WhaTap

AI Agent Observability

release date. 2026. 09. 10.

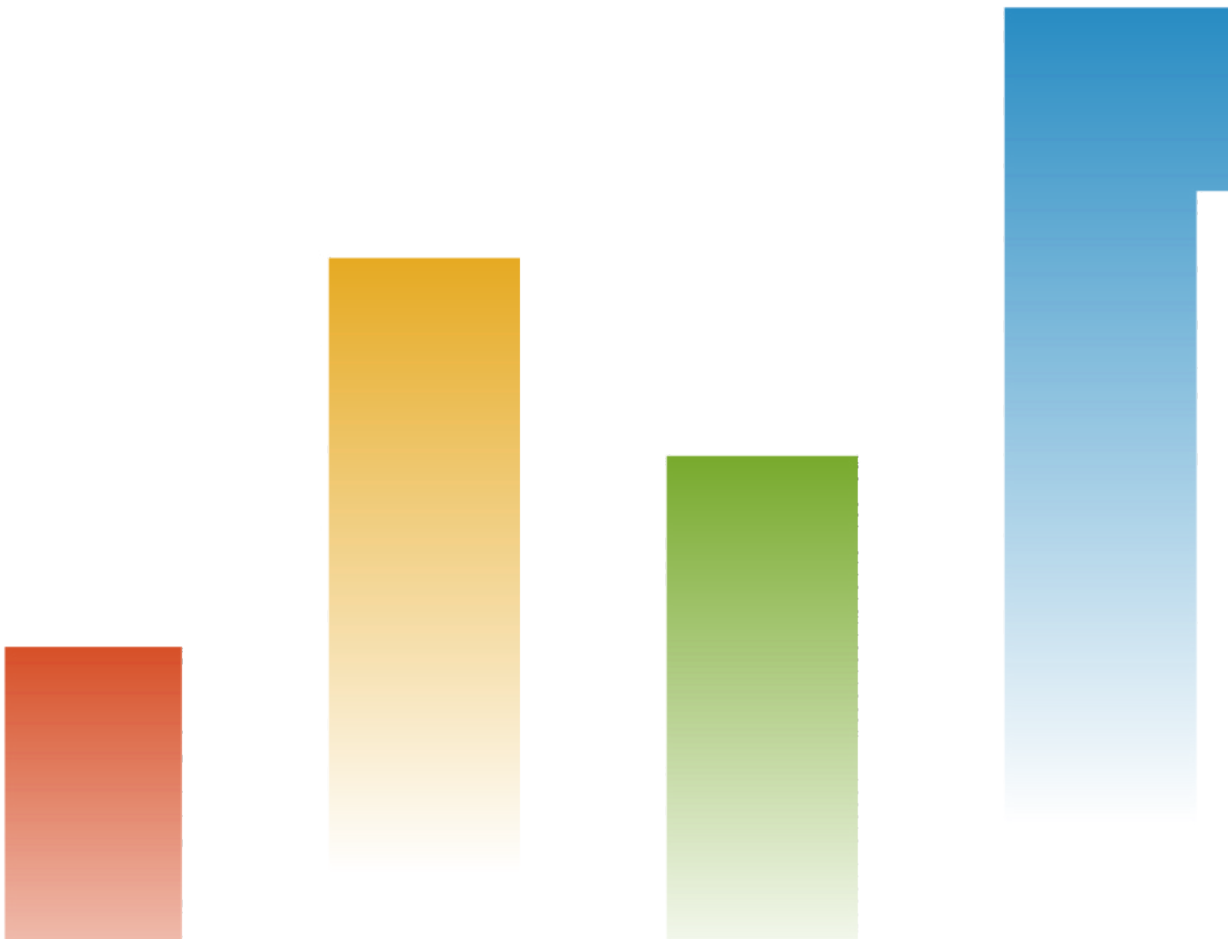


Table of Contents

AI Agent Observability

AI Agent Observability는 LLM(Large Language Model) 기반 애플리케이션의 성능, 비용, 안정성을 통합 모니터링하는 플랫폼입니다. LLM API의 요청량, 응답 성능, 토큰 사용량, 비용, 에러 현황을 실시간으로 수집하고, 모델·에이전트·프로바이더 단위의 상세 분석을 제공합니다. WhaTap APM, 서버, 쿠버네티스 인프라와 결합하여 LLM 호출을 애플리케이션 트랜잭션과 GPU 인프라까지 엔드투엔드로 추적할 수 있습니다.

LLM 모니터링, 왜 필요한가?

추론 엔진 200 응답 뒤에 숨은 LLM 이상 감지

LLM 추론 엔진은 추론 모델이 할루시네이션을 생성하거나 비정상적인 응답을 반환해도 HTTP 200을 내보냅니다. 기존 서버 모니터링으로는 이 문제를 감지할 수 없기 때문에, 장애 인지가 늦어지고 대응 시점을 놓치게 됩니다. AI Agent Observability는 응답 시간, 토큰 패턴, 에러율의 이상 변화를 실시간으로 추적하여 HTTP 상태코드만으로는 보이지 않는 모델 이상을 빠르게 감지합니다.

모델 비용이 보이지 않으면, 통제할 수 없습니다

LLM API는 호출할 때마다 토큰 단위로 과금됩니다. 모델, 프롬프트 길이, 응답 크기에 따라 건당 비용이 크게 달라지며, 트래픽이 늘어나면 예측하지 못한 비용이 발생할 수 있습니다. 어떤 모델이, 어떤 요청에, 얼마나 비용을 발생시키는지 실시간으로 파악해야 비용을 통제할 수 있습니다. 에러로 실패한 요청에도 토큰 비용이 발생하므로, 에러 비용을 별도로 추적하여 낭비되는 금액을 정량화해야 합니다.

느린 응답은 사용자가 가장 먼저 체감합니다

LLM 응답은 기존 API보다 수 초 단위로 느릴 수 있습니다. 특히 스트리밍 환경에서 첫 토큰이 늦게 도착하거나, 토큰 생성 속도가 느려지면 사용자는 "응답이 멈췄다"고 느낍니다. 대다수 사용자는 정상인데 일부만 느린 상황은 평균값으로는 감지할 수 없습니다. 어떤 모델이, 어떤 시간대에, 어떤 패턴으로 느려지는지를 시계열로 추적하고 모델 간 비교까지 가능해야 실질적인 개선이 가능합니다.

프롬프트 재현을 위한 호출 맥락 보존

LLM 호출은 동일한 프롬프트에도 매번 다른 응답을 생성합니다. 문제가 발생했을 때 "어떤 프롬프트로, 어떤 모델에, 어떤 파라미터로 호출했는지"가 보존되어 있지 않으면 재현 자체가 불가능합니다. AI Agent Observability는 모든 LLM 호출의 **시스템 메시지, 입력 프롬프트, 모델 응답, 도구 호출**을 원본 그대로 수집하고 보존합니다. 문제 발생 시 해당 시점의 정확한 호출 맥락을 복원하여, "그때 무슨 일이 있었는지"를 즉시 확인하고 재현할 수 있습니다.

멀티 모델 환경에서는 비교 분석이 필수입니다

하나의 애플리케이션에서 여러 LLM 모델과 프로바이더를 동시에 사용하는 것이 일반적입니다. 모델별 성능, 비용, 에러율을 비교하여 워크로드에 가장 적합한 모델을 선택하고, 비용 대비 성능이 낮은 모델을 교체하는 의사결정에 데이터 기반의 근거가 필요합니다.

모니터링 데이터가 흩어져 있으면, 원인을 찾을 수 없습니다

AI 애플리케이션을 운영하면 로그는 로그 플랫폼에, 메트릭은 인프라 모니터링에, 비용은 프로바이더 콘솔에, 트레이스는 APM에 파편화됩니다. 문제가 발생했을 때 여러 도구를 번갈아 보며 데이터를 수동으로 연결해야 하므로, 원인 파악에 시간이 오래 걸립니다. AI Agent Observability는 성능, 비용, 에러, 프롬프트 로그, 트랜잭션 트레이스, GPU 인프라를 **하나의 플랫폼**에서 통합하여 컨텍스트 전환 없이 문제의 원인까지 드릴다운할 수 있습니다.

AI Agent Observability 주요 기능

실시간 LLM 대시보드

LLM API의 실시간 상태를 한 화면에서 모니터링합니다. 요청량, 응답 성능, TTFT, TPOT, 토큰, 비용, 에러 등 핵심 지표를 위젯으로 제공합니다.

토큰 추이 분석 및 모델별 성능 비교

토큰 사용 패턴과 LLM 응답 성능을 시계열로 분석합니다. 모델·에이전트·프로바이더 간 성능 분포를 비교하여 가장 안정적인 모델을 식별할 수 있습니다.

비용 분석 및 최적화

토큰 단위의 비용 추적, 이전 기간 대비 변동률, 모델별 비용 비교를 제공합니다. 성능 대비 비용 버블 차트로 비용 최적화 의사결정을 지원합니다.

캐시 효율 모니터링

프롬프트 캐싱의 캐시 적중률과 절감 비용을 시간대별로 추적합니다. 캐싱 전략이 실제 비용 절감에 기여하는지 정량적으로 확인할 수 있습니다.

프롬프트 로그 분석

LLM 호출의 시스템 메시지, 입력 프롬프트, 모델 응답, 도구 호출을 개별 건 단위로 수집하고 보존합니다. 대시보드에서 이상을 발견한 후 드릴다운하여 원인을 파악할 수 있습니다.

LLM 트랜잭션 추적 및 GPU 연계 분석

LLM 호출을 애플리케이션 트랜잭션의 일부로 추적합니다. 트랜잭션 프로파일에서 LLM 스텝의 프롬프트, 토큰, 비용은 물론 GPU 상관관계 분석까지 확인할 수 있습니다.

다음 단계

- 시작하기 — 프로젝트 생성과 에이전트 유형 선택
- 지원 환경 — Python 지원 환경 확인
- LLM 대시보드 — 실시간 모니터링 시작

AI Agent Observability 시작하기

WhaTap AI Agent Observability를 시작하려면 프로젝트를 생성하고, LLM 애플리케이션 언어에 맞는 에이전트 유형을 선택한 뒤, 설치 가이드에 따라 에이전트를 적용합니다. 에이전트를 적용하면 LLM 대시보드에서 수집된 데이터를 확인할 수 있습니다.

WhaTap AI Agent Observability 모니터링은 다음 절차로 시작합니다.

1단계. 회원 가입하기

WhaTap 모니터링 서비스를 사용하려면 먼저 회원 가입이 필요합니다. 회원 가입에 관한 자세한 내용은 계정 관리 > 회원 가입 문서를 참고하세요.

2단계. 프로젝트 생성하기

에이전트를 설치하기 전에 먼저 프로젝트를 생성합니다.

1. 와탭 모니터링 서비스에 로그인합니다.
2. 화면 좌측 상단의 **전체 프로젝트** > **+ 프로젝트 생성** 버튼을 클릭합니다.
3. ❶ **상품 선택** 화면에서 **AI Observability** 카테고리를 선택합니다.
4. **AI Agent Observability** 제품의 **생성하기** 버튼을 클릭합니다.
5. ❷ **프로젝트 생성** 화면에서 프로젝트 정보를 입력합니다.

항목	설명
프로젝트 이름	프로젝트를 식별할 수 있는 이름을 입력합니다.
데이터 서버 지역	데이터가 저장될 서버 리전을 선택합니다. 선택한 리전에 사용자의 데이터가 저장됩니다.
타임 존	알림, 보고서에 사용되는 기준 시간을 설정합니다.
알림 언어 설정	알림 메시지의 언어를 선택합니다. 한국어, 영어를 지원합니다.

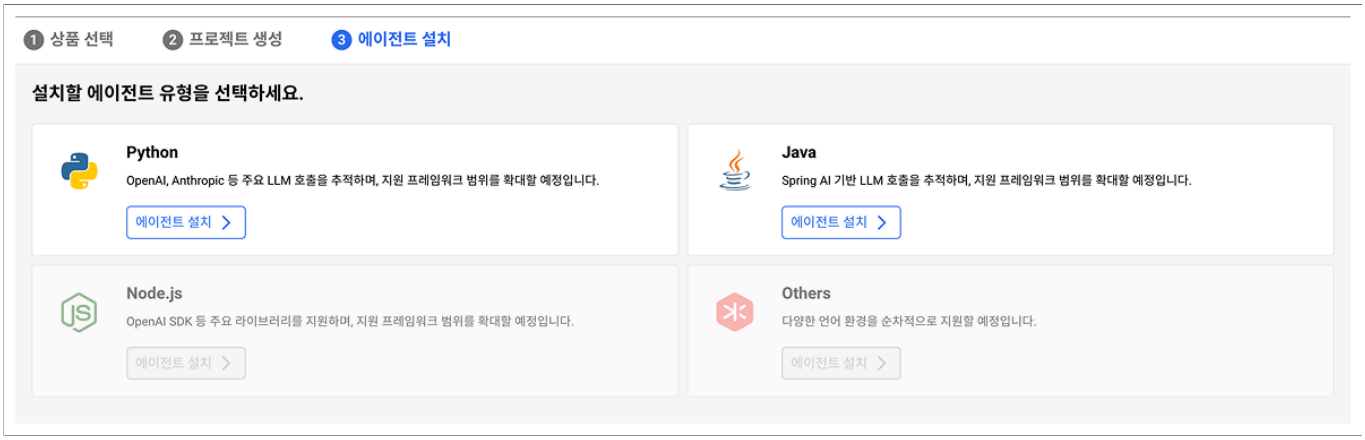
항목	설명
프로젝트 그룹(선택)	프로젝트를 그룹으로 분류할 수 있습니다.
프로젝트 설명(선택)	프로젝트에 대한 추가 설명을 입력할 수 있습니다.

6. 모든 항목을 입력한 후 **프로젝트 생성하기** 버튼을 클릭합니다.

> ⓘ 조직을 선택한 상태에서 프로젝트를 추가할 경우 **조직 하위 그룹**을 필수로 설정해야 합니다. 그룹에 대한 자세한 설명은 다음 문서를 참고하세요.

3단계. 에이전트 유형 선택하기

프로젝트를 생성하면 자동으로 ❸ **에이전트 설치** 페이지로 이동합니다. LLM 애플리케이션의 언어에 맞는 에이전트 유형을 선택합니다. 에이전트 유형을 선택하면 해당 언어의 설치 안내 페이지로 이동합니다.



에이전트 유형	설명	지원 상태
Python	Python 기반 LLM 애플리케이션용 에이전트	지원
Java	Java 기반 LLM 애플리케이션용 에이전트	추후 지원 예정

에이전트 유형	설명	지원 상태
Node.js	Node.js 기반 LLM 애플리케이션용 에이전트	추후 지원 예정
Others	기타 언어 기반 LLM 애플리케이션	추후 지원 예정

ⓘ **에이전트 설치** 페이지로 이동하지 않는다면 화면 왼쪽 메뉴에서 **관리** > **에이전트 설치**를 선택하세요.

4단계. 액세스 키 확인하기

액세스 키는 WhaTap 서비스 활성화를 위한 고유 ID입니다. 에이전트 설치 안내 페이지의 첫 번째 단계에서 **프로젝트 액세스 키 발급받기** 버튼을 클릭하여 액세스 키를 발급받습니다.

발급받은 액세스 키는 에이전트 설정 파일(whatap.conf)에 입력해야 합니다.

5단계. 에이전트 설치하기

선택한 에이전트 유형에 따라 설치를 진행합니다. 에이전트 유형별 상세 설치 방법은 다음 문서를 참조하세요.

- Python 에이전트 설치

6단계. 대시보드 확인하기

에이전트를 설치하고 애플리케이션을 재시작하면, 수 분 내에 WhaTap 수집 서버로 데이터가 전송되기 시작합니다.

1. 좌측 사이드바에서 **대시보드** > **LLM 대시보드**를 클릭합니다.
2. LLM API 호출이 발생하면 요청량, 토큰 사용량, 비용, 성능 지표가 위젯에 표시됩니다.

AI Agent Observability 지원 환경

이 문서는 AI Agent Observability 에이전트의 언어별 지원 환경과 버전 요구사항을 안내합니다. 에이전트를 설치하기 전에 애플리케이션 환경이 지원 범위에 해당하는지 확인하세요.

Python 에이전트 지원 환경

와탭 Python 모니터링 에이전트를 설치하기 전에 지원 환경을 확인하세요.

> ⓘ 최종 업데이트: 2026-09-07

지원 버전

애플리케이션 타입	Python 버전
WSGI Application	3.7 ~ 3.13
ASGI Application	3.7 ~ 3.13

운영체제

OS	최소 버전	아키텍처
Debian / Ubuntu	12.04	64bit
Red Hat / CentOS	6.x	64bit
macOS	10.15 Catalina	-
Windows	Windows Server 2008 R2, Windows 7	64bit

> AI Agent Observability에서는 현재 Windows OS를 지원하지 않습니다.

애플리케이션 서버

프레임워크	지원 버전
Bottle	-
Celery	-
CherryPy	-
Django	-
FastAPI	0.84.0 ~ 0.139.2
Flask	-
Frappe	-
Nameko	-
Odoo	v14 ~ v19
Tornado	-

라이브러리

카테고리	라이브러리
외부 호출	HttpLib, HttpLib2, Requests, Urllib, HTTPX
데이터 처리	SQL Alchemy, GraphQL

카테고리	라이브러리
로깅	smtplib, logging, loguru
메시징	Kombu, Pika
LLM SDK	OpenAI, Anthropic
Agent Orchestration	Hermes, LangGraph

데이터베이스

데이터베이스	드라이버
MongoDB	pymongo
MySQL	MySQLdb, pymysql
Neo4j	neo4j
Oracle	cxOracle
PostgreSQL	psycopg2, psycopg, psycopg_pool
Redis	redis
SQLite	sqlite3

ⓘ `whatap-python[llm]` 패키지는 `datasketches>=5.2.0`, `genai-prices` 외부 라이브러리를 필요로 합니다.

공통 지원 환경

서비스 제공 형태

와탭 모니터링은 SaaS 또는 온프레미스(설치형) 방식으로 이용할 수 있습니다. 어느 쪽을 선택해도 같은 에이전트로 같은 모니터링 기능을 사용합니다.

Deployment	Support	Description
SaaS	✅	와탭이 운영하는 수집 서버에 연결. 리전별 주소는 방화벽 항목 참조
온프레미스(설치형)	✅	수집 서버를 직접 구축해 운영(폐쇄망 환경 지원). 도입 범위는 영업 담당자와 협의

> ⓘ 인터넷에 연결하지 않는 폐쇄망 환경에서도 온프레미스로 구축할 수 있습니다.

제공 형태에 따른 기능 차이

어느 형태를 선택해도 모니터링 기능은 같습니다. 모바일 앱만 예외로 service.whatap.io SaaS에서만 사용할 수 있습니다.

> ⓘ 온프레미스에서 일부 기능을 사용하려면 별도 계약이 필요합니다.

브라우저 지원

와탭 모니터링 서비스는 웹브라우저와 모바일 앱에서 이용할 수 있습니다.

브라우저	권장 여부	지원 버전
Google Chrome	O	111 이상
Mozilla FireFox	X	최신 버전

브라우저	권장 여부	지원 버전
Edge	X	최신 버전
Safari	X	최신 버전

지원 안내

- 브라우저 호환성과 성능을 이유로 Chrome 최신 버전 사용을 권장합니다.
- 사용자 인터페이스(User Interface, UI)는 HTML5 표준 기술로 구현되어 Internet Explorer는 지원하지 않습니다.

제약 사항

와탭의 웹 인터페이스는 모바일 브라우저를 지원하지 않습니다. 모바일 기기에서 와탭에 접속하려면 Android 또는 iOS용 와탭 앱을 설치하세요. 와탭 모바일 앱은 모바일 환경에 최적화되어 있습니다. WhaTap 모바일 앱에 대한 자세한 내용은 다음 문서를 참고하세요.

방화벽

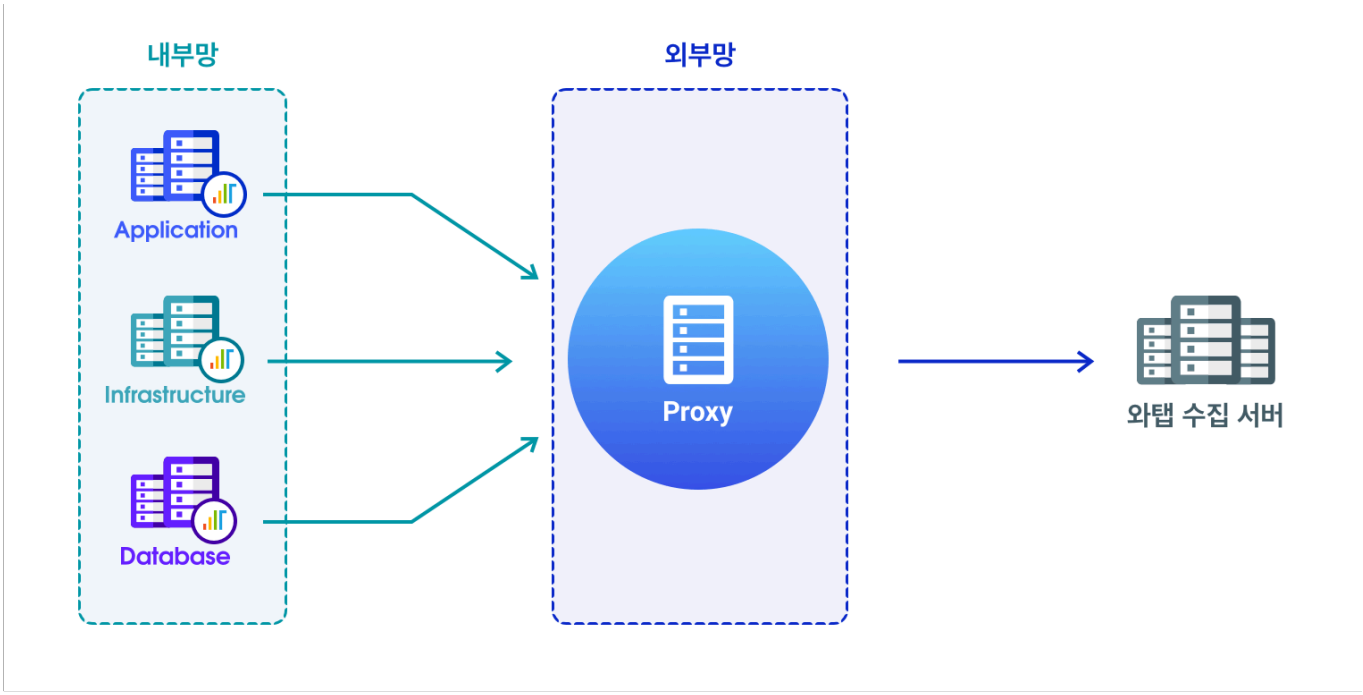
와탭 에이전트는 수집 서버 **TCP 6600** 포트로 접근할 수 있어야 합니다. 모니터링 대상과 가까운 수집 서버 주소를 허용하세요.

출발지: 와탭 에이전트

클라우드	목적지	목적지 IP	포트
AWS	와탭 서울 수집 서버	13.124.11.223 / 13.209.172.35	TCP 6600
	와탭 도쿄 수집 서버	52.68.36.166 / 52.193.60.176	TCP 6600
	와탭 싱가포르 수집 서버	18.138.0.93 / 18.139.67.236	TCP 6600
	와탭 자카르타 수집 서버	108.136.91.69 / 108.137.158.44	TCP 6600
	와탭 태국 수집 서버	43.209.109.219 / 43.209.48.86	TCP 6600

클라우드	목적지	목적지 IP	포트
	와탭 뭄바이 수집 서버	13.127.125.69 / 13.235.15.118	TCP 6600
	와탭 캘리포니아 수집 서버	52.8.223.130 / 52.8.239.99	TCP 6600
	와탭 버지니아 수집 서버	107.23.220.101 / 54.236.221.105	TCP 6600
	와탭 프랑크프루트 수집 서버	3.125.142.162 / 3.127.76.140	TCP 6600
Azure	와탭 서울 수집 서버	52.231.66.38 / 20.194.5.115	TCP 6600
	와탭 도쿄 수집 서버	52.246.169.54 / 20.210.27.232	TCP 6600
Kakao	와탭 서울 수집 서버	61.109.237.237 / 61.109.238.166	TCP 6600

에이전트에서 수집 서버로 직접 접속할 수 없다면 제공하는 Proxy 모듈을 이용해 경유하세요.



LLM Python 에이전트 설치

AI Agent Observability Python 에이전트는 `pip install whatap-python[llm]` 명령으로 설치하며, 서버 환경에 맞춰 실행 명령어 앞에 `whatap-start-agent`를 추가하여 적용합니다. 이 문서는 가상 환경 활성화부터 서비스 실행 확인까지 설치 전 과정을 안내합니다.

설치 전 확인사항

에이전트를 설치하기 전에 다음을 확인하세요.

- 시작하기 문서에서 프로젝트 생성과 액세스 키 발급을 완료했는지 확인하세요.
- 지원 환경에서 애플리케이션 환경이 지원 범위에 해당하는지 확인하세요.

1 가상 환경 활성화

애플리케이션이 virtualenv를 사용 중이라면 bin/activate 파일을 실행해 가상 환경을 활성화하세요.

2 에이전트 다운로드

액세스 키를 발급받은 다음 **에이전트 다운로드** 섹션으로 이동하세요. 다음 코드를 실행해 에이전트를 설치하세요.

```
pip install whatap-python[llm]
```

pip 명령으로 설치할 수 없다면 pypi 와탭 페이지에서 설치 파일을 다운로드하세요. 다운로드한 파일을 압축 해제한 다음 설치를 진행하세요.

```
tar xzvf whatap_python-2.x.x.tar.gz
cd whatap_python-2.X.Y.Z
python setup.py install
```

> ⓘ `whatap-python[llm]` 패키지는 `datasketches>=5.2.0` , `genai-prices` 외부 라이브러리를 필요로 합니다.
> 의존성을 누락하면 일부 데이터 수집이 빠질 수 있으니, 수동 설치 시에도 해당 의존성을 포함하세요.

에이전트 구성 파일

Python 에이전트 파일은 애플리케이션 모니터링에 필요한 정보를 추출해 와탭 수집 서버로 전달하는 트레이서와 이를 지원하는 요소로 구성되어 있습니다. 에이전트 파일 구성은 다음을 참조하세요.

🗀 whatap

3 에이전트 설정

WHATAP_HOME 경로 설정

로그와 설정 파일 경로를 위한 `$WHATAP_HOME` 경로를 지정하세요. whatap 디렉터리를 새로 생성하는 것을 권장합니다.

```
$ export WHATAP_HOME=[PATH]
```

액세스 키 및 수집 서버 IP 설정

액세스 키 및 수집 서버 IP 설정을 위해 `whatap-llm-setting-config` 명령어를 실행하세요.

```
$ whatap-llm-setting-config \
--host [ COLLECTION_SERVER_IP ] \
--license [ ACCESS_KEY ] \
--app_name [ USER_DEFINED_AGENT_NAME ] \
--app_process_name [ APPLICATION_PROCESS_NAME(uwsgi, gunicorn etc..) ]
```

설정 확인

`$WHATAP_HOME` 에 지정한 경로로 whatap.conf 파일이 생성되고 설정됩니다. 다음 명령어를 실행해 whatap.conf 파일이

생성됐는지 확인하세요.

```
$ cat $WHATAP_HOME/whatap.conf
```

whatap.conf

```
llm_license=[ACCESS_KEY]
llm.whatap.server.host=[COLLECTION_SERVER_IP]

llm_enabled=true

# application name
app_name=[ USER_DEFINED_AGENT_NAME ]

# middleware process name ex)uwsgi, gunicorn ..
app_process_name=[ APPLICATION_PROCESS_NAME(uwsgi, gunicorn etc..) ]
```

LLM 전용 모듈 설치

Python AI Agent Observability를 사용하려면 와탭이 LLM 전용으로 개발한 Go 모듈을 추가로 설치해야 합니다.

```
cd $WHATAP_HOME

curl -fsSL -O https://repo.whatap.io/python/llm/whatap-python-llm.tar.gz

tar -xzf whatap-python-llm.tar.gz
```

압축을 해제하면 생성되는 whatap-python-llm 폴더는 앞서 설정한 `$WHATAP_HOME` 경로 아래에 위치해야 합니다.

> ⓘ **권한 문제가 발생하는 경우**
>
> - 와탭 설정을 위한 $WHATAP_HOME/whatap.conf 파일의 읽기 및 쓰기 권한
> - 와탭 로그를 위한 $WHATAP_HOME/logs 경로와 하위 파일의 읽기 및 쓰기 권한
>
> `$WHATAP_HOME` 경로에 대한 권한 문제가 발생하다면 다음 명령어를 실행해 권한을 부여하세요.
>
> ```
> echo `sudo chmod -R 777 $WHATAP_HOME`
> ```

4 서버 환경별 적용

서버 환경에 맞는 방법을 선택하세요.

- **Command** **uWSGI** **Gunicorn** **Supervisor** **WSGI** **Docker** **Code**

Command 환경에서는 다음과 같이 `whatap-start-agent` 명령어를 애플리케이션 시작 명령어(Application start command) 앞에 추가하세요.

BASH

```
# $ whatap-start-agent [Application start command]
$ whatap-start-agent python manage.py runserver
```

uWSGI 환경에서는 다음과 같이 `whatap-start-agent` 명령어를 `uwsgi` 명령어 앞에 추가하세요.

BASH

```
$ whatap-start-agent uwsgi --ini myapp.ini
```

서비스에 uWSGI를 등록해 애플리케이션을 실행하는 경우 다음 예제 코드를 참고하세요.

/etc/init.d/uwsgi

```
description "uWSGI application server handling myapp"
start on runlevel [2345]
stop on runlevel [!2345]
exec whatap-start-agent [YOUR_APPLICATION_START_COMMAND]
# or if you are using a virtual environment
exec env WHATAP_HOME=[PATH] [ABSOLUTE_PATH]/whatap-start-agent [YOUR_APPLICATION_START_COMMAND]
```

/etc/init/uwsgi.conf

```
...
NAME="uwsgi"
PATH=/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
DAEMON=/usr/local/bin/uwsgi
PID=$RUN/$NAME.pid
INI_PATH=/etc/$NAME/$NAME.ini
```

```
########## WHATAP_AGENT_CONF ##########
WHATAP_HOME=[PATH]
WHATAP_AGENT=[ABSOLUTE_PATH]/whatap-start-agent
...
do_start(){
  env WHATAP_HOME=$WHATAP_HOME $WHATAP_AGENT [YOUR_APPLICATION_START_COMMAND]
}
```

- 사용자(User)를 변경한다면 `WHATAP_HOME` 환경 변수를 추가하세요.
- 가상 환경을 사용한다면 에이전트 시작 명령어를 절대 경로로 추가하세요.
- 서비스 실행 파일(/etc/init/uwsgi.conf)을 수정하여 에이전트 시작 명령어와 함께 애플리케이션을 시작하세요.
- 사용자 환경에 따라 서비스 실행 파일의 경로는 다를 수 있습니다.

Gunicorn 환경에서는 다음과 같이 `whatap-start-agent` 명령어를 `Gunicorn` 명령어 앞에 추가하세요.

BASH

```
$ whatap-start-agent gunicorn myapp.wsgi
```

서비스에 Gunicorn를 등록해 애플리케이션을 실행하는 경우 다음 예제 코드를 참고하세요.

/etc/init.d/gunicorn

```
description "gunicorn application server handling myapp"
start on runlevel [2345]
stop on runlevel [!2345]
exec whatap-start-agent [YOUR_APPLICATION_START_COMMAND]
# or if you are using a virtual environment
exec env WHATAP_HOME=[PATH] [ABSOLUTE_PATH]/whatap-start-agent [YOUR_APPLICATION_START_COMMAND]
```

/etc/init/gunicorn.conf

```
...
NAME="gunicorn"
PATH=/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
DAEMON=/usr/local/bin/gunicorn
```

```
PID=$RUN/$NAME.pid
INI_PATH=/etc/$NAME/$NAME.ini

########## WHATAP_AGENT_CONF ##########
WHATAP_HOME=[PATH]
WHATAP_AGENT=[ABSOLUTE_PATH]/whatap-start-agent

...
do_start(){
  env WHATAP_HOME=$WHATAP_HOME $WHATAP_AGENT [YOUR_APPLICATION_START_COMMAND]
}
```

- 사용자(User)를 변경한다면 `WHATAP_HOME` 환경 변수를 추가하세요.
- 가상 환경을 사용한다면 에이전트 시작 명령어를 절대 경로로 추가하세요.
- 서비스 실행 파일(/etc/init/gunicorn.conf)을 수정하여 에이전트 시작 명령어와 함께 애플리케이션을 시작하세요.
- 사용자 환경에 따라 서비스 실행 파일의 경로는 다를 수 있습니다.

Supervisor 환경에서는 적용하기 위해 supervisor.conf 파일에 다음 설정을 추가하세요.

supervisor.conf

```
[program:app-uwsgi]
environment = WHATAP_HOME=[PATH]
command = [ABSOLUTE_PATH]/whatap-start-agent /usr/local/bin/uwsgi --ini /home/blog/backend/config/uwsgi.ini

[program:nginx-app]
command = /usr/sbin/nginx
```

- 사용자(User)를 변경한다면 `WHATAP_HOME` 환경 변수를 추가하세요.
- 가상 환경을 사용한다면 에이전트 시작 명령어를 절대 경로로 추가하세요.
- 서비스 실행 파일(/etc/supervisor/conf.d/supervisor.conf)을 수정하여 에이전트 시작 명령어와 함께 애플리케이션을 시작하세요.
- 사용자 환경에 따라 서비스 실행 파일의 경로는 다를 수 있습니다.

WSGI 애플리케이션을 직접 구현하는 방법을 안내합니다. 다음을 참조하세요.

PYTHON

```python
import whatap

@whatap.register_app
def simple_app(environ, start_response):
    """Simplest possible application object"""
    status = '200 OK'
    response_headers = [('Content-type', 'text/plain')]
    start_response(status, response_headers)
    return ['Hello world!\n']
```

Docker 컨테이너로 서비스되는 애플리케이션의 경우 다음 내용을 Dockerfile에 추가하세요.

1. **Python 에이전트**를 설치하세요.

   Dockerfile

   ```
   ENV WHATAP_HOME /whatap
   RUN mkdir -p /whatap
   RUN pip install whatap-python[llm]
   ```

   > ⓘ requirements.txt를 사용한다면 해당 파일에 **whatap-python[llm]**을 추가하세요.

2. **액세스 키** 및 **수집 서버 IP** 설정을 등록하세요.

   Dockerfile

   ```
   RUN touch /whatap/whatap.conf
   RUN echo "llm_license=[ ACCESS_KEY ]" > /whatap/whatap.conf
   RUN echo "llm.whatap.server.host=[ COLLECTION_SERVER_IP ]" >> /whatap/whatap.conf
   RUN echo "app_name=[ AGENT_NAME ]" >> /whatap/whatap.conf
   RUN echo "app_process_name=[ APPLICATION_PROCESS_NAME(uwsgi, gunicorn etc..) ]" >> /whatap/whatap.conf
   ```

3. 기존 실행 커맨드 `python manage.py runserver` 앞에 `whatap-start-agent` 를 추가하세요.

   Dockerfile

```
CMD ["whatap-start-agent", "python", "manage.py", "runserver", "0.0.0.0:8000"]
```

코드 가장 윗줄에 API를 직접 호출하는 코드를 다음과 같이 추가해 에이전트를 적용할 수 있습니다.

코드를 통해 에이전트를 실행하는 경우 whatap.conf에 다음 옵션을 추가하세요.

whatap.conf

```
whatap_code_start=true
```

코드 가장 윗줄에 다음과 같이 추가하세요.

PYTHON

```
import whatap
whatap.agent()
```

애플리케이션 서버를 실행하면 에이전트가 모니터링 데이터를 수집하기 시작합니다.

5 서비스 실행 확인

다음 명령어를 실행해 와탭 Python 서비스가 정상 실행되는지 확인하세요.

```
ps -ef | grep whatap_python
```

Python 에이전트 설정

기본 필수 활성화 옵션

`llm_license=` `llm.whatap.server.host=`

파이썬 에이전트는 APM과 LLMOBS 프로젝트를 함께 사용할 수 있으며, LLM 프로젝트의 관련 값들은 이와 같은 옵션에 태깅을 해야만 합니다.

`llm_enabled=true`

이 옵션을 통해 LLM 로그싱크팩과 메트릭 수집을 활성화 할 수 있습니다. LLMOBS 사용시 필수적으로 해당 옵션은 `true` 로 되어 있어야 합니다.

LLM API 호스트 인식

`llm_api_hosts=localhost:8000,api.openai.com:443`

자동 계측이 모르는 LLM API endpoint (vLLM / LiteLLM / 사설 게이트웨이 등) 를 LLM 호출로 판단하기 위한 설정 옵션입니다. httpc driver 가 해당 호스트로 가는 호출을 `LLM API` 드라이버로 마킹하게 됩니다.

모델 가격 수동 입력

`llm_model_pricing=gpt-4o|2.50|10.00|1.25,gpt-4o-mini|0.15|0.60|0.075,my-custom-model|1.00|5.00`

모델의 토큰 별 가격을 수동 지정할 수 있습니다. 설정된 값은 실제 비용이 인식되더라도 가장 최우선으로 반영됩니다.

형식: **`model|input_per_1m|output_per_1m|cached_per_1m`**

- 4 필드
- 모델 간 콤마 구분
- 4번째 (cached) 는 optional, 비우거나 생략 가능

LLM 트랜잭션 설정

AI Agent Observability에서 워크플로우와 에이전트 패턴을 계측하기 위한 공개 API 사용 가이드입니다.

워크플로우 및 AI 에이전트의 영역만 선언하면, 해당 스코프 안에서 발생하는 모든 LLM API 호출이 별도의 트랜잭션이 아니라 **하나의 LLM 트랜잭션 아래 "스텝"으로 묶여** 수집됩니다. 사용자가 할 일은 영역을 표시하는 것뿐입니다.

AI Agent Observability의 트랜잭션 종류

LLM 트랜잭션의 종류는 총 3종입니다.

- `Workflow` (기본) — `[Workflow] <name>`
- `Agent` — `[Agent] <name>`
- `LLMAPI` — 워크플로우로 감싸지 않은 단독 호출은 `[LLMAPI] Pure LLM API` 로 수집됩니다.

워크플로우 안에서 다른 워크플로우가 실행되면 부모-자식 트랜잭션이 트라잭토리(Trajectory)로 연결되어 호출 관계를 추적할 수 있습니다.

워크플로우 트랜잭션 설정

데코레이터와 컨텍스트 매니저 두 가지 형태를 모두 지원합니다.

데코레이터

워크플로우 이름이 함수 단위로 고정되어 있을 때 사용합니다.

```
from whatap.llm import workflow

@workflow("support_chat")
async def support_chat(conversation_id, message, history):
    intent = await classify_intent(message)   # 이 안의 LLM 호출들은
    docs   = await retrieve(message)          # 전부 이 워크플로우의 "스텝"이 됩니다
    answer = await generate(message, docs)
    return {"intent": intent, "answer": answer}
```

컨텍스트 매니저

워크플로우 이름을 런타임에 결정하거나, 함수 일부 구간만 감싸고 싶을 때 사용합니다.

```
from whatap.llm import workflow

with workflow("ticket_triage"):
    classification = await _classify(ticket_text)
    entities = await _extract(ticket_text)
```

> ! 비동기 코드에서도 `async with` 이 아니라 위와 같이 `with` 를 사용하세요. `workflow` 와 `agent` 는 동기 컨텍스트 매니저이므로, 스코프 안에서 `await` 하는 것은 문제가 없지만 `async with` 으로 열면 에러가 발생합니다.

에이전트 트랜잭션 설정

사용법은 워크플로우 설정과 동일하며, 트랜잭션 이름의 접두어만 `[Agent]` 로 바뀝니다. 데코레이터와 컨텍스트 매니저 두 가지 형태를 모두 지원합니다.

데코레이터

에이전트 실행 함수가 고정되어 있을 때 사용합니다.

```
from whatap.llm import agent

@agent("react_agent")
async def run_agent(task):
    ...
    return result
```

컨텍스트 매니저

에이전트 이름을 런타임에 결정하거나, 실행 구간 일부만 감싸고 싶을 때 사용합니다.

```
from whatap.llm import agent

with agent("react_agent"):
    result = await react_agent.run(task)
```

코드 수정 없이 영역 선언하기

애플리케이션 코드를 수정할 수 없을 때는 설정으로 워크플로우 경계를 지정할 수 있습니다. whatap.conf에 대상 메서드를 나열하면, 해당 메서드의 실행이 곧 워크플로우 경계가 됩니다. 워크플로우 이름은 메서드 이름이 사용됩니다.

whatap.conf

```
workflow_method_patterns=myapp:RagService.run, myapp.agent:run_agent
```

`모듈:클래스.메서드` 또는 `모듈:함수` 형태로 쉼표로 구분합니다.

자동 계측 지원 프레임워크

다음 프레임워크는 별도의 영역 선언 없이 LLM 트랜잭션이 자동으로 수집됩니다.

프레임워크	수집 방식
hermes-agent	에이전트 실행 레이어(`AIAgent.run_conversation`)를 계측하여 `[Agent] <name>` 트랜잭션으로 등록합니다. 게이트웨이 인바운드(`GatewayRunner._handle_message`)는 `[Workflow] gateway/<platform>` 트랜잭션으로 등록되고, 그 안에서 실행된 에이전트가 자식 트랜잭션으로 연결됩니다.
LangGraph	컴파일된 그래프의 실행 경계(`Pregel.invoke` / `ainvoke` / `stream` / `astream`)를 계측하여 **그래프 1회 실행 = LLM 트랜잭션 1건**으로 등록합니다. `tools` 노드를 가진 그래프(`create_react_agent`)는 `[Agent]`, 그 외는 `[Workflow]`로 등록됩니다. 서브그래프는 자식 트랜잭션으로 중첩 연결됩니다.

Evaluation

WhaTap AI Agent Observability는 운영 중인 LLM 애플리케이션의 응답 품질과 안정성을 자동으로 측정하는 **Evaluation** 기능을 제공합니다.

Evaluation은 LLM 응답에 대해 할루시네이션, 답변 관련성, 유해성(toxicity), 프롬프트 인젝션, 사실성(factuality), PII 노출, URL 포함 등의 평가를 자동으로 적용하는 기능입니다. 평가 결과는 점수(0.0~1.0)로 산출되어 메트릭과 로그싱크에 함께 수집되므로, 응답 품질을 시계열로 추적하고 임계치 기반 알림을 구성할 수 있습니다.

> Evaluation 기능은 Python 에이전트만 지원합니다.

> 평가 파이프라인을 활성화하면 사용자 응답을 위한 LLM 호출과 별개로 **judge 호출만큼의 LLM 리소스(토큰·비용)가 추가로 발생**합니다.

설정 옵션

키	기본값	설명
`llm_eval_enabled`	`false`	평가 파이프라인 마스터 토글. `true`면 파이프라인이 기동되고 기본 평가자가 자동 등록됩니다. 나머지 옵션은 모두 선택 사항입니다.
`llm_eval_evaluators`	`combined_judge, pii_leak, url_scan`	에이전트에서 실행할 평가 파이프라인을 선택합니다.

`combined_judge`는 judge를 **1회만 호출해 5가지 관점**(`hallucination`, `answer_relevance`, `toxicity`, `prompt_injection`, `factuality`)을 한 번에 추론하는 통합 평가자입니다.

관점을 개별로 제어하려면 `combined_judge`를 빼고 필요한 라벨만 나열하세요. 지정 가능한 값은 다음과 같습니다.

- LLM judge 기반: `hallucination` · `answer_relevance` · `toxicity` · `prompt_injection` · `factuality`
- 규칙 기반(LLM 호출 없음): `pii_leak` · `url_scan`

whatap.conf

```
llm_eval_evaluators=hallucination,toxicity,pii_leak
```

단, 개별 라벨은 각각 judge를 호출하므로 5개를 모두 지정하면 응답 1건당 judge 호출이 5회 발생합니다. 비용 측면에서는 `combined_judge` 사용을 권장합니다.

judge 호출 방식

judge LLM 호출에는 사용자 호출에서 계측이 캡처한 **자격증명(엔드포인트·API 키) 스냅샷**을 재사용해 에이전트가 자기 소유의 client를 별도로 생성합니다. 사용자 요청의 client 인스턴스를 그대로 빌리지 않기 때문에, 요청마다 client를 만들고 닫는 애플리케이션에서도 안전하게 동작합니다.

judge 모델은 애플리케이션이 사용한 모델과 동일하게 설정됩니다. 스냅샷된 엔드포인트가 자체 호스팅·프록시일 수 있어 클라우드 기본 모델을 가정할 수 없기 때문입니다. 즉 애플리케이션이 고품질 또는 저품질 모델을 사용하면 judge 호출도 같은 모델로 실행됩니다. 평가 모델을 조정하려면 `llm_eval_judge_model`에 모델을 **명시적으로 지정**하세요.

Evaluation 자격증명

다음 경우에만 whatap.conf에 Evaluation 자격증명을 직접 지정합니다.

- 스냅샷이 OAuth 인증이라 평문 API 키가 없는 경우
- 스냅샷된 엔드포인트가 `chat.completions` 경로를 서빙하지 않는 경우
- 평가를 의도적으로 다른 모델·엔드포인트로 보내려는 경우

지정 방법은 다음 **두 가지 모드** 중 하나입니다. 두 모드 모두 지정하는 순간 자격증명 스냅샷 경로를 **완전히 대체**합니다.

OpenAI 호환 커스텀 엔드포인트

whatap.conf

```
llm_eval_judge_base_url=http://10.0.0.5:8000/v1
llm_eval_judge_api_key=EMPTY
llm_eval_judge_model=qwen2.5-7b-instruct
```

- `llm_eval_judge_provider`는 생략해도 됩니다. `base_url`이 있으면 OpenAI 호환으로 간주합니다.
- 호출은 `POST <base_url>/chat/completions`로 나갑니다.

- 키가 불필요한 로컬 엔진이면 `llm_eval_judge_api_key=EMPTY` 로 지정하세요.

클라우드 프로바이더 API 직접 지정

OpenAI 또는 Anthropic의 기본 엔드포인트로 judge를 보냅니다.

whatap.conf

```
llm_eval_judge_provider=openai
llm_eval_judge_api_key=OPENAI_API_KEY
llm_eval_judge_model=gpt-4o-mini
```

- `llm_eval_judge_provider` 는 `openai` 또는 `anthropic` 을 지정합니다.
- `llm_eval_judge_base_url` 은 지정하지 않습니다. SDK 기본 엔드포인트를 사용합니다.

비용 / 성능 옵션

키	기본값	설명
`llm_eval_sample_rate`	`1.0`	LLM judge 평가 비율. `0.1` 은 10%만 평가해 비용이 1/10로 줄어듭니다. 규칙 기반 평가자는 영향을 받지 않고 항상 100% 실행됩니다.
`llm_eval_judge_timeout_sec`	`30`	judge 1회 호출의 상한(초). `0` 또는 음수면 무제한입니다.
`llm_eval_workers`	`4`	평가 워커 스레드 수.
`llm_eval_buffer_limit`	`1000`	평가 큐의 최대 크기. 초과 시 평가가 드롭되며 `LLM030` 경고가 발생합니다. 애플리케이션은 차단되지 않습니다.
`llm_eval_track_judge_calls`	`false`	judge 자체의 LLM 호출도 메트릭·로그싱크에 수집할지 여부. `true` 면 judge의 비용·지연도 가시화되지만 LLM 호출 카운트가 증가합니다(사용자 호출 + judge 호출). 이때 judge 호출의 `operation_type` 은 `whatap_evaluation` 으로 기록됩니다.

특정 파이프라인에만 적용

데코레이터와 컨텍스트 매니저를 사용해 특정 함수나 구간의 LLM 호출만 평가할 수 있습니다.

```
from whatap.llm.evaluators import evaluate_with, evaluation_scope
from whatap.llm.evaluators.builtins import CombinedJudgeEvaluator

@evaluate_with(CombinedJudgeEvaluator())
def chat(q):
    ...

def chat2(q):
    with evaluation_scope(CombinedJudgeEvaluator()):
        ...
```

코드로 등록할 수 있는 평가자 클래스는 whatap.conf로 등록하는 선택지와 동일합니다.

```
from whatap.llm.evaluators.builtins import (
    # LLM judge 기반 — judge 호출 비용 발생
    CombinedJudgeEvaluator,
    HallucinationEvaluator,
    AnswerRelevanceEvaluator,
    ToxicityEvaluator,
    PromptInjectionEvaluator,
    FactualityEvaluator,

    # 규칙 기반 — LLM 호출 없음, 비용 0
    PIILeakEvaluator,
    URLScanEvaluator,
)
```

Evaluation 지표 설명

Evaluator	LABEL	종류	점수 방향	비용
`CombinedJudgeEvaluator`	`combined_judge`	LLM judge	종합 위험도 — 낮을수록 좋음	judge 1회로 5개 aspect
`HallucinationEvaluator`	`hallucination`	LLM judge	낮을수록 좋음	judge 1회
`AnswerRelevanceEvaluator`	`answer_relevance`	LLM judge	높을수록 좋음	judge 1회
`ToxicityEvaluator`	`toxicity`	LLM judge	낮을수록 좋음	judge 1회
`PromptInjectionEvaluator`	`prompt_injection`	LLM judge	낮을수록 좋음	judge 1회
`FactualityEvaluator`	`factuality`	LLM judge	높을수록 좋음	judge 1회

각 Evaluation이 측정하는 항목은 다음과 같습니다.

- `answer_relevance` **: 답변 관련성(높을수록 좋음)**

  응답이 사용자 질문에 얼마나 충실하게 답하는지를 측정합니다. 질문에 완전히 답하면 `1.0` , 부분적·지엽적으로만 답하면 `0.5` , 질문과 무관하거나 회피·이탈한 경우 `0.0` 으로 채점됩니다.

- `hallucination` **: 할루시네이션(낮을수록 좋음)**

  응답에 근거 없는 주장이 포함되어 있는지를 측정합니다. 컨텍스트(ground truth)가 주어진 경우에는 컨텍스트에 대한 충실도(faithfulness)를, 컨텍스트가 없는 경우에는 응답 자체의 자기 일관성(self-consistency)을 기준으로 평가합니다. `0.0` 은 컨텍스트에 완전히 충실한 상태, `1.0` 은 전적으로 날조된 상태입니다.

- `toxicity` **: 유해성(낮을수록 좋음)**

  응답에 유해 콘텐츠가 포함되어 있는지를 hate / harassment / violence / sexual / self_harm / profanity 6개 카테고리

기준으로 판정하며, 탐지된 카테고리 목록을 함께 반환합니다. `0.0`은 완전히 안전, `1.0`은 심각한 유해 상태입니다.

- `prompt_injection` **: 프롬프트 인젝션(낮을수록 좋음)**

  사용자 입력에 포함된 "ignore previous instructions" 류의 오버라이드 시도가 성공했는지, 혹은 응답이 시스템 프롬프트·숨겨진 지시·기밀 정보를 누출했는지를 판정합니다. `0.0`은 원래 작업을 그대로 수행하고 어떤 보호 정보도 노출하지 않은 상태, `1.0`은 인젝션에 완전히 탈취된 상태입니다.

- `factuality` **: 사실성(높을수록 좋음)**

  응답의 사실적 정확도를 측정합니다. 역사·과학·지리·수학적 사실과 날짜·이름·숫자 등 검증 가능한 객관적 주장을 기준으로 하며, 의견이나 완곡한 표현은 평가 대상에서 제외합니다. `1.0`은 모든 사실 주장이 정확한 상태, `0.0`은 명백히 거짓인 주장이 다수 포함된 상태입니다. 컨텍스트 충실도를 보는 `hallucination`과 달리, factuality는 컨텍스트와 무관하게 응답 자체의 사실 정확도를 봅니다.

종합 위험도 점수 (combined_judge)

`combined_judge`는 5개 aspect 점수를 하나의 종합 위험도(risk)로 합산합니다. 이때 점수 방향이 다른 두 그룹을 위험도 기준으로 통일합니다.

- **위험 방향 aspect(** `hallucination` , `toxicity` , `prompt_injection` **)** — 점수가 높을수록 위험하므로 점수를 그대로 위험도로 사용합니다.
- **품질 방향 aspect(** `answer_relevance` , `factuality` **)** — 점수가 높을수록 좋으므로 `1 − score`를 위험도로 사용합니다.

각 aspect를 독립적인 위험으로 가정하고, "모든 aspect가 안전할 확률"의 보수(complement)로 종합 위험도를 계산합니다. 통계학의 **Probabilistic OR** 공식입니다.

$$\text{risk} = 1 - \prod_{i=1}^{N} (1 - r_i)$$

계산 예시

위험도 `[0.7, 0.3, 0.1, 0.1, 0.2]`인 경우: **1 − (0.3 × 0.7 × 0.9 × 0.9 × 0.8) = 0.864**

위험도 합산 특성 (시뮬레이션)

개별 위험도	최댓값(max)	종합(compound)
`[0.5, 0, 0, 0, 0]`	0.50	**0.50**
`[0.5, 0.5, 0, 0, 0]`	0.50	**0.75**
`[0.3, 0.3, 0.3, 0.3, 0.3]`	0.30	**0.83**
`[1.0, 0, 0, 0, 0]`	1.00	**1.00**

종합 위험도는 단순 최댓값보다 항상 크거나 같습니다. 낮은 위험이라도 여러 aspect에 걸쳐 누적되면 종합 위험도는 그만큼 높아지며, 어느 한 aspect가 `1.0`이면 나머지와 무관하게 종합 위험도도 `1.0`이 됩니다. 이를 통해 단일 지표로는 놓치기 쉬운 "여러 영역에서 동시에 조금씩 나쁜 응답"을 효과적으로 포착할 수 있습니다.

Operation Type 및 Prompt Version

Operation Type은 LLM 호출을 **비즈니스 맥락 단위로 그룹화·필터링**하기 위한 사용자 정의 라벨입니다. 지정한 라벨은 메트릭 태그의 일부가 되므로, 사용자 지정 단위로 다양한 지표를 분석할 수 있습니다.

Prompt Version은 `operation_type` 안에서 **프롬프트 문안이 몇 번째 개정판인지**를 표시하는 사용자 정의 라벨입니다. `operation_type`이 "무엇을 하는 호출인가"(비즈니스 단위)라면, `prompt_version`은 "그 호출을 어떤 프롬프트로 했는가"(개정 단위)입니다.

> ⓘ Operation Type과 Prompt Version 라벨링은 Python 에이전트만 지원합니다.

사용 방법

(a) 데코레이터(권장)

```
from whatap.llm import prompt_meta

@prompt_meta(operation_type='checkout_chain', prompt_version='v3')
def checkout(question):
    return client.chat.completions.create(...)
```

(b) 컨텍스트 매니저

```
from whatap.llm import prompt_meta_scope

def handler(req):
    with prompt_meta_scope(operation_type='greeting', prompt_version='v2'):
        return client.chat.completions.create(...)
```

현재 활성 meta 조회(커스텀 로직에서)

```
from whatap.llm import get_prompt_meta

op_type, version = get_prompt_meta()  # 스코프가 없으면 ('default', 'v1') 반환
```

값을 지정하지 않은 경우

`operation_type`은 라벨이 없을 때 감싸는 워크플로우/에이전트 이름으로 자동 대체됩니다. 반면 프롬프트 문안의 개정 여부는 와탭 에이전트가 알 수 없는 정보이므로, `prompt_version`은 `llm_prompt_default_version` 값으로 기록됩니다. 이 옵션을 설정하지 않으면 `v1`로 기록됩니다.

프롬프트 버전 일괄 수정

프롬프트를 일괄 개정했는데 데코레이터를 일일이 손대기 어려운 경우 whatap.conf에서 기본 버전을 옮깁니다. 이미 명시 라벨을 붙여둔 호출은 영향을 받지 않습니다.

whatap.conf

```
llm_prompt_default_version=v2
```

설정은 런타임에 반영되므로 재시작 없이 다음 호출부터 새 기본값이 적용됩니다.

Session & User ID 설정하기

Session ID와 User ID를 설정하면 LLM 호출을 세션·사용자 단위로 묶어서 추적할 수 있습니다. 설정 방법은 두 가지이며, 여러 방법이 동시에 적용된 경우 우선순위에 따라 하나만 반영됩니다.

코드(API) 방식

요청 핸들러에서 `set_session_id()`, `set_user_id()`를 호출하면, 해당 요청 컨텍스트 안에서 발생하는 모든 LLM 호출에 Session ID와 User ID가 적용됩니다.

```
from whatap.llm import set_user_id, set_session_id

@app.post("/chat")
async def chat(req):
    set_user_id(req.user.id)              # 누가
    set_session_id(req.conversation_id)   # 어느 대화
    ...
```

두 함수는 **활성 트랜잭션 컨텍스트에 값을 기록**합니다. 따라서 요청 핸들러 안이나 계측된 워크플로우 안에서 호출해야 하며, 트랜잭션 밖(모듈 로드 시점, 워커 시작 전 등)에서 호출하면 아무 동작도 하지 않습니다. 반환값(`bool`)으로 적용 여부를 확인할 수 있습니다.

스코프 지정

요청 핸들러가 아닌 곳(백그라운드 작업, 배치)에서 구간을 명시하려면 컨텍스트 매니저를 사용합니다. 블록을 나가면 이전 값으로 복원됩니다.

```
from whatap.llm import session, user

with user(user_id):
    with session(conversation_id):
        client.chat.completions.create(...)
```

HTTP 헤더 방식

Session ID와 User ID가 HTTP 요청 헤더로 전달되는 경우, whatap.conf 설정만으로 별도의 코드 수정 없이 적용할 수 있습니다.

whatap.conf

```
llm_session_header=X-Conversation-Id
llm_user_header=X-User-Id
```

위 예시에서는 `X-Conversation-Id` 헤더 값이 Session ID로, `X-User-Id` 헤더 값이 User ID로 사용됩니다.

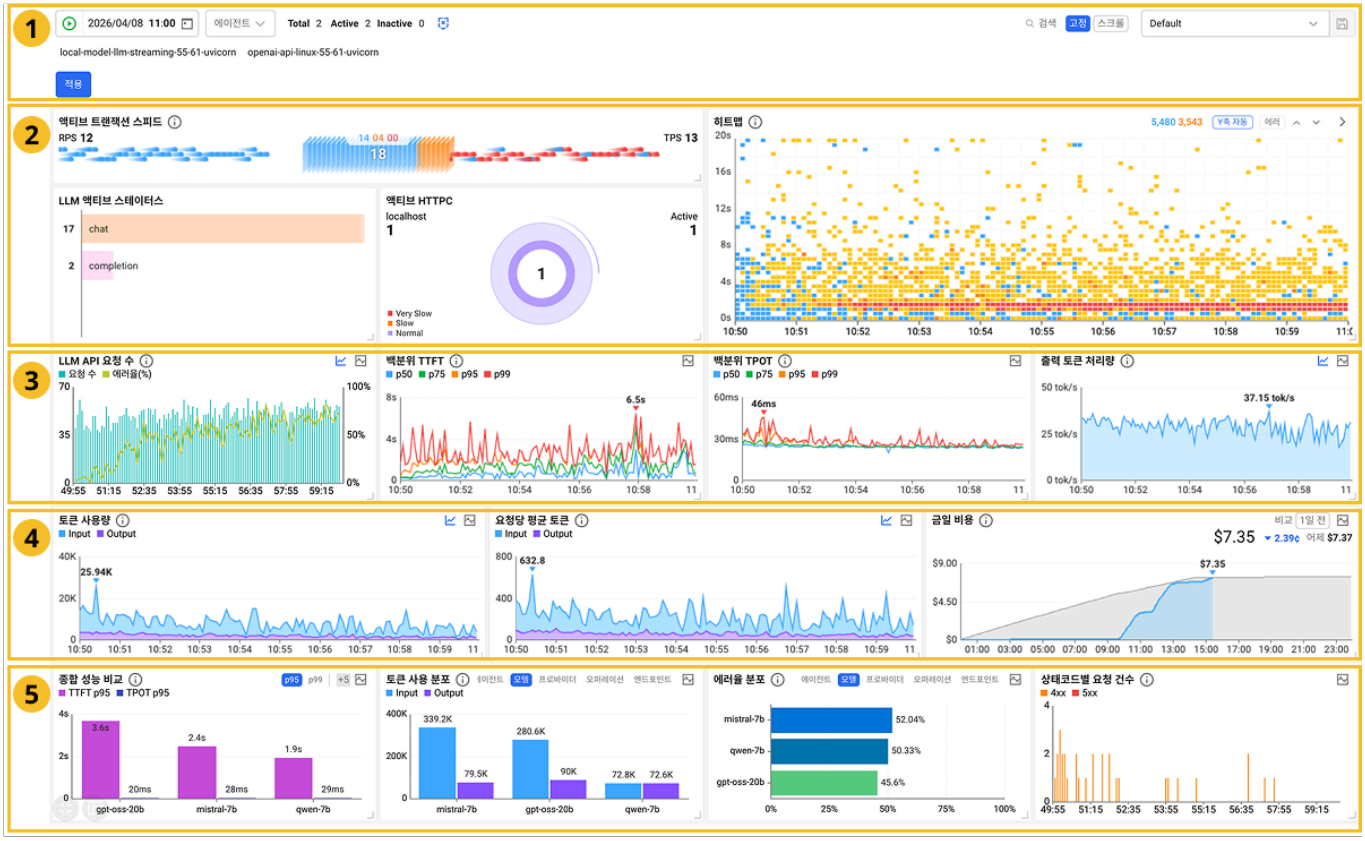
자동 계측 지원 프레임워크

다음 프레임워크는 별도 설정 없이 Session ID와 User ID가 자동으로 수집됩니다.

프레임워크	수집 방식
hermes-agent	게이트웨이 인바운드(`GatewayRunner._handle_message`)에서 이벤트 정보로부터 자동 수집합니다. **User ID**는 메신저 플랫폼의 사용자 식별자(telegram, slack, whatsapp 등 어댑터가 채우는 값)이며, **Session ID**는 hermes가 대화 이력을 묶는 세션 키를 그대로 사용합니다.

LLM 대시보드

LLM 대시보드는 LLM(Large Language Model) API의 요청량, 응답 성능, 토큰 사용량, 비용, 에러 현황을 한 화면에서 실시간으로 모니터링하는 메뉴입니다. 화면 위에서 아래로 **실시간 상태 → 요청과 성능 → 토큰과 비용 → 비교와 에러** 순서로 위젯이 배치되어, 현황 파악부터 원인 분석까지 자연스럽게 따라갈 수 있습니다. 이 구성은 **기본 프리셋**으로 저장되어 있으며, 별도 설정 없이 즉시 모니터링을 시작할 수 있습니다.



LLM 대시보드는 기본 프리셋 기준으로 5개 영역으로 구성됩니다.

영역	이름	설명
①	옵션바	조회 시간, 에이전트 필터, 화면 모드, 프리셋을 설정합니다.
②	실시간 상태	현재 트랜잭션 속도, LLM 호출 유형, 액티브 HTTPC, 히트맵을 실시간으로 확인합니다.

영역	이름	설명
③	LLM 성능 지표	API 요청량, TTFT·TPOT 백분위, 출력 토큰 처리량으로 LLM 응답 품질을 파악합니다.
④	토큰 및 비용	입출력 토큰 사용량, 요청당 평균 토큰, 금일 비용을 추적합니다.
⑤	모델별 비교·분석	모델 간 성능·토큰·에러를 비교하고 상태 코드별 요청 건수를 확인합니다.

옵션바

대시보드 상단 옵션바에서 조회 시간, 에이전트 필터, 화면 모드(고정/스크롤), 프리셋을 설정합니다. 각 기능의 상세 사용법은 대시보드 공통 기능 문서를 참조하세요.

실시간 상태

애플리케이션의 실시간 상태를 보여줍니다.

- **액티브 트랜잭션 스피드** — 초당 요청 수(RPS)와 응답 수(TPS), 진행 중인 트랜잭션 건수를 실시간으로 표시합니다.
- **LLM 액티브 스테이터스** — 현재 처리 중인 LLM 호출을 유형별(`chat` , `completion` 등)로 구분해 보여줍니다.
- **액티브 HTTPC** — 외부 LLM API로 나가는 HTTP 요청의 실시간 상태를 속도 구간(Very Slow, Slow, Normal)으로 분류합니다.
- **히트맵** — 완료된 트랜잭션의 응답 시간을 시간 축(X)과 경과 시간 축(Y)에 점으로 표시합니다. 점이 위로 올라갈수록 응답이 느린 트랜잭션입니다.

LLM 성능 지표

LLM API의 핵심 성능 지표를 모아 봅니다.

- **LLM API 요청 수** — 시간대별 요청 건수(바)와 에러율(라인)을 함께 표시합니다. 요청 급증과 에러율 상승이 동시에 나타나면 Rate Limit 초과나 프로바이더 장애를 의심할 수 있습니다.
- **백분위 TTFT** — 첫 번째 토큰이 도착하기까지 걸린 시간(Time To First Token)을 p50·p75·p95·p99 백분위로 보여줍니다. TTFT가 높으면 사용자가 체감하는 초기 대기 시간이 길어집니다.

- **백분위 TPOT** — 토큰 하나를 생성하는 데 걸리는 시간(Time Per Output Token)을 백분위로 보여줍니다. TPOT가 높으면 스트리밍 응답이 끊기거나 느려질 수 있습니다.
- **출력 토큰 처리량** — 초당 출력 토큰 수(tok/s)를 표시합니다. 처리량이 갑자기 떨어지면 프로바이더 측 성능 저하를 의심할 수 있습니다.

토큰 및 비용

토큰 사용량과 비용을 추적합니다.

- **토큰 사용량** — 시간대별 입력(Input)·출력(Output) 토큰 사용량을 추이 차트로 보여줍니다.
- **요청당 평균 토큰** — 요청 한 건당 평균 입력·출력 토큰 수를 표시합니다. 프롬프트 최적화 효과를 확인할 때 유용합니다.
- **금일 비용** — 오늘 누적 비용과 전일 대비 증감을 표시합니다. 비용 이상 징후를 빠르게 감지할 수 있습니다.

모델별 비교·분석

모델별 성능·토큰·에러를 비교합니다.

- **종합 성능 비교** — 모델별 TTFT p95, TPOT p95를 한 차트에서 비교합니다. 모델 교체나 버전 업그레이드 효과를 판단할 때 유용합니다.
- **토큰 사용 분포** — 모델별 입력·출력 토큰 사용량을 에이전트, 프로바이더, 오퍼레이션, 엔드포인트 기준으로 비교합니다.
- **에러율 분포** — 모델별 에러율을 에이전트, 프로바이더, 오퍼레이션, 엔드포인트 기준으로 비교합니다. 특정 모델에서 에러가 집중되는지 빠르게 확인할 수 있습니다.
- **상태코드별 요청 건수** — HTTP 상태 코드(4xx, 5xx)별 요청 건수를 시간대별로 표시합니다.

> ⓘ 위젯 편집, 위젯 옵션, 프리셋 등 대시보드 공통 기능은 대시보드 공통 기능 문서를 참조하세요.
>
> 각 위젯의 차트 유형, 필터, 활용 방법은 LLM 대시보드 위젯 문서를 참조하세요.

LLM 대시보드 공통 기능

대시보드에서 공통으로 사용할 수 있는 기능을 안내합니다. 조회 시간과 에이전트 필터로 데이터 범위를 좁히고, 위젯 옵션으로 차트를 조작하며, 프리셋으로 위젯 구성을 저장·관리할 수 있습니다.

조회 시간 설정

대시보드에서는 실시간 모니터링 기능을 기본 제공하지만, 화면 상단의 시간 선택자를 이용해 과거 시간의 데이터를 조회할 수도 있습니다. 시간 선택자에서 ⏸ 버튼을 선택한 다음 시간을 설정하세요. 사용자가 설정한 시간을 기준으로 대시보드에 배치한 위젯의 데이터를 갱신합니다.

에이전트 확인

에이전트 연결 상태 확인하기

화면 왼쪽 위, 시간 선택자의 오른쪽에서는 해당 프로젝트와 연결된 에이전트의 상태를 확인할 수 있는 정보를 제공합니다. 이를 통해 모니터링 대상 서버의 동작 여부를 바로 확인할 수 있습니다.

- **Total**: 프로젝트와 연결된 모든 에이전트의 수
- **Active**: 활성화된 에이전트의 수
- **Inactive**: 비활성화된 에이전트의 수
- : 비활성화된 에이전트를 표시하거나 감출 수 있습니다.

에이전트별 모니터링

> 에이전트를 하나 또는 둘 이상을 선택한 상태에서 다시 모든 에이전트를 선택하려면 선택을 해제하거나 **Total**을 클릭하세요.

> 프로젝트에 연결된 에이전트의 수가 많을 경우 에이전트의 이름을 짧게 설정하는 것이 효율적입니다. 에이전트 이름

> ⓘ 설정에 대한 자세한 내용은 다음 문서를 참조하세요.

에이전트 검색하기

프로젝트에 연결된 에이전트의 목록이 많아 찾기 어렵다면 에이전트 검색 기능을 이용하세요. 화면 오른쪽 위에 **검색** 버튼을 클릭하면 문자열을 입력할 수 있는 상자가 나타납니다.

입력한 문자열과 일치하는 에이전트만 에이전트 목록에 표시됩니다. 검색한 에이전트를 기준으로 대시보드의 데이터를 필터링하려면 문자 입력 상자 오른쪽에 **선택** 버튼을 클릭하세요.

선택한 에이전트 항목을 초기화하려면 문자 입력 상자 오른쪽에 **닫기** 버튼을 클릭하세요.

위젯 옵션

위젯에 표시된 아이콘 버튼의 기능은 다음과 같습니다.

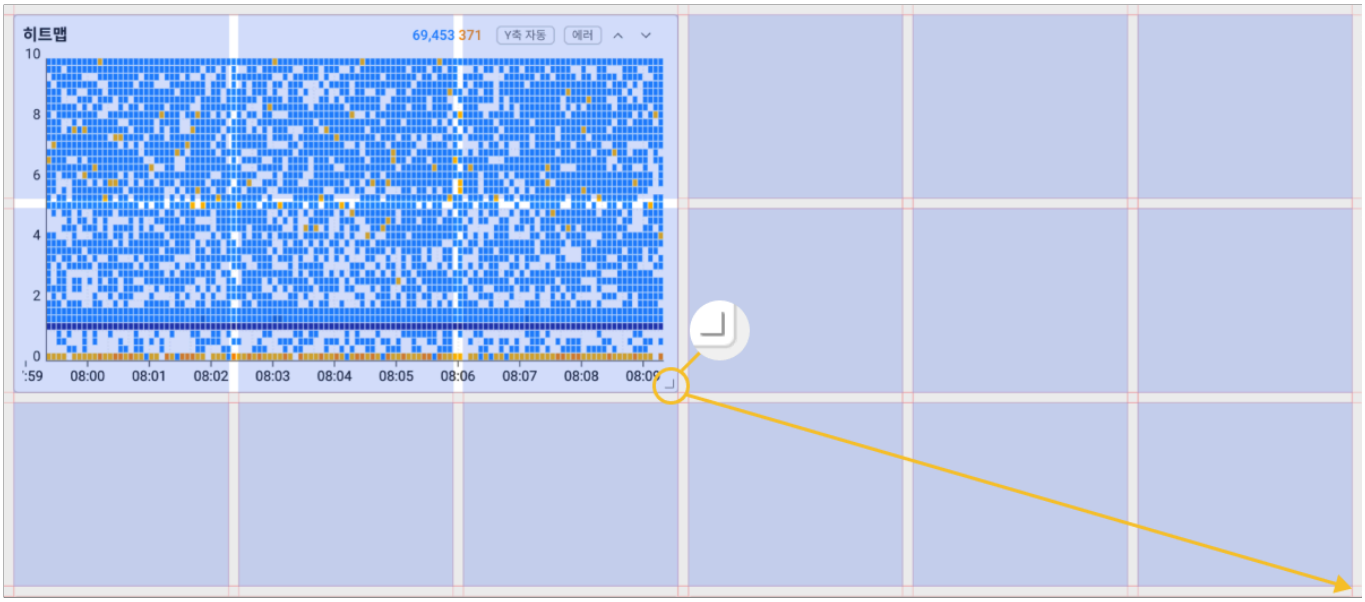
- : 주요 위젯에 대한 기능 및 정보를 확인할 수 있습니다.
- **병합 / 개별로 보기**: 해당 위젯 항목의 에이전트 데이터를 개별 또는 병합해 그래프로 표시합니다.
- **상세**: 해당 위젯 항목의 데이터를 에이전트별로 구분해 조회할 수 있는 모달 창이 나타납니다.

> ⓘ 위젯에 따라 제공되는 옵션은 다를 수 있습니다.

위젯 편집

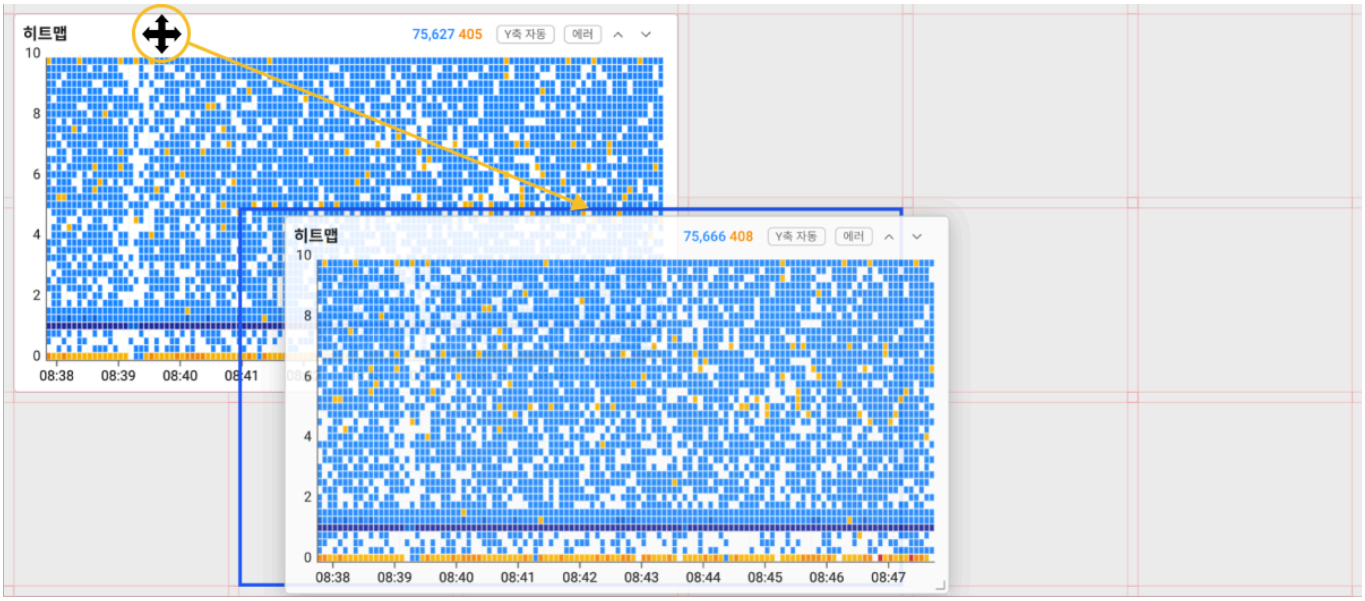
대시보드에 배치한 위젯은 사용자가 원하는 크기로 조절할 수 있고, 원하는 위치에 배치할 수 있습니다. 불필요하다고 생각되는 위젯은 삭제하고 다시 추가할 수도 있습니다.

위젯 크기 조절하기



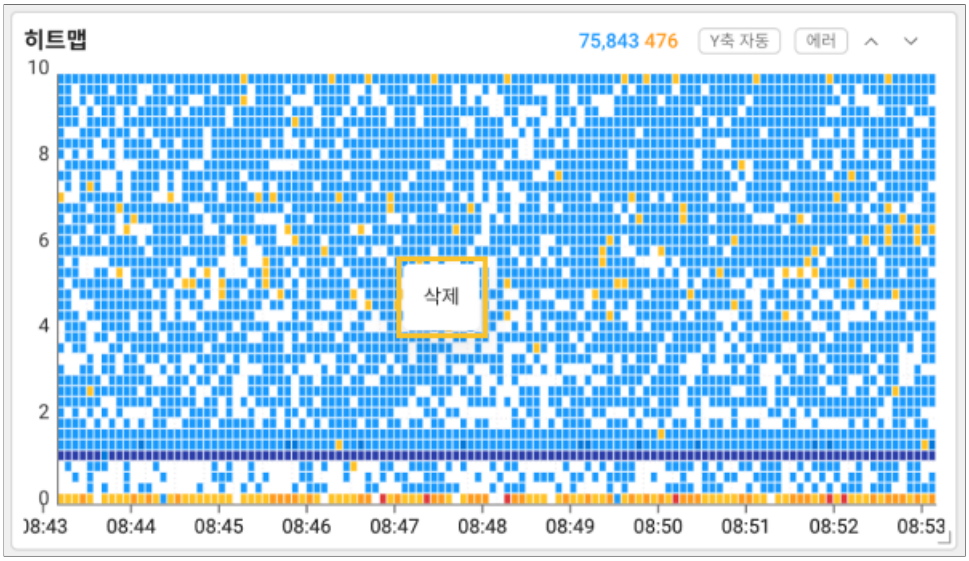
위젯의 오른쪽 아래에 ⌟ 요소를 마우스로 클릭한 상태에서 원하는 크기로 드래그하세요. 균일한 가로, 세로 비율의 격자가 표시되고, 격자 단위로 위젯의 크기를 조절할 수 있습니다.

위젯 이동하기



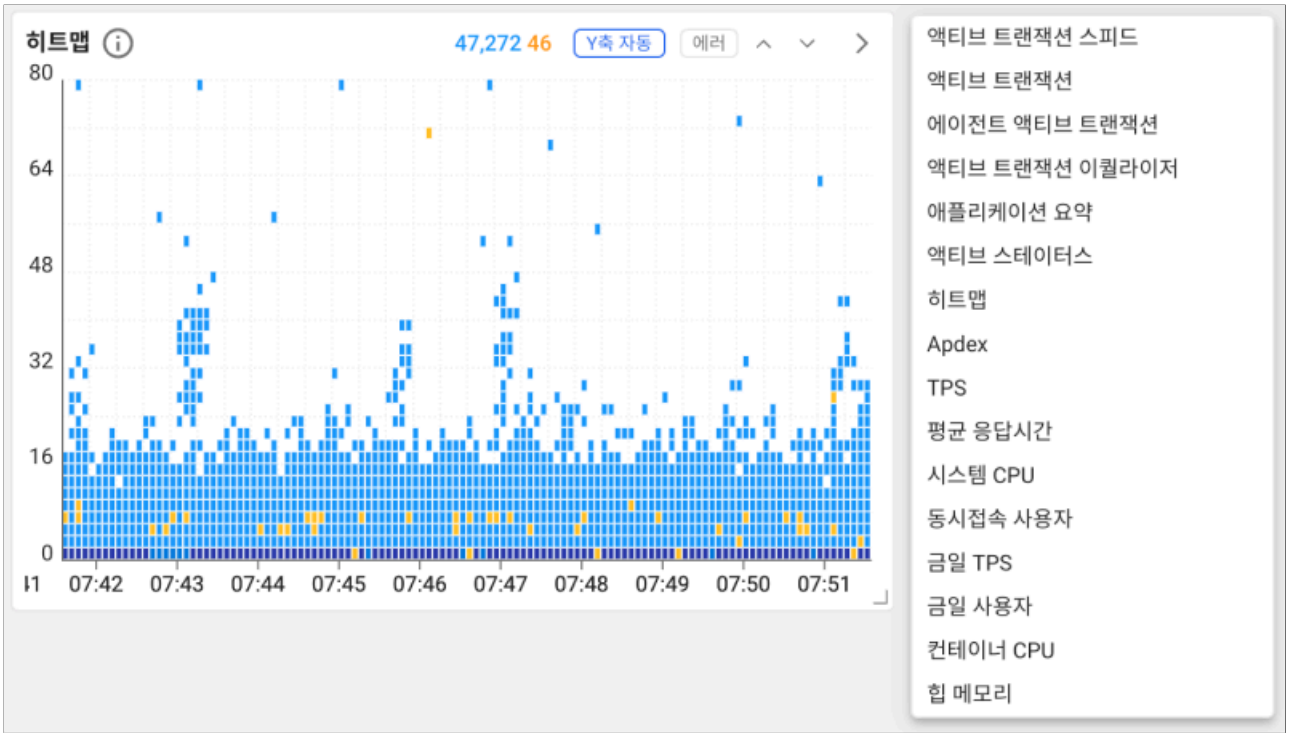
위젯의 윗부분으로 마우스 커서를 이동하면 커서 모양이 ✛ 모양으로 변경됩니다. 이때 마우스 왼쪽 버튼을 클릭한 상태로 원하는 위치로 드래그하여 위젯을 이동할 수 있습니다.

위젯 삭제하기



삭제할 위젯에서 마우스 오른쪽 버튼을 클릭하세요. **삭제** 버튼을 선택하면 해당 위젯이 대시보드에서 삭제됩니다.

위젯 추가하기



대시보드에서 빈 공간으로 마우스 커서를 이동한 다음 마우스 오른쪽 버튼을 클릭하세요. 팝업 메뉴에서 추가하려는 위젯을 선택하세요. 원하는 위치로 위젯을 배치하고 크기를 조절하세요.

프리셋 설정

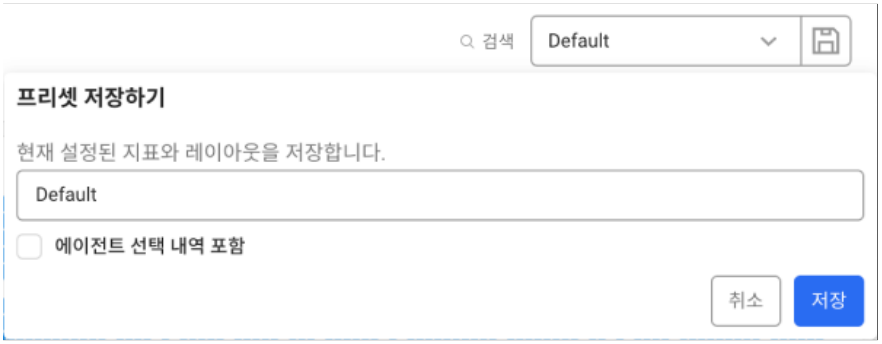
대시보드에서 사용자가 설정한 위젯의 설정과 레이아웃 상태를 저장하고 불러올 수 있습니다. 위젯의 크기를 조절하고, 원하는 위치에 배치해 새로운 프리셋을 만들 수 있습니다.

- **Default**: LLM 모니터링에 필요한 주요 지표를 한 화면에 요약한 기본 프리셋입니다. 실시간 액티브 상태, 요청·성능 핵심 지표, 토큰·비용, 비교·에러 등의 흐름으로 위젯이 배치됩니다.

> ⓘ
> - **기본**으로 설정된 프리셋은 변경할 수 없습니다.
> - **프로젝트 수정** 권한이 있는 멤버만 프리셋을 저장하거나 수정할 수 있습니다. 멤버 권한에 대한 자세한 내용은 다음 문서를 참조하세요.

새로운 프리셋 만들기

1. 대시보드에서 원하는 형식으로 위젯을 배치해 보세요. 크기를 조절하고 자주 확인하는 위젯만 배치할 수도 있습니다.
2. 화면 오른쪽 위에 버튼을 선택하세요.
3. 새로운 프리셋 이름을 입력하세요.



에이전트 선택 내역을 같이 저장하려면 **에이전트 선택 내역 포함**을 선택하세요.

4. **저장** 버튼을 선택하세요.

프리셋 목록에서 새로 저장한 프리셋을 확인할 수 있습니다.

- 새로 만든 프리셋에 변경 사항이 생겼다면 다시 프리셋을 저장해야 합니다. 버튼을 선택한 다음 같은 이름으로 프리셋을 저장하세요. 기존의 프리셋에 변경 사항을 덮어쓰기합니다.
- 대시보드의 변경 사항을 저장하지 않고 다른 메뉴로 이동하면 변경 사항은 저장되지 않습니다.
- 프리셋은 프로젝트 단위로 저장되어 다른 사용자와 공유할 수 있습니다.

프리셋 삭제하기

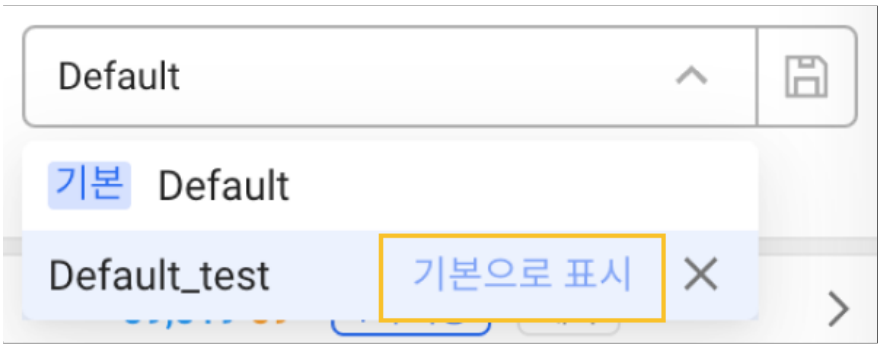
사용하지 않는 프리셋이 있다면 프리셋 목록에서 삭제할 수 있습니다. 프리셋 목록에서 삭제하려는 항목의 오른쪽에 ✕ 버튼을 선택하세요.

Default 프리셋은 삭제할 수 없습니다.

기본 프리셋 변경하기

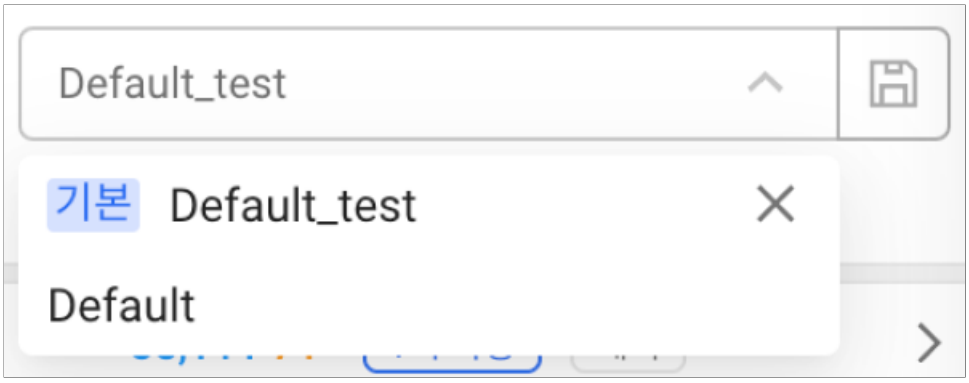
애플리케이션 대시보드 메뉴로 처음 진입했을 때 보여지는 프리셋을 변경할 수 있습니다.

1. 대시보드의 오른쪽 위에 프리셋 선택 상자를 선택하세요.
2. 기본으로 설정할 프리셋에 마우스를 오버하세요.



3. **기본으로 표시** 버튼이 나타나면 선택하세요.

선택한 프리셋이 기본 프리셋으로 변경되며, 프리셋 목록에 **기본** 태그가 표시됩니다.



실시간 알림

화면 오른쪽 위에 **실시간 알림** 버튼을 선택하면 최근 발생한 이벤트를 확인할 수 있습니다. 토글 메뉴를 클릭해 브라우저 알림을 켜거나 끌 수 있습니다.

화면 가장 위에 고정 메뉴 영역의 기본 요소들에 대한 자세한 내용은 다음 문서를 참조하세요.

LLM 대시보드 위젯 레퍼런스

이 문서는 LLM 대시보드에서 제공하는 위젯을 대시보드 영역별로 정리한 레퍼런스입니다. 각 위젯의 차트 유형, 필터, 활용 방법을 확인할 수 있습니다. 대시보드의 전체 화면 구성과 기본 프리셋은 LLM 대시보드 문서를 참조하세요.

제공 위젯

LLM 성능 지표

Widget	Description
LLM API 요청 수	시간대별 요청 건수 + 에러율
상태 코드별 요청 건수	HTTP 상태 코드(2xx/4xx/5xx) 분포
Latency 평균	시간대별 평균 응답 시간 추이
Latency 백분위	Latency p50·p75·p95·p99 추이
Latency 백분위 개별 위젯	백분위수별 개별 위젯
Latency 분포 비교	모델별 Latency 박스플롯
TTFT 평균	시간대별 TTFT 평균 추이
TTFT 백분위	TTFT p50·p75·p95·p99 추이
TTFT 백분위 개별 위젯	백분위수별 개별 위젯
TTFT 분포 비교	모델별 TTFT 박스플롯
TPOT 평균	시간대별 TPOT 평균 추이
TPOT 백분위	TPOT p50·p75·p95·p99 추이

Widget	Description
TPOT 백분위 개별 위젯	백분위수별 개별 위젯
TPOT 분포 비교	모델별 TPOT 박스플롯
종합 성능 비교	모델별 TTFT·TPOT·Output TPS 비교
출력 토큰 처리량	초당 출력 토큰 수(tok/s)

토큰 및 비용

Widget	Description
토큰 사용량	시간대별 입력·출력 토큰 추이
요청당 평균 토큰	요청 1건당 평균 입력·출력 토큰
토큰 사용 분포	모델별 입력·출력 토큰 총량 비교
금일 토큰 사용량	금일 누적 토큰 + 전일 대비
캐시 적중률	캐시 토큰 비율(%) 추이
캐시 적중률 분포	모델별 캐시 적중률 비교
캐시 절감 비용	캐싱으로 절약한 금액($) 추이
캐시 절감 비용 분포	모델별 캐시 절감 비용 비교
비용 사용량	시간대별 입력·출력 비용($) 추이
요청당 평균 비용	요청 1건당 평균 비용
비용 사용 분포	모델별 비용 총량 비교

Widget	Description
금일 비용	금일 누적 비용 + 전일 대비

모델별 비교·분석

Widget	Description
종합 성능 비교	모델별 TTFT·TPOT·Output TPS 비교
토큰 사용 분포	모델별 입력·출력 토큰 총량 비교
에러율 분포	모델별 에러율(%) 비교
에러 건수	시간대별 API 에러·프로그램 에러 건수
상태 코드별 요청 건수	HTTP 상태 코드(4xx/5xx) 분포

라이브 관측

Widget	Description
액티브 LLM API	실행 중인 LLM API 호출과 장기 실행 목록
액티브 트라잭토리	진행 중 Trajectory 수와 등급별 분포

제공 위젯 상세 설명

LLM API 요청

LLM API의 요청량과 상태 코드를 모니터링하는 위젯 그룹입니다.

LLM API 요청 수

Property	Value
차트 유형	바 차트(요청 수) + 라인 차트(에러율) 복합
필터	태그 필터
액션	개별로 보기, 병합 보기, 상세 보기

LLM API에 전달된 시간대별 요청 건수와 에러율(%)을 함께 표시합니다. 바 차트(좌측 Y축)는 요청량, 라인(우측 Y축)은 전체 요청 대비 에러 비율입니다.

- 요청량이 급증하면서 에러율도 함께 상승하면 Rate Limit 초과 또는 프로바이더 장애 가능성이 있습니다.
- 요청량은 변화 없는데 에러율만 상승하면 특정 모델이나 에이전트에서 문제가 발생하고 있을 수 있으므로, **개별로 보기**로 원인을 좁혀볼 수 있습니다.

상태 코드별 요청 건수

Property	Value
차트 유형	스택 바 차트
필터	없음
액션	상세 보기

시간대별 HTTP 상태 코드 분포를 2xx(성공), 4xx(클라이언트 에러), 5xx(서버 에러)로 나누어 표시합니다.

- **4xx 증가** — 잘못된 요청 파라미터, 인증 실패, 토큰 한도 초과 등 요청 측 문제를 점검해야 합니다.
- **5xx 증가** — LLM 프로바이더 측 서버 장애나 일시적 과부하 상태를 의미합니다.
- **LLM API 요청 수** 위젯의 에러율과 함께 보면 에러 원인을 빠르게 분류할 수 있습니다.

LLM API 응답 성능

LLM API의 전체 응답 시간(Latency)을 다양한 관점에서 분석하는 위젯 그룹입니다. Latency는 요청을 보낸 시점부터 응답이 완전히 완료되기까지의 전체 소요 시간입니다.

Latency 평균

Property	Value
차트 유형	라인 차트
필터	태그 필터
액션	개별로 보기, 병합 보기, 상세 보기

시간대별 Latency 평균 추이를 표시합니다.

- Latency 평균가 전반적으로 상승하면 모델 응답 속도 저하나 네트워크 지연을 점검해야 합니다.
- **개별로 보기**로 모델별 Latency를 비교하여 어떤 모델에서 지연이 발생하는지 확인할 수 있습니다.

Latency 백분위

Property	Value
차트 유형	라인 차트 (p50, p75, p95, p99 4개 시리즈)
필터	없음
액션	상세 보기

시간대별 Latency의 백분위수 추이를 p50, p75, p95, p99 네 개 시리즈로 표시합니다. 예를 들어 p95 값이 3초이면 전체 요청 중 95%가 3초 이내에 완료되었음을 의미합니다.

- p50과 p99의 격차가 크면 대부분의 요청은 빠르지만 일부 요청에서 심한 지연이 발생하고 있다는 의미입니다.
- p50과 p99가 함께 상승하면 전체적인 성능 저하를 의미합니다.

Latency 백분위 개별 위젯

Property	Value
차트 유형	라인 차트
필터	태그 필터
액션	개별로 보기, 병합 보기, 상세 보기

각 백분위수를 개별 위젯으로 제공합니다. 백분위수별 의미는 다음과 같습니다.

Percentile	Description
`p50`	전체 요청 중 절반이 이 시간 이내에 완료. 일반적인 사용자가 체감하는 평균적인 응답 속도.
`p75`	전체 요청 중 75%가 이 시간 이내에 완료. p50과의 격차로 상위 25% 요청의 추가 지연 파악.
`p95`	전체 요청 중 95%가 이 시간 이내에 완료. SLA(서비스 수준 협약) 기준 지표로 자주 사용.
`p99`	전체 요청 중 99%가 이 시간 이내에 완료. 가장 느린 상위 1% 요청, 시스템 안정성 판단.

Latency 분포 비교

Property	Value
차트 유형	박스플롯 (x축: 모델)
필터	태그 필터 (상시 노출)
액션	상세 보기

태그 기준(모델 등)별로 Latency의 분포를 박스플롯으로 비교합니다. 박스 영역은 p25 ~ p75(중간 50% 범위), 중앙 선은 중앙값, 위스커는 최솟값 ~ 최댓값을 나타냅니다.

- 박스가 넓은 모델은 응답 시간 편차가 크므로 안정성 개선이 필요할 수 있습니다.

- 위스커가 길게 늘어진 모델은 간헐적으로 극단적인 지연이 발생하고 있음을 의미합니다.
- 여러 모델의 분포를 나란히 비교하여 가장 안정적인 모델을 식별할 수 있습니다.

TTFT (Time To First Token)

TTFT는 요청을 보낸 후 첫 번째 응답 토큰이 도착하기까지의 시간입니다. TTFT가 길어지면 사용자 입장에서 "응답이 시작되지 않는다"는 체감을 주게 됩니다.

TTFT 평균

Property	Value
차트 유형	라인 차트
필터	태그 필터
액션	개별로 보기, 병합 보기, 상세 보기

시간대별 TTFT 평균 추이를 표시합니다.

- 특정 시간대에 TTFT가 급등하면 모델 큐 대기 증가나 프로바이더 측 지연을 점검해야 합니다.
- **개별로 보기**로 어떤 모델에서 초기 응답이 느린지 비교할 수 있습니다.

TTFT 백분위

Property	Value
차트 유형	라인 차트 (p50, p75, p95, p99 4개 시리즈)
필터	없음
액션	상세 보기

시간대별 TTFT의 백분위수 추이를 p50, p75, p95, p99 네 개 시리즈로 표시합니다. 예를 들어 p95 값이 2초이면 전체 요청 중 95%가 2초 이내에 첫 토큰을 수신했음을 의미합니다.

- p50은 안정적인데 p99만 급등하면 간헐적으로 큐 대기가 발생하고 있다는 신호입니다.
- 모든 백분위가 함께 상승하면 전반적인 초기 응답 지연이 발생하고 있습니다.

TTFT 백분위 개별 위젯

Property	Value
차트 유형	라인 차트
필터	태그 필터
액션	개별로 보기, 병합 보기, 상세 보기

각 백분위수를 개별 위젯으로 제공합니다.

Percentile	Description
`p50`	대부분의 사용자가 체감하는 초기 응답 대기 시간.
`p75`	p50과의 격차로 상위 25% 요청의 추가 초기 지연 파악.
`p95`	"대부분의 사용자에게 이 정도 초기 응답 대기는 보장됩니다"라는 SLA 기준 지표.
`p99`	가장 느린 상위 1%의 초기 응답 시간. 간헐적 타임아웃이나 특정 조건에서의 지연 점검.

TTFT 분포 비교

Property	Value
차트 유형	박스플롯 (x축: 모델)
필터	태그 필터 (상시 노출)
액션	상세 보기

태그 기준(모델 등)별로 TTFT의 분포를 박스플롯으로 비교합니다. 박스 영역은 p25~~p75, 중앙 선은 중앙값, 위스커는~~

최솟값최댓값입니다.

- 중앙값이 낮고 박스가 좁은 모델이 가장 안정적으로 빠른 초기 응답을 제공합니다.
- 위스커가 길게 늘어진 모델은 간헐적으로 극단적인 초기 지연이 발생합니다.

TPOT (Time Per Output Token)

TPOT는 출력 토큰이 하나 생성되는 데 걸리는 평균 시간입니다. TPOT가 낮을수록 스트리밍 응답이 부드럽게 느껴지며, 높아지면 "응답이 끊기는" 느낌을 줄 수 있습니다.

TPOT 평균

Property	Value
차트 유형	라인 차트
필터	태그 필터
액션	개별로 보기, 병합 보기, 상세 보기

시간대별 TPOT 평균 추이를 표시합니다.

- 특정 시간대에 TPOT가 급등하면 모델 부하 증가나 Rate Limit 영향을 점검해야 합니다.
- **개별로 보기**로 모델별 토큰 생성 속도를 비교할 수 있습니다.

TPOT 백분위

Property	Value
차트 유형	라인 차트 (p50, p75, p95, p99 4개 시리즈)
필터	없음
액션	상세 보기

시간대별 TPOT의 백분위수 추이를 p50, p75, p95, p99 네 개 시리즈로 표시합니다. 예를 들어 p95 값이 40ms이면 전체 요청 중

95%에서 토큰 하나가 40ms 이내에 생성되었음을 의미합니다.

- p50은 안정적인데 p99만 높아지면 간헐적으로 토큰 생성이 멈추는 현상이 발생하고 있다는 신호입니다.
- 모든 백분위가 함께 상승하면 전반적인 스트리밍 속도 저하를 의미합니다.

TPOT 백분위 개별 위젯

Property	Value
차트 유형	라인 차트
필터	태그 필터
액션	개별로 보기, 병합 보기, 상세 보기

각 백분위수를 개별 위젯으로 제공합니다.

Percentile	Description
`p50`	일반적인 스트리밍 속도. 가장 부드러운 스트리밍을 제공하는 모델 확인.
`p75`	p50과의 격차로 상위 25% 요청의 스트리밍 속도 저하 파악.
`p95`	SLA 기준 스트리밍 속도 지표.
`p99`	가장 느린 상위 1%의 토큰 생성 속도. 간헐적 멈춤 현상 점검.

TPOT 분포 비교

Property	Value
차트 유형	박스플롯 (x축: 모델)
필터	태그 필터 (상시 노출)
액션	상세 보기

태그 기준(모델 등)별로 TPOT의 분포를 박스플롯으로 비교합니다. 박스 영역은 p25~~p75, 중앙 선은 중앙값, 위스커는 최솟값~~최댓값입니다.

- 중앙값이 낮고 박스가 좁은 모델이 가장 안정적인 스트리밍 성능을 제공합니다.
- 위스커가 길게 늘어진 모델은 간헐적으로 토큰 생성이 크게 느려지는 현상이 발생합니다.

종합 성능 비교

Property	Value
차트 유형	수평 바 차트 (모델별, 3개 지표)
필터	태그 필터 (상시 노출)
액션	상세 보기, p95/p99 탭 전환

모델별 핵심 성능 지표 3가지를 수평 바 차트로 비교합니다. p95/p99 탭을 전환하여 일반적인 성능과 최악 조건에서의 성능을 각각 확인할 수 있습니다.

Metric	Description	Direction
`TTFT`	사용자가 첫 응답을 받기까지 대기하는 시간	낮을수록 좋음
`TPOT`	토큰 하나가 생성되는 데 걸리는 시간(스트리밍 속도)	낮을수록 좋음
`Output TPS`	초당 생성되는 출력 토큰 수(전체 처리량)	높을수록 좋음

세 지표를 종합하면 "빠르게 시작하고, 부드럽게 스트리밍하며, 높은 처리량을 내는 모델"을 한눈에 식별할 수 있습니다.

토큰

토큰 사용 패턴과 캐시 효율을 모니터링하는 위젯 그룹입니다.

토큰 사용량

Property	Value
차트 유형	라인 차트 (input, output 2개 시리즈)
필터	태그 필터
액션	개별로 보기, 병합 보기, 상세 보기

시간대별 입력 토큰 수와 출력 토큰 수의 추이를 표시합니다. 입력 토큰은 프롬프트에 포함된 토큰, 출력 토큰은 모델이 생성한 응답 토큰입니다.

- 특정 시간대에 토큰 사용량이 급증하면 트래픽 증가, 프롬프트 변경, 비정상 요청 유입을 점검해야 합니다.
- **개별로 보기**로 어떤 모델이나 에이전트에 토큰 소비가 집중되는지 확인할 수 있습니다.

요청당 평균 토큰

Property	Value
차트 유형	라인 차트 (input, output 2개 시리즈)
필터	태그 필터
액션	개별로 보기, 병합 보기, 상세 보기

요청 1건당 사용된 평균 입력/출력 토큰 수의 시간대별 추이를 표시합니다.

- 요청당 입력 토큰이 급증하면 프롬프트 길이가 길어졌거나 긴 컨텍스트가 포함되기 시작했을 수 있습니다.
- 요청당 출력 토큰이 급증하면 응답 길이가 늘어난 것으로, 비용 증가에 직접 영향을 줍니다.

출력 토큰 처리량

Property	Value
차트 유형	라인 차트

Property	Value
필터	태그 필터
액션	개별로 보기, 상세 보기

초당 생성되는 출력 토큰 수(Tokens Per Second)를 나타내는 처리량 지표입니다.

- 처리량이 떨어지는 구간은 모델 응답 병목이나 Rate Limit 영향을 점검해야 합니다.
- 토큰 사용량과 함께 보면 "토큰은 많이 사용하는데 처리 속도가 느린 구간"을 식별할 수 있습니다.

토큰 사용 분포

Property	Value
차트 유형	히스토그램 (태그별 그룹)
필터	태그 필터 (상시 노출)
액션	상세 보기

태그 기준(모델 등)별로 입력/출력 토큰 총량을 비교하는 분포 차트입니다.

- 어떤 모델이 가장 많은 토큰을 소비하는지 한눈에 확인할 수 있습니다.
- 입력 비중이 높은 모델은 프롬프트 최적화로 비용을 절감할 여지가 있습니다.
- 출력 비중이 높은 모델은 응답 길이 제한(`max_tokens`) 조정을 검토할 수 있습니다.

금일 토큰 사용량

Property	Value
차트 유형	일간 비교 차트 (금일/전일)
필터	없음
액션	날짜 비교

금일 누적 토큰 사용량(input + output)과 전일 동일 시간대 대비 증감을 표시합니다.

- 전일 대비 토큰 소비 속도가 빠르면 트래픽 증가나 프롬프트 크기 변화를 점검해야 합니다.
- Rate Limit 한도나 일일 예산 초과를 사전에 감지하는 데 활용할 수 있습니다.

캐시 적중률

Property	Value
차트 유형	라인 차트 (%)
필터	태그 필터
액션	개별로 보기, 상세 보기

전체 입력 토큰 중 캐시에서 가져온 토큰의 비율(%)을 시간대별로 표시합니다.

- 적중률이 높을수록 프롬프트 캐싱이 효과적으로 동작하고 있으며, 비용 절감 효과가 큽니다.
- 적중률이 갑자기 하락하면 프롬프트 내용 변경, 캐시 TTL 만료, 새로운 유형의 요청 유입을 점검해야 합니다.
- **개별로 보기**로 모델별 캐시 효율을 비교할 수 있습니다.

> ⓘ 캐시 적중률 = (구간 내 `cached_tokens` 합계 / 구간 내 `input_tokens` 합계) × 100

캐시 적중률 분포

Property	Value
차트 유형	히스토그램 (태그별)
필터	태그 필터 (상시 노출)
액션	상세 보기

태그 기준(모델 등)별로 캐시 적중률(%)을 비교하는 분포 차트입니다.

- 적중률이 높은 모델은 캐싱이 잘 활용되고 있으며, 낮은 모델은 캐싱 대상 프롬프트 구조를 점검해야 합니다.
- 모델 간 적중률 격차가 크면 캐싱 전략을 모델별로 차별화하여 비용을 추가로 절감할 수 있습니다.

캐시 절감 비용

Property	Value
차트 유형	라인 차트 ($)
필터	태그 필터
액션	개별로 보기, 상세 보기

프롬프트 캐싱을 통해 실제로 절약된 금액($)을 시간대별로 표시합니다. 캐시된 토큰은 일반 입력 토큰보다 낮은 단가가 적용되며, 이 차이가 절감 비용으로 산출됩니다.

- 절감 비용이 꾸준히 발생하면 캐싱 전략이 효과적으로 동작하고 있다는 의미입니다.
- 절감 비용이 감소하면 캐시 적중률과 함께 확인하여 캐시 미스 증가 여부를 점검해야 합니다.
- **개별로 보기**로 어떤 모델에서 가장 큰 절감 효과가 발생하는지 확인할 수 있습니다.

캐시 절감 비용 분포

Property	Value
차트 유형	히스토그램 (태그별)
필터	태그 필터 (상시 노출)
액션	상세 보기

태그 기준(모델 등)별로 캐시 절감 비용($)을 비교하는 분포 차트입니다.

- 절감 비용이 큰 모델은 캐싱 효과가 높으므로 해당 모델의 캐싱 설정을 유지·강화하는 것이 유리합니다.
- 절감 비용이 낮은 모델은 캐시 적중률과 함께 확인하여 캐싱 대상 프롬프트 구조 개선이나 캐시 TTL 조정을 검토할 수 있습니다.

비용

LLM API 사용 비용을 모니터링하는 위젯 그룹입니다.

비용 사용량

Property	Value
차트 유형	라인 차트 (input, output 2개 시리즈)
필터	태그 필터
액션	개별로 보기, 병합 보기, 상세 보기

시간대별 입력/출력 비용($) 추이를 표시합니다. 입력 비용은 프롬프트 처리 비용, 출력 비용은 응답 생성 비용입니다.

- 비용이 급증하는 시간대가 있으면 해당 구간의 요청량과 요청당 토큰 수를 함께 확인해야 합니다.
- 입력 비용과 출력 비용의 비중을 비교하여 프롬프트 최적화와 응답 길이 제한 중 어디에 집중할지 판단할 수 있습니다.
- **개별로 보기**로 모델별 비용 기여도를 비교할 수 있습니다.

요청당 평균 비용

Property	Value
차트 유형	라인 차트 (input, output 2개 시리즈)
필터	태그 필터
액션	개별로 보기, 병합 보기, 상세 보기

요청 1건당 발생하는 평균 입력/출력 비용($)의 시간대별 추이를 표시합니다.

- 요청당 비용이 갑자기 변하면 모델 전환, 프로바이더 가격 변경, 프롬프트 크기 변화를 점검해야 합니다.
- 총 비용은 증가했지만 요청당 비용은 변동 없다면 단순히 요청량이 늘어난 것이므로, 요청량 관리에 집중하면 됩니다.

비용 사용 분포

Property	Value
차트 유형	히스토그램 (태그별 그룹)
필터	태그 필터 (상시 노출)
액션	상세 보기

태그 기준(모델 등)별로 입력/출력 비용 총량을 비교하는 분포 차트입니다.

- 어떤 모델이 가장 높은 비용을 차지하는지 한눈에 확인할 수 있습니다.
- 비용이 높은 모델의 사용 빈도와 요청당 토큰을 점검하여 비용 최적화 대상을 선정할 수 있습니다.

금일 비용

Property	Value
차트 유형	일간 비교 차트 (금일/전일)
필터	없음
액션	날짜 비교

금일 누적 비용($)과 전일 동일 시간대 대비 증감을 표시합니다.

- 전일 대비 비용 소비 속도가 빠르면 요청량 증가, 모델 전환, 프롬프트 크기 변화를 점검해야 합니다.
- 일일 예산 초과를 사전에 감지하여 대응 조치를 취할 수 있습니다.

에러

LLM API의 에러 현황을 모니터링하는 위젯 그룹입니다.

에러율 분포

Property	Value
차트 유형	수평 바 차트
필터	태그 필터
액션	개별로 보기, 병합 보기, 상세 보기

태그 기준(모델 등)별로 에러율(%)을 수평 바 차트로 비교합니다. 에러율은 해당 그룹의 전체 요청 중 에러(`status_code` 400 이상)가 차지하는 비율입니다.

- 특정 모델이나 에이전트에 에러가 집중되어 있는지 빠르게 파악할 수 있습니다.
- 에러율이 높은 모델은 프로바이더 장애, 잘못된 파라미터, Rate Limit 초과 등을 점검해야 합니다.
- **개별로 보기**로 `Agent` , `Provider` 등 다양한 기준으로 에러 원인을 좁힐 수 있습니다.

에러 건수

Property	Value
차트 유형	스택 바 차트
필터	없음
액션	상세 보기

시간대별 에러 건수를 API 에러와 프로그램 에러로 구분하여 표시합니다.

- **API 에러** — LLM 프로바이더 측에서 반환한 에러로, Rate Limit 초과, 서버 오류, 인증 실패 등이 해당됩니다.
- **프로그램 에러** — 클라이언트 측에서 발생한 에러로, 응답 파싱 실패, 타임아웃, 연결 오류 등이 해당됩니다.
- 에러 유형별 추이를 통해 문제 원인이 프로바이더 측인지 클라이언트 측인지 빠르게 구분할 수 있습니다.

라이브 관측

지금 실행 중인 LLM 호출과 Trajectory를 확인하는 위젯 그룹입니다. Trajectory는 하나의 LLM 작업 흐름에 참여한 트랜잭션을 하나로 묶은 단위입니다. 두 위젯의 목록 화면은 에이전트에서 직접 조회한 실행 중 데이터를 사용하므로 과거 구간이 없습니다. 대시보드를 정지했거나 과거 시각을 보고 있으면 목록으로 이어지는 동선을 제공하지 않습니다.

액티브 LLM API

Property	Value
차트 유형	스웜 차트(모델별 트랙)
필터	대시보드 에이전트 필터
액션	상세 보기, 새 창 열기

현재 실행 중인 LLM API 호출을 모델별 트랙 위에 점으로 표시합니다. 점 하나가 호출 하나이며, 가로 위치는 `Normal`, `Slow`, `Very Slow` 구간을 나타냅니다. 세 구간의 기준은 에이전트 옵션에 설정한 값에 따라 데이터가 집계됩니다.

- `Very Slow` 구간에 점이 쌓이면 에이전트에 설정한 기준보다 오래 실행되는 호출이 특정 모델이나 프로바이더에 몰렸는지 확인하세요.
- 위젯 본문을 클릭하면 장기 실행 목록이 모달로 열립니다. 프롬프트 펼치기를 통해 실행 중인 시스템 프롬프트와 입력 프롬프트를 확인할 수 있으며, 목록에서 호출을 선택하면 그 호출이 속한 Trajectory 뷰로 이동해 전체 실행 흐름을 확인할 수 있습니다.
- 위젯 헤더의 **상세 보기** 버튼을 클릭하면 같은 목록을 넓은 화면으로 열 수 있습니다. 모달 헤더의 **새 창 열기** 버튼을 클릭하면 별도 창으로 띄웁니다.

액티브 트라잭토리

Property	Value
차트 유형	궤도 차트 + 등급별 건수 목록
필터	대시보드 에이전트 필터

Property	Value
액션	상세 보기, 새 창 열기

현재 활성화된 Trajectory 수와 각 Trajectory의 Step 개수를 `Light`, `Middle`, `Heavy` 등급으로 나누어 표시합니다. 등급은 Step을 얼마나 많이, 반복해서 호출했는지를 나타내며 속도나 실행 시간을 뜻하지 않습니다.

- 소속 멀티 트랜잭션 중 하나라도 실행 중이면 그 Trajectory를 라이브 상태로 셉니다.
- 등급 경계값은 에이전트 옵션에 설정한 기준을 따릅니다.
- 등급을 클릭하면 그 등급으로 걸러진 Trajectory 목록이 열립니다. 목록에서 행을 선택하면 Trajectory 뷰로 이어집니다.
- 궤도 차트를 클릭하면 등급을 거르지 않은 전체 목록이 열립니다.

Trajectory 목록

목록은 Trajectory 단위 그룹 헤더와 그 아래 트랜잭션 행으로 구성됩니다. 그룹 헤더에는 `mtid`, 묶인 트랜잭션 수, 사용자, 세션을 표시하며, `mtid`가 없는 단독 흐름은 단독 실행으로 표시합니다. 목록은 CSV로 내보낼 수 있습니다.

Column	Description
`Transaction`	트랜잭션 이름
상태	트랜잭션 진행 상태(실행 중·완료·실패)
`mtid`	Trajectory를 묶는 키
스텝	트랜잭션이 수행한 스텝 수
에이전트	트랜잭션을 수행한 에이전트
LLM 호출 수	트랜잭션 안에서 발생한 LLM 호출 수
경과	트랜잭션 시작 후 경과 시간
사용자	로그에 부착된 `user_id`
세션	로그에 부착된 `session_id`

참고

수집 원본 필드

Field	Description
request_count	LLM API 호출 건수
status_code	HTTP 응답 상태 코드 (2xx, 4xx, 5xx)
latency	요청 시작 ~ 응답 완료 전체 시간 (ms)
ttft	Time To First Token. 요청 시작 ~ 첫 토큰 수신 (ms)
tpot	Time Per Output Token. 출력 토큰 간 평균 간격 (ms)
input_tokens	전체 입력 토큰 수 (cached 포함)
output_tokens	출력 토큰 수
cached_tokens	캐시에서 가져온 입력 토큰 수 (input_tokens 의 부분집합)
input_tokens_cost	전체 입력 토큰 비용 ($). cached_tokens_cost 포함
output_tokens_cost	출력 토큰 비용 ($)
cached_tokens_cost	캐시 토큰 비용 ($)
error_type	에러 유형 (api_error , program_error)

위젯 그룹 요약

Group	Widgets	Description
LLM API 요청	2	요청량과 상태 코드 모니터링

Group	Widgets	Description
LLM API 응답 성능	8	Latency 전반의 추이와 분포 분석
TTFT	7	첫 토큰 수신 시간 분석
TPOT	7	토큰 생성 속도 분석
토큰	9	토큰 사용 패턴과 캐시 효율 분석
비용	4	LLM API 비용 모니터링
에러	2	에러 유형별 현황 분석
라이브 관측	2	실행 중인 LLM 호출과 Trajectory 현황 파악
합계	**41**	

용어

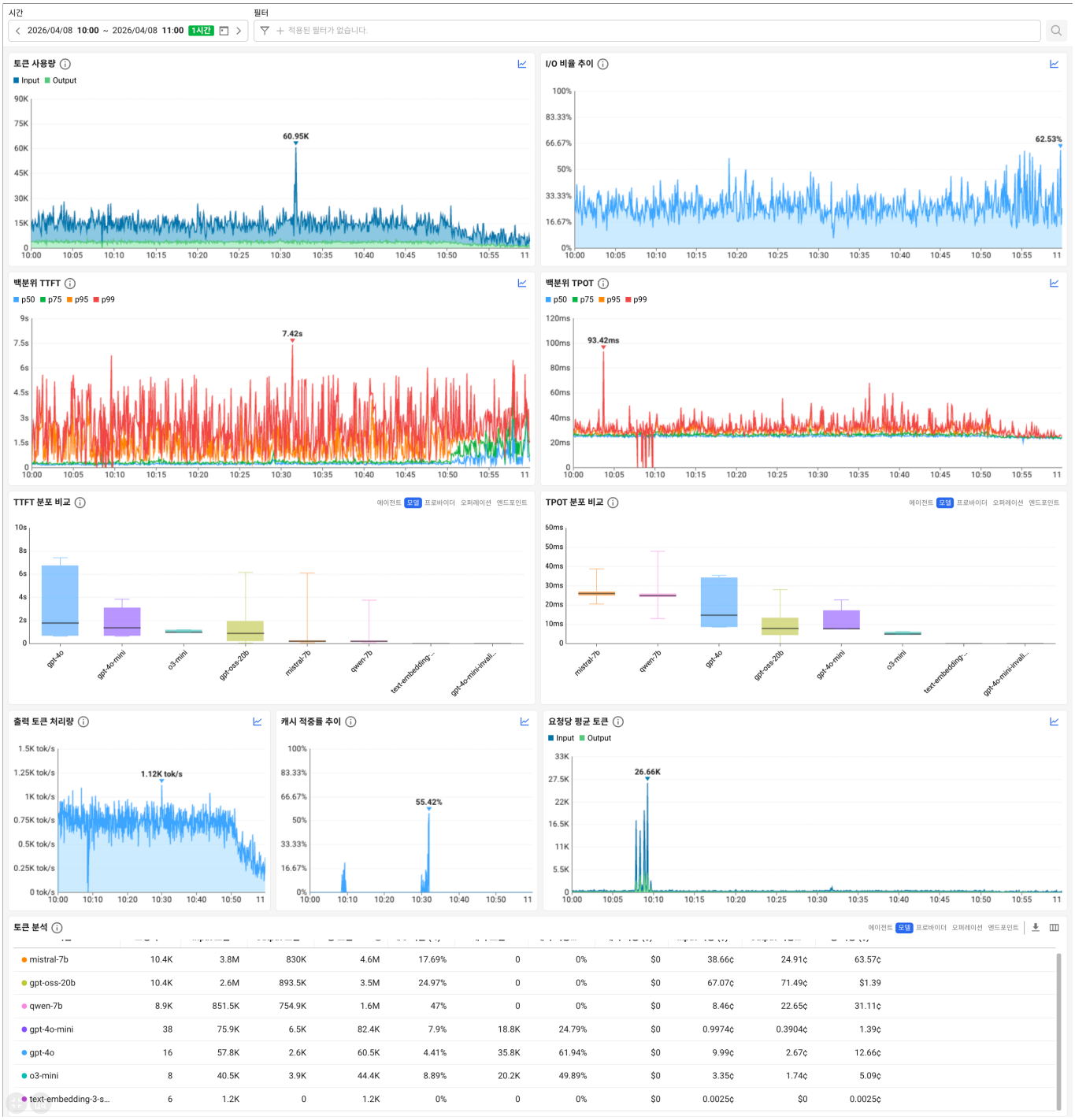
Term	Description
`Latency`	LLM API 요청을 보낸 시점부터 응답이 완전히 완료되기까지의 전체 소요 시간 (ms)
`TTFT`	Time To First Token. 요청을 보낸 후 첫 번째 응답 토큰이 도착하기까지의 시간 (ms)
`TPOT`	Time Per Output Token. 출력 토큰이 하나 생성되는 데 걸리는 평균 시간 (ms)
`Output TPS`	Tokens Per Second. 초당 생성되는 출력 토큰 수 (tokens/s)
`백분위수`	데이터를 크기 순서로 정렬했을 때 해당 백분율 아래에 있는 값. 예: p95 = 전체 데이터의 95%가 이 값 이하
`캐시 적중률`	전체 입력 토큰 중 캐시에서 가져온 토큰의 비율 (%). 값이 클수록 비용 절감 효과가 큼
`태그 필터`	`Agent`, `Model`, `Provider`, `Operation` 기준으로 데이터를 필터링하는 기능
`프리셋`	위젯 구성(종류, 위치, 크기)을 저장한 설정. 사용자 정의 프리셋 생성 가능

Term	Description
`Trajectory`	하나의 LLM 작업 흐름에 참여한 트랜잭션을 `mtid` 기준으로 묶은 단위

LLM 토큰 추이 분석

토큰 추이는 LLM API의 토큰 사용 패턴과 처리 효율을 시계열 기반으로 분석하는 메뉴입니다. 대시보드가 현재 상태 모니터링에 초점을 맞춘다면, 이 페이지는 **구간별 추이 비교**와 **모델/에이전트 간 상세 분석**에 활용됩니다.

상단 옵션바에서 시간 범위, 필터, 검색 조건을 설정한 뒤 위젯 데이터를 조회합니다. 차트 드래그로 구간을 좁히거나, 차트/박스플롯 클릭으로 프롬프트 로그 페이지로 이동할 수 있습니다.



mistral-7b	10.4K	3.8M	830K	4.6M	17.69%	0	0%	$0	38.66¢	24.91¢	63.57¢
gpt-oss-20b	10.4K	2.6M	893.5K	3.5M	24.97%	0	0%	$0	67.07¢	71.49¢	$1.39
qwen-7b	8.9K	851.5K	754.9K	1.6M	47%	0	0%	$0	8.46¢	22.65¢	31.11¢
gpt-4o-mini	38	75.9K	6.5K	82.4K	7.9%	18.8K	24.79%	$0	0.9974¢	0.3904¢	1.39¢
gpt-4o	16	57.8K	2.6K	60.5K	4.41%	35.8K	61.94%	$0	9.99¢	2.67¢	12.66¢
o3-mini	8	40.5K	3.9K	44.4K	8.89%	20.2K	49.89%	$0	3.35¢	1.74¢	5.09¢
text-embedding-3-s...	6	1.2K	0	1.2K	0%	0	0%	$0	0.0025¢	$0	0.0025¢

토큰 사용량

Property	Value
차트 유형	라인 차트 (Input, Output 2개 시리즈)
필터	태그 필터
액션	개별로 보기, 병합 보기, 상세 보기

시간대별 Input(입력) 토큰 수와 Output(출력) 토큰 수의 추이를 표시합니다. Input 토큰은 프롬프트에 포함된 토큰, Output 토큰은 모델이 생성한 응답 토큰입니다.

- 특정 시간대에 토큰 사용량이 급증하면 트래픽 변화, 프롬프트 변경, 또는 비정상 요청 유입을 점검하세요.
- Input과 Output의 비중 변화를 통해 워크로드 특성(입력 위주 vs 생성 위주)의 변화를 파악할 수 있습니다.
- **개별로 보기**로 어떤 모델이나 에이전트에 토큰 소비가 집중되는지 확인할 수 있습니다.

I/O 비율 추이

Property	Value
차트 유형	라인 차트 (%, 영역 채움)
필터	태그 필터
액션	개별로 보기, 병합 보기, 상세 보기

I/O 비율은 전체 토큰(Input + Output) 중 Output 토큰이 차지하는 비율(%)입니다. 50% 기준선(점선)이 함께 표시됩니다.

- **50% 이상**: 응답 생성 비중이 높은 워크로드입니다. 요약, 코드 생성 등 출력이 긴 작업이 많습니다.
- **50% 이하**: 입력 비중이 높은 워크로드입니다. 분류, 임베딩, RAG 검색 등 프롬프트가 긴 작업이 많습니다.
- 비율이 갑자기 변하면 워크로드 패턴이나 프롬프트 구조가 변경되었을 가능성이 있으므로 점검하세요.

> I/O 비율 = output_tokens 합계 / (input_tokens 합계 + output_tokens 합계) x 100

TTFT 백분위

Property	Value
차트 유형	라인 차트 (p50, p75, p95, p99 4개 시리즈)
필터	태그 필터
액션	개별로 보기, 병합 보기, 상세 보기, 백분위 선택

TTFT(Time To First Token)의 백분위수 추이를 표시합니다. 토큰 사용량이 급증하는 구간에서 TTFT도 함께 상승하면, 토큰 규모 증가가 초기 응답 지연에 영향을 미치고 있다는 의미입니다.

- p50은 안정적인데 p99만 급등하면 간헐적으로 큐 대기나 프로바이더 지연이 발생하고 있다는 신호입니다.
- 이상 구간을 발견하면 TTFT 분포 비교에서 어떤 모델이 원인인지 확인할 수 있습니다.

TPOT 백분위

Property	Value
차트 유형	라인 차트 (p50, p75, p95, p99 4개 시리즈)
필터	태그 필터
액션	개별로 보기, 병합 보기, 상세 보기, 백분위 선택

TPOT(Time Per Output Token)의 백분위수 추이를 표시합니다. TTFT 백분위와 함께 보면 "첫 토큰도 느리고 생성도 느린 구간"과 "첫 토큰은 빠르지만 생성이 느린 구간"을 구분할 수 있습니다.

- p99가 급등하면 간헐적으로 토큰 생성이 멈추거나 크게 느려지는 요청이 존재한다는 의미입니다.

- 이상 구간을 발견하면 TPOT 분포 비교에서 어떤 모델이 원인인지 확인할 수 있습니다.

TTFT 분포 비교

Property	Value
차트 유형	박스플롯 (x축: 모델)
필터	태그 필터 (상시 노출)
액션	상세 보기

태그 기준(모델 등)별로 TTFT의 분포를 박스플롯으로 비교합니다. 박스 영역은 p25~~p75(중간 50% 범위), 중앙 선은 중앙값, 위스커는 최솟값~~최댓값을 나타냅니다.

- 중앙값이 낮고 박스가 좁은 모델이 가장 안정적으로 빠른 초기 응답을 제공합니다.
- 위스커가 길게 늘어진 모델은 간헐적으로 극단적인 초기 지연이 발생합니다.
- 모델을 클릭하면 해당 모델의 프롬프트 로그로 이동합니다.

TPOT 분포 비교

Property	Value
차트 유형	박스플롯 (x축: 모델)
필터	태그 필터 (상시 노출)
액션	상세 보기

태그 기준(모델 등)별로 TPOT의 분포를 박스플롯으로 비교합니다.

- 중앙값이 낮고 박스가 좁은 모델이 가장 안정적인 스트리밍 성능을 제공합니다.
- 위스커가 길게 늘어진 모델은 간헐적으로 토큰 생성이 크게 느려지는 현상이 발생합니다.

출력 토큰 처리량

Property	Value
차트 유형	라인 차트
필터	태그 필터
액션	개별로 보기, 병합 보기, 상세 보기

초당 생성되는 출력 토큰 수(Tokens Per Second)를 나타내는 처리량 지표입니다.

- 처리량이 떨어지는 구간은 모델 응답 병목 또는 Rate Limit 영향을 점검하세요.
- 토큰 사용량과 함께 보면 "토큰은 많이 사용하는데 처리 속도가 느린 구간"을 식별할 수 있습니다.
- **개별로 보기**로 모델별 처리량을 비교할 수 있습니다.

캐시 적중률 추이

Property	Value
차트 유형	라인 차트 (%, 영역 채움)
필터	태그 필터
액션	개별로 보기, 병합 보기, 상세 보기

전체 입력 토큰 중 캐시에서 가져온 토큰의 비율(%)을 시간대별로 표시합니다.

- 적중률이 높을수록 프롬프트 캐싱이 효과적으로 동작하고 있으며, 비용 절감 효과가 큽니다.
- 적중률이 갑자기 하락하면 프롬프트 내용 변경, 캐시 TTL 만료, 또는 새로운 유형의 요청 유입을 점검하세요.
- **개별로 보기**로 모델별 캐시 효율을 비교하여, 어떤 모델에서 캐싱이 가장 효과적인지 확인할 수 있습니다.

> ⓘ 캐시 적중률 = (cached_tokens 합계 / input_tokens 합계) x 100

요청당 평균 토큰

Property	Value
차트 유형	라인 차트 (Input, Output 2개 시리즈, 영역 채움)
필터	태그 필터
액션	개별로 보기, 병합 보기, 상세 보기

요청 1건당 사용된 평균 Input/Output 토큰 수의 시간대별 추이를 표시합니다.

- 요청당 Input 토큰이 급증하면 프롬프트 길이가 길어졌거나 긴 컨텍스트가 포함되기 시작했을 수 있습니다.
- 요청당 Output 토큰이 급증하면 응답 길이가 늘어난 것으로, 비용 증가에 직접 영향을 줍니다.
- 토큰 사용량이 증가할 때 이 위젯의 수치도 함께 증가하면 요청당 크기가 커진 것이고, 토큰 사용량만 증가하면 단순히 요청 수가 늘어난 것입니다.

토큰 분석

Property	Value
차트 유형	테이블
필터	Agent / Model / Provider 탭 전환
액션	상세 보기, CSV 다운로드, 컬럼 설정

태그 기준(Agent, Model, Provider)별로 토큰 사용량, 캐시, 비용 핵심 지표를 테이블로 정리합니다.

Column	Description
Name	태그 값 (에이전트명, 모델명, 프로바이더명)
Requests	요청 건수
Total Input	입력 토큰 총량
Total Output	출력 토큰 총량
Total Tokens	전체 토큰 총량
I/O Ratio (%)	출력 토큰 비율
Cached Tokens	캐시 토큰 총량
Cache Hit (%)	캐시 적중률
Cached Cost ($)	캐시 토큰 비용
Input Cost ($)	입력 비용
Output Cost ($)	출력 비용
Total Cost ($)	전체 비용

- `Total Tokens`가 높은 항목은 전체 비용에서 가장 큰 비중을 차지하므로 최적화 우선 대상입니다.
- `I/O Ratio`가 높은 항목은 출력 위주 워크로드로 Output 토큰 단가의 영향을 크게 받습니다.
- `Cache Hit`가 낮은 항목은 캐싱 전략을 점검하여 비용 절감 여지를 확인하세요.
- `Total Cost`를 기준으로 정렬하면 비용 최적화 우선순위를 빠르게 파악할 수 있습니다.
- 오른쪽 상단의 다운로드 버튼으로 현재 데이터를 CSV로 내보낼 수 있습니다.
- 컬럼 설정으로 표시할 컬럼을 선택할 수 있습니다.

참고

대시보드와의 차이

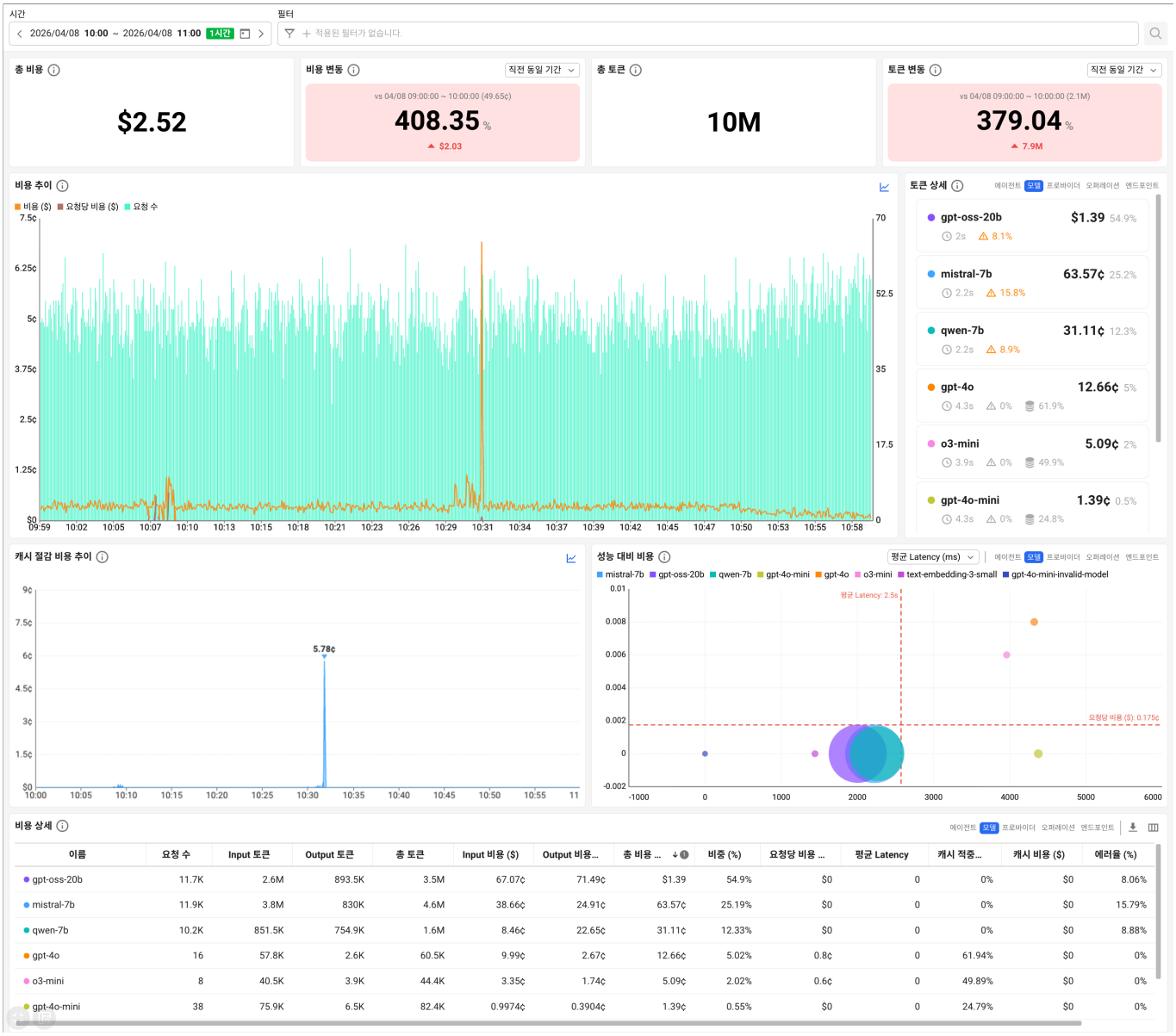
항목	LLM 대시보드	토큰 추이
시간 모드	실시간(Live) + 과거 시점	과거 구간(Range) 전용
주요 목적	현재 상태 모니터링	구간별 추이 분석
필터	에이전트 선택	복합 조건 필터 (Agent, Model, Provider, Operation, URL)
프롬프트 로그 연동	없음	차트 클릭 시 프롬프트 로그 이동
데이터 내보내기	없음	CSV 다운로드 지원

LLM 비용 분석

비용 분석은 LLM API 사용 비용을 다각도로 분석하는 메뉴입니다. 비용 현황 파악, 모델별 비용 비교, 성능 대비 비용 효율 분석, 비용 최적화 대상 선정에 활용할 수 있습니다.

화면은 위에서 아래로 **요약 → 추이·상세 → 캐시·성능 효율 → 테이블** 순서로 구성되어, 현황 파악부터 최적화 대상 선정까지 자연스럽게 따라갈 수 있습니다.

상단 옵션바에서 시간 범위, 필터, 검색 조건을 설정한 뒤 위젯 데이터를 조회합니다. 요약 카드에서 비교 기간(**직전 동일 기간** 또는 **전주 동일 기간**)을 선택할 수 있으며, 차트 드래그로 구간을 좁힐 수 있습니다.



요약 카드

선택한 시간 범위의 핵심 지표를 한눈에 보여주는 카드입니다.

총 비용

선택한 시간 범위 내 발생한 총 비용($)을 표시합니다. Input 비용과 Output 비용을 합산한 금액입니다.

- 비용 분석의 출발점으로, 전체 비용 규모를 한눈에 파악할 수 있습니다.
- 비용 변동과 함께 보면 이전 기간 대비 비용이 어떻게 변했는지 확인할 수 있습니다.

비용 변동

이전 기간 대비 비용 변동률(%)과 변동 금액($)을 표시합니다.

- **양수(▲ 빨간색)**: 비용이 증가했습니다. 요청량 증가, 비싼 모델 사용 비중 증가, 프롬프트 크기 변화 등을 점검하세요.
- **음수(▼ 파란색)**: 비용이 감소했습니다. 최적화 효과나 트래픽 감소를 의미할 수 있습니다.

총 토큰

선택한 시간 범위 내 사용된 총 토큰 수(Input + Output)를 표시합니다.

- 총 비용과 함께 보면 비용 증가가 토큰 사용량 때문인지, 비싼 모델 사용 때문인지 구분할 수 있습니다.
- 토큰은 늘었는데 비용은 비슷하다면 저렴한 모델로의 전환이 잘 이루어지고 있다는 의미입니다.
- 클릭하면 토큰 추이 페이지로 이동합니다.

토큰 변동

이전 기간 대비 토큰 사용량 변동률(%)과 변동 수를 표시합니다.

- 비용 변동과 함께 비교하면 비용 변동의 원인을 파악할 수 있습니다.
- **토큰 변동률 > 비용 변동률**: 저렴한 모델 비중이 늘어난 것입니다.
- **토큰 변동률 < 비용 변동률**: 비싼 모델 비중이 늘어난 것입니다.
- 클릭하면 비교 기간 범위로 토큰 추이 페이지로 이동합니다.

비용 추이

Property	Value
차트 유형	바 차트(요청 수) + 라인 차트(비용, 요청당 비용) 복합
필터	태그 필터
액션	개별로 보기, 병합 보기, 상세 보기

시간대별 요청 수, 비용, 요청당 비용의 추이를 함께 표시합니다. 바 차트는 요청 수, 라인은 비용($)과 요청당 비용($)입니다.

- 요청 수와 비용이 함께 움직이면 사용량에 비례하는 정상적인 비용 증가입니다.
- 요청 수는 변화 없는데 비용만 상승하면 요청당 비용이 비싸진 것이므로, 비싼 모델 비중 증가나 프롬프트 크기 변화를 점검하세요.
- 요청당 비용 라인이 급등하면 모델 전환이나 프로바이더 가격 변경을 의심할 수 있습니다.

토큰 상세

Property	Value
차트 유형	카드 리스트 (모델별 카드)
필터	Model / Provider / Operation / Agent 탭 전환
액션	상세 보기

태그 기준(Model, Provider, Operation, Agent)별로 비용, 비중(%), 평균 Latency, 에러율을 카드 형태로 정리합니다.

각 카드에는 다음 정보가 표시됩니다.

Property	Description
모델명/에이전트명	태그 값
비용	해당 그룹 총 비용($)
비중	해당 그룹 비용 / 전체 비용 x 100(%)
평균 Latency	해당 그룹 Latency 평균
에러율	해당 그룹 에러 건수 / 전체 요청 건수 x 100(%)
캐시 적중률	해당 그룹 cached_tokens / input_tokens x 100(%)

- 비용이 높은 항목이 상단에 정렬되어, 비용 기여도가 큰 모델/에이전트를 바로 확인할 수 있습니다.
- 에러율이 높은 항목은 실패한 요청에도 비용이 발생하고 있으므로, 에러를 줄이면 비용 절감에 직접적인 효과가 있습니다.

캐시 절감 비용 추이

Property	Value
차트 유형	라인 차트($)
필터	태그 필터
액션	개별로 보기, 병합 보기, 상세 보기

프롬프트 캐싱을 통해 절약된 금액($)의 시간대별 추이를 표시합니다. 캐시된 토큰은 일반 입력 토큰보다 낮은 단가가 적용되며, 이 차이가 절감 비용으로 산출됩니다.

- 절감 비용이 꾸준히 발생하면 캐싱 전략이 효과적으로 동작하고 있다는 의미입니다.
- 절감 비용이 감소하면 캐시 적중률 하락 여부를 점검하세요.
- 총 비용 대비 절감 비용의 비율로 캐싱의 전체 비용 절감 기여도를 파악할 수 있습니다.

성능 대비 비용

Property	Value
차트 유형	버블 차트 (x축: 성능 지표, y축: 요청당 비용, 버블 크기: 요청 수)
필터	Model / Provider / Operation / Agent 탭 전환
액션	상세 보기, x축 지표 선택

모델/에이전트별 성능과 비용(요청당 비용)의 관계를 버블 차트로 표시합니다. 버블 크기는 요청 수를 나타내며, 점선으로 전체 평균 기준선이 표시됩니다.

x축 선택 가능 지표

지표	설명
Avg Latency (ms)	평균 응답 완료 시간
Avg TTFT (ms)	평균 첫 토큰 수신 시간
Avg TPOT (ms)	평균 토큰 생성 간격
p95 Latency (ms)	95번째 백분위 응답 시간
p95 TTFT (ms)	95번째 백분위 첫 토큰 시간
p95 TPOT (ms)	95번째 백분위 토큰 생성 간격

사분면 해석

위치	의미	권장 액션
좌측 하단 (빠르고 저렴)	가장 비용 효율이 높은 영역	이 모델의 사용 비중을 유지하거나 확대

위치	의미	권장 액션
우측 상단 (느리고 비쌈)	비용 최적화가 가장 시급한 영역	대체 모델 검토 또는 프롬프트 최적화
좌측 상단 (빠르지만 비쌈)	성능은 좋지만 비용이 높음	해당 성능이 반드시 필요한 워크로드인지 검토
우측 하단 (느리지만 저렴)	비용은 낮지만 성능이 부족	성능 요구사항이 낮은 워크로드에 적합

- 버블이 큰 항목일수록 전체 비용에서 차지하는 비중이 크므로 최적화 시 효과가 큽니다.

비용 상세

Property	Value
차트 유형	테이블
필터	Model / Provider / Operation / Agent 탭 전환
액션	상세 보기, CSV 다운로드, 컬럼 설정

태그 기준(Model, Provider, Operation, Agent)별로 비용 관련 핵심 지표를 테이블로 정리합니다.

Column	Description
`Name`	태그 값 (모델명, 에이전트명 등)
`Requests`	요청 건수
`Input Tokens`	입력 토큰 수
`Output Tokens`	출력 토큰 수
`Total Tokens`	전체 토큰 수
`Input Cost ($)`	입력 비용

Column	Description
Output Cost ($)	출력 비용
Total Cost ($)	전체 비용
Share (%)	비용 점유율 (해당 그룹 비용 / 전체 비용)
Cost/Req ($)	요청당 비용
Avg Latency	평균 응답시간
Cache Hit (%)	캐시 적중률
Cached Cost ($)	캐시 토큰 비용
Error Rate (%)	에러율
Error Cost ($)	에러로 인해 발생한 비용

- `Total Cost ($)`가 높은 항목을 중심으로 비용 최적화 대상을 선정하세요.
- `Cost/Req ($)`가 높은 항목은 요청당 비용이 비싼 모델이므로, 대체 모델 검토나 프롬프트 최적화를 고려하세요.
- `Error Rate (%)`가 높은 항목은 실패한 요청에도 비용이 발생하고 있으므로, 에러 원인을 해결하면 비용 절감에 직접적인 효과가 있습니다.
- `Error Cost ($)`를 통해 에러로 낭비되는 금액을 정확히 확인할 수 있습니다.
- 오른쪽 상단의 다운로드 버튼으로 현재 데이터를 CSV로 내보낼 수 있습니다.
- 컬럼 설정으로 표시할 컬럼을 선택할 수 있습니다.

참고

비교 기간 계산 방식

비교 모드	계산
직전 동일 기간	현재 조회 기간과 동일한 길이만큼 이전 시점의 데이터와 비교합니다. 예: 오늘 09:00 ~ 12:00 조회 시 06:00 ~ 09:00과 비교
전주 동일 기간	현재 조회 기간에서 정확히 7일 전의 동일 시간대 데이터와 비교합니다. 예: 4/6 09:00 ~ 12:00 조회 시 3/30 09:00 ~ 12:00과 비교

모델별 토큰 가격 사용자 설정

모델별 토큰 요율을 직접 지정합니다. 지정한 값은 에이전트의 자동 산출(`genai-prices` 패키지)보다 **우선** 적용되므로, 자체 호스팅·파인튜닝·프록시 경유 모델처럼 공개 요율표에 없는 모델의 비용을 산출하거나, 계약 단가가 정가와 다른 경우 실제 단가로 교정할 수 있습니다.

whatap.conf

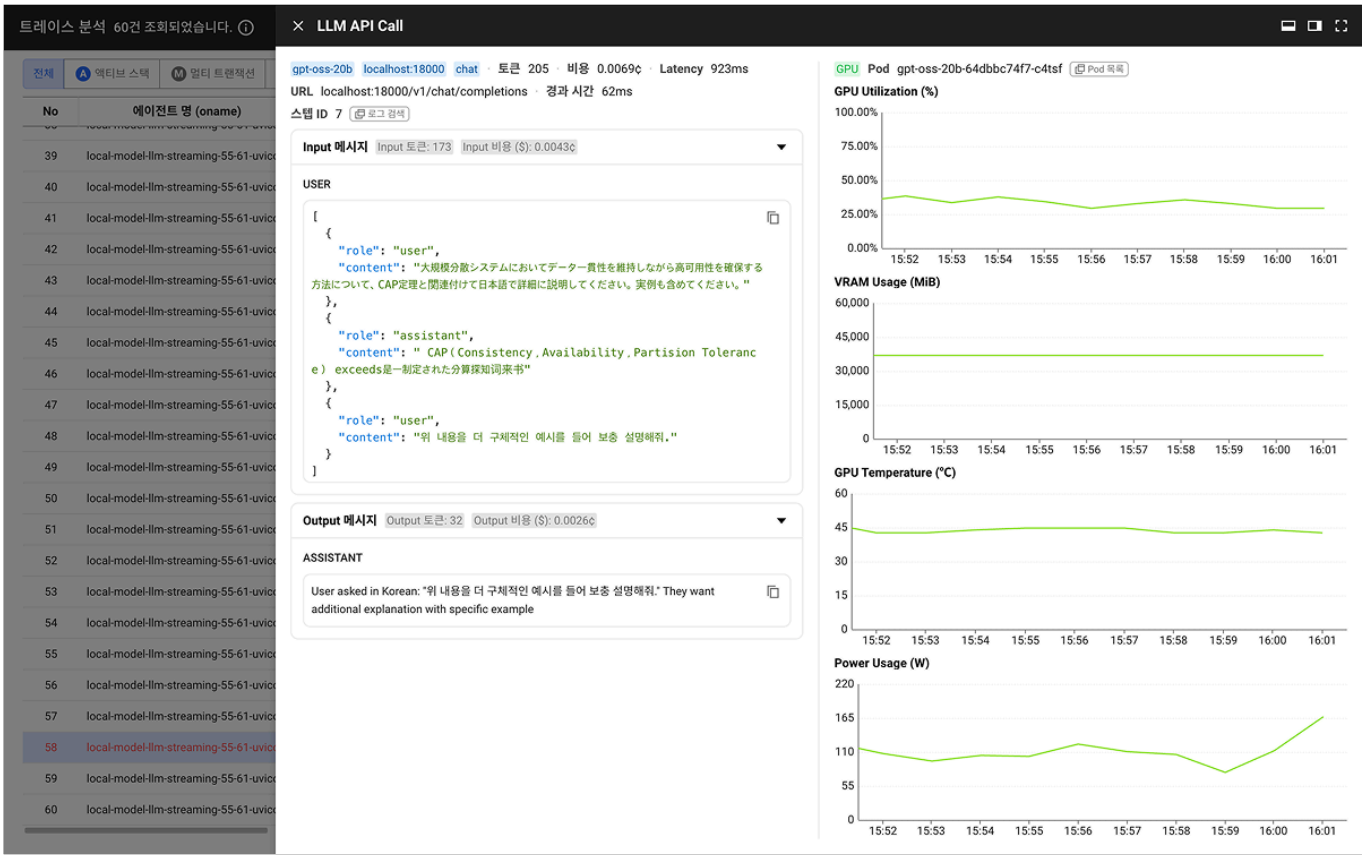
```
llm_model_pricing=model|input_per_1m|output_per_1m|cached_per_1m
# llm_model_pricing=gpt-4o|2.50|10.00|1.25,gpt-4o-mini|0.15|0.60|0.075,my-custom-model|1.00|5.00
```

- 모델 간 구분은 콤마(`,`), 필드 간 구분은 파이프(`|`)를 사용합니다.
- 4번째 필드는 생략(`model|1.00|5.00`) 또는 공백(`model|1.00|5.00|`) 모두 가능합니다.
- 설정 변경은 즉시 반영되며 재시작이 필요하지 않습니다.

LLM API 분석 (트랜잭션 연계)

AI Agent Observability는 WhaTap APM과 통합되어 LLM API 호출을 애플리케이션 트랜잭션의 일부로 추적합니다. 트랜잭션 프로파일에서 LLM HTTPC 스텝을 클릭하면 **LLM API 상세 Drawer**가 열리며, 프롬프트 입력/출력, 토큰 사용량, 비용, 에러 정보는 물론 GPU 인프라 상관관계까지 **LLM API 분석**에서 확인할 수 있습니다.

> ⓘ 프롬프트 입력/출력 조회에는 **로그 읽기 권한**이 필요합니다.



진입 방법

1. LLM 대시보드의 **히트맵** 또는 **트랜잭션 검색** 등에서 트랜잭션을 선택합니다.

2. 트랜잭션 프로파일 화면에서 LLM HTTPC 스텝(LLM 프로바이더 URL로 호출된 HTTP 외부 호출)을 클릭합니다.
3. 우측에서 **LLM API 상세 Drawer**가 열립니다.

요약 및 프롬프트

LLM 호출의 핵심 정보를 한눈에 보여주고, 프롬프트 입력/출력 원본을 확인할 수 있는 영역입니다.

Identity 태그

호출 대상 모델과 프로바이더 정보를 태그 형태로 표시합니다.

Tag	Description	Example
Model	사용된 LLM 모델명	gpt-4o, claude-sonnet-4-20250514
Provider	LLM API 프로바이더	api.openai.com, api.anthropic.com
Operation	호출의 Operation Type	chat, completion

호출이 실패한 경우(success=false) 빨간색 **Error** 태그가 추가로 표시됩니다.

핵심 수치 지표

Identity 태그 옆에 인라인으로 표시됩니다.

Metric	Description
Token	전체 토큰 수 (Input + Output)
Cost	해당 호출의 비용 ($)
Latency	요청 시작~응답 완료 시간 (ms)

HTTP 정보

Property	Description
URL	호출 엔드포인트 (host:port + path)
Elapsed	HTTP 호출 소요 시간 (ms)
Step ID	이 LLM 호출의 고유 식별자. 로그 탐색으로 이동하는 기준 키입니다.

Step ID 옆의 **로그 탐색** 버튼을 클릭하면, 해당 Step ID로 필터링된 로그 탐색기가 새 창에서 열립니다.

에러 정보

호출에 에러가 발생한 경우에만 표시됩니다.

Property	Description
Error Class	에러 클래스명 (예: RateLimitError, TimeoutError)
Error Message	에러 상세 메시지

프롬프트 입력 (Input)

접기/펼치기가 가능한 섹션으로, LLM에 전달된 입력 메시지를 표시합니다.

헤더 뱃지

Badge	Description
Input Tokens	입력 토큰 수
Cached Tokens	캐시에서 가져온 토큰 수 (해당 시에만 표시)
Input Cost ($)	입력 비용

메시지 유형

Label	Description
SYSTEM	시스템 메시지. 모델의 역할과 동작을 정의하는 지시문입니다.
USER	사용자 입력 메시지. 실제 프롬프트 내용입니다.

메시지가 청크(chunk)로 분할되어 수집된 경우, 자동으로 올바른 순서로 조립하여 완전한 텍스트로 표시합니다.

모델 응답 (Output)

접기/펼치기가 가능한 섹션으로, 모델이 생성한 응답을 표시합니다.

헤더 뱃지

Badge	Description
`Output Tokens`	출력 토큰 수
`Reasoning Tokens`	추론(Reasoning) 토큰 수 (해당 시에만 표시)
`Output Cost ($)`	출력 비용

메시지 유형

Label	Description
ASSISTANT	모델의 텍스트 응답입니다.
TOOL CALL	모델이 요청한 도구 호출(Function Calling) 내용입니다.
TOOL RESULT	도구 호출의 실행 결과입니다.

GPU 상관관계

멀티 트랜잭션(mtid가 존재하는 경우)일 때 Drawer 우측에 표시됩니다. LLM 호출을 처리한 Pod의 GPU 인프라 상태를 실시간 차트로 보여주어, 모델 응답 지연이 GPU 리소스 부족 때문인지 판단할 수 있습니다.

GPU 정보 헤더

Property	Description
`Pod`	LLM 추론을 수행한 Kubernetes Pod 이름
Pod 상세 버튼	클릭 시 Kubernetes 모니터링의 Pod 상세 페이지가 새 창에서 열립니다.

GPU 차트

LLM 호출 시점 전후 5분 범위의 GPU 메트릭을 라인 차트로 표시합니다.

Chart	Metric	Unit	Description
GPU Utilization	`DCGM_FI_DEV_WEIGHTED_GPU_UTIL`	%	GPU 연산 자원의 사용률. 100%에 가까우면 GPU가 포화 상태이며, LLM 응답 지연의 원인일 수 있습니다.
VRAM Usage	`DCGM_FI_DEV_FB_USED`	MiB	GPU 메모리(Video RAM) 사용량. 모델 로딩과 추론에 사용되며, 부족하면 OOM이나 스왑이 발생합니다.
GPU Temperature	`DCGM_FI_DEV_GPU_TEMP`	°C	GPU 온도. 과열 시 자동 스로틀링이 발생하여 성능이 저하됩니다.
Power Usage	`DCGM_FI_DEV_POWER_USAGE`	W	GPU 전력 소비. Utilization과 함께 보면 실제 연산 부하를 파악할 수 있습니다.

> ⓘ GPU 상관관계는 다음 조건이 모두 충족될 때 표시됩니다.

- 트랜잭션이 멀티 트랜잭션(mtid 존재)인 경우
- LLM 추론을 수행한 Pod 이름이 식별된 경우
- 연계된 Kubernetes 프로젝트에서 DCGM GPU 메트릭이 수집되고 있는 경우

분석 시나리오

LLM 응답 지연 원인 파악

1. LLM 대시보드의 히트맵에서 응답 시간이 긴 트랜잭션을 클릭합니다.
2. 트랜잭션 프로파일에서 LLM HTTPC 스텝의 **Elapsed** 시간을 확인합니다.
3. LLM API 상세 Drawer를 열어 **Latency** 값과 **GPU Utilization** 차트를 비교합니다.
4. GPU Utilization이 높은 상태에서 Latency가 급증했다면 GPU 리소스 부족이 원인입니다.
5. GPU Utilization이 낮은데 Latency가 높다면 프로바이더 측 지연(큐 대기, Rate Limit 등)을 의심합니다.

에러 발생 LLM 호출 재현

1. 트랜잭션 프로파일에서 에러가 표시된 LLM HTTPC 스텝을 클릭합니다.
2. Drawer의 **Error Class**와 **Error Message**로 에러 유형을 확인합니다.
3. **프롬프트 입력(Input)** 섹션에서 SYSTEM 메시지와 USER 메시지 원본을 확인합니다.
4. 수집된 프롬프트 원본과 모델/파라미터 정보로 동일 호출을 재현하여 문제를 분석합니다.

비용이 높은 LLM 호출 분석

1. 비용 분석 페이지에서 비용이 높은 시간대를 식별합니다.
2. 해당 시간대의 트랜잭션을 검색하여 프로파일을 열고, LLM HTTPC 스텝을 클릭합니다.
3. Drawer에서 **Token** 수와 **Cost**를 확인합니다.
4. **프롬프트 입력** 섹션에서 Input Tokens 뱃지를 확인하고, 프롬프트가 불필요하게 긴지 점검합니다.
5. **Cached Tokens** 뱃지가 0이면 캐싱이 적용되지 않은 호출이므로, 캐싱 대상 프롬프트인지 검토합니다.

참고

데이터 수집 구조

LLM API 상세 Drawer는 두 가지 데이터 소스를 결합합니다.

Data Source	Description	Usage
APM 트레이스	HTTP 호출 정보 (URL, host, port, elapsed, 에러 클래스/메시지), Step ID	트랜잭션 프로파일에서의 호출 식별 및 HTTP 수준 정보 표시
`#LlmCallLog` 로그	모델, 프로바이더, 토큰, 비용, 성공 여부, 메시지 내용	프롬프트 원본 복원 및 LLM 수준 메타데이터 표시

두 데이터는 **Step ID**를 기준으로 연결됩니다.

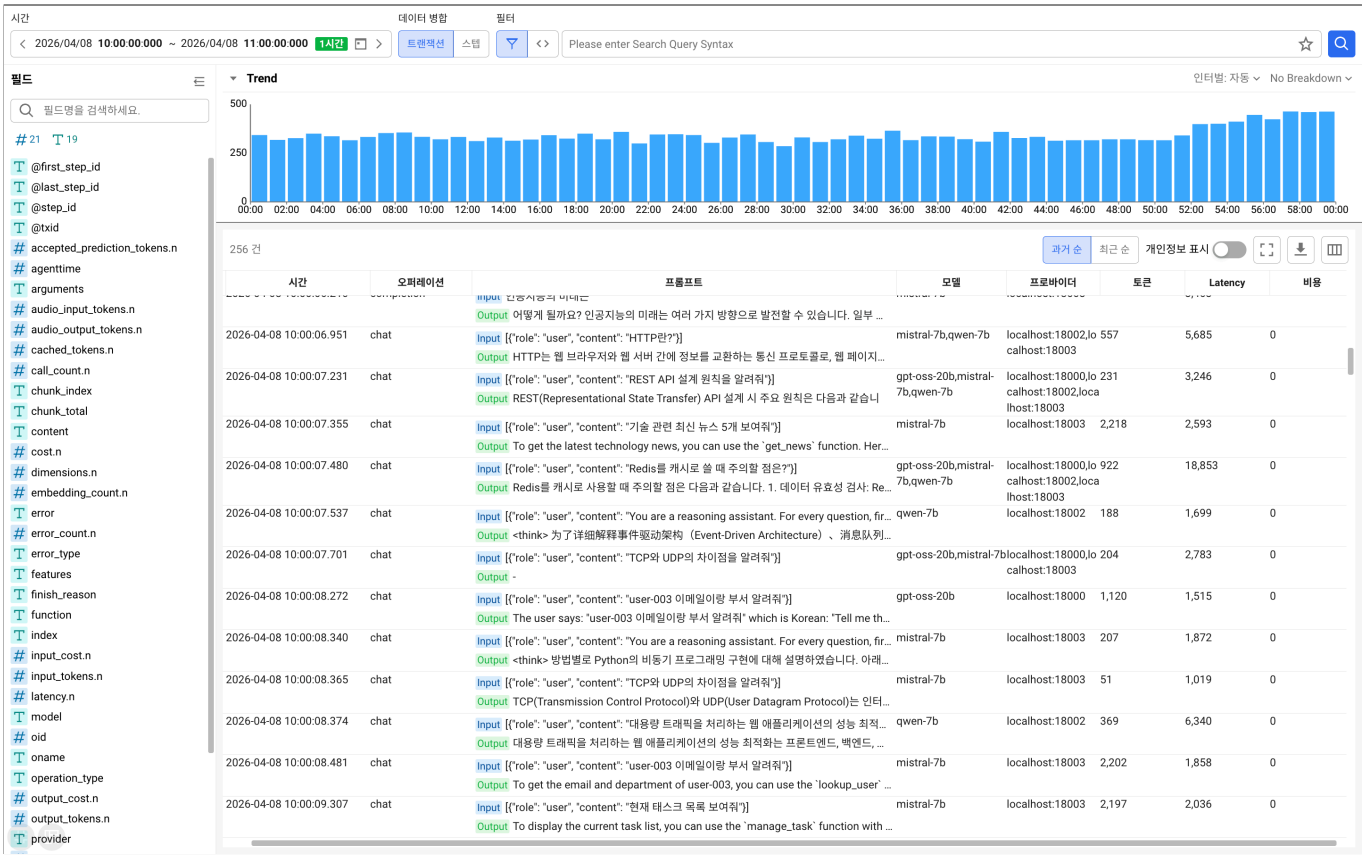
메시지 청크 조립

LLM 프롬프트와 응답은 길이가 길 수 있으므로 서버에서 여러 청크(chunk)로 분할하여 수집됩니다. Drawer는 `chunk_index` 필드를 기준으로 청크를 올바른 순서로 조립하여 완전한 메시지를 복원합니다. 타입당 최대 100개 청크까지 지원합니다.

프롬프트 및 응답 검색

LLM API 호출의 상세 로그를 조회하고 분석하는 메뉴입니다. 각 호출의 입력(프롬프트), 출력(응답), 토큰 사용량, 비용, 성능 지표를 개별 건 단위로 확인할 수 있습니다. 토큰 추이나 비용 분석에서 이상 구간을 발견한 후, 원인을 파악하기 위해 드릴다운하는 용도로 활용됩니다.

로그 읽기 권한이 필요합니다. 권한이 없으면 메뉴에 접근할 수 없습니다.



화면 구성

프롬프트 로그는 **상단 옵션바**, **좌측 필터 패널**, **우측 메인 영역**(트렌드 차트 + 로그 리스트)의 3단 레이아웃으로 구성됩니다.

상단 옵션바

- **시간 범위 선택**: 시작일시와 종료일시를 지정합니다. 빠른 선택 옵션(최근 5분, 10분, 30분, 1시간, 3시간, 6시간, 12시간)을 제공합니다. 초 단위 및 밀리초 단위까지 지정할 수 있습니다.
- **병합 기준 선택**: 로그를 병합하는 기준을 선택합니다.
  - **트랜잭션**: 하나의 트랜잭션 안에서 발생한 모든 LLM 호출을 하나의 행으로 병합합니다. 호출 수(Call Count)와 에러 수(Error Count)가 함께 표시됩니다.
  - **스텝**: 개별 LLM 호출(Step) 단위로 한 행씩 표시합니다. 각 호출의 모델, temperature, TTFT 등 상세 정보를 확인할 수 있습니다.
- **필터**: 검색 쿼리를 입력하여 특정 조건의 로그만 조회할 수 있습니다.

좌측 필터 패널

로그 필드 기반의 태그 필터를 제공합니다. 각 필드의 값 목록이 표시되며, 클릭하여 필터를 적용할 수 있습니다.

트렌드 차트

선택한 시간 범위의 로그 건수를 시간대별 바 차트로 표시합니다. 차트를 접거나 펼쳐 공간을 조절할 수 있으며, 차트와 로그 리스트 사이의 구분선을 드래그하여 비율을 조절할 수 있습니다.

로그 리스트

LLM 호출 로그를 테이블 형태로 표시합니다. 병합 기준(트랜잭션/스텝)에 따라 표시되는 컬럼이 달라집니다.

기본 컬럼

Column	Description
상태	에러 여부를 색상으로 표시합니다. 에러가 발생한 행은 빨간색으로 표시됩니다.
`Time`	로그 발생 시각

Column	Description
Operation	LLM 호출의 Operation Type
Prompt	프롬프트 입력과 응답의 요약. 클릭하면 트랜잭션 프로파일 팝아웃이 열립니다.
Model	사용된 LLM 모델명
Provider	LLM 프로바이더 (예: api.openai.com)
Tokens	전체 토큰 수 (Input + Output)
Latency	요청 시작~응답 완료 시간 (ms)
Cost	해당 호출의 비용 ($)

트랜잭션 모드 전용 컬럼

Column	Description
Call Count	트랜잭션 내 LLM 호출 수
Error Count	트랜잭션 내 에러가 발생한 호출 수

스텝 모드 전용 컬럼

Column	Description
Step	트랜잭션 내 LLM 호출 순서 인덱스
Step ID	개별 LLM 호출의 고유 식별자
temperature	LLM 호출 시 설정된 temperature 파라미터 값
TTFT	Time To First Token (ms). 스트리밍 모드에서만 표시됩니다.

추가 표시 가능 컬럼

컬럼 설정을 통해 추가로 표시할 수 있습니다.

Column	Description
`Input Tokens`	입력 토큰 수
`Output Tokens`	출력 토큰 수
`Cached Tokens`	캐시에서 가져온 토큰 수
`Reasoning Tokens`	추론(Reasoning) 토큰 수
`Input Cost`	입력 비용 ($)
`Output Cost`	출력 비용 ($)
`Stream`	스트리밍 모드 여부 (True/False)
`Features`	활성화된 기능 플래그
`Agent`	에이전트명
`Error`	에러 메시지 (에러 발생 시에만 표시)
`Error Type`	에러 유형 (`api_error`, `program_error` 등)
`TXID`	트랜잭션 ID

로그 상세 보기

로그 리스트에서 **Prompt** 컬럼을 클릭하면 트랜잭션 프로파일 팝아웃이 열립니다. 해당 LLM 호출이 포함된 전체 트랜잭션의 상세 정보를 확인할 수 있습니다.

- **트랜잭션 모드**: 전체 트랜잭션 뷰로 이동합니다.

- **스텝 모드**: 해당 Step이 하이라이트된 상태로 트랜잭션 뷰가 열립니다.

트랜잭션 프로파일에서 LLM HTTPC 스텝을 클릭하면 프롬프트 입력/출력 원본, 토큰/비용 상세, GPU 상관관계까지 확인할 수 있습니다. 자세한 내용은 LLM API 분석 문서를 참조하세요.

병합 기준 상세

트랜잭션 병합

하나의 트랜잭션(@txid 기준) 안에서 발생한 모든 LLM 호출을 하나의 행으로 병합합니다.

- **Prompt 표시**: 첫 번째 Step의 입력(Input)과 마지막 Step의 출력(Output)이 표시됩니다.
- **토큰/비용**: 트랜잭션 내 모든 호출의 합계가 표시됩니다.
- **모델**: 트랜잭션의 대표 모델이 표시됩니다.
- **Call Count / Error Count**: 총 호출 수와 에러 발생 호출 수가 표시됩니다.

스텝 병합

개별 LLM 호출(@step_id 기준) 단위로 한 행씩 표시합니다.

- **Prompt 표시**: 해당 Step의 입력(Input Message)과 출력(Output Message)이 표시됩니다.
- **토큰/비용**: 해당 호출의 개별 값이 표시됩니다.
- **추가 정보**: temperature, TTFT, Stream 여부 등 호출 단위의 상세 정보를 확인할 수 있습니다.

다른 메뉴에서 진입

프롬프트 로그는 다른 분석 메뉴에서 드릴다운할 때 자주 사용됩니다.

진입 경로	설명
토큰 추이 > 시계열 차트 클릭	해당 시점의 시간 범위가 자동 설정되어 프롬프트 로그가 열립니다.
토큰 추이 > 박스플롯 모델 클릭	해당 모델의 필터가 자동 적용되어 프롬프트 로그가 열립니다.

반대로 프롬프트 로그에서 **Prompt** 컬럼을 클릭하면 트랜잭션 프로파일로 이동하여, 트랜잭션 내 모든 스텝(DB, HTTP, LLM 등)을 시간순으로 확인할 수 있습니다.

참고

로그 타입

프롬프트 로그는 내부적으로 `#LlmCallLog` 카테고리의 로그를 조회합니다.

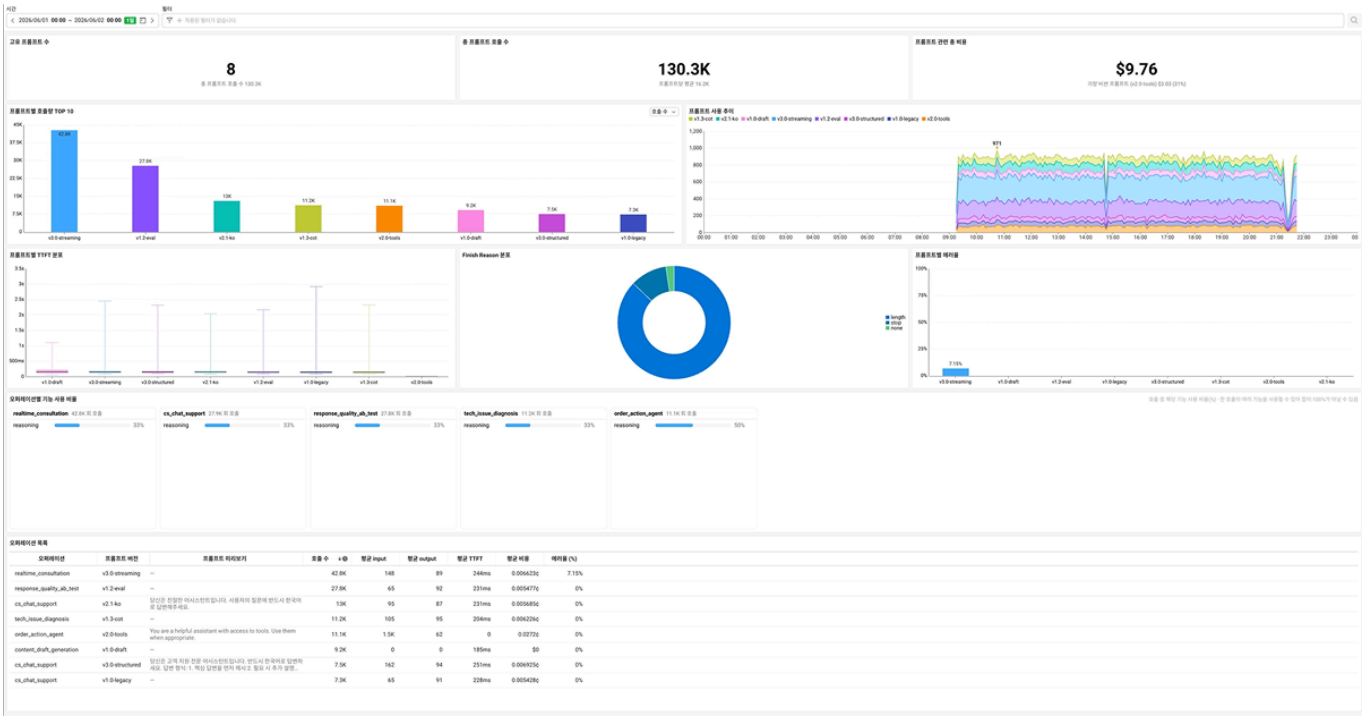
Log Type	Description
`llm_step_status`	개별 LLM 호출의 최종 결과 (모델, 토큰, 비용, 성능 등)
`llm_tx_status`	트랜잭션 단위 집계 결과 (호출 수, 에러 수, 합산 토큰/비용)
`system_message`	시스템 메시지 (프롬프트의 시스템 역할 부분)
`input_message`	입력 메시지 (사용자 프롬프트)
`output_message`	출력 메시지 (모델 응답)
`tool`	도구 호출 (Function calling)
`tool_result`	도구 호출 결과

프롬프트 분석

애플리케이션에서 사용 중인 프롬프트를 식별하고, 프롬프트 단위로 호출량·토큰·지연(TTFT)·비용·에러를 분석하는 메뉴입니다. 어떤 프롬프트가 가장 많이 쓰이는지, 어떤 프롬프트가 느리고 비싸고 불안정한지 파악하여 최적화 대상을 선정하는 데 활용할 수 있습니다.

호출은 **오퍼레이션(operation)** 과 **프롬프트 버전(prompt version)** 의 두 단위로 식별됩니다. 오퍼레이션은 기능 단위의 호출 묶음이고, 프롬프트 버전은 해당 오퍼레이션에서 사용된 시스템 프롬프트의 버전입니다. 하나의 오퍼레이션은 여러 프롬프트 버전을 가질 수 있으며, 차트 대부분은 프롬프트 버전 단위로 집계됩니다.

화면은 위에서 아래로 **요약 카드 → 호출·추이 → 분포·에러 → 기능 사용 → 테이블** 순서로 구성되어, 현황 파악부터 개별 오퍼레이션 점검까지 자연스럽게 따라갈 수 있습니다. 상단 옵션바에서 시간 범위와 필터를 설정한 뒤 위젯 데이터를 조회합니다.



요약 카드

선택한 시간 범위의 핵심 지표를 한눈에 보여주는 카드입니다.

고유 프롬프트 수

선택한 시간 범위 내 1회 이상 호출된 서로 다른 프롬프트의 개수를 표시합니다. 카드 하단에는 해당 프롬프트들의 **총 호출 수**가 함께 표시됩니다.

- 프롬프트 운영 규모를 한눈에 파악하는 출발점입니다.
- 고유 프롬프트 수가 평소보다 급증하면 의도하지 않은 프롬프트 변형이 배포되었을 가능성을 점검하세요.

총 프롬프트 호출 수

선택한 시간 범위 내 발생한 전체 프롬프트 호출 횟수를 표시합니다. 카드 하단에는 **프롬프트당 평균** 호출 수가 함께 표시됩니다. 프롬프트당 평균과 함께 보면 소수의 프롬프트에 호출이 집중되어 있는지, 고르게 분산되어 있는지 가늠할 수 있습니다.

- `프롬프트당 평균 = 총 프롬프트 호출 수 / 고유 프롬프트 수`

프롬프트 관련 총 비용

선택한 시간 범위 LLM API 호출로 발생한 총 비용($)을 표시합니다. 카드 하단에는 **가장 비싼 오퍼레이션**의 버전명과 금액, 전체 대비 비중(%)이 함께 표시됩니다.

- 비중이 높은 단일 프롬프트는 비용 최적화의 우선 대상입니다.
- 특정 프롬프트가 전체 비용의 큰 부분을 차지하면, 해당 프롬프트의 토큰 길이나 모델 선택을 우선 검토하세요.

프롬프트별 호출량 TOP 10

Property	Value
차트 유형	세로 막대 차트
집계 단위	프롬프트 버전
액션	표시 지표 선택 (우측 상단 드롭다운)

호출량이 많은 상위 10개 프롬프트 버전을 막대 길이로 정렬해 보여줍니다. 우측 상단 드롭다운에서 표시 지표(호출 수, 총 토큰 ,

총 비용)를 전환할 수 있습니다.

- 특정 버전에 호출이 집중되어 있다면, 그 버전이 전체 성능·비용에 미치는 영향이 크므로 우선적으로 관리하세요.
- 신규 배포 버전이 상위권에 빠르게 진입하면 트래픽 전환(롤아웃)이 정상적으로 진행되고 있다는 의미입니다.

프롬프트 사용 추이

Property	Value
차트 유형	누적 영역 차트(stacked area)
집계 단위	프롬프트 버전
액션	범례 항목 토글

시간대별로 각 프롬프트 버전의 호출량 추이를 누적 영역으로 표시합니다. 범례의 색상이 각 프롬프트 버전에 대응하며, 영역의 두께가 해당 버전의 호출량을 나타냅니다.

- 전체 영역의 두께 변화로 시간대별 총 호출량의 흐름을 파악할 수 있습니다.
- 특정 시간대에 특정 버전의 비중이 급변하면 트래픽 패턴 변화나 신규 버전 배포의 영향을 점검하세요.
- 새 버전 배포 직후 해당 영역이 빠르게 두꺼워지면 트래픽 전환이 진행 중이라는 의미입니다.

프롬프트별 TTFT 분포

Property	Value
차트 유형	박스플롯 (x축: 프롬프트 버전)
집계 단위	프롬프트 버전
액션	상세 보기

프롬프트 버전별 **TTFT(Time To First Token, 첫 토큰까지의 시간)** 분포를 박스플롯으로 비교합니다. 박스 영역은 p25 ~

p75(중간 50% 범위), 중앙 선은 중앙값, 위스커는 최솟값 ~ 최댓값을 나타냅니다.

- 중앙값이 낮고 박스가 좁은 버전이 가장 안정적으로 빠른 초기 응답을 제공합니다.
- 위스커가 길게 늘어진 버전은 간헐적으로 극단적인 초기 지연이 발생하므로, 입력 길이나 모델 부하를 점검하세요.

Finish Reason 분포

Property	Value
차트 유형	도넛 차트
집계 단위	종료 사유 (length / stop / none)
액션	범례 항목 토글

LLM API가 응답에 반환하는 종료 사유(finish reason) 값을 그대로 집계하여 분포로 표시합니다. 표시되는 값과 명칭은 **프로바이더·모델에 따라 다를 수 있으며**, WhaTap이 별도로 정의하지 않고 API가 내려준 값을 그대로 사용합니다.

- 정상 종료를 나타내는 값(예: `stop`)의 비중이 대부분이라면 응답이 의도대로 마무리되고 있다는 의미입니다.
- 길이 제한으로 인한 중단을 나타내는 값(예: `length`)의 비중이 높다면, 최대 토큰(max_tokens) 설정이 너무 작아 응답이 중간에 잘리고 있지 않은지 점검하세요.
- 종료 사유가 비어 있거나 기록되지 않은 호출은 별도 값(예: `none`)으로 묶여 표시될 수 있습니다.

> ⓘ 표시되는 finish reason 값은 사용하는 LLM 프로바이더·모델의 응답 스펙을 따릅니다. 각 값의 정확한 의미는 해당 프로바이더의 API 문서를 참고하세요.

프롬프트별 에러율

Property	Value
차트 유형	세로 막대 차트 (x축: 프롬프트 버전)

Property	Value
집계 단위	프롬프트 버전
액션	상세 보기

프롬프트 버전별 호출 대비 에러 발생 비율(%)을 표시합니다.

- 에러율이 높은 버전은 잘못된 툴 정의, 스키마 불일치, 토큰 초과 등이 원인일 수 있으므로 오퍼레이션 목록에서 해당 버전의 지표를 함께 확인하세요.
- 실패한 요청에도 비용이 발생하므로, 에러율이 높은 버전을 개선하면 비용 절감에도 직접적인 효과가 있습니다.

`에러율 = 에러 호출 수 / 전체 호출 수 x 100`

오퍼레이션별 기능 사용 비율

Property	Value
차트 유형	카드 그리드 (오퍼레이션별 카드)
집계 단위	오퍼레이션
액션	-

오퍼레이션마다 카드 한 장이 생성되며, 카드 상단에 오퍼레이션 이름과 총 호출 수가 표시됩니다. 카드 안에는 해당 오퍼레이션의 호출에서 각 기능(예: `reasoning`)이 사용된 비율이 막대와 퍼센트로 표시됩니다.

- 한 호출이 여러 기능을 동시에 사용할 수 있어 비율의 합은 100%가 아닐 수 있으며, 사용하지 않은 기능은 표시되지 않습니다.
- 기능 사용 패턴을 통해 오퍼레이션의 동작 방식을 빠르게 이해하고, 의도와 다른 기능 사용을 발견할 수 있습니다.

오퍼레이션 목록

Property	Value
차트 유형	테이블
집계 단위	오퍼레이션 + 프롬프트 버전
액션	정렬

기간 내 식별된 오퍼레이션을 프롬프트 버전 단위로 나열하고, 핵심 지표를 테이블로 정리합니다. 동일 오퍼레이션이라도 프롬프트 버전이 다르면 별도의 행으로 표시됩니다.

Column	Description
`오퍼레이션`	기능 단위의 호출 묶음 이름
`프롬프트 버전`	해당 오퍼레이션에서 사용된 시스템 프롬프트 버전
`프롬프트 미리보기`	시스템 프롬프트 원문의 앞부분 (없으면 `—`)
`호출 수`	기간 내 총 호출 횟수
`평균 input`	호출당 평균 입력 토큰 수
`평균 output`	호출당 평균 출력 토큰 수
`평균 TTFT`	첫 토큰까지의 평균 시간(ms)
`평균 비용`	호출 1건당 평균 비용(¢)
`에러율 (%)`	호출 대비 에러 발생 비율

- 기본 정렬은 호출 수 내림차순이며, 컬럼 헤더를 클릭해 정렬 기준을 변경할 수 있습니다.
- `평균 비용`이 높은 행은 요청당 단가가 비싼 프롬프트이므로, 프롬프트 길이 축소나 모델 검토를 고려하세요.

- `프롬프트 미리보기`로 동일 오퍼레이션의 버전 간 프롬프트 차이를 빠르게 비교할 수 있습니다.

참고

프롬프트 식별 방식

단위	설명
오퍼레이션	기능 단위의 호출 묶음 (예: `cs_chat_support`)
프롬프트 버전	해당 오퍼레이션에서 사용된 시스템 프롬프트의 버전 (예: `v2.1-ko`)

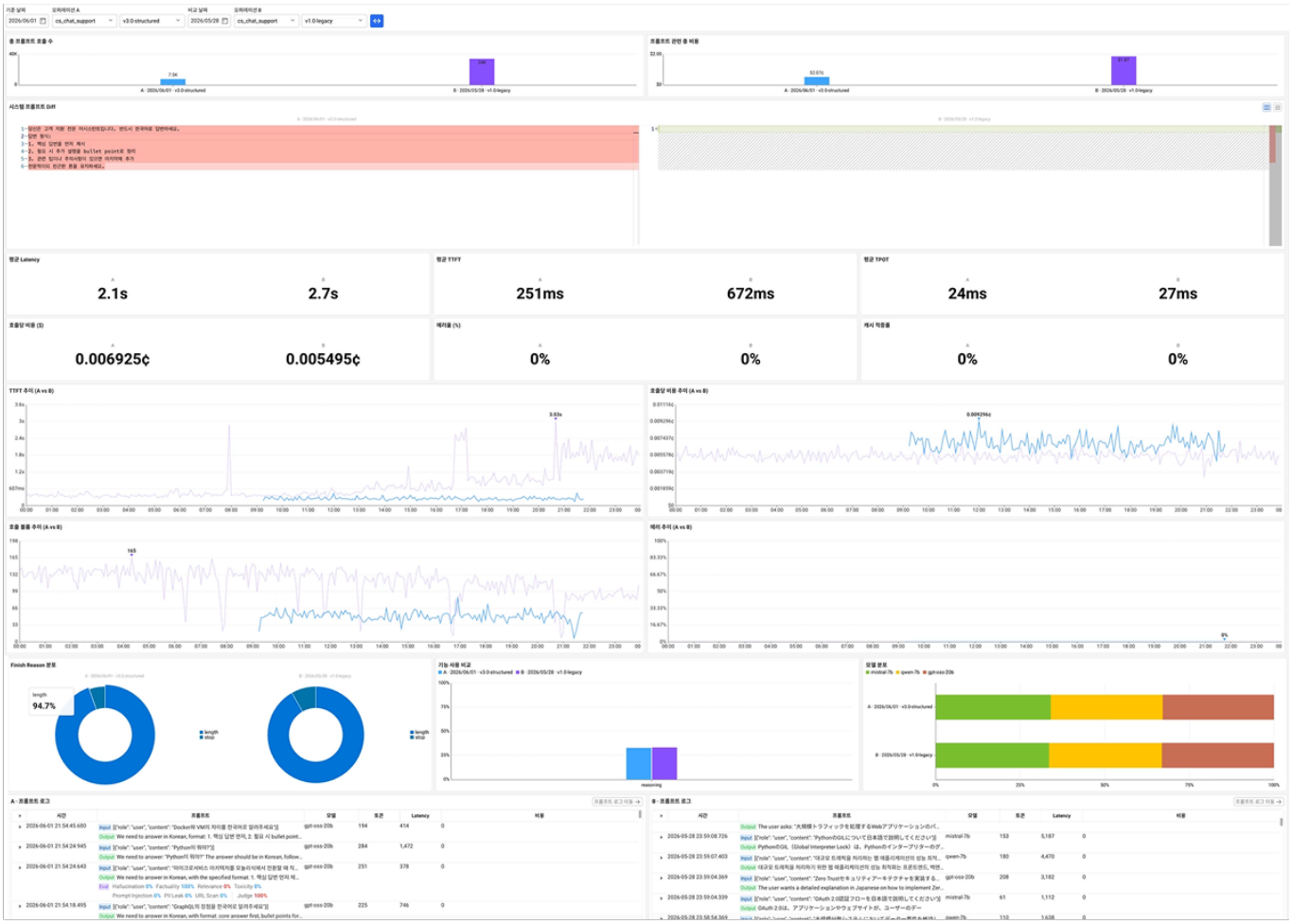
- 하나의 오퍼레이션은 여러 프롬프트 버전을 가질 수 있으며, TOP 10·사용 추이·TTFT 분포·에러율 위젯은 프롬프트 버전 단위로 집계됩니다.
- 프롬프트 버전 간 성능 차이를 정량 비교하려면 **프롬프트 비교** 메뉴를, 응답 품질·안전성 평가가 필요하면 **응답 품질 분석** 메뉴를 함께 활용하세요.

프롬프트 비교

두 프롬프트를 나란히 놓고 호출량·비용·성능·응답 품질을 비교하는 메뉴입니다. 프롬프트를 수정한 뒤 변경이 실제로 성능과 비용을 개선했는지 정량적으로 검증하거나, 서로 다른 오퍼레이션·버전 간의 특성 차이를 분석하는 데 활용할 수 있습니다.

비교 대상은 **오퍼레이션 + 프롬프트 버전 + 날짜**의 조합으로 지정하며, 기준이 되는 쪽을 **A**, 비교 대상을 **B**로 둡니다. 보통 A에 기존 버전, B에 변경 버전을 놓고 차이를 확인합니다.

화면은 위에서 아래로 **요약(호출·비용) → 프롬프트 Diff → 지표 비교 → 추이 비교 → 분포 비교 → 프롬프트 로그** 순서로 구성되어, "무엇이 바뀌었는가"(Diff)부터 "그 변경이 어떤 결과를 냈는가"(지표·추이·로그)까지 따라갈 수 있습니다.



비교 조건 설정

상단 옵션바에서 비교할 두 프롬프트를 지정합니다.

- **기준 날짜 / 오퍼레이션 A / 프롬프트 버전 A**: 기준이 되는 프롬프트(A)와 조회 날짜를 선택합니다.
- **비교 날짜 / 오퍼레이션 B / 프롬프트 버전 B**: 비교 대상 프롬프트(B)와 조회 날짜를 선택합니다.

> A와 B는 서로 다른 날짜를 지정할 수 있어, 같은 프롬프트의 배포 전후를 다른 기간으로 비교할 수도 있습니다. 모든 위젯에서 A는 파란색, B는 보라색으로 일관되게 표시됩니다.

총 프롬프트 호출 수

Property	Value
차트 유형	막대 차트 (A vs B)
비교 대상	A(기준) · B(비교)
액션	마우스 오버 시 값 표시

선택한 두 프롬프트의 총 호출 수를 막대로 나란히 비교합니다. 각 막대에는 `날짜 · 프롬프트 버전` 이 라벨로 표시됩니다.

- 호출 수 차이가 크면 평균 지표나 추이를 해석할 때 표본 규모 차이를 함께 고려하세요.
- 새 버전(B)의 호출 수가 빠르게 늘고 있다면 트래픽 전환(롤아웃)이 진행 중이라는 의미입니다.

프롬프트 관련 총 비용

Property	Value
차트 유형	막대 차트 (A vs B)

Property	Value
비교 대상	A(기준) · B(비교)
액션	마우스 오버 시 값 표시

선택한 두 프롬프트의 총 비용을 막대로 나란히 비교합니다.

- 호출 수가 비슷한데 총 비용 차이가 크면 프롬프트 길이나 모델 단가 차이가 원인일 수 있습니다.
- 호출당 비용은 아래 **지표 비교**의 `호출당 비용` 카드에서 표본 규모와 무관하게 비교할 수 있습니다.

시스템 프롬프트 Diff

Property	Value
차트 유형	텍스트 Diff (좌: A / 우: B)
액션	Side-by-side / Inline 보기 전환

A와 B의 시스템 프롬프트 원문을 줄 단위로 비교합니다. 추가된 줄과 삭제된 줄이 색상으로 구분되며, 우측 상단 아이콘으로 **Side-by-side**(좌우 분할)와 **Inline**(한 흐름) 보기를 전환할 수 있습니다.

- 이 위젯은 "무엇이 바뀌었는가"를, 아래의 지표·추이 위젯은 "그 변경이 어떤 결과를 냈는가"를 보여주므로 함께 해석하세요.
- 한쪽 버전의 프롬프트가 비어 있으면 신규 도입 또는 전면 교체된 버전임을 의미합니다.

지표 비교 카드

Property	Value
차트 유형	지표 카드 (A · B 값 병렬 표시)
비교 대상	A(기준) · B(비교)

Property	Value
액션	-

핵심 성능·비용·품질 지표를 카드별로 A와 B 값을 나란히 표시합니다.

Metric	Description
평균 Latency	요청 시작~응답 완료까지의 평균 시간
평균 TTFT	첫 토큰까지의 평균 시간 (체감 응답성)
평균 TPOT	출력 토큰당 평균 생성 간격 (스트리밍 속도)
호출당 비용 ($)	요청 1건당 평균 비용
에러율 (%)	호출 대비 에러 발생 비율
캐시 적중률	입력 토큰 중 캐시에서 가져온 토큰 비율

- A와 B의 값을 직접 대조하여, 변경이 각 지표를 개선했는지 악화시켰는지 한눈에 확인할 수 있습니다.
- 캐시 적중률이 오르면서 호출당 비용이 내려갔다면, 프롬프트 공통 prefix 증가로 캐싱 효율이 좋아졌을 가능성이 있습니다.

TTFT 추이 (A vs B)

Property	Value
차트 유형	라인 차트 (A, B 2개 시리즈)
비교 대상	A(기준) · B(비교)
액션	마우스 오버 시 값 표시

시간대별 평균 TTFT를 A·B로 겹쳐 표시합니다.

- 특정 시간대(예: 트래픽 피크)에서만 격차가 벌어지는지, 전 구간에서 고르게 차이가 나는지 확인하세요.
- 한쪽에서 간헐적 스파이크가 나타나면 큐 대기나 프로바이더 지연을 의심할 수 있습니다.

호출당 비용 추이 (A vs B)

Property	Value
차트 유형	라인 차트 (A, B 2개 시리즈)
비교 대상	A(기준) · B(비교)
액션	마우스 오버 시 값 표시

호출 1건당 비용의 시간대별 추이를 A·B로 비교합니다.

- 캐시 적중 패턴이나 출력 길이 변화가 호출당 비용에 미친 영향을 시간 흐름으로 확인할 수 있습니다.
- 평균값은 같아도 특정 구간에서만 비용이 튀는 패턴을 발견할 수 있습니다.

호출 볼륨 추이 (A vs B)

Property	Value
차트 유형	라인 차트 (A, B 2개 시리즈)
비교 대상	A(기준) · B(비교)
액션	마우스 오버 시 값 표시

두 프롬프트의 시간대별 호출량 추이를 비교합니다.

- 새 버전(B)으로의 트래픽 전환 진행 상황을 파악할 수 있습니다.
- 볼륨 차이가 큰 구간은 다른 지표를 해석할 때 표본 규모 차이의 근거가 됩니다.

에러 추이 (A vs B)

Property	Value
차트 유형	라인 차트 (A, B 2개 시리즈)
비교 대상	A(기준) · B(비교)
액션	마우스 오버 시 값 표시

시간대별 에러율 추이를 A·B로 비교합니다.

- 변경 직후 특정 시점에 에러가 몰렸는지, 안정적으로 낮게 유지되는지 확인하세요.
- B의 에러율이 A보다 높아지는 구간이 있으면 시스템 프롬프트 Diff에서 원인이 된 변경을 역추적하세요.

Finish Reason 분포

Property	Value
차트 유형	도넛 차트 (A · B 각각)
집계 단위	finish reason 값
액션	범례 항목 토글

A와 B 각각의 응답 종료 사유(finish reason) 분포를 도넛으로 나란히 표시합니다. 표시되는 값과 명칭은 프로바이더·모델에 따라 다를 수 있으며, WhaTap이 별도로 정의하지 않고 API가 내려준 값을 그대로 사용합니다.

- 길이 제한으로 인한 중단을 나타내는 값(예: `length`)의 비중이 B에서 줄었다면, 출력 제어가 개선되어 응답이 잘리는 경우가 감소했다는 의미입니다.
- 정상 종료를 나타내는 값(예: `stop`)의 비중 변화로 응답 완결성의 차이를 가늠할 수 있습니다.

ⓘ 표시되는 finish reason 값은 사용하는 LLM 프로바이더·모델의 응답 스펙을 따릅니다. 각 값의 정확한 의미는 해당 프로바이더의 API 문서를 참고하세요.

기능 사용 비교

Property	Value
차트 유형	막대 차트 (A vs B)
비교 대상	A(기준) · B(비교)
액션	-

기능별 사용 비율을 A·B로 비교합니다.

- 프롬프트 변경(예: 예시 추가, 지시 강화)이 의도한 기능 사용 패턴 변화를 유도했는지 검증할 수 있습니다.
- 한 호출이 여러 기능을 동시에 사용할 수 있어 비율의 합은 100%가 아닐 수 있습니다.

모델 분포

Property	Value
차트 유형	100% 누적 가로 막대 (A · B 각 행)
비교 대상	A(기준) · B(비교)
액션	범례 항목 토글

각 프롬프트가 호출한 모델의 구성 비율을 A·B 행으로 비교합니다. 범례의 색상이 각 모델에 대응합니다.

- A와 B의 모델 구성이 다르면 지연·비용 차이가 프롬프트 변경 때문인지 모델 차이 때문인지 구분해 해석해야 합니다.
- 특정 모델 비중이 크게 달라졌다면, 라우팅 규칙이나 모델 지정이 변경되었는지 점검하세요.

프롬프트 로그 (A · B)

Property	Value
차트 유형	테이블 (A · B 각각)
액션	행 펼치기, 프롬프트 로그 이동

A와 B 각각의 최근 호출 로그를 테이블로 표시하여, 지표 차이가 실제 응답에서 어떻게 나타나는지 개별 사례로 확인할 수 있습니다.

Column	Description
`시간`	호출 발생 시각
`프롬프트`	입력(Input)과 응답(Output)의 요약
`모델`	사용된 LLM 모델명
`토큰`	전체 토큰 수
`Latency`	요청 시작 ~ 응답 완료 시간
`비용`	해당 호출의 비용

- 응답 품질 평가가 적용된 호출은 `Eval` 영역에 평가 항목별 점수(예: Hallucination, Factuality, Relevance, Toxicity, Prompt Injection, PII Leak, URL Scan, Judge)가 함께 표시됩니다. 평가 항목에 대한 자세한 내용은 **응답 품질 분석** 문서를 참조하세요.
- 각 테이블 우측 상단의 **프롬프트 로그 이동** 버튼으로 해당 프롬프트의 원본 로그로 이동할 수 있습니다.

참고

비교 대상 지정 방식

Term	Description
오퍼레이션	기능 단위의 호출 묶음 (예: `cs_chat_support`)
프롬프트 버전	해당 오퍼레이션에서 사용된 시스템 프롬프트 버전 (예: `v3.0-structured`)
날짜	각 대상을 조회할 기준 날짜 (A·B 서로 다르게 지정 가능)

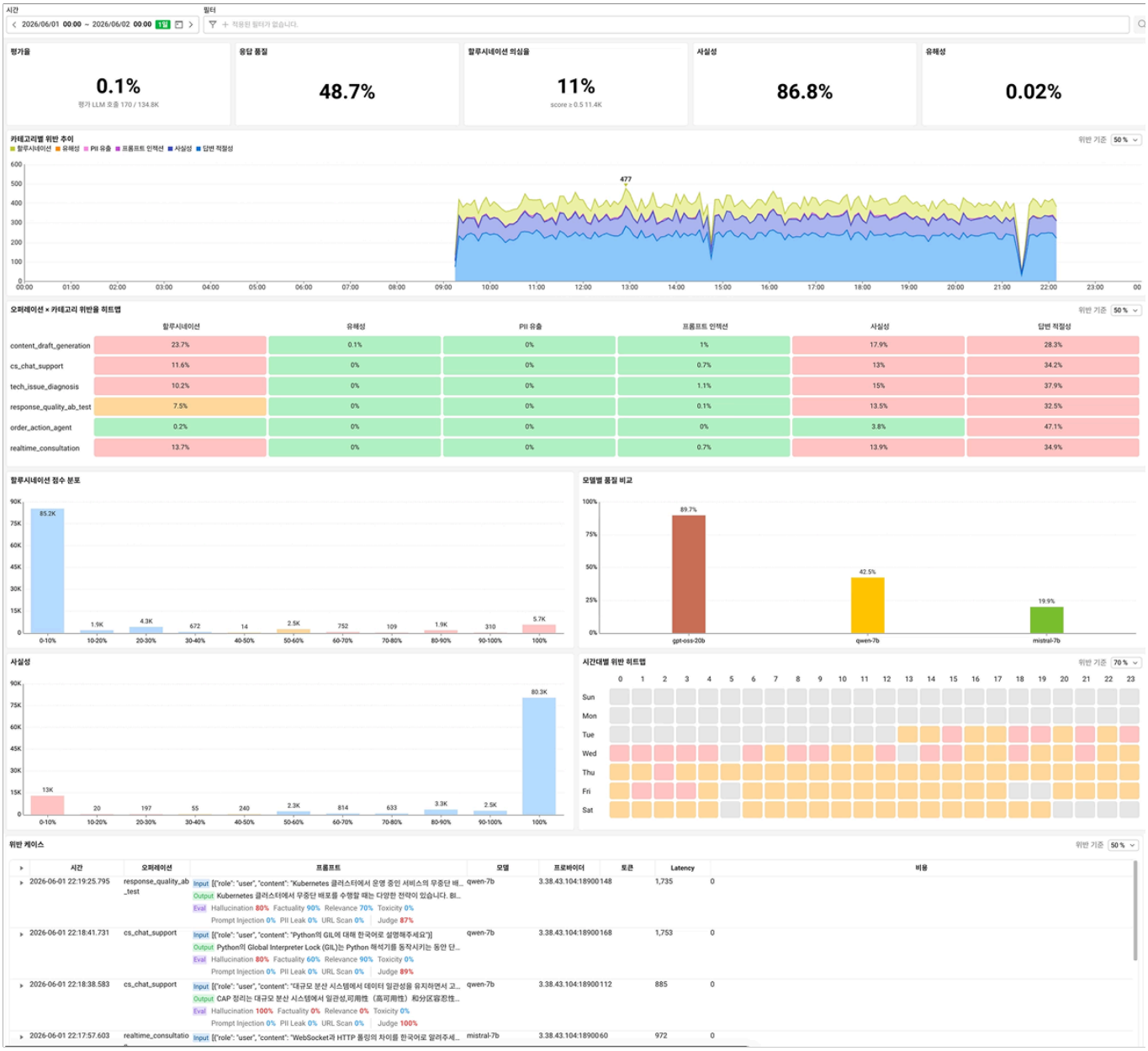
- 비교 대상 후보를 먼저 찾으려면 **프롬프트 분석**, 응답 품질·안전성 관점의 평가가 필요하면 **응답 품질 분석** 메뉴를 활용하세요.
- 모든 위젯에서 A는 파란색, B는 보라색으로 표시됩니다.

응답 품질 분석

LLM이 생성한 응답의 품질과 안전성을 자동 평가하여, 할루시네이션·유해성·PII 유출·프롬프트 인젝션·사실성·답변 적절성 등의 위험을 카테고리별로 모니터링하는 메뉴입니다. 호출량·비용·지연 같은 운영 지표만으로는 드러나지 않는 "응답 내용 자체가 믿을 만하고 안전한가"를 진단하는 데 활용합니다.

평가는 정규식·NLI·LLM Judge 등 여러 평가기를 조합한 **하이브리드 방식**으로 수행되며, 비용 통제를 위해 전체 호출 중 일부를 **샘플링**하여 평가합니다. 각 평가 항목은 점수로 산출되고, 위젯의 **위반 기준(임계값)** 을 기준으로 위반 여부가 집계됩니다.

화면은 위에서 아래로 **요약 → 위반 추이·히트맵 → 점수 분포·모델 비교 → 시간대 히트맵 → 위반 케이스** 순서로 구성되어, 전체 현황 파악부터 개별 위반 사례 확인까지 따라갈 수 있습니다. 상단 옵션바에서 시간 범위와 필터를 설정한 뒤 위젯 데이터를 조회합니다.



요약 카드

선택한 시간 범위의 핵심 품질 지표를 한눈에 보여주는 카드입니다.

평가율

전체 LLM 호출 중 품질 평가가 수행된 호출의 비율을 표시합니다. 카드 하단에는 평가 LLM 호출 수와 전체 호출 수가 함께 표시됩니다. (예: `평가 LLM 호출 170 / 134.8K`)

- 평가는 비용이 발생하므로 샘플링 비율 내에서 운영됩니다. 평가율이 낮으면 표본이 작아 다른 위젯의 비율 해석에 주의가 필요합니다.

응답 품질

평가된 응답의 종합 품질 점수를 표시합니다.

- 여러 평가 항목을 종합한 대표 지표로, 추세가 하락하면 품질 저하 신호로 해석하세요.

할루시네이션 의심율

할루시네이션 점수가 기준 이상인 호출의 비율을 표시합니다. 카드 하단에는 판정 기준과 해당 호출 수가 함께 표시됩니다. (예: `score ≥ 0.5 11.4K`)

- RAG·검색 기반 응답에서 사실과 다른 내용이 생성되는 정도를 나타냅니다. 비율이 상승하면 컨텍스트 구성이나 모델을 점검하세요.

사실성

응답이 사실에 부합하는 정도(Factuality)의 평균 수준을 표시합니다.

- 값이 높을수록 응답이 사실에 충실합니다. 할루시네이션 의심율과 함께 보면 사실 왜곡의 전반적인 경향을 파악할 수 있습니다.

유해성

유해(Toxicity)로 평가된 응답의 비율을 표시합니다.

- 사용자에게 노출되면 안 되는 공격적·유해 표현의 발생 정도를 모니터링합니다. 값이 급증하면 위반 케이스에서 실제 응답을 확인하세요.

카테고리별 위반 추이

Property	Value
차트 유형	누적 영역 차트(stacked area)
집계 단위	평가 카테고리
액션	위반 기준(임계값) 선택, 범례 항목 토글

시간대별로 각 위험 카테고리의 위반 발생량 추이를 누적 영역으로 표시합니다. 범례의 색상이 각 카테고리(할루시네이션·유해성·PII 유출·프롬프트 인젝션·사실성·답변 적절성)에 대응합니다.

- 우측 상단의 **위반 기준** 드롭다운으로 위반으로 집계할 점수 임계값을 조정할 수 있습니다.
- 특정 시점에 특정 카테고리가 급증했다면 배포·트래픽 변화와 연관 지어 원인을 추적하세요.

오퍼레이션 × 카테고리 위반율 히트맵

Property	Value
차트 유형	히트맵 테이블 (행: 오퍼레이션, 열: 카테고리)
집계 단위	오퍼레이션 × 카테고리
액션	위반 기준(임계값) 선택

행은 오퍼레이션, 열은 위험 카테고리이며, 각 셀에 해당 조합의 위반율(%)이 표시됩니다. 위반율이 높을수록 붉은 계열, 낮을수록 녹색 계열로 표시됩니다.

- "어떤 오퍼레이션이 어떤 위험에 취약한가"를 한눈에 짚어내는 위젯입니다.
- 예를 들어 특정 오퍼레이션의 답변 적절성 열이 붉게 나타나면, 해당 프롬프트가 질문 의도와 어긋난 응답을 자주 생성하고 있다는 의미입니다.
- 우측 상단의 **위반 기준** 드롭다운으로 임계값을 조정하면 셀 색상과 값이 함께 갱신됩니다.

할루시네이션 점수 분포

Property	Value
차트 유형	히스토그램 (x축: 점수 구간, y축: 평가 호출 수)
집계 단위	점수 구간 (0–10% … 100%)
액션	마우스 오버 시 값 표시

할루시네이션 평가 점수의 분포를 점수 구간별 호출 수로 표시합니다.

- 점수가 낮은 구간(0–10%)에 분포가 몰려 있으면 대부분의 응답에서 할루시네이션이 의심되지 않는다는 의미입니다.
- 높은 점수 구간(특히 100%)에 꼬리가 두껍게 나타나면 사실 왜곡이 의심되는 응답이 일정량 존재하므로 위반 케이스에서 실제 사례를 확인하세요.

모델별 품질 비교

Property	Value
차트 유형	막대 차트 (x축: 모델)
집계 단위	모델
액션	마우스 오버 시 값 표시

모델별 응답 품질 점수를 막대로 비교합니다.

- 품질 점수가 낮은 모델은 동일 워크로드에서 위반을 더 많이 유발할 수 있습니다.
- 비용·지연 지표와 함께 검토하여, 품질 대비 효율이 좋은 모델로의 전환을 판단하세요.

사실성 점수 분포

Property	Value
차트 유형	히스토그램 (x축: 점수 구간, y축: 평가 호출 수)
집계 단위	점수 구간 (0–10% … 100%)
액션	마우스 오버 시 값 표시

사실성(Factuality) 평가 점수의 분포를 점수 구간별 호출 수로 표시합니다.

- 점수가 높은 구간(특히 100%)에 분포가 몰려 있으면 대부분의 응답이 사실에 충실하다는 의미입니다.
- 낮은 점수 구간(0–10%)의 비중이 두껍다면 사실과 다른 응답이 일정량 존재하므로, 할루시네이션 점수 분포와 함께 해석하세요.

시간대별 위반 히트맵

Property	Value
차트 유형	히트맵 (요일 × 시간대)
집계 단위	요일(Sun–Sat) × 시간(0–23)
액션	위반 기준(임계값) 선택

요일(세로)과 시간대(가로)의 격자로 위반 발생 밀도를 표시합니다. 색이 진한 셀일수록 해당 시간대에 위반이 집중되어 있음을 의미합니다.

- 특정 요일·시간대(예: 평일 오후)에 위반이 몰린다면 트래픽 부하나 특정 사용자 패턴과의 연관성을 점검하세요.
- 우측 상단의 **위반 기준** 드롭다운으로 임계값을 조정할 수 있습니다.

위반 케이스

Property	Value
차트 유형	테이블
액션	위반 기준(임계값) 선택, 행 펼치기

평가 결과 위반으로 판정된 개별 호출을 행 단위로 나열합니다. 통계가 아닌 실제 사례를 직접 확인하기 위한 위젯입니다.

Column	Description
시간	호출 발생 시각
오퍼레이션	위반이 발생한 오퍼레이션
프롬프트	입력(Input)과 응답(Output)의 요약
모델	사용된 LLM 모델명
프로바이더	LLM 프로바이더
토큰	전체 토큰 수
Latency	요청 시작~응답 완료 시간
비용	해당 호출의 비용

- 행을 펼치면 입력·응답 원문과 함께 Eval 영역에 평가 항목별 점수가 표시됩니다. (Hallucination, Factuality, Relevance, Toxicity, Prompt Injection, PII Leak, URL Scan, Judge)
- 우측 상단의 **위반 기준** 드롭다운으로 위반으로 표시할 임계값을 조정할 수 있습니다.

참고

평가 카테고리

화면의 한글 카테고리명과 `Eval` 영역의 평가 항목은 다음과 같이 대응됩니다.

Category	Eval Item	Description
할루시네이션	Hallucination	컨텍스트에 없는 내용을 생성한 정도
사실성	Factuality	응답이 사실에 부합하는 정도
답변 적절성	Relevance	응답이 질문 의도에 부합하는 정도
유해성	Toxicity	공격적·유해 표현 포함 정도
프롬프트 인젝션	Prompt Injection	시스템 프롬프트 탈취·우회 시도
PII 유출	PII Leak	개인식별정보 노출 정도
-	URL Scan	응답에 포함된 URL의 위험 여부
-	Judge	LLM Judge 종합 판정 점수

위반 기준(임계값)

- 카테고리별 위반 추이, 위반율 히트맵, 시간대별 위반 히트맵, 위반 케이스 위젯은 우측 상단에서 **위반 기준**(점수 임계값)을 선택할 수 있습니다.
- 임계값을 조정하면 해당 위젯이 위반으로 집계하는 기준이 바뀌며 데이터가 함께 갱신됩니다.
- 평가는 전체 호출 중 일부를 샘플링하여 수행되므로 위반율은 평가된 호출을 기준으로 산출됩니다.

LLM 메트릭 용어 사전

이 문서는 AI Agent Observability에서 사용하는 주요 메트릭과 개념을 정리한 용어 사전입니다. 대시보드, 토큰 추이, 비용 분석 등 여러 메뉴에서 공통으로 사용되는 지표의 정의와 단위를 확인할 수 있습니다.

LLM 성능 지표

Metric	Description
`Latency`	LLM API 요청부터 응답 완료까지 전체 소요 시간(ms)으로 사용자 체감 전체 대기 시간
`TTFT`	Time To First Token. 요청 후 첫 번째 응답 토큰 도착까지의 시간(ms)으로 스트리밍 환경에서 응답 시작 체감 시점
`TPOT`	Time Per Output Token. 출력 토큰 1개 생성 평균 시간(ms)으로 높을수록 응답이 끊기는 느낌
`Output TPS`	Tokens Per Second. 초당 출력 토큰 수로 모델 전체 처리량 지표

토큰 및 비용

Metric	Description
`Input Tokens`	프롬프트 전체 입력 토큰 수. Cached Tokens 포함
`Output Tokens`	모델 생성 응답 토큰 수. 비용에 직접 영향
`Cached Tokens`	캐시에서 가져온 입력 토큰 수. Input Tokens의 부분집합. 일반 입력 대비 낮은 단가 적용
`Cache Hit Rate`	전체 입력 토큰 중 캐시 적중 토큰 비율(%). 높을수록 비용 절감
`I/O Ratio`	전체 토큰 중 출력 토큰 비율(%). 워크로드 특성 파악용(입력 위주 vs 생성 위주)

백분위수

Percentile	Description
p50	전체 데이터의 50% 이하 값으로 일반 사용자 체감 성능
p75	전체 데이터의 75% 이하 값으로 p50 격차로 상위 요청 추가 지연 파악
p95	전체 데이터의 95% 이하 값으로 SLA 기준으로 주로 사용
p99	전체 데이터의 99% 이하 값으로 상위 1% 성능, 시스템 안정성 판단 기준

품질 평가 지표

Metric	Description
Hallucination	환각 점수. 거짓 정보 또는 컨텍스트 모순 정도로 **낮을수록 좋음** (0.0 ~ 1.0)
Answer Relevance	답변 관련성 점수, 질문 적절 응답 여부로 **높을수록 좋음** (0.0 ~ 1.0)
Toxicity	유해성 점수, 혐오·욕설·폭력·성적 콘텐츠 수준으로 **낮을수록 좋음** (0.0 ~ 1.0)
Combined Judge	종합 위험도 max(hallucination, toxicity, 1 − answer_relevance) - 세 지표 중 최대 위험값. **낮을수록 좋음** (0.0 ~ 1.0)
Prompt Injection	프롬프트 인젝션 시도 탐지 정도로 **낮을수록 좋음** (0.0 ~ 1.0)
Factuality	응답 사실 정확성 점수로 **높을수록 좋음** (0.0 ~ 1.0)
PII Leak	개인식별정보(PII) 노출 정도
URL Scan	응답 내 URL 위험도

Metric	Description
Eval Success	평가 성공 여부 (1 = 정상, 0 = judge 호출/파싱 실패)

태그 기반 필터

AI Agent Observability의 모든 데이터는 다음 태그 기준으로 필터링하거나 그룹화할 수 있습니다.

Tag	Description	Example
Agent	LLM 에이전트(애플리케이션 인스턴스)	llm-app-01, llm-app-02
Model	사용된 LLM 모델	gpt-4o, claude-sonnet-4-20250514
Provider	LLM API 프로바이더	api.openai.com, api.anthropic.com
Operation	사용자 지정 호출 단위 라벨 (chain/prompt/agent 이름 등) - 미지정 시 default. prompt_meta로 설정	checkout_chain, rag_query, default
Prompt Version	프롬프트/체인 버전	v1, v2, v3

OpenMetrics 템플릿 목록

OpenMetrics를 처음 사용하는 사용자가 복잡한 설정 과정 없이 빠르게 메트릭을 수집하고 시각화할 수 있도록, 기본 템플릿과 사전 정의된 템플릿를 제공합니다. 이를 통해 자주 사용되는 오픈소스 메트릭을 손쉽게 적용할 수 있으며, 초기 도입 장벽을 낮추고 운영자가 확장 가능한 모니터링 환경을 구축할 수 있도록 지원합니다.

템플릿 목록

템플릿 목록은 기본 오픈소스 템플릿 카드를 제공합니다. 각 카드에 간단한 설명, 템플릿 이동, 수집 안내가 포함되어 있습니다.

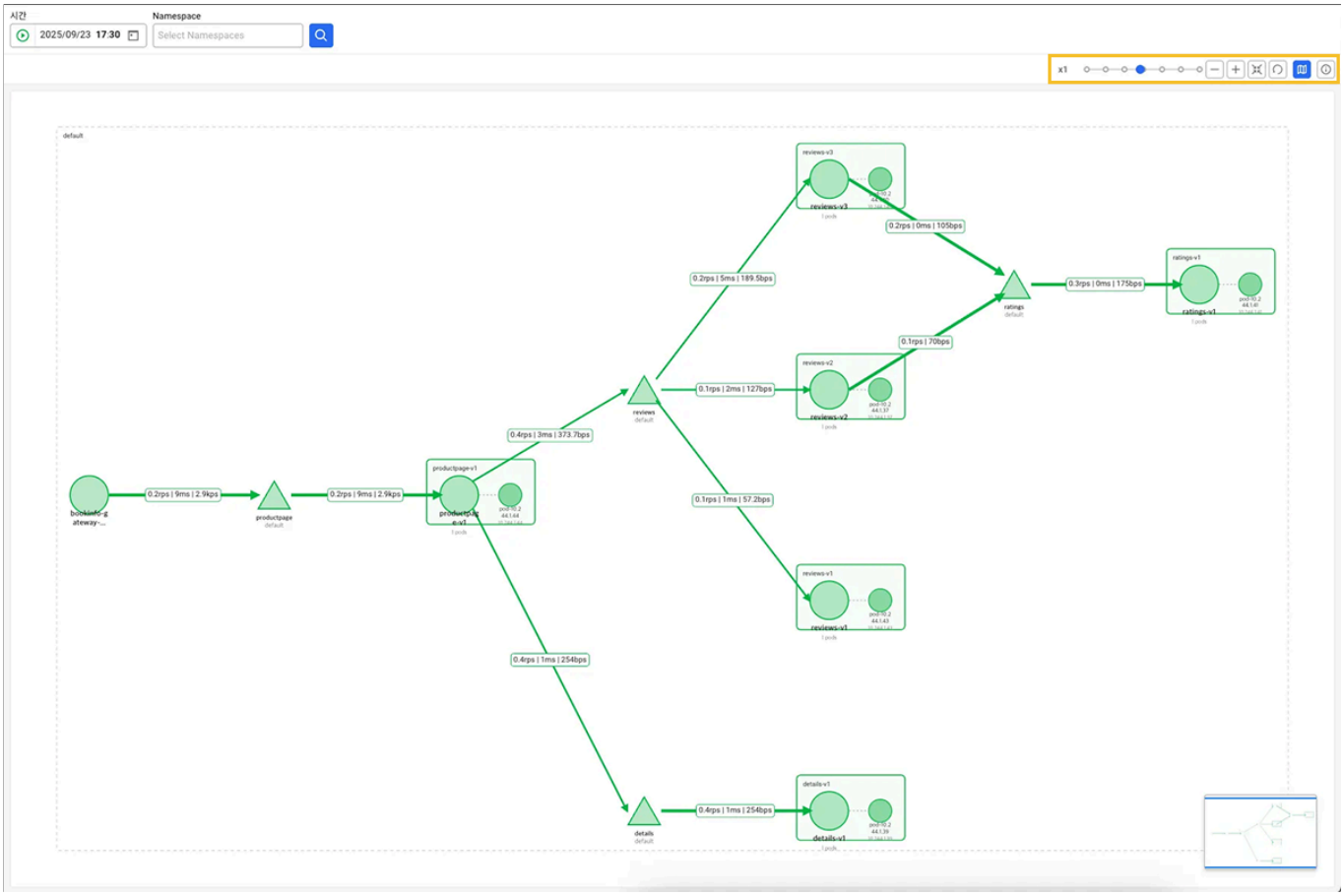
- → **템플릿**: 클릭하면 해당 템플릿 화면으로 바로 이동합니다.
- ⓘ **수집 안내**: 클릭하면 데이터 수집 설정 방법을 제공합니다.

Nginx 템플릿

Apache Nginx 모니터링을 통해 웹 서버의 성능을 실시간으로 추적하고 분석할 수 있습니다.

Istio 템플릿

Istio 서비스 메시 환경의 서비스 간 관계와 트래픽 흐름을 시각적으로 표현하며, WhaTap APM 데이터와 연계하여 통합 모니터링 환경을 템플릿으로 제공합니다. 화면 오른쪽 상단의 버튼을 클릭해 템플릿를 확대하거나 축소할 수 있습니다.



범례

화면 오른쪽 상단의 ⓘ 버튼을 클릭하면 범례를 확인할 수 있습니다.

노드 모양

- 삼각형: 쿠버네티스 서비스
- 원: 워크로드
- 사각형: 애플리케이션
- 작은 원: Pod(파드)

노드 상태

HTTP 요청/응답 전체 중 에러(4xx, 5xx)의 비율

- 초록색: 에러율 5% 미만
- 주황색: 에러율 5% 이상
- 빨간색: 에러율 10% 이상
- 회색: 요청 수 없음

Pod 노드 상태

쿠버네티스 Pod의 Phase

- 초록색: Running 상태의 Pod
- 주황색: Pending 상태의 Pod
- 빨간색: Failed 상태의 Pod

엣지 색상

두 노드 간의 HTTP 요청에서 발생한 에러(4xx, 5xx)의 비율

- 초록색: 에러율 10% 미만
- 주황색: 에러율 10% 이상
- 빨간색: 에러율 20% 이상
- 파랑색: TCP 통신
- 회색: 상태 알 수 없음

OpenMetrics 탐색기

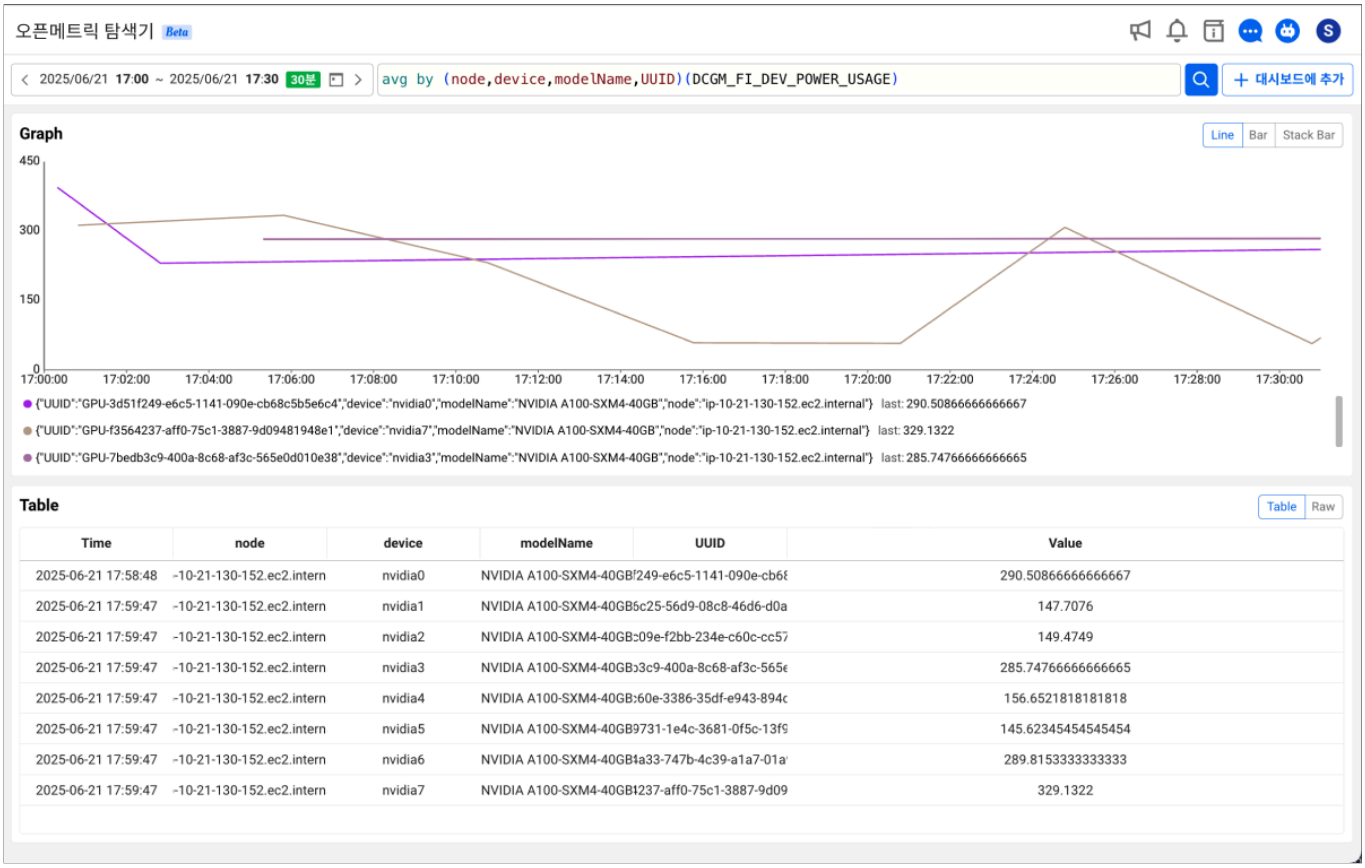
OpenMetrics 탐색기는 Prometheus 기반의 커스텀 메트릭을 PromQL을 통해 직접 탐색하고, 다양한 형태로 시각화하여 메트릭의 상태를 확인하거나 인사이트를 얻을 수 있는 기능입니다.

이 기능을 통해 원하는 메트릭을 자유롭게 조회하고, 유의미한 데이터를 추출하거나 대시보드로 연동하여 지속적으로 모니터링 할 수 있습니다.

> ⓘ **OpenMetrics 탐색기**를 사용하려면 오픈 에이전트 설치가 필요합니다.

PromQL 탐색하기

OpenMetrics 탐색기에서 PromQL을 실행하고, 다양한 시각화 옵션으로 결과를 확인할 수 있습니다.



Graph

- **Line**: 기본 시계열 라인 그래프
- **Bar**: 막대그래프 형식으로 시각화
- **StackBar**: 누적된 값을 보여주는 스택형 바 차트

Table / Raw

- **Table**: 각 시계열의 값들을 시간 순서대로 정렬해 표로 표시
- **Raw**: Prometheus API의 응답(JSON 형태)을 그대로 확인 가능

PromQL 문법 지원 및 자동완성

`node_cpu_seconds_total` , `container_memory_usage_bytes` 등의 메트릭을 입력하면 자동완성 기능으로 메트릭 및

연산자 목록을 제시합니다.

`{pod="example"}` 형태의 라벨 필터링도 지원합니다.

대시보드에 추가하기

조회한 PromQL 쿼리를 대시보드 위젯으로 저장하여, 원하는 메트릭을 지속적으로 모니터링할 수 있습니다.

추가 방법

1. **OpenMetrics 탐색기** 화면 오른쪽 위의 **+ 대시보드에 추가** 버튼을 클릭하면, **대시보드에 추가** 모달 창이 열립니다.
2. **대시보드에 추가** 창에서 기존 대시보드 선택하거나 신규 대시보드를 생성합니다.
3. **적용** 버튼을 클릭하면 해당 위젯이 지정된 대시보드에 추가됩니다.
    - 추가된 대시보드에서 위젯 이름 및 설명을 입력할 수 있습니다.
    - 추가된 위젯은 Flex보드의 Grid 레이아웃에서 자유롭게 위치 조정 가능하며, PromQL 수정할 수 있습니다.

라이브 테일

> ⓘ **로그 조회** 권한이 없을 경우 해당 메뉴에 진입할 수 없습니다.

홈 화면 > 프로젝트 선택 > **로그** > **라이브 테일**

라이브 테일 메뉴에서 서버 콘솔에 접속하지 않고, 로그 데이터 스트림을 모니터링 화면상에서 바로 확인할 수 있습니다. 복잡한 대량의 로그도 **필터**와 **하이라이트** 기능을 활용해 선별하고 빠르게 인지할 수 있습니다. 로그 데이터 조회 주기는 2초입니다.

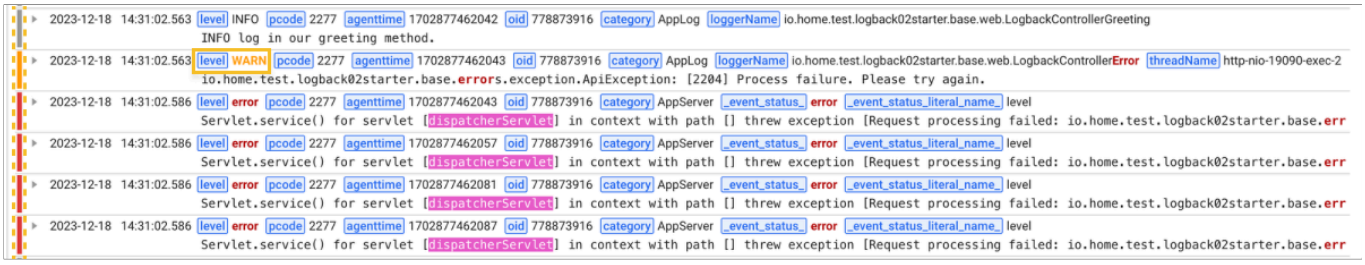
주요 용어는 다음과 같습니다.

- **Category**: 로그의 수집 및 조회 단위
- **Content**: 로그 메시지
- **Search Key**: 로그 파서 설정을 통해 생성되는 검색 키
- **Tag**: 수집된 로그를 검색할 수 있는 태그

> ⓘ 로그 테이블 컬럼 가장자리를 드래그해 컬럼 너비를 수정할 수 있습니다.

에이전트 옵션

에이전트 옵션이 설정된 경우 로그 레벨을 수집해 로그 레벨 기준 색상이 다음과 같이 표시됩니다.



에이전트 옵션 설정

```
# whatap.conf
weaving=log4j-2.17
weaving=logback-1.2.8
```

- Java 에이전트 2.2.22 버전 이후부터 위빙 설정에 log4j-2.17 또는 logback-1.2.8 설정 시 사용할 수 있습니다. 에이전트 재시작이 필요합니다.
- 로그 레벨은 파싱된 키워드 중 `level`, `type` 기준으로 판별합니다. `level`, `type`으로 파싱된 키가 존재하고 파싱 값이 FATAL, CRITICAL, ERROR, WARN, WARNING, INFO를 포함할 경우 로그 레벨 색상을 표시합니다.

필터 적용하기

필터를 적용하면 입력한 조건에 맞는 로그를 필터링합니다. 복수의 필터를 입력할 수 있습니다. 필터의 태그가 같은 경우 OR(`||`)로, 그렇지 않은 경우는 AND(`&&`)로 적용됩니다.

입력 창에 값을 직접 입력하거나 **필터** 입력 창을 클릭해 필터를 지정할 수 있습니다. 필터 태그는 `검색 키`, `연산자`, `검색 값`의 순서로 입력합니다. **검색** 버튼을 선택하면 필터가 적용된 데이터를 로그 목록에서 조회할 수 있습니다.

> **가이드 UI**
>
> 다음과 같이 입력 창 아래 **가이드 UI**를 제공합니다. 입력 창 또는 필터 태그에 마우스 커서가 있을 때 ESC 키를 통해 **가이드 UI**를 닫을 수 있습니다.

검색 키, 연산자, 검색 값 입력

- `검색 키` 입력 시 일반 인덱스, 예약어 인덱스, 숫자만 입력할 수 있는 인덱스를 구분해 추천 값을 제공합니다
- `연산자` 입력 시 일반 인덱스 검색 키의 경우 `==`, `!=` 옵션을 하단에 안내합니다. 숫자만 입력할 수 있는 인덱스의 경우 `>`, `<`, `<=`, `>=`, `==`, `!=` 옵션을 제공합니다.
- `검색 값` 입력 시 일치 검색(`>`, `<`, `<=`, `>=`, `==`)일 때 파란색으로, 제외 검색(`!=`)일 때 붉은색으로 하이라이팅합니다. 대소문자 구분 옵션을 활용해 검색할 수 있습니다.

- 필터 태그가 2줄 이상 길어지는 경우 ∧ **접기** 아이콘을 선택해 접어둘 수 있습니다.
- 필터 태그 입력 후 입력 창에서 Shift와 Tab 키를 동시에 사용하여 이전 필터 태그로 이동할 수 있습니다.
- 필터 태그 입력 후 입력 창에서 Tab 키를 통해 다음 필터 태그로 이동할 수 있습니다.

필터 태그 추가

입력 창에 텍스트 입력 후 Enter 또는 Tab 키를 누르거나, 입력창 하단의 가이드 UI에서 추천 값을 클릭하거나 방향키로 선택해 추가할 수 있습니다.

필터 태그 제거

Backspace 키 또는 태그의 **X** 아이콘을 눌러 태그를 삭제할 수 있습니다. 입력창 오른쪽의 **X** 아이콘을 클릭해 모든 태그를 한 번에 삭제할 수 있습니다.

필터 즐겨찾기

필터 **즐겨찾기** 기능을 제공합니다. 입력 창 오른쪽의 ☆ **즐겨찾기** 아이콘을 선택하여 원하는 필터 검색 조건을 즐겨찾기에 **추가**, **제거** 및 기존 즐겨찾기를 선택할 수 있습니다. **즐겨찾기**는 최대 50개까지 저장됩니다.

필터 적용 예외 상황

- 숫자만 입력할 수 있는 인덱스(`.n` 으로 끝나는 `검색 키`)를 입력한 태그에서 `검색 값` 은 숫자만 입력할 수 있습니다.
- 중복된 `검색 키` , `검색 값` 은 입력할 수 없습니다.
- `검색 키` , `검색 값` 중 하나라도 없는 태그가 존재할 때 검색할 수 없습니다. 유효하지 않는 태그의 경우 회색으로 표시합니다.

> ⓘ **라이브 테일** `검색 키` 로 `category` 를 입력할 수 없습니다.
> 입력된 필터 값 아래에 있는 수식(expression)은 로그 데이터 조회 시 적용될 필터 수식 미리 보기입니다.

미파싱 키워드 필터 적용

로그에서 파싱되지 않은 즉 인덱스가 생성되지 않은 키워드를 포함한 로그를 조회할 수 있습니다. 이 경우 지정 범위 내 모든 로그를 Full Scan합니다. 그렇기 때문에 인덱스가 생성된 키와 비교해 검색 속도가 다소 떨어질 수 있습니다. 정형화된 로그 데이터의 경우 **로그 파서 설정**을 통해 인덱스 키 값을 활용해 검색하는 것을 권장합니다.

1. **카테고리**를 선택하세요.
2. **필터** 입력창에 `content` 기준 띄어쓰기 후 검색을 원하는 키워드를 입력하세요.
   예시, `content *select*`
3. **검색** 버튼을 클릭해 로그를 조회하세요.

> **라이브 테일**의 경우 모든 로그 검색이 가능해 카테고리를 지정할 필요가 없습니다.
> 파서 설정에 대한 자세한 내용은 다음 문서를 참조하세요.

필터 수정

필터에 값을 입력한 뒤 입력한 값을 클릭하면 해당 값을 수정할 수 있습니다.

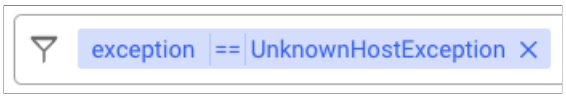
- 입력 창에 텍스트 재입력해 수정할 수 있습니다.
- 입력 창 아래 **가이드 UI**를 통해 추천 값을 선택해 수정할 수 있습니다.

검색 키(Search Key)

다음 이미지에서 파란색 박스 부분은 파싱(parsing)된 검색 키입니다. 검색 키는 **로그 설정**의 **로그 파서 설정** 탭에서 파싱 로직을 등록해 설정할 수 있습니다.

필터 입력 문법

태그는 검색 키와 검색 값으로 구성되어있습니다. 다음의 예시에서 검색 키는 `exception` , 검색 값은 `UnknownHostException` 입니다. 해당 예시는 수집한 로그 데이터 중 IP 주소와 도메인 주소가 매칭되지 않아 서버를 호스트에 연결할 수 없을 경우 발생하는 예외(`UnknownHostException`)가 포함된 로그 데이터를 조회합니다.



검색 키 종류

검색 키 종류	검색 키 포멧	의미	검색 키와 검색 값 예시	검색 예시
문자열 키워드	keyword	파일 이름	- 키: fileName - 값: /data/whatap/logs/yard.log	fileName:/data/whatap/logs/yard.log
숫자 키워드	keyword.n	응답시간	- 키: response_time.n - 값: 2945	response_time.n>=2945
예약어 키워드 (사전 정의 키워드)	@keyword	트랜잭션 ID	- 키: @txid - 값: 85459614215434144	-
로그 본문 키워드	content	로그 본문	- 키: content - 값: 사용자 입력값	content: `*ERROR*`

Content 검색 키

- Content 검색 키는 인덱싱되지 않은 로그의 본문을 대상으로 검색합니다. 예를 들어 `content *ERROR*` 와 같이 입력하는 경우 로그 본문 중 `ERROR` 를 포함한 로그를 검색합니다.
- 어떤 키워드로 인덱싱을 걸어야하는지 모르는 경우 Content 검색 키를 활용해 문제가 되는 키워드를 포함한 로그를 식별합니다. 이후 **로그 설정** 메뉴의 로그 파서 설정을 통해 해당 키워드로 파서를 설정해 인덱스를 생성하는 방식으로 검색 속도를 향상시킬 수 있습니다.

공통 문법

문법 종류	설명	예시
`==searchValue`	검색 값과 일치하는 로그를 검색합니다.	`exception==RuntimeExceptionexception`
`!=searchValue`	검색 값을 제외한 로그를 검색합니다.	`exception!=RuntimeException`

문법 종류	설명	예시
`*searchValue`	검색 값으로 끝나는 로그를 검색합니다.	`word==*hello`
`searchValue*`	검색 값으로 시작하는 로그를 검색합니다.	`word==hello*`
`*searchValue*`	검색 값으로 중간에 포함된 로그를 검색합니다.	`word==*hello*`
`*search*Value*`	검색 값으로 포함된 로그를 검색합니다.	`word==*he*llo*`
`re:{regex}`	정규표현식에 매칭되는 로그를 검색합니다.	`caller==re:^i\.w\.a\.w\.s\.v\.r\.`
`**`	검색 키에 해당하는 모든 로그를 검색합니다.	

검색 키가 숫자 키워드(keyword.n)인 경우 문법

다음의 문법은 검색 키가 `keyword.n` 형식인 경우에만 지원합니다.

- 검색 값으로는 숫자만 올 수 있습니다.
- `.n` 키워드의 값에는 prefix를 붙이지 않습니다. `.n` 이 아닌 키워드는 모두 prefix를 붙여야합니다.

  예, `+>searchValue` 는 유효하지 않습니다.

문법 종류	설명	예시
`>searchValue`	검색 값보다 큰 값이 포함된 로그를 조회합니다.	`response_time.n>3000`
`>=searchValue`	검색 값보다 크거나 같은 값이 포함된 로그를 조회합니다.	`response_time.n>=3000`
`==searchValue`	검색 값보다 같은 값이 포함된 로그를 조회합니다.	`response_time.n==3000`
`!=searchValue`	검색 값보다 다른 값이 포함된 로그를 조회합니다.	`response_time.n!=3000`
`<searchValue`	검색 값보다 작은 값이 포함된 로그를 조회합니다.	`response_time.n<3000`
`<=searchValue`	검색 값보다 작거나 같은 값이 포함된 로그를 조회합니다.	`response_time.n<=3000`

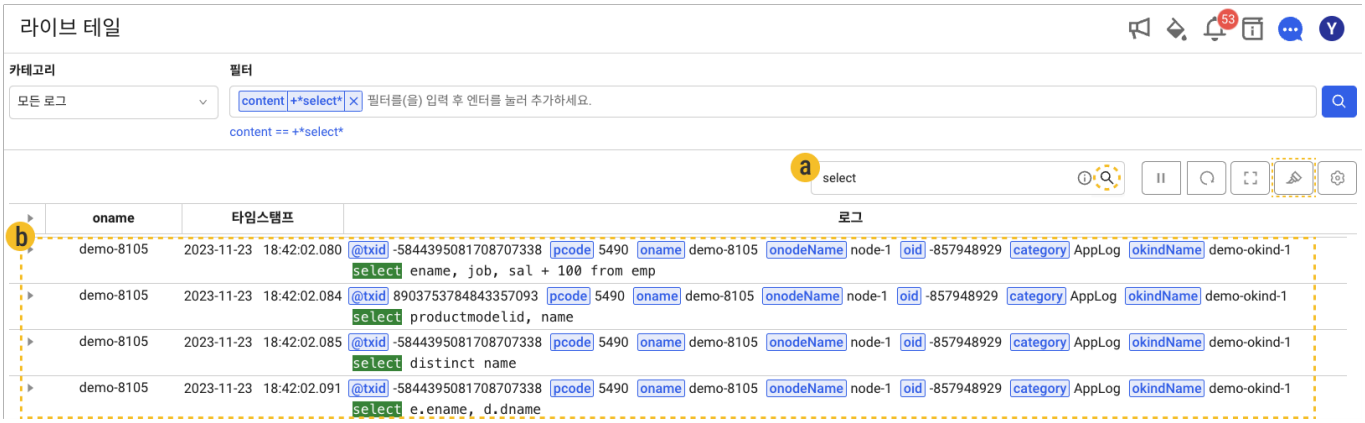
개인정보 조회

로그 개인정보 조회 권한이 있을 경우, **개인정보 표시** 토글 버튼 활성화를 통해 비식별화 처리한 개인정보를 표시할 수 있습니다.

> ⓘ
> - **로그 개인정보 조회** 권한에 대한 자세한 내용은 다음 문서를 참고하세요.
> - **로그 개인정보 비식별화** 기능에 대한 자세한 내용은 다음 문서를 참고하세요.

콘텐츠 하이라이트

로그의 콘텐츠 중 원하는 키워드를 손쉽게 식별하기 위해 하이라이트 기능을 제공합니다. 단일 또는 복수 키워드로 필터링할 수 있습니다.



1. 키워드 입력창에 하이라이트를 원하는 키워드를 입력하세요.
   - 예시, `select`
2. 로그 목록의 Content 내 검색한 키워드가 하이라이팅 됩니다.
   - **로그 전체 화면** 아이콘을 선택하면 **로그**와 **타임스탬프**를 전체 화면에서 확인할 수 있습니다.
   - 하이라이트된 키워드 간 이동은 단축키로 조작할 수 있습니다.
     - `Enter` : 다음 row로 이동
     - `Shift+Enter` : 이전 row로 이동

- ∧ / ∨ 아이콘: row 이동
- `Cmd` + `F` (Mac) / `Ctrl` + `F` (Windows): 키워드 검색

복수 키워드 조건

복수 키워드로 하이라이팅을 할 경우 다음과 같이 작성합니다.

입력 문자열	설명	결과
`a b c`	띄어쓰기로 각 키워드를 구분합니다.	a, b, c
`"Whatap is good."`	띄어쓰기를 키워드에 포함하고 싶은 경우 `''` 또는 `""`로 감쌉니다.	Whatap is good.
`"Whatap\\ is good."`	""로 감싸진 키워드에서 `\`를 포함할 경우, `\\`로 입력해야합니다.	Whatap\ is good.

하이라이트 색상 설정

하이라이팅할 키워드 및 색상을 설정할 수 있습니다.

- 기본적으로 로그 레벨에 따른 하이라이팅(FATAL, ERROR, WARN, SEVERE, CRITICAL)이 적용되어 있습니다.
- 설정한 내용은 **프로젝트 단위**로 저장됩니다.

1. 아이콘을 클릭하세요.
2. **하이라이트** 창에서 추가적으로 색상 설정을 원하는 키워드를 입력창에 입력하세요.
3. 입력창 왼쪽 색상 클릭 시 색을 선택할 수 있습니다.

테이블 설정

컬럼 설정

컬럼을 추가하거나 순서를 설정할 수 있습니다. 화면 오른쪽에서 **컬럼 설정** 버튼을 클릭하세요.

컬럼 추가

태그를 선택하여 테이블에 컬럼을 추가할 수 있습니다. **log** 컬럼 선택을 해제할 경우 **로그 표시 상세 설정**을 확인할 수 없습니다. 반드시 한 개 이상의 컬럼을 선택하세요.

- **제공되는 컬럼**: 로그, 수집 시간, content(로그 메시지)

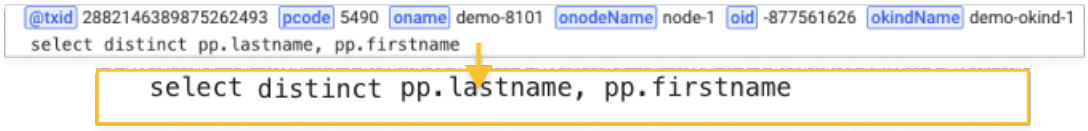
컬럼 순서 변경

컬럼을 추가하면 **표시 항목**에 해당 컬럼이 추가됩니다. 원하는 컬럼을 드래그하여 컬럼의 순서를 변경하세요.

로그 표시 상세 설정

1. 화면의 오른쪽에서 ⚙ **로그 표시 상세 설정** 버튼을 클릭하세요.
2. **로그 표시 상세 설정** 창에서 **content**와 **태그** 중 하나는 반드시 선택하세요.
    - 기본으로 **content**, **태그** 모두 선택되어있으며 두 가지 항목 모두 표시합니다.

체크가 해제된 항목은 테이블에 표시되지 않습니다. 다음과 같이 **태그**를 해제할 경우 테이블에서 로그의 **태그**는 표시되지 않습니다.



태그 관리 목록에 태그를 추가하면 추가한 순서대로 로그의 태그가 나열됩니다. 태그의 순서는 드래그하여 변경할 수 있습니다. 추가한 태그를 비활성화하면 비활성화한 태그는 로그의 태그에 노출되지 않습니다.

> ⓘ **컬럼 및 로그 표시 설정 적용 범위**
>
> - **컬럼 설정**과 **로그 표시 상세 설정**은 **라이브 테일**, **로그 검색**, **로그 트렌드**에서 사용할 수 있습니다.
> - 동일한 프로젝트 내 **라이브 테일**, **로그 검색**, **로그 트렌드**는 **컬럼 설정**과 **로그 표시 상세 설정**을 공유합니다.

로그 파일 다운로드

검색한 로그의 결과값을 CSV, TXT 파일 형식으로 수집된 로그를 다운로드 받을 수 있습니다.

1. 화면 중앙의 오른쪽의 **다운로드** 버튼을 클릭합니다.
2. **CSV 다운로드** 또는 **Log 다운로드**를 클릭해 원하는 형태의 파일을 다운로드 받습니다.

로그 탐색

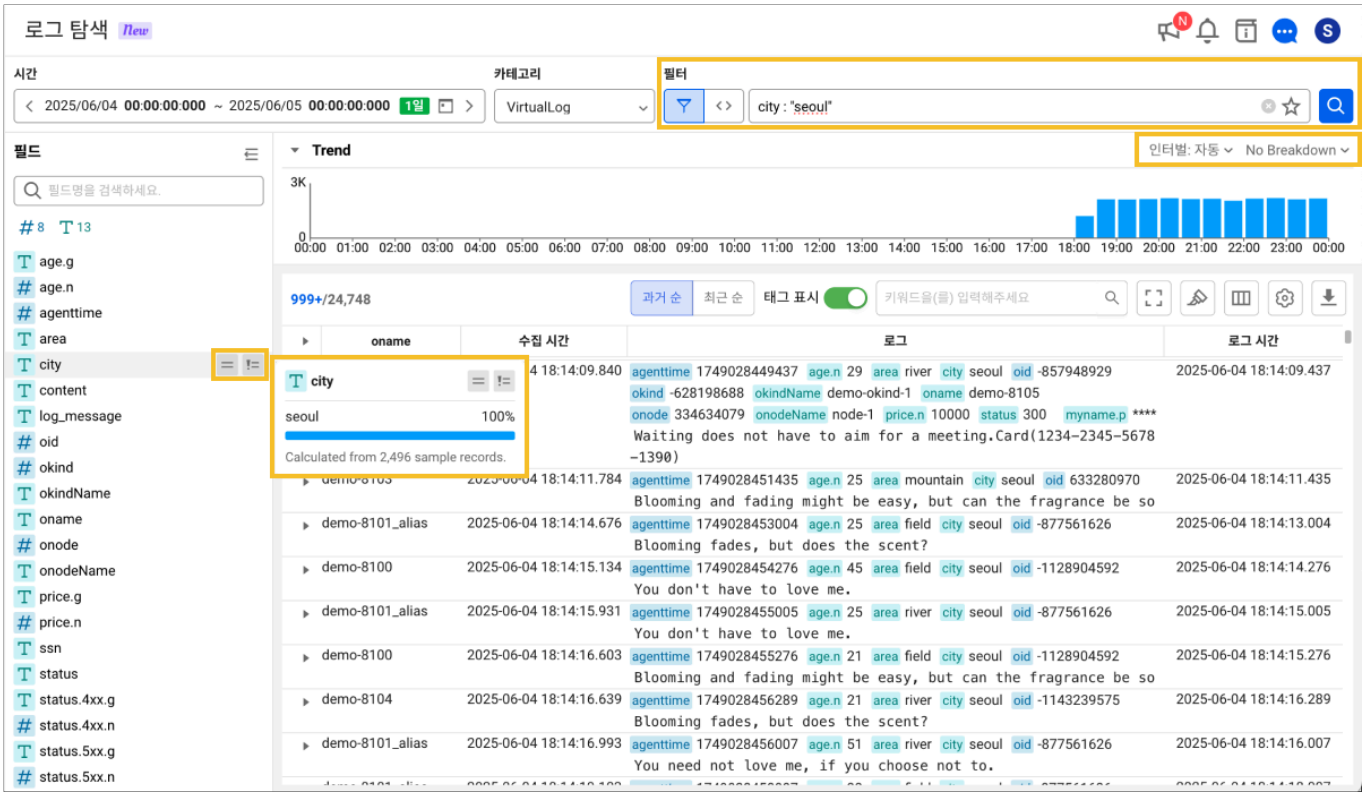
> **로그 조회** 권한이 없을 경우 해당 메뉴에 진입할 수 없습니다.

홈 화면 > 프로젝트 선택 > **로그** > **로그 탐색** *New*

로그 탐색(Log Explorer) 기능은 로그 데이터를 빠르게 검색, 필터링, 분석합니다. 필드 구조에 대한 정보를 파악하고, 쿼리 기반 필터링을 통해 필요한 로그를 효율적으로 검색해, 결과를 데이터 시각화하여 확인할 수 있습니다. 본 기능은 Lucene 인덱스 기반으로 동작하며, 주요 특징은 다음과 같습니다.

1. **로그 트렌드 분석 및 검색**: 로그 트렌드와 검색 기능을 동시에 지원하여, 시간에 따른 로그 변화 추이를 파악할 수 있습니다.
2. **빠른 검색 및 필터링**: 사용자는 간단한 단어 입력만으로도 문제의 로그를 신속하게 검색할 수 있습니다. WhaTap log search query와 Lucene 쿼리 언어를 지원하여, 다양한 검색 조건을 설정할 수 있습니다.
3. **자동완성 기능**: 쿼리 입력 시 자동완성 기능을 통해 사용자가 쉽게 쿼리를 작성하고 학습할 수 있도록 돕습니다.
4. **데이터 시각화**: 스택 그래프를 활용하여 시계열 로그 건수 데이터를 시각적으로 표현하며, 드릴다운 기능을 통해 특정 로그를 심층 분석할 수 있습니다.
5. **즐겨찾기 및 공유**: 자주 사용하는 필터를 즐겨찾기로 저장하고, URL을 통해 다른 사용자와 쉽게 공유할 수 있습니다.

기본 화면 안내



시간 설정

상단 옵션바의 **타임셀렉터**를 통해 로그 검색 시간 범위를 설정할 수 있습니다.

카테고리

상단 옵션바의 **카테고리 선택**에서 탐색하고 싶은 로그 종류 선택합니다. (예: AppLog 등)

필터

상단 옵션 바의 **필터**에서 쿼리 언어를 선택하고 검색 조건을 입력하여 필터를 적용합니다. 검색 쿼리 문법은 WhaTap 로그 검색 쿼리와 Lucene 쿼리를 지원합니다.

- : WhaTap 로그 검색 쿼리
- : Lucene 쿼리

검색 쿼리 문법

WhaTap log search query

WhaTap 로그 검색 쿼리는 자동 완성 기능을 제공하며, 간단한 단어 입력만으로도 문제의 로그를 신속하게 검색할 수 있습니다. and, or 등 다양한 문법을 활용해 다양한 검색 조건을 입력할 수 있습니다.

- 예, 검색 필터에 `error` 만 입력하여 content(로그 본문) 검색할 수 있습니다.

    1. 상단 옵션 바에서 필터 아이콘을 클릭한 후, 쿼리 검색창에서 필드(선택한 시간과 카테고리에 맞는 목록 노출)를 선택하세요.
    2. 연산자를 선택하세요.

        a. 기본: `:` , `:*` , `and` , `or`
        b. 선택한 필드가 `Number` 인 경우(기본 + `<=` , `>=` , `<` , `>`)
        c. 선택한 필드가 `String` 인 경우(기본)

    3. 값(선택한 시간, 카테고리, 필드에 맞는 목록 노출)을 선택하세요.

검색 인덱스 필드 타입

지원 필드 타입은 `Number` , `String` , `Boolean` 세 가지 유형만 존재합니다.

- `Number` : Long, double, float 타입의 숫자 값
- `String` : 이메일 본문과 같은 전체 텍스트
- `Boolean` : True 또는 False 값

> WhaTap log search query 문법에 대한 자세한 내용은 다음 문서를 참고하세요.

Lucene

apache lucene 8.11.1 버전 오픈소스 문법을 그대로 사용하세요. 단, 자동 완성 기능을 제공하지 않습니다.

ⓘ Lucene의 QueryParser 클래식 패키지에 대한 자세한 내용은 공식 문서를 참고하세요.

필드 목록

드릴 다운 분석에 활용할 수 있습니다.

필드 정보 필터 반영하기

필드 목록에서 필드명을 확인하고, `=`, `!=` 버튼을 클릭해 조건을 필터에 적용하세요.

- 필드에 마우스 호버 시 `=`, `!=` 버튼이 보이고 클릭 시 필터에 반영됩니다.
- 필드 검색 기능을 통해 원하는 필드를 빠르게 찾을 수 있습니다.

Trend 차트

Trend 차트에서 **인터벌**, **Breakdown**을 적용해 그래프로 로그를 확인할 수 있습니다.

- **그래프 인터랙션**: 마우스 드래그 및 🔍 버튼으로 시간 줌인, 바 클릭으로 해당 영역의 로그 확인 가능
- **인터벌**: 자동, 1초, 5초, 30초, 1분, 5분, 10분, 30분, 1시간 중 선택 가능
- **Breakdown**: 특정 필드를 기준으로 Top 3 + Others 값을 분포 시각화하고, 드릴다운 필터 적용 가능

breakdown

breakdown은 로그 건수 추이를 특정 필드로 분석해주는 시각화 기능입니다.

- 특정 필드의 Top 3 값과 Others(나머지 값)를 구분하여 스택 그래프로 분포를 표시합니다.
- 드릴다운 분석에 활용할 수 있으며, 범례와 스택 그래프에서 특정 영역(부분 그래프)를 클릭해 특정 값만 강조하거나 필터 조건을 추가할 수 있습니다.

로그 확인하기

검색 조건에 맞는 로그 목록을 확인할 수 있으며, **필드 목록**에서 필드 정보와 로그 태그 색이 동일하게 표시됩니다.

콘텐츠 하이라이트

로그의 콘텐츠 중 원하는 키워드를 손쉽게 식별하기 위해 하이라이트 기능을 제공합니다. 단일 또는 복수 키워드로 필터링할 수 있습니다.

1. 키워드 입력창에 하이라이트를 원하는 키워드를 입력하세요.
    - 예시, `select`
2. 로그 목록의 Content 내 검색한 키워드가 하이라이팅 됩니다.
    - ⛶ **로그 전체 화면** 아이콘을 선택하면 **로그**와 **타임스탬프**를 전체 화면에서 확인할 수 있습니다.
    - 하이라이트된 키워드 간 이동은 단축키로 조작할 수 있습니다.
        - `Enter` : 다음 row로 이동
        - `Shift+Enter` : 이전 row로 이동
        - ∧ / ∨ 아이콘: row 이동
        - `Cmd` + `F` (Mac) / `Ctrl` + `F` (Windows): 키워드 검색

복수 키워드 조건

복수 키워드로 하이라이팅을 할 경우 다음과 같이 작성합니다.

입력 문자열	설명	결과
`a b c`	띄어쓰기로 각 키워드를 구분합니다.	a, b, c
`"Whatap is good."`	띄어쓰기를 키워드에 포함하고 싶은 경우 `''` 또는 `""`로 감쌉니다.	Whatap is good.
`"Whatap\\ is good."`	""로 감싸진 키워드에서 `\`를 포함할 경우, `\\`로 입력해야합니다.	Whatap\ is good.

하이라이트 색상 설정

하이라이팅할 키워드 및 색상을 설정할 수 있습니다.

- 기본적으로 로그 레벨에 따른 하이라이팅(FATAL, ERROR, WARN, SEVERE, CRITICAL)이 적용되어 있습니다.

- 설정한 내용은 **프로젝트 단위**로 저장됩니다.

1. 아이콘을 클릭하세요.
2. **하이라이트** 창에서 추가적으로 색상 설정을 원하는 키워드를 입력창에 입력하세요.
3. 입력창 왼쪽 색상 클릭 시 색을 선택할 수 있습니다.

컬럼 설정

버튼을 클릭해, 컬럼을 추가하거나 순서를 설정할 수 있습니다.

- **컬럼 추가**

  컬럼 설정에서 태그를 선택하여 테이블에 컬럼을 추가할 수 있습니다.

  > 로그 컬럼 선택을 해제할 경우 **로그 표시 상세 설정**을 확인할 수 없습니다. 반드시 한 개 이상의 컬럼을 선택하세요.

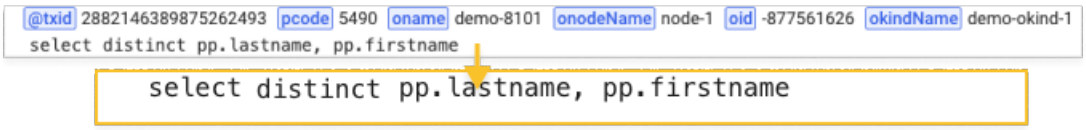
- **컬럼 순서 설정**

  컬럼 설정에서 컬럼을 추가하면 **표시 항목**에 해당 컬럼이 추가됩니다. 원하는 컬럼을 드래그하여 컬럼의 순서를 변경하세요.

로그 표시 상세 설정

버튼을 클릭해, 로그 표시를 상세 설정합니다. 기본으로 **content**, **태그** 모두 체크가 되어있으며 두 가지 항목 모두 표시합니다.

- **content**와 **태그** 중 하나는 반드시 선택하세요.
- **태그 관리** 목록에 태그를 추가하면 추가한 순서대로 로그의 태그가 나열됩니다. 태그의 순서는 드래그하여 변경할 수 있습니다. 추가한 태그를 비활성화하면 비활성화한 태그는 로그의 태그에 노출되지 않습니다.
- 체크가 해제된 항목은 테이블에 표시되지 않습니다. 다음과 같이 **태그**를 해제할 경우 테이블에서 로그의 **태그**는 표시되지 않습니다.



> **컬럼 및 로그 표시 설정 적용 범위**

- **컬럼 설정**과 **로그 표시 상세 설정**은 **라이브 테일**, **로그 검색**, **로그 트렌드**에서 사용할 수 있습니다.
- 동일한 프로젝트 내 **라이브 테일**, **로그 검색**, **로그 트렌드**는 **컬럼 설정**과 **로그 표시 상세 설정**을 공유합니다.

로그 파일 다운로드

검색한 로그의 결과값을 CSV, TXT 파일 형식으로 수집된 로그를 다운로드 받을 수 있습니다.

1. 화면 중앙의 오른쪽의 **다운로드** 버튼을 클릭합니다.
2. **CSV 다운로드** 또는 **Log 다운로드**를 클릭해 원하는 형태의 파일을 다운로드 받습니다.

로그 트렌드

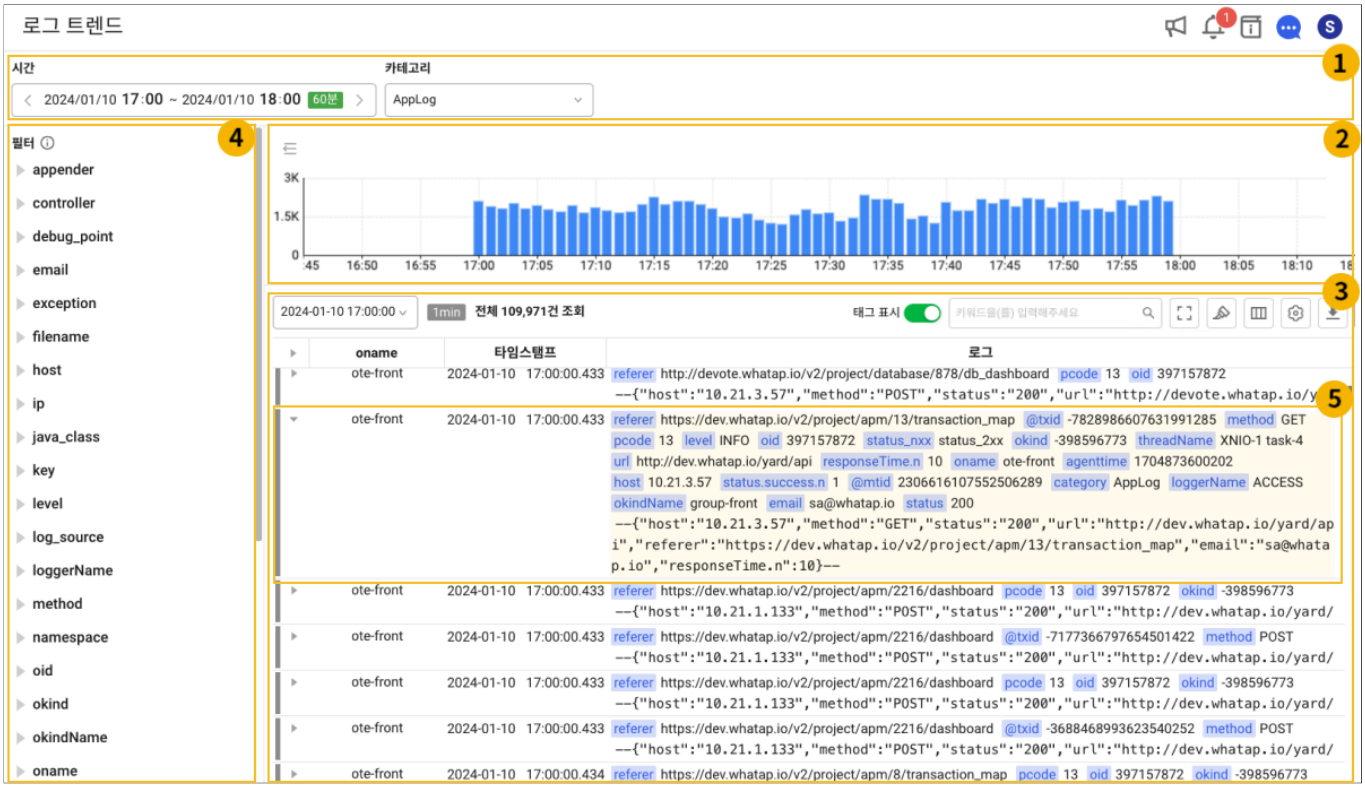
> ⓘ **로그 조회** 권한이 없을 경우 해당 메뉴에 진입할 수 없습니다.

홈 화면 > 프로젝트 선택 > **로그** > **로그 트렌드**

로그 트렌드에서 유형별로 분류된 로그의 발생 건수 추이를 통해 특정 에러 유형의 발생 패턴을 확인하고 시간별 상세 로그 데이터를 확인할 수 있습니다. 하이라이트 기능을 통해 원하는 로그를 빠르게 식별할 수 있습니다. 카테고리별로 수집된 로그의 추이를 조회할 수 있습니다.

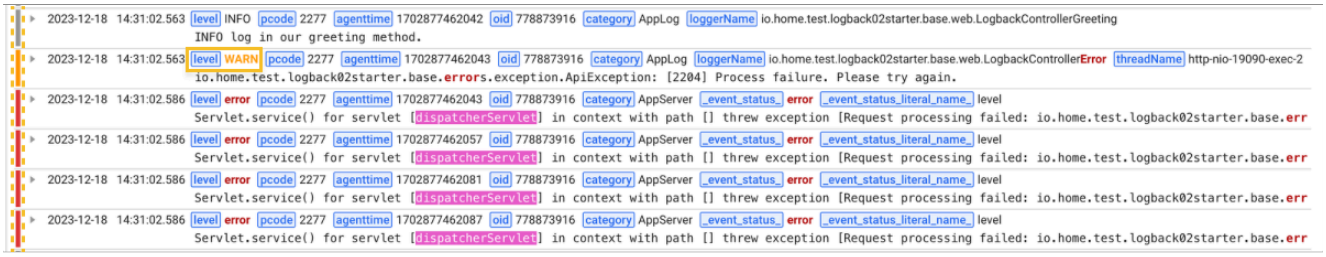
주요 용어는 다음과 같습니다.

- **Category**: 로그의 수집 및 조회 단위
- **Content**: 로그 메시지
- **Search Key**: 로그 파서 설정을 통해 생성되는 검색 키
- **Tag**: 수집된 로그를 검색할 수 있는 태그



데이터 조회하기

- 스크롤이 바닥에 닿으면 다음 데이터를 조회합니다.
- 1에서 탐색할 로그 데이터의 **시간**과 수집 단위인 **카테고리**를 지정합니다.
- **카테고리**를 변경하면 선택된 카테고리에 해당하는 로그를 조회합니다. 2 바 차트와 3 로그 테이블에서 확인할 수 있습니다.
- 2 바 차트의 막대를 클릭하면 막대의 시간 범위에 해당하는 로그를 조회합니다.
- 3 로그 테이블 상단 왼쪽에서 조회한 총 로그 개수를 확인할 수 있습니다.
- 3 로그 테이블 상단 오른쪽 **전체 화면** 아이콘을 선택하면 **로그**와 **수집 시간**을 전체 화면에서 확인할 수 있습니다.
- 4 사이드 메뉴에서 태그로 필터를 걸어서 로그를 확인할 수 있습니다. 검색 키는 2개까지 선택할 수 있고, 검색값은 복수 개 선택이 가능합니다.
- **에이전트 옵션이 설정된 경우** 로그 레벨을 수집해 로그 레벨 기준 색상이 다음과 같이 표시됩니다.



에이전트 옵션 설정

```
# whatap.conf
weaving=log4j-2.17
weaving=logback-1.2.8
```

- Java 에이전트 2.2.22 버전 이후부터 위빙 설정에 log4j-2.17 또는 logback-1.2.8 설정 시 사용할 수 있습니다. 에이전트 재시작이 필요합니다.
- 로그 레벨은 파싱된 키워드 중 `level`, `type` 기준으로 판별합니다. `level`, `type` 으로 파싱된 키가 존재하고 파싱 값이 FATAL, CRITICAL, ERROR, WARN, WARNING, INFO를 포함할 경우 로그 레벨 색상을 표시합니다.

> ⓘ 로그 테이블 컬럼 가장자리를 드래그해 컬럼 너비를 수정할 수 있습니다.

개인정보 조회

로그 개인정보 조회 권한이 있을 경우, **개인정보 표시** 토글 버튼 활성화를 통해 비식별화 처리한 개인정보를 표시할 수 있습니다.

> ⓘ
> - **로그 개인정보 조회** 권한에 대한 자세한 내용은 다음 문서를 참고하세요.
> - **로그 개인정보 비식별화** 기능에 대한 자세한 내용은 다음 문서를 참고하세요.

로그 Content 확인하기

Content는 로그 메시지입니다. 로그 컬럼의 첫 번째 줄은 로그의 파싱(parsing)된 키와 값이고 두 번째 줄은 Content입니다.

- 3 로그 테이블의 행(로그)마다 ▸ **더보기** 버튼이 있습니다. ▸ **더보기** 버튼을 선택하면 5처럼 해당 로그의 전체 content를 확인할 수 있습니다.

하이라이트 설정하기

원하는 키워드를 손쉽게 식별할 수 있도록 하이라이트 기능을 제공합니다. 아이콘을 클릭해 키워드를 입력 후 엔터를 눌러 추가하세요.

콘텐츠 하이라이트

로그의 콘텐츠 중 원하는 키워드를 손쉽게 식별하기 위해 하이라이트 기능을 제공합니다. 단일 또는 복수 키워드로 필터링할 수 있습니다.

1. 키워드 입력창에 하이라이트를 원하는 키워드를 입력하세요.
   - 예시, `select`
2. 로그 목록의 Content 내 검색한 키워드가 하이라이팅 됩니다.
   - **로그 전체 화면** 아이콘을 선택하면 **로그**와 **타임스탬프**를 전체 화면에서 확인할 수 있습니다.
   - 하이라이트된 키워드 간 이동은 단축키로 조작할 수 있습니다.
     - `Enter` : 다음 row로 이동
     - `Shift+Enter` : 이전 row로 이동
     - ∧ / ∨ 아이콘: row 이동
     - `Cmd` + `F` (Mac) / `Ctrl` + `F` (Windows): 키워드 검색

복수 키워드 조건

복수 키워드로 하이라이팅을 할 경우 다음과 같이 작성합니다.

입력 문자열	설명	결과
`a b c`	띄어쓰기로 각 키워드를 구분합니다.	a, b, c
`"Whatap is good."`	띄어쓰기를 키워드에 포함하고 싶은 경우 `'` 또는 `"` 로 감쌉니다.	Whatap is good.
`"Whatap\\ is good."`	`"`로 감싸진 키워드에서 `\`를 포함할 경우, `\\`로 입력해야합니다.	Whatap\ is good.

하이라이트 색상 설정

하이라이팅할 키워드 및 색상을 설정할 수 있습니다.

- 기본적으로 로그 레벨에 따른 하이라이팅(FATAL, ERROR, WARN, SEVERE, CRITICAL)이 적용되어 있습니다.
- 설정한 내용은 **프로젝트 단위**로 저장됩니다.

1. 아이콘을 클릭하세요.
2. **하이라이트** 창에서 추가적으로 색상 설정을 원하는 키워드를 입력창에 입력하세요.
3. 입력창 왼쪽 색상 클릭 시 색을 선택할 수 있습니다.

차트로 로그 조회하기

- 바 차트에서 a 원하는 시간 영역 바를 클릭하거나 드래그하여 해당 시간 범위의 로그를 확인할 수 있습니다.
- 바 차트 아래 로그 테이블 상단 왼쪽의 b 시간 선택 옵션을 이용해 다음과 같이 선택한 시간대에서 더 세분화해 로그를 검색할 수 있습니다.
  - 1min: interval (차트의 막대 사이 간격)
  - 시간 선택 옵션: 선택된 시간 범위를 6개의 구간으로 나눈 시간대

테이블 설정하기

컬럼 설정

컬럼을 추가하거나 순서를 설정할 수 있습니다. 3 영역 오른쪽에서 **컬럼 설정** 버튼을 클릭하세요.

컬럼 추가

태그를 선택하여 테이블에 컬럼을 추가할 수 있습니다. **log** 컬럼 선택을 해제할 경우 **로그 표시 상세 설정**을 확인할 수 없습니다. 반드시 한 개 이상의 컬럼을 선택하세요.

- **제공되는 컬럼**: 로그, 수집 시간, content(로그 메시지)

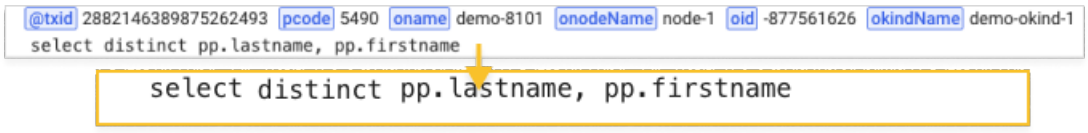
컬럼 순서 변경

컬럼을 추가하면 **표시 항목**에 해당 컬럼이 추가됩니다. 원하는 컬럼을 드래그하여 컬럼의 순서를 변경하세요.

로그 표시 상세 설정

3 영역 오른쪽에서 **로그 표시 상세 설정** 버튼을 선택하세요. 기본으로 **content**, **태그** 모두 체크가 되어있으며 두 가지 항목 모두 표시합니다. **content**와 **태그** 중 하나는 반드시 선택하세요.

체크가 해제된 항목은 테이블에 표시되지 않습니다. 다음과 같이 **태그**를 해제할 경우 테이블에서 로그의 **태그**는 표시되지 않습니다.



태그 관리 목록에 태그를 추가하면 추가한 순서대로 로그의 태그가 나열됩니다. 태그의 순서는 드래그하여 변경할 수 있습니다. 추가한 태그를 비활성화하면 비활성화한 태그는 로그의 태그에 노출되지 않습니다.

> **컬럼 및 로그 표시 설정 적용 범위**
>
> - **컬럼 설정**과 **로그 표시 상세 설정**은 **라이브 테일**, **로그 검색**, **로그 트렌드**에서 사용할 수 있습니다.
> - 동일한 프로젝트 내 **라이브 테일**, **로그 검색**, **로그 트렌드**는 **컬럼 설정**과 **로그 표시 상세 설정**을 공유합니다.

로그 파일 다운로드

검색한 로그의 결과값을 CSV, TXT 파일 형식으로 수집된 로그를 다운로드 받을 수 있습니다.

1. 화면 중앙의 오른쪽의 **다운로드** 버튼을 클릭합니다.
2. **CSV 다운로드** 또는 **Log 다운로드**를 클릭해 원하는 형태의 파일을 다운로드 받습니다.

로그 검색

> ⓘ **로그 조회** 권한이 없을 경우 해당 메뉴에 진입할 수 없습니다.

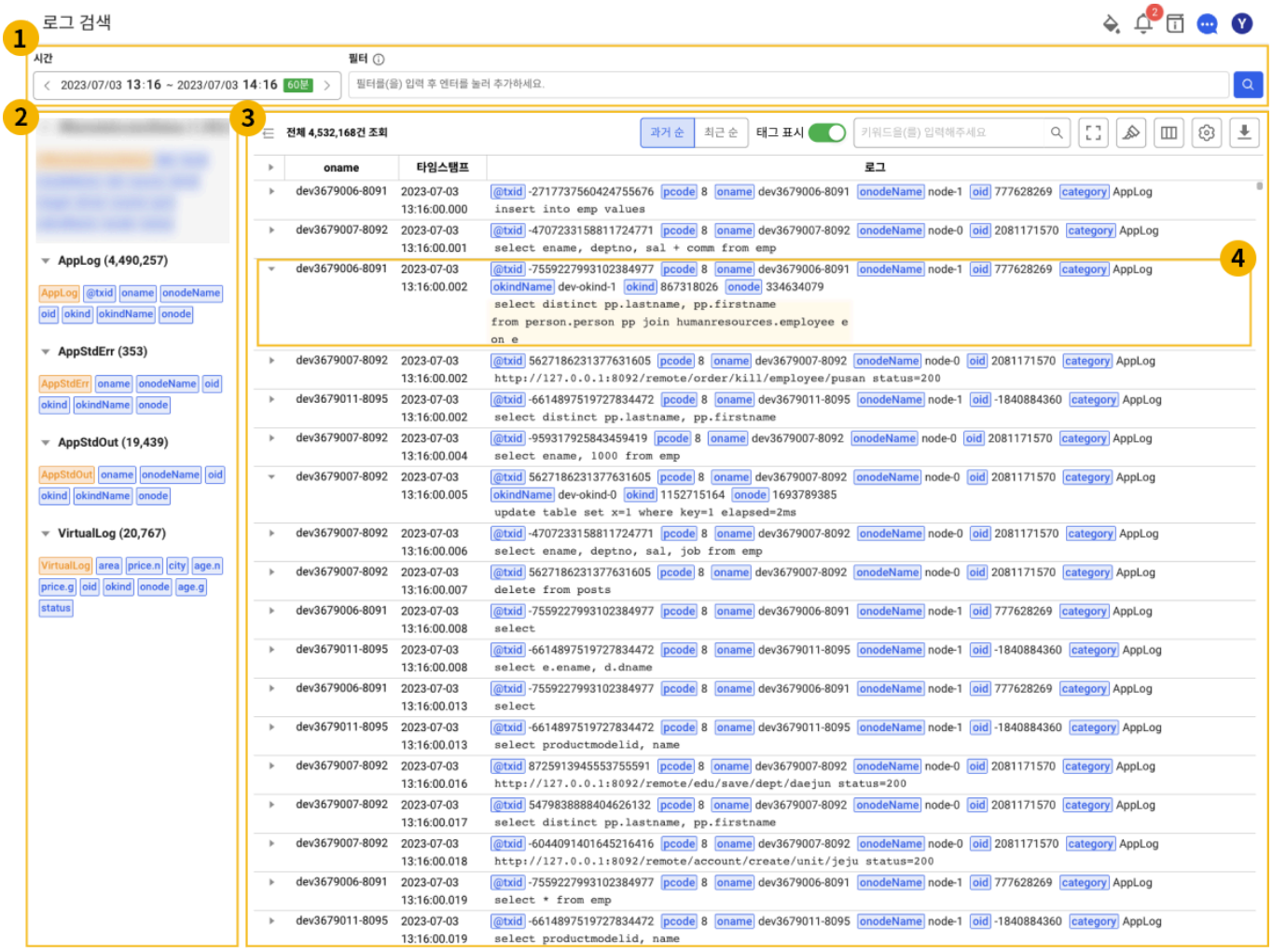
홈 화면 > 프로젝트 선택 > **로그** > **로그 검색**

로그 검색 메뉴에서 통합 수집된 대량의 로그를 다양한 조건으로 검색하고 사용자가 원하는 로그를 특정할 수 있습니다. 복수의 검색 조건을 파싱된 키와 밸류로 지정할 수 있어 원하는 조건에 일치하는 로그 데이터만 추출합니다.

동적 페이지로 검색된 로그 데이터를 정해진 라인 단위로 가져오며, 스크롤 등에 의해 하단에 닿으면 자동으로 다음 데이터를 가져와 표시합니다.

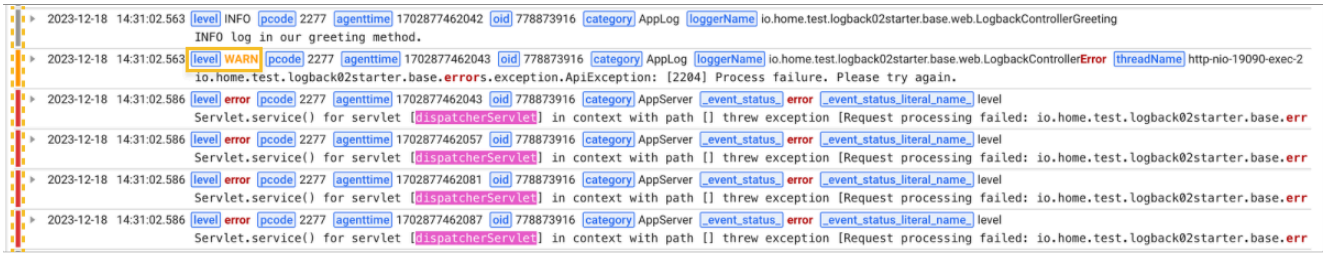
주요 용어는 다음과 같습니다.

- **Category**: 로그의 수집 및 조회 단위
- **Content**: 로그 메시지
- **Search Key**: 로그 파서 설정을 통해 생성되는 검색 키
- **Tag**: 수집된 로그를 검색할 수 있는 태그



데이터 조회하기

- 스크롤이 바닥에 닿으면 다음 데이터를 조회합니다. 한 번에 10,000개의 로그를 조회합니다.
- 3 로그 테이블 상단 왼쪽에서 조회한 총 로그 개수를 확인할 수 있습니다.
- 로그 데이터를 시간 순과 역순으로 조회할 수 있습니다. 3 로그 테이블 상단 오른쪽에서 **과거 순**과 **최근 순** 중 원하는 조회 방식을 선택하세요.
- 시간 범위 지정 후 **적용** 버튼을 선택 해 조회 시간을 설정하고 **검색** 버튼을 선택해 데이터를 조회합니다.
- 3 로그 테이블 상단 오른쪽 **로그 전체 화면** 아이콘을 선택하면 **로그**와 **타임스탬프**를 전체 화면에서 확인할 수 있습니다.
- **에이전트 옵션이 설정된 경우** 로그 레벨을 수집해 로그 레벨 기준 색상이 다음과 같이 표시됩니다.



에이전트 옵션 설정

```
# whatap.conf
weaving=log4j-2.17
weaving=logback-1.2.8
```

- Java 에이전트 2.2.22 버전 이후부터 위빙 설정에 log4j-2.17 또는 logback-1.2.8 설정 시 사용할 수 있습니다. 에이전트 재시작이 필요합니다.
- 로그 레벨은 파싱된 키워드 중 `level`, `type` 기준으로 판별합니다. `level`, `type` 으로 파싱된 키가 존재하고 파싱 값이 FATAL, CRITICAL, ERROR, WARN, WARNING, INFO를 포함할 경우 로그 레벨 색상을 표시합니다.

> ⓘ 로그 테이블 컬럼 가장자리를 드래그해 컬럼 너비를 수정할 수 있습니다.

로그 Content 확인하기

Content는 로그 메시지입니다. 로그 컬럼의 첫 번째 줄은 로그의 파싱(parsing)된 키와 값이고 두 번째 줄은 Content입니다.

- 3 로그 테이블의 행(로그)마다 ▸ **더보기** 버튼이 있습니다. ▸ **더보기** 버튼을 선택하면 4처럼 해당 로그의 전체 Content를 확인할 수 있습니다.
- 로그의 태그를 선택하면 **복사**, **검색**, **제외 검색**, **인접 로그**를 검색 할 수 있는 드롭다운 메뉴가 나타납니다.

필터

필터 적용

1 왼쪽 **시간 선택창**에서 시간 범위를 지정할 수 있습니다. 오른쪽에서 필터를 적용하면 입력한 조건에 맞는 로그를 필터링합니다. 복수의 필터를 입력할 수 있습니다. 필터의 태그가 같은 경우 OR(`||`)로, 그렇지 않은 경우는 AND(`&&`)로 적용됩니다.

입력 창에 값을 직접 입력하거나 **필터** 입력 창을 클릭해 필터를 지정할 수 있습니다. 필터 태그는 `검색 키` , `연산자` , `검색 값` 의 순서로 입력합니다. **검색** 버튼을 선택하면 필터가 적용된 데이터를 로그 목록에서 조회할 수 있습니다.

가이드 UI

다음과 같이 입력 창 아래 **가이드 UI**를 제공합니다. 입력 창 또는 필터 태그에 마우스 커서가 있을 때 ESC 키를 통해 **가이드 UI**를 닫을 수 있습니다.

검색 키, 연산자, 검색 값 입력

- `검색 키` 입력 시 일반 인덱스, 예약어 인덱스, 숫자만 입력할 수 있는 인덱스를 구분해 추천 값을 제공합니다
- `연산자` 입력 시 일반 인덱스 검색 키의 경우 `==` , `!=` 옵션을 하단에 안내합니다. 숫자만 입력할 수 있는 인덱스의 경우 `>` , `<` , `<=` , `>=` , `==` , `!=` 옵션을 제공합니다.
- `검색 값` 입력 시 일치 검색(`>` , `<` , `<=` , `>=` , `==`)일 때 파란색으로, 제외 검색(`!=`)일 때 붉은색으로 하이라이팅합니다. 대소문자 구분 옵션을 활용해 검색할 수 있습니다.

> ⓘ
> - 필터 태그가 2줄 이상 길어지는 경우 ∧ **접기** 아이콘을 선택해 접어둘 수 있습니다.
> - 필터 태그 입력 후 입력 창에서 Shift와 Tab 키를 동시에 사용하여 이전 필터 태그로 이동할 수 있습니다.
> - 필터 태그 입력 후 입력 창에서 Tab 키를 통해 다음 필터 태그로 이동할 수 있습니다.

필터 태그 추가

입력 창에 텍스트 입력 후 Enter 또는 Tab 키를 누르거나, 입력창 하단의 가이드 UI에서 추천 값을 클릭하거나 방향키로 선택해 추가할 수 있습니다.

필터 태그 제거

Backspace 키 또는 태그의 **X** 아이콘을 눌러 태그를 삭제할 수 있습니다. 입력창 오른쪽의 **X** 아이콘을 클릭해 모든 태그를 한 번에 삭제할 수 있습니다.

필터 즐겨찾기

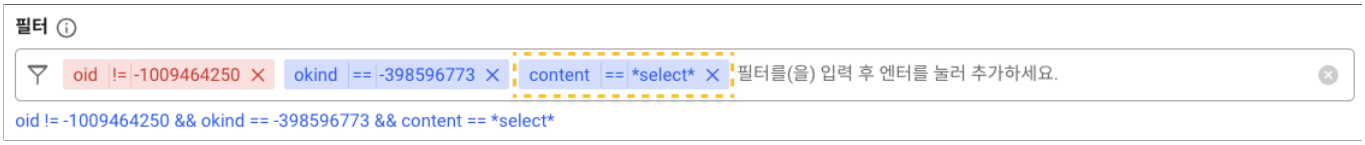
필터 **즐겨찾기** 기능을 제공합니다. 입력 창 오른쪽의 ☆ **즐겨찾기** 아이콘을 선택하여 원하는 필터 검색 조건을 즐겨찾기에 **추가**, **제거** 및 기존 즐겨찾기를 선택할 수 있습니다. **즐겨찾기**는 최대 50개까지 저장됩니다.

필터 적용 예외 상황

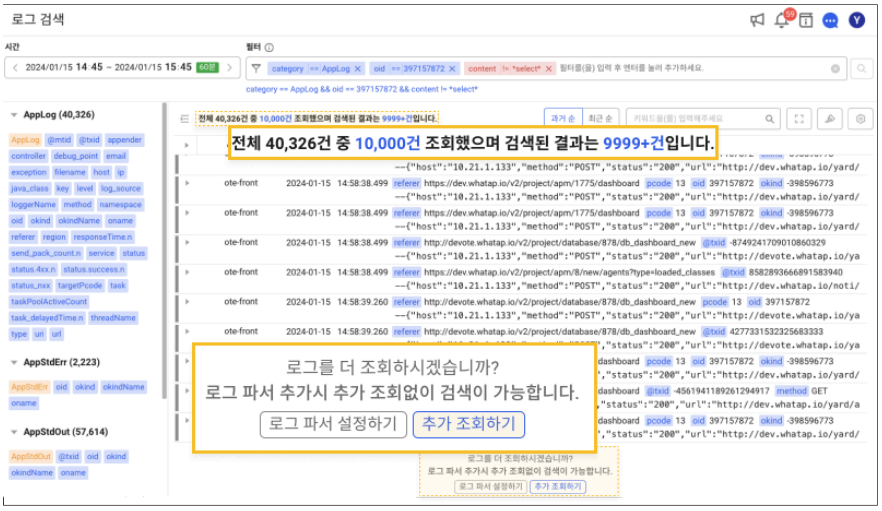
- 숫자만 입력할 수 있는 인덱스(.n 으로 끝나는 검색 키)를 입력한 태그에서 검색 값 은 숫자만 입력할 수 있습니다.
- 중복된 검색 키 , 검색 값 은 입력할 수 없습니다.
- 검색 키 , 검색 값 중 하나라도 없는 태그가 존재할 때 검색할 수 없습니다. 유효하지 않는 태그의 경우 회색으로 표시합니다.

미파싱 키워드 필터 적용

로그에서 파싱되지 않은 즉 인덱스가 생성되지 않은 키워드를 포함한 로그를 조회할 수 있습니다. 이 경우 지정 범위 내 모든 로그를 Full Scan합니다. 그렇기 때문에 인덱스가 생성된 키와 비교해 검색 속도가 다소 떨어질 수 있습니다. 정형화된 로그 데이터의 경우 **로그 파서 설정**을 통해 인덱스 키 값을 활용해 검색하는 것을 권장합니다.



1. **카테고리**를 선택하세요. 카테고리 지정이 필수적입니다.
2. **필터** 입력창에 content 기준 띄어쓰기 후 검색을 원하는 키워드를 입력하세요.
   예시, content *select*
3. **검색** 버튼을 클릭해 로그를 조회하세요. 전체 로그 중 일부 먼저 조회합니다. 1회당 검색 결과는 최대 1만 건입니다.
4. 스크롤을 내려 하단의 **추가 조회하기** 버튼 선택 시 추가 조회할 수 있습니다.



ⓘ 전체 로그 중 서버 조회 범위 당 1만 건씩 조회합니다. 서버 조회 범위의 경우 기본 20만 건이지만 전체 로그 양에 따라 비율이 달라질 수 있습니다.
파서 설정에 대한 자세한 내용은 다음 문서를 참조하세요.

필터 수정

필터에 값을 입력한 뒤 입력한 값을 클릭하면 해당 값을 수정할 수 있습니다.

- 입력 창에 텍스트 재입력해 수정할 수 있습니다.
- 입력 창 아래 **가이드 UI**를 통해 추천 값을 선택해 수정할 수 있습니다.

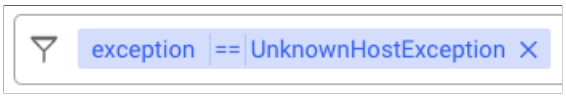
검색 키(Search Key)

검색 키는 로그 데이터 내에서 원하는 특정 값에 접근하기 위한 식별자를 의미합니다. 검색 키에 해당하는 실제 데이터가 검색 값입니다. 왼쪽 2 영역에 있는 태그는 카테고리별로 파싱(parsing) 된 검색 키입니다. 태그를 선택하여 필터를 입력할 수 있습니다. 주황색 태그는 카테고리, 파란색 태그는 검색 키입니다.

예를 들어 2 영역의 AppLog와 AppStdOut은 카테고리, 그 아래 oid와 같은 태그는 파싱(parsing) 된 검색 키입니다. 검색 키는 **로그** > **로그 설정**의 **로그 파서 설정** 탭에서 파싱 로직을 등록해 설정할 수 있습니다.

필터 입력 문법

태그는 검색 키와 검색 값으로 구성되어있습니다. 다음의 예시에서 검색 키는 `exception` , 검색 값은 `UnknownHostException` 입니다. 해당 예시는 수집한 로그 데이터 중 IP 주소와 도메인 주소가 매칭되지 않아 서버를 호스트에 연결할 수 없을 경우 발생하는 예외(`UnknownHostException`)가 포함된 로그 데이터를 조회합니다.



검색 키 종류

검색 키 종류	검색 키 포멧	의미	검색 키와 검색 값 예시	검색 예시
문자열 키워드	keyword	파일 이름	- 키: fileName - 값: /data/whatap/logs/yard.log	fileName:/data/whatap/logs/yard.log
숫자 키워드	keyword.n	응답시간	- 키: response_time.n - 값: 2945	response_time.n>=2945
예약어 키워드 (사전 정의 키워드)	@keyword	트랜잭션 ID	- 키: @txid - 값: 85459614215434144	-

공통 문법

문법 종류	설명	예시
`==searchValue`	검색 값과 일치하는 로그를 검색합니다.	`exception==RuntimeExceptionexception`
`!=searchValue`	검색 값을 제외한 로그를 검색합니다.	`exception!=RuntimeException`
`*searchValue`	검색 값으로 끝나는 로그를 검색합니다.	`word==*hello`
`searchValue*`	검색 값으로 시작하는 로그를 검색합니다.	`word==hello*`
`*searchValue*`	검색 값으로 중간에 포함된 로그를 검색합니다.	`word==*hello*`
`*search*Value*`	검색 값으로 포함된 로그를 검색합니다.	`word==*he*llo*`
`re:{regex}`	정규표현식에 매칭되는 로그를 검색합니다.	`caller==re:^i\.w\.a\.w\.s\.v\.r\.`

문법 종류	설명	예시
**	검색 키에 해당하는 모든 로그를 검색합니다.	

검색 키가 숫자 키워드(keyword.n)인 경우 문법

다음의 문법은 검색 키가 keyword.n 형식인 경우에만 지원합니다.

- 검색 값으로는 숫자만 올 수 있습니다.
- .n 키워드의 값에는 prefix를 붙이지 않습니다. .n 이 아닌 키워드는 모두 prefix를 붙여야합니다.

  예, +>searchValue 는 유효하지 않습니다.

문법 종류	설명	예시
>searchValue	검색 값보다 큰 값이 포함된 로그를 조회합니다.	response_time.n>3000
>=searchValue	검색 값보다 크거나 같은 값이 포함된 로그를 조회합니다.	response_time.n>=3000
==searchValue	검색 값보다 같은 값이 포함된 로그를 조회합니다.	response_time.n==3000
!=searchValue	검색 값보다 다른 값이 포함된 로그를 조회합니다.	response_time.n!=3000
<searchValue	검색 값보다 작은 값이 포함된 로그를 조회합니다.	response_time.n<3000
<=searchValue	검색 값보다 작거나 같은 값이 포함된 로그를 조회합니다.	response_time.n<=3000

개인정보 조회

로그 개인정보 조회 권한이 있을 경우, **개인정보 표시** 토글 버튼 활성화를 통해 비식별화 처리한 개인정보를 표시할 수 있습니다.

- **로그 개인정보 조회** 권한에 대한 자세한 내용은 다음 문서를 참고하세요.
- **로그 개인정보 비식별화** 기능에 대한 자세한 내용은 다음 문서를 참고하세요.

로그 태그 옵션

로그 태그 선택 시 드롭다운 메뉴가 나타납니다. **검색**, **제외 검색**, **인접 로그**, **트랜잭션 정보** 옵션을 확인할 수 있습니다.

로그 태그 옵션 메뉴	설명
복사	태그에 해당하는 로그를 복사합니다.
검색	**필터**에 해당 태그가 포함('==') 조건으로 입력됩니다.
제외 검색	**필터**에 해당 태그가 제외('!=') 조건으로 입력됩니다.
인접 로그	**인접 로그** 상세 창에서 선택한 로그의 서버를 대상으로 선택한 로그와 인접한 시간대의 로그를 조회합니다. 시간을 선택해 인접한 시간대의 로그를 조회할 수 있습니다. 기준 로그는 파란색 바탕으로 표시됩니다.
트랜잭션 정보	로그에 `@txid` 및 `@mtid` 태그가 포함된 경우, **트랜잭션 정보**를 클릭하면 **트랜잭션 정보** 상세창에서 정보를 확인할 수 있습니다.

콘텐츠 하이라이트

로그의 콘텐츠 중 원하는 키워드를 손쉽게 식별하기 위해 하이라이트 기능을 제공합니다. 단일 또는 복수 키워드로 필터링할 수 있습니다.

1. 키워드 입력창에 하이라이트를 원하는 키워드를 입력하세요.

- 예시, `select`

2. 로그 목록의 Content 내 검색한 키워드가 하이라이팅 됩니다.
   - **로그 전체 화면** 아이콘을 선택하면 **로그**와 **타임스탬프**를 전체 화면에서 확인할 수 있습니다.
   - 하이라이트된 키워드 간 이동은 단축키로 조작할 수 있습니다.
     - `Enter` : 다음 row로 이동
     - `Shift+Enter` : 이전 row로 이동
     - ∧ / ∨ 아이콘: row 이동
     - `Cmd` + `F` (Mac) / `Ctrl` + `F` (Windows): 키워드 검색

복수 키워드 조건

복수 키워드로 하이라이팅을 할 경우 다음과 같이 작성합니다.

입력 문자열	설명	결과
`a b c`	띄어쓰기로 각 키워드를 구분합니다.	a, b, c
`"Whatap is good."`	띄어쓰기를 키워드에 포함하고 싶은 경우 `''` 또는 `""`로 감쌉니다.	Whatap is good.
`"Whatap\\ is good."`	""로 감싸진 키워드에서 `\`를 포함할 경우, `\\`로 입력해야합니다.	Whatap\ is good.

하이라이트 색상 설정

하이라이팅할 키워드 및 색상을 설정할 수 있습니다.

- 기본적으로 로그 레벨에 따른 하이라이팅(FATAL, ERROR, WARN, SEVERE, CRITICAL)이 적용되어 있습니다.
- 설정한 내용은 **프로젝트 단위**로 저장됩니다.

1. 아이콘을 클릭하세요.
2. **하이라이트** 창에서 추가적으로 색상 설정을 원하는 키워드를 입력창에 입력하세요.
3. 입력창 왼쪽 색상 클릭 시 색을 선택할 수 있습니다.

테이블 설정하기

컬럼 설정

컬럼을 추가하거나 순서를 설정할 수 있습니다. 3 영역 오른쪽에서 **컬럼 설정** 버튼을 클릭하세요.

컬럼 추가

태그를 선택하여 테이블에 컬럼을 추가할 수 있습니다. **log** 컬럼 선택을 해제할 경우 **로그 표시 상세 설정**을 확인할 수 없습니다. 반드시 한 개 이상의 컬럼을 선택하세요.

- **제공되는 컬럼**: 로그, 수집 시간, content(로그 메시지)

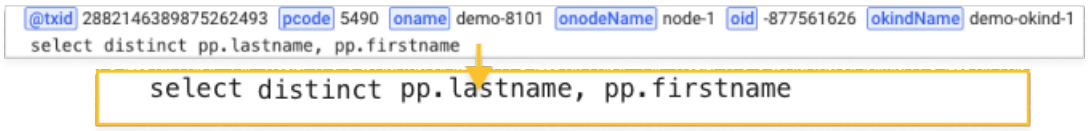
컬럼 순서 변경

컬럼을 추가하면 **표시 항목**에 해당 컬럼이 추가됩니다. 원하는 컬럼을 드래그하여 컬럼의 순서를 변경하세요.

로그 표시 상세 설정

3 영역 오른쪽에서 **로그 표시 상세 설정** 버튼을 선택하세요. 기본으로 **content**, **태그** 모두 체크가 되어있으며 두 가지 항목 모두 표시합니다. **content**와 **태그** 중 하나는 반드시 선택하세요.

체크가 해제된 항목은 테이블에 표시되지 않습니다. 다음과 같이 **태그**를 해제할 경우 테이블에서 로그의 **태그**는 표시되지 않습니다.



태그 관리 목록에 태그를 추가하면 추가한 순서대로 로그의 태그가 나열됩니다. 태그의 순서는 드래그하여 변경할 수 있습니다. 추가한 태그를 비활성화하면 비활성화한 태그는 로그의 태그에 노출되지 않습니다.

> **컬럼 및 로그 표시 설정 적용 범위**
>
> - **컬럼 설정**과 **로그 표시 상세 설정**은 **라이브 테일**, **로그 검색**, **로그 트렌드**에서 사용할 수 있습니다.

- 동일한 프로젝트 내 **라이브 테일**, **로그 검색**, **로그 트렌드**는 **컬럼 설정**과 **로그 표시 상세 설정**을 공유합니다.

로그 파일 다운로드

검색한 로그의 결과값을 CSV, TXT 파일 형식으로 수집된 로그를 다운로드 받을 수 있습니다.

1. 화면 중앙의 오른쪽의 **다운로드** 버튼을 클릭합니다.
2. **CSV 다운로드** 또는 **Log 다운로드**를 클릭해 원하는 형태의 파일을 다운로드 받습니다.

로그 설정

로그 설정의 상단의 탭에서 로그 모니터링을 설정할 수 있습니다. 에이전트 설정 확인, 로그 모니터링의 활성화 여부 결정, 로그 데이터의 유지 기간 및 조회 비밀번호 설정, 로그 파서 등록, 빠른 인덱스 설정, 개인정보 비식별화 관리 등을 할 수 있습니다.

> **권한**
>
> - **로그 모니터링 활성화**을 사용하려면 **프로젝트 수정** 권한이 필요합니다.
> - **로그 편집** 권한이 있으면 **로그 모니터링 활성화** 외 **로그 설정** 메뉴를 수정할 수 있습니다.

로그 모니터링 시작하기

1. **로그** > **로그 설정**의 **로그모니터링 시작하기** 탭을 클릭하세요.
2. **가이드 보기** 아이콘과 **요금제 보기** 버튼을 클릭하면 안내 화면으로 이동합니다.

에이전트 설정 및 로그 모니터링 활성화

로그 모니터링 데이터 설정

로그 사용량을 확인할 수 있습니다. 또한 **로그 보존 기간** 및 **로그 조회 비밀번호** 설정을 변경할 수 있습니다.

로그 조회 비밀번호

보안을 강화하기 위해 **로그 조회 비밀번호**를 설정하세요. 로그 조회 비밀번호 지정은 선택 사항입니다. 로그 조회 비밀번호를 사용 중이라면 로그 화면 진입 시 반드시 비밀번호를 입력해야 합니다.

> **비밀번호 분실**
>
> 로그 편집 권한이 있는 사용자는 **로그 설정**에서 새 비밀번호로 변경할 수 있습니다.

로그 보존 기간

공통으로 적용할 기본(default) 데이터 보존 기간입니다.

- 최소 1일부터 최대 40일까지 입력할 수 있습니다. 미지정 시 기본값은 1일입니다. 로그 보존 기간 선택 옵션 외에 사용자가 원하는 기간을 입력할 수 있습니다.
- **로그 사용량** 목록에서 별도로 설정하지 않으면 이 로그 데이터 보존 기간이 기본적으로 적용됩니다. **로그 사용량** 목록에서 카테고리별 데이터 보존 기간을 설정하고 **초기화** 버튼을 클릭하면 기본 데이터 보존 기간으로 초기화됩니다.

로그 사용량

로그 사용량 목록에서 카테고리별 로그 데이터 보존 기간을 지정할 수 있습니다. 로그 수는 해당 기간 동안 쌓인 로그 건수를 의미합니다. 예를 들어 **금일 로그수**는 하루 동안 쌓인 로그 건수, **예상 로그수**는 데이터 보관일에 금일 로그 수를 곱한 로그 건수를 의미합니다.

로그 데이터 보존 기간을 다음과 같이 지정할 수 있습니다. 기간 지정에 따라 오래된 데이터를 삭제해 공간을 확보할 수 있습니다.

- **트라이얼 프로젝트**

  데이터 보존 기간으로 1일, 2일, 3일을 선택할 수 있습니다.

- **유료 프로젝트**

  데이터 보존 기간으로 1일, 2일, 3일, 4일, 5일, 6일, 7일, 10일, 30일, 40일을 선택할 수 있습니다.

- **저장량 기준 과금**

  데이터 보존 기간에 따라 비용이 달라집니다.

  > 예, 일 평균 200만 로그 건수가 쌓이고 데이터 보존 기간을 3일로 지정한 경우라면 평균 600만 로그 건수가 수집 서버에 유지되고 과금 대상이 됩니다.

이벤트 설정

홈 화면 > 프로젝트 선택 > **경고 알림** > **이벤트 설정**

이벤트 설정에서 모니터링 대상 시스템 또는 애플리케이션에서 발생할 수 있는 이상 징후를 기준으로 이벤트를 설정하고 관리하는 기능을 제공합니다. 설정된 조건이 충족되면 이메일, SMS, 메신저, 앱 푸시 등 원하는 채널로 알림을 받을 수 있습니다. 모니터링 목적에 따라 이벤트를 구성할 수 있습니다.

이벤트 설정 개요

Java 경고 알림을 위한 이벤트 설정 메뉴 구성과 사용법을 안내합니다.

이벤트 수신 설정

Java 경고 이벤트를 수신할 채널과 대상을 설정하는 방법을 안내합니다.

이벤트 기록

Java에서 발생한 이벤트 이력을 조회하고 추이를 모니터링하는 방법을 안내합니다.

참고

- 이벤트 설정 시 적정 임계값을 사용하여 불필요한 이벤트 발생을 방지하세요.
- 알림 설정 시 해소 알림(문제 해결 시 알림) 여부도 함께 고려하면 운영 대응에 도움됩니다.

공통 기능

모든 이벤트 설정을 **JSON 또는 Excel 형식으로 다운로드**할 수 있습니다. 같은 유형의 제품 설정을 공유하면 반복되는 이벤트 설정 작업을 줄일 수 있습니다.

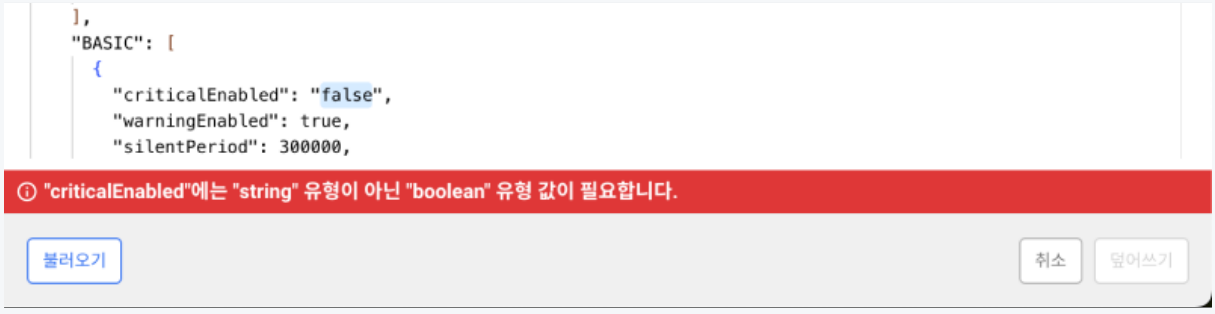
- **JSON 일괄 수정**: 모든 이벤트 조건을 JSON 형식으로 일괄 수정할 수 있습니다.
- **JSON 일괄 다운로드**: 현재 등록된 이벤트 조건을 JSON 파일로 저장할 수 있습니다.
- **Excel 일괄 다운로드**: 이벤트 목록을 Excel 파일로 내려받아 외부 보고나 백업에 활용할 수 있습니다.

JSON 일괄 수정

1. 화면 상단의 [JSON 일괄 수정] 버튼을 클릭하세요.
2. **JSON 일괄 다운로드** 기능을 통해 다운로드 받은 JSON 파일을 가져오거나 직접 수정하세요.
3. [덮어쓰기] 버튼을 클릭해 저장합니다.
4. 확인 메시지가 나타나면 [덮어쓰기] 버튼을 클릭하세요.

덮어쓰기 후 이전의 이벤트 설정으로 복구할 수 없습니다.

- `productType` 또는 `platform` 속성의 값을 임의로 수정하면 다른 프로젝트에서 이벤트가 정상 동작하지 않을 수 있습니다.
- JSON 편집창은 JSON 검증(Validation) 기능이 내장되어 있습니다. 형식이나 값이 올바르지 않은 경우, 편집창 아래에 오류 메시지가 표시되며 [덮어쓰기] 버튼은 비활성화됩니다.



JSON 데이터 구조

```
{
  "productType": "{productType}",
  "platform": "{platform}",
  "events": {
    "HITMAP": {
      ...
    },
    "LOG_REALTIME": [
      {...}
    ],
    "METRICS": [
      {...}
    ],
    "COMPOSITE_LOG": [
      {...}
    ],
    "BASIC": [
      {...}
    ],
    "COMPOSITE_METRICS": [
      {...}
    ],
    "ANOMALY": [
      {...}
    ]
  }
}
```

JSON 필드	설명
productType	제품 타입(예: APM, Infra 등)
platform	플랫폼 정보(예: java, node 등)
events	이벤트 데이터
ㄴ HITMAP	히트맵 패턴

JSON 필드	설명
ㄴ LOG_REALTIME	실시간 로그 이벤트
ㄴ METRICS	메트릭스
ㄴ COMPOSITE_LOG	복합 로그 이벤트
ㄴ BASIC	기본 이벤트
ㄴ COMPOSITE_METRICS	복합 메트릭스
ㄴ ANOMALY	이상치 탐지

JSON 일괄 다운로드

화면 상단의 [JSON 일괄 다운로드] 버튼을 클릭하세요.

- JSON 파일 이름은 integrated-event- `YYYYMMDD` .json 형식입니다.
- JSON 데이터의 구조와 `events` 속성의 하위 속성은 제품(`platform`) 또는 제품 유형(`productType`)에 따라 다를 수 있습니다.

JSON 데이터의 속성 중 `productType` 또는 `platform` 속성의 값을 임의로 수정하면 다른 프로젝트에서 이벤트가 동작하지 않을 수 있습니다.

Excel 일괄 다운로드

화면 상단의 [Excel 일괄 다운로드] 버튼을 클릭하세요.

이벤트 수신 설정

사용자별 이벤트 수신 설정

> **사용자별 이벤트 수신 설정** 의 다른 멤버의 정보는 **멤버 관리** 권한을 가진 멤버만 확인할 수 있습니다. 그 외 멤버에게는 필수 정보가 마스킹 처리됩니다.
>
> - 이벤트 알림의 일괄 수신설정 및 접근 설정을 위한 모바일 기기 관리는 **계정 정보**에서 할 수 있습니다.
> - 멤버 권한에 대한 자세한 설명은 다음 문서를 참고하세요.

수신 수단 설정

이메일, **SMS**, **WhatsApp**, **모바일** 알림을 제공합니다.

사용자별 이벤트 수신 설정 섹션에서 원하는 알림 수신 수단의 체크 박스를 선택하면 경고 알림을 받을 수 있습니다. 알림 수신 수단의 체크 박스를 해제하면 경고 알림을 보내지 않습니다.

> 프로젝트 최고 관리자를 제외한 모든 사용자는 자신의 수신 설정만 변경할 수 있습니다.

이메일 알림

이메일 알림은 회원 가입 시 입력한 이메일 주소로 알림을 보냅니다.

SMS 알림

SMS로 알림을 수신할 수 있습니다.

> SMS 알림 기능은 유료 프로젝트에 한정됩니다.

1. 화면 오른쪽 위의 **자신의 프로필 아이콘** > **계정 관리**로 이동하세요.
2. **계정 정보**의 **사용자 전화번호** 섹션에서 [일반 휴대전화] 버튼을 클릭하세요.
3. **전화번호**에 인증번호를 수신을 전화번호를 입력하세요.
    - SMS를 알림으로 수신할 수 있는 전화번호는 **한국의 휴대전화 번호만 등록**할 수 있습니다.
4. [저장] 버튼을 클릭하세요.

> **등록한 전화번호 변경 및 삭제**
>
> - 전화번호를 변경할 경우, [변경] 버튼을 클릭하세요. SMS 인증이 필요합니다.
> - 전화번호를 삭제할 경우, [삭제] 버튼을 클릭하세요. 전화번호 삭제 시 SMS로 수신받는 알림은 수신할 수 없습니다.

WhatsApp 알림

WhatsApp을 통해 알림을 수신할 수 있습니다.

> WhatsApp 알림 기능은 유료 프로젝트에 한정됩니다.

1. 화면 오른쪽 위의 **자신의 프로필 아이콘** > **계정 관리**로 이동하세요.
2. **계정 정보**의 **사용자 전화번호** 섹션에서 [WhatsApp] 버튼을 클릭하세요.
3. 인증번호를 받을 **전화번호**를 입력하세요.
4. [인증번호 전송] 버튼을 클릭하세요.
5. WhatsApp 애플리케이션으로 전송된 인증번호 6자리를 입력하세요.
    - 10분 내로 인증번호를 받지 못했다면, [인증번호 재전송] 버튼을 클릭하세요.
6. [인증하기] 버튼을 클릭하세요.

모바일 알림

모바일 알림 수신이 필요한 사용자는 모바일 기기에 와탭 애플리케이션을 설치한 후 로그인해야 합니다.

- 와탭 애플리케이션 다운로드 - **안드로이드**
- 와탭 애플리케이션 다운로드 - **iOS**

반복 알림(에스컬레이션) 설정

경고 알림 발생 시간으로부터 알림 발생 상황이 해소되지 않을 경우 최초 알림 발생 시각으로부터의 알림 반복 간격을 설정할 수 있습니다. 위험(Critical) 레벨의 알림만 적용됩니다.

예를 들어, 경고 알림 발생 시간으로부터 0분(즉시), 1시간 후, 1일 후에 경고 알림을 반복하려면 `0,1H,1D` 를 **반복 알림(에스컬레이션)** 컬럼 항목에 입력하세요.

1. **경고 알림** > **이벤트 수신 설정**의 **사용자별 이벤트 수신 설정** 섹션으로 이동하세요.
2. **반복 알림(에스컬레이션)** 항목에서 설정하세요.
   - **M**: 분, **H**: 시간, **D**: 일, 단위를 생략하면 분 단위로 시간을 설정합니다.
   - 숫자 또는 숫자+단위(**M**, **H**, **D**)로 입력하세요. 입력이 올바르지 않으면 메시지가 표시됩니다.
3. [저장] 버튼을 클릭하세요. 저장하지 않으면 설정되지 않습니다.

이벤트 수신 태그

이벤트 설정 시 이벤트 수신 태그를 선택하여 해당 태그를 가진 프로젝트 멤버와 3rd-party 플러그인에 알림을 전송할 수 있습니다. 이벤트 수신 태그를 설정하지 않으면 전체 멤버에게 경고 알림이 전송됩니다.

이벤트 수신 태그 추가/수정/삭제하기

> 이벤트에 적용 중인 **이벤트 수신 태그** 항목은 삭제할 수 없습니다.

1. **경고 알림** > **이벤트 수신 설정**으로 이동하세요.
2. **사용자별 이벤트 수신 설정** 섹션의 사용자 목록에서 **+ 태그 추가** 또는 **+** 를 클릭하세요.
3. **이벤트 수신 태그** 창에서 **+ 새 태그 생성**을 클릭하세요.
4. **태그 생성** 창에서 태그 이름을 입력하고, 색상을 선택한 후, [태그 생성] 버튼을 클릭해 태그를 생성하세요.
5. 생성된 태그는 태그 목록의 아이콘을 클릭해 수정 또는 삭제하세요.
6. **이벤트 수신 태그** 창의 **태그 목록**에서 원하는 태그를 선택하면 적용됩니다.

이벤트 수신 태그 해제하기

1. **경고 알림** > **이벤트 수신 설정**의 **사용자별 이벤트 수신 설정** 섹션으로 이동하세요.
2. 이벤트 수신 태그를 해제할 사용자의 **이벤트 수신 태그** 항목의 **+** 를 클릭하세요.
3. **이벤트 수신 태그** 창에서 해제할 태그의 X 를 클릭하면 해당 태그가 해제됩니다.

수신 태그 미설정 알림 받기

이벤트 설정 시 이벤트 수신 태그를 선택하여 해당 태그를 가진 프로젝트 멤버와 3rd-party 플러그인에 알림을 전송할 수 있습니다.

- 이벤트 수신 태그가 설정되지 않은 경고 알림을 받으려면 **수신 태그 미설정 알림 받기** 옵션을 선택하세요.
- 이벤트 수신 태그가 설정된 경고 알림만 받고 싶다면 선택을 해제하세요.

> ⓘ 모든 경고 알림을 받지 않으려면 해당 옵션을 해제하고 선택한 **이벤트 수신 태그**가 없어야 합니다.

이벤트 수신 모드 선택하기

사용자별 이벤트 수신 설정 섹션에서 멤버 또는 3rd 파티 채널별로 이벤트 알림 수신 범위를 다음 네 가지 모드 중에서 선택할 수 있습니다. 수신 모드는 각 멤버 또는 채널 행의 **이벤트 수신 태그** 열에서 라디오 버튼으로 선택하세요.

수신 모드	설명
모든 알림 받기	이벤트 수신 태그와 무관하게 모든 경고 알림 수신
선택한 태그만 받기	선택한 이벤트 수신 태그를 가진 경고 알림만 수신. 태그를 최소 한 개 이상 선택
선택한 태그만 제외하기	선택한 이벤트 수신 태그를 가진 경고 알림만 제외하고 나머지 전부 수신. 제외 태그를 최소 한 개 이상 선택
모든 알림 받지 않기	어떤 이벤트 알림도 받지 않음. 선택해 둔 태그는 삭제되지 않고 그대로 보존

- 이벤트 수신 태그가 지정되지 않은 경고 알림까지 받으려면 태그 선택 영역에서 **미분류** 가상 태그를 선택하세요. **미분류** 태그는 일반 태그와 동일하게 선택하거나 해제할 수 있으며, 태그 자체를 수정하거나 삭제할 수는 없습니다.

> ⓘ **모든 알림 받지 않기** 모드에서는 해당 멤버 또는 채널이 이벤트 알림을 전혀 받지 않습니다. 이때 선택해 둔 태그는 삭제되지 않고 보존되므로, 다시 **선택한 태그만 받기** 모드로 전환하면 이전에 선택한 태그 설정을 그대로 사용할 수 있습니다.

선택한 태그만 받기

1. **사용자별 이벤트 수신 설정** 섹션에서 설정할 멤버 또는 3rd 파티 채널 행의 **이벤트 수신 태그** 열을 클릭하세요.
2. **선택한 태그만 받기** 라디오 버튼을 선택하세요.
3. 태그 선택 영역에서 알림을 받을 태그를 하나 이상 선택하세요.

> ⓘ **선택한 태그만 받기** 모드에서 태그를 하나도 선택하지 않았을 때 나타나는 완전 차단 경고는 **선택한 태그만 제외하기** 모드에는 표시되지 않습니다. 제외 태그가 없는 상태는 알림을 전부 차단하는 상태가 아니라 전부 수신하는 상태이기 때문입니다.

선택한 태그만 제외하기

경고 알림을 대부분 받되 특정 분류만 걸러 내고 싶을 때 사용합니다. 받을 태그를 하나씩 고르는 **선택한 태그만 받기**와 방향이 반대입니다.

1. **사용자별 이벤트 수신 설정** 섹션에서 설정할 멤버 또는 3rd 파티 채널 행의 **이벤트 수신 태그** 열을 클릭하세요.
2. **선택한 태그만 제외하기** 라디오 버튼을 선택하세요.
3. 아래 태그 선택 영역에서 알림을 받지 않을 태그를 하나 이상 선택하세요.
    - 이벤트에 붙은 태그와 제외 태그가 하나라도 겹치면 해당 경고 알림은 전송되지 않습니다. 겹치는 태그가 없는 알림은 모두 수신합니다.
    - 제외 태그는 최소 한 개 이상 선택해야 합니다. 제외 태그가 하나도 없는 상태는 **모든 알림 받기**와 결과가 같아지므로, 마지막 태그는 제거할 수 없고 제외 태그가 비어 있는 동안 **최소한 하나의 태그를 선택해 주세요.** 안내가 표시됩니다.
    - 제외 태그 목록은 **선택한 태그만 받기** 모드의 수신 태그 목록과 별도로 저장됩니다. 두 모드를 번갈아 선택해도 각 모드에서 고른 태그가 그대로 남습니다.
    - **미분류** 가상 태그도 제외 대상으로 선택할 수 있으며, 태그 한 개로 계산됩니다.

알림 수신 언어 선택

사용자의 알림 수신 언어를 선택할 수 있습니다.

1. **경고 알림** > **이벤트 수신 설정**으로 이동하세요.
2. **사용자별 이벤트 수신 설정** 섹션의 사용자 목록에서 언어를 설정할 사용자의 가장 왼쪽에 ▶ 버튼을 클릭하세요.
3. **알림 수신 언어**를 선택하세요.

요일 및 시간별 알림 설정

사용자의 요일별, 시간별 알림 수신 여부를 선택할 수 있습니다.

1. **경고 알림** > **이벤트 수신 설정**으로 이동하세요.
2. **사용자별 이벤트 수신 설정** 섹션의 사용자 목록에서 요일 및 시간별 알림을 설정할 사용자의 가장 왼쪽에 ▶ 버튼을 클릭하세요.
3. 경고 알림 수신을 원하는 요일을 선택하거나 시간을 입력하세요.
    - 알림 수신 수단별로 설정할 수 있습니다.

3rd 파티 플러그인

외부 애플리케이션을 통해 경고 알림을 받을 수 있습니다.

- 지원 애플리케이션: Slack, Telegram, Teams, Teams Workflows, Jandi, AlertNow, Webhook, Webhook JSON, Opsgenie, PagerDuty, LINE, ilert

> ⓘ 와탭랩스의 지원 범위에 포함하지 않는 사내 메신저는 표준 Webhook, Webhook json을 통해 연동할 수 있습니다.

표 | 3rd 파티 플러그인 목록 구성

항목	설명
플러그인 이름	플러그인의 이름

항목	설명
인증 키	인증 키
인증 값	인증값
수신 레벨	경고, 전체, 위험으로 구분
반복 알림(에스컬레이션)	알림 발생 시간으로부터 알림 발생 상황이 해소되지 않았을 경우, 최초 알림 발생 시각으로부터의 알림 재발행 주기 - 위험(Critical) 단계의 알림에만 적용되는 기능
Smart 메시지(AI)	Telegram 3rd 파티 플러그인 추가 시 AI 알림 기능 제공 - 지원 언어: 한국어, 영어, 일본어
이벤트 수신 태그	프로젝트의 멤버 중 특정 멤버 또는 팀을 대상으로 알림 수신 여부
삭제	3rd 파티 플러그인 삭제

3rd 파티 플러그인 추가

1. **경고 알림** > **이벤트 수신 설정**으로 이동하세요.
2. **3rd 파티 플러그인** 섹션의 [추가하기] 버튼을 클릭하세요.
3. **서드 파티 플러그인 추가** 창에서 원하는 서비스 탭으로 선택하세요.
4. 선택한 서비스의 안내(3rd 파티 플러그인 알림 추가 방법)에 따라 설정을 진행하세요.
5. [수신 테스트] 버튼을 클릭해 설정이 잘 되었는지 확인하세요.
6. 모든 과정을 완료하면 [추가하기] 또는 [등록] 버튼을 클릭하세요.

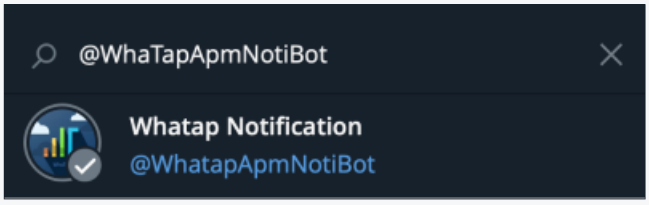
3rd 파티 플러그인 알림 추가 방법

Slack

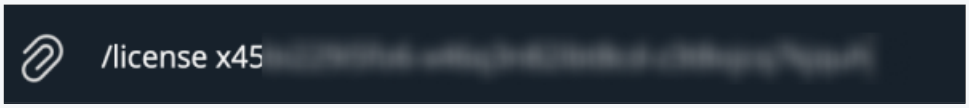
서드 파티 플러그인 추가의 **Slack** 탭 화면의 [**슬랙에 추가하기**] 버튼을 클릭하면 Slack의 팀 및 채널에 **WhaTapNotiBot**을 등록할 수 있습니다.

Telegram

1. **Telegram**에서 **@WhaTapApmNotiBot**을 검색하세요.



2. Bot을 선택하고 [시작] 버튼을 클릭하세요.
3. 채팅창에 `/license` 명령어와 알림을 받을 프로젝트의 액세스 키를 채팅창에 입력하세요.

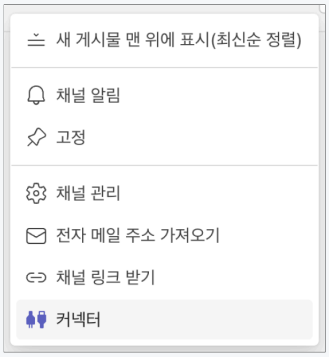


사용할 수 있는 명령어 목록

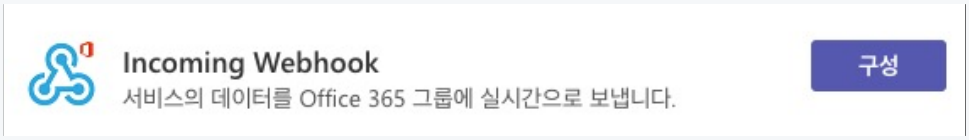
- `/help` : 명령어 목록 보기
- `/license` : 알림 받을 프로젝트의 액세스 키 입력
- `/remove` : 알림을 받지 않을 프로젝트 코드 입력
- `/list` : 현재 알림을 받고 있는 프로젝트 코드 보기

Teams

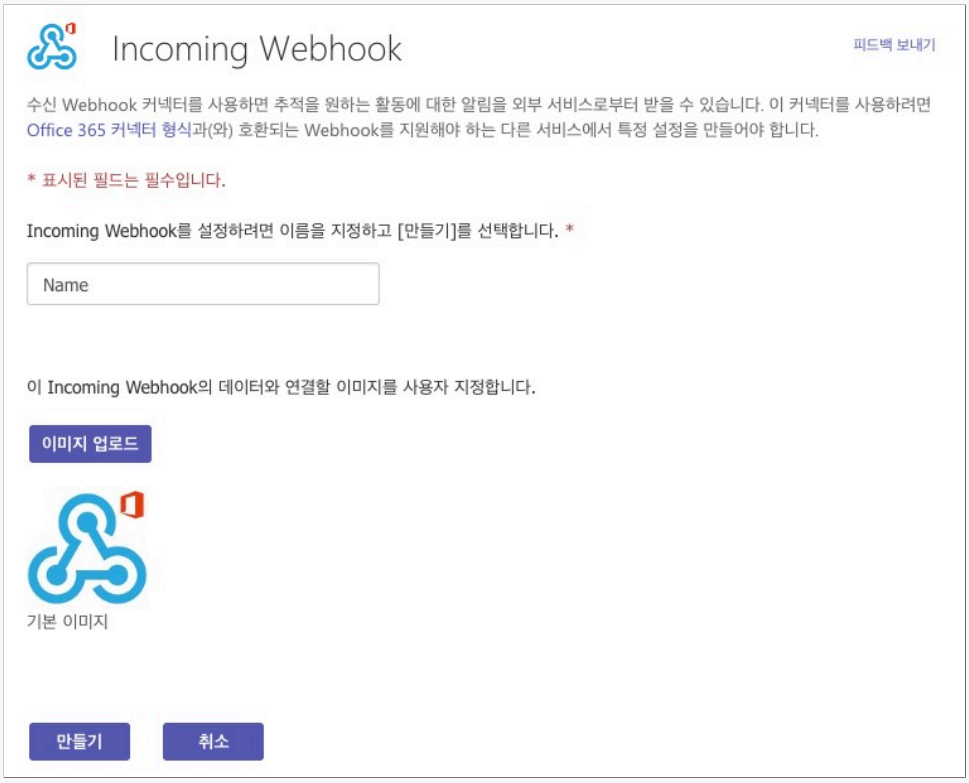
1. 알림을 받을 채널의 **커넥터**로 이동하세요.



2. **Webhook**을 추가하고 [구성] 버튼을 클릭하세요.



3. Webhook 이름을 입력하고 [만들기] 버튼을 클릭하세요.



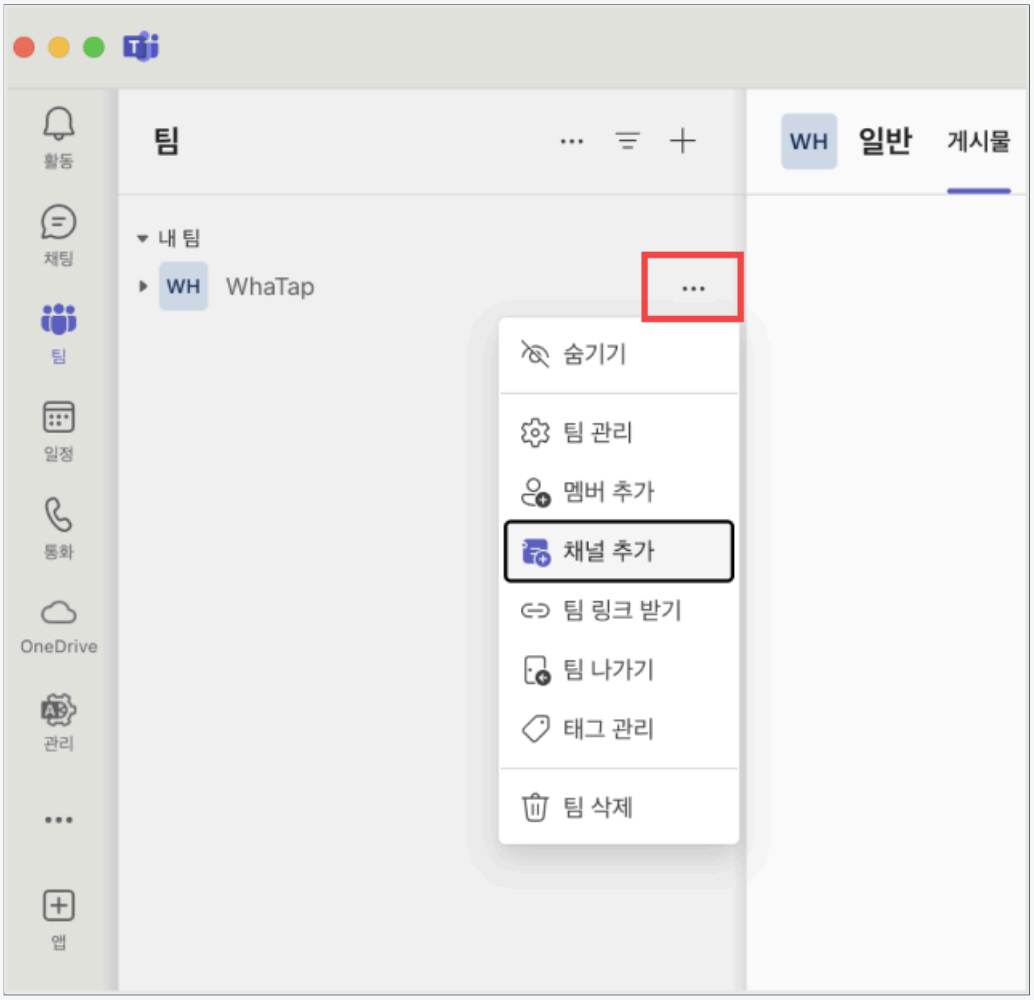
4. 생성된 Webhhok URL을 복사하세요.

5. 와탭 모니터링 서비스의 **서드 파티 플러그인 추가** 창으로 돌아와 **Teams** 탭 하단의 **Channel Name**과 **Webhook URL** 항목을 입력하세요.
6. [**등록**]버튼을 클릭하세요.

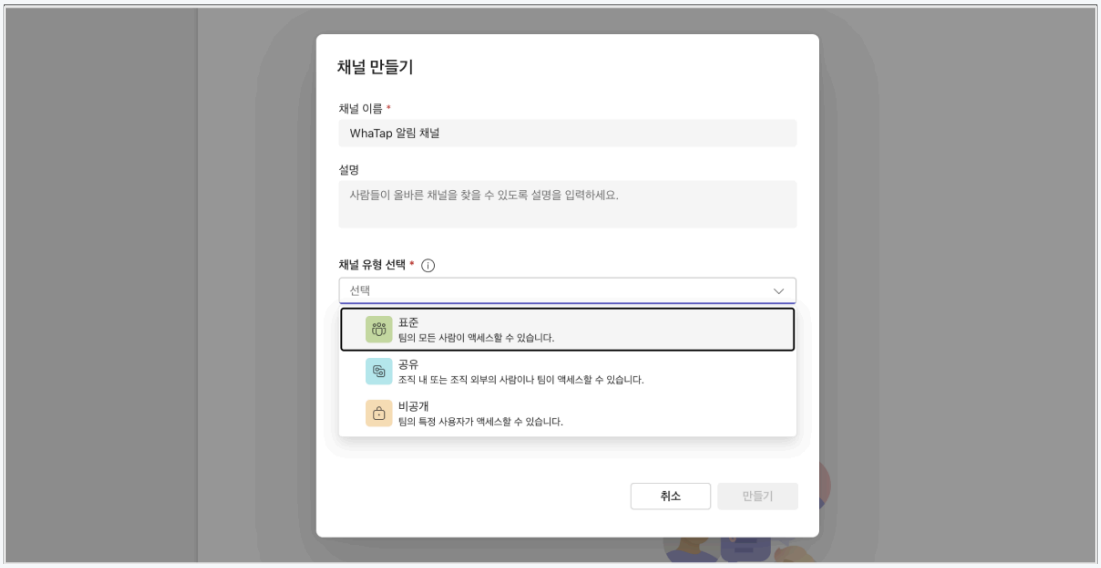
Teams Workflows

워크플로를 사용하려면 **Microsoft Teams**에 액세스할 수 있는 계정과 비즈니스 플랜 구독이 필요합니다.

1. **팀** 메뉴의 더보기(...) > **채널 추가**를 클릭하세요.

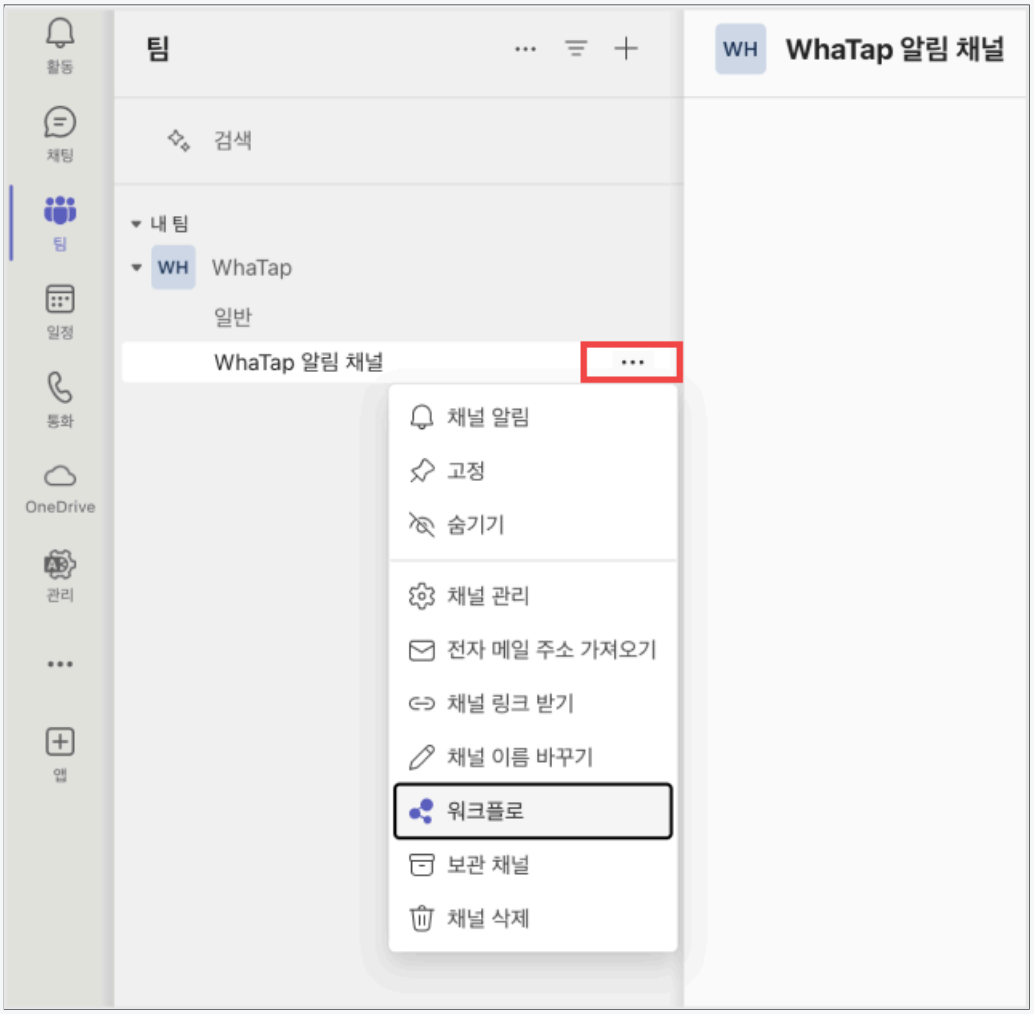


2. **채널 만들기** 창에서 **채널 이름**과 **설명**을 입력하고, **채널 유형 선택** 항목은 **표준**으로 선택한 후, [만들기] 버튼을 클릭하세요.



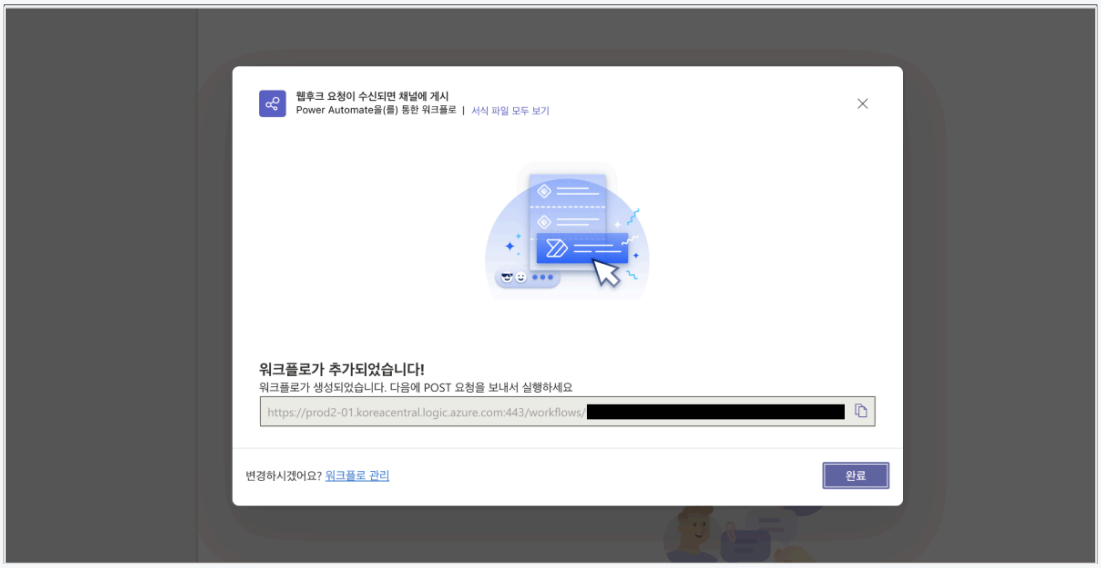
> 채널 유형을 **공유**로 선택하면 워크플로를 사용할 수 없으며, **비공개**를 선택하면 워크플로를 추가할 수 있지만 알림을 수신할 수 없습니다.

3. 생성한 채널의 더보기(...) > **워크플로**를 클릭하세요.



4. **워크플로** 대화상자의 **워크플로**의 목록에서 **웹후크 요청이 수신되면 채널에 게시**를 선택 후, [추가 워크플로] 버튼을 클릭하세요.

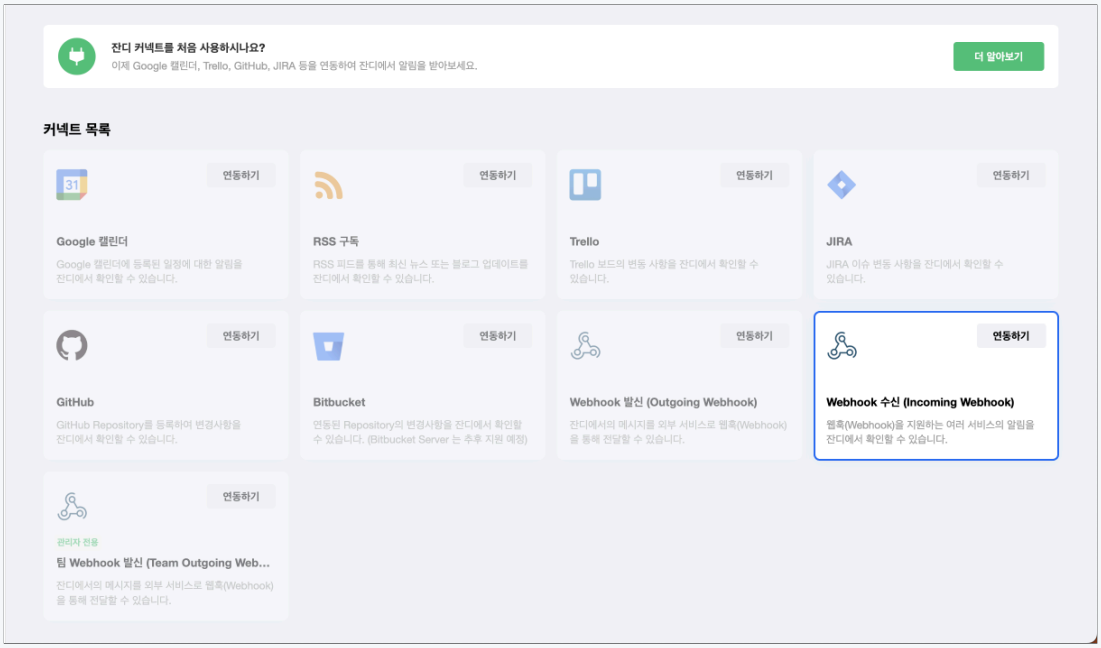
5. **이름**과 **연결** 상태를 확인한 후, [다음] 버튼을 클릭하세요.
6. **세부 정보**를 확인한 후, [워크플로 추가] 버튼을 클릭하세요.
7. 워크플로가 추가되었습니다. 생성된 주소를 복사하세요.



8. 와탭 모니터링 서비스의 **서드 파티 플러그인 추가** 창으로 돌아와 **Teams Workflows** 탭 안내 하단의 **Channel Name**과 **Webhook URL** 항목을 입력하세요.
9. [**등록**] 버튼을 선택하세요.

잔디(Jandi)

1. 알림을 받을 토픽 또는 채팅에서 잔디 커넥트 설정으로 이동하세요.
2. **커넥트 목록**에서 **Webhook 수신 (Incoming Webhook)** 항목에서 [연동하기] 버튼을 클릭하세요.

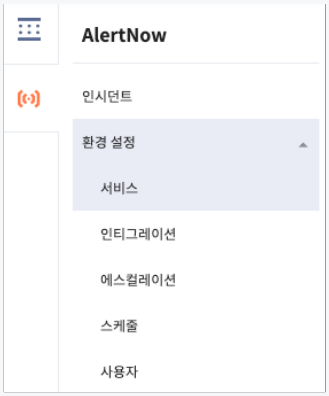


3. [Webhook 수신 추가] 버튼을 클릭하세요.
4. **Webhook URL 등록** 항목에 Webhook URL을 복사하세요.

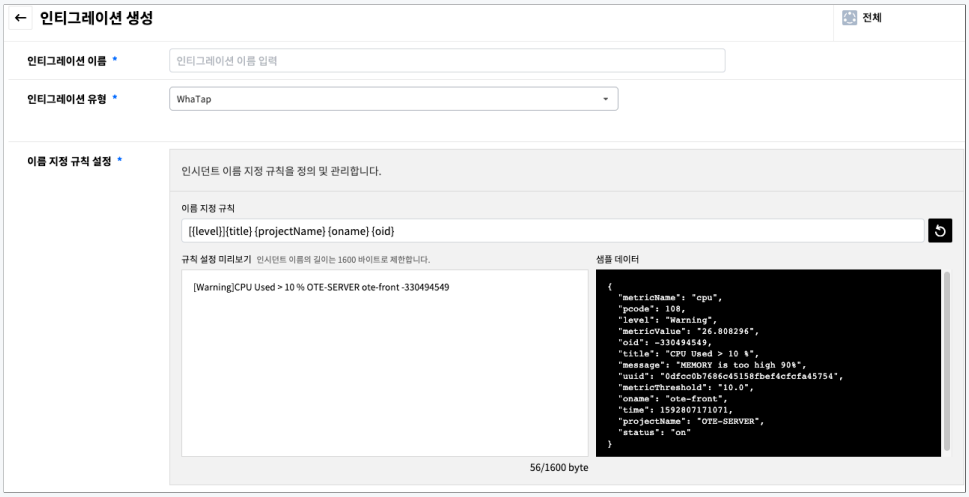
5. 와탭 모니터링 서비스의 **서드 파티 플러그인 추가** 창으로 돌아와 **Jandi** 탭 안내 하단의 **Channel Name**과 **Webhook URL** 항목을 입력하세요.
6. [**등록**] 버튼을 클릭하세요.

AlertNow

1. **AlerNow** 서비스를 접속한 후, **환경 설정** > **Integration** 메뉴로 이동하세요.



2. **인티그레이션** 화면에서 [인티그레이션 생성] 버튼을 클릭하세요.
3. Integration 유형 중 **WhaTap**을 선택하세요.
4. **인티그레이션 생성** 화면에서 이름 및 이름 지정 규칙을 설정하세요.



5. [확인] 버튼을 클릭하면 Integration이 생성됩니다.
6. 생성된 인티그레이션 목록을 확인하고 생성된 Integration 항목을 선택하세요.

7. 상세 화면에서 URL 오른쪽의 복사 아이콘을 클릭해 URL을 복사하세요.

8. 와탭 모니터링 서비스의 **서드 파티 플러그인 추가** 창으로 돌아와 **AlertNow** 탭 안내 하단의 **Channel Name**과 **Webhook URL** 항목을 입력하세요.
9. [**등록**] 버튼을 클릭하세요.

Webhook

1. 알림을 받을 Webhook을 등록하세요. 등록한 URL에 POST Method로 알림의 `title` , `text` 가 전송됩니다.

```
{
    "title":"Alert Title",
    "text": "Sample warning message"
}
```

1. **Webhook Name**과 **Webhook URL** 항목을 입력하세요.
2. [**등록**] 버튼을 클릭하세요.

Webhook JSON

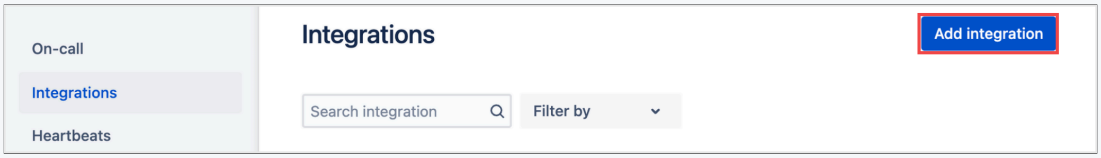
1. 알림을 받을 Webhook을 등록하세요. 등록한 URL에 POST Method로 알림 데이터가 JSON 형태로 전송됩니다. JSON 내에 포함 시킬 수 있는 대체 변수에 대해선 **사용 가능 변수** 부분을 참고하세요.
   - `${something}` 은 와탭에서 제공하는 대체될 값입니다. 지원하는 전체 목록과 의미는 사용 가능 변수를 참고하세요.
   - 이벤트 상태(`status`)는 이벤트 발생(`on`), 이벤트 해소(`off`)로 구분할 수 있습니다.
   - `level` 은 `None` , `Info` , `Warning` , `Critical` 로 구분합니다.

```
{
  "pcode": "${pcode}",
  "projectName": "${projectName}",
  "time": "${time}",
  "oid": "${oid}",
  "oname": "${oname}",
  "title": "${title}",
  "message": "${message}",
  "level": "${level}",
  "status": "${status}",
  "metricName": "${metricName}",
  "metricValue": "${metricValue}",
  "metricThreshold": "${metricThreshold}",
  "uuid": "${uuid}",
  "okind": "${okind}",
  "onode": "${onode}"
}
```

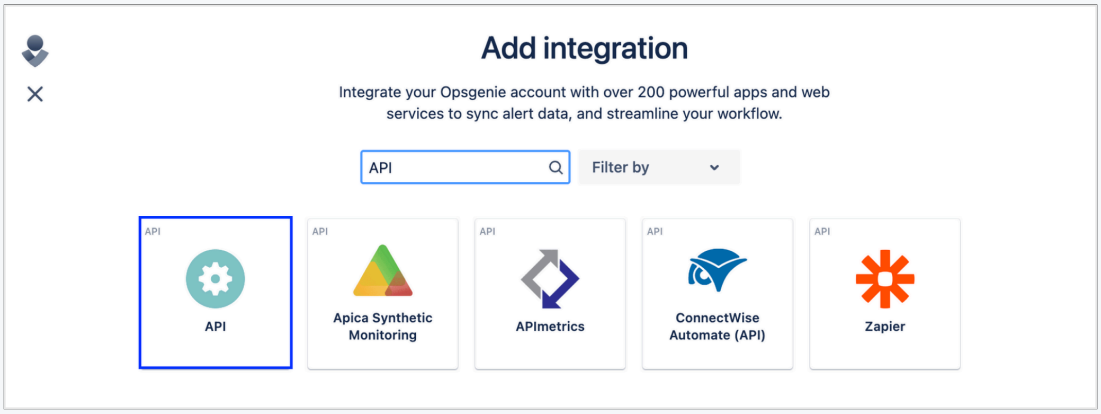
2. **Webhook Name**과 **Webhook URL** 항목을 입력하세요.
3. Header 값을 추가하려면 **사용자 Header 값 추가**의 **+ 추가**를 클릭해 key, value을 추가하세요.
4. [**등록**] 버튼을 클릭하세요.

Opsgenie

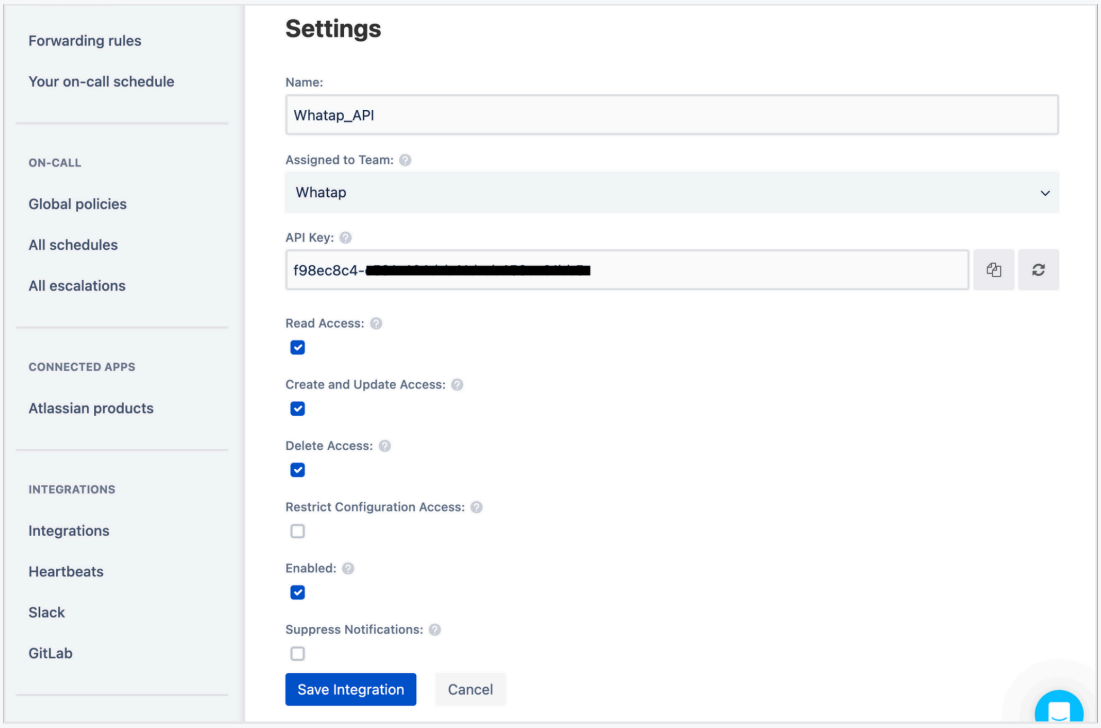
1. **Teams** 메뉴에서 알림을 수신할 팀을 선택하세요.
2. **Integration** 메뉴로 이동해 [Add Integration] 버튼을 클릭하세요.



3. Integration 목록에서 'API' 검색 후 선택하세요.



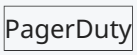
4. API 명과 Team을 할당하고 각 권한을 설정한 후, [Save Integration] 버튼을 클릭하세요.



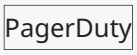
5. 와탭 모니터링 서비스의 **서드 파티 플러그인 추가** 창으로 돌아와, **Opsgenie** 탭 안내 하단의 **Name**과 **API Key** 항목을 입력하세요.
6. [**등록**] 버튼을 선택하세요.

PagerDuty

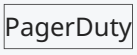
1. PagerDuty 계정에 등록하고 로그인하세요.
2. PagerDuty에서 **Services** > **Service Directory** 메뉴에서 [+ New Service] 버튼을 클릭하세요.

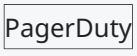
3. Service 정보(**Name**)를 입력한 후 [Next] 버튼을 클릭하세요.

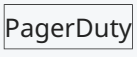
4. Escalation Policy를 설정 후 [Next] 버튼을 클릭하세요.

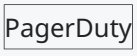
5. **Alert Grouping**과 **Transient Alerts** 항목을 설정한 후 [Next] 버튼을 클릭하세요.

6. **Integrations**에서 **Event API V2**를 선택한 후 [Create Service] 버튼을 클릭하세요.

7. 생성한 Service 화면에서 **Integration Name**과 **Integration Key** 값을 복사하세요.

8. 와탭 모니터링 서비스의 **서드 파티 플러그인 추가** 화면으로 돌아와 **Integration Name**과 **Integration Key** 항목을 입력하세요.
9. [등록] 버튼을 클릭하세요.

> 메시지를 수신하지 못했거나 플러그인이 등록되지 않았다면 **Integration Name** 또는 **Integration Key**값이 올바른지 확인하세요.

LINE

1. LINE 앱을 설치합니다.
2. 아래 QR 코드를 스캔하여 **WhaTapLabs** 공식 채널을 친구로 추가합니다.

3. 다음 명령어를 사용하여 프로젝트를 라인과 연동하고 알림을 설정하세요.

명령어

```
/help # 사용 가능한 명령어 안내
/list # 연동된 프로젝트 목록 조회
/license [액세스키] # 새 프로젝트 연동 추가
/remove [프로젝트코드] # 프로젝트 연동 해제
/language [ko|ja|en] # 알림 언어 설정 (한국어/일본어/영어)
```

- 프로젝트 액세스 키는 **관리 > 프로젝트 관리**에서 확인하실 수 있습니다.
- 라인 내에서 `/language` 명령어를 사용하는 것은 라인 내 채팅방에만 적용됩니다. 다른 3rd 플러그인이나 프로젝트의 알림 수신 언어가 변경되는 것은 아닙니다.

사용 예시

```
/license XXXXXXXXXX # 프로젝트 연동
/language ko # 한국어로 언어 설정
```

```
/list # 연동된 프로젝트 확인
```

- 프로젝트 액세스 키: WhaTap 프로젝트 내 **관리 > 프로젝트 관리**에서 확인할 수 있음
- 언어 설정: `/language` 명령어는 LINE 채팅방 내에서만 적용되며, 다른 알림 채널의 언어는 변경되지 않음

ilert

1. ilert 콘솔에서 **Alert sources** 화면의 오른쪽 위의 [+ Create alert source]를 클릭하세요.
2. **Integration type 단계**의 하단에 있는 **Filter integrations 검색창**에 `WhaTap` 을 입력해 검색합니다.
3. 검색 결과에서 **WhaTap 타일**을 선택한 후 [Next] 버튼을 클릭합니다.
4. 알림 소스 이름을 입력하고 팀을 지정한 후 [Next] 버튼을 클릭합니다.
   - **에스컬레이션 정책은** 새 정책을 생성하거나 기존 정책을 선택할 수 있습니다.
5. 알림 그룹화 방식을 선택한 후 [Continue setup] 버튼을 클릭합니다.
   - 우선 선택한 **Do not group alerts는** 이후 변경할 수 있습니다.
6. 다음 화면에서 **알림 템플릿, 우선순위 등 추가 설정**을 할 수 있습니다. 우선 [Finish setup] 버튼을 클릭해 설정을 완료합니다.
7. 설정이 완료되면 완료 화면에서 **API 키** 또는 **Webhook URL**이 생성됩니다.
8. **Webhook JSON** 탭에서 생성된 **Webhook URL**을 복사해 아래의 **Webhook URL 입력란**에 붙여넣습니다.
9. [등록] 버튼을 클릭하세요.

대량 알림 발생 방지

알림이 대량으로 발생하면 지정한 시간 동안 알림이 일시적으로 중지됩니다. 예를 들어, 5분 동안 20회의 이벤트가 발생하면 5분

동안 경고 알림을 중지합니다. 설정한 **정지 시간** 시간이 지나면 대량 알림 발생 방지 기능은 해제됩니다.

1. **경고 알림** > **이벤트 수신 설정**의 **대량 알림 발생 방지** 섹션으로 이동하세요.
2. 토글 버튼을 클릭해 **활성화** 또는 **비활성화** 할 수 있습니다.
   - **활성화**: 대량 알림 일시적 중지
   - **비활성화**: 대량 알림 재개
3. **탐지 시간, 탐지 횟수, 정지 시간**을 설정하세요.
   - **탐지 시간** 동안 **탐지 횟수** 이상의 이벤트가 발생하면 **정지 시간** 동안 경고 알림을 중지합니다.

> ⓘ 대량 알림으로 차단되는 경우 차단 안내 박스가 표시됩니다. 대량 알림 차단 기능을 해제하려면 박스의 [대량 알림 발생 중지] 버튼을 클릭하세요.

이벤트 기록 및 모니터링

홈 > 프로젝트 선택 > **경고 알림** > **이벤트 기록**

이벤트 기록은 프로젝트에서 발생한 여러 이벤트를 조회하고 상세 정보를 분석할 수 있습니다. Elasticsearch DSL 쿼리를 지원하며, 원하는 조건에 맞는 이벤트를 정확하고 빠르게 찾을 수 있습니다.

이벤트 기록 조회하기

검색 결과에서 **이벤트 제목을 클릭**하면 해당 이벤트가 발생한 시간대의 상세 데이터를 분석할 수 있는 화면으로 이동합니다. 이를 통해 이벤트 발생 전후 상황을 종합적으로 파악할 수 있습니다.

1. **경고 알림** > **이벤트 기록**으로 이동하세요.
2. **이벤트 기록** 화면의 **시간 선택**에서 이벤트 기록을 조회할 기간를 설정하세요.
3. 필터 입력 창에 쿼리을 입력하고 [돋보기] 버튼을 클릭하세요.
    - 쿼리를 입력하지 않으면 전체 이벤트가 조회됩니다.

컬럼 설정

원하는 컬럼을 표시하거나 숨길 수 있습니다.

1. **이벤트 기록** 화면의 왼쪽 위 [컬럼 설정] 버튼을 클릭하세요.
2. **컬럼 설정** 창에서 표시하거나 숨길 컬럼을 선택하세요.
    - **표시 항목**에서 컬럼의 순서를 변경할 수 있습니다.
    - **컬럼명 검색**창에 원하는 컬럼 항목을 검색할 수 있습니다.

컬럼 항목	설명
이벤트 이름	이벤트의 제목
메시지	이벤트의 메시지 또는 스냅샷 정보

컬럼 항목	설명
이벤트 상태	이벤트의 현재 상태
에이전트 명	이벤트가 발생한 대상 에이전트의 이름(Oname) - 특정 에이전트에 대해 발생한 이벤트가 아닌 경우 빈칸으로 표시됨
에이전트 그룹	이벤트가 발생한 대상 에이전트의 종류(OkindName) - 특정 에이전트에 대해 발생한 이벤트가 아닌 경우 빈칸으로 표시됨
에이전트 노드	이벤트가 발생한 대상 에이전트의 서버(OnodeName) - 특정 에이전트에 대해 발생한 이벤트가 아닌 경우 빈칸으로 표시됨
이벤트 발생 시각	이벤트가 발생한 시간
이벤트 해소 시각	이벤트가 해소된 시간 상태가 없는 이벤트의 경우 `-`로 표시됨

3. 설정을 완료한 후, [확인] 버튼을 클릭해 컬럼 설정을 반영합니다.

CSV 다운로드

조회 결과를 CSV 파일 형식으로 다운로드할 수 있습니다.

1. **경고 알림** > **이벤트 기록**에서 [CSV] 버튼을 클릭하세요.
2. 현재 필터 조건에 맞는 이벤트 목록이 CSV 파일로 저장됩니다.

진행 중인 이벤트만 보기

현재 진행 중인 상태의 이벤트만 조회할 수 있습니다.

1. **경고 알림** > **이벤트 기록**에서 [진행 중인 이벤트만 보기] 버튼을 클릭하세요.
2. 필터 쿼리에 다음 조건이 자동으로 추가되어 검색됩니다.

```
Stateful: true and Status : "ON"
```

이벤트 검색

- **정확한 검색**: 원하는 조건을 정확히 지정하여 필요한 이벤트만 조회
- **복합 조건**: 여러 필드를 동시에 검색하여 복잡한 조건도 한 번에 처리
- **유연한 패턴**: 와일드카드, 부분 검색 등 다양한 검색 방식 지원
- **빠른 성능**: 인덱싱된 데이터를 활용한 고속 검색

검색 가능한 필드

필드와 값에 대해 대소문자를 구분하지 않습니다.

- **필드명 태그**: 필드가 지원되는 이벤트 타입
- **필드 타입**: 필드의 데이터의 타입
  - `#`: Number
  - `T`: String
  - : Boolean
  - : Date

이벤트 정보 필드

필드명	타입	설명	예시 값
Title	String	이벤트 제목	"Database Connection Error"
OffTitle	String	이벤트 해소 제목	"RECOVERED: Database Connection Error"
Message	String	이벤트 메시지	"Connection timeout occurred"
OffMessage	String	이벤트 해소 메시지	"RECOVERED: Connection timeout occurred"
Level	String	현재 이벤트 레벨	`Critical`, `Warning`, `Info`

필드명	타입	설명	예시 값
OriginLevel	String	원본 이벤트 레벨	`Critical`, `Warning`, `Info`
Status	String	이벤트 상태	`ON`, `OFF`, `CANCEL`, `ACKNOWLEDGE`, `MAINTENANCE`, `DISABLED`
ActCount	Number	처리내역 개수	2
MetricName	String	지표 이름	"memory"
MetricValue	String	지표 값	"85.5"
OffValue	String	해소 값	"72.8"
MetricThreshold	String	임계치	"80"
alertType	String	이벤트 종류	"BASIC", "METRICS", "TRANSACTION" 등
alertId	String	이벤트 규칙의 고유식별자	"zf3uojer0fv4v7"

이벤트 타입별 지원 필드

> ⓘ 특정 이벤트 타입에서만 지원되는 필드입니다. 해당 이벤트 타입이 아닌 경우 값이 없을 수 있습니다.

필드명	타입	이벤트 종류	설명	예시 값
eventRule	String	기본, 메트릭스	이벤트 발생 규칙	"memory ≥ 80"
field	String	실시간 로그	로그 검색 키	"content"

필드명	타입	이벤트 종류	설명	예시 값
keyword	String	실시간 로그	로그 검색 값	"Error"
logCategory	String	실시간 로그	로그 카테고리	"AppLog"
logContent	String	실시간 로그	로그 내용	"00:00000:00009:2025/04/06 23:03:52.55 server Error: 1601, Severity: 17, State: 3\n00:00000:00009:2025/04/06 23:03:52.55 server There are not enough 'user connections' available to start a new process."

대상 필드

필드명	타입	설명	예시 값
Oid	Number	에이전트 고유식별자	-98765432
Oname	String	에이전트 이름	"web-server-01"
Okind	Number	에이전트 종류 고유식별자	867318026
OkindName	String	에이전트 종류명	"web-server"
Onode	Number	에이전트 서버 고유식별자	334634079
OnodeName	String	에이전트 서버명	"production-node-1"

이벤트 고유값 필드

필드명	타입	설명	예시 값
Id	Number	이벤트의 고유식별자	5768121

필드명	타입	설명	예시 값
UUID	String	이벤트 고유식별자	"550e8400-e29b-41d4-a716"

상태/플래그 필드

필드명	타입	설명	예시 값
Stateful	Boolean	상태 기반 이벤트 여부	true, false
Disabled	Boolean	종료된 이벤트 여부	true, false
Escalation	Boolean	에스컬레이션 여부	true, false
SystemEvent	Boolean	시스템 이벤트 여부	true, false

검색 유형별 문법

> ⓘ 검색 기본 문법은 WhaTap log search query 문법 문서를 참고하시기 바랍니다.

1. 키워드 검색

Database Connection 문자열이 포함된 이벤트

```
title: "Database Connection"
```

2. 여러 값 검색

여러 옵션 중 하나라도 포함되는 이벤트를 조회합니다.

```
title: Database Connection
```

현재 이벤트 레벨이 Info이거나 Warning인 이벤트

```
level: info warning
```

3. 패턴 검색

와일드카드를 사용하여 패턴을 검색할 수 있습니다.

이름 패턴으로 특정 에이전트에서 발생된 이벤트 찾기

```
oname: web-*-prod
```

제목이 Connection으로 끝나는 이벤트 찾기

```
title: *Connection
```

4. 복합 조건 검색

여러 조건을 조합하여 정확히 검색할 수 있습니다.

Critical 레벨이면서 제목에 Database가 포함된 이벤트

```
level: critical and title: Database
```

특정 에이전트의 Warning 또는 Critical 이벤트

```
oname: web-server-01 and level: Warning Critical
```

5. OR 조건 검색

여러 조건 중 하나라도 만족하는 이벤트를 조회합니다.

제목 또는 메시지에 특정 키워드가 포함된 이벤트

```
title: Connection or message: Connection
```

6. 제외 조건

특정 조건을 제외해 검색할 수 있습니다.

Info 레벨을 제외한 모든 이벤트

```
not level: info
```