시보드에서도 분석 대상을 빠르게 좁힐 수 있습니다.

운영자는 큰 숫자로 현재 상태를 확인하고, 시계열로 변화 시점을 찾으며, 표와 로그에서 구체적인 근거를 살펴봅니다. 필요한 권한이 있으면 데이터 쿼리, 필드 표현, 시각화 방식과 패널 배치를 업무에 맞게 편집할 수 있습니다.

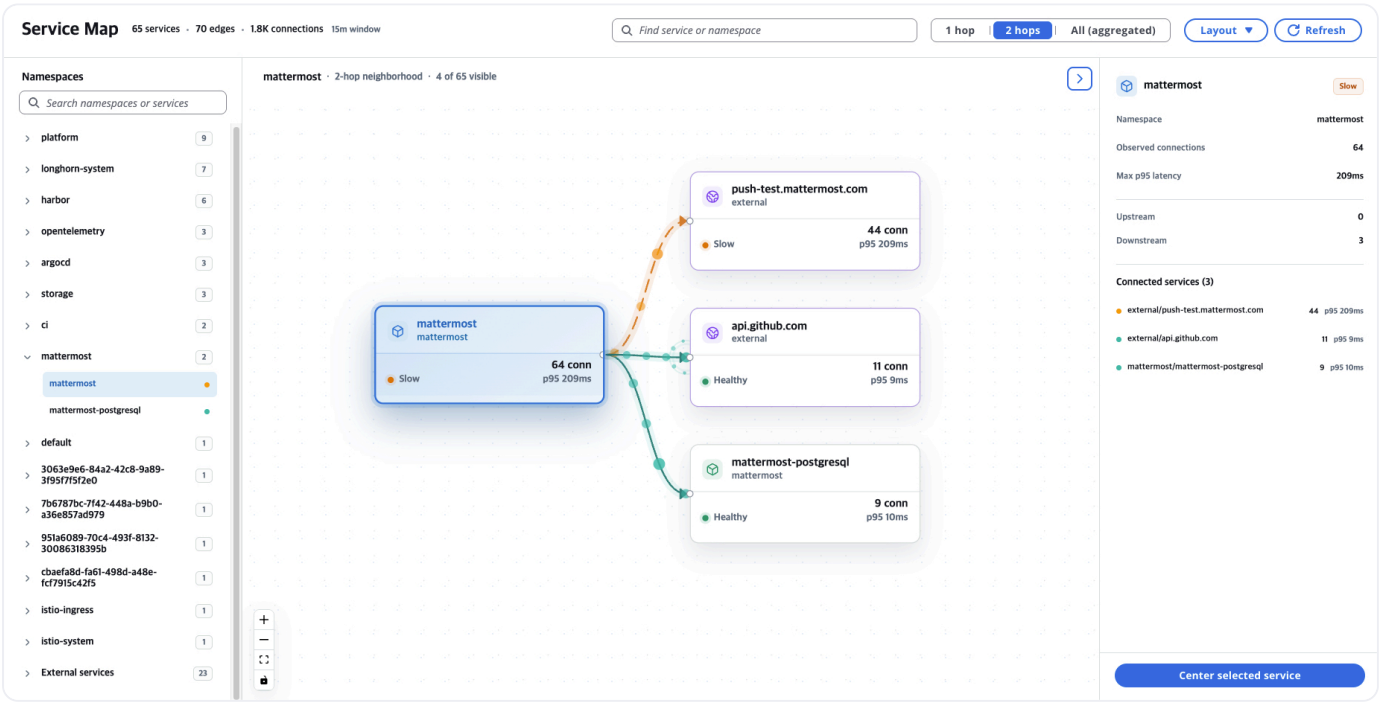
- **대시보드 사용 방법 →**
- **패널 편집과 시각화 유형 →**

5.3. 서비스 맵: 영향이 퍼지는 방향을 이해합니다

서비스 맵은 선택한 서비스를 중심으로 직접 연결된 upstream과 downstream을 보여 줍니다. 서비스 상태, 호출 방향과 지연 시간을 함께 비교하면 "어느 서비스가 느린가?"뿐 아니라 "어느 연결에서 문제가 시작되었고 다음에 무엇이 영향을 받는가?"까지 살펴볼 수 있습니다.

연결이 복잡한 환경에서는 1-hop부터 확인하고 필요할 때 2-hop으로 넓히는 것이 좋습니다. 전체 토폴로지로 큰 흐름을 비교하되, 실제 문제는 영향받은 서비스를 중심으로 범위를 좁혀 분석합니다.

서비스 맵에서는 왼쪽에서 네임스페이스와 서비스를 선택하고, 가운데에서 연결 흐름을 살펴보며, 오른쪽에서 연결 수와 지연 시간 같은 핵심 정보를 확인할 수 있습니다.

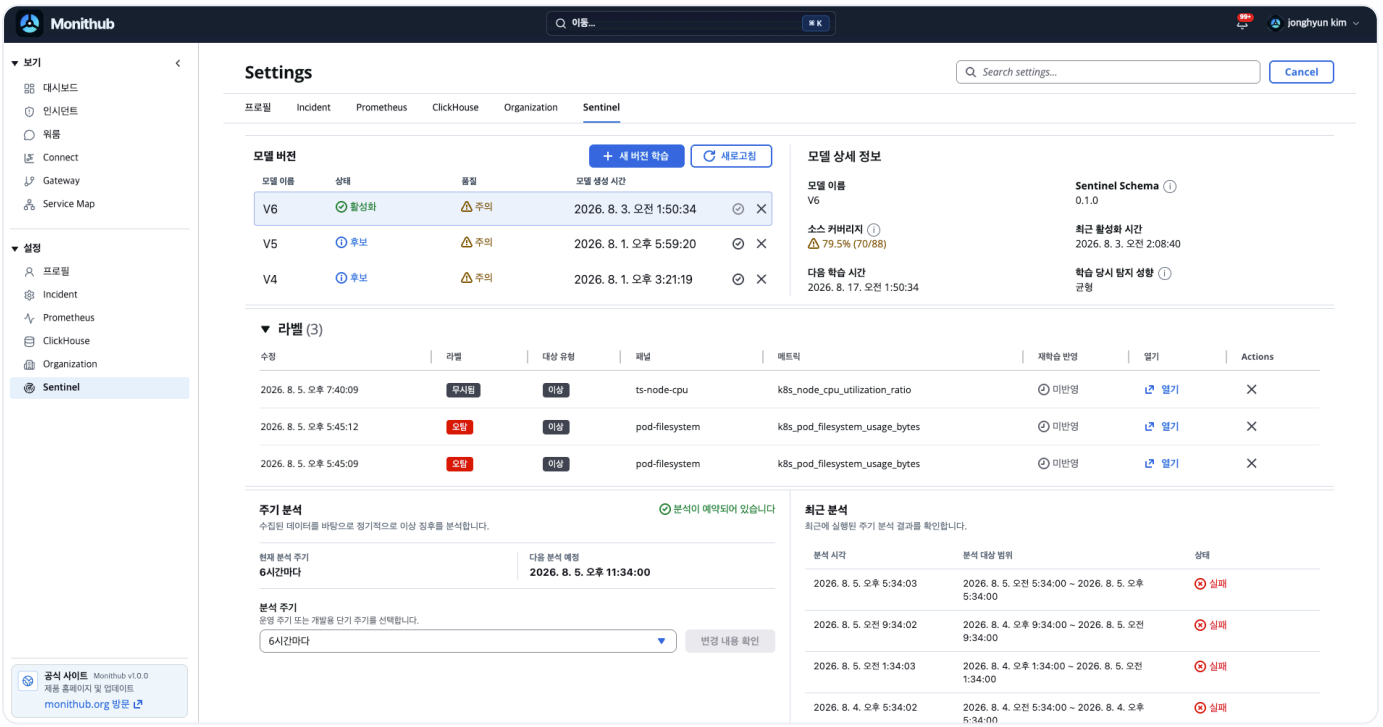


서비스 맵 사용 방법 알아보기 →

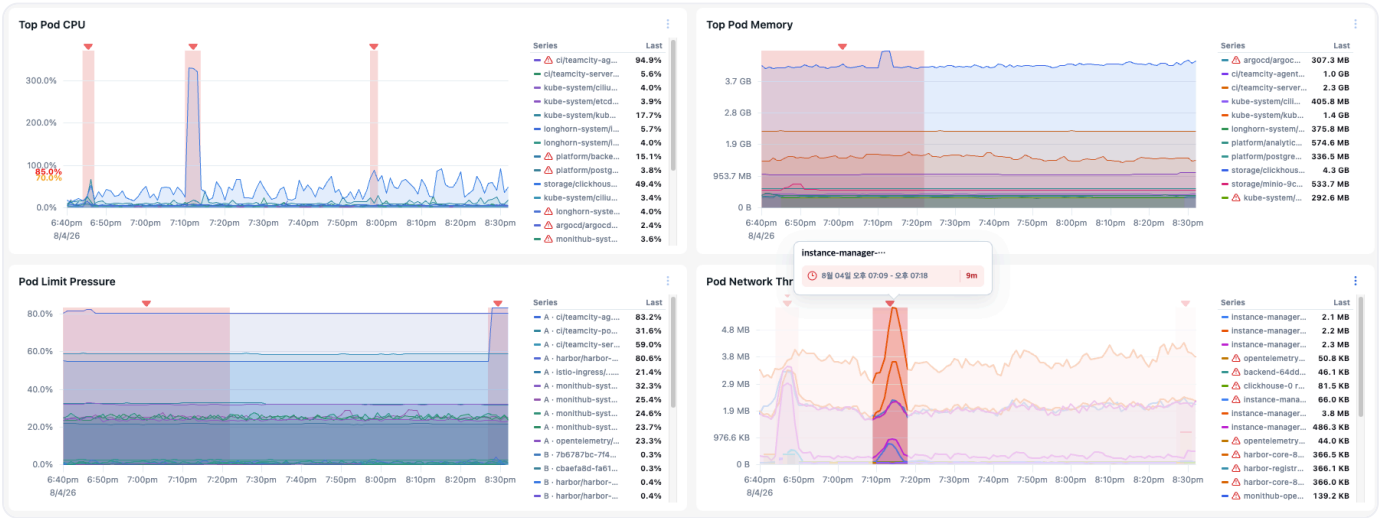
5.4. Sentinel: 이상탐지를 운영 경험에 맞게 관리합니다

Sentinel은 관측 데이터에서 이상 징후를 찾고 모델 버전을 관리하는 기능입니다. 운영자는 Sentinel이 제시한 탐지 결과를 실제 인시던트 해결 기록과 비교하고 라벨을 남긴 뒤, 이를 근거로 새 모델 후보를 학습·검토해 탐지 기준을 운영 환경에 맞게 다듬습니다.

모델 튜닝 화면에서는 활성·후보 버전, 다음 학습 시각과 라벨의 재학습 반영 여부를 한눈에 확인할 수 있습니다. 주기 분석 일정과 최근 실행 결과도 같은 화면에서 살펴보며 학습과 탐지가 계획대로 이어지는지 점검합니다. 탐지가 지나치게 민감하거나 중요한 변화를 놓친다면 운영자가 라벨과 해결 결과를 근거로 학습 조건과 분석 주기를 조정한 뒤 새 후보를 다시 검증합니다.



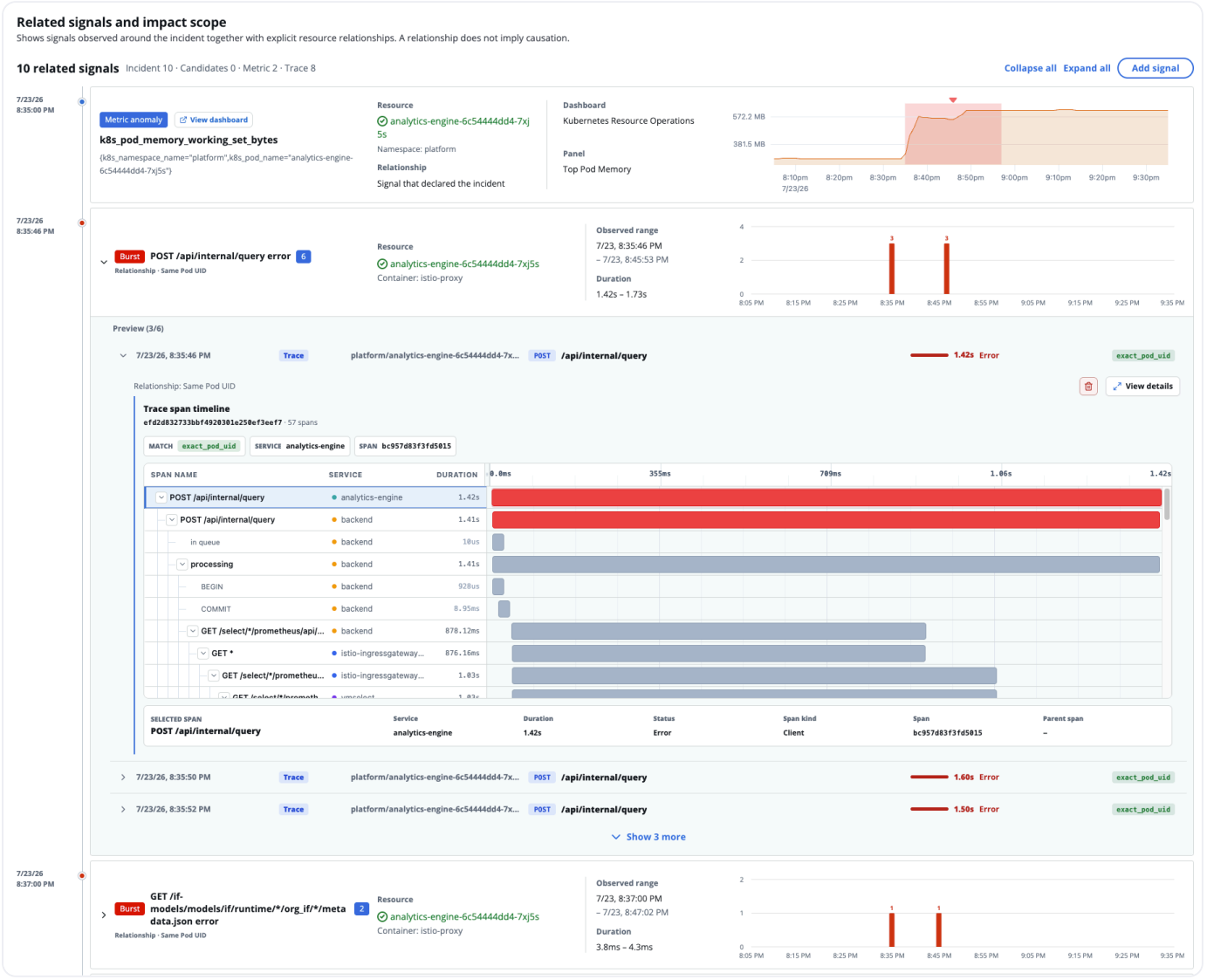
대시보드에서는 탐지된 이상 구간을 원본 시계열 위에 표시합니다. 구간을 선택하면 대상 시계열, 시작·종료 시각과 지속 시간을 확인할 수 있어 값의 변화와 탐지 결과를 같은 맥락에서 판단하기 쉽습니다.



[Monithub Sentinel 이해하고 사용하기 →]

5.5. 인시던트: 신호를 실제 문제의 맥락으로 정리합니다

인시던트에서는 문제의 상태와 심각도, 영향 대상, 담당자와 관련 근거를 함께 확인합니다. 목록에서 우선순위를 판단하고 상세 화면에서 시간 흐름과 텔레메트리 맥락을 살펴본 뒤, 필요하면 문제를 분리하거나 관련 항목을 함께 다룰 수 있습니다.



목표는 단순히 경고 개수를 줄이는 것이 아닙니다. 팀이 같은 근거를 바탕으로 **지금 대응해야 하는 문제인지, 누가 무엇을 확인할지, 언제 해결되었다고 볼지** 합의하고 그 판단과 결과를 다음 대응에 남기는 것이 중요합니다.

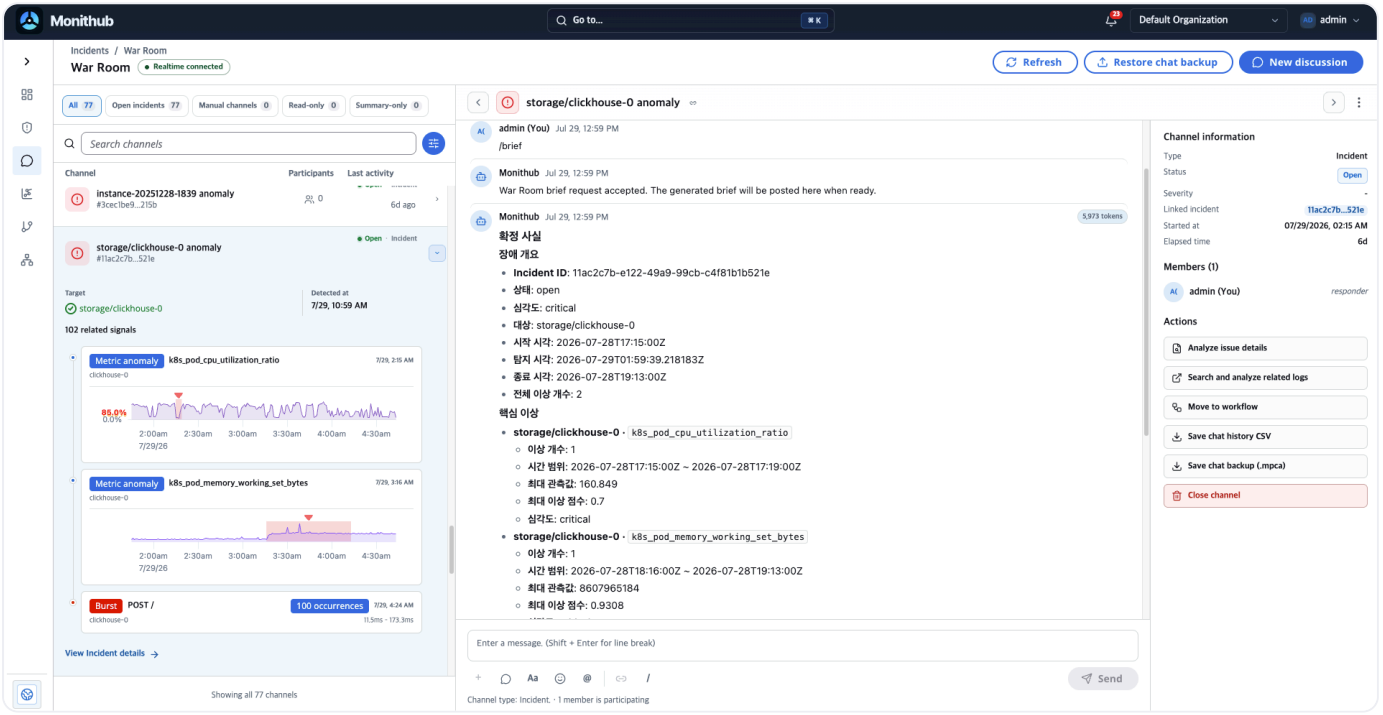
[인시던트와 워룸의 전체 흐름 알아보기 →]

5.6. 워룸과 Monithub AI: 대응 맥락을 공유합니다

워룸에서는 인시던트별로 대화를 나누고 조직 멤버를 부르며, 확인한 사실과 다음 행동을 기록합니다. 단순한 대화방이 아니라 담당자, 판단 근거, 조치와 결과를 인시던트에 연결해 대응 과정을 조직이 다시 활용할 수 있는 기록으로 남기는 공간입니다.

Monithub AI의 요약과 브리프는 새로 참여한 사람이 긴 대화를 모두 읽지 않고도 현재 상황을 파악하도록 돕습니다. 관찰된 문제, 지금까지 확인한 내용과 남은 질문을 정리한 뒤 실제 근거와 비교해 활용합니다.

대응이 끝난 뒤에도 이 기록을 통해 새 담당자는 당시 상황을 다시 설명받지 않고 원인과 해결 과정을 파악할 수 있습니다. 반복되는 문제에서는 과거에 효과가 있었던 조치와 남은 후속 작업을 바로 찾아볼 수 있습니다.

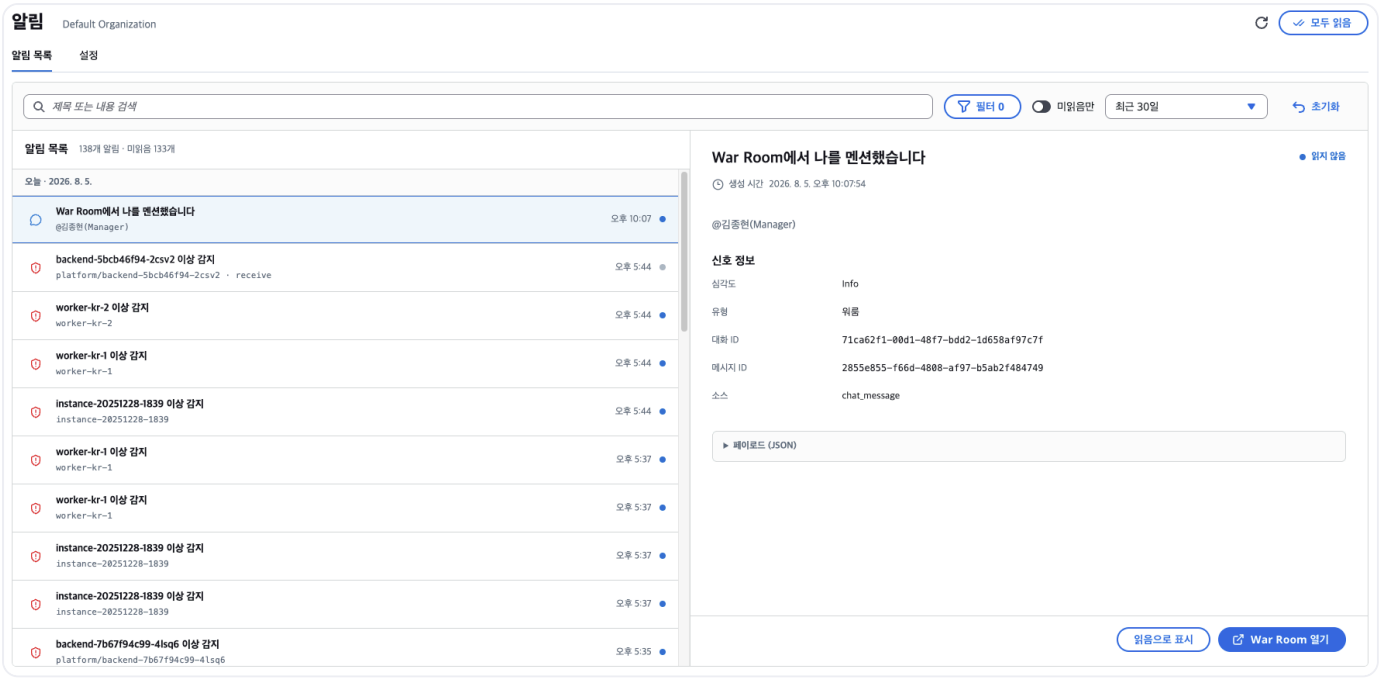


[워룸에서 협업하는 방법 알아보기 →]

5.7. 알림: 필요한 변화를 놓치지 않게 합니다

알림함은 새 인시던트, Connect 수집 상태와 워룸 메시지를 한곳에서 확인하는 기본 채널입니다. 읽음 상태, 유형, 심각도, 기간과 키워드로 항목을 좁혀 보고 해당 인시던트나 워룸으로 바로 이동할 수 있습니다.

사용 환경에 따라 Web Push, Telegram과 앱 알림음을 설정할 수 있습니다. 인시던트는 항상 받되, 워룸 메시지는 멘션만 받을지, 모든 메시지를 받을지, 받지 않을지 선택해 불필요한 알림을 줄일 수 있습니다.



- [알림 확인과 설정 →]
- [Telegram 연결과 전송 →]

5.8. Connect와 Monithub K8S Operator: 관측 기반을 마련합니다

Connect에서는 수집 환경과 대상을 구분해 연결 계획을 만듭니다. Bare Metal과 Kubernetes 중 환경을 선택하고, OpenTelemetry Collector, Linux, Docker, Java 애플리케이션이나 Custom OTLP 등 필요한 대상을 고를 수 있습니다.

Kubernetes 환경에서는 Monithub K8S Operator가 Collector와 Agent를 설치하고 관리합니다. 처음에는 수집 범위를 작게 설정하고 데이터가 정상적으로 들어오는지 확인한 뒤, 필요한 신호와 범위를 점진적으로 넓히는 것이 좋습니다.

- [Connect와 Gateway 화면 알아보기 →]
- [Monithub K8S Operator 설치하기 →]
- [Telemetry 필터링하기 →]

5.9. 조직과 설정: 운영 범위와 사용 환경을 관리합니다

Monithub의 데이터와 권한은 조직을 기준으로 나뉩니다. 사용자는 새 조직을 만들거나 기존 조직에 참여하고, 관리자는 멤버를 초대하거나 참여 요청을 처리합니다. 화면 상단에서 현재 조직을 바꾸면 대시보드, 인시던트, 워룸과 요금·구독의 데이터 범위도 함께 달라집니다.

Settings에서는 프로필, 인시던트, 데이터소스, 조직과 Sentinel 관련 설정을 관리합니다.

- **가입과 조직 참여 →**
- **멤버 초대와 참여 요청 →**
- **설정 →**

5.10. 요금과 구독: 사용 현황과 청구 상태를 관리합니다

요금과 구독에서는 현재 요금제와 갱신일, 사용량과 한도를 확인합니다. 운영자는 요금제와 부가 기능을 조정하고, 수동 청구 요약을 생성·확정·내보내거나 구독 상태를 관리할 수 있습니다.

- **요금과 구독 →**

6. 실제 상황에서는 어떻게 사용하나요?

6.1. 서비스 응답이 갑자기 느려진 경우

1. 홈에서 열린 인시던트와 영향 대상, 최근 활동을 확인합니다.
2. 관련 대시보드에서 시간 범위를 문제 발생 시점으로 맞추고 응답 시간, 오류와 리소스 변화를 비교합니다.
3. 서비스 맵에서 느린 서비스를 선택해 직접 연결된 upstream과 downstream 상태를 확인합니다.
4. 인시던트 상세에서 관련 메트릭, 로그, 트레이스와 Kubernetes 맥락을 함께 살펴봅니다.
5. 워룸을 열어 담당자를 부르고, 확인된 사실과 원인 가설, 다음 행동을 기록합니다.
6. 대응 대화가 길어지면 AI 요약이나 브리프로 현재 상황을 정리하고 실제 근거와 비교합니다.
7. 복구 후 원인, 영향 범위, 확인 근거, 조치 결과와 반복 방지 작업을 정리하고, 탐지 결과 라벨과 모델 학습·검토 필요 여부를 남깁니다.

이 흐름의 목적은 모든 분석을 자동화하는 것이 아니라, 사람이 판단하는 데 필요한 근거와 협업 맥락을 끊김 없이 연결하는 것입니다. 축적된 해결 기록은 다음 담당자와 다음 인시던트가 다시 사용할 수 있는 조직의 운영 자산이 됩니다.

6.2. 이상탐지 알림이 너무 자주 발생하는 경우

1. 이상 구간이 실제 운영 문제였는지 대시보드와 인시던트 근거로 확인합니다.
2. 정탐, 오탐 또는 무시됨 라벨을 남겨 판단 결과를 구분합니다.

3. 같은 유형의 오탐이 반복되는지 기간과 데이터 소스별로 살펴봅니다.
4. 반복된 오탐의 해결 기록과 라벨을 근거로 운영자가 Sentinel의 탐지 성향이나 학습 조건을 조정해 새 후보 버전을 학습합니다.
5. 후보의 품질과 소스 커버리지를 검토한 뒤, 기존 활성 버전보다 적합할 때만 승격합니다.
6. 변경 후 대시보드의 이상 구간과 새 인시던트 흐름을 관찰해 개선 여부를 확인합니다.

6.3. 새 Kubernetes 클러스터를 연결하는 경우

1. 올바른 조직을 선택하고 Connect에서 Kubernetes 연결 계획을 만듭니다.
2. 발급된 식별 정보와 토큰, OTLP endpoint와 registry 인증 정보를 안전하게 준비합니다.
3. 클러스터에 Monithub K8S Operator와 기본 수집 컴포넌트를 설치합니다.
4. Monithub K8S Operator, Collector와 Agent 상태가 정상인지 확인합니다.
5. Monithub 연결 목록과 서비스 맵에서 실제 데이터가 들어오는지 검증합니다.
6. 기본 수집이 안정된 뒤 필요한 telemetry preset과 필터를 조정합니다.

7. 사용할 때 기억할 점

- **현재 조직을 먼저 확인합니다.** 조직이 다르면 같은 메뉴에서도 보이는 데이터와 권한이 달라집니다.
- **시간 범위와 필터를 함께 확인합니다.** 데이터가 없거나 평소와 다르게 보일 때는 수집 문제로 단정하기 전에 조회 범위를 확인합니다.
- **이상 신호와 AI 결과는 판단을 돕는 근거입니다.** 실제 장애 여부와 실행할 조치는 시스템 상태와 팀의 운영 기준을 바탕으로 결정합니다.
- **대응 기록은 다음 대응에서 다시 쓸 수 있게 남깁니다.** 확인 근거, 판단 이유, 조치 결과와 후속 작업을 인시던트와 워룸의 같은 맥락에 정리하면 특정 사람의 기억에 의존하지 않고 운영 경험을 이어갈 수 있습니다.
- **보이는 기능은 권한과 연결 상태에 따라 달라질 수 있습니다.** 필요한 메뉴가 없거나 데이터를 볼 수 없다면 조직 관리자에게 권한과 데이터소스 상태를 확인합니다.
- **민감한 연결 정보는 안전하게 관리합니다.** Connect token과 registry token은 문서, 이슈나 메신저에 남기지 않고 Secret으로 주입합니다.

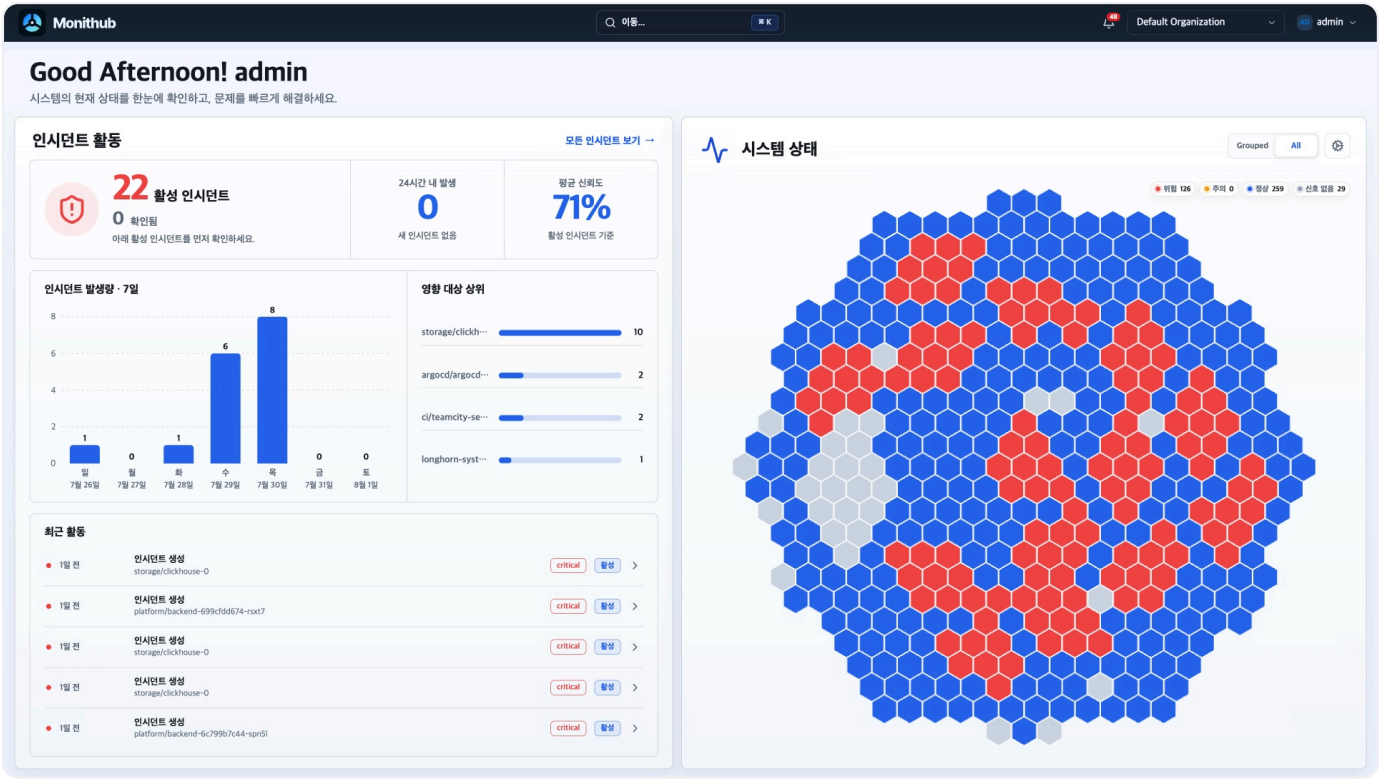
8. 역할에 맞게 시작하기

지금 하려는 일	추천 시작 문서
처음 가입하고 화면을 익히고 싶습니다.	**가입과 조직 참여** → **홈과 기본 탐색**
우리 시스템의 상태를 관측하고 싶습니다.	**대시보드** → **서비스 맵**
인시던트 대응 흐름을 준비하고 싶습니다.	**인시던트 및 워룸 개요** → **알림 확인과 설정**
이상탐지 모델을 운영하고 싶습니다.	**Monithub Sentinel** → **실습 가이드** → **학습과 튜닝**
Kubernetes 데이터를 처음 연결하고 싶습니다.	**Connect와 Gateway** → **Monithub K8S Operator 설치**
프로필, 데이터소스와 조직 설정을 관리하고 싶습니다.	**설정**
요금제, 사용량, 청구 요약과 구독 상태를 확인하고 싶습니다.	**요금과 구독**

제품의 역할과 전체 흐름을 이해했다면 **제품 사용 가이드**에서 필요한 작업을 골라 시작해 보세요.

제품 사용 가이드

이 가이드는 Monithub 화면에서 원하는 작업을 실제로 수행할 수 있도록 메뉴 위치, 확인할 내용과 실행 순서를 안내합니다. 제품의 목적과 기능이 서로 어떻게 이어지는지 먼저 알고 싶다면 **Monithub 소개**를 읽어 보세요.



기본 사용 흐름

1. 처음 사용하는 경우 **새 조직을 만들거나 기존 조직에 참여**합니다.
2. 로그인 후 상단에서 사용할 조직을 선택합니다.
3. 홈에서 현재 상태와 주요 메뉴를 확인합니다.
4. 대시보드에서 메트릭, 로그, 이벤트, 이상탐지 상태를 봅니다.
5. 필요한 경우 패널 편집 화면에서 쿼리, 표시 형식, 시각화 옵션, 레이아웃을 조정합니다.
6. 인시던트에서 문제 후보를 확인하고 상세 화면으로 들어갑니다.
7. 워룸에서 대응 대화, 분석 요청, 채팅 백업을 관리합니다.
8. **알림**에서 새 인시던트, Connect 수집 상태, 워룸 메시지를 확인하고 받을 방식을 설정합니다.
9. **Connect**에서 데이터 수집 연결과 최근 수집 상태를 확인합니다.
10. Monithub Sentinel에서 이상탐지 모델, 재학습, 라벨을 관리합니다.
11. **설정**에서 운영 환경, 데이터소스, 조직과 프로필을 관리합니다.

12. **요금과 구독**에서 요금제, 사용량, 청구 요약과 구독 상태를 관리합니다.

메뉴 한눈에 보기

메뉴	사용하는 때
계정과 조직	처음 가입하거나 기존 조직에 참여하고, 관리자가 멤버 초대와 참여 요청을 처리할 때
홈	로그인 직후 인시던트 활동 추이, 영향 대상, 최근 활동과 시스템 상태를 빠르게 볼 때
대시보드	Kubernetes, 로그, 트레이스, Gateway 등 운영 지표를 패널 단위로 보고 패널 편집으로 표시 방식을 조정할 때
인시던트	이상 신호가 묶인 문제 후보를 확인하고 담당자, 상태, 근거를 추적할 때
워룸	인시던트별 채팅방에서 대화, 요약, 백업, 분석 요청을 처리할 때
알림	새 인시던트, Connect 수집 상태, 워룸 메시지를 확인하고 Web Push, Telegram, 앱 알림음을 설정할 때
Connect	Collector, Java, Linux, Docker, Kubernetes, Custom OTLP 연결 계획을 만들 때
Gateway	Gateway 헬스, 서비스 레지스트리, 라우팅, 요청 테스트를 확인할 때
Service Map	서비스와 네트워크 edge 상태를 그래프로 볼 때
Monithub Sentinel	이상탐지 모델을 만들고 재학습, 후보 검토, 활성 승격, 라벨을 관리할 때
설정	데이터소스, 인시던트, 조직, Sentinel, 프로필과 화면 환경을 관리할 때
요금과 구독	현재 요금제, 갱신일, 사용량, 한도, 청구 요약과 구독 상태를 확인할 때

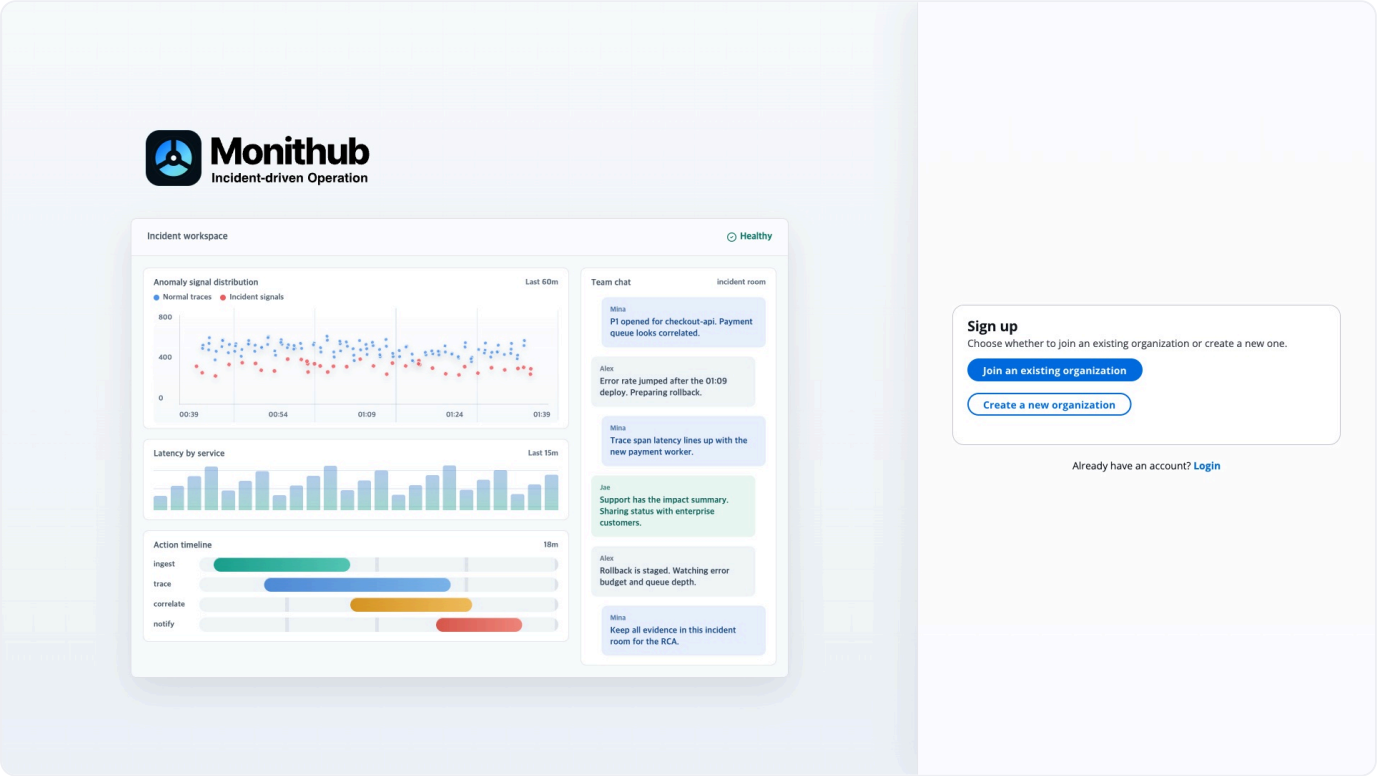
먼저 알아둘 것

- 상단의 조직 선택값에 따라 대시보드, 인시던트, 워룸, 요금과 구독 데이터가 달라집니다.
- `이동...` 또는 `⌘ K`를 사용하면 주요 메뉴와 대시보드를 빠르게 찾을 수 있습니다.
- 좌측 메뉴에서는 주요 기능과 조직의 대시보드를 바로 선택할 수 있습니다.
- 대부분의 화면에는 `Refresh` 또는 `새로고침` 버튼이 있어 최신 상태를 다시 불러올 수 있습니다.
- War Room을 사용하려면 Platform의 워룸 메뉴에서 원하는 방을 선택합니다.

가입과 조직 참여

Monithub에 처음 가입할 때는 새 조직을 만들지, 이미 만들어진 조직에 참여할지 먼저 선택합니다. 회사나 팀에서 초대 메일을 받았다면 메일의 링크로 시작하는 것이 가장 빠릅니다.

내 상황	선택할 항목
새 팀이나 프로젝트의 공간이 필요함	Create a new organization
관리자가 알려 준 조직에 참여하려고 함	Join an existing organization
조직 초대 메일을 이미 받음	메일의 초대 링크



새 조직을 만들면서 가입하기

1. 새 조직 만들기 선택

가입 화면에서 Create a new organization을 누릅니다. Create organization 화면이 열리면 조직과 계정 정보를 입력합니다.

입력 항목	입력할 내용
Organization name	화면에 표시할 팀이나 회사 이름
Organization slug	조직 주소에 사용할 고유한 이름
Email	인증 메일을 받을 이메일 주소
Display name	다른 멤버에게 표시할 이름. 비워 두면 이메일의 `@` 앞부분이 사용됩니다.
Password	8자 이상의 비밀번호
Confirm password	위에서 입력한 비밀번호

`Organization slug`에는 영문 소문자, 숫자, 하이픈(`-`)만 사용할 수 있습니다. 조직 이름을 입력하면 추천값이 자동으로 채워지며, 필요하면 직접 바꿀 수 있습니다.

2. 인증 메일 받기

입력을 마치면 `Send verification email`을 누릅니다. `Check your email` 화면이 보이면 입력한 이메일의 받은편지함을 확인합니다.

메일의 인증 링크를 열거나, 메일에 표시된 8자리 코드를 `Verification code`에 입력한 뒤 `Verify code`를 누릅니다.

- 메일이 오지 않으면 스팸함과 입력한 이메일 주소를 확인한 뒤 `Resend email`을 누릅니다.
- 이메일이나 조직 정보를 고쳐야 하면 `Edit details`를 누릅니다.
- 이미 사용 중인 `Organization slug`라는 안내가 나오면 다른 값을 입력합니다.

인증이 끝나고 새 조직의 대시보드가 열리면 가입과 조직 생성이 완료된 것입니다.

기존 조직에 참여 요청하기

조직 초대 링크가 없지만 관리자가 조직의 `Organization slug`를 알려 준 경우 다음 순서로 참여를 요청합니다.

1. 가입 화면에서 `Join an existing organization`을 누릅니다.
2. `Organization slug`에 관리자가 알려 준 값을 입력합니다.
3. `Email`에 초대받을 이메일 주소를 입력합니다.

4. `Send invitation email`을 누릅니다.
5. 다음 단계가 메일로 안내된다는 메시지가 보이면 받은편지함을 확인합니다.

화면에는 조직이나 요청 상태를 구분하지 않고 같은 안내가 표시됩니다. 초대 메일이 바로 오지 않으면 조직 관리자에게 참여 요청을 확인해 달라고 요청하세요. 관리자가 승인하면 초대 메일이 발송됩니다.

> **Note**
>
> 전에 받은 초대 메일을 잃어버렸다면 `Organization slug`를 비워 두고 초대받은 이메일만 입력해 다시 요청할 수 있습니다.

초대 메일로 참여하기

초대 메일의 링크를 열면 `Organization 초대` 화면에서 조직 이름, 초대받은 이메일, 초대 만료일을 확인할 수 있습니다.

Monithub 계정이 없는 경우

1. `가입해서 수락`을 누릅니다.
2. 초대받은 이메일이 맞는지 확인합니다. 이 이메일은 다른 주소로 바꿀 수 없습니다.
3. `Display name`, `Password`, `Confirm password`를 입력합니다.
4. `Send verification email`을 누릅니다.
5. 메일의 링크를 열거나 8자리 코드를 입력해 이메일 인증을 마칩니다.
6. 초대받은 조직의 대시보드가 열렸는지 확인합니다.

Monithub 계정이 있는 경우

1. `로그인하여 수락`을 누릅니다.
2. 초대받은 이메일과 같은 계정으로 로그인합니다.
3. 초대 화면으로 돌아오면 `초대 수락`을 누릅니다.
4. “초대를 수락했습니다”라는 메시지가 보이면 `대시보드로 이동`을 누릅니다.

> **⚠ Caution**
>
> 다른 이메일의 계정으로 로그인하면 초대를 수락할 수 없습니다. 로그아웃한 뒤 초대 메일을 받은 계정으로 다시 로그인하세요.

초대가 진행되지 않을 때

화면 또는 상황	다음에 할 일
인증 메일이나 초대 메일이 보이지 않음	스팸함을 확인하고 이메일 주소가 맞는지 확인합니다. 계속 오지 않으면 조직 관리자에게 새 초대를 요청합니다.
`만료된 초대입니다.` 또는 `취소된 초대입니다.`	기존 링크 대신 조직 관리자에게 새 초대 메일을 요청합니다.
`이미 수락된 초대입니다.`	`대시보드로 이동`을 눌러 조직 화면을 확인합니다.
`이미 이 organization의 멤버입니다.`	`대시보드로 이동`을 누릅니다.
“이 초대는 다른 이메일 주소로 발송되었습니다.”	현재 계정에서 로그아웃하고 초대받은 이메일의 계정으로 다시 로그인합니다.

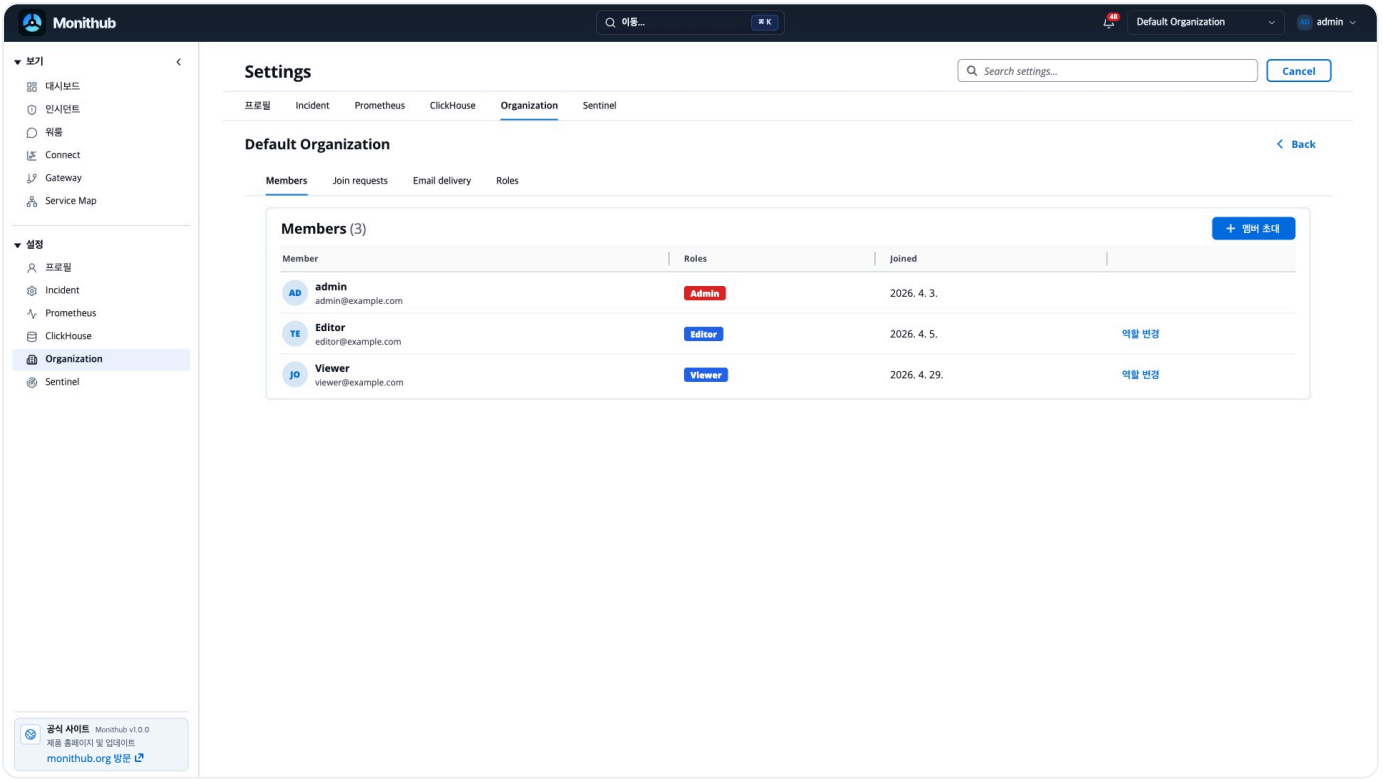
조직 관리자가 멤버를 직접 초대하거나 참여 요청을 처리하는 방법은 **멤버 초대와 참여 요청**에서 확인할 수 있습니다.

멤버 초대와 참여 요청

조직 관리자는 이메일로 멤버를 직접 초대하거나, 가입 화면에서 들어온 참여 요청을 확인할 수 있습니다. 두 방법 모두 상대방이 초대 메일을 열고 수락해야 조직 참여가 완료됩니다.

조직의 멤버 화면 열기

1. Monithub에 로그인한 뒤 우측 상단의 사용자 메뉴를 엽니다.
2. `Settings`를 누릅니다.
3. 설정 메뉴에서 `Organization`을 선택합니다.
4. `My Organizations`에서 관리할 조직의 `Manage →`를 누릅니다.
5. `Members` 탭이 선택되어 있는지 확인합니다.



이메일로 멤버 초대하기

1. `Members` 탭에서 `멤버 초대`를 누릅니다.
2. `Email address`에 초대할 사람의 이메일을 입력합니다.
3. `Send Invitation`을 누릅니다.

4. `Invitation Sent` 창에 입력한 이메일과 “초대 메일을 발송했습니다”라는 메시지가 보이는지 확인합니다.
5. `Done`을 눌러 창을 닫습니다.

초대를 보낸 직후에는 상대방이 멤버 목록에 나타나지 않습니다. 상대방이 메일의 링크에서 초대를 수락한 뒤 `Members` 목록을 다시 확인하세요.

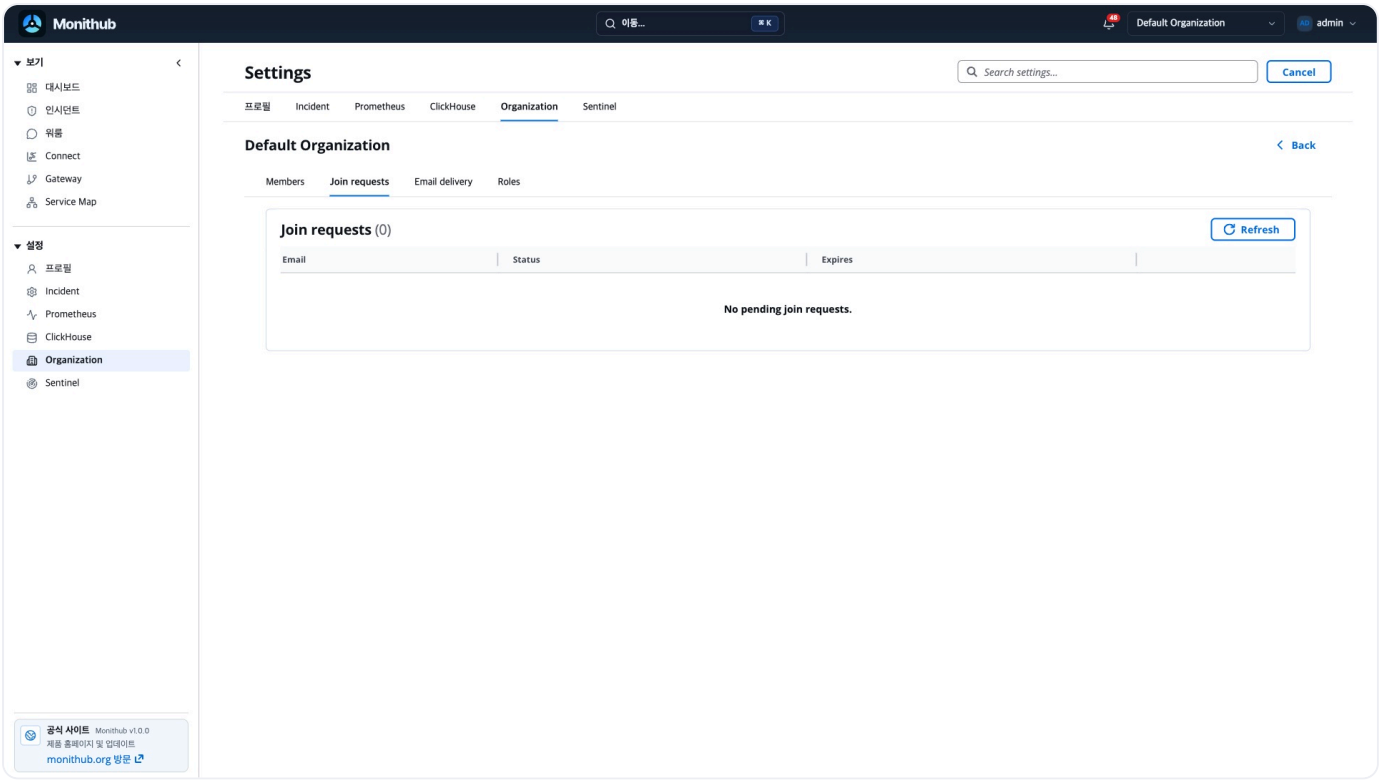
> **Note**
>
> “이미 대기 중인 초대가 있는 이메일입니다”라는 안내가 나오면 같은 주소로 새 초대를 만들 수 없습니다. 초대받은 사람은 가입 화면에서 `Join an existing organization`을 선택하고 이메일을 입력해 다음 단계 메일을 다시 요청할 수 있습니다.

초대받은 사람이 계정 유무에 따라 진행하는 방법은 **가입과 조직 참여**에서 확인할 수 있습니다.

참여 요청 확인하기

가입 화면에서 조직의 `Organization slug`와 이메일을 입력한 사용자의 요청은 관리자의 `Join requests` 탭에 표시됩니다.

1. `Settings → Organization`에서 관리할 조직의 `Manage →`를 누릅니다.
2. `Join requests` 탭을 누릅니다.
3. 요청한 이메일, 요청 시각, 상태, 만료 시각을 확인합니다.
4. 조직에 참여해도 되는 사용자라면 `Approve`, 그렇지 않으면 `Reject`를 누릅니다.



승인한 경우

“Invitation email sent”라는 메시지가 보이고 해당 요청이 목록에서 사라지면 처리가 완료된 것입니다. 승인된 이메일로 초대 메일이 발송됩니다.

승인은 멤버 추가를 즉시 완료하지 않습니다. 요청한 사람이 메일의 초대 링크를 열어 로그인 또는 가입한 뒤 `초대 수락`까지 마쳐야 `Members` 목록에 나타납니다.

거절한 경우

거절되었다는 메시지가 보이고 해당 요청이 목록에서 사라지는지 확인합니다. 잘못 거절했다면 사용자가 가입 화면에서 다시 참여를 요청해야 합니다.

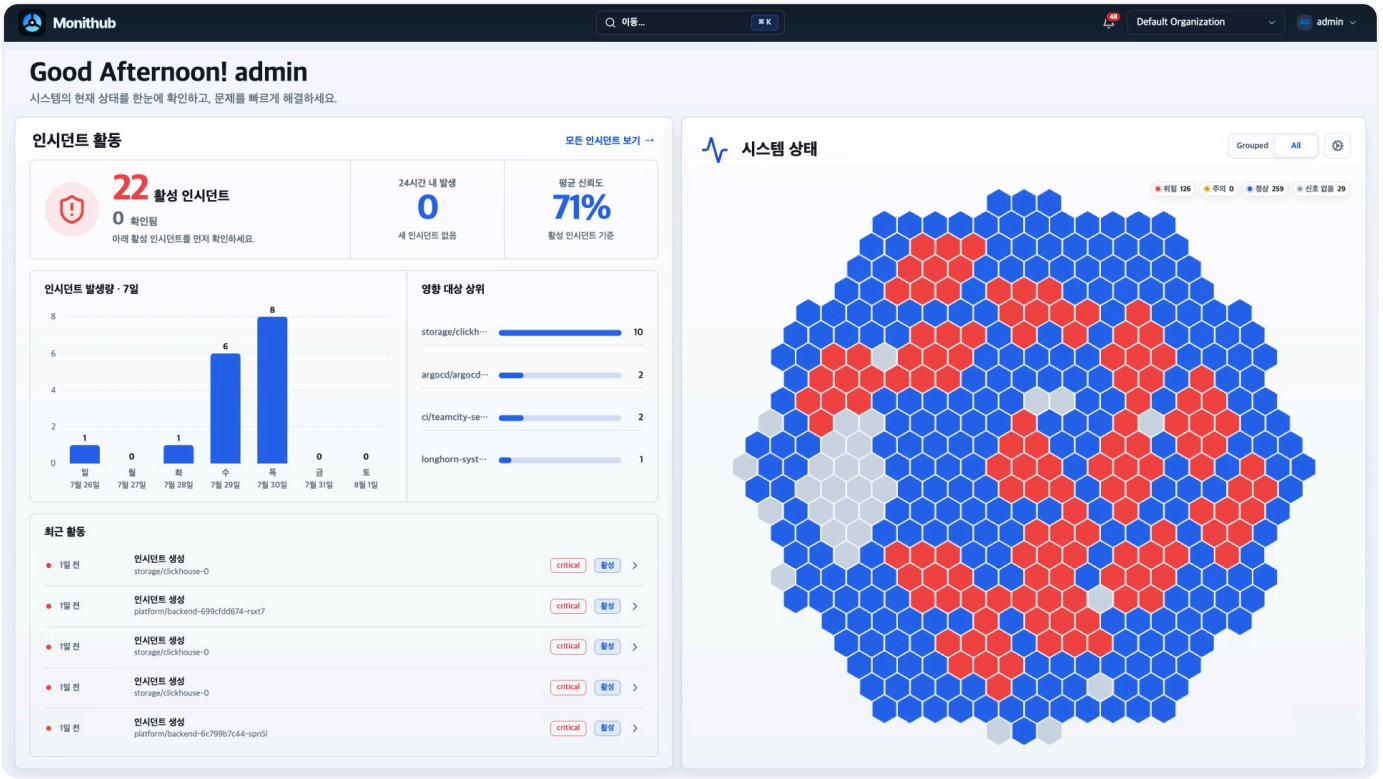
요청이 보이지 않거나 처리할 수 없을 때

화면 또는 상황	다음에 할 일
`Join requests` 탭이 보이지 않음	조직 관리자 계정으로 로그인했는지 확인합니다.
`No pending join requests.`가 표시됨	대기 중인 요청이 없습니다. 요청자에게 `Organization slug`와 이메일을 확인한 뒤 다시 요청해 달라고 안내합니다.
승인 후에도 멤버 목록에 사용자가 없음	요청자에게 초대 메일을 열고 수락까지 마쳤는지 확인해 달라고 안내합니다.
초대 메일이 도착하지 않음	요청자에게 스팸함과 입력한 이메일 주소를 확인하도록 안내합니다. 필요하면 가입 화면에서 다음 단계 메일을 다시 요청합니다.

처리가 끝나면 `Members` 탭으로 돌아가 새 멤버의 이름과 이메일이 목록에 표시되는지 확인합니다.

홈과 기본 탐색

홈은 Monithub에 들어온 직후 가장 먼저 보는 운영 콘솔입니다. 현재 조직에서 발생한 인시던트 활동과 리소스 상태를 한 화면에서 요약합니다.



상단 바

영역	설명
빠른 이동	이동...을 누르거나 ⌘ K를 입력해 메뉴와 대시보드를 찾습니다.
알림	읽지 않은 알림 수를 표시합니다.
조직 선택	현재 보고 있는 조직을 바꿉니다. 조직을 바꾸면 주요 데이터 범위도 바뀝니다.
사용자 메뉴	로그인 사용자와 계정 메뉴를 확인합니다.

인시던트 활동

인시던트 활동은 현재 열린 인시던트와 최근 변화량을 빠르게 판단하는 영역입니다.

영역	확인할 내용
Active incidents	현재 해결되지 않은 인시던트 수를 확인합니다.
Acknowledged	담당자가 확인한 인시던트 수를 확인합니다.
Past 24h	최근 24시간에 감지된 활동량을 봅니다.
Avg confidence	묶인 이상 신호의 평균 신뢰도를 참고합니다.
인시던트 발생량 · 7일	최근 7일 동안 인시던트가 증가하거나 감소한 시점을 봅니다.
영향 대상 상위	반복해서 영향을 받은 리소스를 우선 확인합니다.
최근 활동	상태 변경, 담당자 지정 등 최근 대응 이력을 확인합니다.

시스템 상태

`시스템 상태`는 조직의 리소스를 계층형 도형으로 묶어 보여 줍니다. 색상은 `위험`, `주의`, `정상`, `신호 없음` 상태를 나타냅니다.

- `Grouped`는 리소스를 그룹별로 묶어 보고, `All`은 전체를 한 번에 펼쳐 봅니다.
- 중앙 또는 바깥쪽 조각을 선택해 해당 그룹과 하위 리소스 상태를 좁혀 봅니다.
- `Options`에서 표시 기준을 조정한 뒤 상태 분포를 다시 확인합니다.

좌측 메뉴

좌측 아이콘 메뉴는 모든 주요 화면으로 이동하는 기본 경로입니다.

1. 메뉴가 접혀 있으면 상단의 펼치기 버튼을 누릅니다.
2. `대시보드`, `인시던트`, `워룸`, `Connect`, `Gateway`, `Service Map` 중 필요한 화면을 선택합니다.
3. 대시보드 화면에서는 좌측 목록에서 원하는 대시보드도 바로 선택할 수 있습니다.

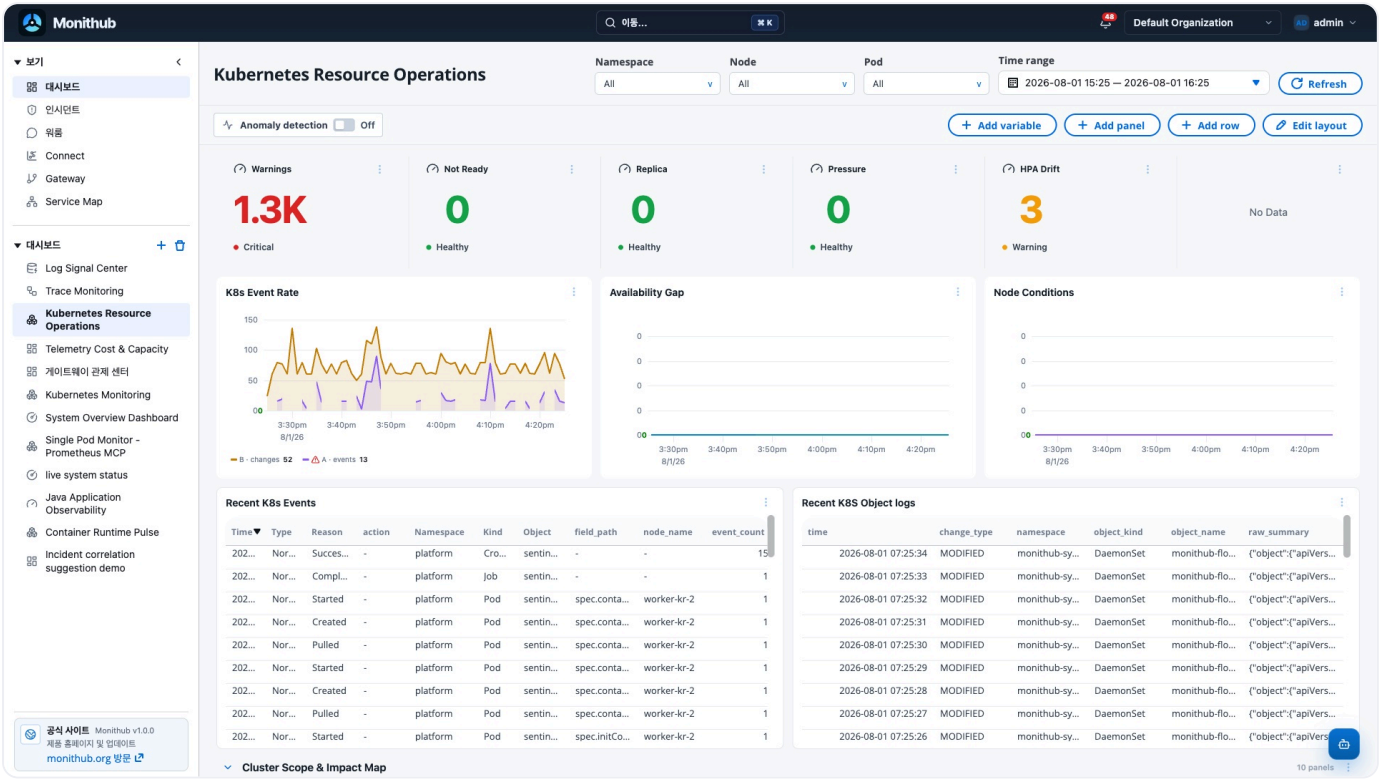
홈에서 바로 판단할 것

- 7일 추이에서 인시던트가 급증한 구간이 있는지 확인합니다.
- `영향 대상 상위`에 중요한 리소스가 반복해서 등장하는지 봅니다.

- 시스템 상태에 `위험` 또는 `주의` 조각이 있는지 확인합니다.
- 최근 활동에서 아직 담당자가 없거나 장시간 이어진 인시던트를 찾습니다.

대시보드

대시보드는 운영 지표를 패널 단위로 모아 보는 화면입니다. 왼쪽 목록에서 대시보드를 선택하고 상단 변수와 시간 범위로 필요한 데이터만 좁혀 봅니다.



대시보드 선택

좌측 대시보드 목록에서 원하는 대시보드를 선택합니다. 확인한 화면에는 아래 대시보드가 보였습니다.

- Kubernetes Resource Operations
- Trace Monitoring
- Log Signal Center
- Telemetry Cost & Capacity
- Single Pod Monitor - Prometheus MCP
- 게이트웨이 관제 센터
- System Overview Dashboard
- Container Runtime Pulse
- Java Application Observability
- live system status
- Incident correlation suggestion demo

필터와 시간 범위

대시보드 상단 필터에서 데이터를 좁힙니다.

필터	설명
Namespace	Kubernetes namespace 기준으로 패널 데이터를 좁힙니다.
Node	특정 노드만 봅니다.
Pod	특정 Pod만 봅니다.
Time range	조회 시간을 바꿉니다.
Refresh	현재 필터와 시간 범위로 데이터를 다시 불러옵니다.

패널 읽기

대시보드는 row와 panel로 구성됩니다.

- 큰 숫자 패널은 현재 상태를 즉시 판단하는 데 사용합니다.
- 차트 패널은 시간 흐름에 따른 변화와 급증 구간을 봅니다.
- 테이블 패널은 이벤트, 로그, object 변경 내역처럼 행 단위 근거를 확인합니다.
- 패널 우측 메뉴에서 세부 동작을 열 수 있습니다.

이상탐지

`Anomaly detection` 토글이 켜져 있으면 지원하는 패널에 이상탐지 결과가 겹쳐 표시됩니다. 결과가 갱신 중이면 현재 시간 범위와 변수를 유지한 채 잠시 뒤 `Refresh`로 다시 확인합니다.

편집 기능

권한이 있으면 상단 버튼으로 대시보드를 수정할 수 있습니다.

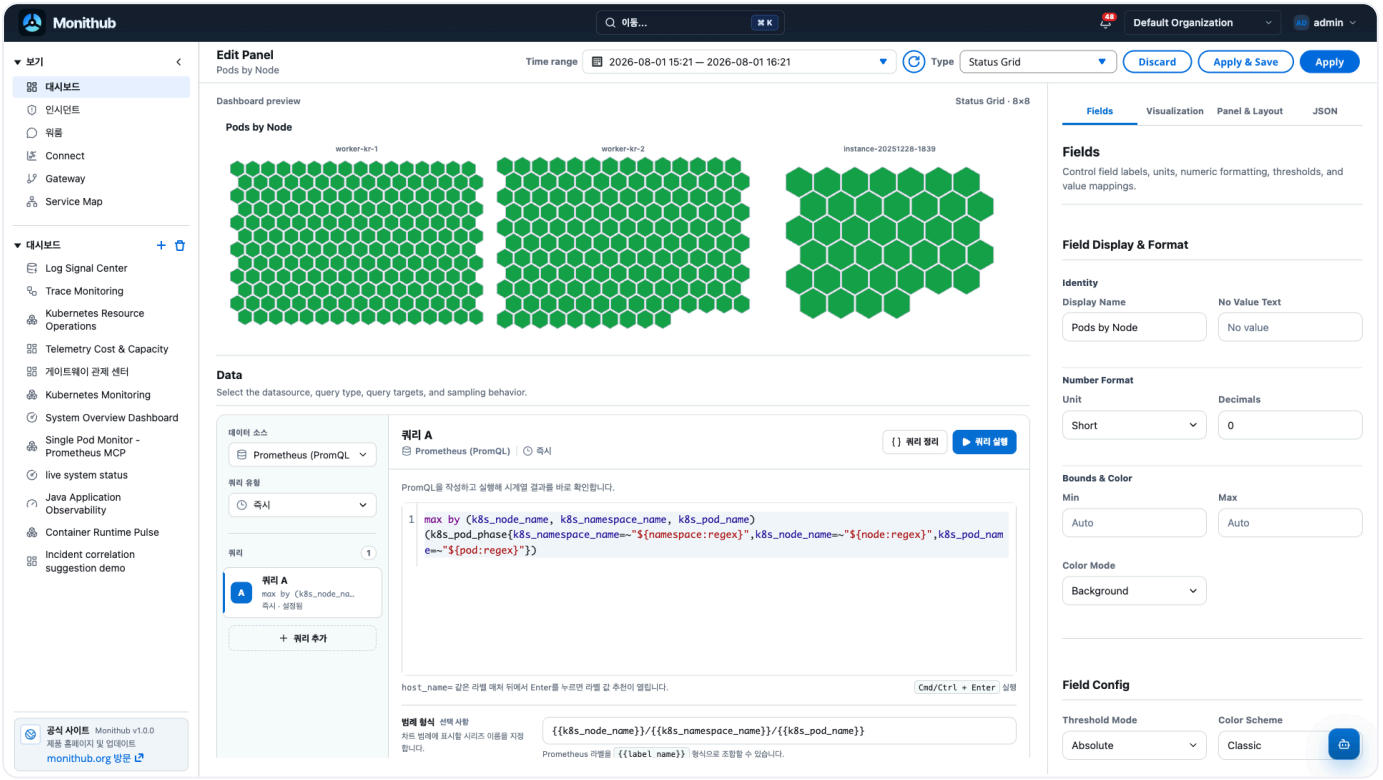
버튼	용도
Add variable	Namespace, Node, Pod 같은 재사용 필터를 추가합니다.
Add panel	새 차트, 숫자, 테이블 패널을 추가합니다.
Add row	패널 묶음을 추가합니다.
Edit layout	패널 배치를 편집합니다.

패널 타입별 쿼리, 표시 형식, 시각화 옵션은 **패널 편집** 문서에서 확인합니다.

운영 중인 대시보드를 수정할 때는 먼저 현재 필터와 시간 범위가 맞는지 확인한 뒤 저장합니다.

패널 편집

패널 편집 화면은 대시보드 패널의 데이터 쿼리, 값 표시 형식, 시각화 옵션, 레이아웃, JSON 설정을 한 곳에서 조정하는 화면입니다. 타입별 조작법은 하위 문서로 분리했고, 예시는 현재 조직의 Kubernetes, 로그, 트레이스 대시보드 패널을 기준으로 캡처했습니다.



진입 방법

대시보드에서 기존 패널 우측 상단의 Panel actions 메뉴를 열고 Edit을 선택합니다. 새 패널을 만들 때는 대시보드 상단의 Add panel을 사용합니다. 패널 묶음은 Add row로 만들며, row는 일반 지표 패널과 달리 데이터 쿼리보다 그룹 접기와 배치가 중심입니다.

편집 화면은 크게 다섯 영역으로 나뉩니다.

영역	역할
상단 바	패널 제목, 편집용 시간 범위, 새로고침, 패널 타입, 저장/취소 버튼을 제공합니다.
미리보기	현재 draft 설정으로 패널이 어떻게 보이는지 확인합니다.
Data	datasource, query type, query target, query 실행 테스트를 조작합니다.
오른쪽 탭	`Fields`, `Visualization`, `Panel & Layout`, `JSON` 설정을 전환합니다.
저장 버튼	`Apply`, `Apply & Save`, `Discard`로 draft 반영 범위를 정합니다.

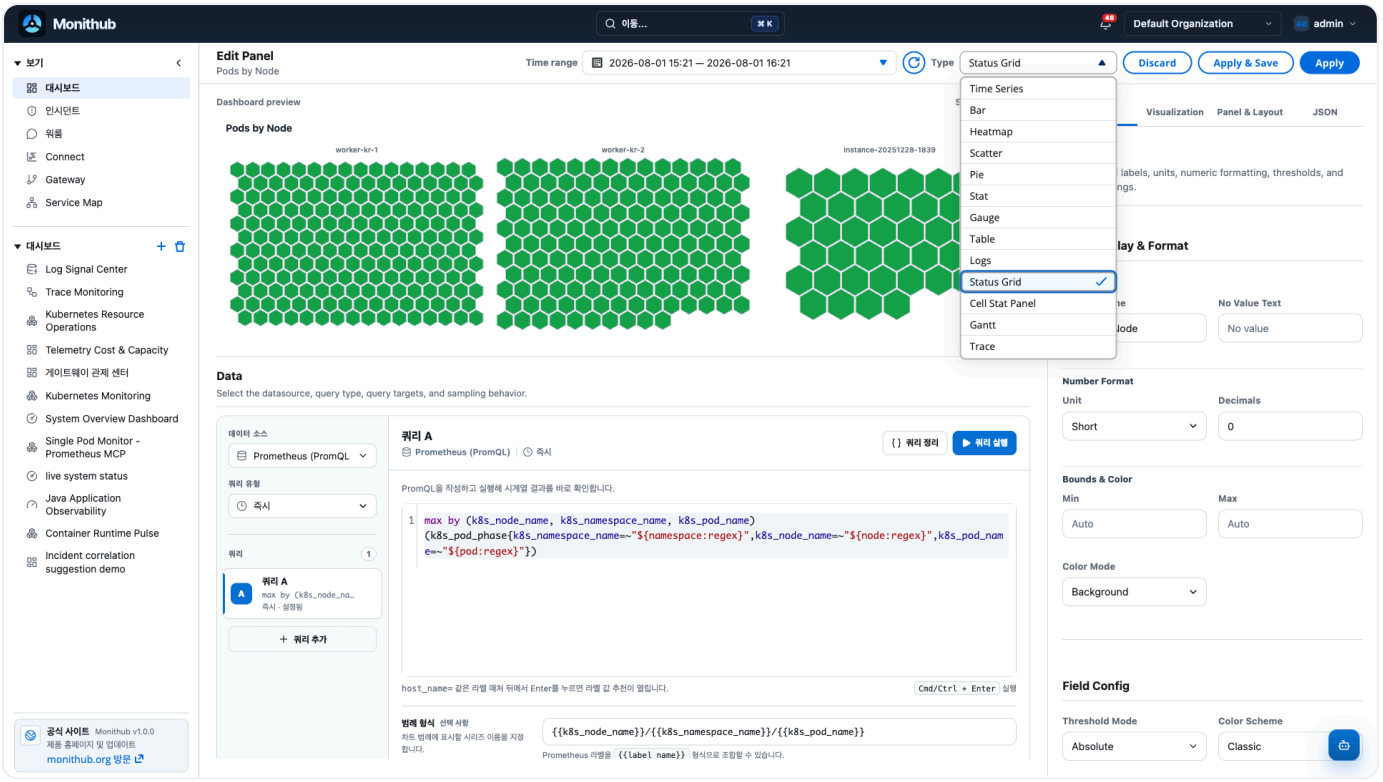
저장과 취소

버튼	동작
Discard	현재 편집 draft를 버리고 편집 화면을 닫습니다.
Apply	편집 화면의 draft를 현재 대시보드 화면에 반영합니다. 대시보드 저장까지 필요한 화면에서는 별도 저장 흐름을 확인합니다.
Apply & Save	draft를 적용하고 backend 저장까지 요청합니다.
Refresh	현재 편집용 시간 범위로 미리보기와 쿼리 결과를 다시 불러옵니다. 패널 정의 자체를 저장하지는 않습니다.

패널 타입을 바꾸면 같은 패널 id 안에서 모델이 새 타입에 맞게 정규화됩니다. 예를 들어 `Trace` 타입은 ClickHouse 계열 데이터에 맞게 datasource가 바뀔 수 있습니다. 타입 변경도 draft이므로 저장 버튼을 누르기 전까지 운영 대시보드에 확정되지 않습니다.

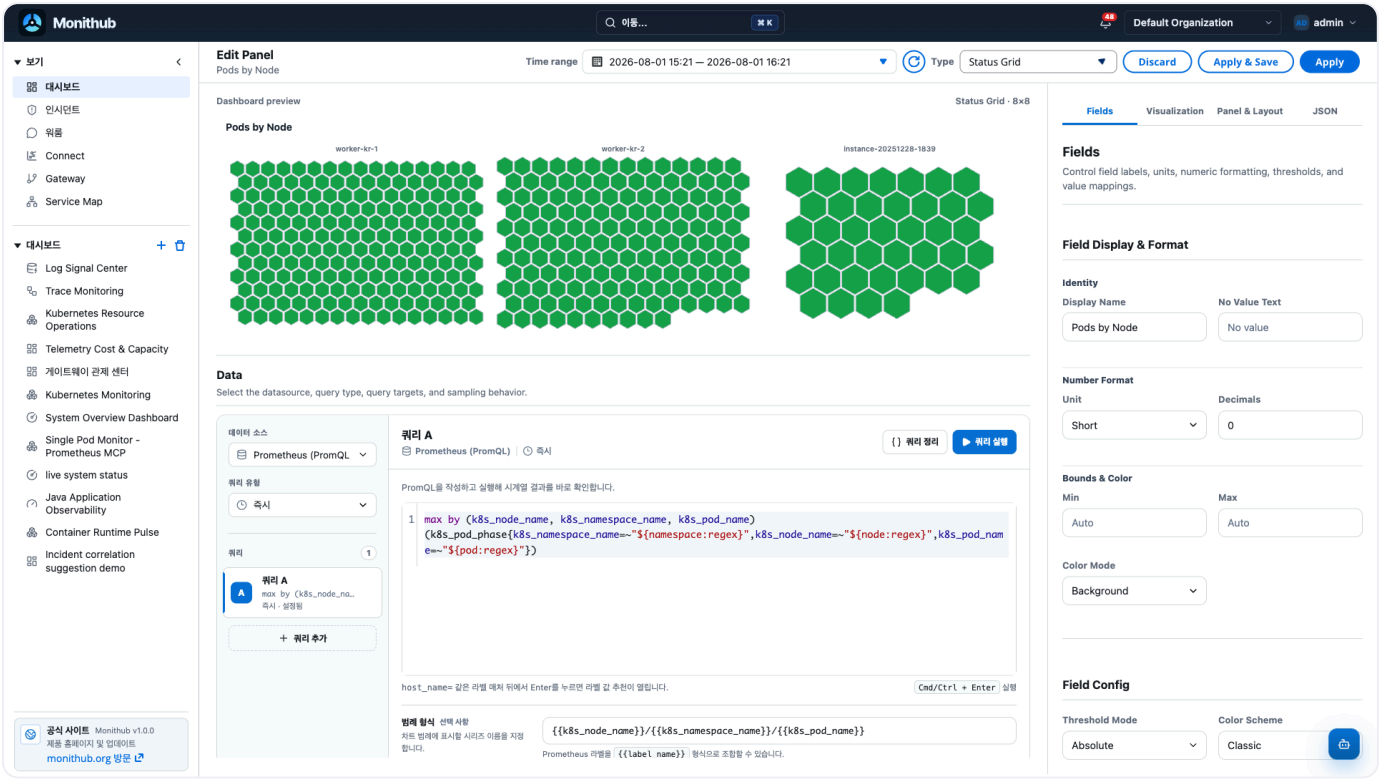
타입 선택

상단 `Type` 선택에서 패널 종류를 바꿉니다. 타입을 바꾸면 오른쪽 `Visualization` 탭의 입력 항목이 타입에 맞게 달라집니다.



타입	주 사용 상황
Time Series	시간 흐름에 따른 메트릭 변화, 이상탐지 overlay, series 비교를 볼 때
Bar	Time Series 계열 데이터를 막대 스타일로 비교할 때
Heatmap	시간과 bucket, label, value가 있는 밀도/분포 데이터를 볼 때
Scatter	trace duration처럼 행 단위 관측치를 x/y 점으로 볼 때
Pie	적은 수의 category 비중을 볼 때
Stat	하나 또는 여러 series의 최신값, 합계, 평균 같은 대표값을 볼 때
Gauge	단일 값이 기준 범위 안에서 어디에 있는지 볼 때
Table	로그, 이벤트, trace, ClickHouse 결과처럼 행과 열을 자세히 볼 때
Logs	로그성 텍스트 결과를 표시할 때
Status Grid	여러 대상의 정상/비정상 상태를 타일로 볼 때
Cell Stat Panel	작은 셀 단위로 여러 상태 값을 요약해 비교할 때
Gantt	시작/종료/기간이 있는 작업, span, 배치 흐름을 시간 막대로 볼 때
Trace	trace 목록과 trace 요약을 볼 때

Data 탭



Data 영역은 모든 데이터 패널의 출발점입니다. 먼저 datasource와 query type을 정한 뒤, query를 실행해서 반환 형태가 패널 타입과 맞는지 확인합니다.

설정	역할
Queries	패널이 사용하는 query target 목록입니다. A, B 같은 refId로 구분합니다.
Datasource	Prometheus (PromQL) 또는 ClickHouse (SQL)을 선택합니다. 상단 datasource를 바꾸면 현재 query target에도 같은 datasource가 적용됩니다.
Query type	Prometheus에서는 Instant와 Range를 고릅니다. ClickHouse는 table query로 동작합니다.
Add Query	같은 패널에 query target을 추가합니다. Time Series, Table, Status Grid처럼 여러 query를 비교하거나 join할 때 사용합니다.
Query editor	PromQL 또는 SQL을 작성합니다. 대시보드 변수와 시간 매크로를 함께 사용할 수 있습니다.
Format / snippet 버튼	PromQL/SQL 포맷 정리, trace query template, group by, time filter, limit snippet을 넣습니다.
Run Query	저장하지 않고 현재 query를 실행해 결과 형태를 확인합니다.

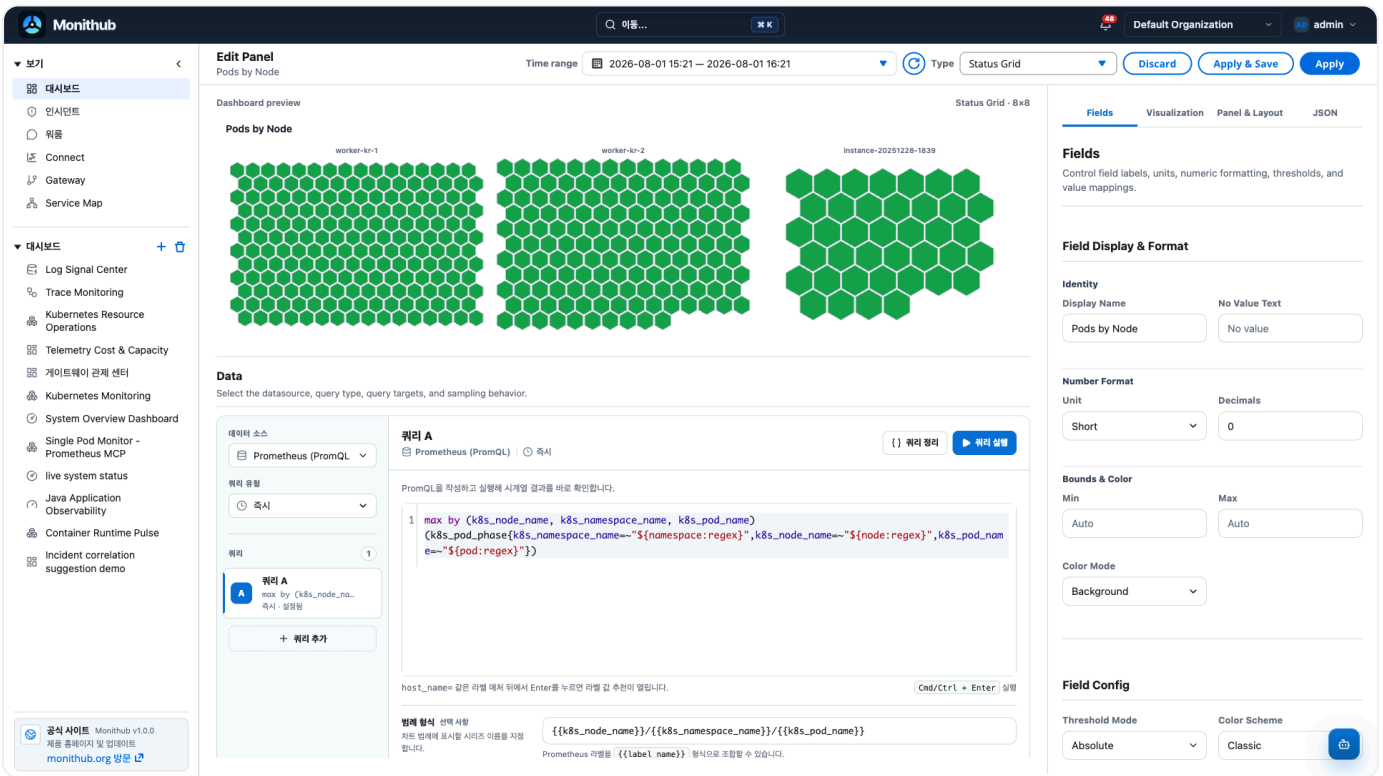
ClickHouse 쿼리는 `"$__fromTime"`, `"$__toTime"` 매크로를 지원합니다. 이 값은 편집 화면 상단의 `Time range` 기준으로 치환됩니다. Prometheus range query는 시간 범위와 start/end가 중요하고, instant query는 특정 시점의 대표값을 가져옵니다.

Table 타입에서 query가 둘 이상이면 join 설정이 나타납니다.

Join 설정	설명
Enable join across queries	여러 query 결과를 하나의 table로 합칩니다.
Join Queries	join에 포함할 refId를 고릅니다. 선택하지 않으면 전체 query가 대상입니다.
Join Key Field	여러 결과를 같은 행으로 묶을 기준 field입니다.
Join Type	inner, left, outer join 중 의도에 맞는 방식을 고릅니다.

중복 설정을 피하려면 datasource는 가능한 한 query target에서 한 번에 맞춥니다. 패널 타입 전환 후 datasource가 자동 변경된 경우, query editor와 datasource 표시가 서로 맞는지 먼저 확인합니다.

Fields 탭



`Fields`는 값이 화면에 표시되는 방식을 정합니다. 타입별 시각화가 달라도 단위, 소수점, min/max, threshold, value mapping은 대부분 공통으로 영향을 줍니다.

Field Display & Format

설정	역할
Display Name	legend, tooltip, stat title처럼 field 이름을 사람이 읽기 쉬운 이름으로 바꿉니다.
No Value Text	값이 없거나 null일 때 표시할 문구입니다. 기본값은 `No value`입니다.
Unit	숫자 단위를 지정합니다. `percent`, `bytes`, `ms`, `short` 등으로 값 formatter가 달라집니다.
Decimals	소수점 자리수를 고정합니다. 비워두면 자동으로 정합니다.
Min / Max	값 범위를 정합니다. Gauge, axis, threshold 색상 판단의 기준과 함께 쓰입니다.
Color Mode	값 자체 색상, 배경 색상, 색상 미사용 여부를 정합니다.

`Unit`, `Decimals`, `Min`, `Max`, `Display Name`, `No Value Text`는 내부적으로 `fieldConfig.defaults`에도 같이 반영됩니다. 같은 의미의 값을 JSON에서 따로 바꾸면 UI 값과 JSON 값이 어긋날 수 있으므로, 일반 편집에서는 Fields 탭을 기준으로 관리합니다.

Field Config

설정	역할
Threshold Mode	threshold step 값을 절대값으로 볼지 percentage로 볼지 정합니다.
Color Scheme	threshold나 value 색상 기본 팔레트를 정합니다.
Series Palette	Time Series/Bar 계열에서 여러 series 색상 순서를 정합니다.
Threshold Steps	값 범위별 색상을 순서대로 정의합니다. 각 step은 해당 값 이상부터 적용됩니다.
Value Mappings	특정 값 또는 범위를 텍스트와 색상으로 바꿉니다. 상태 코드, boolean, sentinel 상태를 사람이 읽는 문구로 바꿀 때 사용합니다.

중복 관계는 아래 원칙으로 정리합니다.

상황	권장 기준
Field Unit과 Axis Unit이 모두 있는 경우	값 표시와 tooltip은 Field Unit, 축 눈금은 Axis Unit을 따릅니다. 특별한 이유가 없으면 같은 단위로 맞춥니다.
Field Min/Max와 Axis Min/Max가 모두 있는 경우	Axis 입력은 chart 축에 직접 영향을 주며, 현재 UI는 min/max/decimals를 Field/Standard 값에도 동기화합니다. 한 곳에서 바꾼 뒤 다른 탭에서 값이 맞는지 확인합니다.
Global threshold와 Table column threshold가 모두 있는 경우	Table column threshold가 해당 column 표시를 더 구체적으로 제어합니다. 공통 기준은 Fields, 특정 column 예외는 Table column에서 둡니다.
Value Mapping과 Threshold가 모두 있는 경우	Mapping은 표시 텍스트/색상을 바꾸고, Threshold는 값 범위 기반 상태 색상을 만듭니다. 같은 값을 서로 다른 의미로 색칠하지 않도록 색상 기준을 하나로 맞춥니다.

Axes & Legend

`Visualization` 그룹 안에는 타입에 따라 `Axes & Scale`과 `Legend`도 함께 나타납니다.

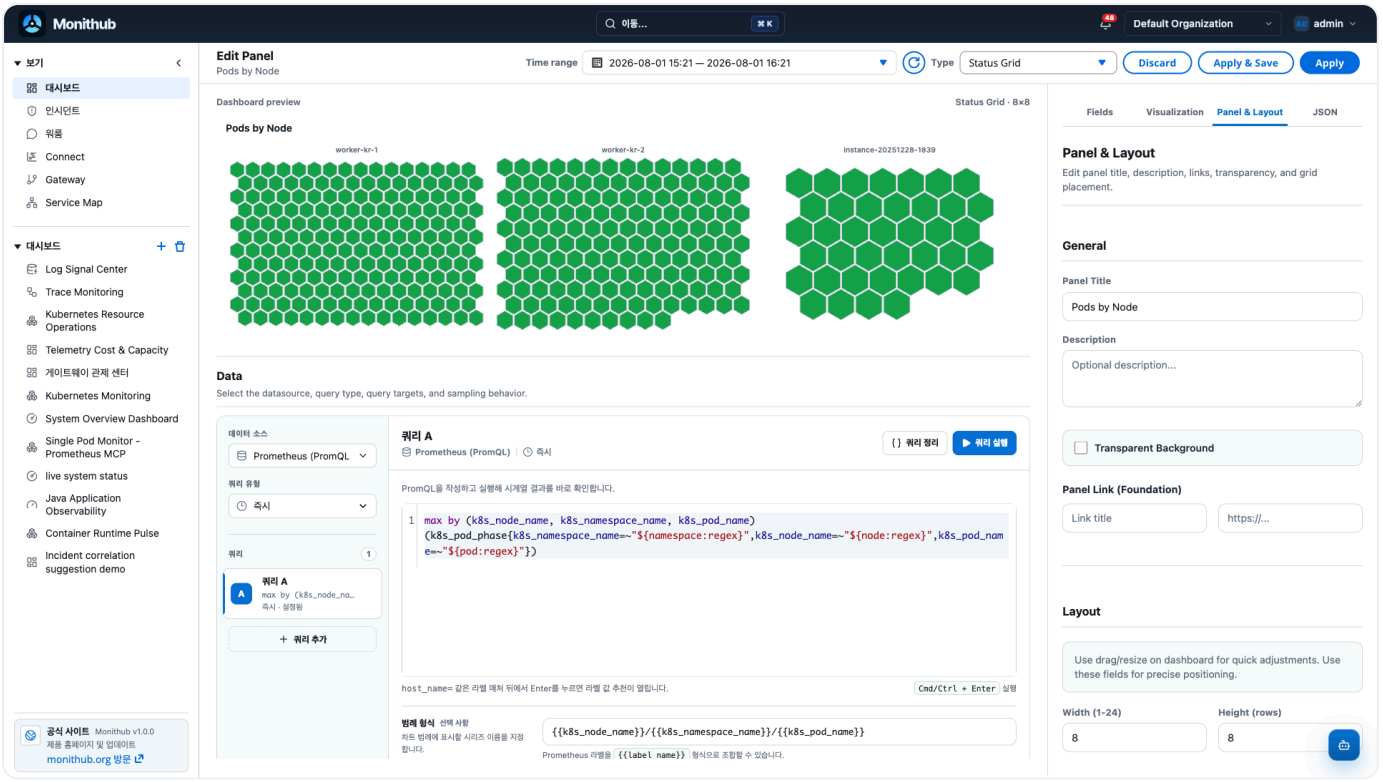
설정	적용 타입	설명
Axis Label	Bar, Scatter, Time Series, Pie, Gauge	축에 표시할 이름입니다.
Scale	Bar, Scatter, Time Series, Pie, Gauge	`Linear` 또는 `Logarithmic`을 고릅니다. 0 또는 음수가 포함된 데이터에는 log scale을 피합니다.
Placement	Bar, Scatter, Time Series, Pie, Gauge	축을 왼쪽 또는 오른쪽에 둡니다.
Axis Unit	Bar, Scatter, Time Series, Pie, Gauge	축 눈금 단위를 정합니다. Field Unit과 맞추는 것이 기본입니다.
Decimals / Min / Max	Bar, Scatter, Time Series, Pie, Gauge	축과 값 범위의 정밀도를 정합니다.
Soft Min / Soft Max	Bar, Scatter, Time Series, Pie, Gauge	데이터가 범위를 벗어나면 자동 확장할 수 있는 느슨한 범위입니다.
Show Legend	Bar, Scatter, Time Series, Pie	legend 표시 여부입니다.
Mode	Bar, Scatter, Time Series, Pie	`List` 또는 `Table` legend를 고릅니다.
Placement	Bar, Scatter, Time Series, Pie	legend를 아래 또는 오른쪽에 둡니다.
Values	Time Series	table legend에 Last, Min, Max, Avg 같은 계산값을 column으로 표시합니다. 최소 하나는 남아야 합니다.

타입별 문서

타입별 조작 설명은 개별 페이지에서 확인합니다. 각 페이지는 query 결과 형태를 맞추는 `Data`, 단위와 threshold를 맞추는 `Fields`, 타입 전용 렌더링을 조정하는 `Visualization` 순서로 읽을 수 있게 구성했습니다.

타입 문서	대표 사용 상황	먼저 확인할 설정
Time Series	시간 흐름에 따른 메트릭 추세, series 비교	range query, draw style, null 처리, 축, legend
Bar	bucket, category, series를 막대 스타일로 비교	draw style, stack, fill, 축 단위
Stat	최신값, 평균, 합계 같은 대표값 표시	instant query, reducer, trend, color mode
Gauge	현재값이 기준 범위 안에서 어디에 있는지 표시	reducer, min/max, threshold, unit
Table	행과 열로 근거 데이터 검토	join, detected columns, cell type, filter
Heatmap	시간과 bucket별 밀도/분포 확인	x/y/value field, color scale, empty cell
Scatter	row 단위 관측치를 x/y 점으로 비교	x field, y field, series field
Status Grid	여러 대상의 정상/비정상 상태 스캔	reducer, reverse status, group, true/false text
Trace	trace 목록과 요약 확인	summary card, trace columns, column visibility
Gantt	시작/종료/기간이 있는 작업을 시간 막대로 표시	time unit, row grouping, row height
Pie	적은 수의 category 비중 표시	category query, unit, decimals, legend
Logs	로그성 텍스트 결과 표시	query, field formatting, panel layout, JSON

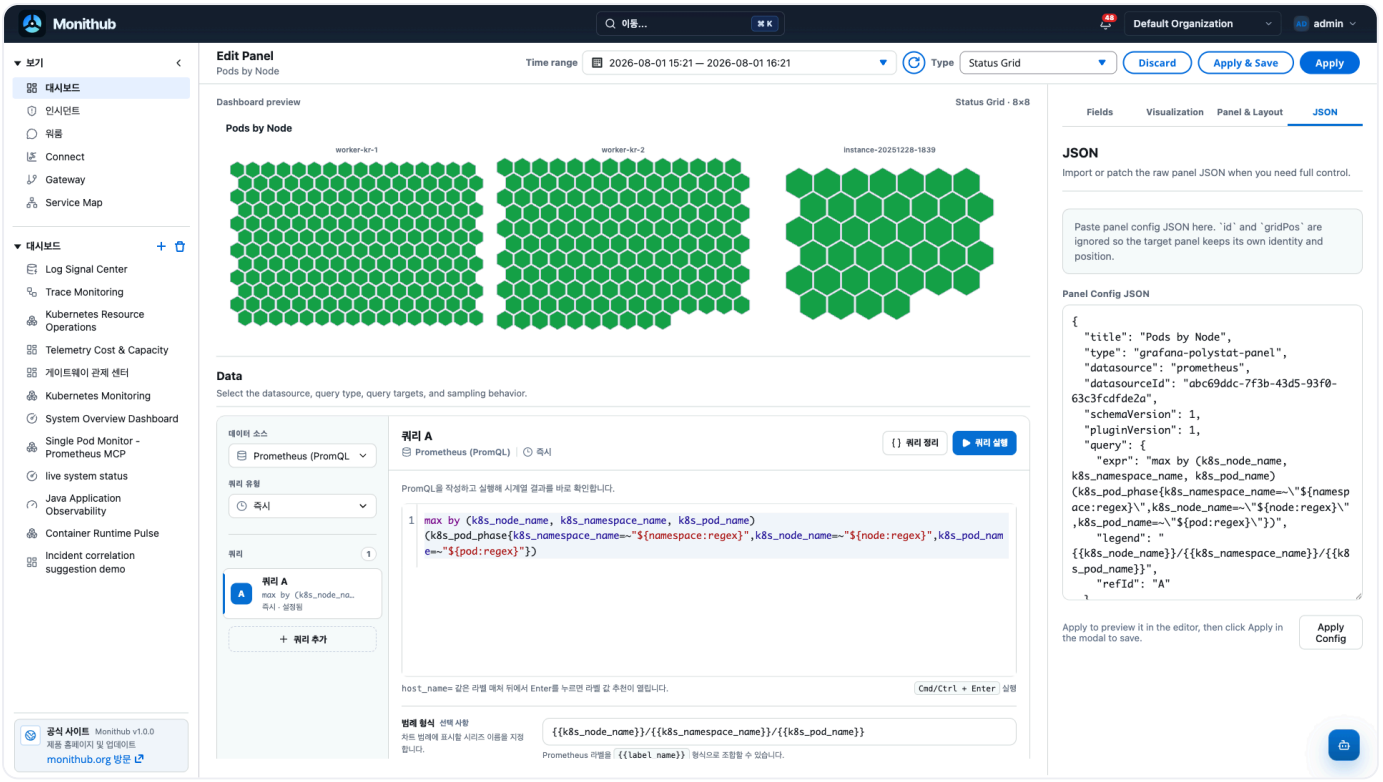
Panel & Layout



설정	역할
Panel Title	대시보드와 패널 header에 표시되는 제목입니다.
Description	패널 설명입니다. 미리보기와 문서화 목적에 사용됩니다.
Transparent Background	패널 배경 투명 여부입니다. 주변 row 배경과 맞춰야 합니다.
Panel Link	패널 메뉴에서 열 수 있는 외부 또는 내부 링크입니다.
Width	24-column grid 기준 너비입니다.
Height	row 단위 높이입니다.
X	0부터 시작하는 좌우 위치입니다.
Y	0부터 시작하는 세로 위치입니다.

빠른 배치는 대시보드의 `Edit layout`에서 drag/resize로 조정하고, 정확한 좌표가 필요할 때 이 탭에서 숫자를 맞춥니다. Width와 X의 합이 24를 넘지 않도록 관리합니다.

JSON



JSON 탭은 UI에 아직 노출되지 않은 panel config를 가져오거나 수정할 때 사용합니다.

설정	역할
Panel Config JSON	패널 config payload를 직접 편집합니다.
Apply Config	JSON을 현재 미리보기 draft에 적용합니다.
error message	JSON parse 또는 적용 실패 이유를 표시합니다.

`Apply Config`는 편집기 내부 draft에만 반영합니다. 실제 대시보드에 확정하려면 이후 `Apply` 또는 `Apply & Save`를 눌러야 합니다. JSON 적용 시 `id`와 `gridPos`는 대상 패널의 identity와 위치를 유지하기 위해 무시됩니다.

JSON을 쓸 때는 아래 순서를 권장합니다.

1. 현재 패널 JSON을 백업합니다.
2. type, queryModel, visualization, options, fieldConfig 중 필요한 부분만 바꿉니다.
3. `Apply Config`로 미리보기와 오른쪽 탭 값이 맞는지 확인합니다.
4. `Apply` 또는 `Apply & Save`로 확정합니다.

Row와 Text 패널

Row 패널은 대시보드의 `Add row`로 추가하는 패널 묶음입니다. 일반 지표 패널처럼 query를 실행하지 않고, row 제목과 접힘 상태, 묶인 패널 배치가 핵심입니다. Row 편집에서는 `General`, `Layout`, `JSON` 중심으로 확인합니다.

Text 패널은 현재 타입 선택 메뉴에는 기본 노출되지 않을 수 있습니다. 기존 대시보드에 Text 패널이 있다면 markdown/text 표시 모드가 JSON에 들어있을 수 있으므로, `Panel & Layout`과 `JSON`을 함께 확인합니다.

추천 편집 순서

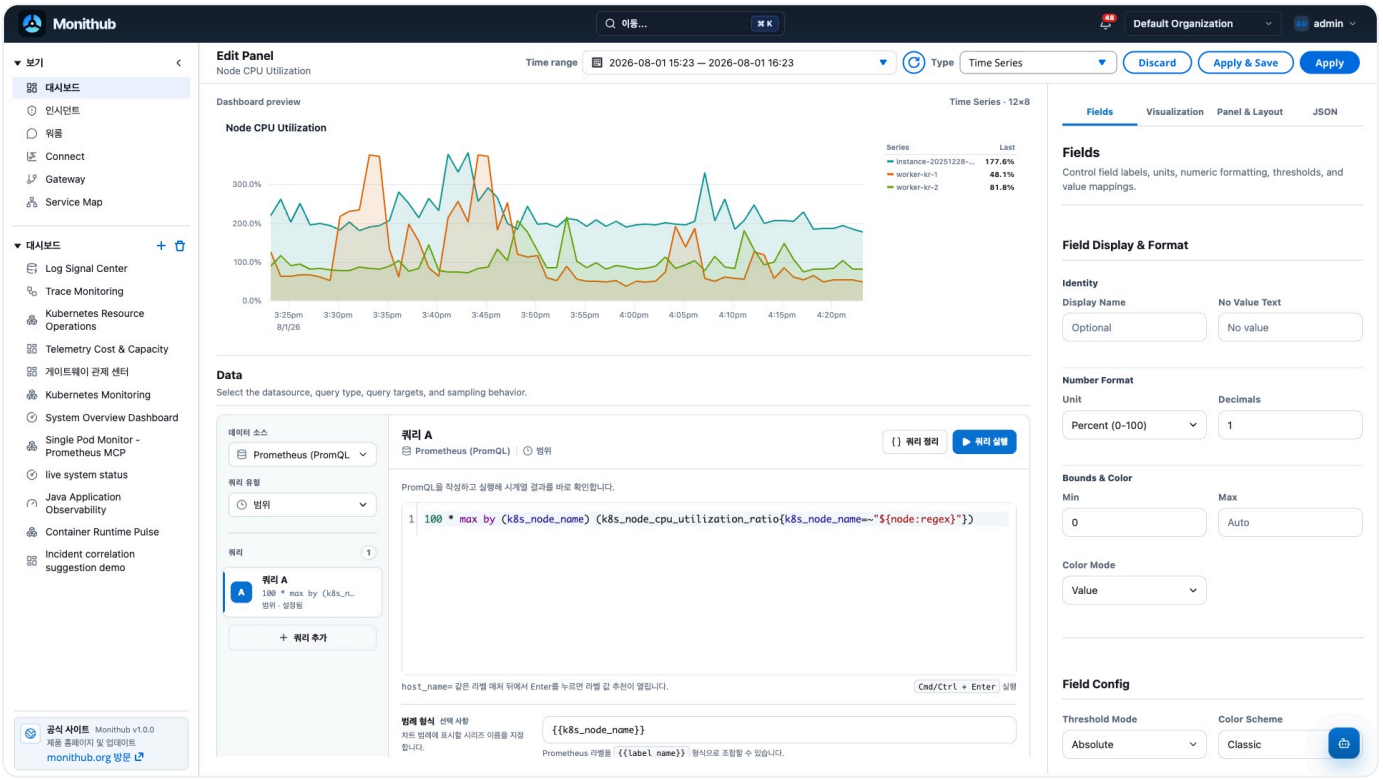
1. `Type`을 먼저 고릅니다.
2. `Data`에서 datasource와 query 결과 형태를 맞춥니다.
3. `Fields`에서 unit, decimals, threshold, value mapping을 맞춥니다.
4. `Visualization`에서 타입별 표현 방식을 조정합니다.
5. `Axes & Scale`과 `Legend`가 있는 타입은 축과 legend를 맞춥니다.
6. `Panel & Layout`에서 제목, 설명, 링크, grid 위치를 정리합니다.
7. 필요한 경우에만 `JSON`을 사용합니다.
8. 미리보기를 확인한 뒤 `Apply & Save`로 저장합니다.

운영 대시보드에서는 query 변경과 type 변경을 한 번에 크게 바꾸지 않는 것이 좋습니다. 먼저 query 결과가 원하는 형태인지 확인하고, 그 다음 시각화와 field 설정을 조정하면 원인을 추적하기 쉽습니다.

Time Series 패널

Time Series는 시간 흐름에 따른 메트릭 변화, 이상탐지 overlay, 여러 series 비교를 볼 때 사용합니다.
예시는 Node CPU Utilization 편집 화면입니다.

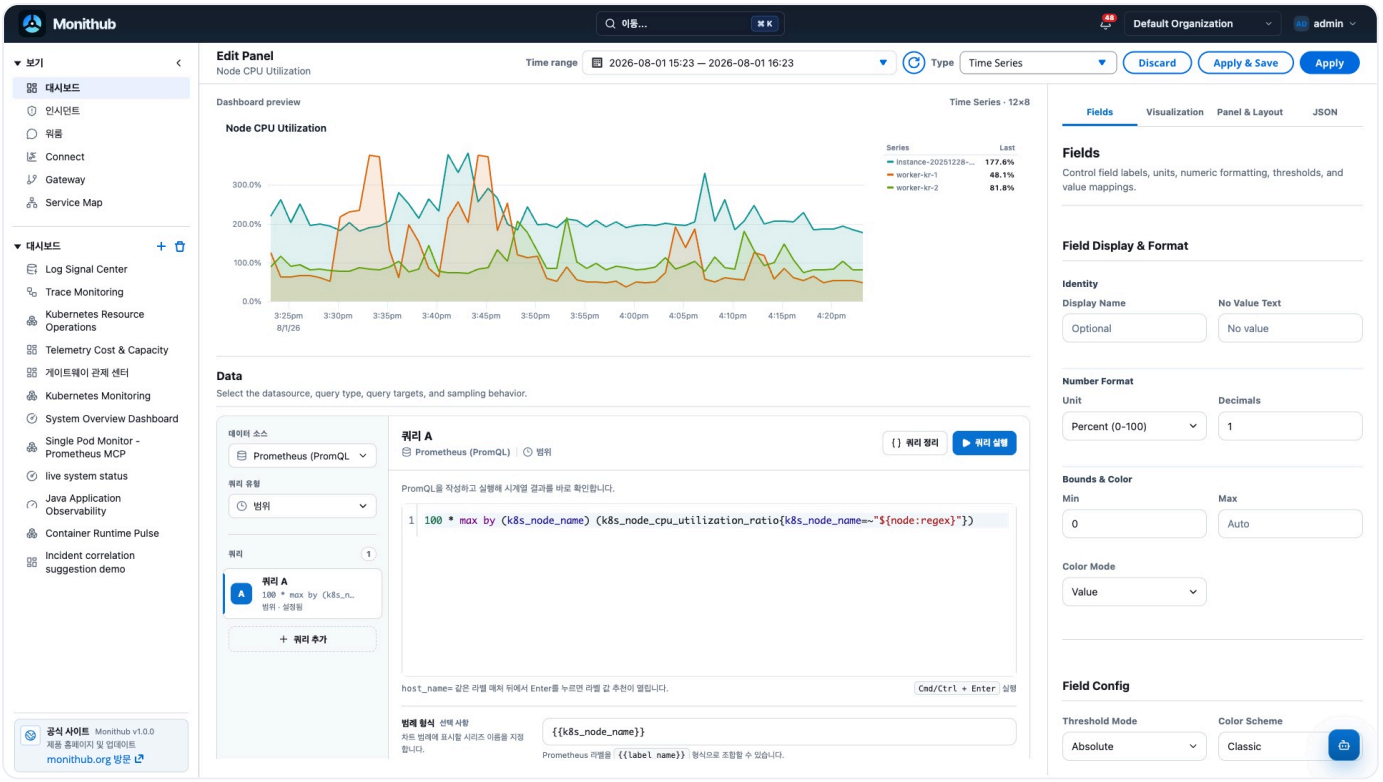
Data



Data에서는 시간축을 만들 query를 먼저 맞춥니다. 추세를 보는 패널이므로 Prometheus는 보통 Range query를 사용하고, ClickHouse를 쓰면 시간 bucket column이 결과에 포함되어야 합니다.

설정	역할
Datasource	Prometheus (PromQL) 또는 ClickHouse (SQL)을 선택합니다.
Query type	추세는 Range, 현재값은 Instant를 사용합니다.
Queries	여러 query를 추가해 series를 겹쳐 비교합니다.
Query editor	PromQL/SQL을 작성하고 Run Query로 반환 series를 확인합니다.
Legend Format	{{pod}}, {{service}}처럼 legend 이름을 읽기 쉽게 정합니다.

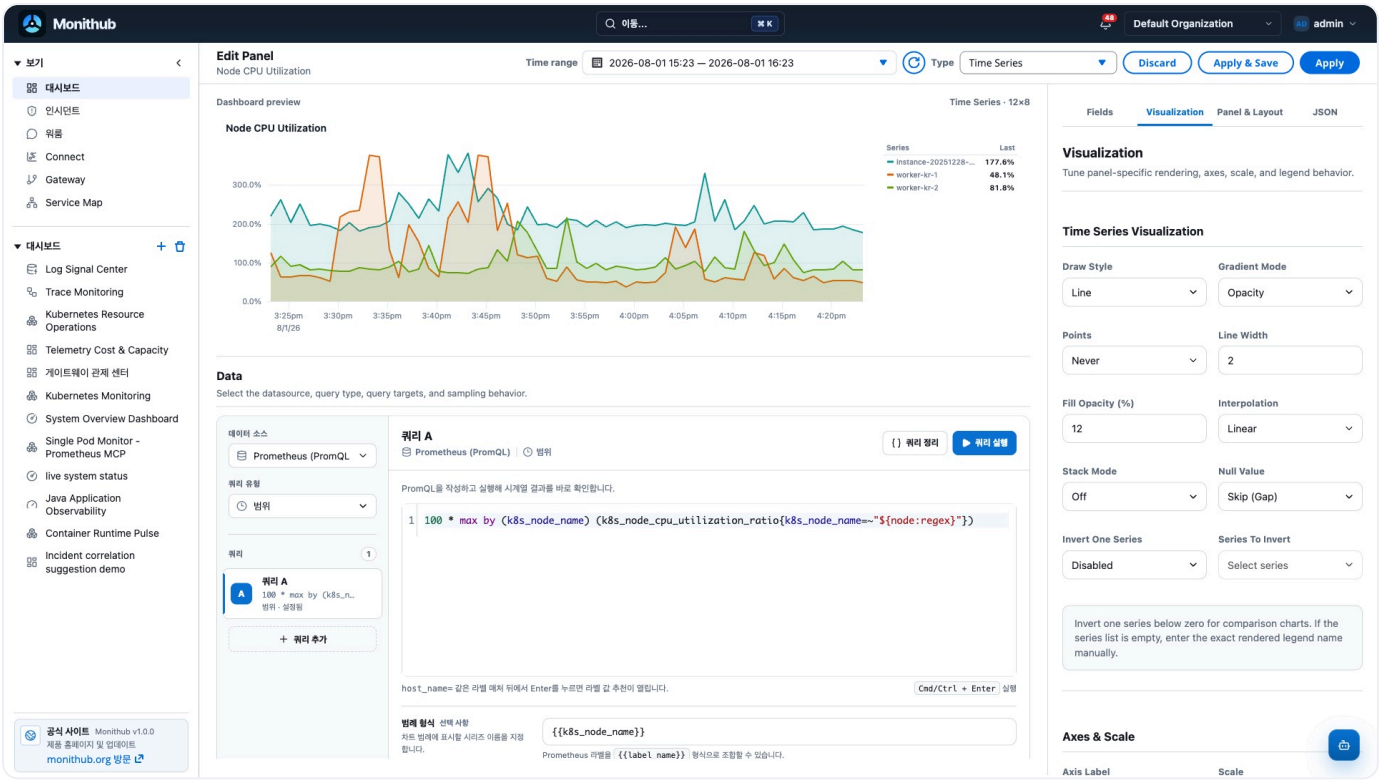
Fields



Fields는 line 값이 어떤 단위와 상태 색으로 보이는지를 정합니다. 축 단위와 tooltip 단위가 다르면 운영자가 값을 잘못 읽기 쉬우므로 여기서 기본 포맷을 먼저 맞춥니다.

설정	역할
Display Name / No Value Text	legend, tooltip, 빈 값 표시를 정리합니다.
Unit / Decimals	tooltip, legend table, axis와 맞춰야 하는 기본 단위입니다.
Min / Max	값 범위와 threshold 판단 기준입니다.
Threshold Steps	warning/critical 같은 값 구간 색상을 정의합니다.
Value Mappings	특정 값이나 범위를 텍스트와 색상으로 바꿉니다.
Series Palette	여러 series의 색상 순서를 정합니다.

Visualization



Visualization은 같은 query 결과를 어떤 그래프 형태로 그릴지 정합니다. 값 자체의 의미는 Fields에서, 선/점/면/stack 표현은 여기서 조정합니다.

설정	역할
Draw Style	line, bars, points 중 표현 방식을 고릅니다.
Gradient Mode	단색, 투명도 gradient, hue gradient를 선택합니다.
Points / Line Width	점 표시와 선 두께를 조정합니다.
Fill Opacity	area처럼 채우는 정도입니다.
Interpolation	linear, smooth, step before, step after 중 선 연결 방식을 고릅니다.
Stack Mode	series를 쌓아서 총량 또는 비중을 비교합니다.
Null Value	누락 구간을 0으로 채울지, 끊을지, 연결할지 정합니다.
Invert One Series	inbound/outbound처럼 한 series를 0 아래로 뒤집습니다.

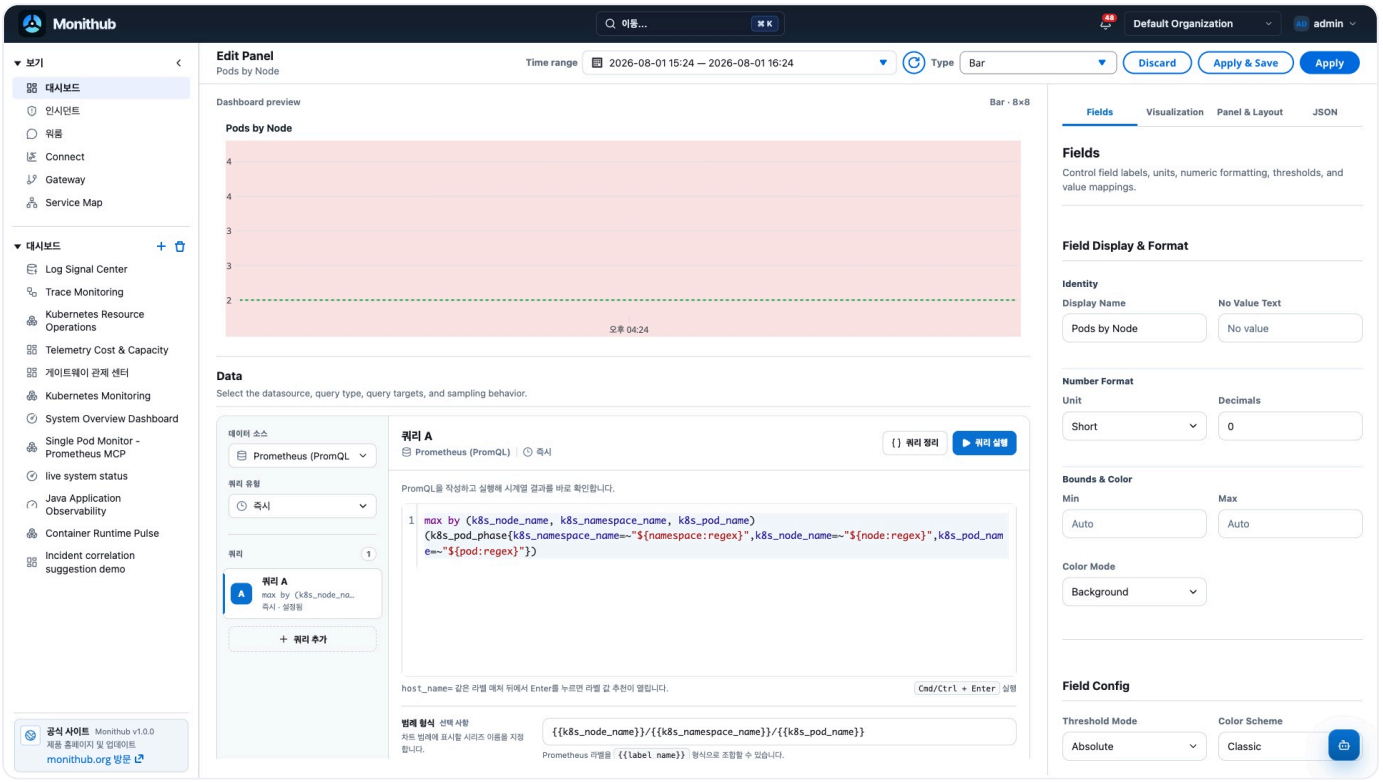
확인 순서

먼저 Data에서 range query와 legend를 안정화하고, Fields에서 unit과 threshold를 맞춘 뒤, Visualization에서 null 처리와 stack 여부를 정합니다.

Bar 패널

Bar는 Time Series 계열 데이터를 막대 스타일로 비교할 때 사용합니다. 같은 query라도 막대 화면에서는 구간별 크기와 series 간 비교가 더 중요합니다.

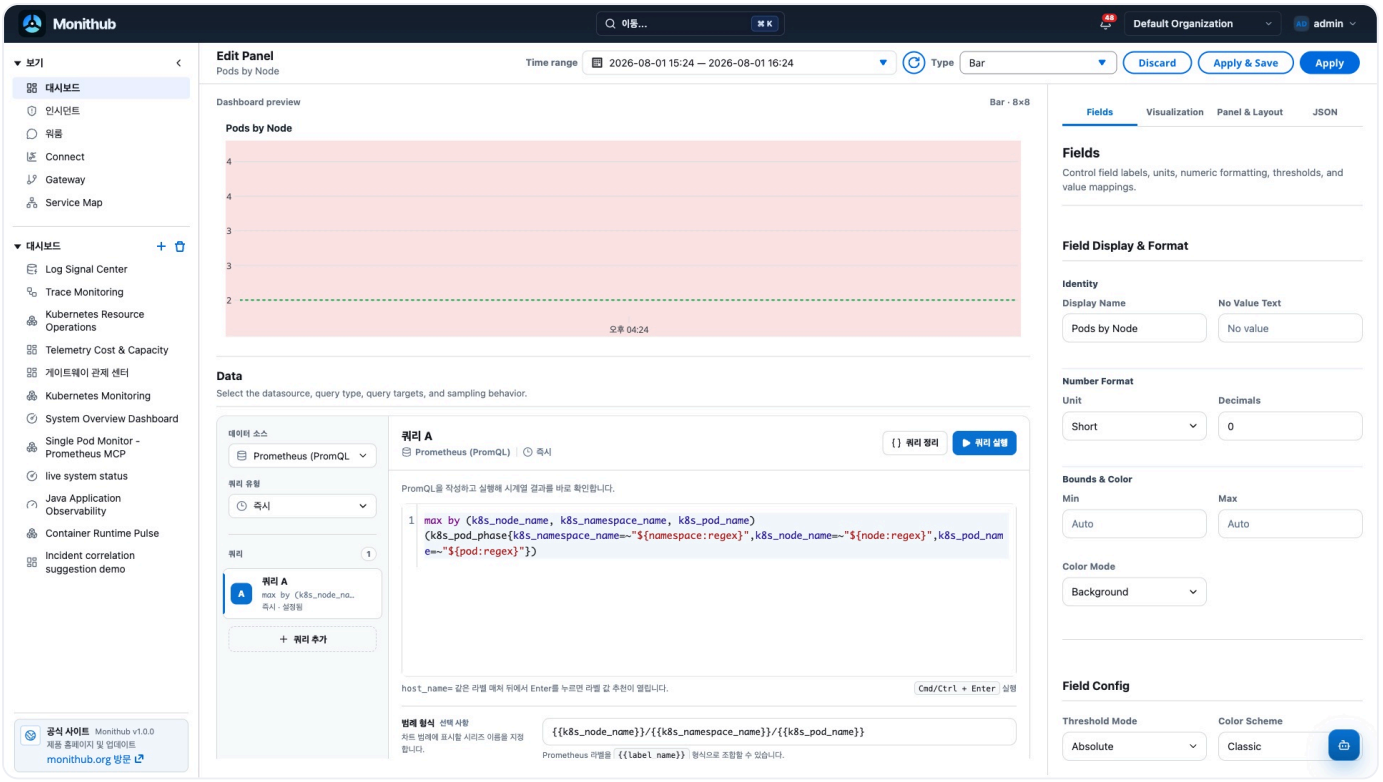
Data



Data에서는 막대 하나가 무엇을 의미하는지 먼저 정합니다. 시간 bucket 비교라면 range query, category 비교라면 query에서 category별 집계를 만들어야 합니다.

설정	역할
Query type	시간 bucket 막대는 `Range`, category 집계는 table query나 PromQL 집계를 사용합니다.
Queries	여러 series를 비교할 때 query를 추가합니다.
Query editor	bucket 크기와 time range가 너무 촘촘하지 않은지 확인합니다.
Legend Format	막대 색상이 어떤 series인지 바로 읽히도록 label을 줄입니다.

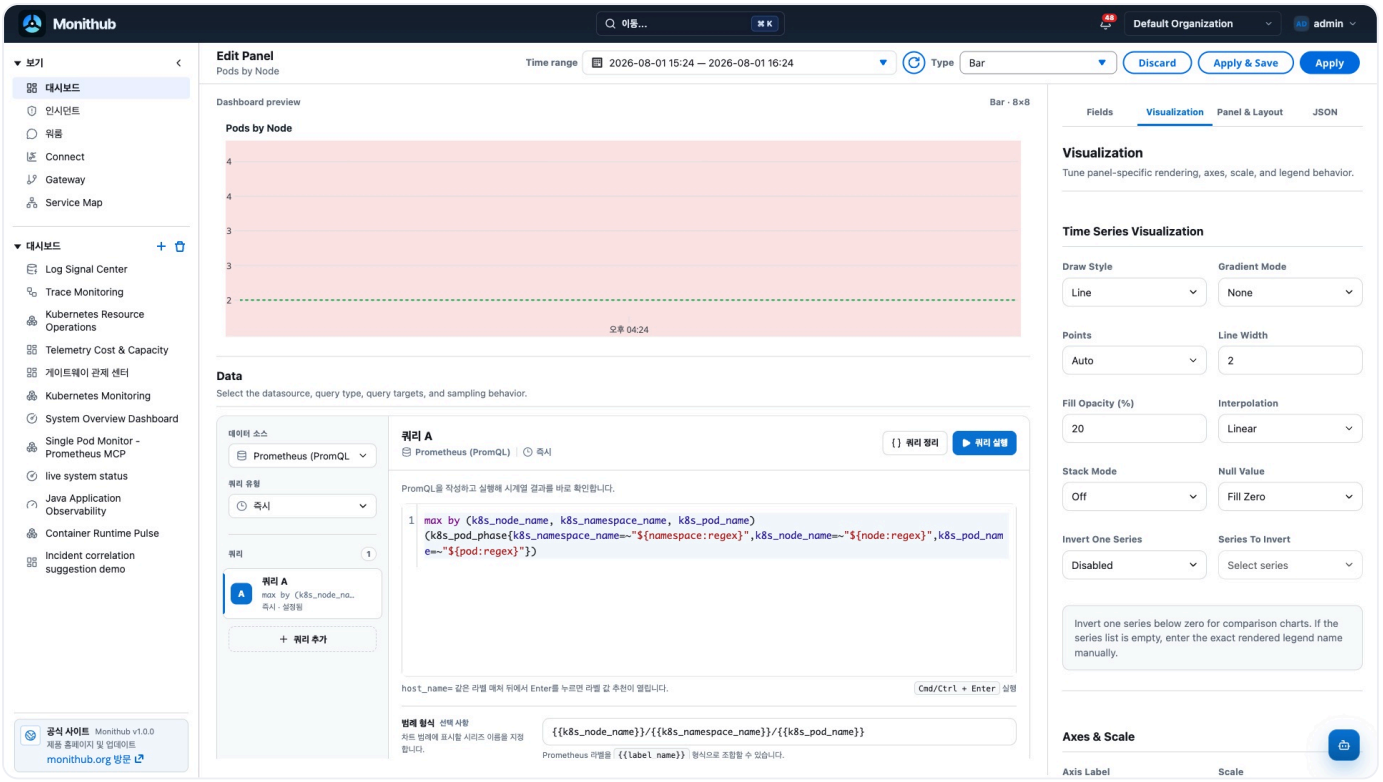
Fields



Fields에서는 막대 높이와 tooltip 값의 단위를 맞춥니다. Bar는 축 단위가 시각적 판단을 크게 좌우하므로 Field Unit과 Axis Unit을 같은 기준으로 두는 편이 좋습니다.

설정	역할
Unit / Decimals	막대 높이와 tooltip 값을 읽는 기준입니다.
Min / Max	막대 축과 threshold 색상 기준에 영향을 줍니다.
Threshold Steps	값 구간별 상태 색상을 정합니다.
Series Palette	여러 막대 series 색상 순서를 통일합니다.

Visualization



Visualization은 Time Series와 같은 설정을 공유하지만, Bar에서는 `Draw Style`과 `Stack Mode`가 핵심입니다.

설정	역할
Draw Style	Bar에서는 `Bars`로 둡니다.
Stack Mode	series를 쌓아서 총량 또는 비중을 비교합니다.
Fill Opacity	막대 채움 정도입니다.
Gradient Mode	운영 화면에서는 단색이 가장 안정적입니다.
Null Value	누락 구간을 0으로 채우면 실제 0과 수집 공백이 섞일 수 있습니다.

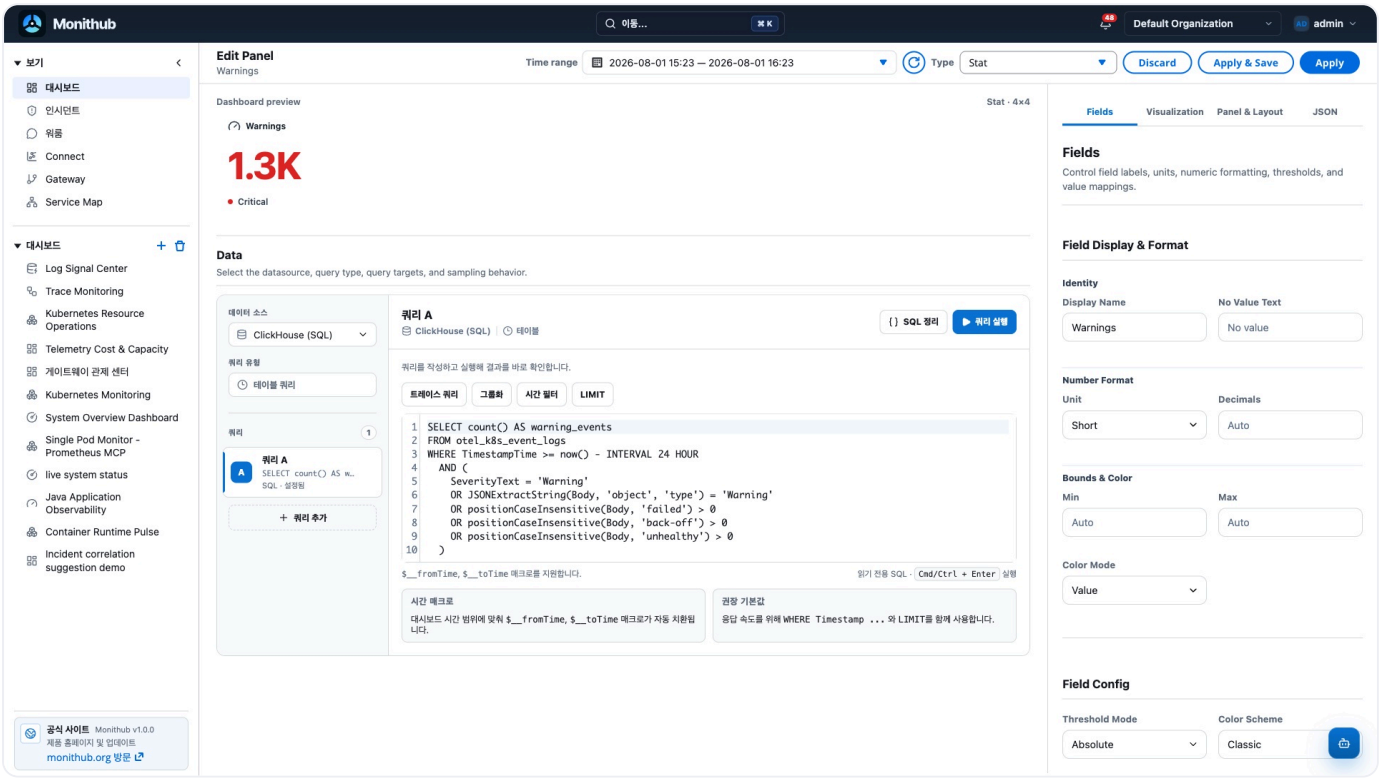
확인 순서

Data에서 비교 단위를 정하고, Fields와 Axes에서 단위를 맞춘 뒤, Visualization에서 stack 여부를 결정합니다.

Stat 패널

Stat은 하나의 대표값 또는 여러 series의 대표값을 간결하게 보여줍니다. 예시는 `Node CPU` 계열 Stat 편집 화면입니다.

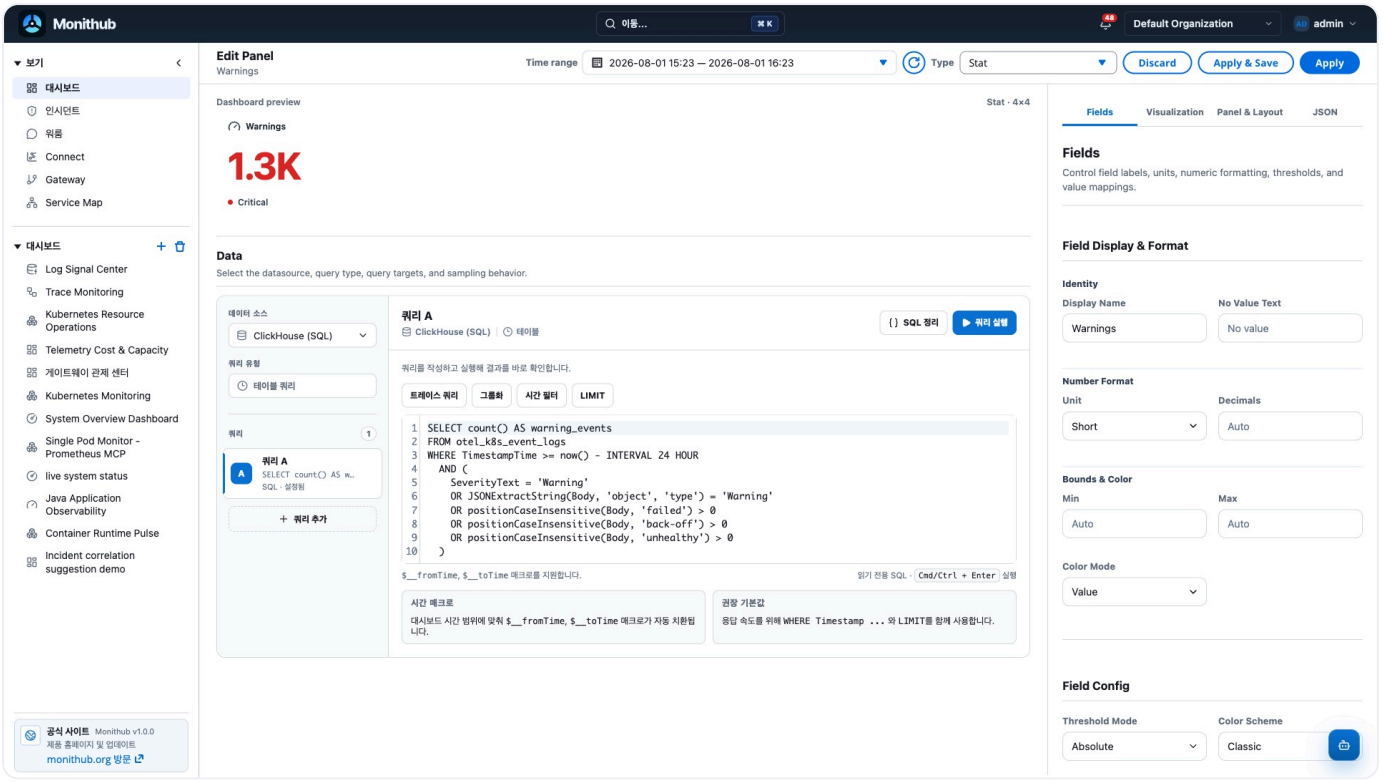
Data



Data에서는 Stat에 들어갈 값의 개수와 query type을 결정합니다. 현재값만 필요하면 `Instant`, 작은 sparkline이나 변화 badge가 필요하면 `Range`를 사용합니다.

설정	역할
Query type	현재값은 `Instant`, 추세와 변화 비교는 `Range`를 사용합니다.
Queries	여러 series가 반환되면 Visualization의 multi-series mode가 표시 방식을 정합니다.
Run Query	series 개수와 값 타입을 확인합니다.
Legend Format	multi-series grid에서 카드 이름으로 쓰입니다.

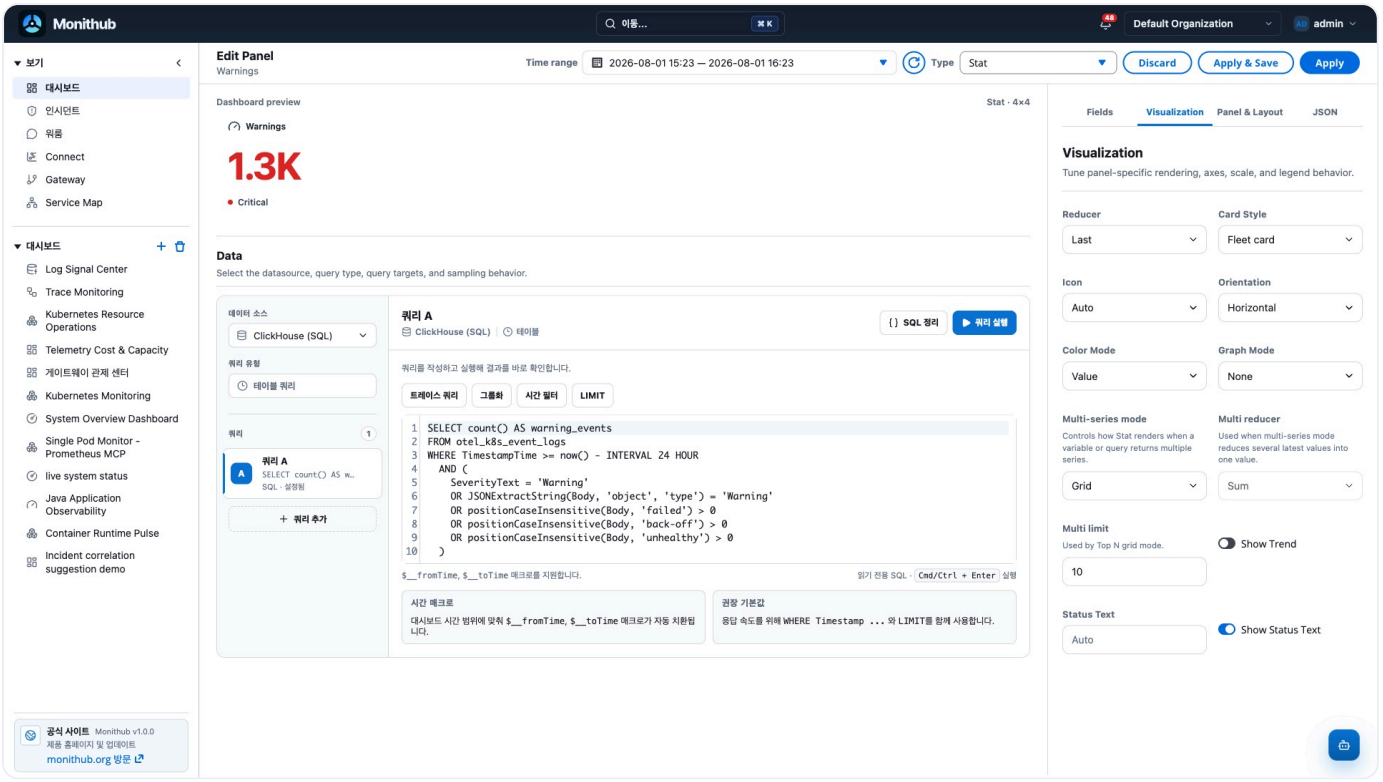
Fields



Fields는 큰 숫자 하나를 어떻게 읽을지 정하는 곳입니다. unit, decimals, threshold가 맞지 않으면 Stat은 가장 빠르게 오해를 만듭니다.

설정	역할
Display Name	단일 Stat 카드의 제목 또는 값 이름입니다.
Unit / Decimals	큰 숫자의 단위와 소수점 표시입니다.
Min / Max	background color와 threshold 판단 기준입니다.
Threshold Steps	값 색상과 상태 색을 나눕니다.
Value Mappings	0/1, 상태 코드, sentinel 상태값을 텍스트로 바꿉니다.

Visualization



Visualization은 series를 어떤 대표값으로 줄이고, 카드 안에서 어떻게 표현할지 정합니다.

설정	역할
Reducer	Last, Mean, Sum, Min, Max 중 대표값을 선택합니다.
Card Style / Icon	카드 밀도와 아이콘 표현을 정합니다.
Orientation	값과 보조 그래프 배치를 정합니다.
Color Mode / Graph Mode	값 색상과 sparkline 표시 여부를 정합니다.
Multi-series mode	여러 series를 grid, reduce, top N, single only로 처리합니다.
Show Trend / Status Text	이전 값 대비 변화와 상태 문구를 표시합니다.

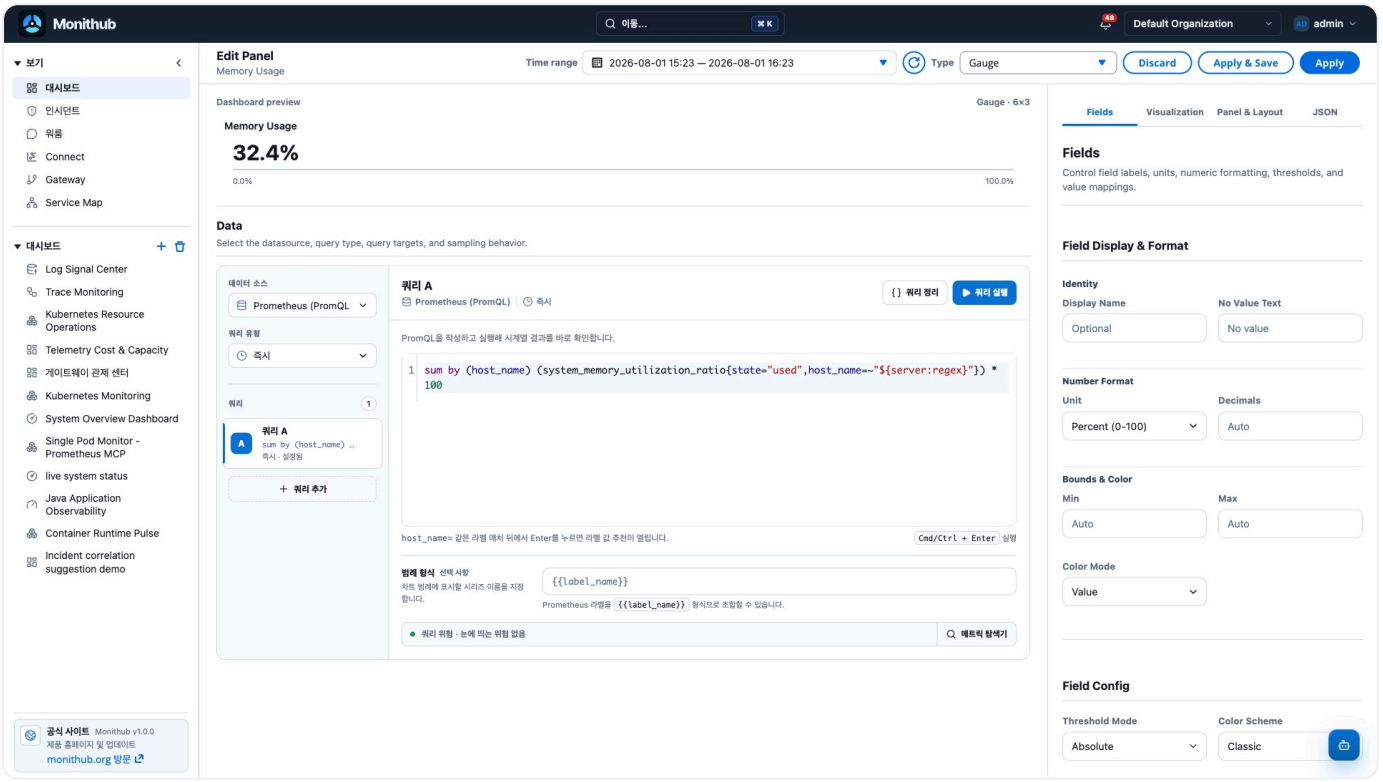
확인 순서

먼저 query 결과가 하나인지 여러 series인지 확인하고, Fields에서 단위를 맞춘 뒤, Reducer와 multi-series mode를 정합니다.

Gauge 패널

Gauge는 현재값이 기준 범위 안에서 어느 위치인지 보여줍니다. 단일 값의 상태를 빠르게 판단해야 할 때 사용합니다.

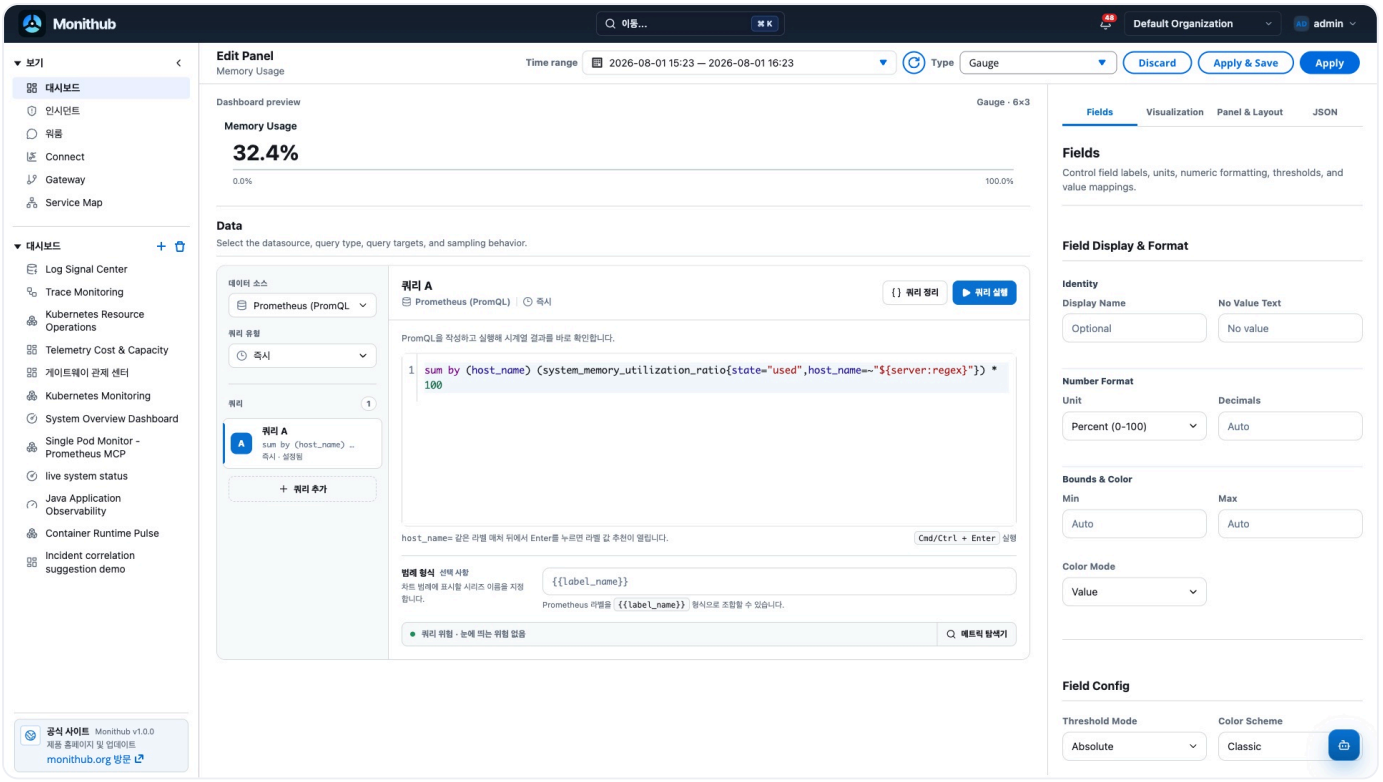
Data



Data에서는 gauge에 들어갈 값을 하나의 대표값으로 만들 수 있는지 확인합니다. 여러 series가 남으면 reducer가 어느 값에 적용되는지 혼란스러울 수 있습니다.

설정	역할
Query type	현재 상태만 보면 `Instant`, 최근 구간 대표값이면 `Range`를 사용합니다.
Run Query	결과가 하나의 series로 줄어드는지 확인합니다.
Legend Format	여러 series가 남을 수 있다면 대표값 이름을 명확히 둡니다.

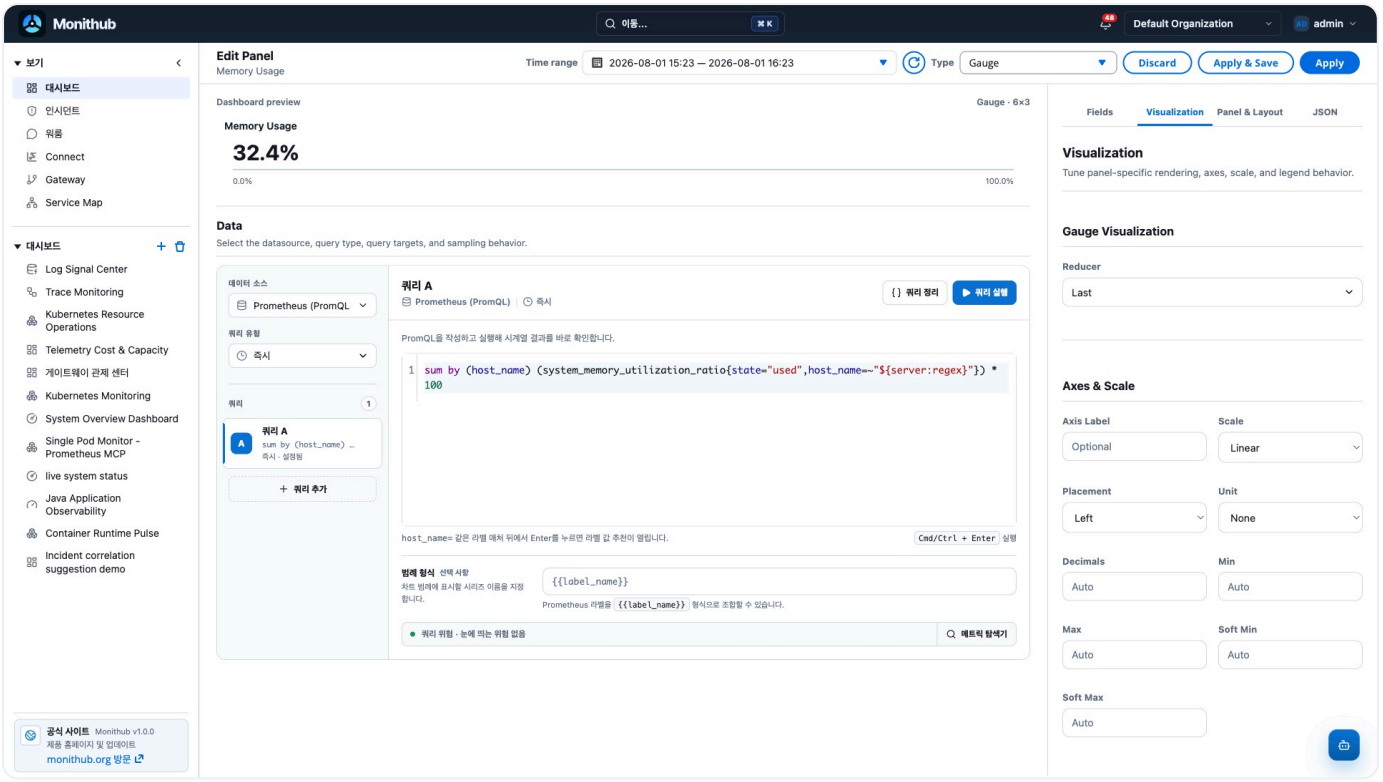
Fields



Fields에서는 gauge의 범위와 색상 기준을 맞춥니다. Gauge는 Min, Max, threshold가 화면 의미를 거의 결정합니다.

설정	역할
Unit / Decimals	중앙 값과 gauge tick의 기본 단위입니다.
Min / Max	gauge 시작/끝 범위입니다.
Threshold Steps	warning/critical 색상 구간입니다.
Value Mappings	특수 상태값을 텍스트로 바꿀 때 사용합니다.

Visualization



Visualization에서는 어떤 대표값을 gauge에 올릴지 정합니다. 범위와 색상은 Fields, 축 단위는 Axes & Scale과 함께 확인합니다.

설정	역할
Reducer	Last, Mean, Sum, Min, Max 중 gauge에 표시할 대표값을 선택합니다.
Axes & Scale	axis unit, min/max, scale을 조정합니다.
Fields Min / Max	gauge 범위와 색상 판단 기준으로 함께 작동합니다.

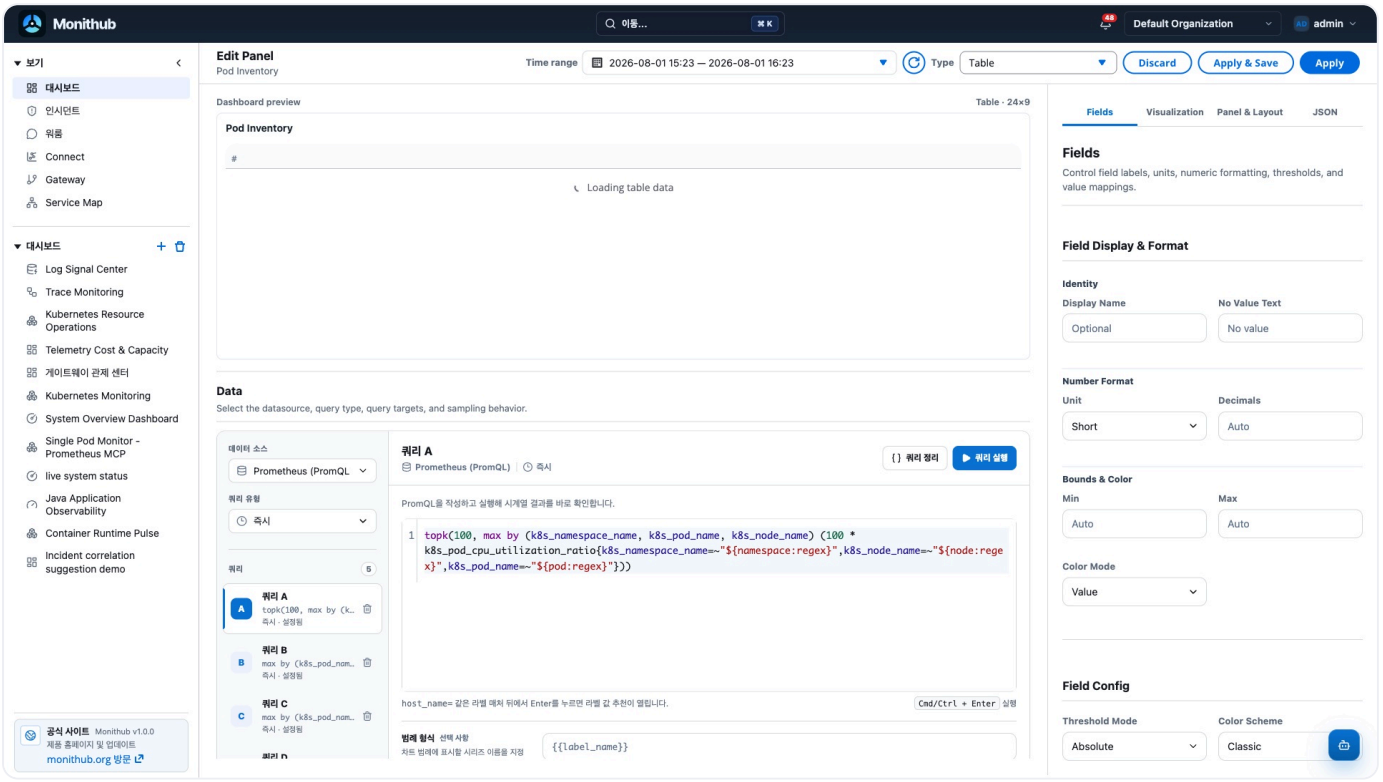
확인 순서

Data에서 단일 값으로 줄어드는지 확인하고, Fields에서 min/max와 threshold를 맞춘 뒤, Visualization의 reducer를 확정합니다.

Table 패널

Table은 query 결과를 row/column으로 검토하는 패널입니다. ClickHouse SQL 결과, Kubernetes event, 로그 요약, trace inventory처럼 근거 데이터를 보여줄 때 사용합니다.

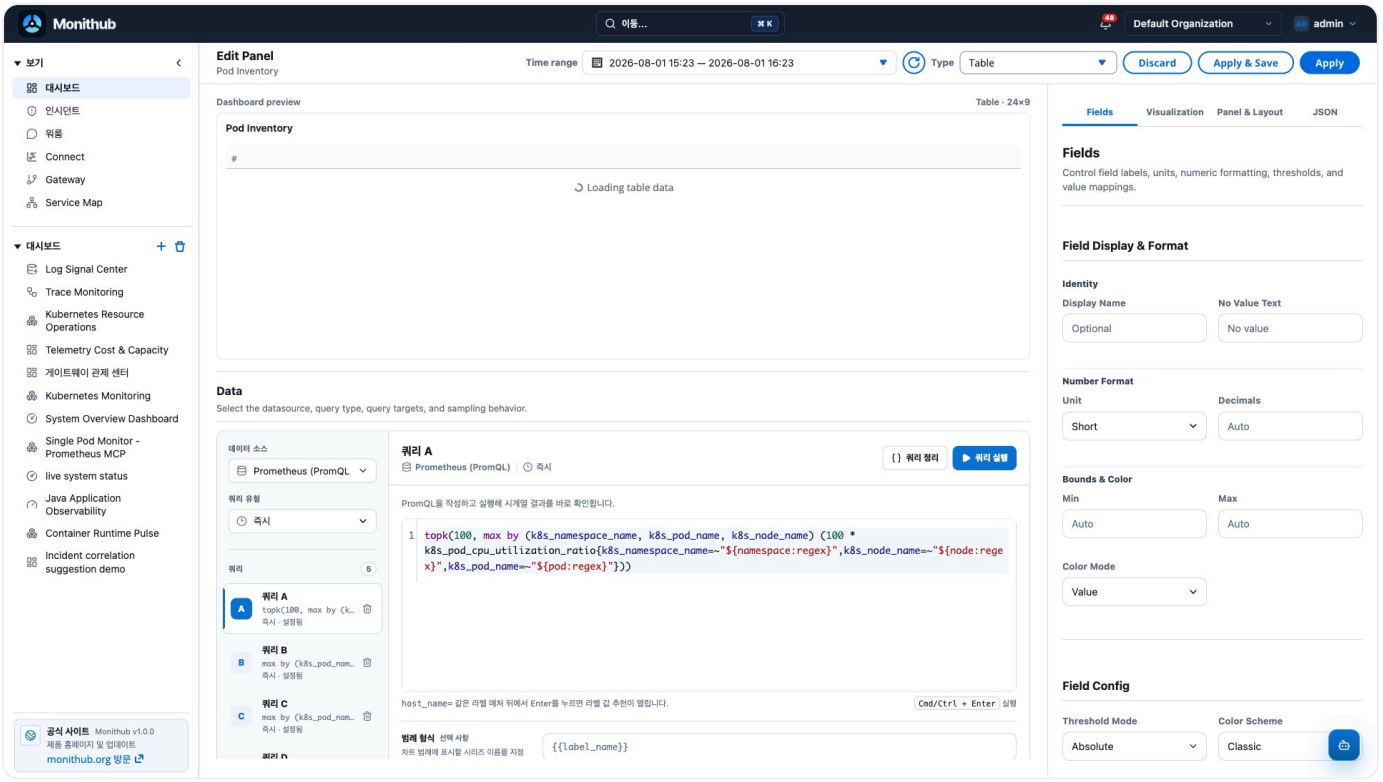
Data



Data에서는 row grain과 column 이름을 먼저 안정화합니다. column alias가 바뀌면 Visualization의 Detected Columns 설정도 다시 확인해야 합니다.

설정	역할
Datasource	로그, trace, inventory는 ClickHouse SQL을 자주 사용합니다.
Queries	`A`, `B`처럼 여러 query를 추가하고 join할 수 있습니다.
Query editor	column alias를 명확히 붙입니다.
Run Query	row 수, column 이름, null 비율을 확인합니다.
Join	query가 둘 이상이면 join 대상, key, type을 정합니다.

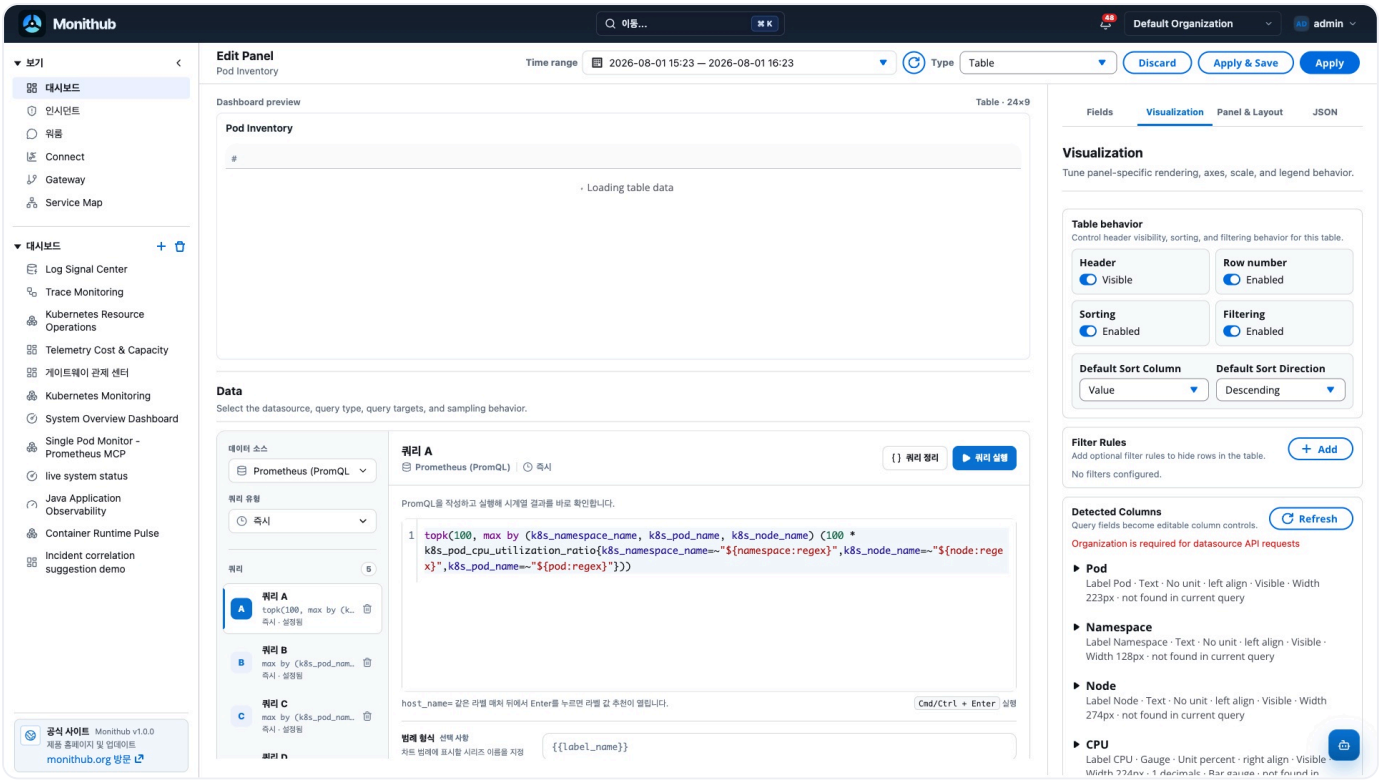
Fields



Fields는 table 전체의 기본 표시 규칙입니다. 특정 column만 다르게 보여야 하면 Visualization의 Detected Columns에서 더 구체적으로 덮어씁니다.

설정	역할
Display Name / No Value Text	기본 field 이름과 빈 값 문구입니다.
Unit / Decimals	column별 unit이 없을 때 쓰는 기본 포맷입니다.
Threshold Steps	column별 threshold가 없을 때 공통 색상 기준입니다.
Value Mappings	상태 코드나 boolean 값을 table cell 문구로 바꿉니다.

Visualization



Visualization은 table behavior와 column별 표시 방식을 조정합니다. 이 탭이 Table 사용 가이드의 핵심입니다.

설정	역할
Header / Row number	table header와 행 번호 표시 여부입니다.
Sorting / Filtering	사용자가 정렬하거나 filter rule로 row를 숨길 수 있게 합니다.
Detected Columns	현재 query 결과의 field를 column 설정으로 엽니다.
Rename Field / Cell Type	column 이름과 text, gauge, sparkline, background color 표시를 정합니다.
Unit / Decimals / Min / Max	column별 숫자 포맷과 gauge 기준입니다.
Thresholds / Data link	column 전용 색상 구간과 drill-down 링크입니다.

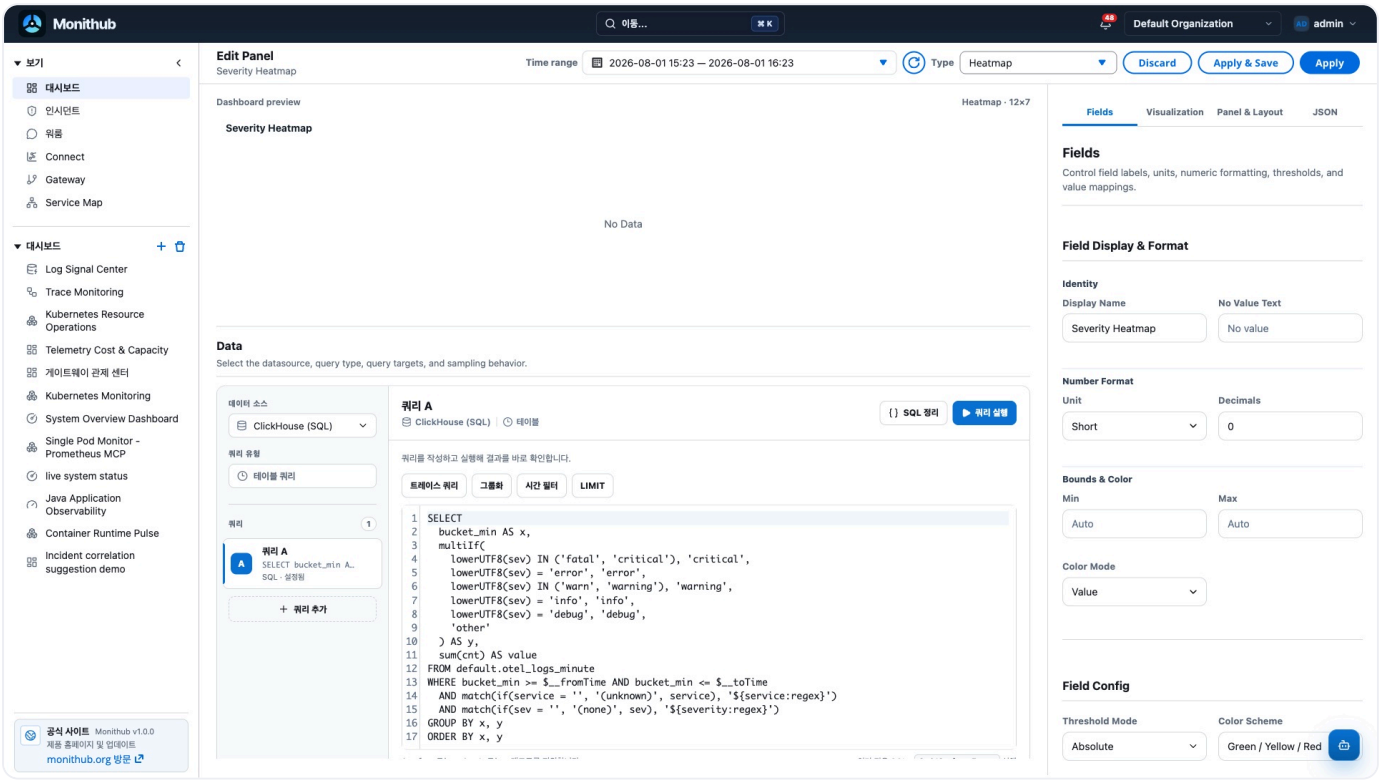
확인 순서

Data에서 column 이름을 고정하고, Fields에서 공통 포맷을 맞춘 뒤, Visualization에서 column별 예외를 정리합니다.

Heatmap 패널

Heatmap은 시간과 bucket, series label, value로 밀도 변화를 볼 때 사용합니다. latency histogram, error bucket, resource distribution처럼 값의 분포가 중요한 화면에 적합합니다.

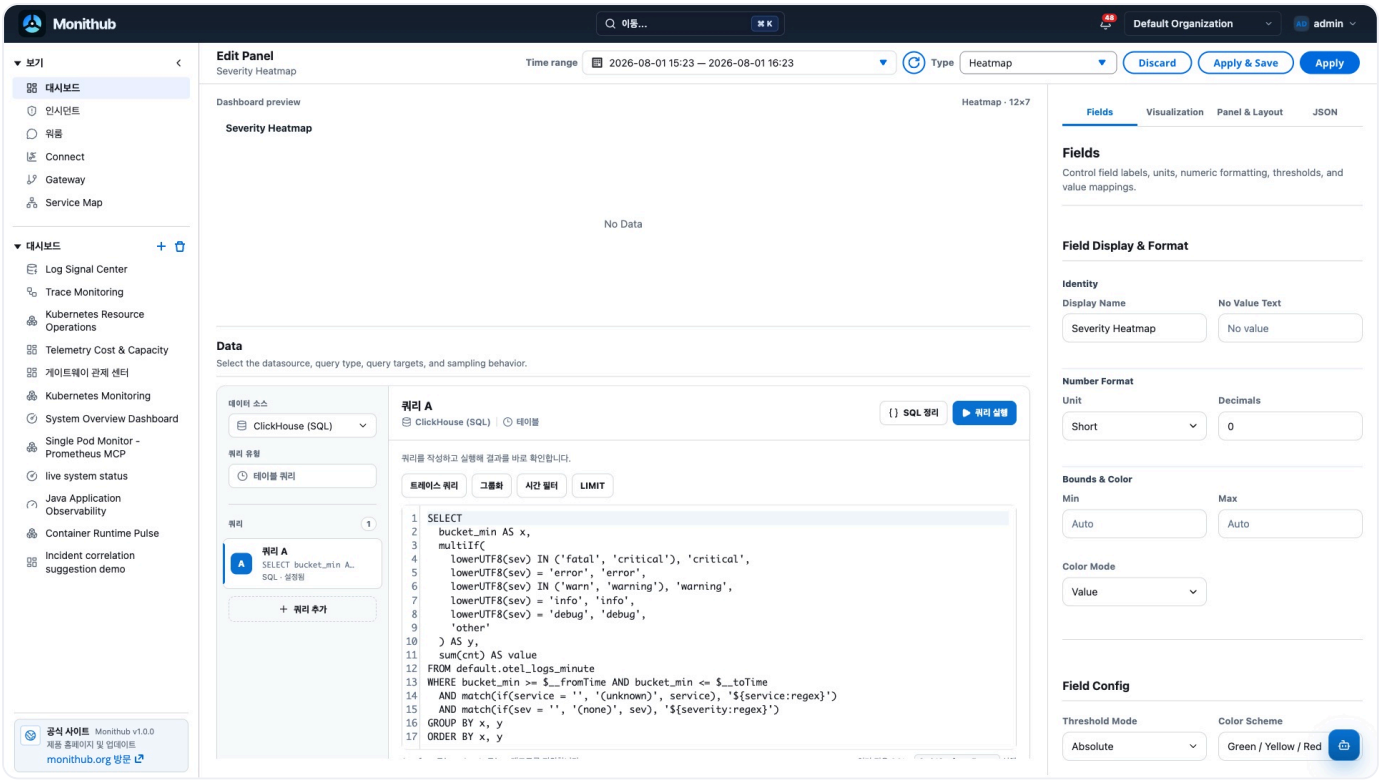
Data



Data에서는 x축 시간, y축 bucket, 색상 값이 한 row에 함께 나오도록 query를 구성합니다. ClickHouse SQL에서는 `$__fromTime`, `$__toTime` 매크로를 time range와 맞춥니다.

설정	역할
Datasource	trace latency는 `ClickHouse (SQL)`, Prometheus histogram은 `Prometheus (PromQL)`을 사용합니다.
Query type	ClickHouse는 table query, Prometheus histogram은 보통 `Range` query입니다.
반환 field	`x`, `bucket`, `value` 또는 `time`, `le`, `value`처럼 이름을 명확히 둡니다.
Run Query	row가 비어 있지 않은지, bucket label이 너무 많지 않은지 확인합니다.

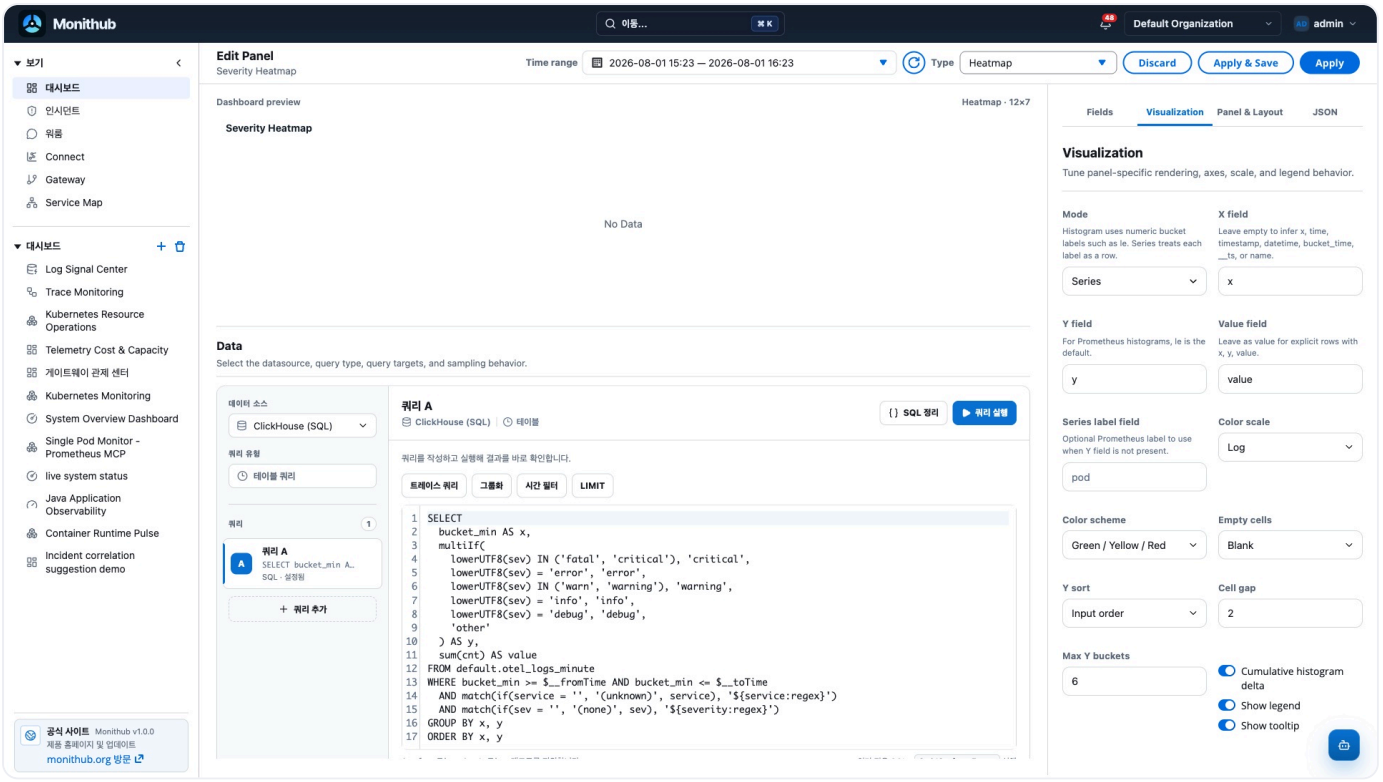
Fields



Fields는 색상 legend와 tooltip 값의 단위, 소수점, 상태 색 기준을 정합니다. Heatmap cell의 위치는 Visualization이 정하지만 값이 어떻게 읽히는지는 Fields가 좌우합니다.

설정	역할
Display Name / No Value Text	tooltip과 빈 cell 문구를 정합니다.
Unit / Decimals	count면 `Short`, latency면 `ms`처럼 맞춥니다.
Min / Max	극단값 때문에 색이 눌릴 때 범위를 조정합니다.
Threshold Steps	특정 값 이상을 warning/critical처럼 읽을 때 사용합니다.
Value Mappings	0, null, 특수 상태값을 사람이 읽는 텍스트나 색상으로 바꿉니다.

Visualization



Visualization에서는 query 결과의 어떤 field를 x/y/value로 쓸지와 heatmap 렌더링 방식을 정합니다.

설정	역할
Mode	histogram, series, value-binning 중 해석 방식을 고릅니다.
X / Y / Value field	시간, bucket, 색상 값 field를 지정합니다.
Series label field	Y field가 없을 때 행 이름으로 쓸 label입니다.
Color scale / scheme	linear/log와 색상 팔레트를 정합니다.
Empty cells	빈 칸을 blank로 둘지 0으로 채울지 정합니다.
Y sort / Max Y buckets	row 정렬과 bucket 수 상한입니다.
Cumulative histogram delta	누적 histogram을 interval delta처럼 해석합니다.

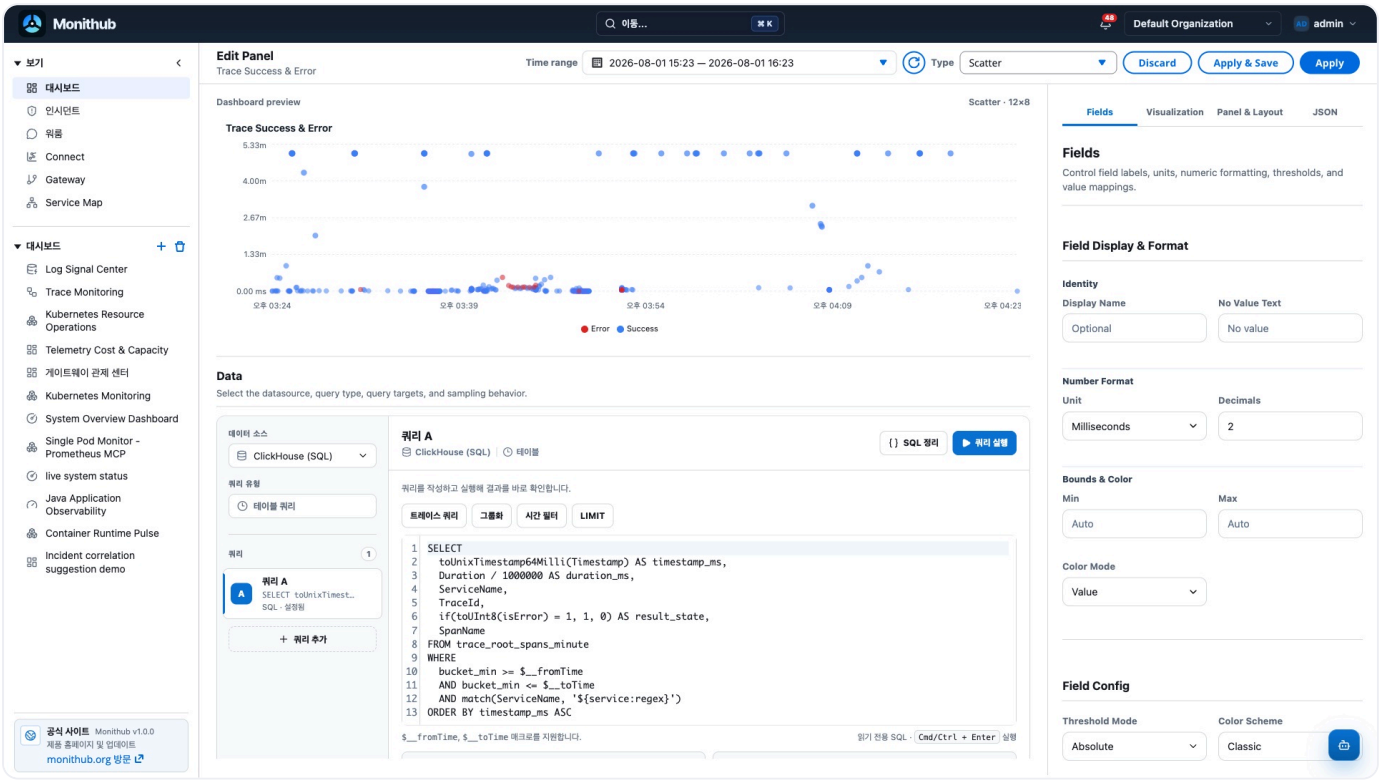
확인 순서

Data에서 x/y/value shape를 확인하고, Fields에서 값 단위를 맞춘 뒤, Visualization에서 field mapping과 empty cell 정책을 정합니다.

Scatter 패널

Scatter는 각 row를 하나의 점으로 표시합니다. trace duration, request size, latency outlier처럼 개별 관측치가 중요할 때 적합합니다.

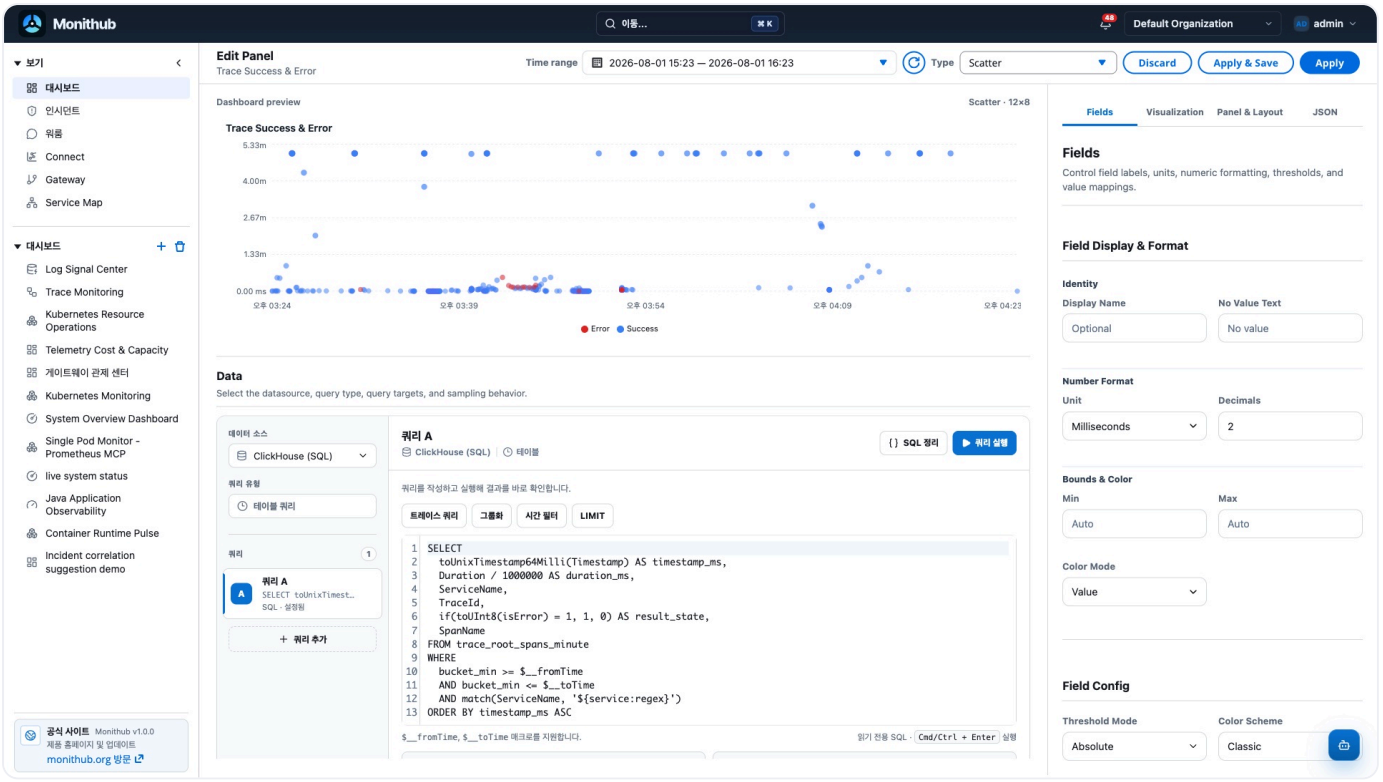
Data



Data에서는 한 row가 한 trace, 요청, 이벤트 같은 하나의 관측치가 되도록 query를 구성합니다.

설정	역할
Datasource	trace duration과 request size 같은 행 단위 데이터는 ClickHouse SQL과 잘 맞습니다.
Query type	ClickHouse는 table query로 동작합니다.
Query Sampling	점이 너무 많을 때 max points와 spike preserve를 조정합니다.
Run Query	x/y field가 있고 y field가 숫자인지 확인합니다.

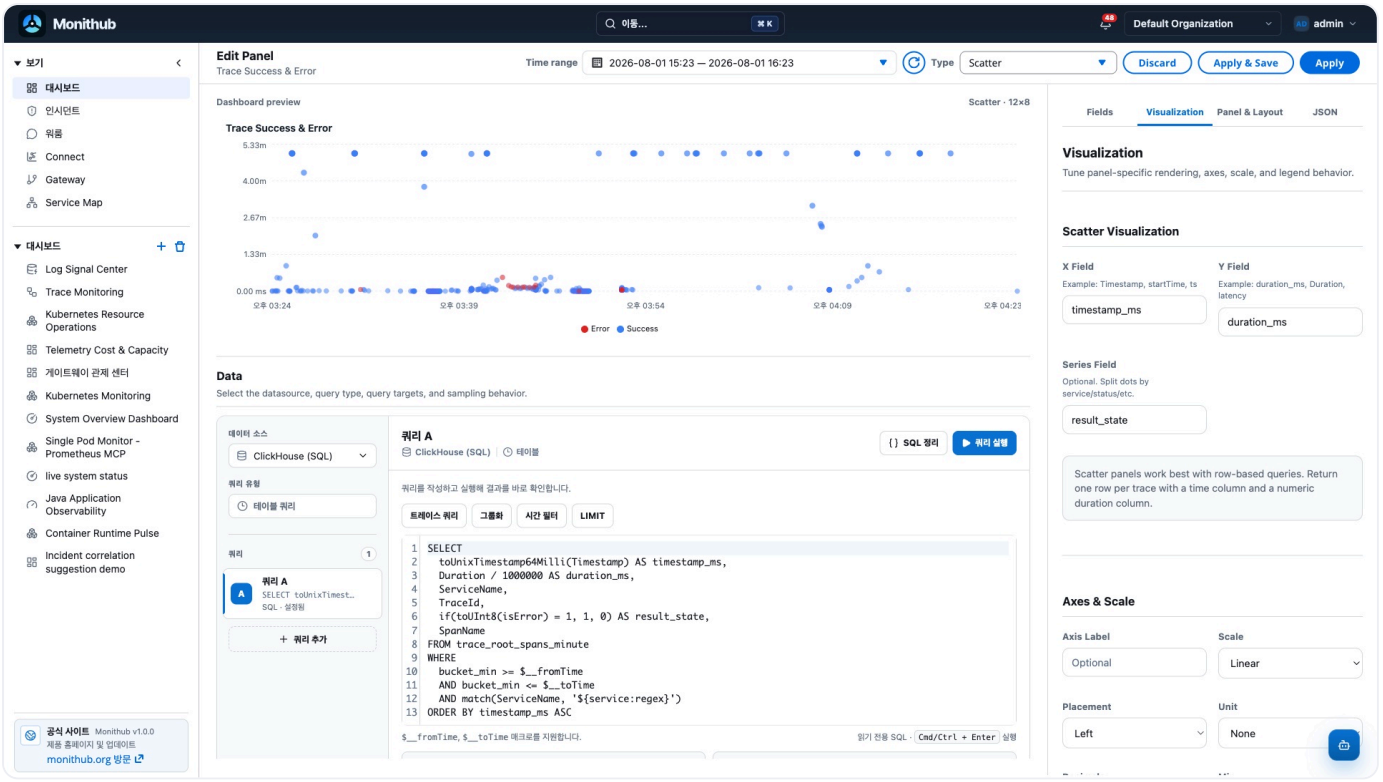
Fields



Fields에서는 y값 tooltip과 축 해석 기준을 맞춥니다. duration이면 `ms`, size면 `bytes`처럼 단위를 먼저 정리합니다.

설정	역할
Unit / Decimals	y값 tooltip과 축 단위의 기본 기준입니다.
Min / Max	y축 범위와 outlier 표시를 조정합니다.
Threshold Steps	점 색상이나 상태 해석 기준으로 쓸 수 있습니다.
Value Mappings	상태값을 tooltip에서 사람이 읽는 문구로 바꿉니다.

Visualization



Visualization에서는 row의 어떤 column을 점의 좌표와 그룹으로 쓸지 정합니다.

설정	역할
X Field	x축 field입니다. 예: `Timestamp`, `startTime`, `ts`
Y Field	y축 numeric field입니다. 예: `duration_ms`, `Duration`, `latency`
Series Field	점 색상/그룹을 나눌 field입니다. 예: `ServiceName`, `status`
Axes & Legend	축 label, scale, legend 표시 여부를 조정합니다.

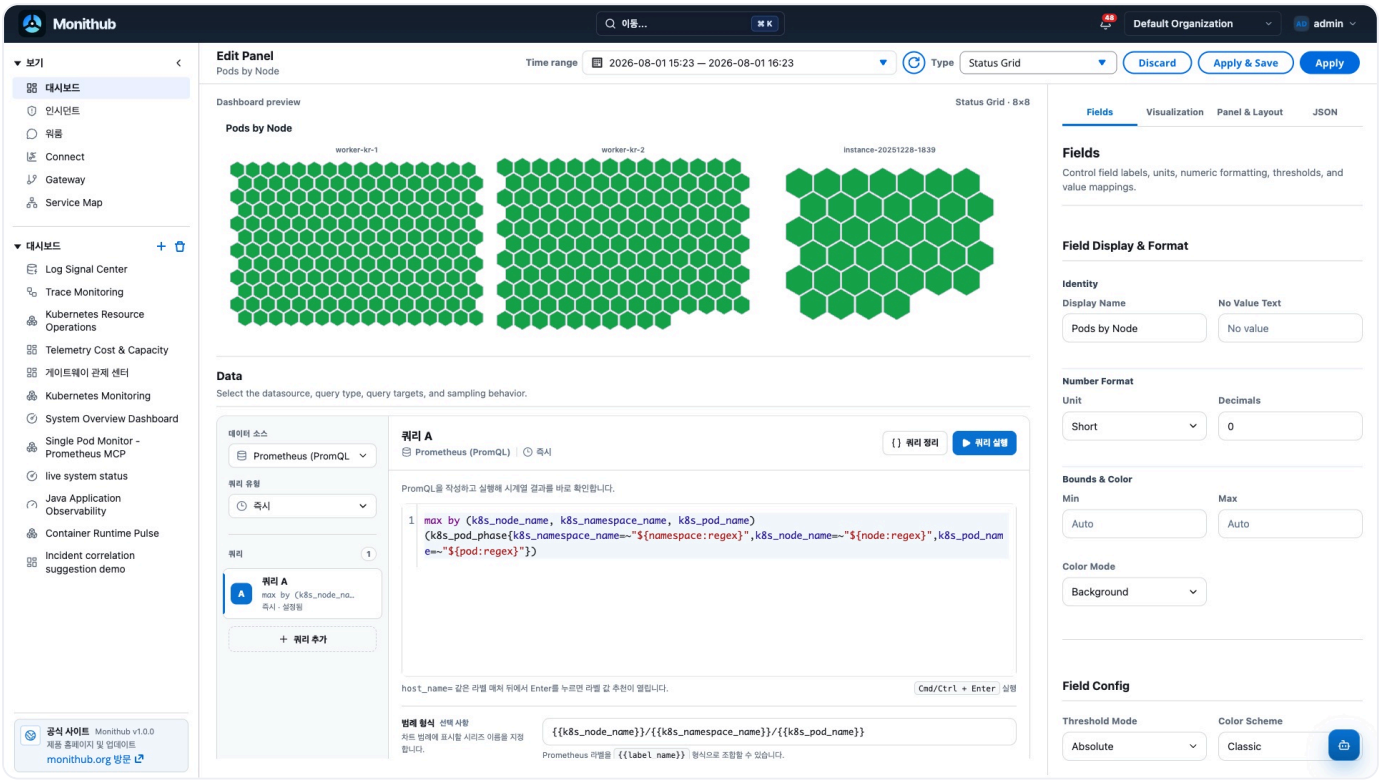
확인 순서

Data에서 row grain과 x/y field를 확인하고, Fields에서 y값 단위를 맞춘 뒤, Visualization에서 field mapping을 지정합니다.

Status Grid 패널

Status Grid는 여러 대상의 정상/비정상 상태를 빠르게 스캔하는 패널입니다. 예시는 `Pod CPU Status Grid` 편집 화면입니다.

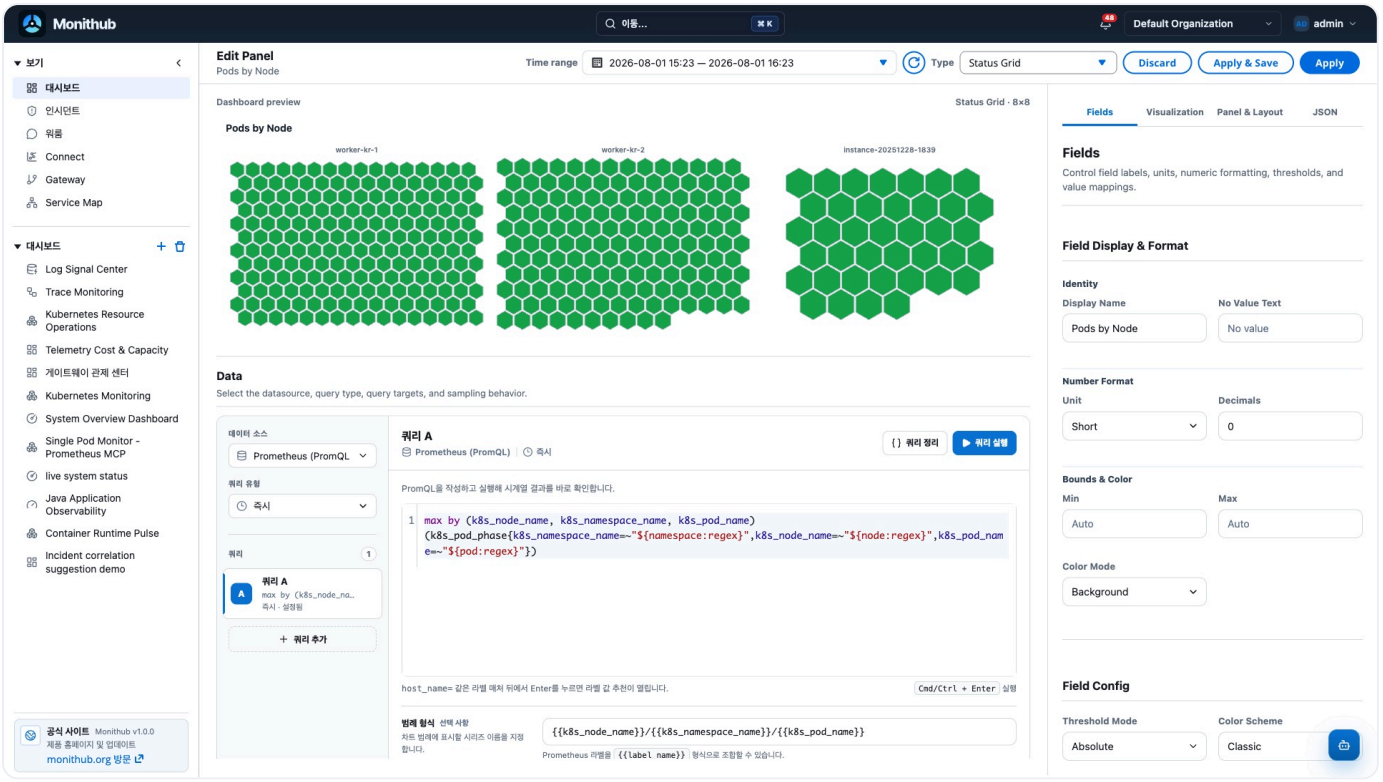
Data



Data에서는 대상 하나가 타일 하나로 나오도록 series label이나 row field를 정리합니다. grouping을 쓸 계획이라면 legend 이름에 `namespace/pod`, `service/instance`처럼 일정한 구분자를 둡니다.

설정	역할
Datasource	상태 metric이 있는 Prometheus나 ClickHouse datasource를 선택합니다.
Query type	현재 상태만 필요하면 `Instant`, 최근 구간 대표값이 필요하면 `Range`를 사용합니다.
Queries	여러 상태 query를 넣을 수 있지만 타일 이름이 겹치지 않게 정리합니다.
Legend Format	group separator로 나눌 이름 규칙을 정합니다.
Run Query	타일 수, label 중복, 비어 있는 series 여부를 확인합니다.

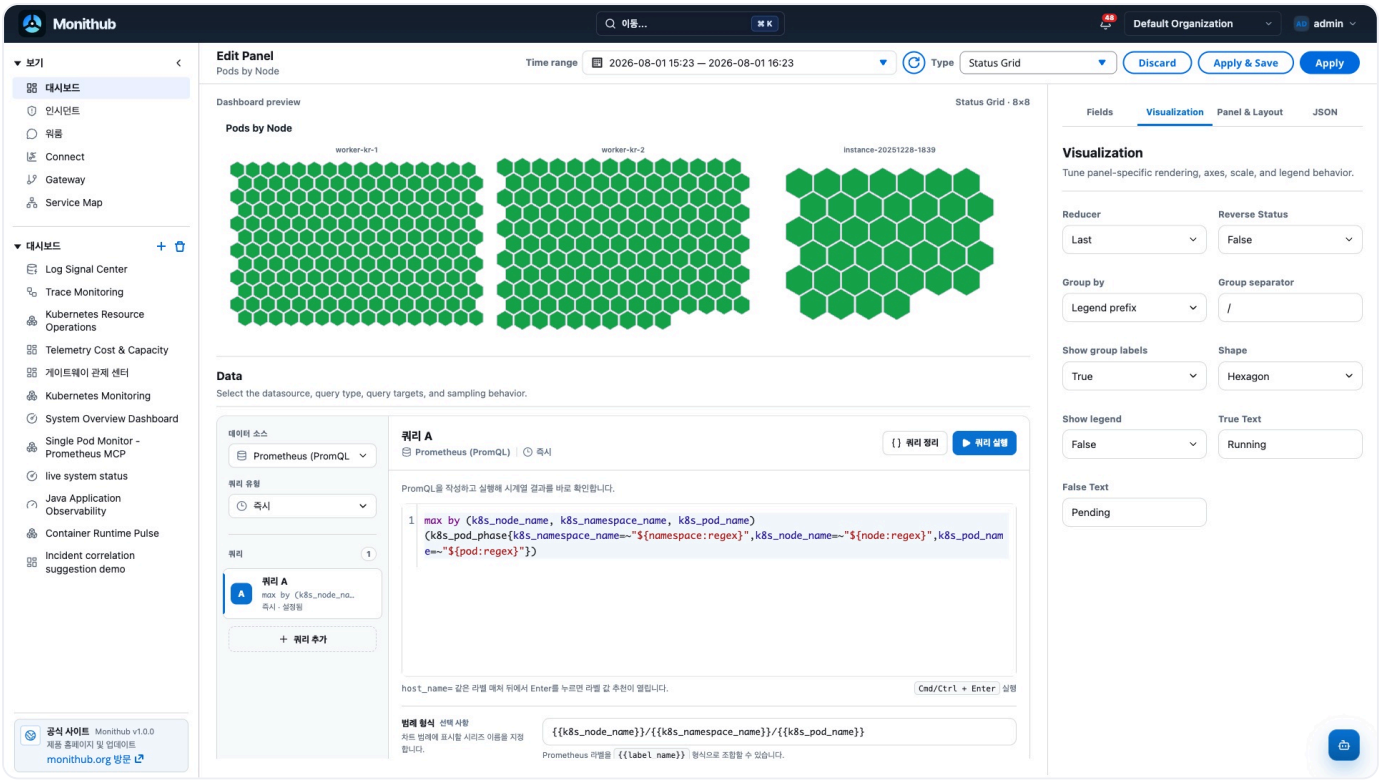
Fields



Fields에서는 상태값을 사람이 읽을 수 있는 문구와 색상 기준으로 바꿉니다. Status Grid는 unit보다 threshold와 value mapping 의미가 더 중요합니다.

설정	역할
Display Name / No Value Text	타일 이름과 데이터가 없을 때의 문구를 정합니다.
Unit / Decimals	숫자 상태값을 그대로 보여줄 때의 표시 기준입니다.
Threshold Steps	정상/비정상 판단 기준입니다. 예를 들어 0은 정상, 1 이상은 장애처럼 둡니다.
Value Mappings	`0`, `1`, boolean 값을 `Up`, `Down`, `Degraded` 같은 문구로 바꿉니다.
Color Scheme	threshold 의미와 타일 색상이 맞는지 확인합니다.

Visualization



Visualization에서는 여러 sample을 하나의 상태로 줄이는 방법, 타일 grouping, 타일 모양과 상태 문구를 정합니다.

설정	역할
Reducer	여러 sample 중 상태 판단에 사용할 대표값입니다.
Reverse Status	1이 정상인지 비정상인지 metric 의미가 반대일 때 뒤집습니다.
Group by	legend prefix를 기준으로 타일을 그룹화합니다.
Group separator	prefix를 나눌 구분자입니다. 기본은 `/`입니다.
Show group labels	그룹 이름 표시 여부입니다.
Shape	hexagon 또는 rounded tile 모양을 선택합니다.
Show legend	상태 색상 범례 표시 여부입니다.
True Text / False Text	true/false 상태 문구입니다.

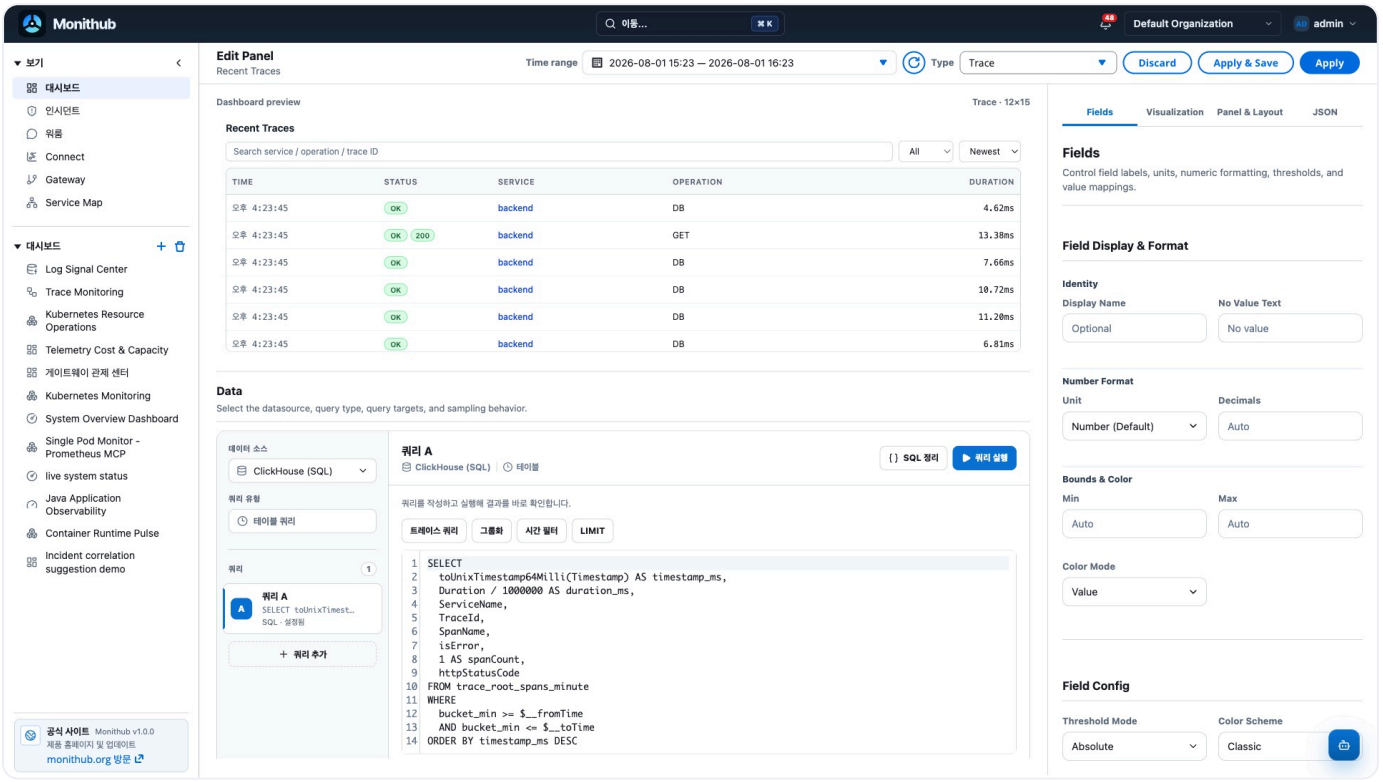
확인 순서

Data에서 타일 단위와 legend 구분자를 맞추고, Fields에서 값과 색상 의미를 정한 뒤, Visualization에서 reducer와 reverse status를 함께 확인합니다.

Trace 패널

Trace 패널은 trace 목록과 요약을 보여줍니다. 일반 PromQL 패널보다 ClickHouse trace 데이터와 더 밀접합니다.

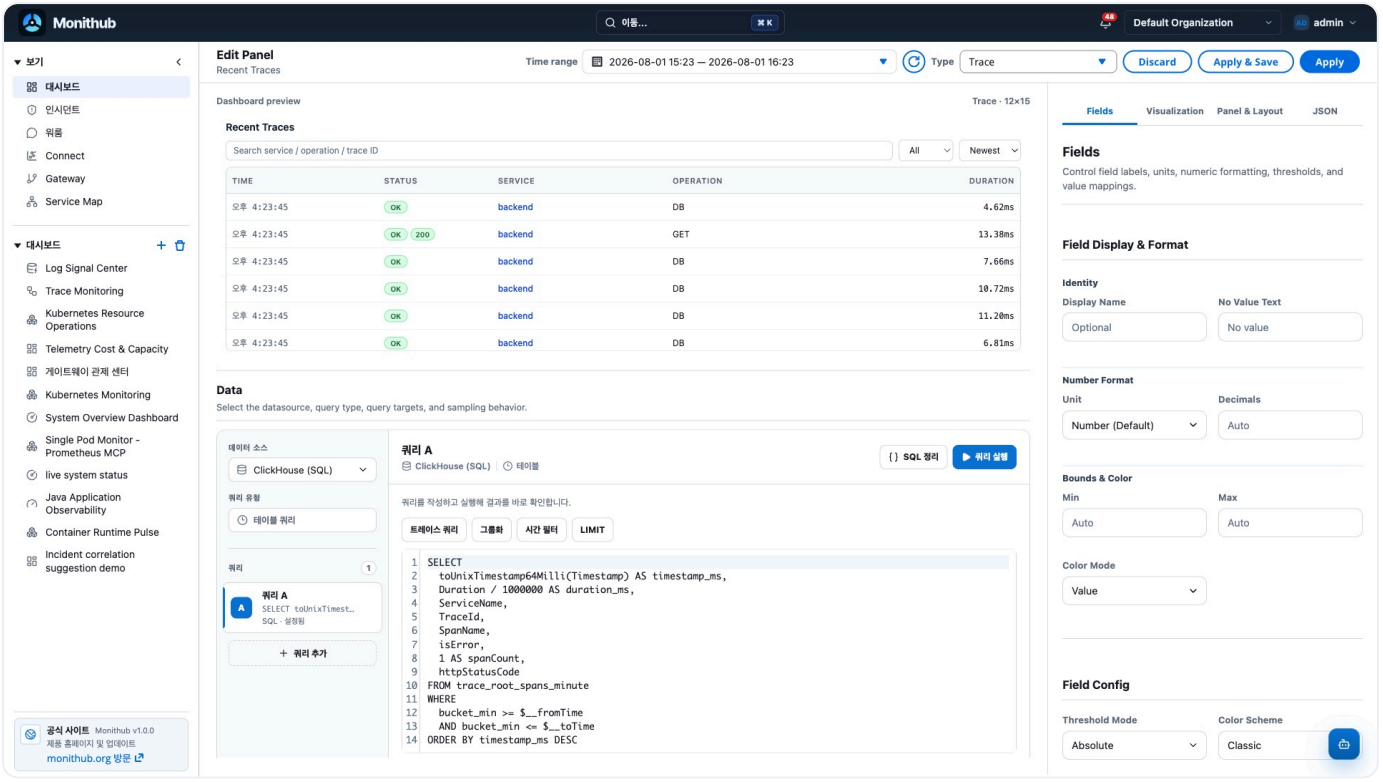
Data



Data에서는 trace 목록을 구성할 column을 안정적으로 반환해야 합니다. `trace_id`, service, operation, duration, status, timestamp 계열 column이 빠지면 Visualization에서 목록을 제대로 구성하기 어렵습니다.

설정	역할
Datasource	보통 `ClickHouse (SQL)`을 사용합니다.
Query editor	trace id, service, operation, duration, status, timestamp column을 반환합니다.
Time Filter	`$__fromTime`, `$__toTime` 매크로로 편집 화면 time range와 맞춥니다.
Trace Query / Group By	trace 목록 또는 service별 집계 template을 빠르게 넣을 때 사용합니다.
Run Query	column 이름과 duration 단위가 Trace 설정과 맞는지 확인합니다.

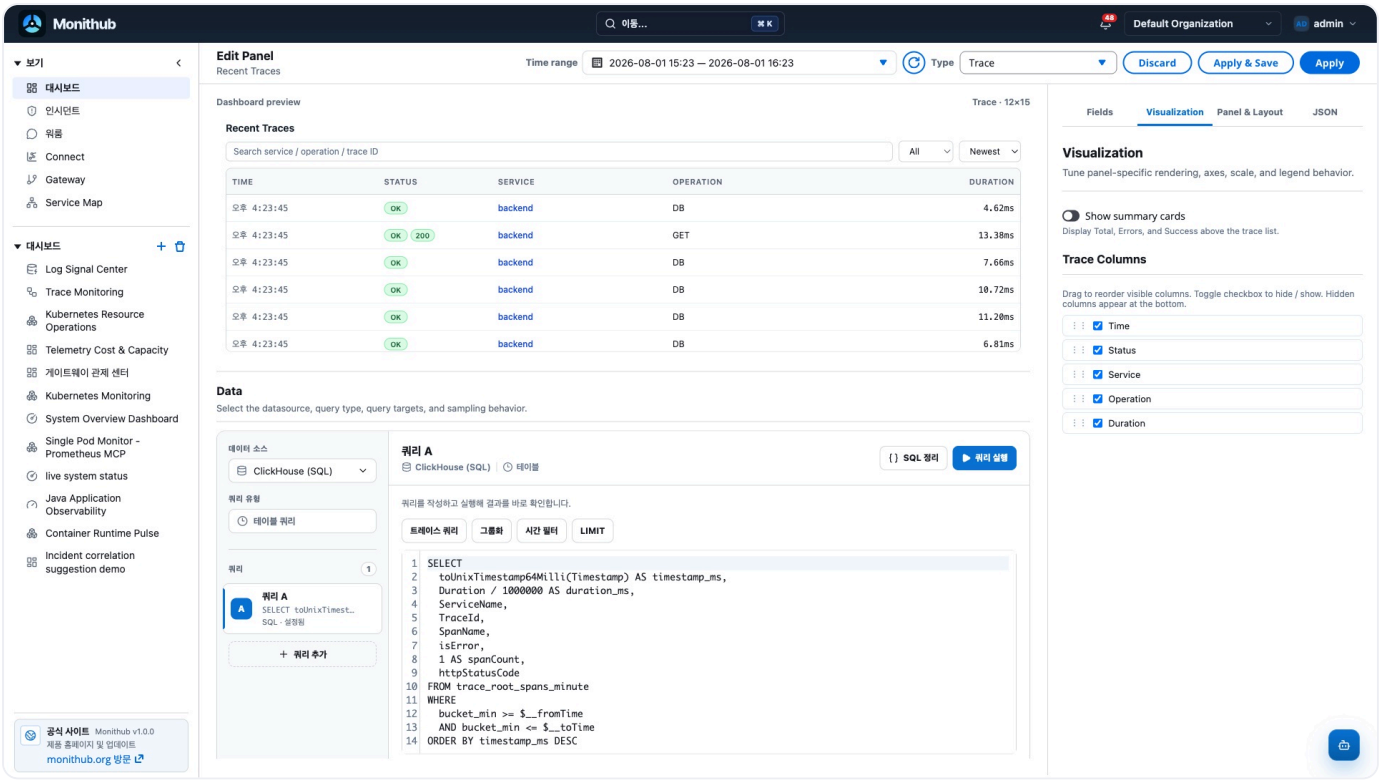
Fields



Fields에서는 trace duration, status code, 빈 값 표시 기준을 정합니다. 목록 column이 많아도 값의 단위와 상태 문구는 여기서 일관되게 맞춥니다.

설정	역할
Display Name / No Value Text	column 이름과 비어 있는 status, operation 문구를 정합니다.
Unit / Decimals	trace duration이 ns/us/ms 중 무엇인지 query와 맞춥니다.
Threshold Steps	duration이나 error count를 상태 색상으로 해석할 때 사용합니다.
Value Mappings	status code를 `Error`, `OK` 같은 문구로 바꿉니다.
Color Scheme	status나 duration 색상이 운영 의미와 맞는지 확인합니다.

Visualization



Visualization에서는 summary card와 trace list column을 정합니다. 장애 triage 화면에서는 status, service, operation, duration처럼 판단 근거가 되는 column을 앞쪽에 둡니다.

설정	역할
Show summary cards	Total, Errors, Success 같은 summary card 표시 여부입니다.
Trace Columns	trace list column 표시 여부와 순서를 정합니다.
Drag to reorder	visible column을 drag해서 순서를 바꿉니다.
Checkbox	column을 숨기거나 다시 표시합니다. 최소 하나의 visible column은 유지합니다.

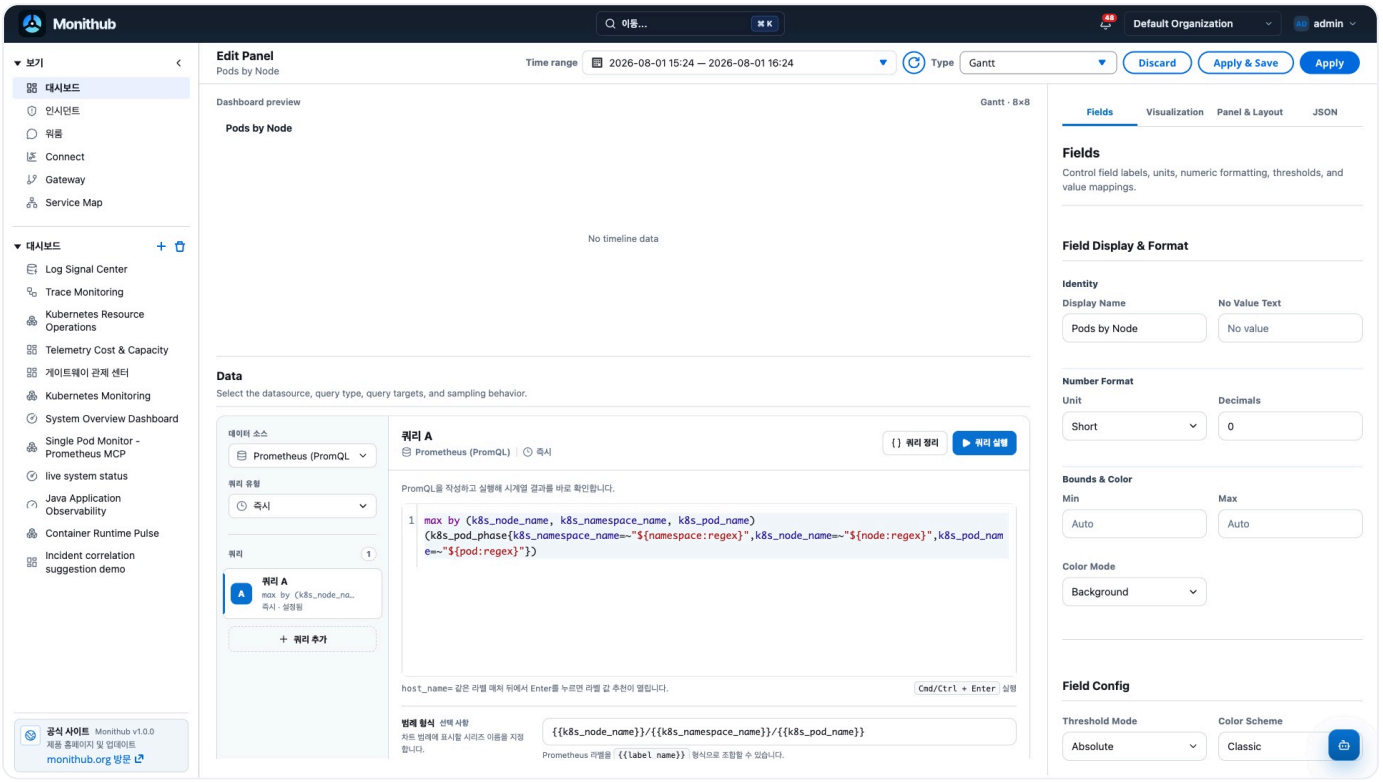
확인 순서

Data에서 trace 목록 column을 확인하고, Fields에서 duration 단위와 status 문구를 맞춘 뒤, Visualization에서 summary card와 column 순서를 줄여 읽기 쉽게 만듭니다.

Gantt 패널

Gantt는 시작/종료/기간이 있는 row를 시간 막대로 보여줍니다. span, 배치 흐름, 작업 실행 시간처럼 길이와 순서가 중요한 데이터를 볼 때 사용합니다.

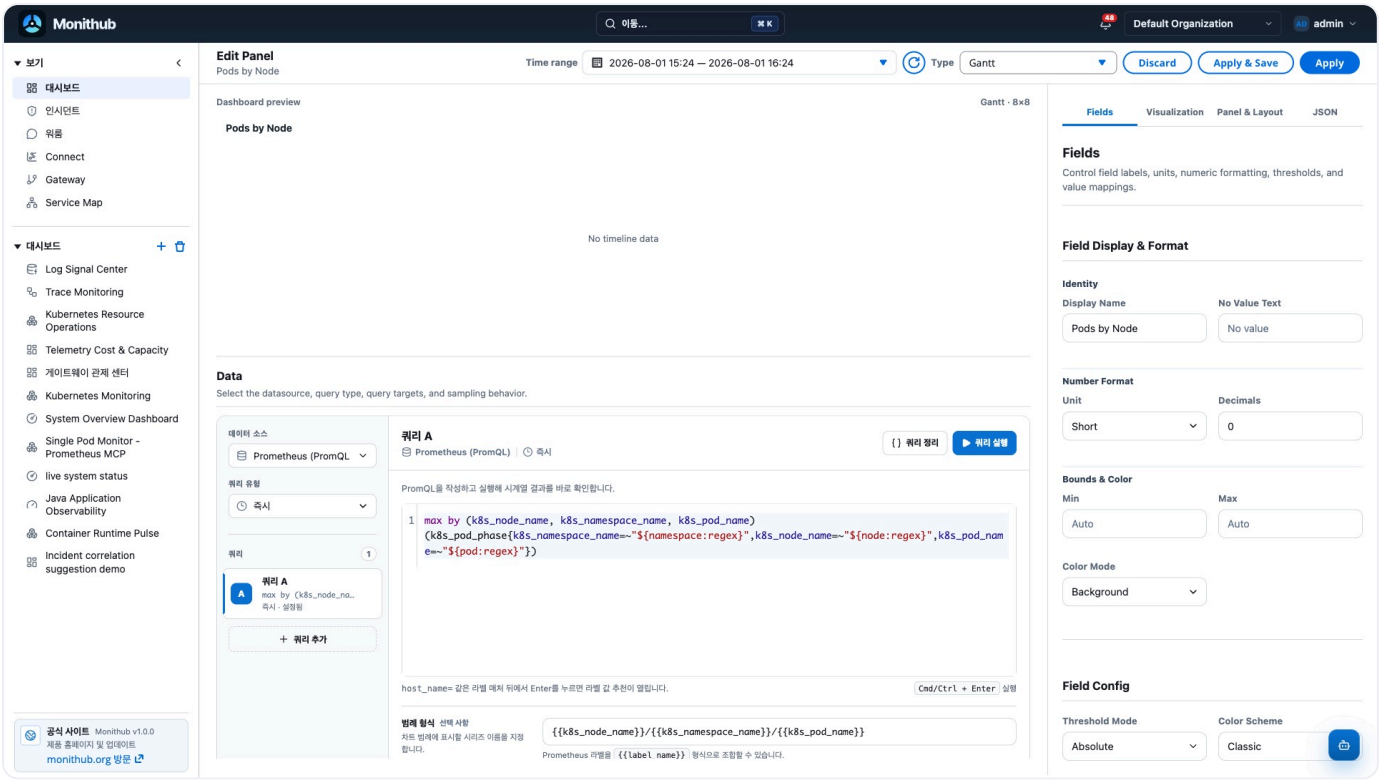
Data



Data에서는 각 row가 하나의 작업, span, 배치 단위가 되도록 query를 구성합니다. start/end 또는 start/duration, label field가 함께 반환되어야 합니다.

설정	역할
Datasource	trace span, 배치 작업, 이벤트 timeline은 ClickHouse SQL과 잘 맞습니다.
Query editor	start, end 또는 start, duration, label field를 반환합니다.
Time Filter	편집 time range와 query 조건을 맞춥니다.
Run Query	시간 단위가 ns/ms/s/unix/ISO 중 무엇인지 확인합니다.
JSON	UI에 보이지 않는 schema mapping이 있으면 함께 확인합니다.

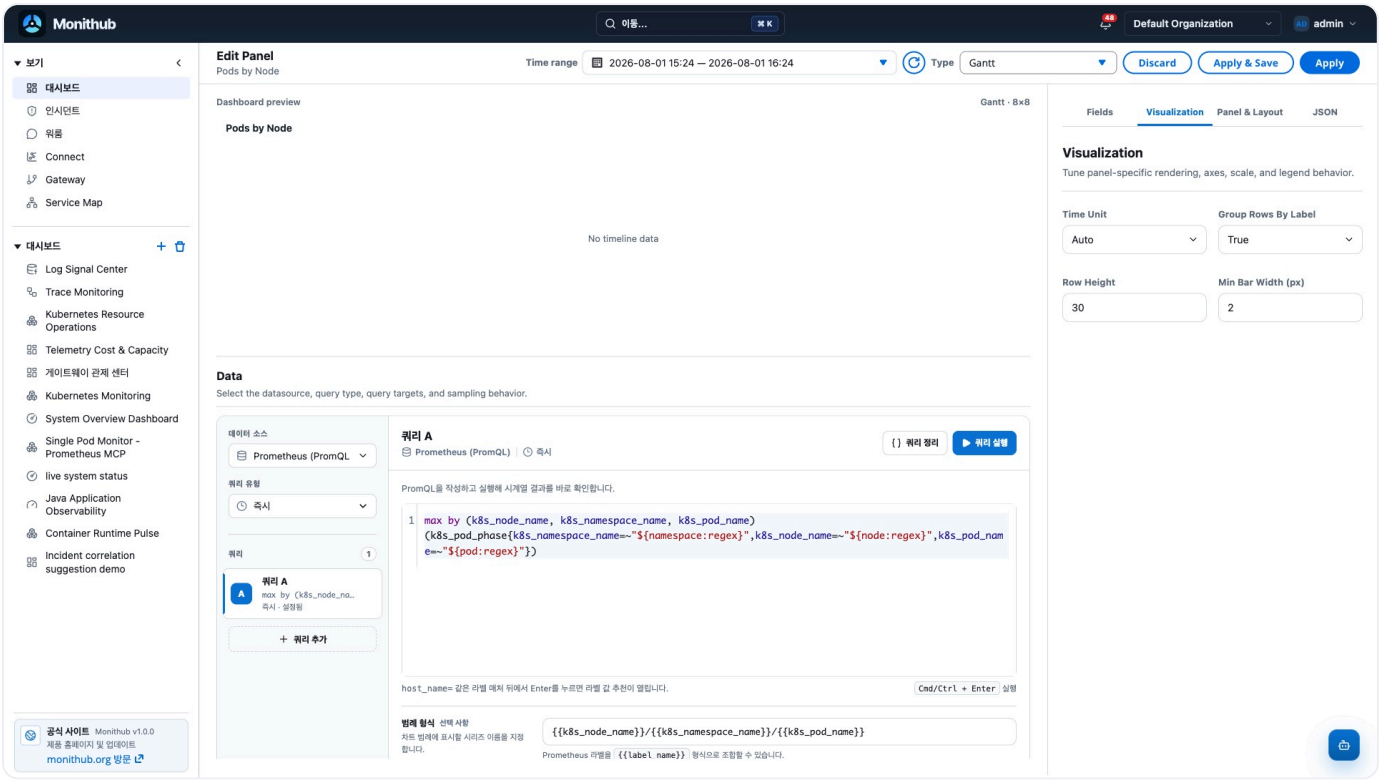
Fields



Fields에서는 duration tooltip과 숫자 표시 기준을 맞춥니다. 긴 작업이나 실패 상태를 색으로 구분하려면 threshold도 함께 정합니다.

설정	역할
Display Name / No Value Text	label이나 end 값이 비어 있을 때 표시할 문구를 정합니다.
Unit / Decimals	duration tooltip과 숫자 표시 기준입니다.
Threshold Steps	긴 작업이나 실패 상태를 색상 기준으로 표시할 때 사용합니다.
Value Mappings	상태 code를 사람이 읽는 문구로 바꿉니다.

Visualization



Visualization에서는 timestamp 단위와 row grouping, 막대 크기를 정합니다. query column 이름을 바꿨다면 JSON mapping도 함께 확인합니다.

설정	역할
Time Unit	timestamp 단위를 auto, ns, ms, s, unix ms, unix s, ISO date 중 고릅니다.
Group Rows By Label	같은 label의 row를 묶어 표시합니다.
Row Height	각 row의 높이입니다. 너무 낮으면 label이 읽히지 않습니다.
Min Bar Width (px)	아주 짧은 duration도 최소 몇 px로 보이게 할지 정합니다.

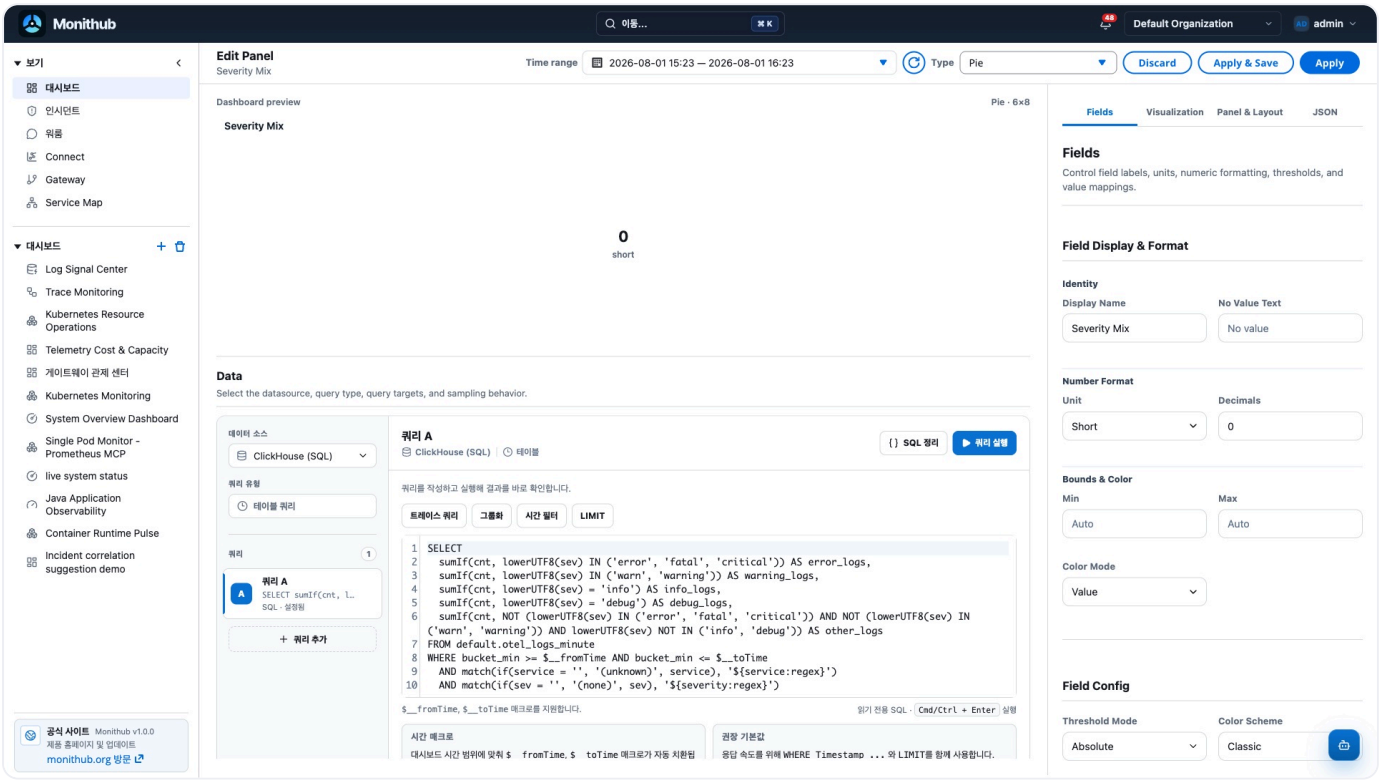
확인 순서

Data에서 start/end/duration/label field를 확인하고, Fields에서 duration 단위를 맞춘 뒤, Visualization에서 time unit과 row 높이를 실제 화면 폭에 맞춥니다.

Pie 패널

Pie는 적은 수의 category 비중을 볼 때 사용합니다. category가 너무 많거나 시간 흐름이 중요한 데이터에는 Time Series, Bar, Table이 더 적합합니다.

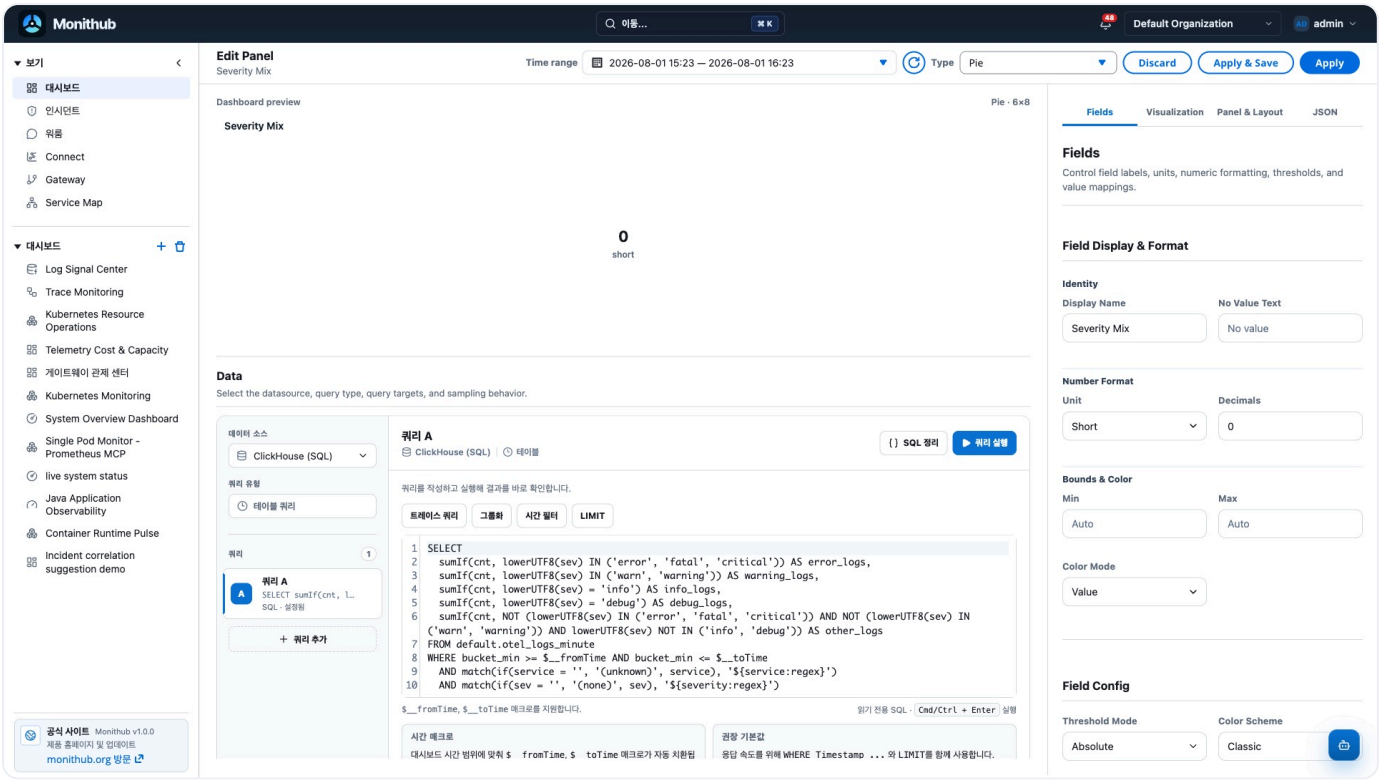
Data



Data에서는 category별 값이 적은 수로 반환되게 query를 줄입니다. 조각 수가 많아지면 pie보다 bar나 table이 더 읽기 쉽습니다.

설정	역할
Datasource	category 집계가 가능한 Prometheus나 ClickHouse datasource를 선택합니다.
Query type	category별 현재 비중이면 `Instant`나 ClickHouse 집계 query를 사용합니다.
Query editor	`GROUP BY service`, `sum by (status)`처럼 category 기준을 query에서 먼저 줄입니다.
Legend Format	category 이름이 길면 legend가 읽기 어려우므로 짧게 정리합니다.
Run Query	category가 너무 많거나 값이 0에 몰리지 않는지 확인합니다.

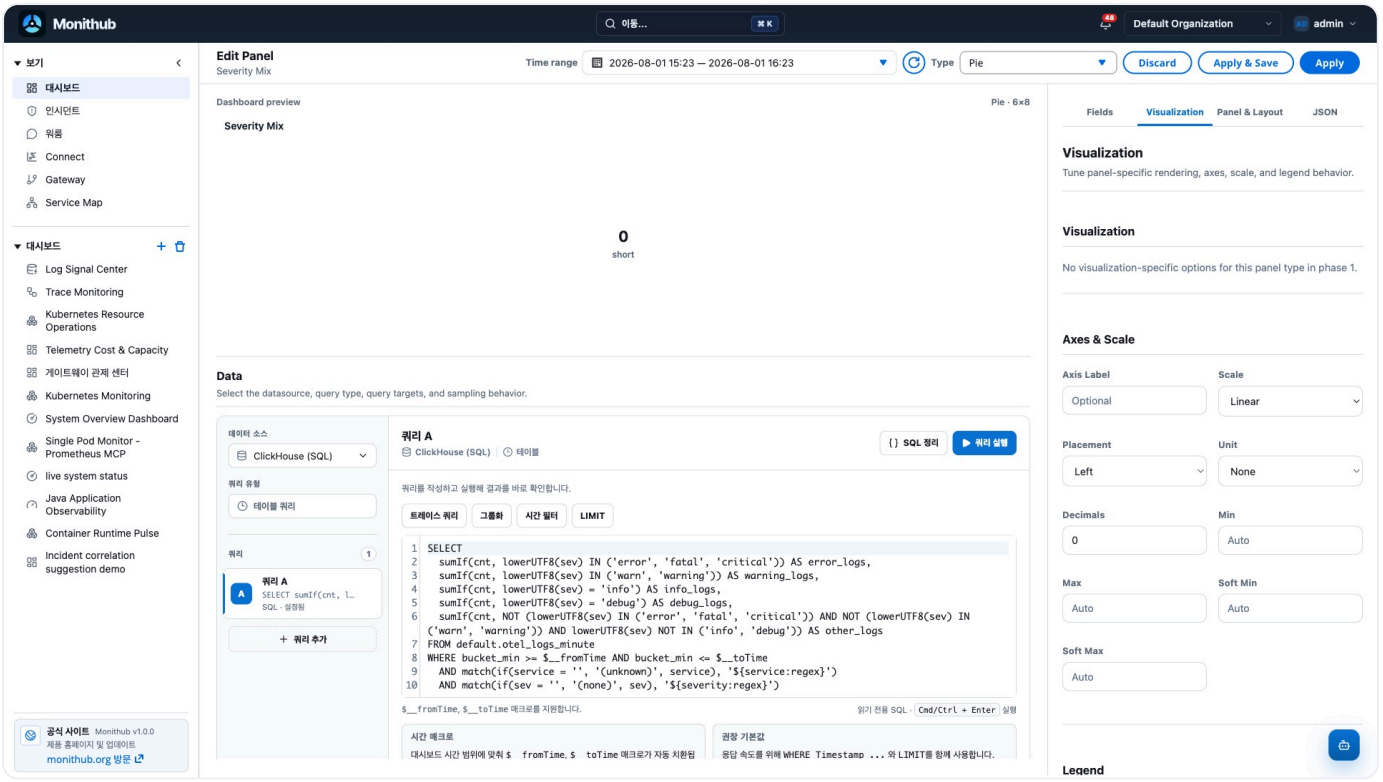
Fields



Fields에서는 조각 tooltip과 legend에 표시되는 값의 이름, 단위, 소수점, 상태 색상을 맞춥니다.

설정	역할
Display Name / No Value Text	값 field 이름과 빈 값 문구를 정합니다.
Unit / Decimals	조각 tooltip과 legend 값을 읽는 기준입니다.
Min / Max	값 범위를 명시해야 하는 경우 사용합니다.
Threshold Steps	pie 자체보다 tooltip/legend 상태 색상 해석에 영향을 줍니다.
Value Mappings	category 값이 숫자 코드일 때 표시 문구를 바꿉니다.

Visualization



Pie 편집 화면에 `No visualization-specific options for this panel type in phase 1.`가 표시되면 현재 UI에서 pie 전용 옵션을 직접 조정하지 않는 상태입니다. 이때는 Data의 category 수와 Fields의 단위, 그리고 Axes & Legend의 legend 표시를 먼저 조정합니다.

설정	역할
안내 문구	전용 옵션이 없는 상태인지 확인합니다.
Axes & Legend	legend 위치, 표시 여부, category 이름 가독성을 조정합니다.
Fields	값 단위와 소수점이 조각 tooltip에 반영됩니다.
Data	category 수와 비중 계산 기준을 결정합니다.

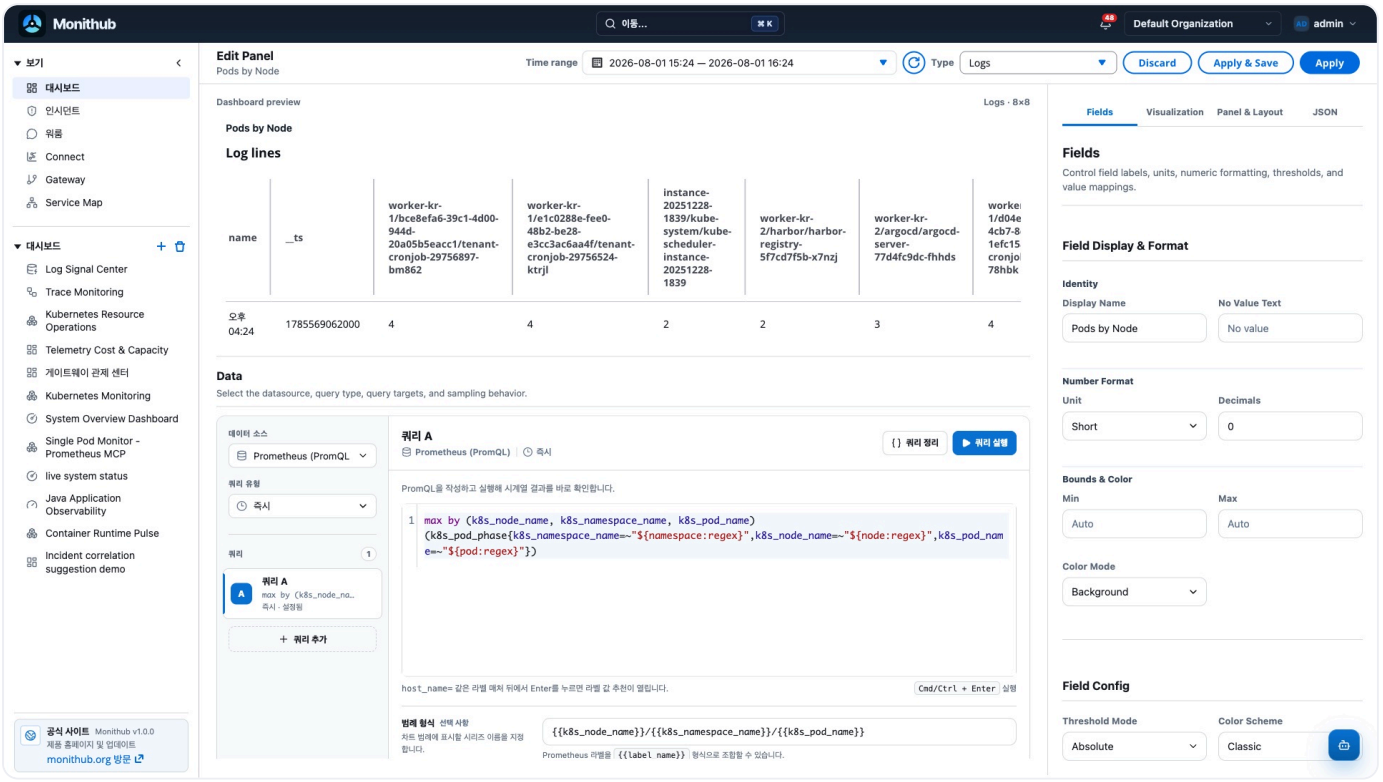
확인 순서

Data에서 category 수를 줄이고, Fields에서 unit과 decimals를 맞춘 뒤, Visualization에 전용 옵션이 없으면 Axes & Legend에서 legend를 정리합니다.

Logs 패널

Logs는 로그성 텍스트 결과를 표시할 때 사용합니다. query에서 결과를 좁히고, field formatting과 패널 크기로 읽기 좋은 로그 목록을 만듭니다.

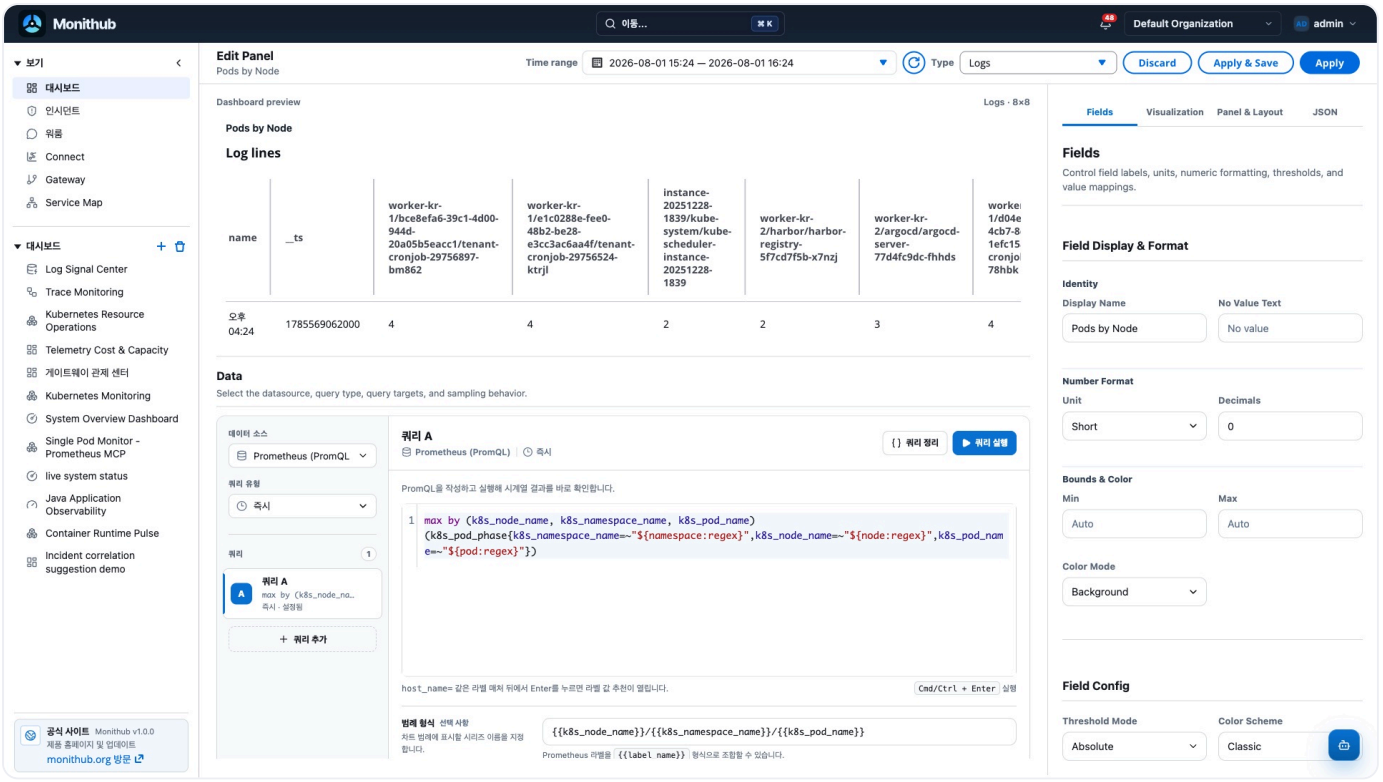
Data



Data에서는 시간 범위와 필터가 로그 row를 충분히 좁히는지 먼저 확인합니다. 로그 패널은 많은 row를 그대로 넣기보다 service, namespace, severity, time range 조건을 query에 넣는 편이 읽기 쉽습니다.

설정	역할
Datasource	로그 저장소가 ClickHouse에 있으면 SQL query를 사용합니다.
Query editor	service, namespace, severity, message, timestamp column을 반환합니다.
Time Filter	ClickHouse SQL에서는 `$__fromTime`, `$__toTime` 매크로로 편집 화면 time range와 맞춥니다.
LIMIT	운영 화면이 무거워지지 않도록 row 수를 제한합니다.
Run Query	반환 row와 message field가 실제 로그 표시 목적에 맞는지 확인합니다.

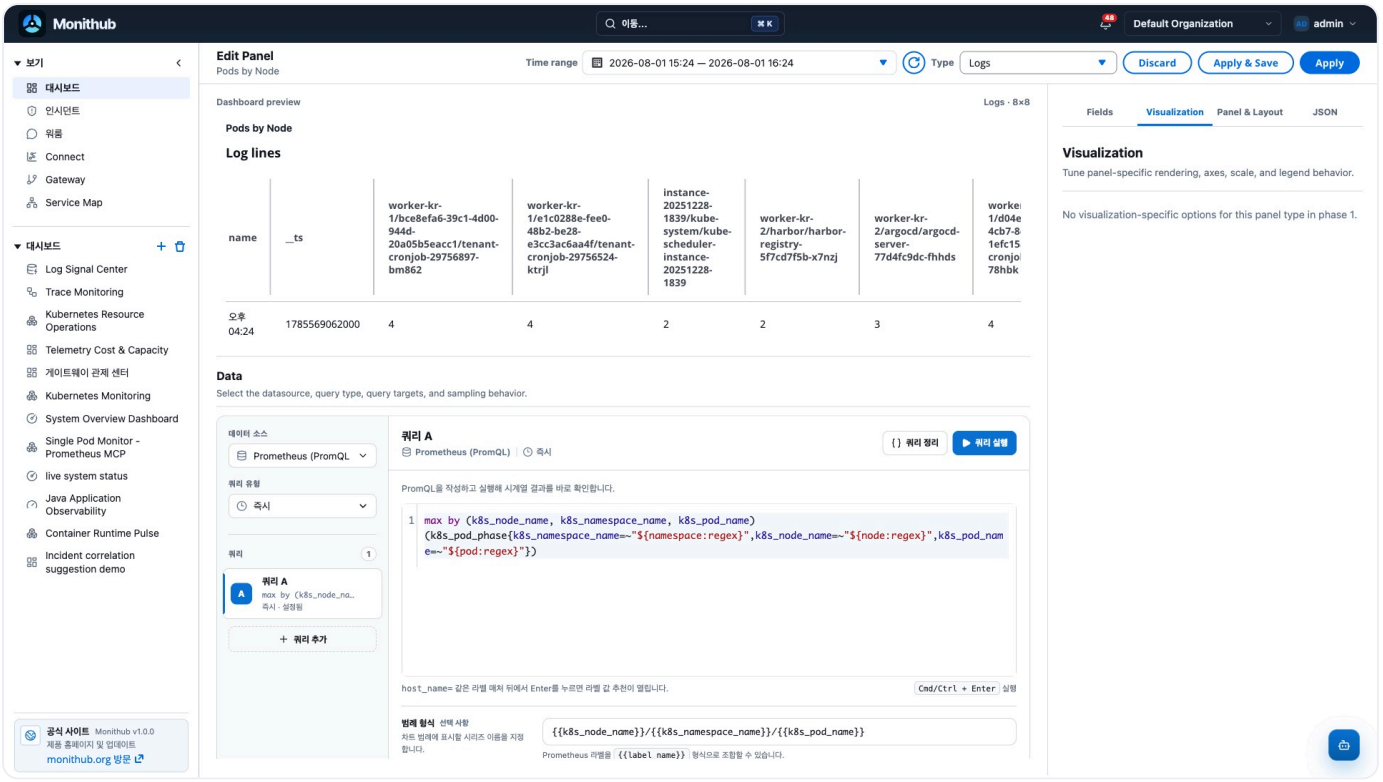
Fields



Fields에서는 로그 row의 field 이름, 빈 값 문구, severity 코드 표시를 정리합니다. 숫자 severity를 색상이나 문구로 바꾸면 목록을 빠르게 훑을 수 있습니다.

설정	역할
Display Name / No Value Text	로그 row의 field 이름과 빈 값 문구를 정합니다.
Unit / Decimals	숫자 field를 함께 표시할 때의 기준입니다.
Threshold Steps	severity 숫자값을 색상 기준으로 읽을 때 사용합니다.
Value Mappings	severity code나 상태값을 사람이 읽는 문구로 바꿉니다.

Visualization



Logs 편집 화면에 No visualization-specific options for this panel type in phase 1.가 표시되면 현재 UI에서 로그 전용 옵션을 직접 조정하지 않는 상태입니다. 이때는 Data에서 row를 줄이고 Fields에서 severity 표시를 맞춘 뒤, 패널 높이와 JSON 설정을 함께 확인합니다.

설정	역할
안내 문구	전용 옵션이 없는 상태인지 확인합니다.
Panel & Layout	로그 줄을 읽을 수 있을 만큼 높이와 폭을 확보합니다.
JSON	UI에 아직 노출되지 않은 로그 표시 옵션을 확인합니다.
Data	row 수와 message column을 안정화합니다.

확인 순서

Data에서 시간과 severity 조건을 좁히고, Fields에서 표시 문구를 맞춘 뒤, Visualization에 옵션이 없으면 Panel & Layout과 JSON에서 표시 방식을 확인합니다.

인시던트 및 워룸 개요

인시던트와 워룸은 문제 발견부터 공동 대응, 해결, 사후 정리까지 한 흐름으로 이어 주는 Monithub의 대응 공간입니다.

- `인시던트`에서는 무엇이 발생했고 어디까지 영향을 주는지 확인합니다.
- `워룸`에서는 담당자가 근거와 판단, 조치 결과를 함께 기록합니다.

두 화면을 오가며 상태와 대응 기록을 최신으로 유지하면, 새로 합류한 사람도 같은 맥락에서 업무를 이어갈 수 있습니다.

권장 대응 흐름

1. 인시던트 목록에서 최근에도 이상이 이어지는 항목을 찾습니다.
2. 담당자를 지정하고 `확인 처리`로 대응 시작을 알립니다.
3. 상세 화면에서 주요 대상, 영향 범위, 지속 시간과 연관 신호를 확인합니다.
4. 필요한 신호를 추가하고, 정탐·오탐·무시 여부를 검토합니다.
5. 영향이 있으면 `완화` 상태로 바꾸고 조치와 관찰 기준을 기록합니다.
6. 여러 사람이 함께 대응해야 하면 연결된 워룸에서 역할과 다음 행동을 나눕니다.
7. 문제가 안정되면 해결 조건을 확인한 뒤 `해결` 처리합니다.
8. 워룸 요약과 후속 작업, 필요한 백업을 남기고 채널을 닫습니다.

먼저 구분해야 할 세 가지 동작

동작	쓰는 상황	결과
오탐 근거 제거	특정 이상 신호가 잘못 감지되었고 현재 인시던트의 근거로 쓰지 않을 때	선택한 이상 근거만 현재 인시던트에서 제거합니다.
해결 또는 닫기	실제 대응이 끝났고 기록을 보존해야 할 때	인시던트와 워룸 기록을 남긴 채 수명 주기를 마칩니다.
인시던트 삭제	잘못 만들었거나 테스트용인 인시던트 자체를 없애야 할 때	연결된 근거와 대응 기록을 영구 삭제하고 워룸을 닫습니다.

일반적인 대응 종료에는 삭제가 아니라 `해결`과 `닫기`를 사용합니다.

문서 묶음

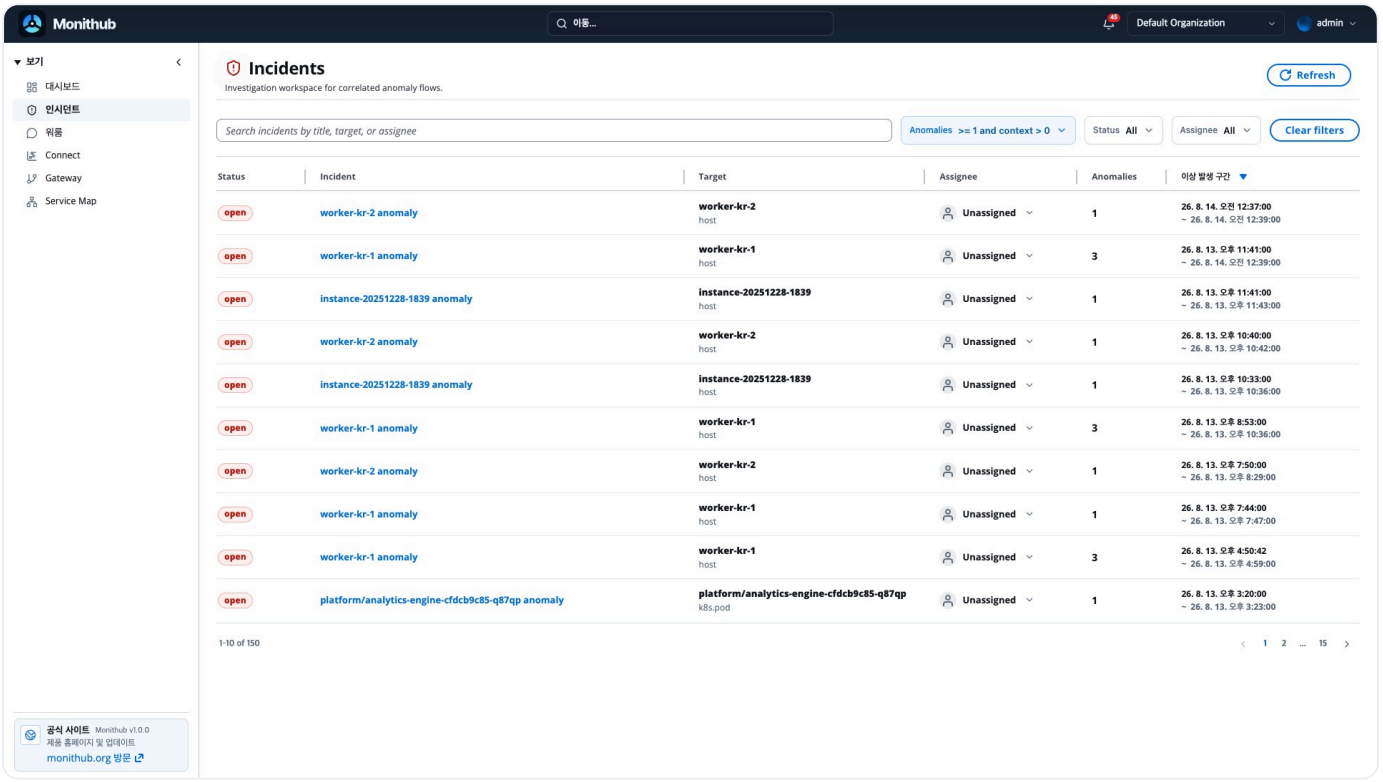
문서	확인할 내용
인시던트 확인	목록 필터, 담당자와 상태 관리, 연관 신호 추가, 오탐 근거 제거, 인시던트 삭제
워룸 협업	채널 선택, 협업 기록, AI 요약과 브리핑, 명령어, CSV·MPCA 백업
종료와 정리	해결 조건, 채널 닫기, 후속 작업과 기록 보존

처음 사용하는 경우 `인시던트 확인` → `워룸 협업` → `종료와 정리` 순서로 읽는 것을 권장합니다.

인시던트 확인

인시던트 화면에서는 대응할 문제를 고르고, 담당자와 상태를 정한 뒤, 판단에 필요한 신호를 한곳에서 검토합니다. 처음부터 원인을 단정하기보다 현재 영향과 시간 흐름을 먼저 확인합니다.

목록에서 대응할 문제 찾기

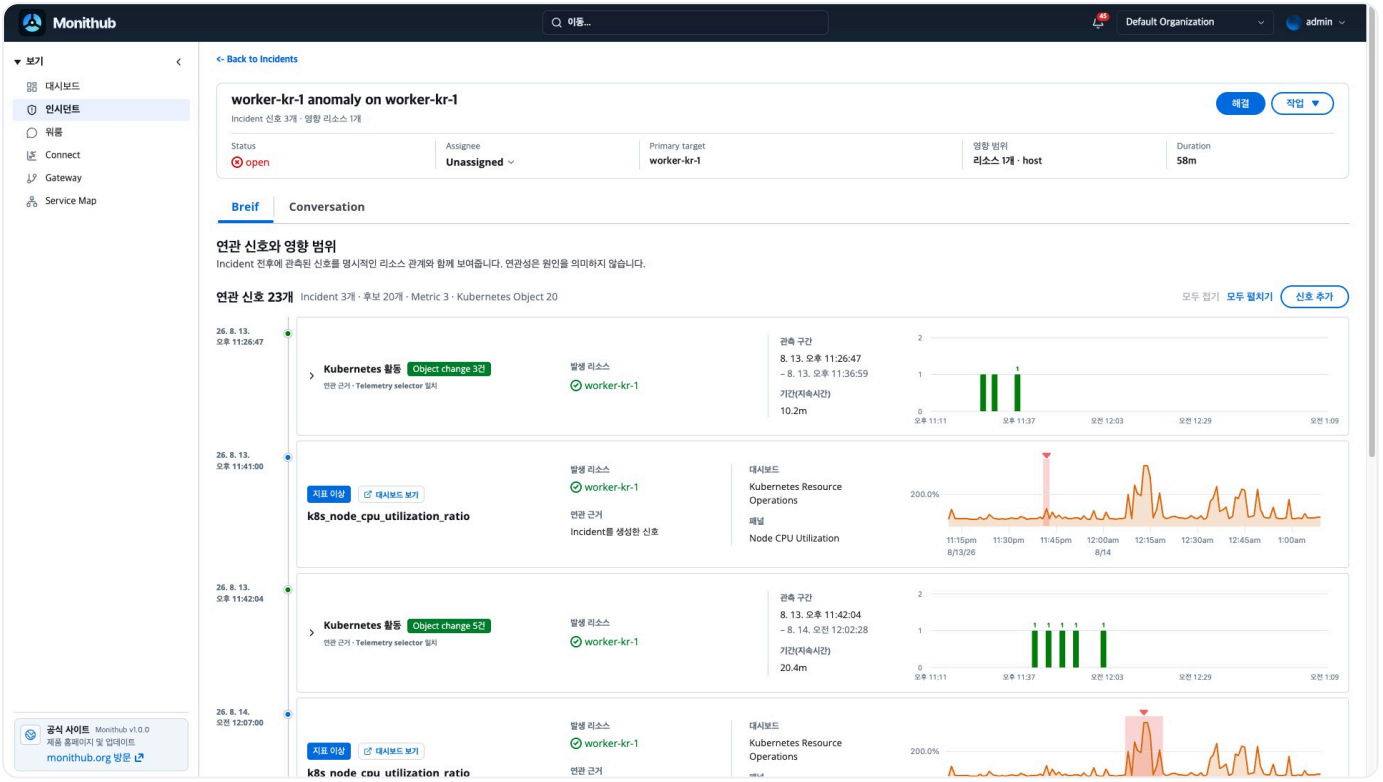


목록 상단의 검색과 필터를 함께 사용하면 대응이 필요한 항목을 빠르게 좁힐 수 있습니다.

기능	사용 방법
검색	인시던트 제목이나 대상 이름을 입력합니다.
이상 신호 수	이상 신호가 있는 인시던트만 보거나, 문맥 신호가 함께 있는 항목을 찾습니다.
상태	열림, 확인, 완화, 해결 등 현재 대응 단계로 좁힙니다.
담당자	내 담당 항목이나 미할당 항목을 찾습니다.
정렬	이상 발생 구간이나 최근 활동을 기준으로 우선순위를 정합니다.
필터 초기화	적용 중인 조건을 한 번에 지웁니다.

각 행에서 담당자를 바로 지정할 수 있습니다. 최근에도 이상이 이어지고 영향 대상이 중요하지만 담당자가 없는 항목을 우선 확인합니다.

상세에서 1차 판단하기

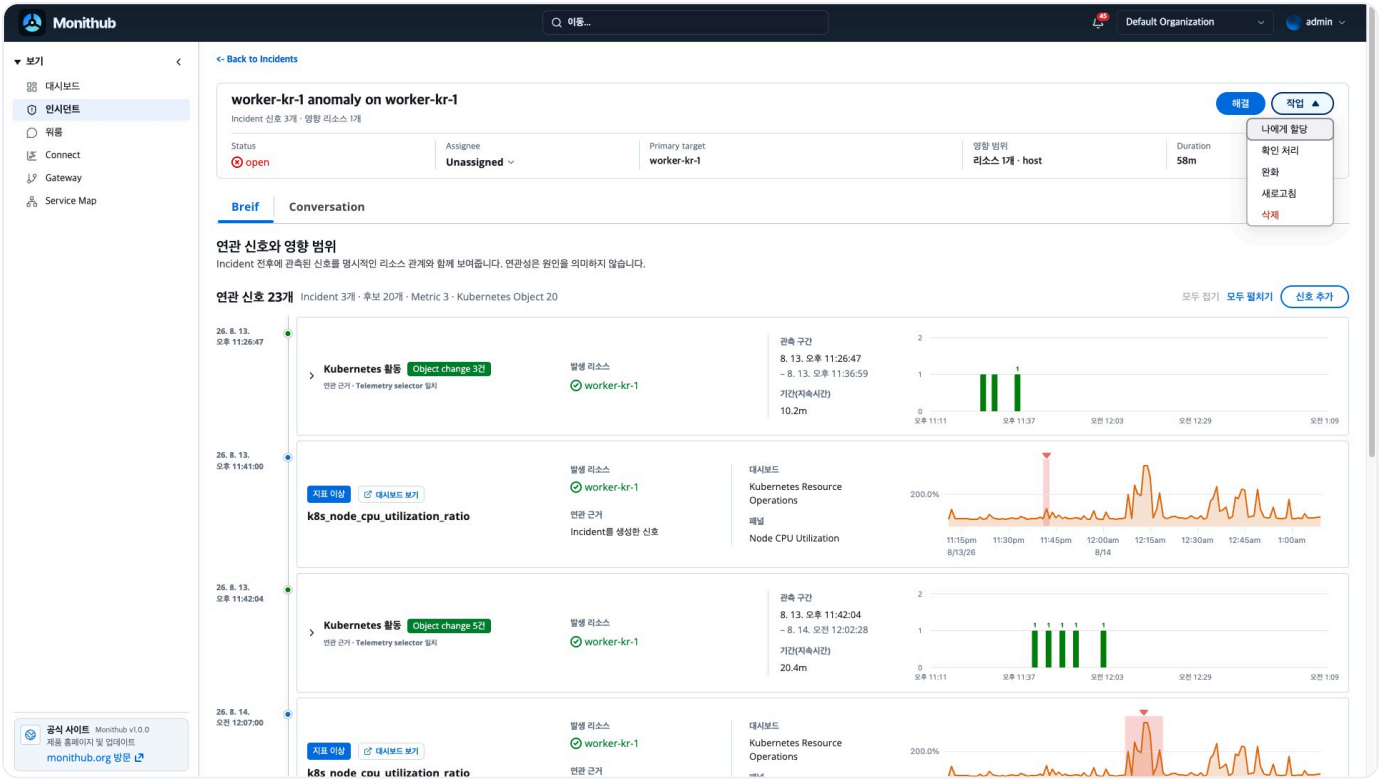


상세 화면을 열면 상단 요약부터 확인합니다.

영역	확인할 내용
상태	아직 열려 있는지, 확인 또는 완화 단계인지 봅니다.
담당자	현재 대응을 이끌 사람이 지정되어 있는지 확인합니다.
주요 대상	문제의 중심이 되는 호스트, 서비스, 워크로드를 확인합니다.
영향 범위	영향을 받는 리소스 수와 유형을 봅니다.
지속 시간	일시적인 흔들림인지 계속 이어지는 문제인지 판단합니다.
Brief	연관 신호와 영향 범위를 시간 순서로 검토합니다.
Conversation	연결된 대응 대화와 기록을 확인합니다.

연관 신호 영역에는 인시던트에 직접 연결된 근거와 후보 신호가 함께 표시됩니다. Kubernetes 활동, 메트릭 등 서로 다른 신호가 같은 시간대와 대상에서 반복되는지 비교합니다. 화면에 표시되는 연관성은 원인 확정을 뜻하지 않으므로, 조치 전에는 실제 영향과 변경 이력을 함께 확인합니다.

담당자와 상태 관리하기

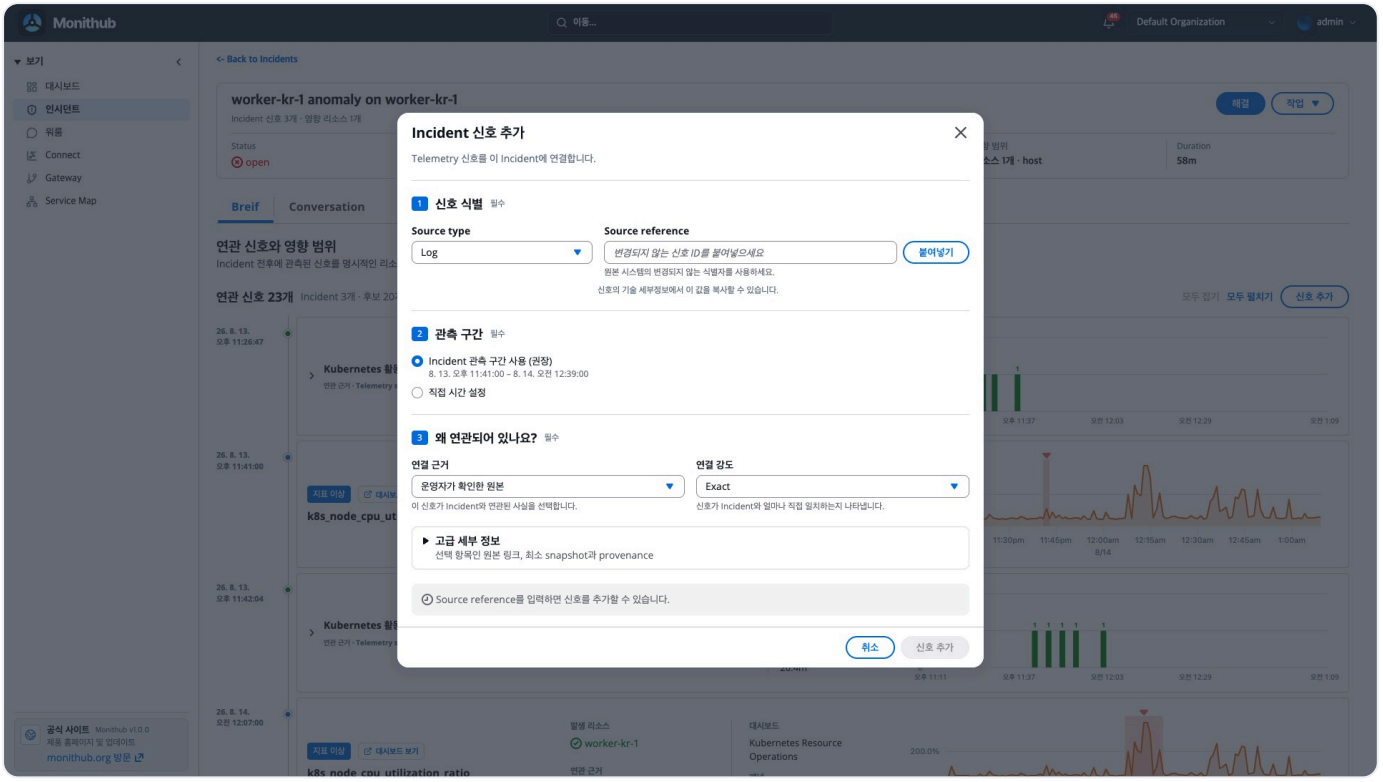


상단의 작업 메뉴에서 현재 상태에 맞는 동작을 선택합니다.

동작	쓰는 시점
나에게 할당	내가 대응을 맡을 때
확인 처리	누군가 문제를 확인하기 시작했음을 기록할 때
완화	영향이 계속되지만 임시 조치로 범위를 줄였을 때
새로고침	최신 상태와 연관 신호를 다시 불러올 때
해결	새 이상이 멈추고 서비스가 정상 범위로 돌아왔을 때
삭제	잘못 생성했거나 테스트용인 인시던트 전체를 없앨 때

상태를 바꿀 때는 워룸에 판단 근거와 조치 시각을 함께 남깁니다. 완화는 해결과 다릅니다. 임시 조치 뒤에도 관찰 기준을 충족할 때까지 대응을 이어갑니다.

필요한 신호 직접 추가하기



자동으로 묶이지 않은 근거가 있다면 연관 신호 영역의 `신호 추가`를 사용합니다.

1. 원본 신호의 `Source type`을 선택합니다.
2. 기술 세부정보에서 복사한 변경되지 않는 `Source reference`를 입력합니다.
3. 기본값인 인시던트 관측 구간을 사용하거나, 필요한 경우 시간을 직접 지정합니다.
4. 연결 근거와 연결 강도를 선택합니다.
5. 원본 링크나 provenance가 필요하면 `고급 세부 정보`에 보완합니다.
6. 내용을 확인한 뒤 `신호 추가`를 누릅니다.

단순히 시간대가 비슷하다는 이유만으로 신호를 추가하지 않습니다. 다른 운영자가 다시 검토해도 연결 이유를 이해할 수 있도록 근거와 강도를 실제 상황에 맞게 선택합니다.

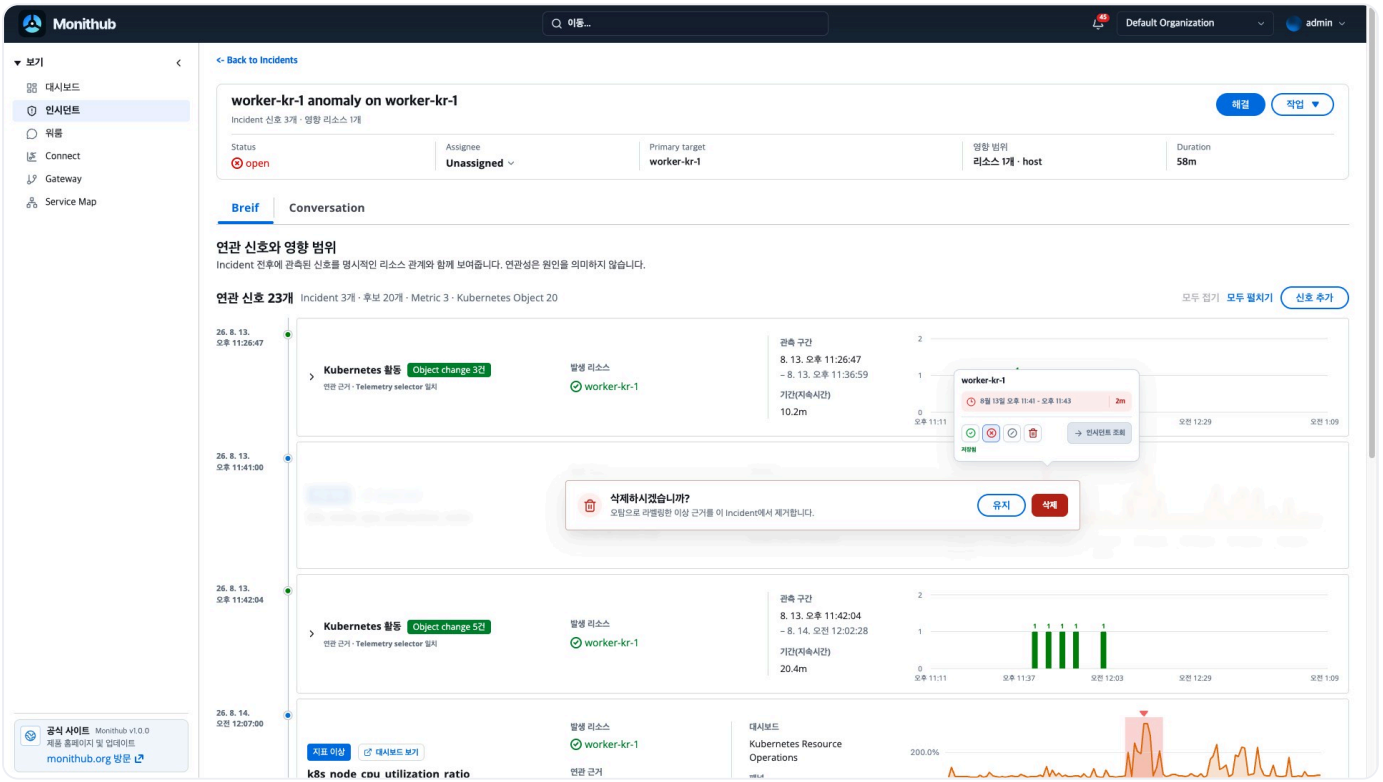
정탐·오탐·무시로 근거 분류하기

신호 카드의 선택 버튼을 누르면 해당 이상을 분류할 수 있습니다.

분류	의미
정탐	실제 이상이며 현재 판단에 유효한 근거입니다.
오탐	이상으로 감지됐지만 실제 문제를 나타내지 않는 근거입니다.
무시	이번 판단에서 사용하지 않되, 바로 삭제하지 않고 남겨 둘 근거입니다.

분류는 원인 확정이 아니라 운영자의 검토 결과입니다. 판단 이유를 워룸에 한 줄이라도 남기면 이후 튜닝과 회고에 도움이 됩니다.

오탐으로 표시한 근거 제거하기

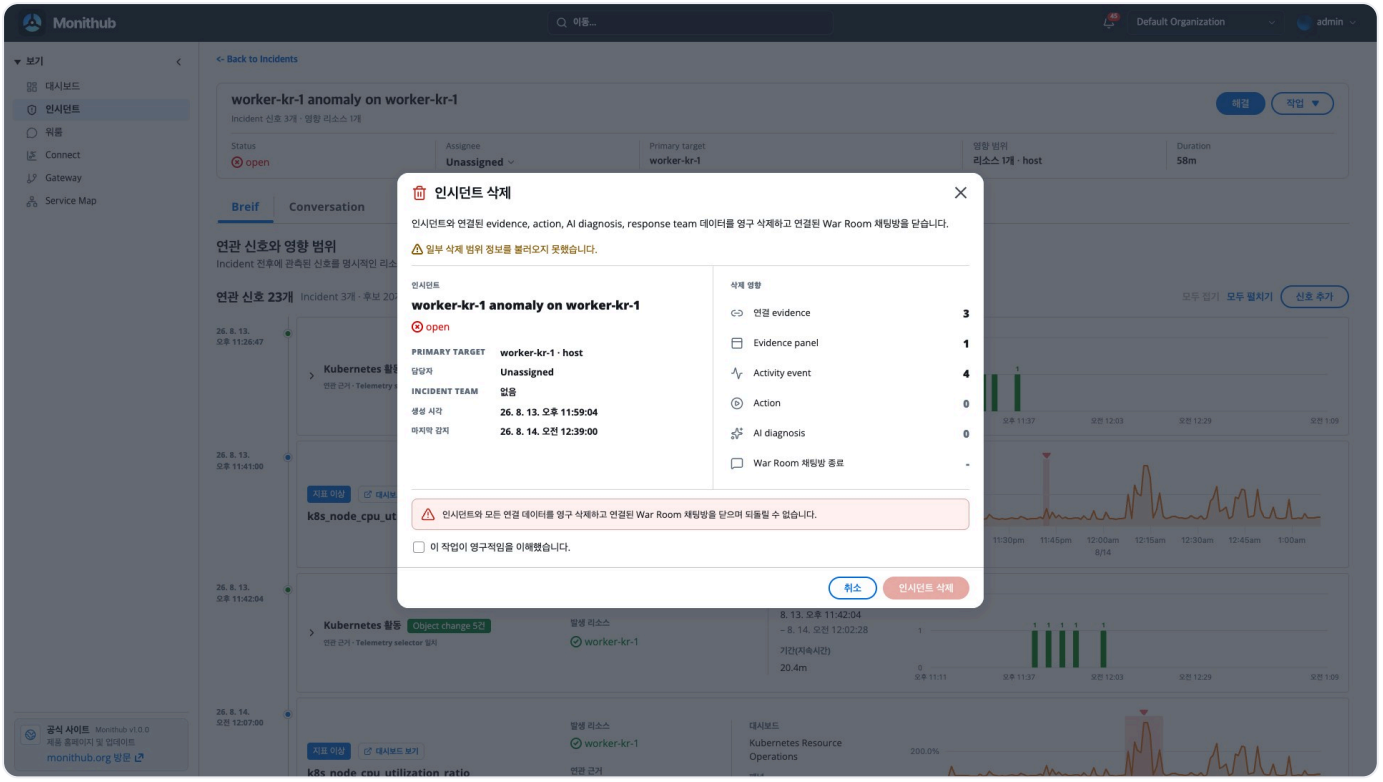


오탐으로 표시한 이상 근거를 현재 인시던트에서 제외하려면 다음 순서로 진행합니다.

1. 신호 카드의 선택 버튼을 엽니다.
2. 오탐을 선택합니다.
3. 휴지통 버튼을 누릅니다.
4. 확인 창의 설명을 읽고 삭제를 선택합니다.

이 동작은 **선택한 오탐 근거만 현재 인시던트에서 제거**합니다. 인시던트 자체를 삭제하는 기능이 아닙니다. 판단을 더 확인해야 한다면 유지를 선택합니다.

인시던트 전체 삭제하기



작업 → 삭제는 오탐 근거 제거와 범위가 다릅니다. 인시던트와 연결된 근거, 액션, AI 진단, 대응 팀 기록이 영구 삭제되고 연결된 워룸 채널도 닫힙니다.

삭제가 꼭 필요한 경우에만 다음 순서로 진행합니다.

1. 확인 창에 표시된 연결 데이터 수를 검토합니다.
2. 필요한 워룸 기록을 CSV 또는 MPCA로 먼저 보관합니다.
3. 영구 삭제 안내에 동의하는 체크박스를 선택합니다.
4. 인시던트 영구 삭제를 실행합니다.

정상적으로 대응을 마친 인시던트는 삭제하지 않습니다. 기록을 보존한 채 해결 또는 닫기 상태로 수명 주기를 마칩니다.

검토를 워룸 대응으로 잇기

담당자와 상태를 최신으로 맞춘 뒤, 연결된 워룸에 아래 내용을 남깁니다.

- 확인된 사실과 아직 검증되지 않은 가설
- 고객 또는 사용자 영향

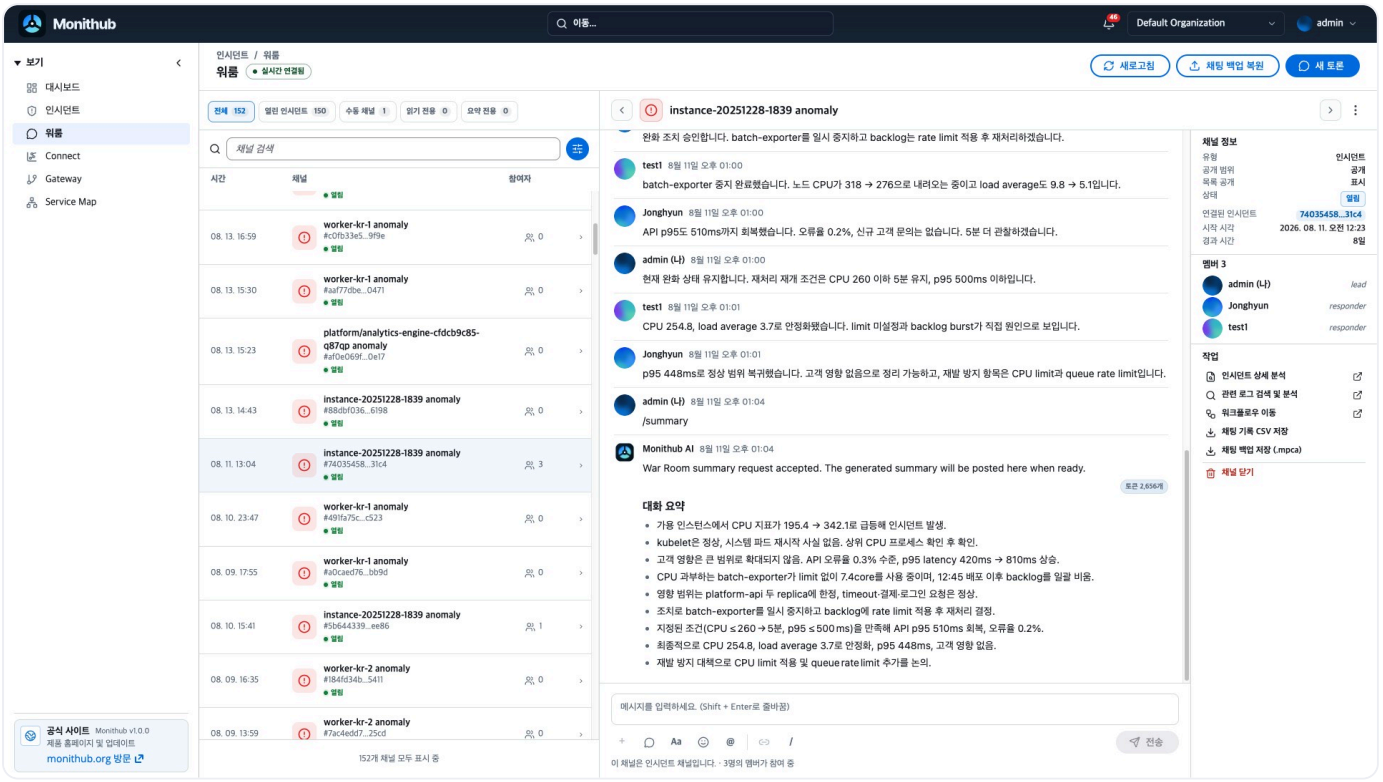
- 조치 담당자와 시작 시각
- 조치 전후의 지표 변화
- 해결 판단 조건과 남은 후속 작업

인시던트 상세는 근거를 모으는 곳이고, 워룸은 그 근거로 내린 판단과 행동을 공유하는 곳입니다.

워룸 협업

워룸은 인시던트 대응에 참여한 사람이 사실, 가설, 결정, 조치 결과를 시간 순서로 공유하는 공간입니다. 새로 합류한 사람도 최근 요약과 채널 정보를 통해 현재 상황을 빠르게 따라잡을 수 있습니다.

채널을 찾고 현재 상황 파악하기



워룸은 세 영역으로 구성됩니다.

영역	확인할 내용
채널 목록	전체, 열린 인시던트, 수동 채널, 읽기 전용, 요약 전용 필터와 검색으로 채널을 찾습니다.
대화	참여 기록, 메시지, 답글, 반응, AI 요약을 시간 순서로 확인합니다.
채널 정보	채널 유형과 상태, 연결된 인시던트, 시작 시각, 경과 시간, 멤버 역할을 봅니다.
작업	인시던트 상세 분석, 관련 로그 분석, 워크플로우 이동, 기록 저장, 채널 닫기를 실행합니다.

먼저 연결된 인시던트와 채널 상태가 맞는지 확인합니다. 채널이 많다면 검색과 필터를 함께 사용하고, 최근 요약이 있으면 전체 대화를 읽기 전에 요약부터 검토합니다.

대응 기록 남기기

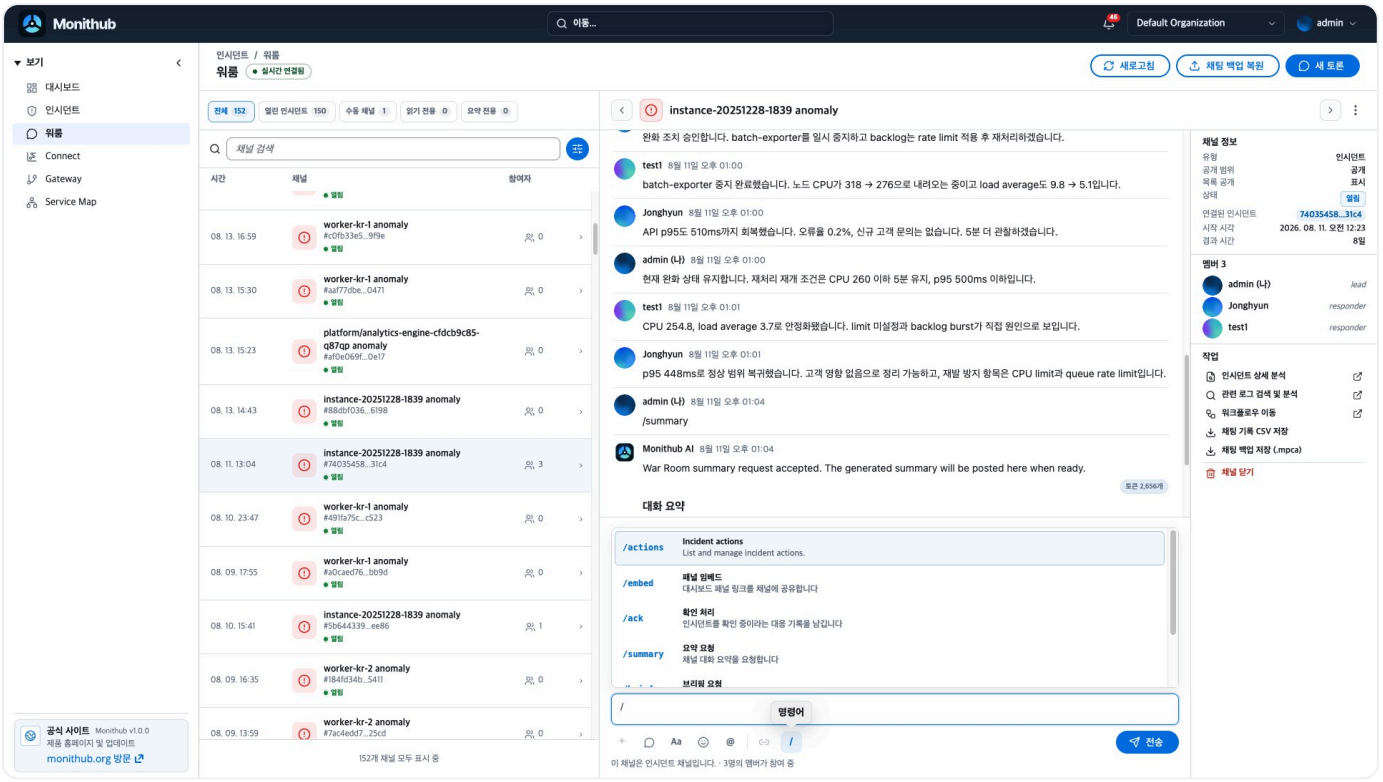
메시지는 짧더라도 사실과 다음 행동이 드러나게 작성합니다.

남길 내용	예시
최초 상황	“12:45 배포 뒤 CPU와 지연이 함께 상승했습니다.”
역할 분담	“저는 최근 배포를, `@이름`은 고객 영향을 확인해 주세요.”
조치	“13:00에 작업을 일시 중지했습니다.”
관찰 기준	“CPU 260 이하와 p95 500ms 이하를 5분간 확인합니다.”
결과	“13:06 기준 CPU와 p95가 정상 범위로 돌아왔습니다.”
후속 작업	“CPU limit과 queue rate limit 적용을 별도 작업으로 남깁니다.”

메시지 입력창에서는 우선순위, 서식, 이모지, 멘션을 선택할 수 있습니다. 특정 담당자는 `@이름`, 현재 참여자는 `@here`, 조직의 활성 멤버 전체는 `@everyone`으로 부릅니다. 긴급 멘션은 실제 영향이 크고 즉시 행동이 필요할 때만 사용합니다.

각 메시지에는 답글과 반응을 남길 수 있습니다. 별도 주제를 답글로 이어 가면 본문 흐름을 유지하면서 세부 논의를 보존할 수 있습니다.

AI 요약과 명령어 사용하기



메시지 입력창의 `/` 버튼을 누르면 자주 쓰는 작업을 선택할 수 있습니다.

명령어	기능
`/actions`	인시던트 액션을 조회하고 관리합니다.
`/embed`	함께 볼 대시보드 패널 링크를 채널에 공유합니다.
`/ack`	인시던트를 확인 중이라는 대응 기록을 남깁니다.
`/summary`	지금까지의 채널 대화를 AI가 요약하도록 요청합니다.
`/brief`	현재 인시던트의 근거를 바탕으로 AI 브리핑을 요청합니다.
`/handoff`	다음 담당자를 위한 인계 메모를 작성합니다.

`/summary`는 대화의 핵심 사실, 결정, 조치와 남은 일을 따라잡을 때 적합합니다. `/brief`는 인시던트 자체를 리드나 다른 팀에 설명할 때 유용합니다.

AI 결과는 대응을 정리하는 초안입니다. 실제 상태, 영향 범위, 원인과 조치 결과는 담당자가 원본 신호와 대화를 다시 확인하고, 틀리거나 빠진 내용은 바로 보완합니다.

오른쪽 작업 메뉴 활용하기

작업	쓰는 시점
인시던트 상세 분석	연결된 인시던트의 근거와 영향 범위를 다시 확인할 때
관련 로그 검색 및 분석	같은 시간대의 로그 단서를 찾아야 할 때
워크플로우 이동	대응 결과를 후속 절차나 작업으로 이어 갈 때
채팅 기록 CSV 저장	회고나 보고서 작성용으로 대화를 검토할 때
채팅 백업 저장 (.mpca)	Monithub에서 복원할 수 있는 형태로 워룸 맥락을 보관할 때
채널 닫기	대응과 기록 정리가 끝나 새 대화를 막을 때

CSV는 사람이 표 형태로 읽고 정리하기 위한 파일입니다. MPCA는 메시지와 답글, 반응, 명령 기록, AI 요약 등 워룸의 맥락을 복원하기 위한 전용 백업입니다. 상단의 `채팅 백업 복원`에서 MPCA 파일을 검토하고 가져올 수 있습니다.

MPCA에는 대화 원문과 참여자 정보가 포함될 수 있으므로 외부 공유용 파일처럼 다루지 않습니다. 조직의 보관 정책과 접근 권한을 확인합니다.

채널을 닫기 전 확인하기

1. 해결 판단의 근거와 최종 조치 시각이 남아 있습니다.
2. 조치 전후 상태와 고객 영향 판단이 기록되어 있습니다.
3. 남은 작업마다 담당자가 지정되어 있습니다.
4. 필요하면 `/summary` 또는 `/handoff`로 마지막 맥락을 정리했습니다.
5. 회고나 복원이 필요하면 CSV 또는 MPCA를 저장했습니다.
6. 연결된 인시던트도 적절한 상태로 변경했습니다.

채널을 닫으면 대화는 읽기 전용이 됩니다. 일반적인 대응 종료에는 채널 닫기를 사용하고, 인시던트 전체 삭제는 잘못 생성된 데이터처럼 기록 자체를 보존할 이유가 없을 때만 사용합니다.

종료와 정리

대응 종료는 상태만 바꾸는 일이 아닙니다. 문제가 안정됐다는 근거를 확인하고, 당시의 판단과 후속 작업을 다음 사람이 이해할 수 있게 남기는 과정입니다.

상태별 의미

상태	의미	다음 행동
열림	새 인시던트가 생성됐지만 담당과 판단이 확정되지 않았습니다.	담당자를 지정하고 영향을 확인합니다.
확인	담당자가 문제를 인지하고 조사 중입니다.	근거와 가설, 역할을 워룸에 남깁니다.
완화	임시 조치로 영향이 줄었지만 관찰이 더 필요합니다.	정상화 기준을 정하고 추이를 확인합니다.
해결	문제 현상이 멈추고 합의한 정상화 기준을 충족했습니다.	후속 작업과 요약을 정리합니다.
닫힘	대응과 기록 정리까지 끝났습니다.	회고와 재발 방지 작업에서 기록을 참조합니다.

조직의 운영 절차에 따라 일부 단계는 생략할 수 있지만, 상태를 바꾼 이유는 워룸에 남기는 것이 좋습니다.

해결 처리 전 확인할 것

1. 같은 이상 신호가 더 이상 늘어나지 않습니다.
2. 영향받던 서비스와 주요 지표가 합의한 정상 범위로 돌아왔습니다.
3. 고객 또는 사용자 영향이 해소됐습니다.
4. 마지막 조치와 관찰 결과가 워룸에 남아 있습니다.
5. 남은 작업과 담당자가 정리됐습니다.
6. 필요한 경우 AI 요약이나 인계 메모를 검토했습니다.
7. 회고 또는 복원이 필요하면 CSV나 MPCA를 저장했습니다.

해결 조건은 “지표가 잠깐 내려갔다”가 아니라, 팀이 정한 수치와 관찰 시간을 충족했는지로 판단합니다.

종료 뒤 남길 기록

- 최초 감지 시각과 실제 영향 시작 시각
- 영향받은 서비스와 사용자 범위

- 주요 판단 근거와 배제한 가설
- 실행한 조치와 담당자
- 조치 전후 지표와 정상화 시각
- 최종 해결 시각
- 늦게 발견했거나 놓친 신호
- 재발 방지 작업과 완료 기준

대화가 길었다면 `/summary` 결과를 초안으로 활용하되, 원인과 영향, 후속 작업은 담당자가 원문과 대조해 확정합니다.

삭제와 종료를 혼동하지 않기

상황	사용할 기능
하나의 이상 근거만 잘못 감지됨	해당 신호를 오탐으로 분류한 뒤 현재 인시던트에서 근거 제거
실제 문제 대응이 정상적으로 끝남	인시던트 해결과 워룸 채널 닫기
잘못 생성되거나 테스트용인 인시던트 전체를 없애야 함	인시던트 영구 삭제

영구 삭제는 연결된 근거와 대응 기록을 되돌릴 수 없게 제거합니다. 정상적으로 대응한 기록은 삭제하지 않고 수명 주기를 종료해 보존합니다.

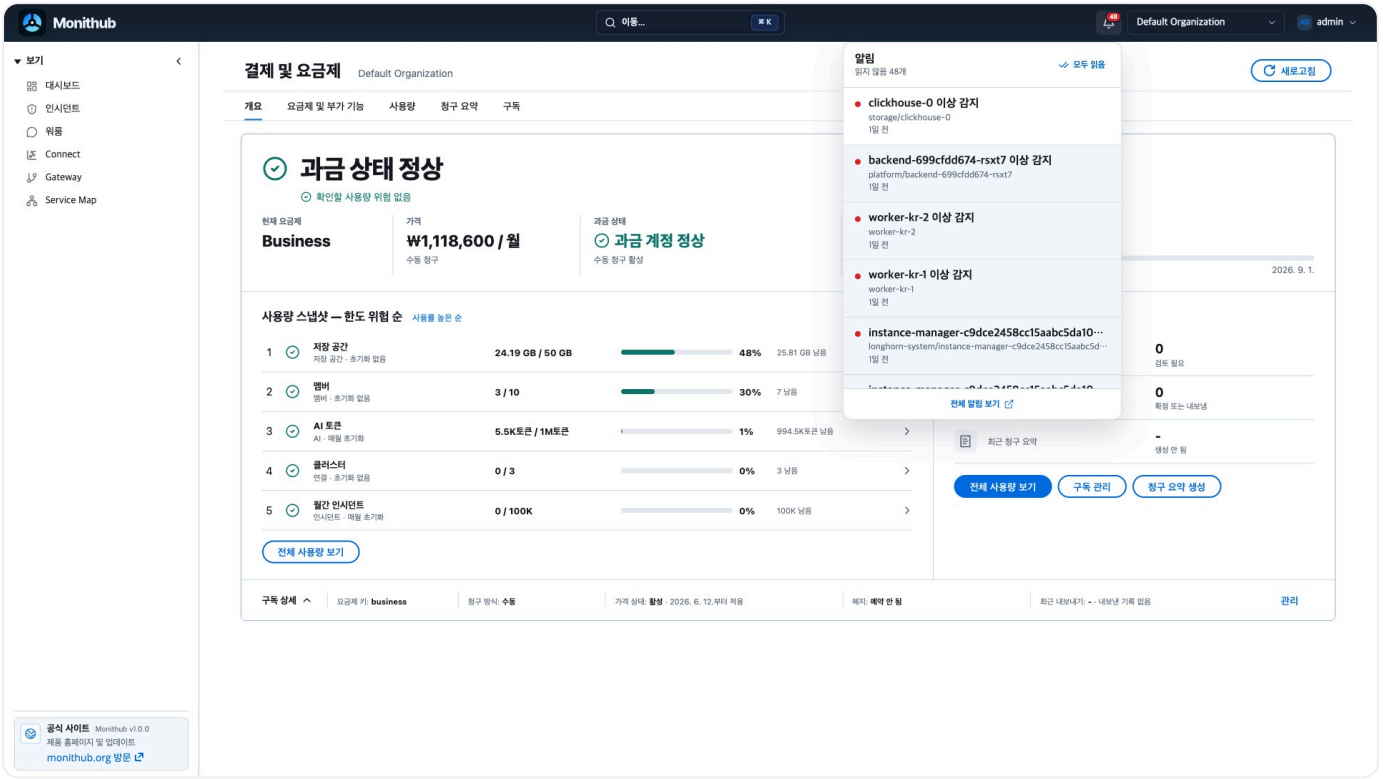
권장 마무리 순서

1. 담당자가 정상화 기준 충족 여부를 확인합니다.
2. 워룸에 마지막 조치, 결과, 남은 작업을 남깁니다.
3. 필요한 요약과 백업을 생성합니다.
4. 인시던트를 `해결` 처리합니다.
5. 후속 작업의 담당자와 완료 기준을 확인합니다.
6. 워룸 채널을 닫습니다.

이 흐름을 지키면 현재 대응을 끝내면서도 다음 대응에 필요한 근거를 잃지 않습니다.

알림 확인과 설정

알림에서는 현재 선택한 조직의 인시던트, Connect 수집 상태, 워룸 메시지를 한곳에서 확인할 수 있습니다. 알림을 열기 전에 상단의 조직 이름이 확인하려는 조직과 일치하는지 먼저 봅니다.



최근 알림 빠르게 확인하기

1. 화면 상단에서 확인할 조직을 선택합니다.
2. 오른쪽 위의 종 모양 `알림` 버튼을 누릅니다. 읽지 않은 알림이 있으면 버튼에 개수가 표시됩니다.
3. 펼쳐진 최근 알림에서 제목, 간단한 내용, 도착 시각을 확인합니다.
4. 바로 확인할 알림을 누릅니다. 해당 인시던트, Connect 또는 워룸 화면이 열리며 알림은 읽음으로 바뀝니다.
5. 최근 알림을 모두 정리하려면 `모두 읽음`을 누릅니다.

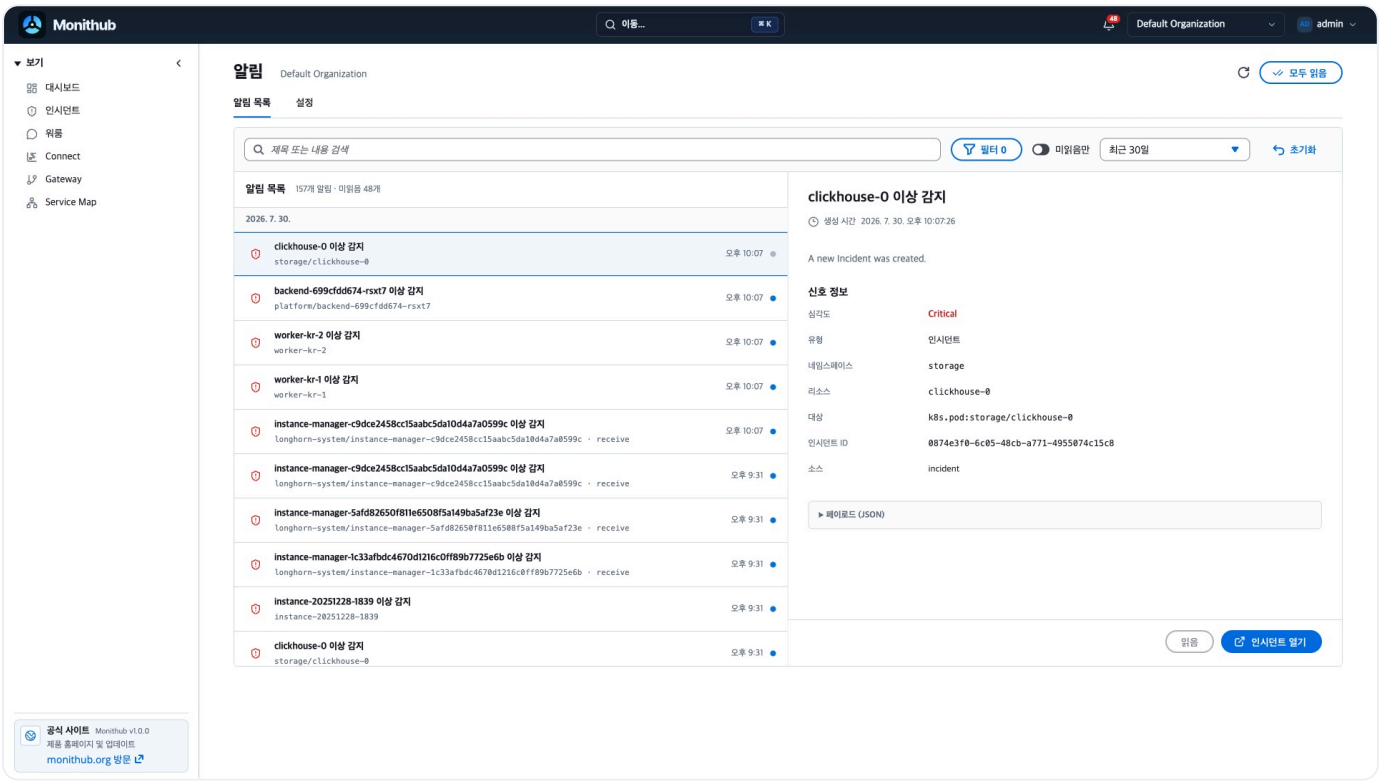
최근 항목만으로 찾기 어렵다면 아래의 `전체 알림 보기`를 눌러 알림 화면으로 이동합니다. 오른쪽 위 사용자 메뉴에서 `알림`을 선택해도 같은 화면을 열 수 있습니다.

필요한 알림 찾기

알림 화면을 열면 `알림 목록` 탭이 먼저 표시됩니다. 처음에는 최근 30일의 알림을 보여줍니다.

1. 제목이나 내용의 일부를 알고 있다면 `제목 또는 내용 검색`에 키워드를 입력합니다.
2. 아직 확인하지 않은 항목만 보려면 `미읽음만`을 켭니다.
3. 더 자세히 좁히려면 `필터`를 누르고 `읽음 상태`, `유형`, `심각도`를 선택합니다.
4. `기간`에서 최근 24시간, 7일, 30일, 전체 기간 또는 사용자 지정 기간을 선택합니다.
5. 조건을 다시 처음으로 돌리려면 `초기화`를 누릅니다.

`유형`은 `인시던트`, `워룸`, `Connect` 중에서 고를 수 있습니다. `심각도`는 인시던트 알림을 찾을 때 사용하며, 심각도를 선택하면 인시던트 유형으로 맞춰집니다.



목록에서 알림을 선택하면 오른쪽에 생성 시간과 신호 정보가 표시됩니다. 화면이 좁을 때는 목록 아래에 상세 내용이 이어집니다.

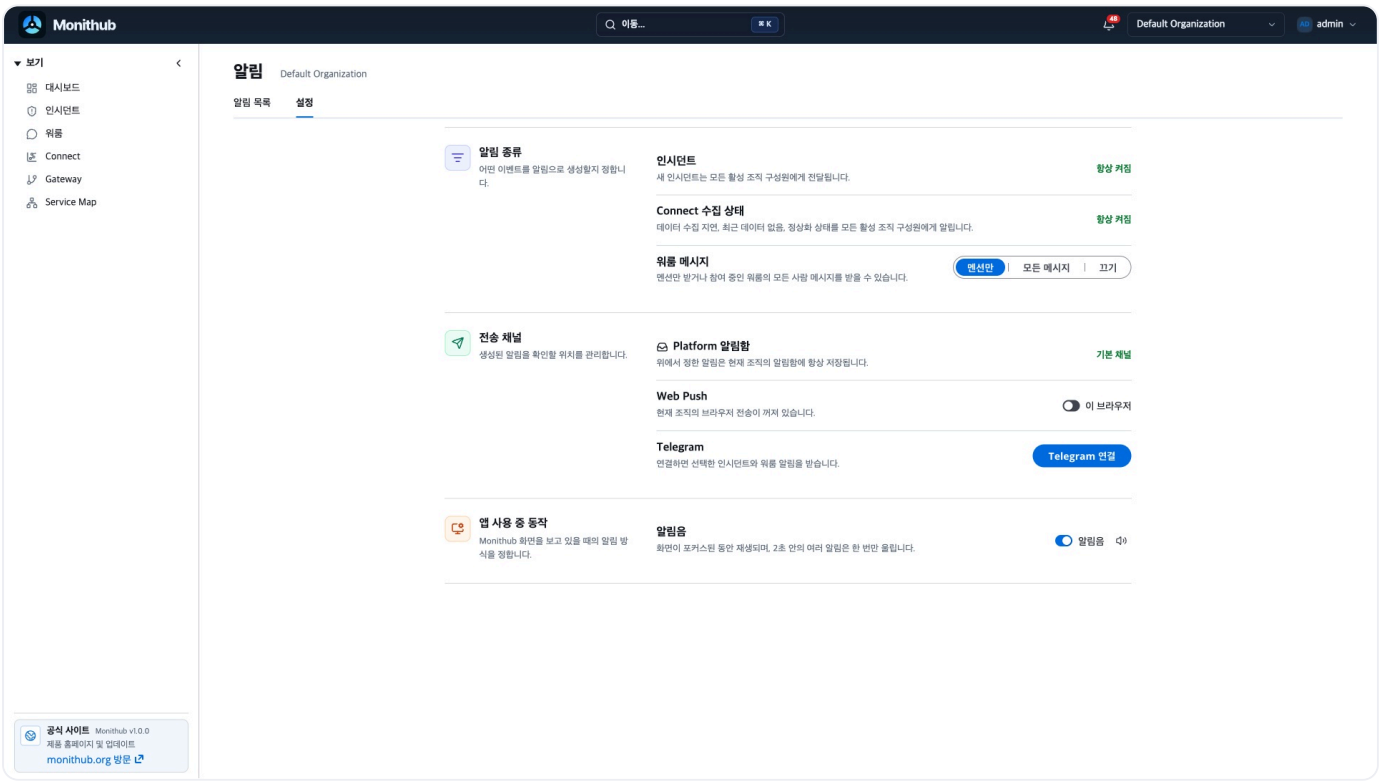
- 내용만 확인하고 목록에 남아 있으려면 `읽음으로 표시`를 누릅니다.
- 관련 화면에서 이어서 확인하려면 `인시던트 열기`, `Connect 열기` 또는 `War Room 열기`를 누릅니다. 이때 알림도 읽음으로 바뀝니다.
- 현재 조건에 보이는 항목과 관계없이 읽지 않은 알림을 모두 정리하려면 화면 위쪽의 `모두 읽음`을 누릅니다.

어떤 알림을 받을지 정하기

알림 화면에서 설정 탭을 선택합니다. 알림 종류에서는 다음 기준을 확인하거나 변경할 수 있습니다.

알림 종류	언제 받는지
인시던트	새 인시던트가 생기면 조직에 참여 중인 모든 멤버가 받습니다. 이 항목은 항상 켜져 있습니다.
Connect 수집 상태	수집이 지연되거나 중단되었을 때와 정상으로 돌아왔을 때 모든 멤버가 받습니다. 이 항목은 항상 켜져 있습니다.
워룸 메시지 멘션만	워룸에서 나를 멘션한 메시지만 받습니다. 기본 선택값입니다.
워룸 메시지 모든 메시지	내가 참여 중인 워룸에서 다른 사람이 보낸 모든 메시지를 받습니다.
워룸 메시지 끄기	워룸 메시지 알림을 받지 않습니다.

워룸 알림 기준을 누르면 바로 저장되고 알림 설정을 저장했습니다.라는 안내가 표시됩니다. 내가 직접 보낸 워룸 메시지는 알림으로 다시 오지 않습니다.



알림을 받을 위치 정하기

전송 채널에서 알림을 어디에서 확인할지 정합니다.

위치	동작
Platform 알림함	위에서 정한 알림을 현재 조직의 알림함에 항상 저장합니다. 끌 수 없는 기본 채널입니다.
Web Push	Monithub 화면을 보고 있지 않을 때 현재 브라우저와 운영체제의 알림으로 알려줍니다. 조직과 브라우저마다 따로 켭니다.
Telegram	연결한 개인 Telegram 계정으로 같은 알림을 받습니다. 연결 방법은 **Telegram 연결과 워룸 전송**에서 확인합니다.

Web Push를 사용하려면 `이 브라우저`를 켠 뒤 브라우저의 알림 권한 요청에서 허용합니다. 설정 아래에 `현재 조직에서 이 브라우저로 전송 중입니다.`가 표시되면 완료입니다.

조직을 바꾼 뒤에도 Web Push가 필요하다면 새로 선택한 조직의 알림 설정에서 다시 켭니다.

Monithub를 보고 있을 때 알림 받기

Monithub 화면에 포커스가 있을 때 새 알림이 오면 화면에 알림 배너가 나타납니다. `앱 사용 중 동작`의 `알림음`을 켜면 배너와 함께 소리가 재생됩니다.

1. `설정` 탭의 `앱 사용 중 동작`으로 이동합니다.
2. `알림음`을 켭니다.
3. 오른쪽의 스피커 모양 `알림음 테스트` 버튼을 눌러 소리를 확인합니다.
4. `테스트 알림음을 재생했습니다.`가 표시되고 소리가 들리면 완료입니다.

2초 안에 여러 알림이 도착하면 배너에는 알림 개수가 함께 표시되고 소리는 한 번만 재생됩니다. 알림함에는 각 알림이 따로 남습니다.

알림이 보이거나 들리지 않을 때

Web Push가 차단된 경우

설정에 `브라우저가 알림 권한을 차단했습니다.`가 표시되면 다음 순서로 확인합니다.

1. 브라우저의 사이트 설정에서 Monithub의 알림 권한을 `허용`으로 바꿉니다.
2. Monithub 화면을 새로고침합니다.
3. `알림 → 설정 → 전송 채널`로 돌아와 `이 브라우저`를 다시 켭니다.

`이 브라우저에서는 Web Push를 지원하지 않습니다.`가 표시되면 Platform 알림함을 사용하거나 Web Push를 지원하는 다른 브라우저에서 다시 설정합니다.

알림음이 들리지 않는 경우

1. Monithub의 `알림음`이 켜져 있는지 확인합니다.
2. Monithub 화면을 한 번 누른 뒤 `알림음 테스트`를 다시 누릅니다.
3. Monithub 탭이 음소거되어 있지 않은지 확인합니다.
4. Monithub 화면이 현재 선택되어 있는지, 기기 음량이 충분한지 확인합니다.

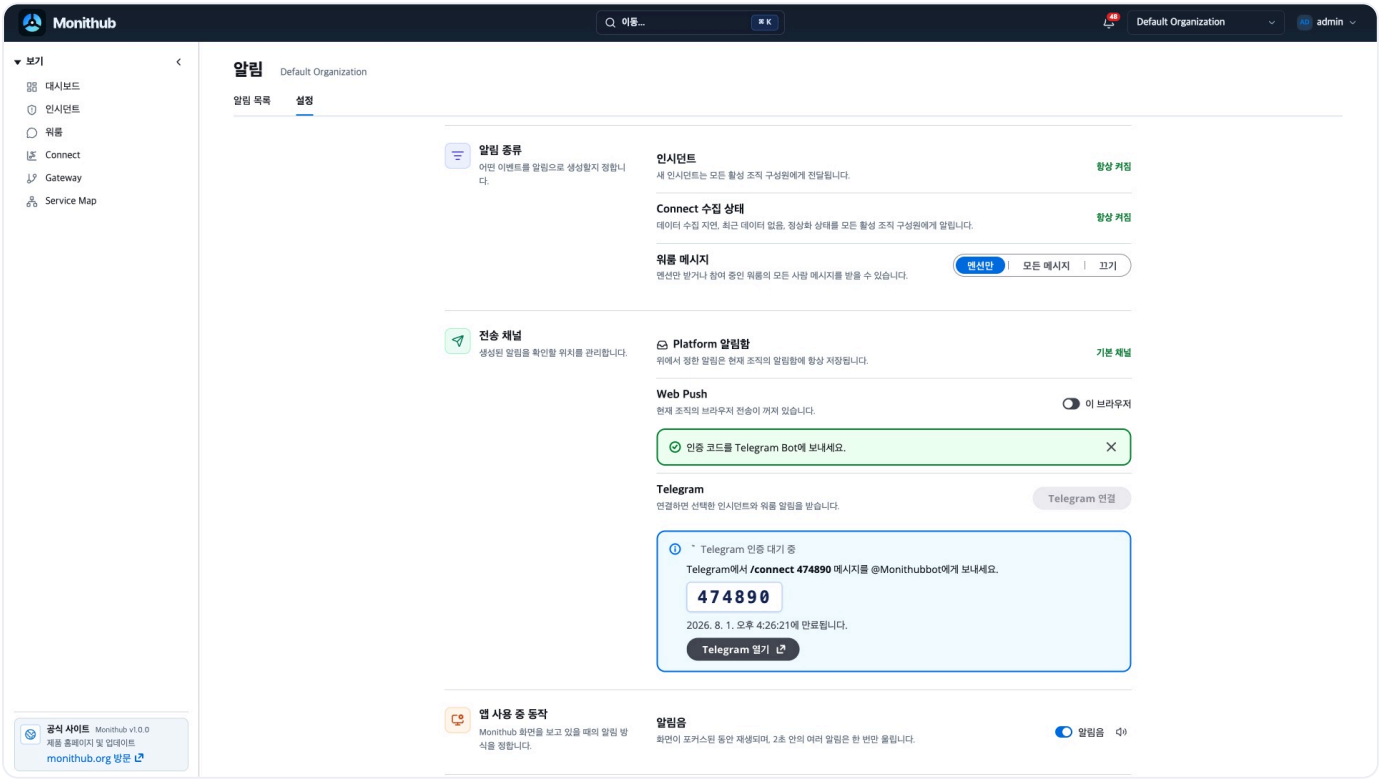
앱 알림음은 Monithub 화면을 보고 있을 때만 재생됩니다. 화면을 보고 있지 않을 때의 소리와 표시 방식은 브라우저와 운영체제의 알림 설정을 따릅니다.

Telegram 연결과 워룸 전송

개인 Telegram 계정을 Monithub에 연결하면 인시던트, Connect 수집 상태, 선택한 워룸 메시지 알림을 Telegram에서 받을 수 있습니다. Telegram Bot과의 개인 대화에서 작성한 텍스트를 접근 가능한 워룸으로 보낼 수도 있습니다.

Telegram 계정 연결하기

1. Monithub 화면 상단에서 연결할 조직을 선택합니다.
2. 오른쪽 위 종 모양 `알림` 버튼을 누르고 `전체 알림 보기`를 선택합니다.
3. `설정` 탭으로 이동합니다.
4. `전송 채널`의 `Telegram`에서 `Telegram 연결`을 누릅니다.
5. 화면에 표시된 Bot 이름, 6자리 코드, 만료 시각을 확인합니다.



6. `Telegram 열기`를 누르거나 Telegram에서 화면에 표시된 Bot을 직접 엽니다.
7. Bot과의 개인 대화에 Monithub 화면에 표시된 `/connect 6자리코드` 메시지를 그대로 보냅니다.
8. Bot이 연결 완료 메시지를 보냈는지 확인합니다.
9. Monithub로 돌아와 Telegram 계정 이름과 `연결됨` 상태가 표시되는지 확인합니다.

연결 코드는 10분 동안만 사용할 수 있습니다. 코드가 만료되었거나 연결에 실패했다면 Monithub에서 `Telegram 연결`을 다시 눌러 새 코드를 만든 뒤 처음부터 진행합니다. 연결 코드는 다른 사람에게 공유하지 않습니다.

Telegram에서 알림 확인하기

연결이 완료되면 현재 조직의 알림 설정에 따라 Telegram으로 알림이 옵니다.

- 새 인시던트와 Connect 수집 상태 알림은 항상 전달됩니다.
- 워룸 메시지는 `멘션만`, `모든 메시지`, `끄기` 중 선택한 기준을 따릅니다.
- Telegram 알림에 포함된 링크를 누르면 관련 인시던트, Connect 또는 워룸 화면으로 이동할 수 있습니다.

Telegram만을 위한 별도 알림 종류 설정은 없습니다. 받을 워룸 메시지를 바꾸려면 Monithub의 `알림 → 설정 → 알림 종류`에서 워룸 기준을 변경합니다. Web Push를 끄더라도 Telegram 연결은 유지됩니다.

Telegram 메시지를 워룸으로 보내기

Telegram Bot과의 개인 대화에서 일반 텍스트를 보내면 메시지를 전달할 워룸을 선택할 수 있습니다.

1. 연결한 Monithub Bot과의 개인 대화를 엽니다.
2. 워룸에 남길 내용을 일반 텍스트로 입력해 보냅니다. `/`로 시작하는 명령어가 아닌 텍스트를 사용합니다.
3. Bot이 보여주는 워룸 목록에서 메시지를 보낼 곳을 선택합니다. 열려 있고 내가 접근할 수 있는 워룸만 표시됩니다.
4. 목록이 여러 페이지라면 `Next →` 또는 `← Previous`로 이동해 원하는 워룸을 찾습니다.
5. Bot이 `Sent to War Room`과 선택한 워룸 이름을 표시하는지 확인합니다.
6. 필요한 경우 Monithub에서 해당 워룸을 열어 메시지가 등록되었는지 확인합니다.

워룸 선택은 메시지를 보낸 뒤 10분 동안만 유효합니다. 시간이 지나 `This message has expired.`가 표시되면 같은 내용을 다시 보내고 워룸을 다시 선택합니다. 전송하지 않으려면 `Cancel`을 누릅니다.

Telegram에서는 일반 텍스트만 워룸으로 보낼 수 있습니다. 파일이나 사진은 이 방법으로 전송할 수 없습니다.

연결된 조직과 도움말 확인하기

Telegram Bot과의 개인 대화에서 다음 명령어를 사용할 수 있습니다.

명령어	사용하는 때
`/start`	연결 방법과 사용할 수 있는 기능을 처음부터 확인할 때
`/connect 6자리코드`	Monithub에서 만든 코드로 개인 계정을 연결할 때
`/sync`	이 Telegram 계정에 연결된 Monithub 조직을 확인할 때
`/help`	계정 연결, 조직 확인, 워룸 전송 방법을 다시 볼 때

여러 조직에서 같은 Telegram 계정을 사용하려면 Monithub에서 조직을 하나씩 선택하고 각 조직의 `알림 → 설정 → Telegram`에서 연결 상태를 확인합니다.

계정을 바꾸거나 연결 해제하기

다른 Telegram 계정으로 바꾸기

1. Monithub에서 바꿀 조직을 선택합니다.
2. `알림 → 설정 → 전송 채널`의 Telegram 영역으로 이동합니다.
3. `재연결`을 누릅니다.
4. 새로 표시된 `/connect 6자리코드`를 바꿀 Telegram 계정의 Bot 대화에 보냅니다.
5. Monithub에 새 계정 이름과 `연결됨` 상태가 표시되는지 확인합니다.

현재 조직의 연결 끊기

1. 연결을 끊을 조직이 상단에 선택되어 있는지 확인합니다.
2. Telegram 영역에서 `해제`를 누릅니다.
3. `Telegram 연결을 해제했습니다.`가 표시되고 `Telegram 연결` 버튼이 다시 나타나는지 확인합니다.

연결을 해제하면 해당 조직의 Telegram 알림을 더 이상 받지 않고, Telegram에서 해당 조직의 워룸으로 메시지를 보낼 수 없습니다. 다른 조직의 연결은 그대로 유지됩니다.

연결이나 전송이 되지 않을 때

화면에 보이는 상황	다음에 할 일
인증 코드가 만료되었다는 안내	Monithub에서 `Telegram 연결` 또는 `재연결`을 눌러 새 코드를 만든 뒤 다시 보냅니다.
코드가 올바르지 않다는 Bot 안내	`/connect` 뒤에 Monithub에 표시된 6자리 숫자를 공백 하나로 구분해 다시 보냅니다.
연결된 조직이 없다는 Bot 안내	Monithub에서 현재 조직의 Telegram 연결 상태를 확인하고 새 코드로 연결합니다.
선택할 워룸이 없다는 Bot 안내	Monithub에서 내가 참여할 수 있는 열린 워룸이 있는지 확인하거나 초대를 요청합니다.
워룸 전송 실패 안내	선택한 워룸이 아직 열려 있고 내가 메시지를 쓸 수 있는지 확인한 뒤 다시 보냅니다.
Telegram 알림이 오지 않음	Monithub에서 올바른 조직과 Telegram 계정이 연결되어 있는지, 워룸 알림 기준이 `끄기`로 되어 있지 않은지 확인합니다.

문제가 계속되면 Telegram에서 `/help`를 보내 현재 사용할 수 있는 명령과 연결 순서를 다시 확인합니다.

Connect 시작하기

Connect는 서버, 애플리케이션, 데이터베이스, Kubernetes의 관측 데이터를 Monithub로 전송하기 위한 설정을 생성합니다. 연결할 카탈로그와 실행 환경을 선택하면 현재 조직에 맞는 설정 파일과 적용 명령을 안내합니다.

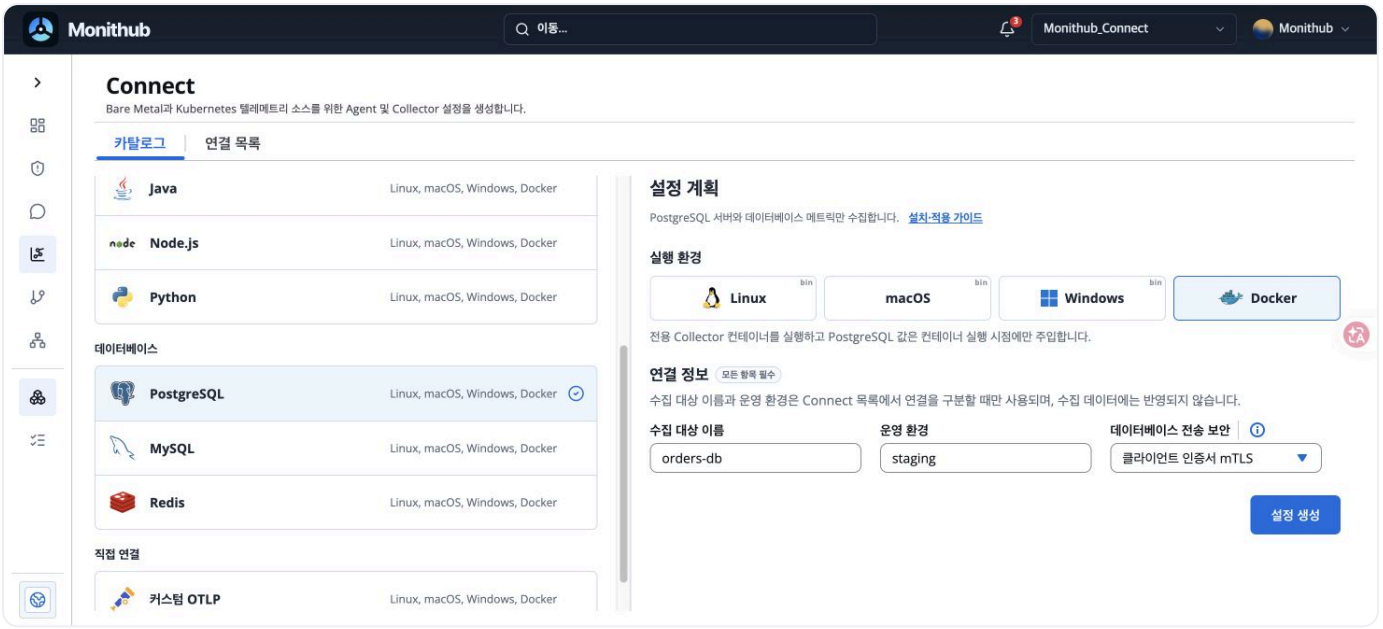
시작하기 전에

- 수집 설정을 적용하고 필요한 프로세스를 실행하거나 재시작할 수 있는 작업 권한이 필요합니다. 데이터베이스 계정처럼 카탈로그별로 필요한 권한은 각 연결 문서에서 확인합니다.
- 데이터베이스 비밀번호와 TLS 인증서는 Connect에 입력하지 않습니다. 생성된 안내에 표시된 환경 변수나 경로를 Collector 실행 환경에 직접 설정합니다.
- 생성된 설정과 토큰은 공개 저장소, 메신저, 티켓에 공유하지 않습니다.

첫 연결 만들기

1. Monithub 상단에서 데이터를 저장할 조직을 선택합니다.
2. `Connect`에서 연결할 카탈로그와 `실행 환경`을 선택합니다.
3. `연결 정보`의 필수 항목과 카탈로그별 추가 항목을 입력합니다.
4. `설정 생성`을 누릅니다.
5. `설치 · 설정 적용`에 표시된 작업을 번호 순서대로 펼쳐 실제 실행 환경에 반영합니다.
6. 카탈로그에 맞게 데이터가 전송되는지 확인한 뒤 `데이터 수집 확인`을 실행합니다.

이름과 운영 환경 입력란에 보이는 값은 형식만 보여주는 예시입니다. 실제 값을 입력해야 설정을 생성할 수 있습니다.



설정을 생성하면 데이터 수집이 확인되기 전까지 설정 초안으로 저장됩니다. 다른 카탈로그를 선택해도 초안은 유지되며, 카탈로그 목록의 진행 중인 설정에서 다시 열 수 있습니다. 데이터 수집이 확인된 설정만 연결 목록에 등록됩니다.

설정 진행 단계

단계	할 일
1 대상 및 연결 정보	카탈로그, 실행 환경과 연결 정보를 확인하거나 수정합니다.
2 설치 · 설정 적용	필요한 작업을 하나씩 펼쳐 설정과 명령을 실제 환경에 적용합니다.
3 데이터 수집 확인	Monithub에 데이터가 도착했는지 확인하고 연결을 등록합니다.

2단계의 작업은 한 번에 하나만 펼쳐집니다. 코드가 길면 코드 박스 안에서 스크롤하고, 오른쪽의 복사를 사용해 전체 내용을 복사합니다.



생성된 설정 적용하기

Connect에 표시된 전체 내용을 복사해 해당 작업이 안내하는 위치에 적용합니다. 문서의 축약 예시를 대신 사용하거나 서로 다른 연결의 설정을 섞지 않습니다.

생성되는 항목	적용 위치
설치 명령	Collector 또는 Agent를 실행할 서버의 터미널
Collector YAML	안내된 파일명의 설정 파일
환경 변수	애플리케이션이나 Collector를 시작하는 셸·배포 환경
실행 명령	실제 애플리케이션 또는 Collector 실행 환경
Dockerfile 조각	애플리케이션 이미지의 Dockerfile
Helm values	연결할 Kubernetes 클러스터의 배포 작업 폴더

파일명과 명령은 현재 설정 계획에 표시된 값을 우선합니다. 설정을 적용한 뒤에는 해당 Collector나 애플리케이션을 기존 운영 방식으로 실행하거나 재시작합니다.

플레이스홀더 바꾸기

Connect 코드에 `{REPLACE_WITH_...}`로 표시된 값은 그대로 실행할 수 없는 플레이스홀더입니다. 복사한 뒤 endpoint, 계정, 파일 경로, Docker network, 이미지 이름 등을 실제 환경 값으로 바꾸세요.

```
{REPLACE_WITH_MYSQL_ENV_FILE_PATH} → /srv/monithub/mysql-collector.env
{REPLACE_WITH_PRIVATE_NETWORK} → observability-private
{REPLACE_WITH_YOUR_IMAGE:TAG} → checkout-api:1.4.0
```

- `${env:MYSQL_ENDPOINT}`는 Collector가 환경 변수를 읽는 설정 문법입니다. 실제 주소·비밀번호를 여기에 직접 넣지 말고 실행 환경의 변수에 설정하세요. 기존 설정에 추가할 때는 안내에 따라 변수 이름의 접두어를 함께 바꿉니다.
- 코드 박스의 `복사`를 사용하고 명령의 실행 순서와 줄바꿈을 유지하세요.

환경 변수는 Collector나 애플리케이션이 시작되는 환경에 설정해야 합니다. 터미널에서 설정한 값은 이미 실행 중인 프로세스나 별도의 서비스에 자동으로 전달되지 않습니다. Docker에서는 코드가 `export KEY=VALUE` 형식인지, `--env-file`용 `KEY=VALUE` 형식인지 구분하세요. 실행 환경별 적용 방법은 각 카탈로그 문서에서 안내합니다.

이미 Collector를 운영 중이고 선택한 카탈로그가 이 방식을 지원한다면, 설정 생성 후 `설치 · 설정 적용`에서 `기존 수집 설정에 추가`를 선택하고 **기존 설정에 적용**을 따릅니다.

카탈로그 선택하기

연결할 대상	카탈로그 문서
이미 OTLP를 보내거나 여러 서비스의 OTLP 데이터를 한곳에 모으는 환경	**Custom OTLP**
Java, Node.js, Python 서비스	**애플리케이션**
Linux 서버, Docker Engine, NGINX	**인프라**
PostgreSQL, MySQL, Redis	**데이터베이스**
Kubernetes 클러스터	**Kubernetes**

지원 신호와 실행 환경을 비교하려면 **전체 카탈로그**를 확인하세요.

데이터 수집 확인하기

Collector나 애플리케이션이 실행된 것만으로 Monithub까지 데이터가 전달됐다고 볼 수는 없습니다. 다음과 같이 데이터를 준비한 뒤 3단계의 `데이터 수집 확인`을 실행합니다.

- 애플리케이션: 계측이 적용된 애플리케이션에서 HTTP 요청이나 기존 작업을 실행합니다. 로그를 확인하려면 지원되는 로깅 라이브러리로 로그도 남깁니다.
- 인프라·데이터베이스: Collector가 대상에 접근하면 메트릭을 주기적으로 수집합니다. 확인을 위해 별도의 부하나 데이터 변경 작업을 만들 필요는 없습니다.
- Custom OTLP: 앞단 송신기에서 실제 OTLP 데이터 전송

확인이 완료되면 설정이 연결 목록에 등록됩니다. 데이터가 확인되지 않는 경우는 **연결 및 수집 확인**에서 원인을 순서대로 점검하세요.

인프라 연결

인프라 카탈로그는 서버, 컨테이너 런타임과 NGINX의 상태를 Collector로 수집합니다. `실행 환경`은 수집 대상이 아니라 **Collector를 실제로 실행할 방식**입니다.

카탈로그	수집 데이터	Collector 실행 위치
Linux 호스트	CPU·메모리·디스크·파일시스템·네트워크 메트릭, 시스템 로그	수집할 Linux 서버
Docker	컨테이너 런타임 메트릭	Docker Engine이 실행되는 Linux 호스트
NGINX	`stub_status` 연결·요청 메트릭	Linux는 NGINX와 같은 호스트, Docker는 NGINX와 같은 네트워크

`Linux`는 Collector를 바이너리로, `Docker`는 컨테이너로 실행하는 방식입니다. 선택한 환경에서 생성된 설정 파일과 실행 명령을 함께 사용하세요. 이미 Collector를 운영하고 있다면 **기존 Collector에 추가**를 따릅니다.

Linux 호스트

Linux 호스트는 Collector가 실행되는 서버 자체의 호스트 메트릭과 표준 시스템 로그를 수집합니다. 다른 서버를 대신 수집하는 방식이 아니므로 대상 Linux 서버에서 실행해야 합니다.

Linux 바이너리로 실행

1. `Linux 호스트`와 `Linux`를 선택합니다.
2. `uname -m`으로 CPU를 확인하고, 설정 생성 후 x86_64 또는 ARM64 중 해당하는 설치 명령 하나를 실행합니다.
3. 생성된 YAML을 `otelcol-monithub.yaml`로 저장하고 Collector를 실행합니다.

```
otelcol-contrib --config ./otelcol-monithub.yaml
```

Docker로 실행

1. `Linux 호스트`와 `Docker`를 선택합니다.

2. 생성된 YAML을 대상 Linux 서버에 저장합니다.
3. Connect가 생성한 명령으로 호스트 파일시스템을 `/hostfs`에 읽기 전용으로 마운트해 Collector를 실행합니다.

두 방식 모두 수집할 Linux 서버에서 실행합니다. Docker Desktop의 내부 Linux VM은 macOS·Windows 호스트 자체와 다릅니다.

시스템 로그는 Collector가 시작된 이후 새로 기록되는 내용을 읽습니다. `/var/log/syslog`, `/var/log/messages`, `/var/log/*.log`에 새 로그가 기록되는지와 Collector의 읽기 권한을 확인하세요. journald에만 저장되는 로그는 기본 파일 수집 설정에 포함되지 않습니다.

- **OpenTelemetry Host Metrics Receiver**
- **OpenTelemetry File Log Receiver**

Docker

Docker 카탈로그는 Docker Engine에서 실행 중인 컨테이너의 CPU, 메모리, 네트워크, 블록 I/O와 상태 메트릭을 수집합니다. 컨테이너 애플리케이션 로그는 이 카탈로그의 수집 범위가 아닙니다.

`Docker` 카탈로그에서 실행 환경을 선택하고, Docker Engine이 실행되는 Linux 호스트에 생성된 Collector 설정을 저장합니다.

- **Linux:** CPU에 맞는 설치 명령을 실행하고, `/var/run/docker.sock`에 접근할 수 있는 계정으로 Collector를 시작합니다.
- **Docker:** 생성된 명령의 socket 마운트와 `--group-add`를 함께 사용합니다. 그룹 ID는 수집할 Docker 호스트에서 확인합니다.

```
docker_socket_gid="$(stat -c '%g' /var/run/docker.sock)"
```

생성된 명령은 위 그룹 ID를 `--group-add`에 전달합니다. 권한 오류가 나면 실제 socket 경로와 그룹 권한을 확인하세요. Rootless Docker처럼 socket 위치가 다르면 바이너리 방식은 YAML의 endpoint를, Docker 방식은 마운트할 호스트 경로와 그룹 ID 확인 명령을 실제 위치에 맞춥니다.

Docker socket 접근은 Docker 제어 권한을 부여합니다. 마운트의 `:ro`는 Docker API를 읽기 전용으로 제한하지 않으므로, 해당 호스트의 socket 접근 정책에 따라 Collector 계정에 권한을 부여하세요.

Collector가 실행되면 기존 실행 중인 컨테이너의 메트릭을 주기적으로 수집합니다. 별도 테스트 컨테이너나 부하는 필요하지 않습니다.

- **OpenTelemetry Docker Stats Receiver**

NGINX

NGINX 카탈로그는 NGINX OSS의 `stub_status`가 제공하는 연결 수와 누적 요청 수를 수집합니다. access/error 로그, 응답 시간, 상태 코드, upstream 정보는 수집하지 않습니다.

`NGINX` 카탈로그에서 실행 환경을 선택합니다. 먼저 NGINX의 상태 주소를 준비한 뒤, 생성된 YAML로 Collector를 실행합니다. 이미 `stub_status` 주소가 있다면 새로 만들지 않고 생성된 YAML의 `nginx.endpoint`를 그 주소로 변경할 수 있습니다.

Linux에서 적용

NGINX와 Collector를 같은 Linux 호스트에서 실행합니다. Connect가 생성한 `monithub-status.conf` 전체를 NGINX의 `http` 블록에서 읽는 include 경로에 저장합니다. 생성되는 상태 주소는 이 호스트에서만 접근할 수 있습니다.

```
server {
    listen 127.0.0.1:18080;
    location = /status { stub_status; }
}
```

NGINX 호스트에서 모듈과 전체 설정을 확인합니다.

```
nginx -V 2>&1 | grep -- --with-http_stub_status_module
sudo nginx -t
```

검증에 성공하면 기존 서비스 관리 방식으로 reload합니다. 다음은 NGINX 명령으로 반영하는 예시입니다.

```
sudo nginx -s reload
```

반영 후 같은 호스트에서 상태 응답을 확인합니다. `Active connections` 등 상태 값이 반환되면 생성된 YAML로 Collector를 시작합니다.

```
curl --fail http://127.0.0.1:18080/status
```

Docker에서 적용

NGINX와 Collector를 같은 사용자 정의 Docker 네트워크에 둡니다. 생성된 `location = /status` 조각은 **외부에 공개되지 않은** NGINX server 블록에 포함하세요. 이미 공개 중인 웹 서비스의 server 블록에 넣으면 `/status`도 외부에서 접근할 수 있습니다.

별도 상태 server를 사용한다면 다음처럼 구성하고, 해당 포트를 호스트에 게시하지 않습니다.

```
server {
  listen 18080;
  location = /status { stub_status; }
}
```

NGINX 컨테이너에서 `nginx -t`를 실행하고, 성공하면 같은 컨테이너에서 `nginx -s reload`를 실행합니다. Docker 호스트에서는 다음처럼 실행할 수 있습니다. `nginx`는 실제 컨테이너 이름으로 바꾸세요.

```
docker exec nginx nginx -t
```

검증에 성공한 뒤 반영합니다.

```
docker exec nginx nginx -s reload
```

Collector 설정과 실행 명령도 맞춥니다.

1. YAML의 기본 주소 `http://nginx:80/status`를 실제 컨테이너 이름 또는 네트워크 별칭과 상태 포트로 변경합니다. 위 예시라면 `http://nginx:18080/status`입니다.
2. 같은 네트워크에서 상태 URL을 요청해 `Active connections` 등 상태 값이 반환되는지 확인합니다. Collector 이미지에 셸이나 `curl`이 없다면 해당 네트워크의 진단용 컨테이너를 사용합니다.
3. 실행 명령의 `{REPLACE_WITH_PRIVATE_NETWORK}`를 해당 네트워크 이름으로 바꾸고 Collector를 시작합니다.

Connect는 NGINX 설정을 직접 수정하거나 reload하지 않습니다. 설정 검증에 실패했다면 오류를 수정한 뒤 다시 검증하세요.

- **OpenTelemetry NGINX Receiver**
- **NGINX stub_status 모듈**
- **NGINX 설정 반영**

연결 후 확인

증상	확인할 내용
Collector가 시작되지 않음	선택한 OS·CPU와 설치 파일이 맞는지, YAML 전체를 저장했는지 확인합니다.
Linux 호스트 로그만 없음	새 로그가 기록되는 파일 경로와 Collector의 파일 읽기 권한을 확인합니다.
Docker 메트릭이 없음	Docker socket mount와 socket 그룹 권한을 확인합니다.
NGINX 메트릭이 없음	`stub_status` 모듈, `nginx -t`, 상태 URL 응답과 Docker network hostname을 확인합니다.

메트릭은 주기적으로 수집됩니다. Collector를 실행한 뒤 `데이터 수집 확인`에서 결과를 확인하세요. 상태의 의미와 계속 수집하기 위한 운영 방법은 **연결 및 수집 확인**에서 안내합니다.