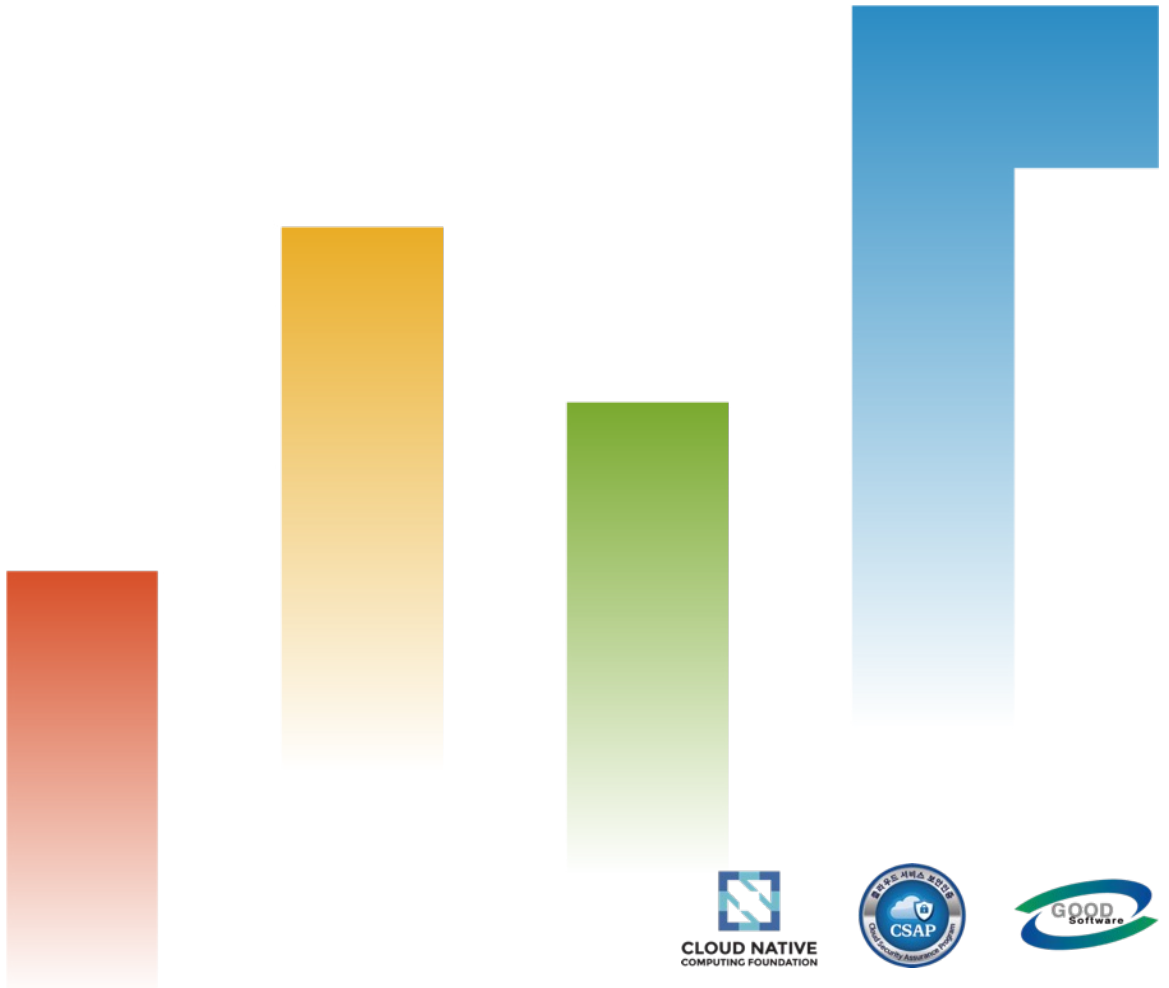


# Java 모니터링

기술 문서 2024.04.02



# Java 모니터링

Web Application Server(WAS), 배치 애플리케이션 등 JVM 환경에서 동작하는 모든 애플리케이션에 적용할 수 있습니다. 애플리케이션 장애분석을 위해 Thread Dump(스레드 덤프)를 확보하던 번거로움에서 벗어나, 이미 확보된 스택을 통해 장애 상황의 재현 없이도 문제의 원인을 정확하게 특정할 수 있습니다.

와탭의 애플리케이션 모니터링은 장애 재현 없이 실시간으로 모니터링이 가능합니다.

## 주요 특징점

### • 실시간 트랜잭션 모니터링

와탭은 현재 진행중인 트랜잭션의 수행 현황을 실시간으로 보여줍니다. 모든 애플리케이션 문제는 진행중인 트랜잭션 증가로 나타납니다. 증가한 순간에 수행중인 SQL, 외부 호출, Method를 즉시 확인 할 수 있다면 문제를 가장 빠르게 인지하고 분석할 수 있습니다.

### • 트랜잭션 성능 Deep Dive

와탭은 모든 트랜잭션의 수행이력을 수집해 분석할 수 있습니다. 트랜잭션 성능은 SQL, Http Call 뿐 아니라 Method 레벨까지 추적할 수 있습니다.

특히 **액티브 스택** 기능으로 번거로운 Method 트레이스 설정 없이 Method 레벨 분석이 가능합니다. **액티브 스택**은 와탭만의 특허 기술로 기존의 다른 상품에서는 발견하지 못했던 숨겨진 영역의 문제 원인을 찾아 개선 포인트를 제공합니다.

ⓘ 액티브스택 기능은 Java, Python에서만 지원합니다.

### • MSA 호출 패턴 분석

MSA 구조의 복잡한 애플리케이션간의 호출 관계를 트랜잭션 URL 기준으로 호출 비중을 통계적으로 시각화하여 제공합니다. 시점별로 변화하는 호출 관계 및 호출 비중의 변화를 통계적으로 조망할 수 있습니다.

### • AI 기반 응답 분포 패턴 분석

와탭은 특정 패턴을 머신 러닝을 통해 자동 인식하여 알림 경고를 발생, 운영자가 인지하지 못한 이슈까지 알려드립니다. (설치형 별도 협의)

출원번호: 10-2020-0037381

### • 멀티 프로젝트 트랜잭션 연계 분석

MSA 환경에서 트랜잭션 연계 추적 기능을 제공합니다. 전체 트랜잭션을 추적하여 어느 애플리케이션에서 지연이 발생하는지를 파악할 수 있습니다.

### • 스택 통계 분석

통계분석에서는 실시간 모니터링이나 트레이스 정보에서 발견하기 힘든 튜닝 포인트를 찾아낼 수 있습니다. 거시적인 문제의 추이 또한 그래프와 표를 통하여 개선 대상을 선별하거나 특정 시간대의 응답 지연 현상을 확인할 수 있습니다.

- **사후 분석(Cube)**

시간을 축으로 다양한 지표를 한번에 확인 할 수 있습니다. 응답시간, 에러 건수 및 처리량 등이 높은 시간대만을 특정할 수 있어 동 시간대 문제 요소를 식별하거나 서비스 현황을 정확하게 조망할 수 있습니다.

- **서비스 폭주 시 부하량 제어**

시스템이 감당할 수 없는 과도한 요청이 들어오는 경우에 일부 요청을 거절하거나 잘못된 서비스 요청은 차단하여 안정적으로 서비스를 운영할 수 있습니다.

- **비동기 트랜잭션 트레이스**

와탭만의 고유한 BCI(Byte Code Instrumentation) 기술로 비동기 트랜잭션의 성능을 효과적으로 추적합니다. WebFlux, Reactor, Hystrix 등을 사용하는 애플리케이션의 트랜잭션 트레이스 정보를 수집합니다.

# 지원 환경

## Java 에이전트 지원 환경

에이전트를 설치하기 전에 JVM 애플리케이션 서버가 다음의 사양을 충족하는지 확인하세요.

Web Application Server (WAS) 뿐 아니라 데몬 및 배치 애플리케이션 등 JVM에서 동작하는 모든 애플리케이션에 적용할 수 있습니다. Java 1.6 미만은 [support@whatap.io](mailto:support@whatap.io)로 문의하거나 [다음 문서](#)를 참조하세요.

### • 지원 버전

지원범위	환경	OS	JVM 버전
Fully support	Java	JVM으로 구동하는 모든 OS	Java 6 이상, Java 20 이하
Limited support	Java	JVM으로 구동하는 모든 OS	Java 1.5 이하

### • 운영체제

- HP-UX 11.x 32bit, 64bit
- Itanium 64bit
- Oracle Solaris 2.8, 2.9, 10, 11 32bit, 64bit
- x86 Intel Linux 32bit
- Red Hat Itanium 64bit
- Microsoft Windows 2000, XP, 2003, 2008, 7, 8

① JDK 6를 설치할 수 있는 모든 운영체제를 지원합니다.

### • 애플리케이션 서버

- BEA WebLogic 9.x, 10.x, 11.x, 12.x
- Tmaxsoft JEUS 6.x, 7.x
- SUN Application Server 8.x, 9.x
- Fujitsu Interstage 5.x, 6.x, 7.x
- Hitachi Cosminexus 7.x, 8.x, 9.x
- Apache Jakarta Tomcat 6.x, 7.x, 8.x, 9.x, 10.x
- Caucho Technology Resin 3.x, 4.x
- Red Hat JBoss Application Server 6.x, 7.x
- GlassFish 3.x, 4.x, 5.x

① 이외 Java EE Spec을 준수하는 모든 애플리케이션 서버를 지원합니다. [다음 링크](#)를 참조하세요.

### • 프레임워크 / 라이브러리

- Spring Boot Project
- Netty, Akka HTTP 및 Play Framework 등 비동기 Framework
- Quarkus, Quarkus-reactive
- Mule Framework 3.9.5, Mule Framework 4.5

① 이외 Java EE Spec을 준수하는 Application Server에서 동작하는 모든 라이브러리를 지원합니다.

### • 데이터베이스

- Generic JDBC (any JDBC compliant driver)
- DB2 JDBC
- Derby JDBC



- H2 JDBC
- jTDS JDBC
- MariaDB JDBC
- Microsoft SQL Server JDBC
- MySQL mysql-connector-java
- Oracle ojdbc5, ojdbc6, ojdbc7, ojdbc8, ojdbc14
- Postgres JDBC
- Tiberio
- Jedis Redis driver

## 애플리케이션 서버별 호환성 및 데이터 소스

### Tomcat

#### 호환성

WAS Ver.	Java SE 6 (jdk1.6.0)	Java SE 7 (jdk1.7.0_80)	Java SE 8 (jdk1.8.0_91)
Tomcat 6	O	O	O
Tomcat 7	O	O	O
Tomcat 8	X	O	O
Tomcat 9	X	X	O

WAS Ver.	OpenJDK 6 (1.6.0_45)	OpenJDK 7 (1.7.0_80)	OpenJDK 8 (1.8.0_91)
Tomcat 6	O	O	O
Tomcat 7	O	O	O
Tomcat 8	X	O	O
Tomcat 9	X	X	O

WAS Ver.	ibm-java-x86_64-60	ibm-java-x86_64-71	ibm-java-x86_64-80
Tomcat 6	O	O	O
Tomcat 7	O	O	O
Tomcat 8	X	O	O
Tomcat 9	X	X	O

### DataSource

DB	JDBC Driver 파일명	JDBC Ver.	JDK Ver.	호환성
MySQL	mysql-connector-java-5.1.39-bin.jar	5.1.39	JDK 1.7.0_80	O
MariaDB	mariadb-java-client-1.4.6.jar	1.4.6	JDK 1.7.0_80	O
PostgreSQL	postgresql-9.4.1209.jre7.jar	9.4.1209	JDK 1.7.0_80	O
Amazon Aurora	mysql-connector-java-5.1.39-bin.jar	5.1.39	JDK 1.7.0_80	O

DB	JDBC Driver 파일명	JDBC Ver.	JDK Ver.	호환성
Oracle	ojdbc6-11.2.0.2.0.jar	11.2.0.2.0	JDK 1.7.0_80	O
DB2	db2jcc.jar, db2jcc_license_cu.jar	1.4.2	JDK 1.7.0_80	O

## JBoss

### 호환성

WAS Ver.	Java SE 6 (jdk1.6.0)	Java SE 7 (jdk1.7.0_80)	Java SE 8 (jdk1.8.0_91)
JBOSS EAP 7.0 (standalone)	X	X	O
JBOSS EAP 6.1.1 (standalone)	X	O	X
JBOSS EAP 6.2 (standalone)	X	O	O
JBOSS EAP 6.3 (standalone)	X	O	O
JBOSS EAP 6.4 (standalone)	X	O	O
JBOSS EAP 7.0 (domain)	X	X	O
JBOSS AS 5.1.0 (default)	O	O	O

WAS Ver.	OpenJDK 6 (1.6.0_45)	OpenJDK 7 (1.7.0_80)	OpenJDK 8 (1.8.0_91)
JBOSS EAP 7.0 (standalone)	X	X	O
JBOSS EAP 6.1.1 (standalone)	X	O	X
JBOSS EAP 6.2 (standalone)	X	O	O
JBOSS EAP 6.3 (standalone)	X	O	O
JBOSS EAP 6.4 (standalone)	X	O	O
JBOSS EAP 7.0 (domain)	X	X	O
JBOSS AS 5.1.0 (default)	O	O	O

WAS Ver.	ibm-java-x86_64-60	ibm-java-x86_64-71	ibm-java-x86_64-80
JBOSS EAP 7.0 (standalone)	X	X	O
JBOSS EAP 6.1.1 (standalone)	X	O	X
JBOSS EAP 6.2 (standalone)	X	O	O
JBOSS EAP 6.3 (standalone)	X	O	O
JBOSS EAP 6.4 (standalone)	X	O	O
JBOSS EAP 7.0 (domain)	-	-	-
JBOSS AS 5.1.0 (default)	O	O	O

## DataSource

DB	JDBC Driver 파일명	JDBC Ver.	JDK Ver.	호환성
MySQL	mysql-connector-java-5.1.39-bin.jar	5.1.39	JDK 1.7.0_80	O
MariaDB	mariadb-java-client-1.4.6.jar	1.4.6	JDK 1.7.0_80	O
PostgreSQL	postgresql-9.4.1209.jre7.jar	9.4.1209	JDK 1.7.0_80	O
Amazon Aurora	mysql-connector-java-5.1.39-bin.jar	5.1.39	JDK 1.7.0_80	O
Oracle	ojdbc6-11.2.0.2.0.jar	11.2.0.2.0	JDK 1.7.0_80	O
DB2	db2jcc.jar, db2jcc_license_cu.jar	1.4.2	JDK 1.7.0_80	O

## JEUS

## 호환성

WAS Ver.	Java SE 6 (jdk1.6.0)	Java SE 7 (jdk1.7.0_80)	Java SE 8 (jdk1.8.0_91)
JEUS 7	O	O	O
JEUS 6	O	O	O

WAS Ver.	OpenJDK 6 (1.6.0_45)	OpenJDK 7 (1.7.0_80)	OpenJDK 8 (1.8.0_91)
JEUS 7	O	O	O
JEUS 6	O	O	O

WAS Ver.	ibm-java-x86_64-60	ibm-java-x86_64-71	ibm-java-x86_64-80
JEUS 7	O	O	O
JEUS 6	O	O	X

## DataSource

DB	JDBC Driver 파일명	JDBC Ver.	JDK Ver.	호환성
MySQL	mysql-connector-java-5.1.39-bin.jar	5.1.39	JDK 1.7.0_80	O
MariaDB	mariadb-java-client-1.4.6.jar	1.4.6	JDK 1.7.0_80	O
PostgreSQL	postgresql-9.4.1209.jre7.jar	9.4.1209	JDK 1.7.0_80	O
Amazon Aurora	mysql-connector-java-5.1.39-bin.jar	5.1.39	JDK 1.7.0_80	O
Oracle	ojdbc6-11.2.0.2.0.jar	11.2.0.2.0	JDK 1.7.0_80	O
DB2	db2jcc.jar, db2jcc_license_cu.jar	1.4.2	JDK 1.7.0_80	O

## WebSphere

### 호환성

WAS Ver.	ibm-java-x86_64-60
8.5.5.10	O

### DataSource

DB	JDBC Driver 파일명	JDBC Ver.	JDK Ver.	호환성
Mysql	mysql-connector-java-5.1.39-bin.jar	5.1.39	JDK 1.7.0_80	O
Oracle	ojdbc6-11.2.0.2.0.jar	11.2.0.2.0	JDK 1.7.0_80	O

## JETTY

### 호환성

WAS Ver.	Java SE 6 (jdk1.6.0)	Java SE 7 (jdk1.7.0_80)	Java SE 8 (jdk1.8.0_91)
Jetty 8.1.21	X	O	O
Jetty 9.2.18	X	O	O
Jetty 9.3.12	X	X	O

WAS Ver.	ibm-java-x86_64-60	ibm-java-x86_64-71	ibm-java-x86_64-80
Jetty 8.1.21	X	O	O
Jetty 9.2.18	X	O	O
Jetty 9.3.12	X	X	O

### DataSource

DB	JDBC Driver 파일명	JDBC Ver.	JDK Ver.	호환성
MySQL	mysql-connector-java-5.1.39-bin.jar	5.1.39	JDK 1.7.0_80	O
Oracle	ojdbc6-11.2.0.2.0.jar	11.2.0.2.0	JDK 1.7.0_80	O

## 공통 지원 환경

### 브라우저 지원

와탭 모니터링 서비스는 웹브라우저와 모바일 앱에서 이용할 수 있습니다.

브라우저	권장여부	지원버전
Google Chrome	O	84 이상
Mozilla FireFox	X	최신 버전

브라우저	권장여부	지원버전
Edge	X	최신 버전
Safari	X	최신 버전

- ① • 브라우저 호환성과 성능을 이유로 Chrome 최신 버전 사용을 권장합니다.
- 사용자 인터페이스(User Interface, UI)는 HTML5 표준 기술로 구현하여 Internet Explorer는 지원하지 않습니다.

### 방화벽

와탭 에이전트는 수집 서버 TCP 6600 포트로 접속 가능해야 합니다. 모니터링 대상과 가까운 수집 서버 주소를 허용하세요.

출발지: 와탭 에이전트


목적지	목적지 IP	포트
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 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 프랑크푸르트 수집 서버	3.125.142.162 / 3.127.76.140	TCP 6600

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

Proxy

### 모바일 앱

와탭 모바일 앱은 안드로이드와 iOS 환경을 지원합니다. 다음 링크로 이동하거나 QR 코드를 스캔해 앱을 설치할 수 있습니다. 모바일 앱에 대한 자세한 설명은 [다음 문서](#)를 참조하세요.

iOS	Android
 <p>iOS 12 버전 이상</p>	 <p>Android 5.0 버전 이상</p>

# 에이전트 설치

와탭 모니터링 서비스를 사용하기 위해서는 [회원 가입](#) 후 프로젝트를 생성하고 대상 서버에 에이전트를 설치해야 합니다. 회원 가입에 관한 자세한 내용은 [다음 문서](#)를 참조하세요.

다음 동영상 가이드를 참조하세요.

## 프로젝트 생성하기

에이전트를 설치하기 전에 먼저 프로젝트를 생성하세요.

1. [와탭 모니터링 서비스](#)로 이동한 다음 로그인하세요.
2. 프로젝트를 생성하려면 화면 왼쪽 사이드 메뉴에서 [전체 프로젝트](#) > [+ 프로젝트](#) 버튼을 선택하세요.
3. [상품 선택](#) 화면에서 프로젝트에 설치할 상품을 선택하세요.
4. [프로젝트 이름](#), [데이터 서버 지역](#), [타임 존](#) 등의 항목을 차례로 설정하세요.

**프로젝트 이름 \***

프로젝트 이름을(를) 입력해주세요

**데이터 서버 지역 \*** AWS demo Azure Kakao Huawei 가까운 지역의 서버를 선택하세요 ▼

데이터 서버 지역을(를) 선택해주세요

데이터 서버 지역은 리전(클라우드 서비스를 제공하기 위해 설치한 데이터 센터의 묶음)을 의미합니다. 특정 리전을 선택하면 해당 리전에 속해있는 데이터 센터에 사용자의 데이터가 저장됩니다.

**타임 존 \*** (GMT +9:00) Seoul, Tokyo, Yakutsk ▼

타임존은 알림, 보고서를 생성할 시 기준시간이 됩니다.

**프로젝트 그룹** 그룹 없음 ▼ 그룹 추가

여러개의 프로젝트를 그룹으로 묶어서 관리할 수 있습니다. 생성할 프로젝트가 속할 그룹이 있으면 선택해주세요.

**프로젝트 설명**

5. 모든 설정을 완료한 다음에는 [프로젝트 생성하기](#) 버튼을 선택하세요.

ⓘ • **데이터 서버 지역**은 리전(클라우드 서비스를 제공하기 위해 설치한 데이터 센터의 묶음)을 의미합니다. 특정 리전을 선택하면 해당 리전에 속한 데이터 센터에 사용자의 데이터를 저장합니다.

- ① **타임 존**은 알림, 보고서를 생성하는 기준 시간입니다.
  - 여러 개의 프로젝트를 그룹으로 묶어 관리하려면 **프로젝트 그룹**에서 그룹을 선택하거나 그룹을 추가하세요. 그룹에 대한 자세한 설명은 [다음 문서](#)를 참조하세요.
  - 조직을 선택한 상태에서 프로젝트를 추가할 경우 **조직 하위 그룹**을 필수로 설정해야 합니다.

## 액세스 키 확인

**액세스 키**는 와탭 서비스 활성화를 위한 고유 ID입니다.

설치 안내 섹션에서 **프로젝트 액세스 키 발급받기** 버튼을 선택하세요. **액세스 키**를 자동으로 발급받은 후 다음 단계를 진행합니다.

- ① 프로젝트를 생성한 다음에는 자동으로 **에이전트 설치** 페이지로 이동합니다. **에이전트 설치** 페이지로 이동하지 않는다면 화면 왼쪽 메뉴에서 **관리 > 에이전트 설치**를 선택하세요.

## 에이전트 다운로드

**액세스 키**를 발급 받으면 **에이전트 다운로드** 섹션에서 **다운로드** 버튼이 활성화됩니다.

- 다운로드** 버튼을 선택해 와탭 에이전트 파일을 다운로드하세요.

- ✔ 다음 명령어를 이용해 서버에서 직접 다운로드할 수도 있습니다.

```
wget https://service.whatap.io/agent/whatap.agent.java.tar.gz
```

- 모니터링 대상 서버에 접속하세요.
- 모니터링 대상 서버의 원하는 경로에 에이전트 파일의 압축을 해제하세요. 압축을 해제한 경로가 `$WHATAP_HOME` 이 됩니다.

```
tar -zxvf whatap.agent.java.tar.gz
```

## 에이전트 설치 파일 구성

에이전트 파일은 애플리케이션 모니터링에 필요한 정보를 추출해 와탭 수집 서버로 전달하는 트레이서와 트레이서가 데이터를 보낼 수 있도록 돕는 요소로 구성되어 있습니다.

- [whatap.conf](#)

모니터링 에이전트의 설정 정보, 액세스 키, 와탭 데이터 수집 서버의 IP 정보 등을 저장하는 설정 파일입니다. 자세한 내용은 [다음 문서](#)를 참조하세요.

- [ping.sh\(bat\)](#)

현재 설정한 액세스 키, 와탭 데이터 수집 서버로 통신을 시도해 설정이나 네트워크 이슈가 있는지 확인하기 위한 스크립트(배치) 파일입니다.

- [proxy.sh\(bat\)](#)

[whatap.agent.proxy-X.Y.Z.jar](#) 파일을 시작하기 위한 스크립트(배치) 파일입니다.

- [resmon.sh\(bat\)](#)

CPU, Memory, Disk 정보를 정상 수집하는지 확인하기 위한 스크립트(배치) 파일입니다.

- [whatap-logsink-lz4-X.Y.Z.jar](#)

애플리케이션 서버의 log 정보를 수집하기 위한 plugin 형식의 파일입니다.

- [whatap.agent-X.Y.Z.jar](#)

애플리케이션 서버의 각종 모니터링 정보를 수집 및 수집 서버에 전송하는 모니터링 에이전트 파일입니다.

- [whatap.agent.proxy-X.Y.Z.jar](#)

애플리케이션 서버가 동작하는 서버에서 WhatTap 측 수집 서버에 직접 연결이 불가능한 경우, 그 사이의 Proxy 서버(경유 서버)에서 경유 역할을 하는 애플리케이션입니다.

## whatap.conf 파일 설정하기

### 기본 정보 입력하기

압축을 푼 `$WHATAP_HOME` 경로로 이동하세요. `$WHATAP_HOME/whatap.conf` 파일에 액세스 키와 데이터 수집 서버 IP 주소를 입력하세요. [와탭 모니터링 서비스](#) 화면으로 이동해 [설치 안내](#) 섹션의 [에이전트 다운로드](#) 항목에서 [액세스 키](#), [수집 서버 IP 주소](#) 값을 복사한 다음 붙여넣기하세요.



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

```
license={엑세스 키}
whatap.server.host={수집 서버 IP}
```

! whantap.conf 파일에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

## 보안키 설정하기

SQL 변수, HTTP 쿼리를 조회하거나 Thread 중지 기능 이용하려면 보안키를 설정하세요.

- Java 에이전트 2.2.2 버전 이상

`$WHATAP_HOME` 경로에 `security.conf` 파일을 생성한 후 다음과 같이 6자리 영어, 숫자로 구성된 암호를 입력하세요.

```
security.conf
```

```
paramkey=ABCDEF # SQL 변수 및 HTTP 쿼리 조회
threadkill=ABCDEF # Thread 중지 기능
```

- Java 에이전트 2.2.2 버전 미만

`$WHATAP_HOME` 경로에 `paramkey.txt` 파일을 생성한 후 다음과 같이 6자리 영어, 숫자로 구성된 암호를 입력하세요.

```
paramkey.txt
```

```
ABCDEF # SQL 변수 및 HTTP 쿼리 조회, Thread 중지 기능
```

✔ Java 에이전트 2.2.2 버전 미만에서 2.2.2 버전으로 업데이트하는 경우 기존 `paramkey.txt` 파일의 키값은 `seuciry.conf` 파일의 키값으로 자동 적용됩니다. 예를 들어, `paramkey.txt`에 `FEDCBA`를 사용한 상태에서 2.2.2 버전으로 업데이트했다면, `security.conf` 파일에는 다음과 같이 적용됩니다.

```
paramkey=FEDCBA
threadkill=FEDCBA
```

`paramkey.txt` 파일이 존재하지 않을 경우 `security.conf` 파일의 키값은 `WHATAP`이라는 키값으로 자동 생성됩니다.

- ① • SQL 파라미터 정보 기록과 관련한 에이전트 설정은 [다음 문서](#)를 참조하세요.
- HTTP 파라미터 정보 기록과 관련한 에이전트 설정은 [다음 문서](#)를 참조하세요.

## JVM 옵션 추가

사용하는 애플리케이션 서버 환경에 따른 설정 방법을 다음과 같이 제공합니다. 에이전트를 설치하고 적용하기 전에 [지원 환경](#)을 먼저 확인하세요.

### Tomcat

와탭 모니터링 솔루션을 톰캣(Tomcat) 애플리케이션에 통합하기 위한 JVM(Java Virtual Machine) 옵션 추가 방법에 대한 상세한 안내를 제

### JBoss

와탭 모니터링 솔루션을 JBoss 애플리케이션에 통합하기 위한 JVM(Java Virtual Machine) 옵션 추가 방법에 대한 상세한 안내를 제공합니

### JEUS

와탭 모니터링 솔루션을 JEUS 애플리케이션에 통합하기 위한 JVM(Java Virtual Machine) 옵션 추가 방법에 대한 상세한 안내를 제공합니다.

### WebLogic

와탭 모니터링 솔루션을 웹로직(WebLogic) 애플리케이션에 통합하기 위한 JVM(Java Virtual Machine) 옵션 추가 방법에 대한 상세한 안내

### WebSphere

와탭 모니터링 솔루션을 WebSphere 애플리케이션에 통합하기 위한 JVM(Java Virtual Machine) 옵션 추가 방법에 대한 상세한 안내를 제공

### Resin

와탭 모니터링 솔루션을 Resin 애플리케이션에 통합하기 위한 JVM(Java Virtual Machine) 옵션 추가 방법에 대한 상세한 안내를 제공합니다.

### Liberty

와탭 모니터링 솔루션을 Liberty 애플리케이션에 통합하기 위한 JVM(Java Virtual Machine) 옵션 추가 방법에 대한 상세한 안내를 제공합니다.

### Jetty

와탭 모니터링 솔루션을 Jetty 애플리케이션에 통합하기 위한 JVM(Java Virtual Machine) 옵션 추가 방법에 대한 상세한 안내를 제공합니다.

### Spring Boot

와탭 모니터링 솔루션을 Spring Boot 애플리케이션에 통합하기 위한 JVM(Java Virtual Machine) 옵션 추가 방법에 대한 상세한 안내를 제공합니다.

### Play2

와탭 모니터링 솔루션을 Play2 애플리케이션에 통합하기 위한 JVM(Java Virtual Machine) 옵션 추가 방법에 대한 상세한 안내를 제공합니다.

### IBM BlueMix

IBM Bluemix 클라우드 플랫폼에서 Java 애플리케이션을 실행하는 사용자가 와탭의 자바(Java) 모니터링 에이전트를 효과적으로 적용하기

### Elastic Beanstalk

AWS Elastic Beanstalk 환경에서 Java 애플리케이션을 실행하는 사용자가 와탭의 자바(Java) 모니터링 에이전트를 효과적으로 적용하기 위

## Docker

도커(Docker) 환경에서 Java 애플리케이션에 와탭 자바(Java) 모니터링 에이전트를 적용하는 방법을 단계별로 안내합니다. 사용자는 기존 Do

## 배치 애플리케이션

와탭의 자바(Java) 에이전트를 배치 애플리케이션에 적용하는 방법을 안내합니다. 배치 애플리케이션의 독특한 시작 및 중지 반복 동작으로 인

## Java 1.6 버전 미만

자바(Java) 1.6 버전 미만에서 자바(Java) 에이전트를 적용하는 방법을 안내합니다. Java 1.5와 1.4 버전 각각에 대해 에이전트 설치 및 설정 절

# Tomcat

## 환경 변수 추가

Tomcat 애플리케이션 서버 환경에서 Java 모니터링 에이전트를 효과적으로 사용하기 위해 필요한 JVM 옵션을 추가하는 방법을 안내합니다. 사용하는 운영체제에 맞게 설정하세요.

### Linux

`catalina.sh` 파일 상단에 `JAVA_OPTS` 환경 변수를 추가하세요.

```
{Tomcat_PATH}/bin/catalina.sh
```

```
1 ##### WHATAP START #####
2 WHATAP_HOME=/whatap
3 WHATAP_JAR=`ls ${WHATAP_HOME}/whatap.agent-*.jar | sort -V | tail -1`
4 JAVA_OPTS="${JAVA_OPTS} -javaagent:${WHATAP_JAR} "
5 ##### WHATAP END #####
```

ⓘ `ls ${WHATAP_HOME}/whatap.agent-*.jar | sort -V | tail -1` 항목은 향후 에이전트 업데이트를 진행할 경우 최신 와탭 에이전트를 적용하기 위해서입니다.

### Windows

- `setup.bat` 파일을 통해 실행하는 경우

`catalina.bat` 파일 상단에 `JAVA_OPT` 환경 변수를 추가하세요.

```
{Tomcat_PATH}/bin/catalina.bat
```

```
1 rem ##### WHATAP START #####
2 set WHATAP_HOME=C:\whatap
3 for /f %%f in ('dir /b /on "%WHATAP_HOME%\whatap.agent-*.jar"') do set last=%%f
```

```

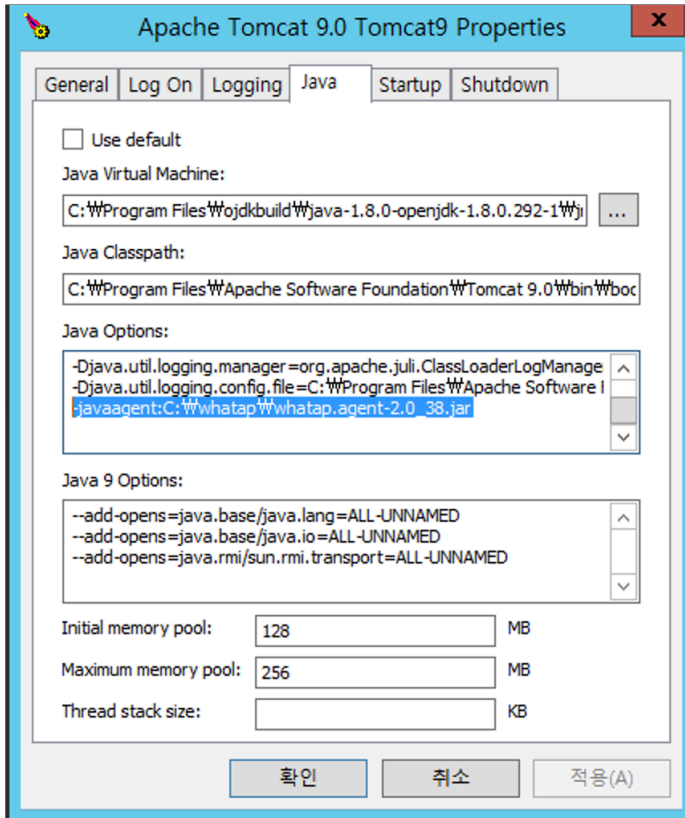
4 set WHATAP_JAR=%last%
5 set WHATAP_OPTS=-javaagent:%WHATAP_HOME%\%WHATAP_JAR%
6
7 if "x%JAVA_OPTS%"=="x" goto setWhatap
8
9 set JAVA_OPTS_TMP=%JAVA_OPTS:"=%
10 if not "x%JAVA_OPTS_TMP:whatap=%"=="x%JAVA_OPTS_TMP%" goto endWhatap
11
12 :setWhatap
13 set JAVA_OPTS=%JAVA_OPTS% %WHATAP_OPTS%
14
15 :endWhatap
16 rem ##### WHATAP END #####

```

- Tomcat을 Windows 인스톨러로 설치한 경우

1. 시작 > Apache Tomcat X.Y.Z. > Configure Tomcat을 선택하세요.
2. Java 탭을 선택하세요.
3. Java Options 텍스트 박스에 -javaagent 옵션을 추가하세요.

```
-javaagent:{와탭 설치 경로}/whatap.agent-X.Y.Z.jar
```



## Java 17 버전 이상

Java 17 버전 이상의 경우 reflection 관련한 다음 옵션을 추가하세요.

```
--add-opens=java.base/java.lang=ALL-UNNAMED
```

## 힙 히스토그램 조회

와탭의 Java 모니터링에서는 JVM 메모리에 올라가 있는 Heap 점유 객체 현황(힙 메모리상의 객체별 사이즈)을 조회할 수 있는 기능을 기본으로 제공합니다. [애플리케이션 > 인스턴스 성능 분석](#) 메뉴에서 **힙 히스토그램** 탭을 선택하세요.

Java 6 ~ 8 버전에서는 JVM 옵션 없이 기본 지원하지만, 일부 Java 버전에 따라 다음과 같이 JVM 옵션을 적용해야 합니다.

- Java 9 ~ Java 15 버전

```
-Djdk.attach.allowAttachSelf=true
```

#### example

```
java -javaagent:{WHATAP_HOME}/whatap.agent-X.Y.Z.jar -Djdk.attach.allowAttachSelf=true -jar {application.jar}
```

- Java 16 버전 이상

```
-Djdk.attach.allowAttachSelf=true
--add-opens=jdk.attach/sun.tools.attach=ALL-UNNAMED
```

#### example

```
java -javaagent:{WHATAP_HOME}/whatap.agent-X.Y.Z.jar -Djdk.attach.allowAttachSelf=true --add-opens=java.base/
java.lang=ALL-UNNAMED --add-opens=jdk.attach/sun.tools.attach=ALL-UNNAMED -jar {application.jar}
```

ⓘ Java 5 버전 이하, IBM Java는 지원하지 않습니다.

## 모니터링 시작하기

모든 설정을 완료한 다음 애플리케이션 서버를 다시 시작하면 에이전트가 정보를 수집하기 시작합니다. [다음 문서](#)를 확인하세요.



# JBoss

## 환경 변수 추가

JBoss 애플리케이션 서버 환경에서 Java 모니터링 에이전트를 효과적으로 사용하기 위해 필요한 JVM 옵션을 추가하는 방법을 안내합니다.

JVM 옵션에 `-javaagent` 및 `-Djboss.modules.system.pkgs` 옵션을 추가하세요.

## JBOS 7.0 이상 EAP 6.0 이상

설정 위치: (JBoss 설치 경로)/bin/standalone.conf(domain.conf)

standalone.sh

```
#!/bin/sh
##### WHATAP #####
WHATAP_HOME=/home/ec2-user/whatap
WHATAP_JAR=`ls ${WHATAP_HOME}/whatap.agent-*.jar | sort -V | tail -1`
JAVA_OPTS="${JAVA_OPTS} -javaagent:${WHATAP_JAR} -Djboss.modules.system.pkgs=whatap"
##### WHATAP #####
```

ⓘ JBoss 5.0 버전 이하는 (JBoss 설치 경로)/bin/run.conf 경로에서 설정하세요. 에러가 발생하면 `-Djboss.platform.mbeanserver=true` 옵션을 추가하세요.

## Java 17 버전 이상

Java 17 버전 이상의 경우 reflection 관련한 다음 옵션을 추가하세요.

```
--add-opens=java.base/java.lang=ALL-UNNAMED
```

## 힙 히스토그램 조회

와탭의 Java 모니터링에서는 JVM 메모리에 올라가 있는 Heap 점유 객체 현황(힙 메모리상의 객체별 사이즈)을 조회할 수 있는 기능을 기본으로 제공합니다. [애플리케이션 > 인스턴스 성능 분석](#) 메뉴에서 [힙 히스토그램](#) 탭을 선택하세요.

Java 6 ~ 8 버전에서는 JVM 옵션 없이 기본 지원하지만, 일부 Java 버전에 따라 다음과 같이 JVM 옵션을 적용해야 합니다.

- Java 9 ~ Java 15 버전

```
-Djdk.attach.allowAttachSelf=true
```

example

```
java -javaagent:{WHATAP_HOME}/whatap.agent-X.Y.Z.jar -Djdk.attach.allowAttachSelf=true -jar {application.jar}
```

- Java 16 버전 이상

```
-Djdk.attach.allowAttachSelf=true
--add-opens=jdk.attach/sun.tools.attach=ALL-UNNAMED
```

example

```
java -javaagent:{WHATAP_HOME}/whatap.agent-X.Y.Z.jar -Djdk.attach.allowAttachSelf=true --add-opens=java.base/
java.lang=ALL-UNNAMED --add-opens=jdk.attach/sun.tools.attach=ALL-UNNAMED -jar {application.jar}
```

ⓘ Java 5 버전 이하, IBM Java는 지원하지 않습니다.

## 모니터링 시작하기

모든 설정을 완료한 다음 애플리케이션 서버를 다시 시작하면 에이전트가 정보를 수집하기 시작합니다. [다음 문서](#)를 확인하세요.

# JEUS

## JVM 옵션 추가

JEUS 애플리케이션 서버 환경에서 Java 모니터링 에이전트를 효과적으로 사용하기 위해 필요한 JVM 옵션을 추가하는 방법을 안내합니다.

1. JEUS 버전에 따른 시작 옵션의 설정 파일의 경로를 참고해 `-javaagent` 옵션을 추가하세요.

**JEUS 7**

JEUS 6

\$JEUS\_HOME/domains/jeus\_domain/config.xml

```

1 <domain>
2   <servers>
3     <server>
4       <name>server1</name>
5       <jvm-config>
6         <jvm-option>
7           -Xmx1024m -XX:MaxPermSize=128m
8           -javaagent:/whatap/whatap.agent-X.Y.Z.jar
9         </jvm-option>
10      </jvm-config>
11    </server>
12  </servers>
13  ...
14 </domain>

```

\$JEUS\_HOME/config/\$hostname/JEUSMain.xml

```

1 <node>
2   <name>node01</name>
3   <engine-container1>
4     <name>container1</name>
5     ...
6     <command-option>-Xmx1024m -XX:MaxPermSize=128m

```

```

7   -javaagent:/whatap/whatap.agent-#.#.#.jar
8   </command-option>
9   ...
10  </engine-container1>
11  </node>

```

2. 애플리케이션 서버를 다시 시작하세요.

```
jdown && jboot
```

3. 애플리케이션 서버 로그 파일과 에이전트 로그 파일을 통해 에이전트가 정상 작동하는지, 에러가 발생하지 않았는지 확인하세요. 로그 파일의 위치는 다음을 참고하세요.
  - 에이전트: `$WHATAP_HOME/logs/whatap-{SERVER_NAME}-{DATE}.log`
  - JEUS 7: `$JEUS_HOME/domains/$HOST_NAME/servers/$NODE_NAME/logs/JeusServer.log`
  - JEUS 6: `$JEUS_HOME/logs/$NODE_NAME/JeusServer.log`
4. 에이전트가 애플리케이션 서버의 종류와 서비스 컨테이너 명을 인식했는지 확인하세요.

① [와탭 모니터링 서비스](#)에 로그인한 다음 프로젝트를 선택하세요. [애플리케이션](#) > [인스턴스 성능 분석](#) 메뉴를 선택한 다음 [실행 환경 변수](#) 탭을 선택하세요. `whatap.name`과 `whatap.type` 항목을 확인하세요. `whatap.name` 항목의 마지막 요소가 컨테이너 이름이어야 합니다. `whatap.type` 항목에는 애플리케이션 서버의 종류가 명시되어야 합니다.



Boot Environment (JU-2-15-server1)

name	detail
java.version	1.7.0_80
java.vendor	Oracle Corporation
java.home	/home/vagrant/jdk1.7.0_80/jre
os.name	Linux
file.encoding	UTF-8
user.country	US
user.timezone	Etc/UTC
user.language	en
user.home	/home/vagrant
user.dir	/home/vagrant/jeus7/bin
whatap.jvm.cmd	jeus.server.ServerBootstrapper -domain ubuntu1204 -server server1 -u administrator -independent
-Xmx	1024m
whatap.boot.start	agentmain
← java.name	4540@ubuntu1204
java.start	1477905163443
whatap.pid	4540
whatap.home	/home/vagrant/whatap
whatap.ip	10.0.2.15
whatap.name	{type}-{ip2}-{ip3}-server1
whatap.version	0.3.0 20161021
whatap.type	JEUS

ⓘ Java 17 버전 이상의 경우 reflection 관련한 다음 옵션을 추가하세요.

```
--add-opens=java.base/java.lang=ALL-UNNAMED
```

## 힙 히스토그램 조회

와탭의 Java 모니터링에서는 JVM 메모리에 올라가 있는 Heap 점유 객체 현황(힙 메모리상의 객체별 사이즈)을 조회할 수 있는 기능을 기본으로 제공합니다. [애플리케이션 > 인스턴스 성능 분석](#) 메뉴에서 **힙 히스토그램** 탭을 선택하세요.

Java 6 ~ 8 버전에서는 JVM 옵션 없이 기본 지원하지만, 일부 Java 버전에 따라 다음과 같이 JVM 옵션을 적용해야 합니다.

- Java 9 ~ Java 15 버전

```
-Djdk.attach.allowAttachSelf=true
```

example

```
java -javaagent:{WHATAP_HOME}/whatap.agent-X.Y.Z.jar -Djdk.attach.allowAttachSelf=true -jar {application.jar}
```

- Java 16 버전 이상

```
-Djdk.attach.allowAttachSelf=true
--add-opens=jdk.attach/sun.tools.attach=ALL-UNNAMED
```

example

```
java -javaagent:{WHATAP_HOME}/whatap.agent-X.Y.Z.jar -Djdk.attach.allowAttachSelf=true --add-opens=java.base/
java.lang=ALL-UNNAMED --add-opens=jdk.attach/sun.tools.attach=ALL-UNNAMED -jar {application.jar}
```

ⓘ Java 5 버전 이하, IBM Java는 지원하지 않습니다.

# WebLogic

## JVM 옵션 추가

WebLogic 애플리케이션 서버 환경에서 Java 모니터링 에이전트를 효과적으로 사용하기 위해 필요한 JVM 옵션을 추가하는 방법을 안내합니다.

사용하는 운영체제를 확인 후 설정을 완료하세요.

Linux

Windows

```
{WebLogic_Path}/user_projects/domains/{User_Domain}/bin/startWebLogic.sh
```

```
##### WHATAP START #####
WHATAP_HOME=/path/to/whatap
WHATAP_JAR=`ls ${WHATAP_HOME}/whatap.agent-*.jar | sort -V | tail -1`
JAVA_OPTIONS="{JAVA_OPTIONS} -javaagent:${WHATAP_JAR}"
##### WHATAP END #####
```

```
{WebLogic_Path}/user_projects/domains/{User_Domain}/bin/startWebLogic.bat
```

```
rem ##### WHATAP START #####
set WHATAP_HOME=\path\to\whatap
for /f %%f in ('dir /b /on "%WHATAP_HOME%\whatap.agent-*.jar"') do set last=%%f
set WHATAP_JAR=%last%
set WHATAP_OPTS=-javaagent:%WHATAP_HOME%\%WHATAP_JAR%
if "x%JAVA_OPTIONS%"=="x" goto setWhatap
set JAVA_OPTIONS_TMP=%JAVA_OPTIONS:="=%
if not "x%JAVA_OPTIONS_TMP:whatap="=="x%JAVA_OPTIONS_TMP%" goto endWhatap
:setWhatap
set JAVA_OPTIONS=%JAVA_OPTIONS% %WHATAP_OPTS%
:endWhatap
rem ##### WHATAP END #####
```

ⓘ Java 17 버전 이상의 경우 reflection 관련한 다음 옵션을 추가하세요.



```
--add-opens=java.base/java.lang=ALL-UNNAMED
```

## 힙 히스토그램 조회

와탭의 Java 모니터링에서는 JVM 메모리에 올라가 있는 Heap 점유 객체 현황(힙 메모리상의 객체별 사이즈)을 조회할 수 있는 기능을 기본으로 제공합니다. [애플리케이션 > 인스턴스 성능 분석](#) 메뉴에서 [힙 히스토그램](#) 탭을 선택하세요.

Java 6 ~ 8 버전에서는 JVM 옵션 없이 기본 지원하지만, 일부 Java 버전에 따라 다음과 같이 JVM 옵션을 적용해야 합니다.

- Java 9 ~ Java 15 버전

```
-Djdk.attach.allowAttachSelf=true
```

### example

```
java -javaagent:{WHATAP_HOME}/whatap.agent-X.Y.Z.jar -Djdk.attach.allowAttachSelf=true -jar {application.jar}
```

- Java 16 버전 이상

```
-Djdk.attach.allowAttachSelf=true
--add-opens=jdk.attach/sun.tools.attach=ALL-UNNAMED
```

### example

```
java -javaagent:{WHATAP_HOME}/whatap.agent-X.Y.Z.jar -Djdk.attach.allowAttachSelf=true --add-opens=java.base/
java.lang=ALL-UNNAMED --add-opens=jdk.attach/sun.tools.attach=ALL-UNNAMED -jar {application.jar}
```

Java 5 버전 이하, IBM Java는 지원하지 않습니다.

## 모니터링 시작하기

모든 설정을 완료한 다음 애플리케이션 서버를 다시 시작하면 에이전트가 정보를 수집하기 시작합니다. [다음 문서](#)를 확인하세요.

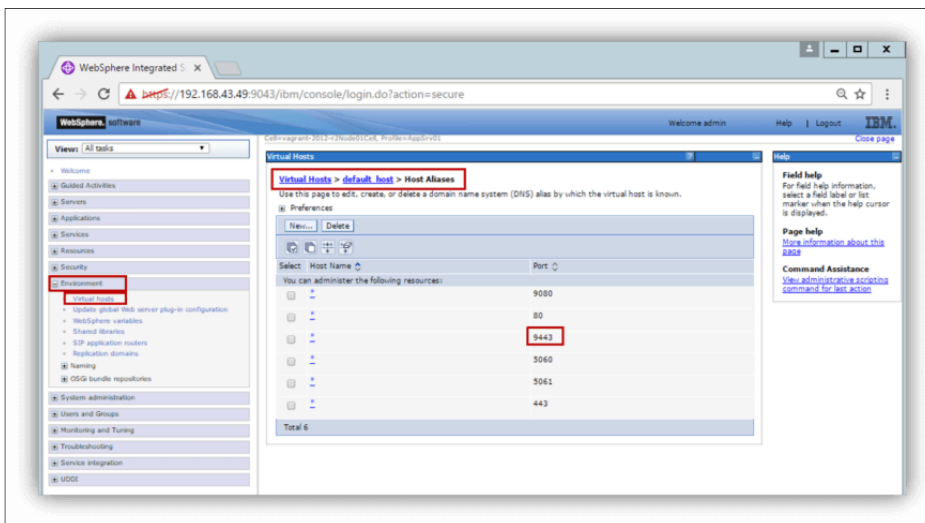


# WebSphere

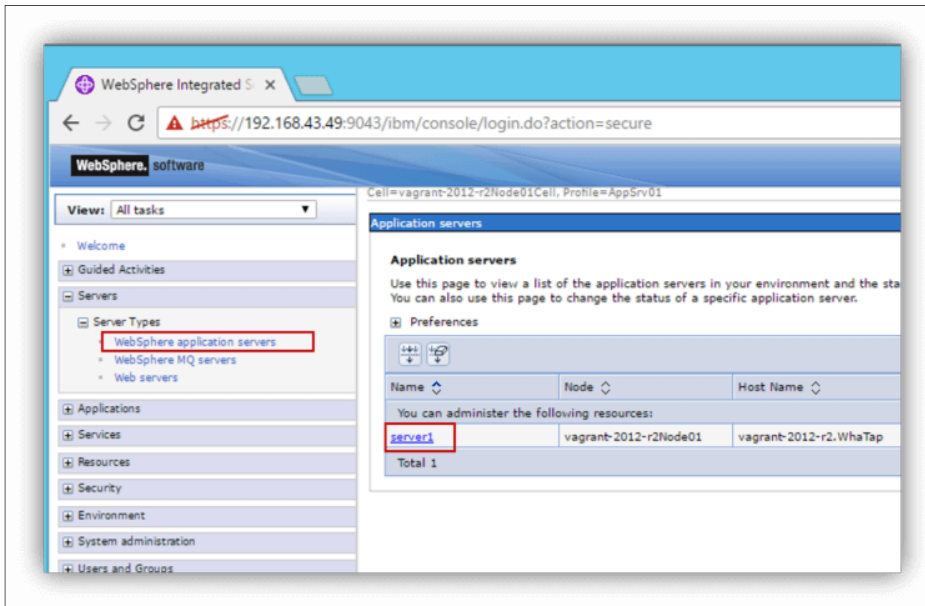
## JVM 옵션 추가

WebSphere 애플리케이션 서버 환경에서 Java 모니터링 에이전트를 효과적으로 사용하기 위해 필요한 JVM 옵션을 추가하는 방법을 안내합니다. JVM 옵션을 추가하려면 관리 콘솔(Admin console)로 접속하세요.

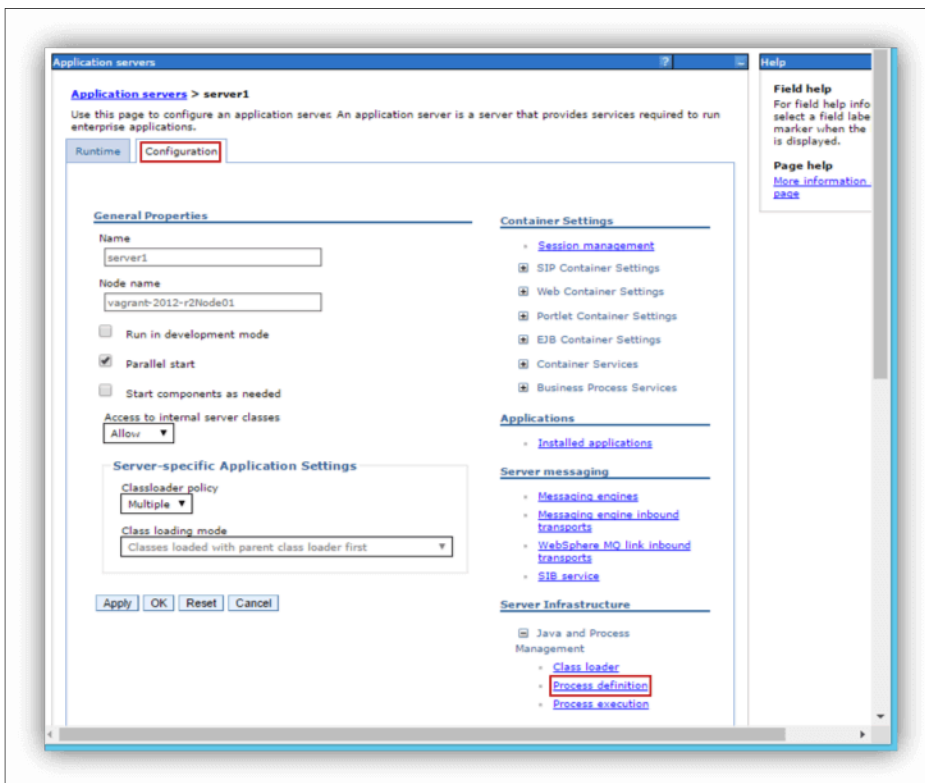
1. 웹브라우저에서 관리 콘솔에 로그인하세요.
2. [Environment](#) > [Virtual Hosts](#) > [default host](#) > [Host Aliases](#) 항목을 선택한 다음 에이전트를 설치할 서버의 포트 번호를 확인하세요.



3. [Servers](#) > [Server Types](#) > [Websphere application servers](#) 항목으로 이동한 다음 에이전트를 설치할 서버를 선택하세요.

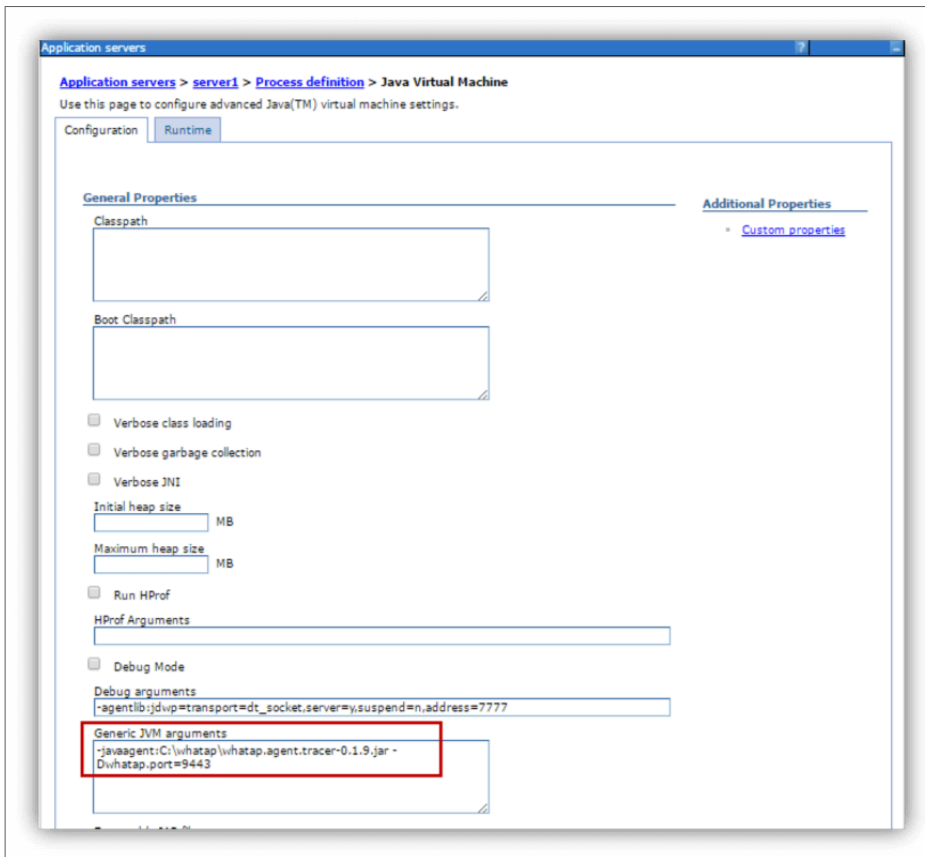


4. 선택한 서버 화면에서 **Configuration** 탭을 선택하고 **Server Infrastructure** 섹션의 **Java and Process Management > Process definition** 메뉴를 선택하세요.



5. 오른쪽 메뉴에서 [Additional Properties](#) > [Java Virtual Machine](#) 메뉴를 선택하세요.
6. [Configuration](#) 탭에서 [Generic JVM arguments](#) 텍스트 박스에 `-javaagent` 와 `-Dwhatap.port` 를 추가하세요.

```
-javaagent:/whatap/whatap.agent-X.Y.Z.jar
-Dwhatap.port={포트번호} ## 2번 항목 포트 번호 ##
```



ⓘ Java 17 버전 이상의 경우 reflection 관련한 다음 옵션을 추가하세요.

```
--add-opens=java.base/java.lang=ALL-UNNAMED
```

## 힙 히스토그램 조회

와탭의 Java 모니터링에서는 JVM 메모리에 올라가 있는 Heap 점유 객체 현황(힙 메모리상의 객체별 사이즈)을 조회할 수 있는 기능을 기본으로 제공합니다. [애플리케이션 > 인스턴스 성능 분석](#) 메뉴에서 [힙 히스토그램](#) 탭을 선택하세요.

Java 6 ~ 8 버전에서는 JVM 옵션 없이 기본 지원하지만, 일부 Java 버전에 따라 다음과 같이 JVM 옵션을 적용해야 합니다.

- Java 9 ~ Java 15 버전

```
-Djdk.attach.allowAttachSelf=true
```

#### example

```
java -javaagent:{WHATAP_HOME}/whatap.agent-X.Y.Z.jar -Djdk.attach.allowAttachSelf=true -jar {application.jar}
```

- Java 16 버전 이상

```
-Djdk.attach.allowAttachSelf=true
--add-opens=jdk.attach/sun.tools.attach=ALL-UNNAMED
```

#### example

```
java -javaagent:{WHATAP_HOME}/whatap.agent-X.Y.Z.jar -Djdk.attach.allowAttachSelf=true --add-opens=java.base/
java.lang=ALL-UNNAMED --add-opens=jdk.attach/sun.tools.attach=ALL-UNNAMED -jar {application.jar}
```

ⓘ Java 5 버전 이하, IBM Java는 지원하지 않습니다.

## 모니터링 시작하기

모든 설정을 완료한 다음 애플리케이션 서버를 다시 시작하면 에이전트가 정보를 수집하기 시작합니다. [다음 문서](#)를 확인하세요.

# Resin

## JVM 옵션 추가

Resin을 사용하는 애플리케이션 서버 환경에서 JVM 옵션을 추가하는 방법을 확인하세요. Java 버전에 따라 적용해야 할 JVM 옵션이 다를 수 있습니다. 사용자의 환경에 맞는 JVM 옵션을 적용하세요. 옵션을 적용한 다음 애플리케이션 서버를 다시 시작하세요.

- 옵션 1: `resin.properties` 파일에 `jvm_args` 속성을 사용해 `javaagent` 인수를 추가하세요.
- 옵션 2: `conf/resin.conf` 또는 `conf/resin.xml` 파일에 `<jvm-args>` 섹션을 추가해 `-javaagent` 인수를 설정하세요.

- ❗ • 애플리케이션 서버 로그 파일과 에이전트 로그 파일을 통해 에이전트가 정상 작동하는지, 에러가 발생하지 않았는지 확인하세요. 로그 파일의 위치는 다음을 확인하세요.
- 에이전트: `$WHATAP_HOME/logs/whatap-{SERVER_NAME}-{DATE}.log`
  - RESIN 4.x: `$RESIN_HOME/log/jvm-app-#.log`
- 에이전트가 정상 작동하지 않거나 에러가 발생한다면 [다음 문서](#)를 확인하세요.

## Java 6 ~ 8 버전

```
resin.properties
```

```
jvm_args : -javaagent:{WHATAP_HOME}/whatap.agent-X.Y.Z.jar
```

```
resin.conf | resin.xml
```

```
<jvm-arg>-javaagent:{WHATAP_HOME}/whatap.agent-X.Y.Z.jar</jvm-arg>
```

## Java 9 ~ 15 버전

```
resin.properties
```

```
jvm_args : -javaagent:{WHATAP_HOME}/whatap.agent-X.Y.Z.jar -Djdk.attach.allowAttachSelf=true
```

```
resin.conf | resin.xml
```

```
<jvm-arg>javaagent:{WHATAP_HOME}/whatap.agent-X.Y.Z.jar</jvm-arg>
<jvm-arg>-Djdk.attach.allowAttachSelf=true</jvm-arg>
```

## Java 16 버전 이상

```
resin.properties
```

```
jvm_args : -javaagent:{WHATAP_HOME}/whatap.agent-X.Y.Z.jar -Djdk.attach.allowAttachSelf=true --add-opens=java.base/
java.lang=ALL-UNNAMED --add-opens=jdk.attach/sun.tools.attach=ALL-UNNAMED
```

```
resin.conf | resin.xml
```

```
<jvm-arg>javaagent:{WHATAP_HOME}/whatap.agent-X.Y.Z.jar</jvm-arg>
<jvm-arg>-Djdk.attach.allowAttachSelf=true</jvm-arg>
<jvm-arg>--add-opens=java.base/java.lang=ALL-UNNAMED</jvm-arg> <!-- Java 17 or later -->
<jvm-arg>--add-opens=jdk.attach/sun.tools.attach=ALL-UNNAMED</jvm-arg>
```

## 힉 히스토그램 조회

와탭의 Java 모니터링에서는 JVM 메모리에 올라가 있는 Heap 점유 객체 현황(힉 메모리상의 객체별 사이즈)을 조회할 수 있는 기능을 기본으로 제공합니다. [애플리케이션](#) > [인스턴스 성능 분석](#) 메뉴에서 [힉 히스토그램](#) 탭을 선택하세요.

Java 6 ~ 8 버전에서는 JVM 옵션 없이 기본 지원하지만, 일부 Java 버전에 따라 다음과 같이 JVM 옵션을 적용해야 합니다.

- Java 9 ~ Java 15 버전

```
-Djdk.attach.allowAttachSelf=true
```

```
example
```

```
java -javaagent:{WHATAP_HOME}/whatap.agent-X.Y.Z.jar -Djdk.attach.allowAttachSelf=true -jar {application.jar}
```

- Java 16 버전 이상

```
-Djdk.attach.allowAttachSelf=true
--add-opens=jdk.attach/sun.tools.attach=ALL-UNNAMED
```

#### example

```
java -javaagent:{WHATAP_HOME}/whatap.agent-X.Y.Z.jar -Djdk.attach.allowAttachSelf=true --add-opens=java.base/
java.lang=ALL-UNNAMED --add-opens=jdk.attach/sun.tools.attach=ALL-UNNAMED -jar {application.jar}
```

ⓘ Java 5 버전 이하, IBM Java는 지원하지 않습니다.

## 모니터링 시작하기

모든 설정을 완료한 다음 애플리케이션 서버를 다시 시작하면 에이전트가 정보를 수집하기 시작합니다. [다음 문서](#)를 확인하세요.



# Liberty

## JVM 옵션 추가

Liberty 애플리케이션 서버 환경에서 Java 모니터링 에이전트를 효과적으로 사용하기 위해 필요한 JVM 옵션을 추가하는 방법을 안내합니다.

`jvm.options` 파일에 에이전트 옵션을 추가하세요. 파일이 없는 경우 신규로 생성하세요.

```
/wasliberty/usr/servers/defaultServer/jvm.options
```

```
1 -javaagent:app/whatap/whatap.agent-2.1.1.jar
2 -Dorg.osgi.framework.bootdelegation=whatap.* ## OSGI 프레임워크 설정을 위한 추가 옵션 ##
```

ⓘ Java 17 버전 이상의 경우 reflection 관련한 다음 옵션을 추가하세요.

```
--add-opens=java.base/java.lang=ALL-UNNAMED
```

## 힙 히스토그램 조회

와탭의 Java 모니터링에서는 JVM 메모리에 올라가 있는 Heap 점유 객체 현황(힙 메모리상의 객체별 사이즈)을 조회할 수 있는 기능을 기본으로 제공합니다. [애플리케이션](#) > [인스턴스 성능 분석](#) 메뉴에서 [힙 히스토그램](#) 탭을 선택하세요.

Java 6 ~ 8 버전에서는 JVM 옵션 없이 기본 지원하지만, 일부 Java 버전에 따라 다음과 같이 JVM 옵션을 적용해야 합니다.

- Java 9 ~ Java 15 버전

```
-Djdk.attach.allowAttachSelf=true
```

```
example
```

```
java -javaagent:{WHATAP_HOME}/whatap.agent-X.Y.Z.jar -Djdk.attach.allowAttachSelf=true -jar {application.jar}
```

- Java 16 버전 이상

```
-Djdk.attach.allowAttachSelf=true
--add-opens=jdk.attach/sun.tools.attach=ALL-UNNAMED
```

#### example

```
java -javaagent:{WHATAP_HOME}/whatap.agent-X.Y.Z.jar -Djdk.attach.allowAttachSelf=true --add-opens=java.base/
java.lang=ALL-UNNAMED --add-opens=jdk.attach/sun.tools.attach=ALL-UNNAMED -jar {application.jar}
```

ⓘ Java 5 버전 이하, IBM Java는 지원하지 않습니다.

## 모니터링 시작하기

모든 설정을 완료한 다음 애플리케이션 서버를 다시 시작하면 에이전트가 정보를 수집하기 시작합니다. [다음 문서](#)를 확인하세요.

# Jetty

## JVM 옵션 추가

Jetty 애플리케이션 서버 환경에서 Java 모니터링 에이전트를 효과적으로 사용하기 위해 필요한 JVM 옵션을 추가하는 방법을 안내합니다.

- `jetty.sh` 파일에 JVM 옵션 추가

```
$JETTY_HOME/bin/jetty.sh
```

```
1 ...
2 JAVA_OPTIONS+=("-javaagent:/home/vagrant/whatap/whatap.agent-X.Y.Z.jar -Dwhatap.port=8080")
3 usage()
4 {
5     echo "Usage: ${0##*/} [-d] {start|stop|run|restart|check|supervise} [ CONFIGS ... ]"
6     exit 1
7 }
8 [ $# -gt 0 ] || usage
9 ...
```

- 셸 스크립트 없이 Jetty 시작하기

```
$ java -javaagent:/home/vagrant/whatap/whatap.agent-X.Y.Z.jar -Dwhatap.port=8080 -jar start.jar &
```

ⓘ Java 17 버전 이상의 경우 reflection 관련한 다음 옵션을 추가하세요.

```
--add-opens=java.base/java.lang=ALL-UNNAMED
```

## 힙 히스토그램 조회

와탭의 Java 모니터링에서는 JVM 메모리에 올라가 있는 Heap 점유 객체 현황(힙 메모리상의 객체별 사이즈)을 조회할 수 있는

기능을 기본으로 제공합니다. [애플리케이션 > 인스턴스 성능 분석](#) 메뉴에서 [힙 히스토그램](#) 탭을 선택하세요.

Java 6 ~ 8 버전에서는 JVM 옵션 없이 기본 지원하지만, 일부 Java 버전에 따라 다음과 같이 JVM 옵션을 적용해야 합니다.

- Java 9 ~ Java 15 버전

```
-Djdk.attach.allowAttachSelf=true
```

example

```
java -javaagent:{WHATAP_HOME}/whatap.agent-X.Y.Z.jar -Djdk.attach.allowAttachSelf=true -jar {application.jar}
```

- Java 16 버전 이상

```
-Djdk.attach.allowAttachSelf=true
--add-opens=jdk.attach/sun.tools.attach=ALL-UNNAMED
```

example

```
java -javaagent:{WHATAP_HOME}/whatap.agent-X.Y.Z.jar -Djdk.attach.allowAttachSelf=true --add-opens=java.base/
java.lang=ALL-UNNAMED --add-opens=jdk.attach/sun.tools.attach=ALL-UNNAMED -jar {application.jar}
```

ⓘ Java 5 버전 이하, IBM Java는 지원하지 않습니다.

## 모니터링 시작하기

모든 설정을 완료한 다음 애플리케이션 서버를 다시 시작하면 에이전트가 정보를 수집하기 시작합니다. [다음 문서](#)를 확인하세요.

# Spring Boot

Spring Boot를 사용하는 애플리케이션 서버 환경에서 JVM 옵션을 추가하는 방법을 확인하세요.

## JVM 옵션 추가

- Java 17 버전 이상의 경우 reflection 관련 JVM 옵션 추가로 애플리케이션을 기동할 수 있습니다.

```
# Java 17 이상만 지원
java -javaagent:{WHATAP_HOME}/whatap.agent-X.Y.Z.jar -Dwhatap.ename={Agent_Name} --add-opens=java.base/
java.lang=ALL-UNNAMED -jar {application.jar}
```

- 시작 스크립트에 JVM 옵션으로 `-javaagent` 를 추가하세요.

```
java -javaagent:{WHATAP_HOME}/whatap.agent-X.Y.Z.jar -jar {application.jar}
```

JVM 옵션을 추가한 다음 사용하는 Spring Boot의 버전과 에이전트 버전에 맞춰 적합한 에이전트 설정을 진행하세요.

## Spring Boot 3

### Spring Boot 3.0.0 버전 이상

- Java 에이전트 v2.2.9 버전 이상([whatap.agent-2.2.9.jar](#))

```
whatap.conf
-----
weaving=spring-boot-3.0
```

- Java 에이전트 v2.2.5 버전 이상([whatap.agent-2.2.5.jar](#))

```
whatap.conf
-----
weaving=spring-boot-3.0,tomcat10
```

- Java 에이전트 v2.2.4 버전 이하([whatap.agent-2.2.4.jar](#))

- `jakarta.servlet.http.HttpServlet` 추적(동기 방식)

```
whatap.conf
```

```
weaving=tomcat10
```

- WebClient 추적(비동기 방식)

```
whatap.conf
```

```
weaving=webflux-6.0
```

## Spring Boot 2

### Spring Boot 2.7.0 버전 이상

- Java 에이전트 v2.2.9 버전 이상([whatap.agent-2.2.9.jar](#))

```
whatap.conf
```

```
weaving=spring-boot-2.7
```

- Java 에이전트 v2.2.5 버전 이상([whatap.agent-2.2.5.jar](#))

```
whatap.conf
```

```
weaving=spring-boot-2.7,tomcat9
```

- Java 에이전트 v2.2.4 버전 이하([whatap.agent-2.2.4.jar](#))

WebClient 추적(비동기 방식)

```
whatap.conf
```

```
weaving=webflux-5.3
```

## Spring Boot 2.5.0 버전 이상

- Java 에이전트 v2.2.9 버전 이상([whatap.agent-2.2.9.jar](#))

```
whatap.conf
```

```
weaving=spring-boot-2.5
```

- Java 에이전트 v2.2.5 버전 이상([whatap.agent-2.2.5.jar](#))

```
whatap.conf
```

```
weaving=spring-boot-2.5,tomcat9
```

- Java 에이전트 v2.2.4 버전 이하([whatap.agent-2.2.4.jar](#))

WebClient 추적(비동기 방식)

```
whatap.conf
```

```
weaving=webflux-5.3
```

## 힙 히스토그램 조회

와탭의 Java 모니터링에서는 JVM 메모리에 올라가 있는 Heap 점유 객체 현황(힙 메모리상의 객체별 사이즈)을 조회할 수 있는 기능을 기본으로 제공합니다. [애플리케이션](#) > [인스턴스 성능 분석](#) 메뉴에서 [힙 히스토그램](#) 탭을 선택하세요.

Java 6 ~ 8 버전에서는 JVM 옵션 없이 기본 지원하지만, 일부 Java 버전에 따라 다음과 같이 JVM 옵션을 적용해야 합니다.

- Java 9 ~ Java 15 버전

```
-Djdk.attach.allowAttachSelf=true
```

```
example
```

```
java -javaagent:{WHATAP_HOME}/whatap.agent-X.Y.Z.jar -Djdk.attach.allowAttachSelf=true -jar {application.jar}
```

- Java 16 버전 이상

```
-Djdk.attach.allowAttachSelf=true  
--add-opens=jdk.attach/sun.tools.attach=ALL-UNNAMED
```

#### example

```
java -javaagent:{WHATAP_HOME}/whatap.agent-X.Y.Z.jar -Djdk.attach.allowAttachSelf=true --add-opens=java.base/  
java.lang=ALL-UNNAMED --add-opens=jdk.attach/sun.tools.attach=ALL-UNNAMED -jar {application.jar}
```

ⓘ Java 5 버전 이하, IBM Java는 지원하지 않습니다.

## 모니터링 시작하기

모든 설정을 완료한 다음 애플리케이션 서버를 다시 시작하면 에이전트가 정보를 수집하기 시작합니다. [다음 문서](#)를 확인하세요.



# Play2

## JVM 옵션 추가

Play2 애플리케이션 서버 환경에서 Java 모니터링 에이전트를 효과적으로 사용하기 위해 필요한 JVM 옵션을 추가하는 방법을 안내합니다.

`$PLAY_HOME/framework/build` 파일에 에이전트 옵션을 설정하세요. 별도로 이용하는 시작 또는 중지 스크립트가 있다면 해당 파일에 적용하세요.

```
$PLAY_HOME/framework/build
```

```
1 #!/usr/bin/env sh
2
3 ##### WHATAP START #####
4 WHATAP_HOME=/apps/whatap
5 WHATAP_JAR=`ls ${WHATAP_HOME}/whatap.agent-*.jar | sort -V | tail -1`
6 JAVA_OPTS="${JAVA_OPTS} -Dwhatap.play2=2.2.6 -javaagent:${WHATAP_JAR}"
7 ##### WHATAP END #####
8
9 if [ -z "${PLAY_VERSION}" ]; then
10   PLAY_VERSION="2.2.6"
11 ...
```

ⓘ Play는 기본 설정 외 `-Dwhatap.play2=2.2.6` 옵션을 추가해야 합니다.

ⓘ Java 17 버전 이상의 경우 reflection 관련한 다음 옵션을 추가하세요.

```
--add-opens=java.base/java.lang=ALL-UNNAMED
```

## 힙 히스토그램 조회

와탭의 Java 모니터링에서는 JVM 메모리에 올라가 있는 Heap 점유 객체 현황(힙 메모리상의 객체별 사이즈)을 조회할 수 있는 기능을 기본으로 제공합니다. [애플리케이션 > 인스턴스 성능 분석](#) 메뉴에서 [힙 히스토그램](#) 탭을 선택하세요.

Java 6 ~ 8 버전에서는 JVM 옵션 없이 기본 지원하지만, 일부 Java 버전에 따라 다음과 같이 JVM 옵션을 적용해야 합니다.

- Java 9 ~ Java 15 버전

```
-Djdk.attach.allowAttachSelf=true
```

example

```
java -javaagent:{WHATAP_HOME}/whatap.agent-X.Y.Z.jar -Djdk.attach.allowAttachSelf=true -jar {application.jar}
```

- Java 16 버전 이상

```
-Djdk.attach.allowAttachSelf=true
--add-opens=jdk.attach/sun.tools.attach=ALL-UNNAMED
```

example

```
java -javaagent:{WHATAP_HOME}/whatap.agent-X.Y.Z.jar -Djdk.attach.allowAttachSelf=true --add-opens=java.base/
java.lang=ALL-UNNAMED --add-opens=jdk.attach/sun.tools.attach=ALL-UNNAMED -jar {application.jar}
```

ⓘ Java 5 버전 이하, IBM Java는 지원하지 않습니다.

## 모니터링 시작하기

모든 설정을 완료한 다음 애플리케이션 서버를 다시 시작하면 에이전트가 정보를 수집하기 시작합니다. [다음 문서](#)를 확인하세요.

# IBM BlueMix

## 에이전트 적용하기

IBM BlueMix를 이용해 배포할 경우 와탭 에이전트를 적용하는 방법입니다.

IBM BlueMix는 컨테이너로 **WebSphere Liberty** 환경을 제공합니다. Liberty는 **WebSphere Application Server**와 다른 경량화 환경으로 **Spring Boot**가 동작하는 방식과 유사합니다.

❗ **Platform as a Service(PaaS)** 환경에서는 배포 대상 애플리케이션에 와탭 에이전트의 설정을 포함해 배포합니다.

다음은 설정 환경 예제입니다. [가이드](#)에 따른 환경을 구성할 경우 로컬 개발 환경에서 생성하는 파일들입니다.

```
whatap@vmwas01:/apps/bluemix/java-helloworld$ ls -alrt
합계 64
drwxrwxr-x 3 whatap whatap 4096 10월 29 13:13 ..
-rw-rw-r-- 1 whatap whatap 1079 10월 29 13:13 .classpath
-rw-rw-r-- 1 whatap whatap 1184 10월 29 13:13 .project
-rw-rw-r-- 1 whatap whatap 39 10월 29 13:13 .gitignore
-rw-rw-r-- 1 whatap whatap 151 10월 29 13:13 CONTRIBUTING.md
drwxrwxr-x 2 whatap whatap 4096 10월 29 13:13 .settings
-rw-rw-r-- 1 whatap whatap 2823 10월 29 13:13 pom.xml
-rw-rw-r-- 1 whatap whatap 122 10월 29 13:13 manifest.yml
-rw-rw-r-- 1 whatap whatap 3522 10월 29 13:13 README.md
-rw-rw-r-- 1 whatap whatap 11323 10월 29 13:13 LICENSE
drwxrwxr-x 3 whatap whatap 4096 10월 29 13:13 src
drwxrwxr-x 2 whatap whatap 4096 10월 29 13:13 target
drwxrwxr-x 8 whatap whatap 4096 10월 29 13:13 .git
drwxrwxr-x 6 whatap whatap 4096 10월 29 15:26 .
```

1. `${APP_HOME}` 에서 `src/main/resources/whatap-agent/` 디렉터리를 생성하고 jar 파일, conf 파일을 복사하세요.

```
$ mkdir -p src/main/resources/whatap-agent/
$ cp /apps/whatap/whatap.agent.tracer-1.5.4.jar src/main/resources/whatap-agent/
$ cp /apps/whatap/whatap.conf src/main/resources/whatap-agent/
```

2. `${APP_HOME}` /`manifest.yml` 파일에 옵션을 추가하세요. yml 파일이므로 공백, 들여쓰기 기준을 잘 맞춰 작성하세요.

```

${APP_HOME}/manifest.yml

```

```

1 ---
2 applications:
3   - name: sample-java-helloworld
4     random-route: true
5     memory: 256M
6     path: target/JavaHelloWorldApp.war
7     # 여기서부터 추가합니다.
8     env:
9       JAVA_OPTS: "-javaagent:{{APPLICATION_DIR}}/WEB-INF/classes/whatap-agent/whatap.agent-X.Y.Z.jar
                  -Dorg.osgi.framework.bootdelegation=whatap.* "

```

- ⓘ Java 17 버전 이상의 경우 reflection 관련한 다음 옵션을 추가하세요.

```
--add-opens=java.base/java.lang=ALL-UNNAMED
```

- ⓘ • [whatap.conf](#) 설정은 PaaS가 아닌 환경과 동일하게 적용합니다. 적용 후 에이전트 명 식별에 어려울 수 있으니 상황에 맞는 에이전트 명을 적용하세요.
- 에이전트 네이밍에 관한 자세한 내용은 [다음 문서](#)를 참조하세요.

## 힙 히스토그램 조회

와탭의 Java 모니터링에서는 JVM 메모리에 올라가 있는 Heap 점유 객체 현황(힙 메모리상의 객체별 사이즈)을 조회할 수 있는 기능을 기본으로 제공합니다. [애플리케이션 > 인스턴스 성능 분석](#) 메뉴에서 [힙 히스토그램](#) 탭을 선택하세요.

Java 6 ~ 8 버전에서는 JVM 옵션 없이 기본 지원하지만, 일부 Java 버전에 따라 다음과 같이 JVM 옵션을 적용해야 합니다.

- Java 9 ~ Java 15 버전

```
-Djdk.attach.allowAttachSelf=true
```

**example**

```
java -javaagent:{WHATAP_HOME}/whatap.agent-X.Y.Z.jar -Djdk.attach.allowAttachSelf=true -jar {application.jar}
```

- Java 16 버전 이상

```
-Djdk.attach.allowAttachSelf=true
--add-opens=jdk.attach/sun.tools.attach=ALL-UNNAMED
```

**example**

```
java -javaagent:{WHATAP_HOME}/whatap.agent-X.Y.Z.jar -Djdk.attach.allowAttachSelf=true --add-opens=java.base/
java.lang=ALL-UNNAMED --add-opens=jdk.attach/sun.tools.attach=ALL-UNNAMED -jar {application.jar}
```

ⓘ Java 5 버전 이하, IBM Java는 지원하지 않습니다.

## 모니터링 시작하기

모든 설정을 완료한 다음 애플리케이션 서버를 다시 시작하면 에이전트가 정보를 수집하기 시작합니다. [다음 문서](#)를 확인하세요.

# Elastic Beanstalk

## 에이전트 적용하기

Elastic Beanstalk를 이용해 배포할 경우 와탭 에이전트를 적용하는 방법입니다.

❗ Platform as a Service(PaaS) 환경에서는 배포 대상 애플리케이션에 와탭 에이전트의 설정을 포함해 배포합니다.

1. Spring Boot 결과물을 jar로 배포할 경우 `.ebextensions`의 내용을 적용할 수 없습니다. 다음 파일들을 압축해 zip 형태로 배포하세요.
  - `.elasticbeanstalk/config.yml`: eb 명령을 실행하는 디렉터리 하위에 자동 생성
  - `.ebextensions/{config_name}.config`: eb 설정 파일
  - `Procfile`: JVM command line 옵션을 설정하기 위한 파일
  - `{application}.jar`: 실행할 applicaion.jar 파일
2. Service에 적용할 와탭 에이전트의 파일을 압축해 S3(혹은 다운로드할 수 있는 public 경로)에 업로드하세요.
  - `paramkey.txt`
  - `whatap.agent-2.0_25.jar`
  - `whatap.conf`

```
# 디렉터리 압축하기
zip -r whatap-agent.zip agent
```

3. `.ebextension/{config_name}.config` 파일에 다운로드할 와탭 에이전트 경로를 입력하세요.

다운로드할 수 있도록 압축 파일 형태로 만들어 둔 경우 eb를 실행해 자동으로 다운로드한 다음 압축을 풉니다.

```
ebextension/{config_name}.config
```

```
sources:
target directory:
S3경로
```

## Example

```
sources:
  /home/webapp:
    http://s3.ap-northeast-2.amazonaws.com/{bucket-name}/whatap-agent.zip
```

4. 와탭 에이전트 옵션을 추가한 JVM command를 입력해 **Procfile**을 작성하세요.

## Procfile

```
web: java -javaagent:${WHATAP_JAR_FILE_PATH} -Dwhatap.name=${WHATAP_NAME}
-Dwhatap.okind=${WHATAP_OKIND_NAME} -Dwhatap.server.home=${APPLICATION_PATH}
-Dwhatap.conf.path=${WHATAP_HOME} -jar ${APPLICATION}.jar
```

## Example

```
web: java -javaagent:/home/webapp/agent/whatap.agent-X.Y.Z.jar -Dwhatap.name=bootTest -Dwhatap.okind=test
-Dwhatap.server.home=/var/app/current -Dwhatap.conf.path=/home/webapp/agent -jar whatap-boot-test.jar
```

5. **Procfile** 파일과 **.ebextensions** 파일, `{application}` **.jar** 파일을 압축하세요.

```
zip -r {application}.zip Procfile .ebextensions/{application}.jar
```

6. **.elasticbeanstalk/config.yml** 파일에 배포할 zip파일 경로를 추가하세요.

```
deploy:
  artifact: /path/to/{application}.zip
```

7. eb 배포 명령어를 실행하세요.

```
eb deploy
```

- ⓘ Java 17 버전 이상의 경우 reflection 관련한 다음 옵션을 추가하세요.

```
--add-opens=java.base/java.lang=ALL-UNNAMED
```



자세한 내용은 [AWS 가이드 문서](#)를 참조하세요.

## 힙 히스토그램 조회

와탭의 Java 모니터링에서는 JVM 메모리에 올라가 있는 Heap 점유 객체 현황(힙 메모리상의 객체별 사이즈)을 조회할 수 있는 기능을 기본으로 제공합니다. [애플리케이션 > 인스턴스 성능 분석](#) 메뉴에서 **힙 히스토그램** 탭을 선택하세요.

Java 6 ~ 8 버전에서는 JVM 옵션 없이 기본 지원하지만, 일부 Java 버전에 따라 다음과 같이 JVM 옵션을 적용해야 합니다.

- Java 9 ~ Java 15 버전

```
-Djdk.attach.allowAttachSelf=true
```

example

```
java -javaagent:{WHATAP_HOME}/whatap.agent-X.Y.Z.jar -Djdk.attach.allowAttachSelf=true -jar {application.jar}
```

- Java 16 버전 이상

```
-Djdk.attach.allowAttachSelf=true
--add-opens=jdk.attach/sun.tools.attach=ALL-UNNAMED
```

example

```
java -javaagent:{WHATAP_HOME}/whatap.agent-X.Y.Z.jar -Djdk.attach.allowAttachSelf=true --add-opens=java.base/
java.lang=ALL-UNNAMED --add-opens=jdk.attach/sun.tools.attach=ALL-UNNAMED -jar {application.jar}
```

Java 5 버전 이하, IBM Java는 지원하지 않습니다.



# 모니터링 시작하기

모든 설정을 완료한 다음 애플리케이션 서버를 다시 시작하면 에이전트가 정보를 수집하기 시작합니다. [다음 문서](#)를 확인하세요.

# Docker

## 에이전트 적용하기

기존 Docker 이미지를 바탕으로 와탭 에이전트 설정을 추가한 이미지를 빌드하세요.

ⓘ 이 문서에서는 Java 에이전트를 설치하는 경우만을 다룹니다. 와탭 쿠버네티스 모니터링과 함께 적용하는 경우 [다음 문서](#)를 참조하세요.

1. Docker 빌드 디렉터리를 생성하세요.

```
mkdir -p {Docker build dir}
```

2. `whatap.conf` 파일을 생성하세요.

```
1 cat >{Docker build Dir}/whatap.conf <<EOL
2 # 액세스 키를 입력하세요.
3 license=XXXXXXXXXXXXXXXX-XXXXXXXXXXXXXXXX-XXXXXXXXXXXXXXXX
4 # 수집 서버 IP 정보를 입력하세요.
5 whatap.server.host=xx.xx.xx.xx/yy.yy.yy.yy
6 EOL
```

3. Dockerfile을 생성하세요.

이미지를 빌드할 경우 와탭 이미지에서 `-javaagent` 옵션에 적용할 jar 파일을 복사할 수 있습니다.

```
cat >/home/silver/whatap/docker/Dockerfile <<EOL
FROM whatap/kube_mon as build
## 실제 이미지 생성 (기존 이미지에 Whatap 추가)
## $Image_Name(이미지명)
FROM $Image_Name
RUN mkdir -p /whatap
COPY --from=build /data/agent/micro/whatap.agent-*.jar /whatap
COPY ./whatap.conf /whatap/
#...
EOL
```

4. `JAVA_OPT` 에 다음 내용을 추가하세요.

```
WHATAP_HOME=/whatap
WHATAP_JAR=ls ${WHATAP_HOME}/whatap.agent-*.jar | sort -V | tail -1
export JAVA_OPTS="-javaagent:${WHATAP_JAR} "
```

5. Docker를 빌드하세요.

```
cd docker
docker build -t $Image_Name
```

ⓘ Java 17 버전 이상의 경우 reflection 관련한 다음 옵션을 추가하세요.

```
--add-opens=java.base/java.lang=ALL-UNNAMED
```

ⓘ Java 에이전트 파일 이름은 `Rename` 기능을 활용해 수정할 수 있습니다. Java 에이전트의 이름을 수정했다면 `JAVA_OPTS` 에 새로운 Java 에이전트 이름을 등록하세요.

**Java 에이전트 이름 수정 방법 예시**

```
java -cp whatap.agent-X.Y.Z.jar whatap.agent.setup.Rename -from whatap.agent-X.Y.Z.jar -to whatap.agent.jar
```

## 힙 히스토그램 조회

와탭의 Java 모니터링에서는 JVM 메모리에 올라가 있는 Heap 점유 객체 현황(힙 메모리상의 객체별 사이즈)을 조회할 수 있는 기능을 기본으로 제공합니다. [애플리케이션](#) > [인스턴스 성능 분석](#) 메뉴에서 [힙 히스토그램](#) 탭을 선택하세요.

Java 6 ~ 8 버전에서는 JVM 옵션 없이 기본 지원하지만, 일부 Java 버전에 따라 다음과 같이 JVM 옵션을 적용해야 합니다.

- Java 9 ~ Java 15 버전

```
-Djdk.attach.allowAttachSelf=true
```

**example**

```
java -javaagent:{WHATAP_HOME}/whatap.agent-X.Y.Z.jar -Djdk.attach.allowAttachSelf=true -jar {application.jar}
```

- Java 16 버전 이상

```
-Djdk.attach.allowAttachSelf=true
--add-opens=jdk.attach/sun.tools.attach=ALL-UNNAMED
```

**example**

```
java -javaagent:{WHATAP_HOME}/whatap.agent-X.Y.Z.jar -Djdk.attach.allowAttachSelf=true --add-opens=java.base/
java.lang=ALL-UNNAMED --add-opens=jdk.attach/sun.tools.attach=ALL-UNNAMED -jar {application.jar}
```

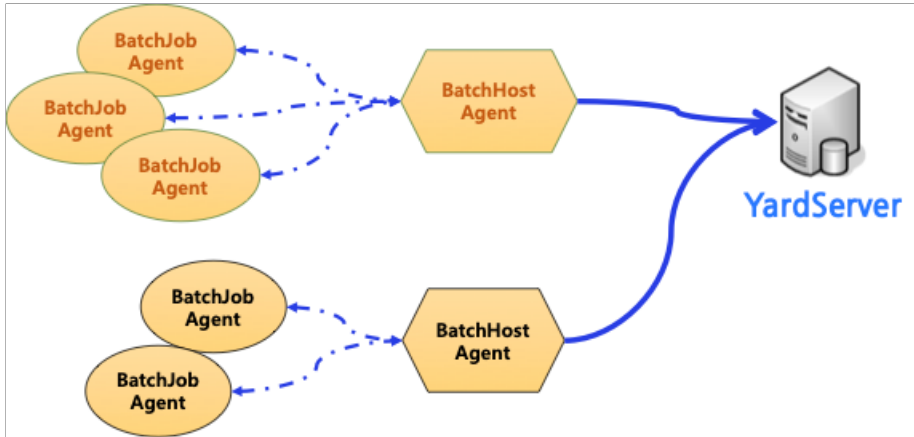
ⓘ Java 5 버전 이하, IBM Java는 지원하지 않습니다.

## 모니터링 시작하기

모든 설정을 완료한 다음 애플리케이션 서버를 다시 시작하면 에이전트가 정보를 수집하기 시작합니다. [다음 문서](#)를 확인하세요.

# 배치 애플리케이션

배치 애플리케이션에서 와탭 에이전트를 적용하는 방법입니다. 배치 애플리케이션은 WAS와는 다르게 애플리케이션이 실행과 중단을 반복하는 특성이 있습니다. 따라서 일반적인 웹 애플리케이션과 다르게 모니터링해야 합니다.



## 에이전트 다운로드

다음 명령어를 실행해 배치 애플리케이션 모니터링을 위한 에이전트를 다운로드할 수 있습니다.

```
wget https://api.whatap.io/agent/whatap.agent.batch.tar.gz
```

> 배치 에이전트 파일 구성

- [whatap.agent.batch.tar.gz](#)
  - └ [whatap.agent-x.y.z.jar](#): 자바 에이전트
  - └ [whatap.agent.batch.host-x.y.z.jar](#): 배치 호스트(Batch Host) - 수집 서버로 데이터 전송
  - └ [whatap.conf](#): 배치 호스트(Batch Host) 설정 파일(파일이 없는 경우 새로 생성하여 [설정 > 에이전트 설치](#) 메뉴의 `license`, `whatap.server.host` 항목을 추가하세요.)
  - └ [agent/whatap.agent.batch-x.y.z.jar](#): 배치 에이전트 - 애플리케이션 데이터를 수집하여 배치 호스트로 전송
  - └ [agent/whatap.conf](#): 배치 에이전트 설정 파일

# 에이전트 설정하기

## 배치 호스트(Batch Host) 설정

license, whatap.server.host 옵션값은 해당 프로젝트의 [설정 > 에이전트 설치](#) 메뉴에서 확인할 수 있습니다.

whatap.conf

```
# default setting
license=
whatap.server.host=

# 스텝을 나누어서 전송
split_trace_enabled=true

# 하나의 트레이스에 표시할 최대 스텝 개수
trace_step_max_count=1024
```

## 배치 에이전트 설정

agent/whatap.conf

```
# default setting
net_udp_listen_ip=0.0.0.0
net_udp_listen_port=6611

# 배치 호스트와 통신 설정
net_udp_listen_ip=0.0.0.0
net_udp_listen_port=6611

# 배치 잡에서 http call 최대 개수
trace_httpc_limit=1000000

# 배치 잡에서 sql 최대 개수
trace_sql_limit=1000000

# 액티브 스택 표시 여부
active_stack_enabled=true
# 5회 이내
```

```
active_stack_time1=5000
# 5회 이후
active_stack_time2=10000
# 최초 스택 시작시간
active_stack_start_wait_time=1000
```

## 배치 잡(Batch Job) 실행하기

배치 잡(Batch Job)을 하나의 트랜잭션 관점에서 모니터링해야 합니다. 그래서 와탭의 Java 에이전트는 Job 프로세스를 위한 Job 에이전트와 이 정보를 서버로 중계하기 위한 Host 에이전트로 분리된 두 개의 에이전트를 실행해야 합니다.

### 배치 호스트(Batch Host) 실행 명령어

#### Batch Host Agent

```
java -cp {BATCH_HOST_HOME}/whatap.agent.batch.host-x.y.z.jar;{BATCH_HOST_HOME}/whatap.agent-x.y.z.jar
-Dwhatap.name=batch whatap.agent.batch.App
```

#### Batch Host Agent, Java 17 or later

```
java --add-opens=java.base/java.lang=ALL-UNNAMED -cp {BATCH_HOST_HOME}/whatap.agent.batch.host-
x.y.z.jar;{BATCH_HOST_HOME}/whatap.agent-x.y.z.jar -Dwhatap.name=batch whatap.agent.batch.App
```

### 배치 에이전트(Batch Agent) 실행 명령어

#### Batch Agent

```
java -javaagent:{BATCH_AGENT_HOME}/whatap.agent.batch-x.y.z.jar -jar {APP_PATH}/batch-application.jar
```

#### Batch Agent, Java 17 or later

```
java -javaagent:{BATCH_AGENT_HOME}/whatap.agent.batch-x.x.x.jar --add-opens=java.base/java.lang=ALL-UNNAMED -jar
{APP_PATH}/batch-application.jar
```

- ❗ 배치 잡(Batch Job) 에이전트와 배치 호스트(Batch Host) 에이전트는 배치 잡(Batch Job)을 수행하는 호스트와 같이 설치 및 실행하세요.

## Time Limit 설정

배치 잡(Batch Job) 에이전트와 배치 호스트(Batch Host) 에이전트 사이는 UDP를 이용해 통신합니다.



배치 잡(Batch Job)은 일반적으로 장시간 수행합니다. 수집초가 넘을 수도 있습니다. 따라서 배치 잡(Batch Job) 별로 `time_limit`의 값을 다를 수 있습니다. 배치 잡(Batch Job)이 실행되는 예상 처리 시간을 `time_limit` 옵션에 설정하세요.

```
whatap.conf
```

```
time_limit=300000
warning_time=70% of time_limit
step_interval=5000
```

## 액티브 이퀄라이저 컬러 설정

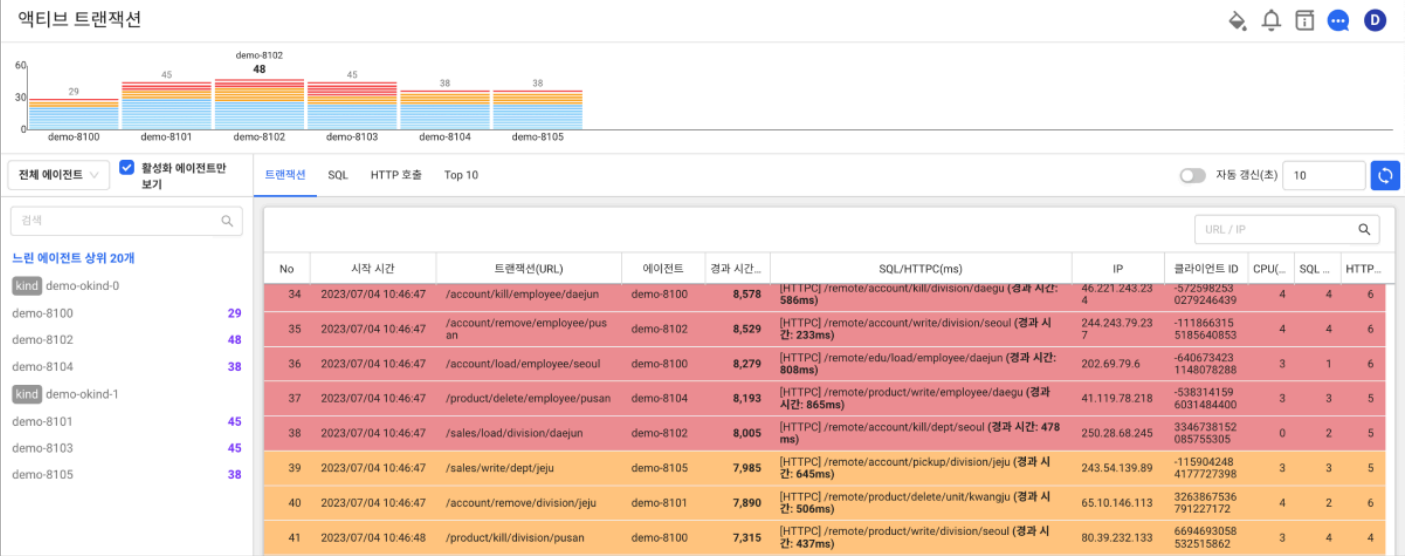
배치 잡(Batch Job)의 실행 시간이 `time_limit`의 70%를 지나면 노란색으로 표시하고, `time_limit`을 초과하면 빨간색으로 표시하도록 다음과 같이 설정하세요.

```
whatap.conf
```

```
1 ## 기대 종료시간의 70%
2 yellow_time = time_limit * 0.7;
```



```
3 ## 기대 종료 시간을 초과
4 red_time = time_limit;
```



ⓘ 서버 운영자는 배치 잡(Batch Job)의 지연 여부를 실시간으로 확인하려면 `time_limit` 값을 적절하게 설정하세요.

## 액티브 스택 수집 간격 설정

배치 잡(Batch Job) 에이전트는 `step_interval` 옵션값의 간격으로 액티브 스택을 수집할 수 있습니다.

```
whatap.conf

step_interval=5000
```

## 배치 에이전트의 메트릭 지표 수집

대시보드에는 배치 호스트(Batch Host)의 Heap Memory를 수집하고, 배치 애플리케이션의 성능 지표는 메트릭(TagCount: `batch_job_counter`)으로 수집합니다.

## batch\_job\_counter

- batch job: pid, hostname, job\_name
- gc: gc\_count, gc\_time\_sum, gc\_oldgen\_count
- heap: heap\_tot, heap\_use, heap\_max, heap\_pending\_final, heap\_perm
- cpu: cputime
- thread count: thread\_total\_started, thread\_count, thread\_daemon, thread\_peak\_count

# Java 1.6 버전 미만

Java 1.6 버전 미만에서 와탭 에이전트를 적용할 수 있습니다. 하지만 일부 기능에 제약이 있습니다.

## Java 1.5 버전

Java 1.5 버전은 `-javaagent` 옵션을 제공합니다. 다만 외부 스레드의 정보를 조회하는데 제약이 있어 일부 기능에 제약이 있습니다.

다음의 기능들은 동작하지 않습니다.

- 액티브 트랜잭션에서 실시간 트레이스
- 스택 분석 (탑 스택, 유니크 스택, 액티브 스택)
- 스레드 목록의 상세 스택

## Java 1.4 버전

Java 1.4 버전은 `-javaagent` 옵션이 없습니다. 다른 방식으로 와탭 에이전트를 설치해야 합니다. 물론 `whatap.agent.jar` 파일은 Java 1.4 버전을 사용해야 합니다.

```
whatap.java14.tracer-X.Y.Z.jar
```

## boot.jar 생성

`setup.sh` 파일을 이용해 `boot.jar` 파일을 생성하세요. 만약 `[JAVA_HOME]`을 입력하지 않으면 현재 경로를 JDK의 위치를 자동으로 인식합니다.

```
$ setup.sh [JAVA_HOME]
```

`setup.sh` 실행 결과: `whatap.java14.boot-X.Y.Z.jar`

## bootclasspath에 설치

`bootclasspath` 옵션을 사용해 빌드된 jar와 tracer jar를 prepend로 추가하세요.

```
-Xbootclasspath/p:${WHATAP_HOME}/whatap.java14.boot-1.0.2.jar:${WHATAP_HOME}/whatap.java14.tracer-1.0.2.jar
```

## 제약 사항

jdk 1.4 버전에서는 다음의 기능들이 동작하지 않습니다.

- 액티브 트랜잭션에서 실시간 트레이스
- 스택 분석 (탑 스택, 유니크 스택, 액티브 스택)
- 컴포넌트 버전
- 스레드 목록, 힙히스토그램
- 로드된 클래스
- 오픈 소켓
- 에이전트 덤프

# 설치 점검 사항

에이전트를 설치한 다음 확인해야 할 사항입니다. 만약 설치에 문제가 생겼다면 [다음 문서](#)를 참조하세요.

## WAS 다시 시작

애플리케이션 서버를 다시 시작하세요. 구동 로그에서 다음과 같은 와탭 캐릭터 로그가 보인다면 `JAVA_OPTS` 설정을 완료한 것입니다.

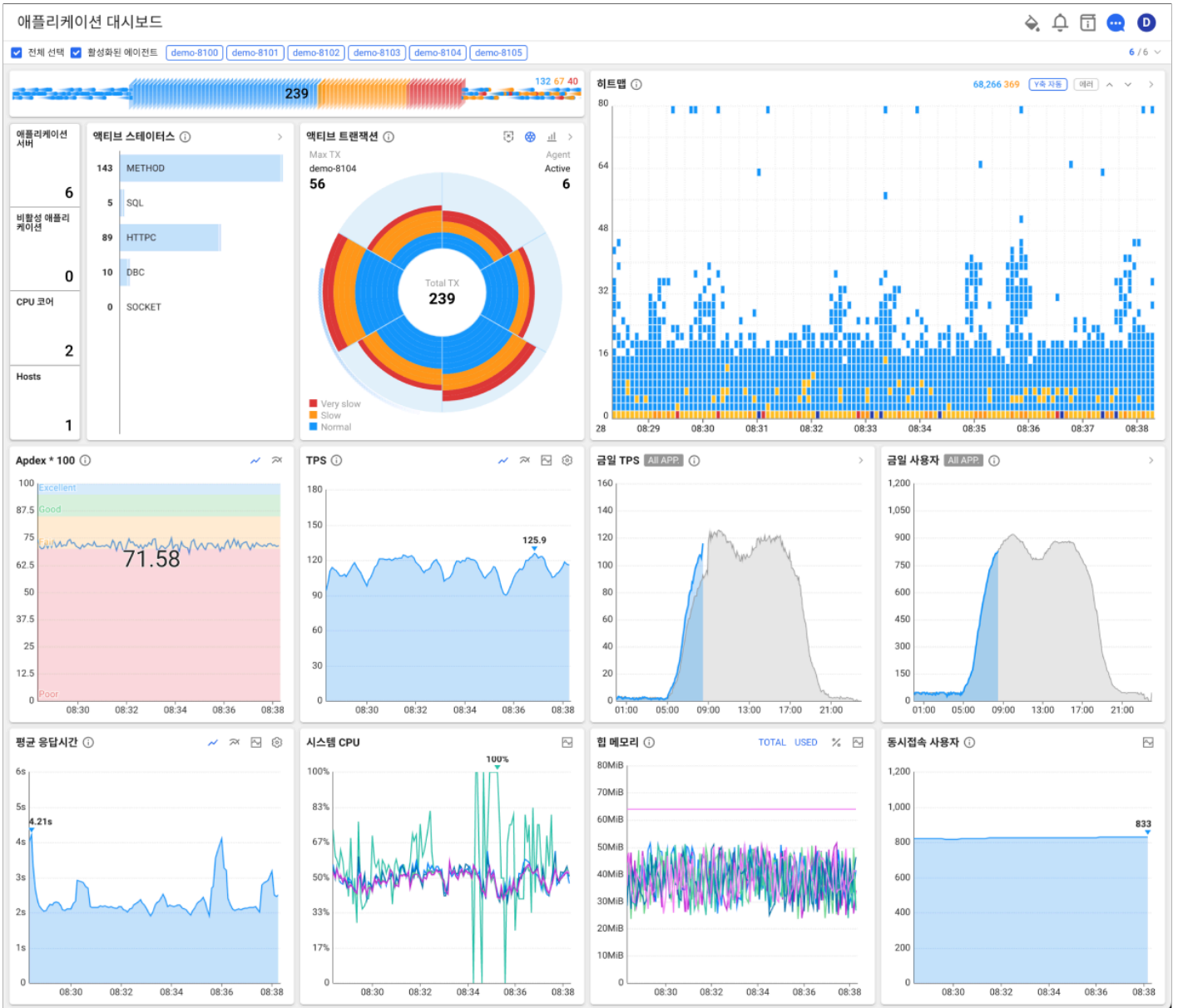
```
Nov 16, 2016 3:06:40 AM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deployment of web application directory /var/lib/tomcat7/webapps/ROOT has finished in 577 ms
Nov 16, 2016 3:06:40 AM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["http-bio-8080"]
Nov 16, 2016 3:06:40 AM org.apache.catalina.startup.Catalina start
INFO: Server startup in 3984 ms

- - - - -
| | / / / _ / _ / _ _
| | / / _ V _ // / / _ \ _ \
| _ | _ / / \ _ / / \ _ / _ /
  / \

Just Tap, Always Monitoring
WhaTap Agent version 0.3.9 20161115
```

## 에이전트 연결 확인

[와탭 모니터링 서비스](#)로 이동하세요. 로그인 후 프로젝트 목록에서 생성한 프로젝트를 선택하세요. [애플리케이션 대시보드](#) 화면에 연결한 서버의 정보가 나타나면 적용을 완료한 것입니다.



- ① 에이전트 이름의 기본값은 {type}-{ip2}-{ip3}-{port} 입니다.
- 에이전트의 이름을 변경하거나 속성을 추가하려면 [다음 문서](#)를 참조하세요.

# 설치 문제 해결

## 방화벽 설정 확인

와탭의 데이터 수집 서버에 대한 TCP 아웃바운드 정책을 설정하지 않으면 모니터링 정보를 전송할 수 없습니다. **방화벽 차단**을 해제하세요.

`telnet` 명령을 사용해 "Connected to `IP`"를 확인하세요.

```
$ telnet 52.193.60.176 6600
Trying 52.193.60.176...
Connected to 52.193.60.176.
Escape character is '^['.
```

① 수집 서버 정보는 [와탭 모니터링 서비스](#)에서 해당 프로젝트를 선택한 다음 **관리 > 에이전트 설치** 메뉴에서 확인할 수 있습니다.

## SpringBoot 2.2 버전 이상, Tomcat JMX가 off 상태

SpringBoot 2.2 버전 이상에서 Embedded Tomcat을 사용하는 경우 기본값으로 JMX 관련 기능이 동작하지 않습니다. 이 경우 JMX 기능을 활성화하세요. [관련 링크](#)

application.properties

```
spring.jmx.enabled=true
server.tomcat.mbeanregistry.enabled=true # tomcat embedded인 경우 #
```

application.yml

```
spring:
  jmx:
    enabled: true
server:
  tomcat:
```

```
mbeanregistry:
  enabled: true #tomcat embedded인 경우
```

## OSGI 프레임워크 사용 애플리케이션 서버의 경우

OSGI 프레임워크 구조의 애플리케이션 서버인 경우 JVM 옵션에 에이전트 패키지 prefix(whatap)를 등록하세요.

Jboss EAP 6.0 버전 이상, Jboss AS 7.0 버전 이상, Wildfly 8.0 버전 이상, IBM WebSphere AS 7.0 버전 이상 등이 해당합니다.

### JBoss AS, Wildfly, JBoss EAP 6.0 이상

다음 파일에 prefix를 등록하세요.

```
$JBOSS_HOME/bin/standalone.conf(domain.conf)
```

```
-Djboss.modules.system.pkgs=whatap
```



```
1. vagrant@ubuntu1204: ~/EAP-6.1.0/jboss-eap-6.1/bin (vagrant)
if [ "$JBOSS_MODULES_SYSTEM_PKGS" = "x" ]; then
  JBOSS_MODULES_SYSTEM_PKGS="org.jboss.byteman,org.jboss.logmanager"
fi

# Uncomment the following line to prevent manipulation of JVM options
# by shell scripts.
#
#PRESERVE_JAVA_OPTS=true
#
# Specify options to pass to the Java VM.
#
if [ "$JAVA_OPTS" = "x" ]; then
  JAVA_OPTS="-Xms1303m -Xmx1303m -XX:MaxPermSize=512m -Djava.net.preferIPv4Stack=true"
  JAVA_OPTS="$JAVA_OPTS -Djboss.modules.system.pkgs=whatap $JBOSS_MODULES_SYSTEM_PKGS -Djava.awt.headless=true"
  JAVA_OPTS="$JAVA_OPTS -Djava.util.logging.manager=org.jboss.logmanager.LogManager"
  JAVA_OPTS="$JAVA_OPTS -Xbootclasspath/p:/home/vagrant/EAP-6.1.0/jboss-eap-6.1/modules/system/layers/base/org.jboss.logmanager/main/jboss-logmanager-1.4.0.Final-redhat-1.jar"
  JAVA_OPTS="$JAVA_OPTS -javaagent:/home/vagrant/whatap/whatap.agent.tracer-0.2.5.jar"
else
  echo "JAVA_OPTS already set in environment; overriding default settings with values: $JAVA_OPTS"
fi

Sample JPDA settings for remote socket debugging
```



## WebSphere

JVM 옵션으로 다음 내용을 추가하세요.

```
-Dcom.ibm.ws.classloader.server.alwaysAllowedPackages=whatap
```

- 기본적으로 '\*'로 지정된 경우 별도 설정이 필요 없습니다.
- 설정 위치는 [WebSphere](#)를 참조합니다.

`security.policy` 권한을 다음과 같이 추가하세요.

```
$WEBSPPHERE_HOME/properties/server.policy || $WEBSPPHERE_PROFILE_HOME/properties/server.policy
```

```
grant codeBase "file:$WHATAP_HOME/-"
{
  permission java.security.AllPermission;
};
```

## Log Manager 관련 에러

JBoss AS 7.0 버전 이상, JBoss EAP 6.0 버전 이상에서 Log Manager 관련 에러가 발생한다면 JVM 옵션을 추가합니다.

- `-Djava.util.logging.manager` 항목에 Log Manager package 명을 설정하세요.
- `-Xbootclassloader` 항목에 JBoss Log Manager JAR file을 설정하세요.

```
(JBoss_Path)/bin/standalone.conf(domain.conf)
```

```
1 # Specify the exact Java VM executable to use.
2 #
3 if [ "x$JBoss_MODULES_SYSTEM_PKGS" = "x" ]; then
4   JBoss_MODULES_SYSTEM_PKGS="org.jboss.byteman,org.jboss.logmanager,whatap"
5 fi
6
7 ...
8 # Specify options to pass to the Java VM.
9 #
```

```

10 if [ "x$JAVA_OPTS" = "x" ]; then
11   JAVA_OPTS="-Xms1303m -Xmx1303m -Djava.net.preferIPv4Stack=true"
12   JAVA_OPTS="$JAVA_OPTS -Djboss.modules.system.pkgs=$JBoss_MODULES_SYSTEM_PKGS -Djava.awt.headless=true"
13   JAVA_OPTS="$JAVA_OPTS -Djava.util.logging.manager=org.jboss.logmanager.LogManager -Xbootclasspath/p:/home/
vagrant/EAP-7.0.0/modules/system/layers/base/org.jboss.logmanager/main/jboss-logmanager-2.0.3.Final-redhat-1.jar"
14 else
15   echo "JAVA_OPTS already set in environment; overriding default settings with values: $JAVA_OPTS"
16 fi

```

## MBeanServerBuilder 에러가 발생하는 경우

JBoss 5.0 버전 이하에서 MBeanServerBuilder 관련 에러가 출력된 경우 JVM 옵션을 추가하세요.

(JBoss\_Path)/bin/run.conf

```

1 # Specify options to pass to the Java VM.
2 #
3 if [ "x$JAVA_OPTS" = "x" ]; then
4   JAVA_OPTS="-Xms128m -Xmx512m -XX:MaxPermSize=256m -Dorg.jboss.resolver.warning=true
-Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000"
5   JAVA_OPTS="$JAVA_OPTS -Djboss.platform.mbeanserver"
6 fi

```

## Permission 오류가 발생하는 경우

Java Security Policy 관련 오류가 발생하면 `$JAVA_HOME/jre/lib/security/java.policy` 파일에 권한 설정을 추가하세요.

모든 권한을 일괄 적용하려면 다음과 같이 `java.policy` 파일에 설정을 추가하세요.

`$JAVA_HOME/jre/lib/security/java.policy`

```

grant {
  permission java.security.AllPermission;
};

```

## java.io.FilePermission 오류가 발생하는 경우

```
org.hyperic.sigar.SigarException: Failed to load sigar-ppc64-aix-7: Access denied (java.io.FilePermission
  at org.hyperic.sigar.Sigar.loadLibrary(Sigar.java:170)
  at org.hyperic.sigar.Sigar.<clinit>(Sigar.java:100)
  at java.lang.J9VMInternals.initializeImpl(Native Method)
  at java.lang.J9VMInternals.initialize(J9VMInternals.java:200)
  at whatap.xtra.sigar.SigarMain.<clinit>(SigarMain.java:31)
  at java.lang.J9VMInternals.initializeImpl(Native Method)
  at java.lang.J9VMInternals.initialize(J9VMInternals.java:200)
  at java.lang.Class.forNameImpl(Native Method)
  at java.lang.Class.forName(Class.java:217)
  at whatap.agent.proxy.SigarFactory.create(SigarFactory.java:62)
  at whatap.agent.counter.task.SystemPerf.<init>(SystemPerf.java:35)
  at whatap.agent.counter.CounterExecutingManager.run(CounterExecutingManager.java:78)
```

다음과 같이 `java.policy` 파일에 설정을 추가하세요.

```
$JAVA_HOME/jre/lib/security/java.policy
```

```
grant {
  ...
  permission java.io.FilePermission {오류 메시지에서 확인된 패키지 경로}, "read"
};
```

## java.util.PropertyPermission 오류가 발생하는 경우

다음과 같이 `java.policy` 파일에 설정을 추가하세요.

```
$JAVA_HOME/jre/lib/security/java.policy
```

```
grant {
  ...
  permission java.util.PropertyPermission {오류 메시지에서 확인된 패키지 경로}, "read"
};
```

## Sigar library를 로딩하지 못하는 경우

`$WHATAP_HOME/lib1/*.so` 파일에 실행 권한이 있는지 확인하세요. 권한이 없다면 실행 권한을 부여하세요.

```
$ sudo chmod +x *.so
```

AIX 7에서 `$WHATAP_HOME/lib1` 하위에 `libsigar-ppc64-aix-7.so` 파일이 없어 오류가 발생한 경우 파일을 복사하세요.

```
org.hyperic.sigar.SigarException: Failed to load sigar-ppc64-aix-7: Access denied (java.io.FilePermission)
  at org.hyperic.sigar.Sigar.loadLibrary(Sigar.java:170)
  at org.hyperic.sigar.Sigar.<clinit>(Sigar.java:100)
  at java.lang.J9VMInternals.initializeImpl(Native Method)
  at java.lang.J9VMInternals.initialize(J9VMInternals.java:200)
  at whatap.xtra.sigar.SigarMain.<clinit>(SigarMain.java:31)
  at java.lang.J9VMInternals.initializeImpl(Native Method)
  at java.lang.J9VMInternals.initialize(J9VMInternals.java:200)
  at java.lang.Class.forNameImpl(Native Method)
  at java.lang.Class.forName(Class.java:217)
  at whatap.agent.proxy.SigarFactory.create(SigarFactory.java:62)
  at whatap.agent.counter.task.SystemPerf.<init>(SystemPerf.java:35)
  at whatap.agent.counter.CounterExecutingManager.run(CounterExecutingManager.java:78)
```

```
$ cp libsigar-ppc64-aix-5.so libsigar-ppc64-aix-7.so
```

# 에이전트 설정

와탭 에이전트는 에이전트 별 필요한 설정을 `whatap.conf` 파일에 작성합니다. 에이전트는 환경변수를 통해 설정 파일의 위치를 파악하고 로딩합니다.

별도 설정이 없는 경우 에이전트는 자기 자신(`whatap.agent-X.Y.Z.jar`)이 위치한 디렉터리에서 `whatap.conf` 파일을 찾도록 구현했습니다. 5초에 한번씩 `whatap.conf` 파일의 변경 여부를 체크하고 리로딩합니다.

사용자의 편의성을 위해 [와탭 모니터링 서비스](#)에서도 에이전트 설정 기능을 제공하고 있습니다.

## 기본 설정

```
whatap.conf
```

```
whatap.home=.
whatap.config=whatap.conf
```

- `whatap.home` : 기본값은 `whatap.agent.jar` 파일의 경로를 입력하세요.
- `whatap.config` : 파일명만 입력할 것을 권고합니다. 상대 패스를 같이 입력할 수는 있습니다.

에이전트가 데이터를 수집하고, 서버로 데이터를 전송하기 위해서는 최소한 액세스 키와 서버 주소는 설정돼 있어야 합니다.

```
whatap.conf
```

```
license=x220g160hgd29-x3qpf0garcus7-z4p0kao58un11a
whatap.server.host=10.10.0.1
```

와탭 에이전트는 최초 실행시 다음과 같이 어떤 jar 파일이 에이전트로 사용됐는지, 어떤 경로에서 `whatap.conf` 파일을 로딩하는지 확인할 수 있습니다.



시작 스크립트에 JVM 옵션을 추가하세요.

Service A

-Dwhatap.config=whatap\_Aservice.conf

Service B

-Dwhatap.config=whatap\_Bservice.conf

! **whatap.conf** 설정 파일은 반드시 `${WHATAP_HOME}` 경로에 위치해야 합니다.

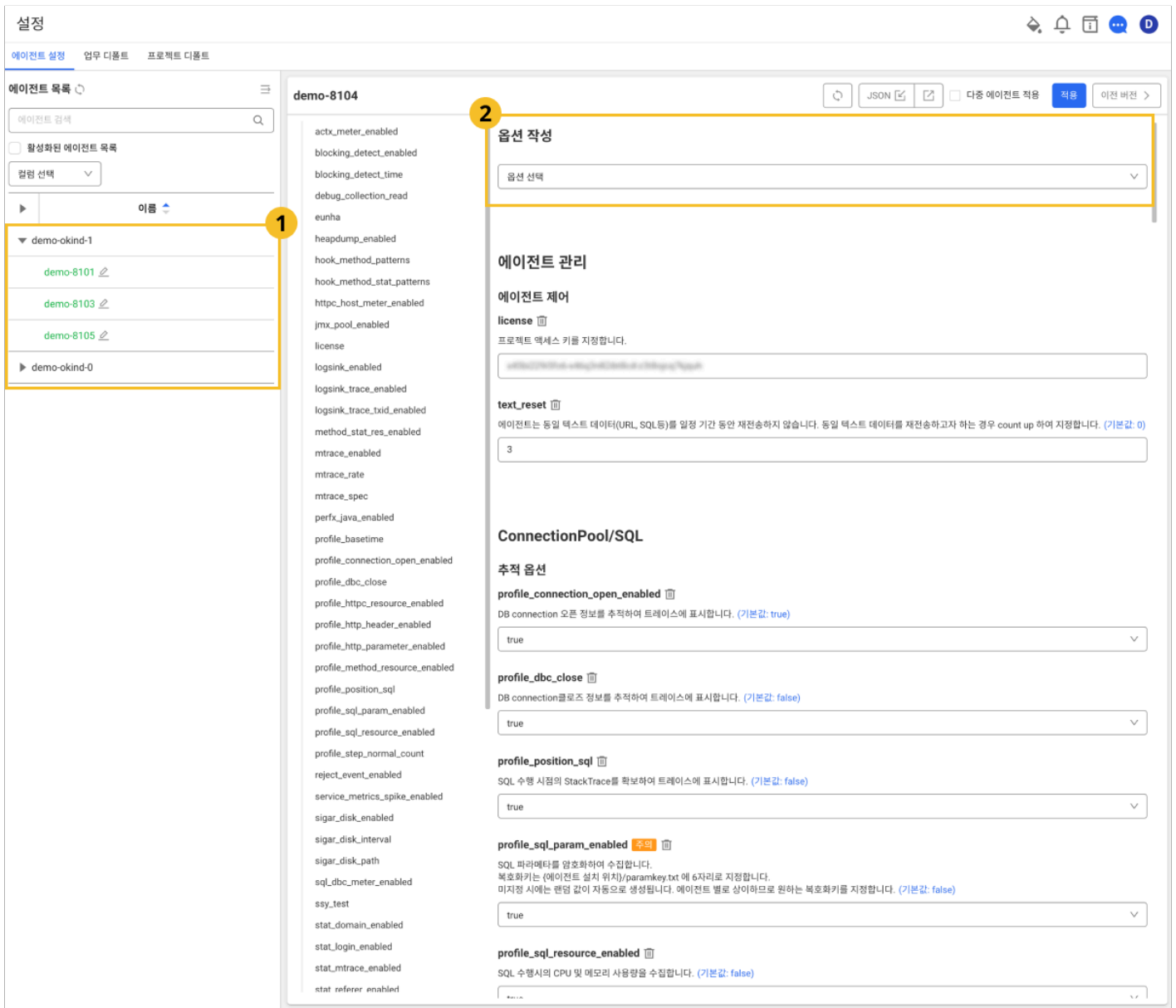
## 서비스 화면에서 에이전트 설정하기

홈 화면 > 프로젝트 선택 > 관리 > 에이전트 설정

모니터링 대상 서버에 위치한 `whatap.conf` 파일을 직접 수정하지 않고 [와탭 모니터링 서비스](#)에서 에이전트 설정 옵션을 추가하거나 수정, 삭제할 수 있습니다.

- ⓘ 이 기능은 수정 권한을 가진 멤버만 이용할 수 있습니다. 수정 권한이 없는 멤버는 설정 내용을 조회만 할 수 있습니다.
- 옵션 값으로 설정할 수 있는 형식은 다음과 같습니다.
  - Boolean 형식의 값은 `true` 또는 `false` 를 선택하세요.
  - 숫자 형식의 값은 숫자만 입력할 수 있습니다.
  - 텍스트(String) 형식의 값을 입력 또는 수정할 경우 옵션 설명을 자세히 확인하세요.
- 수정할 수 없는 옵션은 선택할 수 없습니다. (예, `license` )
- 추가 또는 수정, 삭제한 옵션에 따라 에이전트를 재시작해야 할 수 있습니다.
- 애플리케이션 종류 및 에이전트의 버전에 따라 적용할 수 있는 옵션 키는 다를 수 있습니다.

# 옵션 추가하기



1. 에이전트 목록에서 옵션을 추가하려는 1 에이전트를 선택하세요.
2. 2 옵션 작성에서 추가할 옵션 항목을 선택하세요.



The screenshot shows the 'Option Creation' (옵션 작성) interface. At the top, there is a search bar labeled '옵션 선택' (Option Selection) and a search icon. Below this is the 'Direct Input' (직접 입력) section. The main area contains a list of options with their basic values (기본값):

- Option 1: (기본값: 100)
- Option 2: (기본값: true)
- Option 3: (기본값: 10)
- Option 4: (기본값: false)
- Option 5: (기본값: false)

- **검색**에서 추가할 옵션을 찾을 수 있습니다. 텍스트를 입력하면 일치하는 옵션을 필터링합니다.
- **직접 입력**을 선택하면 옵션 키와 값을 입력할 수 있습니다.

3. 선택한 옵션 키에 대한 설명과 기본값을 확인한 다음 설정값을 입력하세요.

The screenshot shows the 'Option Creation' (옵션 작성) interface with a selected option. The option name is visible, and its basic value (기본값) is 100. Below the option name, there is an input field containing the value '100' and the label 'number'. At the bottom, there is a '+ 추가하기' (Add) button.

- 선택한 옵션을 취소하려면 **🗑** 버튼을 선택하세요.
- 옵션을 추가 설정하려면 **+ 추가하기** 버튼을 선택하고 2번의 과정을 반복하세요.

4. 원하는 모든 옵션을 추가했다면 화면 오른쪽 위에 **적용** 버튼을 선택하세요.

선택한 옵션 및 설정값을 에이전트에 적용합니다.

- ① • 수정 중인 내용을 초기화하려면 화면 오른쪽 위에 **🔄** 버튼을 선택하세요.
- 옵션값으로 아무것도 입력하지 않은 상태에서 **적용** 버튼을 선택하면 해당 옵션을 삭제합니다.
- 이미 추가한 옵션은 옵션 목록에서 선택할 수 없습니다.
- 애플리케이션 종류 및 에이전트의 버전에 따라 적용할 수 있는 옵션 키는 다를 수 있습니다.
- 예시로 제공한 이미지는 애플리케이션 종류 및 에이전트에 따라 다를 수 있습니다.

## 옵션 수정 또는 삭제하기

demo-8104

JSON [ ] [ ]  다중 에이전트 적용 적용 이전 버전 >

- actx\_meter\_enabled
- blocking\_detect\_enabled
- blocking\_detect\_time
- debug\_collection\_read
- eunha
- heapdump\_enabled
- hook\_method\_patterns
- hook\_method\_stat\_patterns
- httpc\_host\_meter\_enabled
- jmx\_pool\_enabled
- license
- logsink\_enabled
- logsink\_trace\_enabled
- logsink\_trace\_txid\_enabled
- method\_stat\_res\_enabled
- mtrace\_enabled
- mtrace\_rate
- mtrace\_spec
- perfx\_java\_enabled
- profile\_basetime
- profile\_connection\_open\_enabled
- profile\_dbc\_close
- profile\_httpc\_resource\_enabled
- profile\_http\_header\_enabled**
- profile\_http\_parameter\_enabled
- profile\_method\_resource\_enabled
- profile\_position\_sql
- profile\_sql\_param\_enabled
- profile\_sql\_resource\_enabled
- profile\_step\_normal\_count
- reject\_event\_enabled
- service\_metrics\_spike\_enabled
- sigar\_disk\_enabled
- sigar\_disk\_interval
- sigar\_disk\_path
- sql\_dbc\_meter\_enabled
- ssy\_test
- stat\_domain\_enabled
- stat\_login\_enabled
- stat\_mtrace\_enabled

**profile\_http\_header\_enabled** 주의 [ ]

트레이스 상세 내역에 HTTP 헤더 정보를 표시할 지 지정합니다. (기본값: false)

true

**HTTP 파라메타**

**profile\_http\_parameter\_enabled** 주의 [ ]

트랜잭션의 쿼리 파라메타를 수집하여 트레이스에 노출할 지를 지정합니다. 복호화키는 (에이전트 설치 위치)/paramkey.txt 에 6자리로 지정합니다. 미지정 시에는 랜덤 값이 자동으로 생성됩니다. 에이전트 별로 상이하므로 원하는 복호화키를 지정합니다. (기본값: false)

true

**트랜잭션 URL**

**정규화**

**trace\_normalize\_enabled** 주의 [ ]

트랜잭션 URL을 정규화합니다. (기본값: true)

true

**trace\_normalize\_urls** 주의 [ ]

정규화 대상 URL을 등록합니다.

/edu/create/employee/{region}

**사용자 추적**

**추적 옵션**

**trace\_user\_using\_ip** [ ]

remote address 기준으로 사용자를 추적합니다. false 지정 시, 쿠키 기반으로 사용자를 추적합니다. (기본값: true)

false

**기타**


**debug\_collection\_read** [ ]

false

eunha [ ]

3

^ 옵션 추가로 이동

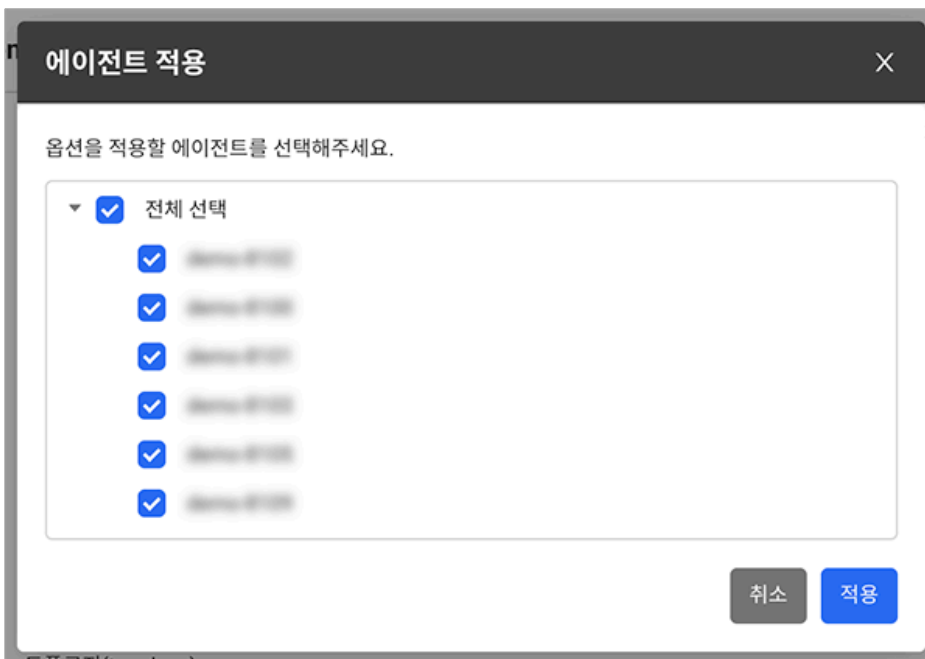
1. 화면을 위 또는 아래로 스크롤하거나 왼쪽의 옵션 목록에서 수정 또는 삭제하려는 옵션을 선택하세요.
2. 변경하려는 옵션에서 원하는 값을 선택하거나 수정하세요. 옵션을 삭제하려면  버튼을 선택하세요.
3. 변경한 사항을 적용하려면 **적용** 버튼을 선택하세요.

- ① • 화면 가장 위로 이동하려면 **3** **옵션 추가로 이동** 버튼을 선택하세요.
  - 옵션값으로 아무것도 입력하지 않은 상태에서 **적용** 버튼을 선택하면 해당 옵션을 삭제합니다.

## 여러 에이전트에 동시 적용하기

프로젝트에 소속된 여러 개의 에이전트에 변경한 옵션을 동시에 적용할 수 있습니다.

1. 화면 오른쪽 위에 **다중 에이전트 적용** 체크박스를 선택하면 각 옵션 항목에 체크박스가 생성됩니다.
2. 동시에 적용하길 원하는 옵션의 체크박스를 선택하세요. 여러 개를 선택할 수 있습니다.
3. 화면 오른쪽 위에 **적용** 버튼을 선택하세요.
4. **에이전트 적용** 창이 나타나면 변경한 옵션을 적용할 에이전트를 선택하세요. 모두 선택하려면 **전체 선택** 체크박스를 선택하세요.



5. **적용** 버튼을 선택하세요.

## 에이전트 기본값 설정하기

프로젝트에 새로운 에이전트를 추가할 경우 기존의 설정값을 반복해서 적용하는 번거로움을 피하고 싶다면 프로젝트별, 업무별 기본 설정값을 만들어 적용할 수 있습니다.



- **업무 디폴트:** `whatap.conf` 파일에서 `whatap.okind` 항목으로 분류한 에이전트들에 옵션을 공통 적용할 수 있고, 적용된 옵션을 확인할 수 있습니다.
- **프로젝트 디폴트:** 프로젝트에 소속된 모든 에이전트들에 옵션을 공통 적용할 수 있고, 적용된 옵션을 확인할 수 있습니다.

- ① • **업무 디폴트** 또는 **프로젝트 디폴트** 탭을 선택한 다음 옵션을 적용하는 것은 에이전트를 공통으로 관리하는 데 유용합니다.
- 옵션의 적용 우선 순위는 **업무 디폴트**, **프로젝트 디폴트**, **에이전트 설정** 순입니다.
    - **프로젝트 디폴트**에서 b 옵션값을 1000을 적용하고, **업무 디폴트**에서 b 옵션값을 2000으로 적용하면, **업무 디폴트**에서 설정한 옵션값 2000을 우선 적용합니다.
    - **프로젝트 디폴트**에서 b 옵션값을 아무것도 입력하지 않았고 **업무 디폴트**에 b 옵션값에 2000이 적용된 경우에도 **업무 디폴트**에서 설정한 옵션값을 우선 적용합니다.

## 에이전트 설정 공유하기

에이전트 설정 내용을 json 형식의 파일로 저장하고 파일을 불러와 다른 에이전트에 적용할 수 있습니다.

1. **에이전트 목록**에서 설정 내용을 json 파일로 내보낼 에이전트를 선택하세요.
2. 화면 오른쪽 위에 버튼을 선택하세요.
3. json 파일을 다운로드하세요.
4. **에이전트 목록**에서 다른 에이전트를 선택하세요.
5. 화면 오른쪽 위에 버튼을 선택하세요.

json 파일을 통해 가져온 에이전트 설정을 확인하세요.

# 에이전트 설정 옵션 안내

다음 링크를 통해 설정할 수 있는 옵션들에 대한 설명을 제공합니다.

## 에이전트 이름 식별

모니터링 대상을 구별하기 위한 애플리케이션 서버의 고유 식별자 설정에 대해 안내합니다. 모니터링 대상 시스템 내에서 애플리케이션 서버를

## 에이전트 기능 제어

자바(Java) 애플리케이션 서버의 모니터링을 위한 에이전트 기능을 세밀하게 조정하는 방법을 안내합니다. 에이전트의 활성화/비활성화, 트랜

## CPU/메모리/디스크 사용량 수집

자바(Java) 에이전트가 CPU, 메모리, 디스크 사용량을 수집하는 방법을 안내합니다. 자바(Java)의 JMX 라이브러리를 기본적으로 활용하며, 필

## 에이전트 통신 설정

자바(Java) 에이전트가 애플리케이션 서버로부터 수집한 데이터를 서버로 전송하기 위한 네트워크 설정 방법을 안내합니다. 주요 설정 옵션으로

## 에이전트 성능

자바(Java) 에이전트의 성능 관련 옵션을 안내합니다. 동시에 처리할 수 있는 최대 트랜잭션 수와 트랜잭션 데이터를 저장하는 버퍼의 초기 크기

## 에이전트 로그 설정

자바(Java) 애플리케이션 환경에서 발생하는 로그 데이터를 에이전트를 통해 관리하는 방법을 안내합니다. 로그 파일의 경로 및 이름 설정, 보관

### 📖 트랜잭션

자바(Java) 애플리케이션 내 각 트랜잭션의 성능 데이터를 에이전트가 추적하고 분석할 수 있는 설정 방법을 안내합니다. 트랜잭션의 요청 처

### 📖 DB, SQL

데이터베이스 및 SQL 성능 데이터 수집을 위한 자바(Java) 에이전트의 다양한 옵션을 안내합니다. DBCP, Hikari, Tomcat 등 다양한 DB 연결

### 📖 HTTPC, API Call

HTTP 외부 호출과 API 호출 관련 데이터를 수집, 분석을 위한 자바(Java) 에이전트의 옵션 설정 방법을 안내합니다. TOO SLOW 에러 처리 및

### 📖 스크립트 플러그인

자바(Java) 에이전트에서 제공하는 플러그인 옵션을 통해 사용자가 원하는 코드를 트레이스 데이터에 주입하거나 메소드 수행 전후로 부가 정

### 📖 사용자 수

자바(Java) 웹 애플리케이션 서버에 연결된 사용자 수를 에이전트 설정을 통해 집계하는 방법을 안내합니다. 실시간 사용자 집계 활성화 여부

### 📖 부하량 제어

자바(Java) 애플리케이션 서버의 트래픽 스로틀링 옵션을 설정하는 에이전트 옵션을 안내합니다. 애플리케이션의 최대 동시 요청 수를 제한하

### 📖 에이전트 알림

자바(Java) 에이전트를 통해 애플리케이션 서버에서 발생하는 다양한 이벤트에 대한 알림 설정 방법을 제공합니다. 트랜잭션 재귀 호출, 서비

## 📖 Apdex

사용자 만족도를 측정하는 Apdex 점수를 설정하는 자바(Java) 에이전트 옵션을 안내합니다. Apdex 지표를 통해 애플리케이션의 성능을 객관

## 📖 통계

자바(Java) 애플리케이션 모니터링을 위한 다양한 통계 수집 기능 관련 에이전트 옵션을 안내합니다. 성능 카운터 확장, 도메인별 트랜잭션 수

## 📖 토폴로지 맵

자바(Java) 에이전트가 수집한 데이터를 사용하여 애플리케이션의 토폴로지 맵을 생성하는 에이전트 설정 옵션을 안내합니다. 트랜잭션 호출자,

## 📖 트랜잭션 에러 스택

자바(Java) 애플리케이션에서 트랜잭션 중 발생하는 에러를 감지하고 이에 대한 스택 트레이스를 표시하는 에이전트 설정 옵션을 안내합니다.

## 📖 부가 기능

자바(Java) 애플리케이션의 모니터링 수준을 사용자의 필요에 맞게 조정할 수 있는 추가적인 에이전트 설정 옵션을 안내합니다. SQL 파라미터

## 📖 비동기 추적

자바(Java) 에이전트 설정을 통해 비동기 애플리케이션 활동을 추적하는 방법을 안내합니다. CompletableFuture 메소드 수집 방법과 사용

## 📖 오픈소스 추적

자바(Java) 애플리케이션에서 사용하는 프레임워크나 오픈소스 라이브러리를 에이전트를 통해 추적하는 설정 방법을 제공합니다. Java 에이전



# 에이전트 이름 식별

와탭은 모니터링 정보 수집 대상인 애플리케이션 서버 식별을 위한 정보로 기본적으로 애플리케이션 서버로부터 수집한 정보를 활용합니다. 기본 활용 정보는 애플리케이션 서버의 종류, IP 주소 등을 조합해 애플리케이션 서버를 고유 식별자로 사용합니다. 필요에 따라 사용자가 설정한 명칭을 이용하거나 패턴을 변경해 고유한 값으로 변경해 이용합니다. 에이전트의 이름은 반드시 고유한 값이어야 합니다.

애플리케이션 서버로부터 추출한 정보를 활용하는 이유는 애플리케이션 서버 정지 또는 네트워크 단절, 에이전트 문제로 인한 수집 서버와 에이전트의 통신 두절 상태가 복구되었을 경우, 재접속된 에이전트로부터 송신되는 정보가 기존 에이전트로부터 송신된 정보와의 연속성을 유지하기 위해서입니다.

- ❗ • 에이전트 이름은 프로젝트 단위로 고유해야 합니다.
- 에이전트 아이디/이름을 변경하면 이전 데이터와 연결되지 않습니다.

- ⓘ • 와탭 모니터링 서비스에서는 모니터링 대상을 오브젝트 또는 에이전트로 혼용해서 사용합니다.

## 기본 이름 결정 방식

JVM 옵션으로 `whatap.oname` 설정이 없는 경우 에이전트 이름을 다음과 같은 우선 순위로 자동 부여합니다.

1. `whatap.type` 을 식별합니다.
2. `whatap.type` 이 서블릿 컨테이너면 애플리케이션이 사용하는 포트를 식별합니다.
3. 포트를 식별할 수 없다면 process ID(PID)를 검색합니다.
4. 식별한 정보에 IPv4 주소 3, 4번째를 추가해 최종 이름을 결정합니다.

예를 들어, IP 주소가 10.11.12.13인 서버에서 Tomcat이 8080 포트로 시작한 경우 `TC-12-13-8080` 명칭으로 에이전트 이름을 자동 부여합니다.

## 에이전트 이름 변수

사용할 수 있는 이름 변수를 다음과 같이 제공합니다.

변수	설명
문자열	입력한 문자열을 사용합니다.
{type}	컨테이너 타입을 자동 식별해 부여합니다. 자동 식별 대상: JAVA, TOMCAT, JBOSS, RESIN, SPRINGBOOT, JETTY, JEUS, WEBLOGIC, WEBSPPHERE, UNDERTOW, PLAY
{ip0}	IPv4 주소 중 첫 번째 단위를 사용합니다. (예시, <b>10</b> .11.12.13 중 <b>10</b> )
{ip1}	IPv4 주소 중 두 번째 단위를 사용합니다. (예시, 10. <b>11</b> .12.13 중 <b>11</b> )
{ip2}	IPv4 주소 중 세 번째 단위를 사용합니다. (예시, 10.11. <b>12</b> .13 중 <b>12</b> )
{ip3}	IPv4 주소 중 네 번째 단위를 사용합니다. (예시, 10.11.12. <b>13</b> 중 <b>13</b> )
{pid}	Java의 Process ID(PID)를 사용합니다.
{port}	애플리케이션이 Listen 하는 포트를 사용합니다.

## 에이전트 이름 설정

시작 스크립트에 JVM 옵션을 추가해 자동으로 부여한 이름 대신 직접 이름을 설정할 수 있습니다.

우선순위	옵션	설정 위치	설명
1	-Dwhatap.name	JVM Options	에이전트 이름을 패턴으로 설정하세요. default: {type}-{ip2}-{ip3}-{port}
2	-Dwhatap.ename	JVM Options	에이전트 이름을 고정값으로 설정합니다. -Dwhatap.ename=front-{ip3}-{port}

- ⓘ • 클라우드의 Scale Out 환경에서는 VM이 복사되어 사용될 수 있습니다. 따라서 오브젝트 이름이 자동 부여되는 것이

❗ 중요합니다.

- 만약 2개의 VPC 환경을 만들고 동일한 private IP를 사용하도록 네트워크를 구성한 경우 두 개의 VPC가 중복되면 안 됩니다. 예를 들어, 별도의 프로젝트로 분리하거나 `-Dwhatap.name` 옵션과 이름 변수를 서로 다른 방식으로 조합해 설정할 수 있습니다.

다음은 `whatap.conf` 파일에 에이전트 이름으로 적용할 수 있는 옵션입니다.

- **whatap.name** String

기본값 `{type}-{ip2}-{ip3}-{port}`

에이전트 이름을 패턴으로 설정합니다.

- **whatap.okind** String

에이전트가 속한 그룹 단위를 설정합니다. 여러 에이전트를 같이 설정해 해당 그룹명으로 토폴로지를 확인할 수 있습니다.

- **whatap.onode** String

에이전트가 속한 노드 단위를 설정합니다.

- **auto\_ename\_enabled** Boolean

기본값 `false`

서버에 등록할 에이전트 이름(`whatap.ename`)을 서버로부터 자동 부여받는 기능을 활성화합니다. 해당 옵션 적용 시 JVM 옵션으로 설정한 `-Dwhatap.name`, `-Dwhatap.ename` 값은 무시합니다. 수집 서버와의 통신을 통해 `whatap.ename` 값을 부여 받은 다음 에이전트의 일반적인 동작을 시작합니다.

- **auto\_ename\_prefix** String

기본값 `agent`

에이전트 이름을 서버로부터 자동 부여할 때 에이전트 이름의 prefix, 보통 업무명을 사용합니다. prefix 일련번호 '1~'을 부여합니다.

❗ 에이전트 이름을 자동으로 부여하는 `auto_ename_enabled` 옵션이 활성화 상태여야(`true`) 해당 옵션을 사용할 수 있습니다.

- **auto\_ename\_reset** Int

기본값 `0`

서버로부터 새로운 에이전트 이름을 부여받기 위해 수정합니다. 에이전트 이름을 자동 부여하면 `whatap.ename`이라는 시스템 환경 변수로 설정됩니다. 한번 정해진 시스템 환경 변수는 자바 인스턴스가 다시 시작할 때까지 유지합니다. 리셋을 원할 때

`auto_otype_reset` 값을 수정하세요. 현재 설정 값과 다른 값으로 변경하면 적용됩니다.

# 에이전트 기능 제어

자바(Java) 애플리케이션 서버의 모니터링을 위한 에이전트 기능을 세밀하게 조정하는 방법을 안내합니다. 에이전트의 활성화/비활성화, 트랜잭션 및 성능 카운터 추적, OS 정보 수집 등의 기능을 포함하며, 각 기능을 제어하는 다양한 설정 옵션들에 대해 확인할 수 있습니다.

- **shutdown** Boolean

기본값  false

true 인 경우 에이전트의 모든 동작을 중지하고 서버와의 연결을 종료합니다.

- **enabled** Boolean

기본값  true

전체 기능을 활성화합니다.  false 인 경우에도 서버와 최소한의 통신을 유지하기 위한 정보는 전송됩니다.

- **transaction\_enabled** Boolean

기본값  true

트랜잭션 추적 기능을 활성화합니다. 히트맵에 기록되는 트랜잭션 정보가 해당합니다.  enabled 옵션이  false 인 경우 비활성화합니다.

- **counter\_enabled** Boolean

기본값  true

성능 카운터 추적 기능을 활성화합니다. 액티브 트랜잭션 수 및 사용자 수, JVM 자원 사용량, Process CPU 사용량, DB Pool 사용량 정보 등이 해당합니다.  enabled 옵션이  false 인 경우 비활성화합니다.

- **stat\_enabled** Boolean

기본값  true

통계 정보 추적 기능을 활성화합니다. 5분 단위로 수집하는 트랜잭션, SQL, HTTPCALL, UserAgent, Client IP 등의 통계 데이터 등이 해당합니다.  enabled 옵션이  false 인 경우 비활성화합니다.

- **sigar\_enabled** Boolean

기본값  true

sigar 라이브러리를 통한 OS 정보 수집을 활성화합니다. sigar 라이브러리를 통해 5초 단위로 수집하는 CPU, Memory, Disk 등의 OS 자원 데이터가 해당합니다.  enabled 옵션이  false 인 경우 비활성화합니다.

- **active\_stack\_enabled** Boolean

기본값 `true`

액티브 스택 추적을 활성화합니다. 스택 메뉴의 탑 스택 및 유니크 스택, 액티브 스택이 해당합니다. `enabled` 또는 `counter_enabled` 옵션이 `false` 인 경우 비활성화합니다.

- **license** `String`

에이전트 설치 시 서버로부터 부여받은 액세스 키를 입력합니다. 액세스 키에는 에이전트가 속한 프로젝트와 보안 통신을 위한 암호 키를 포함합니다.

- **cypher\_level** `aes_bit [128, 256]`

기본값 `128`

AES 보안 알고리즘에 대한 암호 레벨을 설정합니다. 256 비트까지 확장할 수 있습니다. AES 256 암호화 설정에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

- **encrypt\_level** `encrypt_level [1, 2, 3]`

기본값 `2`

와탭 에이전트는 서버로 데이터를 전송할 때 데이터 속성에 따라 선택적으로 암호화합니다. 높은 보안을 유지하면서도 성능상 이점을 가지고 있습니다. 데이터 유형에 상관 없이 일괄적인 암호화 정책을 적용하려면 다음 옵션을 참조하세요.

- `1`: 암호화 전송 기능 사용 안함
- `2`: SQL 파라미터, Plain Text와 같은 민감한 속성에 대한 암호화 전송
- `3`: 모든 항목에 대한 암호화 전송

- **stat\_ip\_enabled** `Boolean`

기본값 `true`

[와탭 모니터링 서비스](#)의 [프로젝트 메뉴](#)에서 [통계/보고서](#) > [통계](#) > [클라이언트 IP](#) 항목의 IP 통계 사용 여부를 활성화합니다.

- **realtime\_user\_enabled** `Boolean`

기본값 `true`

[와탭 모니터링 서비스](#)의 [대시보드](#) > [애플리케이션 대시보드](#) 메뉴에서 확인할 수 있는 지표 중 [동시접속 사용자](#) 지표 수집 여부를 설정합니다.

- **hook\_direct\_patch\_classes** `ClassFile_FullPath`

직접적으로 특정 클래스를 로딩 타임에 바꿔치기할 때 사용합니다. 클래스를 컴파일한 후에 별도 파일을 만들고 그 파일의 전체 경로를 설정합니다.

- **active\_stack\_second** `Seconds`

기본값 `10`

액티브 스택을 추적하는 간격을 설정합니다.

! 주어진 기본값을 변경하지 않을 것을 권장합니다.

- **boot\_redefine\_size** `Int`

기본값 `100`

에이전트를 Attach 방식 또는 Watcher 방식으로 설치했을 경우 이미 로딩된 클래스의 추적을 위해 BCI를 새로 수행합니다. 이때 동시에 redefine하는 클래스의 개수를 설정합니다.

- **trace\_component\_enabled** `Boolean`

기본값 `true`

[와탭 모니터링 서비스](#)의 프로젝트 메뉴에서 [애플리케이션](#) > [인스턴스 성능 분석](#) 메뉴에서 [라이브러리 버전](#) 기능을 활성화합니다.

- **realtime\_user\_thinktime\_max / wclient\_thinktime\_max** `MiliSeconds`

기본값 `300000`

실시간 브라우저 사용자 수를 측정할 경우 동일 사용자로 인정되는 최대 호출 간격을 설정합니다.

! Java 에이전트 2.2.0 버전 이상의 경우 `wclient_thinktime_max` 옵션을 이용하세요.

- **time\_sync\_interval\_ms** `MiliSeconds`

기본값 `300000`

에이전트와 서버 간 동기화하는 시간 간격을 설정합니다. 동기화하지 않을 경우 `0`으로 설정하세요.

- **detect\_deadlock\_enabled** `Boolean`

기본값 `false`

Java 스레드의 교착 상태(DeadLock) 여부를 체크해 감지할 경우 이벤트를 발생합니다. 발생 간격은 5초 단위입니다. 같은 DeadLock 건에 대한 이벤트는 한 시간에 한 번만 발생합니다.

- **text\_reset** `Int`

기본값 `0`

와탭 에이전트는 한번 보낸 텍스트 유형 데이터를 hash 처리하므로 다음 날까지 다시 전송하지 않습니다. 이전 설정 값과 다른 값을 입력하는 경우 다시 전송합니다.

① 트랜잭션 URL, SQL String 등이 텍스트 유형 데이터에 해당합니다.



# CPU/메모리/디스크 사용량 수집

Java 에이전트가 CPU, 메모리, 디스크 사용량을 수집하는 방법을 안내합니다. 자바(Java)의 JMX 라이브러리를 기본적으로 활용하며, 필요에 따라 Linux의 `proc` 디렉터리나 `oshi`, `sigar` 라이브러리를 통한 성능 지표 수집 방법을 확인할 수 있습니다.

- **linux\_proc\_stat\_enabled** Boolean

기본값 `false`

CPU, 메모리, 디스크 사용량을 Linux의 `proc` 디렉터리에서 수집할지 여부를 설정합니다.

- **oshi\_enabled** Boolean

기본값 `false`

CPU, 메모리, 디스크 사용량을 측정할 때 `oshi` 라이브러리 사용 여부를 설정합니다.

ⓘ whatap.agent.2.1.0 버전 이상만 지원합니다.

- **sigar\_enabled** Boolean

CPU, 메모리, 디스크 사용량을 측정할 때 `sigar` 라이브러리 사용 여부를 설정합니다.

ⓘ whatap.agent.2.1.0 버전 미만에서 기본값은 `true` 이며, whatap.agent.2.1.0 버전 이상에서는 기본값이 `false` 입니다.

# 에이전트 통신 설정

에이전트가 애플리케이션 서버로부터 수집한 데이터를 서버로 전송하기 위한 네트워크 설정 방법을 안내합니다. 주요 설정 옵션으로는 수집 서버 호스트, 포트, 타임아웃 값, 네트워크 전송 크기, 데이터 전송 큐 크기 등이 있으며, 에이전트의 효율적인 데이터 관리 및 안정적인 서버 통신을 보장하는 데 중요합니다.

- **whatap\_server\_host** `ip_address`

기본값 `127.0.0.1,127.0.0.1`

에이전트가 수집한 데이터를 전송할 서버를 설정합니다. 수집 서버 이중화로 2개 이상의 IP 주소를 가진 경우 쉼표(,)를 구분자로 이용하세요. 설정한 IP 주소에는 수집 서버 proxy 서버가 리스닝(Listening) 상태로 서비스돼야 합니다.

- **whatap\_server\_port** `tcp_port`

기본값 `6600`

수집 서버 포트를 설정합니다. 포트는 하나만 설정할 수 있으므로 `whatap_server_host` 옵션에 설정한 수집 서버들은 동일 포트를 사용해야 합니다.

- **tcp\_so\_timeout** `MiliSecond`

기본값 `60000`

수집 서버와 통신하는 TCP 세션의 Socket Timeout 값을 설정합니다.

- **tcp\_connection\_timeout** `MiliSecond`

기본값 `5000`

수집 서버와 통신하는 TCP 세션의 Connection Timeout 값을 설정합니다.

- **net\_send\_max\_bytes** `Byte`

기본값 `5242880`

에이전트가 데이터를 수집하고 네트워크로 한 번에 전송할 수 있는 최대 byte 크기입니다.

- **net\_send\_queue1\_size** `Int`

기본값 `256`

트레이스 정보와 액티브 스택을 제외한 나머지 데이터 전송에 사용할 Queue의 크기를 설정합니다. 설정한 크기가 10 이하인 경우 10으로 설정합니다.

- **net\_send\_queue2\_size** `Int`

기본값 `512`

트레이스 정보와 액티브 스택 데이터 전송에 사용할 Queue의 크기를 설정합니다. 설정한 크기가 10 이하인 경우 10으로 설정합니다.

# 에이전트 성능

자바(Java) 에이전트의 성능 관련 옵션을 안내합니다. 동시에 처리할 수 있는 최대 트랜잭션 수와 트랜잭션 데이터를 저장하는 버퍼의 초기 크기 설정을 확인할 수 있습니다. 이는 시스템의 부하와 자원 사용을 최적화하는 데 중요한 역할을 합니다.

## whatap.agent.2.1.0 버전 미만

- **whatap.tx\_max\_count** `Int`

기본값 `5000`

트랜잭션을 최대 저장하는 개수입니다. 동시에 처리하는 트랜잭션이 많을 경우 해당 크기를 더 늘려주세요.

- **whatap.tx\_default\_capacity** `Int`

기본값 `101`

트랜잭션을 저장하는 버퍼의 시작 크기입니다.

# 에이전트 로그 설정

자바(Java) 애플리케이션 환경에서 발생하는 로그 데이터를 에이전트를 통해 관리하는 방법을 안내합니다. 로그 파일의 경로 및 이름 설정, 보관 기간 설정, 로그 모니터링, 사용자 정의 로그 추적 활성화 방법 등을 포함합니다. 로그 관리를 위한 다양한 설정 옵션을 제공해 시스템의 효율적인 로그 관리를 도와줍니다.

## 경로 및 이름 설정

에이전트 로그의 경로와 이름은 `whatap.conf` 파일에 설정할 수 있습니다. `log_root` 의 기본값으로 `${WHATAP_HOME}/logs` 경로를 사용합니다.

```
whatap.conf
```

```
log_root=System.getProperty("whatap.home", ".") + "/logs"
log_name=whatap
```

## 로그 관리 옵션

`whatap.conf` 파일에 로그 관련 옵션을 작성해 로그 내용을 관리할 수 있습니다.

```
whatap.conf
```

```
watchlog_enabled=true
log_rotation_enabled=true
log_otime_enabled=true
log_keep_days=7
```

- **watchlog\_enabled** Boolean

기본값 `false`

값이 `true` 이면 로그 모니터링을 활성화합니다.

- **watchlog\_check\_interval** Int

기본값 `2000`

로그를 감시하는 간격입니다. 매 간격으로 로그 파일에 기록을 추가했는지 확인합니다.

- **watchlog\_read\_count** **Int**

기본값 `8`

로그를 읽는 최대 횟수입니다. 한 번에 `watchlog_buffer_size` 크기 만큼 읽습니다.

- **watchlog\_buffer\_size** **Int**

기본값 `131072(128k)`

로그를 한 번에 읽는 로그 사이즈입니다.

- **watchlog.LOGKEY**

하나의 로그 파일을 감시하기 위한 설정입니다. `file` 이름을 입력하고 감시한 `word` 여러 개를 쉼표(,)로 구분해 설정합니다.

`check_interval` 간격으로 로그를 감시하다가 키워드를 발견하면 경고합니다. 한번 경고가 나가면 `silent` 만큼 경고를 멈춥니다.

- **watchlog.LOGKEY.enabled** **Boolean** 기본값 `true`

- **watchlog.LOGKEY.file** **String**

- **watchlog.LOGKEY.words** **String**

- **watchlog.LOGKEY.silent** **Int** 기본값 `10000`

- **watchlog.LOGKEY.check\_interval** **Int** 기본값 `1000`

- **log\_root** **String**

기본값 `${WHATAP_HOME}/logs`

에이전트 로그 경로를 설정합니다.

- **log\_datasource\_lookup\_enabled** **Boolean**

기본값 `true`

InitialContext Lookup 시 DataSource인 경우 로그를 기록하는 기능을 활성화합니다.

- **log\_rotation\_enabled** **Boolean**

기본값 `true`

에이전트 로그 파일을 날짜 별로 저장하는 기능을 활성화합니다. 로그 파일명은 `whatap-yyyymmdd.log` 형태로 저장합니다.

값이 `false` 이면 로그 파일명은 `whatap.log` 형태로 저장합니다.

- **log\_keep\_days** **Int**

기본값 `7`

로그 파일 보관 기간을 설정합니다. `log_rotation_enabled` 값이 `true` 인 경우에만 동작합니다.

- **log\_ophone\_enabled** **Boolean**

기본값 `false`

값이 true이면 각 로그마다 에이전트 이름을 추가합니다.

```

20220728 09:28:48(GMT) [ote-proxy] logsink send zip-mode enabled=true
20220728 09:28:48(GMT) [ote-notihu][BCI] httpc org/apache/http/impl/client/AbstractHttpClient
20220728 09:28:50(GMT) [ote-notihu][InnerMod] whatap.tracer.http.jar loaded 12962 bytes
20220728 09:28:51(GMT) [ote-notihu] logsink send zip-mode enabled=true
20220728 09:28:53(GMT) [ote-yard] logsink send zip-mode enabled=true
20220728 09:31:00(GMT) [ote-keeper][UrlNorm] Class Scan tot=9292 real=2897 classes 11ms done.
20220728 09:31:00(GMT) [ote-keeper][UrlNorm] Service PathTree done. BuildBy=FullBuild PathTree
20220728 09:31:00(GMT) [ote-gatewa][UrlNorm] Class Scan tot=13269 real=5816 classes 25ms done.

```

## 로그싱크(LogSink)

whatap.agent-2.1.0 버전 이상부터는 애플리케이션 서버를 통합 모니터링하는 로그싱크(LogSink) 기능을 제공합니다.

! LogSink 옵션은 whatap.agent-2.1.0 버전 이상에서 사용 가능합니다.

## 로그 모니터링 기능 활성화

- **logsink\_enabled** Boolean

기본값 `false`

로그 모니터링 기능을 활성화합니다.

- **logsink\_trace\_enabled** Boolean

기본값 `false`

Log에 트랜잭션 ID를 삽입하여, 트랜잭션 트레이스의 로그 탭을 노출할지 여부를 지정합니다.

## 프레임워크 로그내용 수집

프레임워크에서 수집하는 로그를 모니터링하려면 모듈에서 로그를 가로채기 위한 설정과 플러그인을 추가해야 합니다.

프레임워크에서 수집하는 로그의 기본 카테고리 이름을 'AppLog'로 설정합니다. 기본값은 `hooklog_enabled` 옵션의 설정과 동일합니다.

- **hooklog\_enabled** Boolean

기본값 `logsink_enabled`

로그 추적 활성화 여부를 설정합니다.

- ⓘ ◦ Java 에이전트 2.2.4 버전 이상에서는 애플리케이션의 재기동 없이 옵션의 변경 사항을 적용할 수 있습니다.
- Java 에이전트 2.2.4 버전 미만에서는 `hooklog_enabled` 값을 변경하면 재기동이 필요합니다. `hooklog_enabled` 기본값은 `logsink_enabled` 입니다. 따라서 `logsink_enabled=true` 로 설정하고 자바 애플리케이션을 재시작하면 `hooklog_enabled` 는 `true` 로 설정됩니다.

#### • `hooklog_custom_methods`

사용자 정의 로그를 등록합니다. 임의의 로그 프레임워크 내용을 전달합니다. 사이트에서 개별로 만든 로그 모듈의 로그를 추적할 때 사용하세요.

Java

```
package io.home.test;

public class MyLog {
    public void customLog(String log) { ... }
}
```

whatap.conf

```
hooklog_custom_methods=io.home.test.MyLog.customLog
```



# 트랜잭션

트랜잭션이란 사용자 브라우저의 요청을 처리하기 위한 서버 사이드의 Logical Unit of Work(LUW)를 말합니다. 개별 웹서비스(URL) 요청에 대한 처리 과정이 바로 트랜잭션입니다. 웹 애플리케이션에서 트랜잭션은 웹서비스(URL)에 대한 HTTP Request를 받아 Response를 반환하는 과정입니다.

애플리케이션의 성능은 이 트랜잭션들의 성능으로 요약할 수 있습니다. 트랜잭션 성능은 트랜잭션 시작에서부터 종료 시점, 응답시간 및 자원 사용량 혹은 트랜잭션 호출자 속성 등의 정보를 포함합니다.

기본적으로 트랜잭션 응답 분포와 트랜잭션 통계를 통해서 트랜잭션 성능을 분석할 수 있습니다.

## 트랜잭션 트레이싱

- **profile\_basetime** MiliSecond

기본값

트레이스의 상세 정보를 수집할 최소 응답 시간을 설정합니다. 설정한 값 이하의 시간 내에 트랜잭션이 종료된 경우 트레이스 정보를 수집하지 않습니다. 단, 5분 단위로 최초 호출된 URL과 에러가 발생한 트랜잭션에 대한 트레이스 정보는 수집합니다.

!  옵션에 상관없이 에러를 수집하려면  옵션을  로 설정하세요.  
기본값  로 설정하면 에러를 샘플링하게 됩니다

- **profile\_step\_normal\_count** Int

기본값

트랜잭션 트레이스의 최대 스텝 수를 설정합니다.

- **profile\_step\_heavy\_count** Int

기본값

Heavy한 스텝의 경우 트레이스 기본 스텝 수를 초과하더라도 정해진 값만큼 기록합니다.

- **profile\_step\_max\_count** Int

기본값

트레이스 스텝의 최대 수를 설정합니다. 수집된 트레이스 스텝 수가 이 값을 초과하면 이후 수집되는 스텝들은 모두 버려집니다.

을 최대 1000으로 설정하면  만큼 액티브 스택이 수집됩니다.

- **profile\_step\_heavy\_time** MiliSecond

기본값 100

Heavy한 스텝의 기준을 설정합니다. 설정한 값보다 수행 시간이 긴 경우 `profile_step_normal_count` 값을 초과하더라도 `profile_step_heavy_count`에 설정한 값만큼 기록합니다.

- **active\_stack\_count** *Int*

기본값 100

트랜잭션 내에서 수집하는 액티브 스택의 최대 수를 설정합니다.

- **profile\_method\_resource\_enabled** *Boolean*

기본값 false

트레이스에서 method 스텝을 수집할 때 해당 스텝에서 사용한 CPU와 메모리 사용량을 추적합니다.

- **profile\_position\_method** *String*

설정된 메소드를 수행하는 시점의 StackTrace를 기록합니다.

- **profile\_position\_depth** *Int*

기본값 50

position 추적을 위해 StackTrace를 기록할 때 최대 라인 수를 설정합니다.

- **trace\_error\_callstack\_depth** *Int*

기본값 50

Error 발생 시 수집하는 StackTrace의 최대 라인 수를 설정합니다.

- **trace\_active\_callstack\_depth** *Int*

기본값 50

액티브 스택에서 수집하는 StackTrace의 최대 라인수를 설정합니다.

- **trace\_active\_transaction\_yellow\_time** *MiliSecond*

기본값 3000

액티브 트랜잭션의 아카이퀼라이저에서 노란색으로 표현할 기준을 설정합니다.

- **trace\_active\_transaction\_red\_time** *MiliSecond*

기본값 8000

액티브 트랜잭션의 아카이퀼라이저에서 빨간색으로 표현할 기준을 설정합니다.

- **hook\_method\_patterns** *String*

응답시간을 측정할 메소드를 설정합니다. 마지막 마침표(.)를 구분자로 클래스 FullName과 메소드로 구분되며 별표(\*)를 와일드 카드로 사용할 수 있습니다. 대상이 여러 개인 경우 쉼표(,)로 구분합니다.

```
whatap.conf
```

```
hook_method_patterns=a.b.C1.*
```

- **hook\_method\_supers** *String*

특정 클래스를 상속받은 메소드의 응답시간을 측정하고자 할 때 Super Class를 설정합니다. 클래스 FullName을 설정하며 대상이 여러 개인 경우 쉼표(,)로 구분합니다.

```
whatap.conf
```

```
hook_method_supers=a.b.C1
```

- **hook\_method\_interfaces** *String*

특정 인터페이스를 구현한 메소드의 응답시간을 측정하고자 할 때 Interface를 설정합니다. 인터페이스 FullName을 설정하며 대상이 여러 개인 경우 쉼표(,)로 구분합니다.

- **hook\_method\_ignore\_classes** *String*

메소드 트레이스를 설정할 때 트레이스에서 제외하고 싶은 클래스들을 설정합니다.

- **hook\_method\_access\_public\_enabled** *Boolean*

기본값  true

메소드 트레이스를 설정할 때 public 메소드에 대해서만 별도 대상으로 처리할지 여부를 설정합니다.

- **hook\_method\_access\_private\_enabled** *Boolean*

기본값  false

메소드 트레이스를 설정할 때 private 메소드에 대해서만 별도 대상으로 처리할지 여부를 설정합니다.

- **hook\_method\_access\_protected\_enabled** *Boolean*

기본값  true

메소드 트레이스를 설정할 때 protected 메소드에 대해서만 별도 대상으로 처리할지 여부를 설정합니다.

- **hook\_method\_access\_none\_enabled** *Boolean*

기본값  true

메소드 트레이스를 설정할 때 no access indicated 메소드에 대해서만 별도 대상으로 처리할지 여부를 설정합니다.

- **stacklog\_socket\_port** *TCP\_PortNumber*

기본값

목적지의 TCP 포트를 설정하면 Socket.connect() 시점 StackTrace를 에이전트 로그로 기록합니다. 기본 설정으로 확인되지 않는 DB 연결, HTTP 연결 등을 추적할 때 사용할 수 있습니다.

⚠️ 설정한 목적지로 연결 시마다 매번 StackTrace를 기록합니다. 성능 저하를 유발할 수 있습니다. 디버깅 용도로 선별한 에이전트에 한시적으로만 사용해야 합니다.

DB 연결 상태 추적 예시, `stacklog_socket_port=1521`

## 예외 처리

### • biz\_exceptions String

Business Exception을 등록합니다.

✔️ 에러 클래스 수집 / 히트맵 표시 레벨: INFO(파란색) / 에러 통계 포함

ⓘ Java 에이전트 [v2.2.20](#) 이상에서는 등록된 비즈니스 Exception을 에러 통계로 수집합니다. v2.2.20 미만에서는 등록된 비즈니스 Exception을 에러 통계에서 제외합니다. 다만 트레이스 상세에서는 나타납니다.

### • biz\_exceptions\_status String Java Agent v2.2.20 or later

Business Exception 발생 시 이벤트 알림을 억제할 Business Exception과 Status를 설정합니다. 이 옵션에서 설정한 **Business Exception:Status**에 대해서는 이벤트 레벨이 INFO로 변경됩니다. Business Exception과 Status는 콜론(:)으로 구분하며, 하나의 Business Exception에 여러 개의 Status를 사용하는 경우 앰퍼샌드(&)로 구분합니다. 심표를 구분자로 이용해 여러 개를 설정할 수 있습니다.

whatap.conf

# example

biz\_exceptions\_status=java.util.concurrent.TimeoutException:400&404&408,java.lang.Exception:200

✔️ 에러 클래스 수집 / 히트맵 표시 레벨: INFO(파란색) / 에러 통계 포함

### • ignore\_exceptions String

등록한 Exception의 에러 자체를 무시합니다.

✔ 에러 클래스 무시 / 히트맵 표시 레벨: **INFO**(파란색) / 에러 통계 미포함

- **transaction\_status\_error\_enable** Boolean

기본값 `true`

HTTP 401, 403과 같이 정상 응답이 아닌 HTTP 상태 코드를 반환하는 경우 에러로 처리할지 여부를 설정합니다.

- **status\_ignore** String

무시하려는 HTTP 상태 코드를 설정할 수 있습니다. 여러 값을 대상으로 할 경우 쉼표(,)를 구분자로 사용하세요.

```
whatap.conf
# example
status_ignore=408,500
```

✔ Status 에러 무시 / 히트맵 표시 레벨: **INFO**(파란색) / 에러 통계 미포함

- **status\_ignore\_set** String

`whatap.error.STATUS_ERROR` 를 무시합니다. 여러 값을 대상으로 할 경우 쉼표(,)를 구분자로 사용하세요.

```
whatap.conf
# example
status_ignore_set=/api/test/timeout/{time}:408,/api/test/timeout:200,/error:500
```

✔ Status 에러 세트 무시 / 히트맵 표시 레벨: **INFO**(파란색) / 에러 통계 미포함

- **httpc\_status\_error\_enable** Boolean

기본값 `true`

HTTP 상태 코드가 에러인 경우 수집 여부를 설정합니다. 클라이언트 에러 응답(`400` 이상), 서버 에러 응답(`500` 이상)이 해당됩니다.

ⓘ HTTP 상태 코드에 대한 자세한 내용은 [다음 링크](#)를 참조하세요.

- **http\_status\_ignore** String

HTTP 상태 코드가 에러(`HTTPC_ERROR`)인 경우 무시할 수 있습니다. 여러 값을 대상으로 할 경우 쉼표(,)를 구분자로 사용하세요.

✔ HTTP 상태 에러 코드 무시 / 에러 통계 미포함

- **http\_status\_url\_ignore\_set** String

`whatap.error.HTTPC_ERROR` 를 무시합니다. 여러 값을 대상으로 할 경우 쉼표(,)를 구분자로 사용하세요.

`http_status_url_ignore_set=/a/b/c:400,/a/ab/c:404` 와 같이 설정하세요. 이때 url은 통계/에러 분석에서 `HTTPC_URL` 값을 입력합니다.

✔ HTTP 상태 에러 코드 무시 / 에러 통계 미포함

## HTTP 트랜잭션 추적

- **profile\_http\_header\_enabled** Boolean

기본값 `false`

트레이스 내역에 http 헤더 정보를 기록하려면 `true` 로 설정하세요.

- **profile\_http\_parameter\_enabled** Boolean

기본값 `false`

트레이스 내역에 http 파라미터 정보를 기록하려면 `true` 로 설정하세요. 파라미터는 별도 보안키를 입력해야 조회할 수 있습니다.

- ⓘ
  - **Java 에이전트 2.2.2 버전 이전:** 보안 키는 WAS 서버 `${WHATAP_AGENT_HOME}/paramkey.txt` 파일 내에 6자리로 작성합니다. `paramkey.txt` 파일이 존재하지 않는 경우 랜덤 값으로 자동 생성합니다.
  - **Java 에이전트 2.2.2 버전 이후:** 보안 키는 WAS 서버 `${WHATAP_AGENT_HOME}/security.conf` 파일 내에 `paramkey` 키값을 확인하세요. `security.conf` 파일이 존재하지 않을 경우 `paramkey` 키값을 **WHATAP**으로 자동 생성합니다.
  - 보안키 설정 파일에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

- **profile\_http\_header\_url\_prefix** String

트레이스 내역에 http 헤더 정보를 기록할 대상 URL의 prefix를 정의할 때 사용합니다.

- **profile\_http\_parameter\_url\_prefix** String

트레이스 내역에 http 파라미터 정보를 기록할 대상 URL의 prefix를 정의할 때 사용합니다.

- **trace\_transaction\_name\_key** String

HTTP request parameter 값을 해당 옵션에 설정하면, 파라미터 값을 추출하여 트랜잭션의 이름 마지막에 추가합니다.

예를 들어, HTTP 파라미터로 `paramKey` 를 가진 `/api/test` URL을 호출하는 경우 다음 예제와 같이 설정하면 트레이스 내역에 `/api/test$paramKey={value}` 로 트랜잭션 이름을 표시합니다. 다만 여러 개의 파라미터를 등록할 수는 없습니다.

```
whatap.conf
```

```
# example
```

```
trace_transaction_name_key=paramKey
```

- **trace\_normalize\_enabled** Boolean

기본값 `true`

트랜잭션 URL을 파싱해 정규화하는 기능을 활성화합니다.

❗ `false` 로 값을 변경하면 패스 파라미터 파싱을 비활성화합니다. 이 경우 통계 데이터의 의미가 약화됨으로 디버그 용도로만 잠시 사용하는 것을 권장합니다.

- **trace\_auto\_normalize\_enabled** Boolean

기본값 `true`

트랜잭션 URL 정규화할 때 패턴 값을 어노테이션에서 추출해 자동으로 파싱하는 기능을 활성화합니다.

- **trace\_normalize\_urls** String

정규화할 트랜잭션 URL 패턴을 설정합니다. 호출 URL 패턴을 파싱해 패스 파라미터를 제거합니다.

❗ 예시, `/a/{v}/b` 라고 선언하면 `a/123/b` → `a/{v}/b` 로 치환합니다. 여러 개를 등록할 때는 쉼표(,)를 구분자로 사용하세요. 치환 패턴 정리 후 보완이 필요합니다.

- **web\_static\_content\_extensions** String

기본값 `js, htm, html, gif, png, jpg, css, swf, ico`

스태틱 콘텐츠를 판단하는 확장자를 설정합니다. 이 옵션에 설정한 확장자를 가진 트랜잭션들은 트레이스 추적과 카운팅에서 제외합니다.

- **trace\_transaction\_name\_header\_key** String

설정된 HTTP 헤더 키에서 추출한 값을 트랜잭션의 이름 마지막에 추가합니다.

- **recursive\_max** `Int`

기본값 `1000000`

트랜잭션의 재귀 호출 여부 검출을 위한 옵션입니다. 단일 트랜잭션으로부터 파생되는 재귀 호출 횟수를 카운트하여 이벤트 알림을 발행하기 위한 기준을 설정합니다.

ⓘ HTTP URL 재귀 호출을 대상으로 합니다. `jsp:forward`를 통해 재호출하는 케이스도 카운트에 포함합니다.

- **hook\_httpservlet\_classes** `String`

HTTP 트랜잭션의 END POINT를 추가로 설정합니다. 메소드의 첫 번째 2개의 파라미터는 `HttpServletRequest` 와 `HttpServletResponse` 만 설정할 수 있습니다.

- **hook\_jsp\_patterns** `String`

기본값 `org.apache.jasper.servlet.JspServlet.servicejspFile`

JSP 파일을 로딩하는 메소드를 설정합니다. 트랜잭션 호출 결과로 반환하는 JSP 정보를 트레이스에 표시합니다. 이 옵션을 통해 추가한 설정에 기본값이 자동 추가됩니다.

- **trace\_ignore\_url\_set** `String`

트랜잭션 추적에서 제외할 URL을 설정합니다. 2개 이상의 값을 설정하려면 쉼표(,)를 구분자로 이용하세요.

- **trace\_ignore\_url\_prefix** `String`

트랜잭션 추적에서 제외할 URL prefix를 설정합니다. 2개 이상의 값을 설정하려면 쉼표(,)를 구분자로 이용하세요.

- **ignore\_http\_method** `String`

기본값 `PATCH,OPTIONS,HEAD,TRACE`

설정된 HTTP 메소드(Method)로 요청된 트랜잭션 정보는 수집하지 않습니다. 여러 개를 설정하려면 쉼표(,)를 구분자로 이용하세요.

```
whatap.conf
```

```
# e.g. http_method가 OPTIONS, HEAD인 트랜잭션 "추적"
ignore_http_method=PATCH,TRACE
```

```
# e.g. http_method가 OPTIONS, HEAD인 트랜잭션 "무시"
ignore_http_method=OPTIONS,HEAD
```



# NON HTTP 트랜잭션 추적

- **trace\_auto\_transaction\_enabled** Boolean

기본값 `false`

트레이스 대상 메소드가 트랜잭션 시작점( `Javax.http.httpservlet`, `hook_service_*` ) 내에서 수행하는 경우가 아니라면 수집되지 않습니다. 이 경우 트레이스 대상 메소드가 트랜잭션 시작점이 되도록 설정합니다.

! 주로 개발 환경에서 백그라운드 트랜잭션의 END POINT를 찾을 때 사용합니다.

- **trace\_auto\_transaction\_backstack\_enabled** Boolean

기본값 `true`

`trace_auto_transaction_enabled` 옵션의 값이 `true` 인 경우 트랜잭션 시작 시 StackTrace를 기록합니다. 이를 통해 트랜잭션의 시작점을 찾아낼 수 있습니다.

- **trace\_background\_socket\_enabled** Boolean

기본값 `true`

트랜잭션이 아닌 백그라운드 스레드에 의한 소켓이 오픈될 때도 이를 기록합니다.

- **async\_stack\_enabled** Boolean

기본값 `false`

백그라운드 스레드에 대한 Active Stack 기능 사용 여부를 설정합니다.

- **async\_thread\_match** String

액티브 스택을 덤프할 백그라운드 스레드 이름을 설정합니다. 여러 개를 설정하려면 쉼표(,)를 구분자로 사용하세요. 이름을 설정할 때 'Thread-\*' 처럼 별표(\*)를 사용해 비교 패턴을 사용할 수 있습니다.

- **async\_thread\_parking\_class** String

기본값 `sun.misc.Unsafe`

스택의 Top 메소드가 `async_thread_parking` 에 등록된 클래스/메소드일 때 스레드가 파킹 상태에 있다고 판단하고 덤프를 생성하지 않습니다.

- **hook\_service\_patterns** String

NON-Http 트랜잭션 추적을 위한 시작점 패턴을 설정합니다. 와일드카드 문자를 사용할 수 없으며, 2개 이상의 값을 설정하려면 쉼표(,)를 구분자로 이용하세요.

- **hook\_serivce\_ignore\_methods** String

`hook_service_patterns`에서 설정한 내역 중 시작점으로 불필요한 메소드를 추가할 수 있습니다.

- **hook\_service\_supers** String

NON-HTTP 트랜잭션 추적을 위한 시작점의 공통 분모가 특정 클래스의 메소드를 상속 받은 경우라면 공통 분모의 메소드를 설정합니다. 메소드 전체 경로를 입력하세요. 와일드 카드(\*)를 사용할 수 있으며, 쉼표(,)를 구분자로 이용해 여러 개의 메소드를 등록할 수 있습니다.

```
whatap.conf
```

```
hook_service_supers=a.b.C.method,a.b.C.*
```

- **hook\_service\_interfaces** String

NON-Http 트랜잭션 추적을 위한 시작점의 공통 분모가 특정 인터페이스를 구현한 경우라면 이를 설정합니다.

- **hook\_service\_access\_public\_enabled** Boolean

기본값 `true`

Non Http Demon 프로세스의 트랜잭션을 설정할 때 public 메소드에 대해서만 Access 권한 기준을 on/off 설정합니다.

- **hook\_service\_access\_private\_enabled** Boolean

기본값 `true`

Non Http Demon 프로세스의 트랜잭션을 설정할 때 private 메소드에 대해서만 Access 권한 기준을 on/off를 설정합니다.

- **hook\_service\_access\_protected\_enabled** Boolean

기본값 `true`

Non Http Demon 프로세스의 트랜잭션을 설정할 때 protected 메소드에 대해서만 Access 권한 기준을 on/off를 설정합니다.

- **service\_name\_mode** String

기본값 `full`

트랜잭션 명으로 다음의 옵션을 사용할 수 있습니다.

- `full` : Full Class 이름 사용
- `class` : 서비스 명칭을 Class 이름으로 사용
- `method` : 서비스 명칭을 Method 이름으로 사용
- `string` : 서비스 명칭을 문자열 중에서 첫 번째 파라미터로 사용
- `arg` : 파라미터 중에서 `service_name_index` 옵션에 설정한 인덱스에 파라미터를 서비스 명칭으로 사용

## 멀티 트랜잭션 추적

- **mtrace\_enabled** Boolean

기본값 `true`

멀티 트랜잭션 추적 기능(Multi Transaction ID, 이하 MTID) 사용 여부를 설정합니다. MTID를 추적하면 등록된 모든 애플리케이션 간의 호출을 확인할 수 있습니다.

- **mtrace\_rate** Percentage

기본값 `10`

최초 트랜잭션이 발생할 때 발급하는 MTID의 발급 비율을 설정하는 옵션입니다.

- **mtrace\_caller\_key** String

기본값 `x-wtap-mst`

MTID 추적에 사용할 Caller Key Name을 설정합니다.

- **mtrace\_callee\_key** String

기본값 `x-wtap-tx`

MTID 추적에 사용할 Callee Key Name을 정합니다.

- **mtrace\_send\_url\_length** Int

기본값 `80`

Http Caller는 Callee에게 자신의 URL을 넘겨줍니다. 이때 URL 길이를 제한하는데, 이 길이의 값을 설정합니다.

- **mtrace\_callee\_id\_send\_enabled** Boolean

기본값 `false`

MTID 추적 시 HTTP 호출과 함께 발행된 Callee ID를 트레이스에 표현합니다.

- **mtrace\_callee\_id\_rcv\_enabled** Boolean

기본값 `false`

MTID 추적 시 수신한 Callee ID를 트레이스에 표현합니다.

- **mtrace\_alltx\_enabled** Boolean

기본값 `false`

log4j와 같은 로깅 시스템과 연계하기 위해서 모든 트랜잭션의 MTID를 추적할 수 있습니다.

- **mtrace\_basetime** Miliseconds

기본값 100

`mtrace_alltx_enabled` 옵션의 값이 `true` 이면 너무 많은 로그가 남을 수 있습니다. 이때는 트레이스 로깅량을 줄일 필요가 있습니다.

- **stat\_mtrace\_enabled** Boolean

기본값 false

Caller와 Callee의 상관관계 통계를 수집합니다. Caller에 적용하면 Caller의 상세 정보를 보내주고 Callee에 적용하면 url 단위 Caller-Callee 호출 통계를 수집 서버로 전송합니다.

## 트레이스 데이터 샘플링

Java Agent v2.2.4 or later

- **trace\_sampling\_enabled** Boolean

기본값 false

트레이스 데이터 샘플링 수집 여부를 설정할 수 있습니다.

- **trace\_sampling\_tps** Int

기본값 10000

트레이스 데이터 샘플링 수를 설정할 수 있습니다. 설정한 샘플링 수를 초과하는 데이터는 전송하지 않으며, 5초마다 초기화합니다.

- **trace\_send\_enabled** Boolean

기본값 true

트레이스 데이터 전송 여부를 설정할 수 있습니다.

- **debug\_trace\_samling** Boolean

기본값 false

트레이스 데이터 샘플링 디버그 옵션을 켜거나 끌 수 있습니다.

## 트레이스 URL 이름 수정하기

- **hook\_tx\_name\_patterns / hook\_tx\_name\_mode** string Java Agent v2.2.4 or later

특정 URL 및 특정 메소드(method)를 호출 시 트레이스 URL의 이름을 수정할 수 있습니다. 예를 들어

`io.home.test.TestController.test1` 에 등록된 URL 호출 시 `test100000()` 메소드를 호출하는 경우, 서비스 화면에서 `/api/100000+test1+test100000` 으로 확인할 수 있습니다.

`hook_tx_name_mode` 옵션을 통해 **class, method, string, return** 중 하나를 선택할 수 있습니다.

```
whatap.conf
```

```
# 호출 메소드의 전체 경로 (* 사용 가능)
hook_tx_name_patterns=ab.cd.ef.GH.ij

# class, method, string, return 중 택1
hook_tx_name_mode=method
```

## 특정 exception 무시하기

- `ignore_exception_tx_pattern` **String** [Java Agent v2.2.4 or later](#)

특정 서비스에서 특정 exception을 무시할 수 있도록 설정할 수 있습니다. 여러 개를 등록할 때는 쉼표(,)를 구분자로 이용하세요.

```
whatap.conf
```

```
ignore_exception_tx_pattern=exception:service_url
# e.g. java.util.concurrent.TimeoutException:/api/test/timeout,org.springframework.web.util.NestedServletException:/api/
posts/test/cexception
```

## URLConnection 추적하기

- `URLConnection` **Boolean**

기본값 `true`

`URLConnection` 클래스의 메소드를 추적하기 위한 옵션입니다.

- `hook_URLConnection_startup_enabled` **Boolean** [Java Agent v2.2.4 or later](#)

기본값 `false`

`URLConnection` 클래스의 메소드를 최초 호출 시에도 추적할 수 있도록 설정할 수 있습니다.

- `URLConnection_weblogic` **Boolean** [Java Agent v2.2.4 or later](#)

기본값 `true`

Weblogic에서 `URLConnection` 으로 http 호출을 하는 경우 `weblogic.net.http.URLConnection` 클래스의 메소드를 추적하기 위한 옵션입니다.

- `hook_URLConnection_weblogic_startup_enabled` **Boolean** [Java Agent v2.2.4 or later](#)

기본값 `false`

`weblogic.net.http.HttpURLConnection` 클래스의 메소드를 최초 호출 시에도 추적할 수 있도록 설정할 수 있습니다.

# DB, SQL

데이터베이스 및 SQL 성능 데이터 수집을 위한 자바(Java) 에이전트의 다양한 옵션을 안내합니다. DBCP, Hikari, Tomcat 등 다양한 DB 연결 풀(Connection Pool) 정보 추적 및 SQL 실행 세부 사항을 기록할 수 있습니다. 또한, SQL 파라미터 정보 기록, DB 연결 누수 추적 등 성능 최적화에 필수적인 정보 수집 옵션을 제공합니다.

- **dbcp\_pool\_enabled** Boolean

기본값  true

JMX를 사용하지 않고 DBCP의 DB Connection 정보를 추적하기 위해 사용합니다.

- **hikari\_pool\_enabled** Boolean

기본값  false

JMX를 사용하지 않고 hikari pool의 DB Connection 정보를 추적하기 위해 사용합니다.

- **tomcat\_ds\_enabled** Boolean

기본값  false

JMX를 사용해 Tomcat DB Connection Pool 정보를 추적하는 기능을 활성화합니다.

- **tomcat\_pool\_enabled** Boolean

기본값  true

JMX를 사용하지 않고 Tomcat DB Connection Pool 정보를 추적하는 기능을 활성화합니다.

- **weblogic\_ds\_enabled** Boolean

기본값  false

JMX를 사용해 Weblogic DB Connection Pool 정보를 추적하는 기능을 활성화합니다.

- **weblogic\_pool\_enabled** Boolean

기본값  true

JMX를 사용하지 않고 Weblogic DB Connection Pool 정보를 추적하는 기능을 활성화합니다.

- **jeus\_pool\_enabled** Boolean

기본값  true

JMX를 사용하지 않고 JEUS DB Connection Pool 정보를 추적하는 기능을 활성화합니다.

- **profile\_connection\_open\_enabled** Boolean

기본값 `true`

트레이스 내역에 DBConnection 오픈 정보를 기록합니다.

- **profile\_dbc\_close** Boolean

기본값 `false`

`profile_connection_open_enabled`의 값이 `true`인 경우에만 동작합니다. 트레이스 내역에 DBConnection 클로즈 정보를 기록합니다.

- **profile\_sql\_param\_enabled** Boolean

기본값 `false`

트레이스 내역에 SQL 파라미터 정보를 기록할 때 사용합니다. 파라미터는 별도 보안 키를 입력해야 조회할 수 있습니다.

- ⓘ ◦ **Java 에이전트 2.2.2 버전 이전:** 보안 키는 WAS 서버 `${WHATAP_AGENT_HOME}/paramkey.txt` 파일 내에 6자리로 작성합니다. `paramkey.txt` 파일이 존재하지 않는 경우 랜덤 값으로 자동 생성합니다.
- **Java 에이전트 2.2.2 버전 이후:** 보안 키는 WAS 서버 `${WHATAP_AGENT_HOME}/security.conf` 파일 내에 `paramkey` 키값을 확인하세요. `security.conf` 파일이 존재하지 않을 경우 `paramkey` 키값을 **WHATAP**으로 자동 생성합니다.
- 보안키 설정 파일에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

- **profile\_sql\_resource\_enabled** Boolean

기본값 `false`

트레이스에서 SQL을 수집할 때 해당 스텝에서 사용한 CPU와 메모리 사용량을 추적합니다.

- **profile\_update\_count** Boolean

기본값 `false`

`executeUpdate()` 메소드를 통해 SQL UPDATE 문을 수행한 경우 UPDATE 건수를 수집합니다.

- **custom\_pool\_classes** String

pre-define되지 않는 별도의 Connection Pool을 사용하는 경우 해당 클래스 명을 설정합니다.

- **ds\_update\_interval** MiliSeconds

기본값 `5000`

DB Connection 정보 Count 간격을 설정합니다.

- **profile\_position\_sql** Boolean



기본값 `false`

SQL을 수행하는 시점의 StackTrace를 기록합니다.

- **profile\_sql\_param\_length** `Int`

기본값 `40`

SQL 파라미터의 길이를 설정합니다.

❗ 최대값은 128이며 그 이상의 값을 입력하더라도 파라미터는 128까지 저장합니다.

- **trace\_dbc\_leak\_enabled** `Boolean`

기본값 `false`

DBConnection Leak을 추적하는 기능을 활성화합니다.

❗ Connection Wrapper를 사용해 Leak을 추적하기에 운영 서비스에 영향을 미칠 수 있으므로 반드시 테스트 후 적용하세요.

- **trace\_dbc\_leak\_fullstack\_enabled** `Boolean`

기본값 `false`

DBConnection Leak이 감지되는 경우 해당 시점 StackTrace를 수집합니다.

❗ 피크 타임(Peak Time)에는 가급적 적용하지 마세요. 옵션을 적용하면 CPU 사용량이 다소 증가할 수 있습니다. 문제 해결 용도로만 한시적으로 적용할 것을 권고합니다.

- **trace\_sql\_normalize\_enabled** `Boolean`

기본값 `true`

SQL 문에서 리터럴 부분을 추출해 SQL 문을 정규화하는 기능을 활성화합니다.

- **profile\_error\_jdbc\_fetch\_max** `Int`

기본값 `10000`

SQL Fetch Count(`ResultSet.next()` 호출 건 수)가 설정한 값을 초과하면 TOO MANY Fetch 에러로 처리합니다. `0`으로 설정하면 에러 처리하지 않습니다.

- **profile\_error\_sql\_time\_max** `Int`

기본값 30000

SQL 수행 시간이 설정한 값을 초과하면 TOO SLOW 에러로 처리합니다. 0으로 설정하는 경우 에러 처리를 하지 않습니다.

- **hook\_connection\_open\_patterns** String

DB Connection Open 시 호출하는 메소드를 등록합니다. 미리 설정하지 않은 Connection Pool의 getConnection을 등록하는 것이 일반적입니다.

Example

```
hook_connection_open_patterns=mypool.ConPool.getConnection
```

- **hook\_jdbc\_con\_classes** String

미등록한 JDBC Connection 클래스를 지정합니다.

Example

```
hook_jdbc_con_classes=mypool.ConPool
```

- **hook\_jdbc\_pstmt\_classes** String

미등록한 jdbc PreparedStatement 클래스를 설정합니다. 생성자 파라미터에 SQL 문자열이 전달되는 구조여야 한다는 것을 주의하세요.

Example

```
hook_jdbc_pstmt_classes=org.apache.derby.impl.jdbc.EmbedPreparedStatement
```

- **hook\_jdbc\_cstmt\_classes** String

미등록한 jdbc CallableStatement 클래스를 설정합니다.

Example

```
hook_jdbc_cstmt_classes=org.apache.derby.impl.jdbc.EmbedCallableStatement
```

- **hook\_jdbc\_stmt\_classes** String

미등록한 JDBC Statement 클래스를 설정합니다.

Example

```
hook_jdbc_stmt_classes=org.apache.derby.impl.jdbc.EmbedStatement
```

- **hook\_jdbc\_rs\_classes** `String`

미등록한 JDBC ResultSet 클래스를 설정합니다.

Example

```
hook_jdbc_rs_classes=org.apache.derby.impl.jdbc.EmbedResultSet
```

- **hook\_jdbc\_wrapping\_driver\_patterns** `String`

DB2 드라이버처럼 단독 처리한 JDBC 드라이버는 `hook_jdbc_xxx` 옵션으로 직접 BCI가 어렵습니다. 이런 경우 Wrapper 방식으로 SQL을 추적할 수 있습니다. 이때 `Driver.connect` 를 설정해 추적합니다.

- **debug\_dbc\_stack\_enabled** `Boolean`

기본값 `false`

DB Connection 시점의 StackTrace를 트레이스에 저장합니다. 애플리케이션에서 사용하는 Connection Pool 정보를 얻기 위해 사용합니다.

- **ignore\_sql\_hash\_set** `String` `Java Agent v2.2.4 or later`

해당 옵션에 입력한 SQL 해시값과 일치하는 SQL 문은 무시하도록 설정할 수 있습니다.

# HTTPC, API Call

HTTP 외부 호출과 API 호출 관련 데이터를 수집, 분석을 위한 자바(Java) 에이전트의 옵션 설정 방법을 안내합니다. TOO SLOW 에러 처리 및 CPU와 메모리 사용량 추적, 호출 시점의 스택 트레이스 기록, URL 정규화 등 세밀한 모니터링을 위한 다양한 옵션을 제공합니다. 이를 통해 개발자는 애플리케이션의 외부 의존성을 효율적으로 관리하고 성능 문제를 식별할 수 있습니다.

- **profile\_error\_httpc\_time\_max** `Int`

기본값 `10000`

HTTPC 수행 시간이 지정한 값을 초과하면 TOO SLOW 에러로 처리합니다. `0`으로 설정하면 에러 처리하지 않습니다.

- **profile\_httpc\_resource\_enabled** `Boolean`

기본값 `false`

트레이스에서 HTTP Call 스텝을 수집할 때 해당 스텝에서 사용한 CPU와 메모리 사용량을 추적합니다.

- **profile\_position\_httpc** `Boolean`

기본값 `false`

HTTPC가 수행하는 시점의 StackTrace를 기록합니다.

- **trace\_httpc\_normalize\_enabled** `Boolean`

기본값 `true`

트랜잭션 내 HTTPC URL을 파싱해 정규화하는 기능을 활성화합니다.

- **trace\_httpc\_normalize\_urls** `String`

정규화할 HTTPC URL 패턴을 설정합니다. 호출 URL 패턴을 파싱해 패스 파라미터를 제거합니다.

❗ 예시, `/a/{v}/b` 라고 선언하면 `a/123/b` → `a/{v}/b` 로 치환합니다. 여러 개를 등록할 때는 쉼표(,)를 구분자로 사용합니다. 치환 패턴 정리 후 보완이 필요합니다.

- **hook\_httpc\_patterns** `String`

HTTP outbound 호출을 수행하는 `full package 클래스명.메소드` 를 설정합니다. HTTP Call을 수행하는 메소드(method)의 전체 경로(full path)를 등록해 사용합니다. 등록 후 클래스 재정의(class redefine)를 하거나 에이전트를 재시작해야 합니다.

```
whatap.conf
```

```
hook_httpc_patterns=io.home.test.baseapp.app.post.service.HookHttpService.*
```

- ⓘ ◦ 클래스를 여러 개 등록하려면 쉼표(,)를 구분자로 사용하세요.
- 와일드 카드는 별표(\*)만 사용할 수 있으며 정규식은 사용할 수 없습니다.

# 스크립트 플러그인

Java 에이전트에서 제공하는 플러그인 옵션을 통해 사용자가 원하는 코드를 트레이스 데이터에 주입하거나 메소드 수행 전후로 부가 정보를 추가하는 방법을 제공합니다. 메소드 시작/종료 시점에 특정 코드를 실행할 위치를 설정하는 옵션과 사용자 정의 pool을 모니터링하기 위한 클래스 설정 방법을 포함합니다. 실제 플러그인 적용 사례를 통해 구체적인 활용 방안을 제시하며, 모니터링의 유연성과 세밀함을 높일 수 있는 다양한 설정 예시와 API 사용법을 안내합니다.

## 플러그인 옵션

Java 에이전트 플러그인은 메소드 시작/종료 부분에 삽입해 실행합니다. 플러그인을 적용할 위치(클래스, 메소드)는 에이전트 옵션으로 설정할 수 있습니다.

- hook\_trace\_helper\_start\_patterns

메소드 시작 부분에 트레이스 플러그인을 삽입할 지점(클래스, 메소드)을 설정합니다. 플러그인 코드는 `$WHATAP_HOME/plugin/TraceHelperStart.x` 파일에 작성하세요.

- hook\_trace\_helper\_end\_patterns

메소드 종료 부분에 트레이스 플러그인을 삽입할 지점(클래스, 메소드)를 설정합니다. 플러그인 코드는 `$WHATAP_HOME/plugin/TraceHelperEnd.x` 파일에 작성하세요.

- hook\_trace\_helper\_patterns

메소드 시작/종료 양쪽에 트레이스 플러그인을 삽입할 지점(클래스, 메소드)을 설정합니다. 플러그인 코드는 `$WHATAP_HOME/plugin/TraceHelperStart.x`, `$WHATAP_HOME/plugin/TraceHelperEnd.x` 파일에 각각 작성하세요.

- custom\_pool\_classes

사용자 정의 pool을 모니터링 하기 위해 pool 사용량 정보를 가진 클래스를 설정합니다. 플러그인 코드는 `$/WHATAP_HOME/plugin/CustomPool.x`에 작성합니다.

## 플러그인 설정 예시

여러 개의 클래스를 설정하는 경우 쉼표(,)를 구분자로 이용하세요. 패키지 이름의 문자열 일부 혹은 전부를 '\*'로 치환할 수 있습니다.

- 패키지와 메소드 이름을 설정하는 경우

```
whatap.bytecode.instrument.PluginTestA.testA,whatap.bytecode.instrument.PluginTestB.testB
```

- 패키지와 메소드 이름을 \*로 치환하는 경우

```
*PluginTestA.testA,whatap.bytecode.instrument.PluginTestB.*
```

```
*.testA, *PluginTestB.testB
```

- 전체를 대상으로 하는 경우

```
*.*
```

- Custom Pool을 설정하는 경우

Custom Pool은 식별자@패키지 명 형태로 설정합니다.

예시, Class 명 앞에 `whatap_plugin_guide` 라는 식별자를 설정니다. 식별자와 클래스는 @로 구분합니다.

```
whatap_plugin_guide@com.ibm.ws.connectionpool.monitor.ConnectionPoolStats
```

## 플러그인 코드 작성 예시

대부분의 경우 오브젝트를 반환하므로 명시적 type casting 처리가 필요합니다. 다음은 플러그인 코드 작성 예시입니다.

### 설정된 메소드 시작 시간을 트레이스에 기록하기

- Java 에이전트 `hook_trace_helper_patterns` 옵션에 적용할 메소드를 `whatap.conf` 파일에 작성하세요.

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

```
hook_trace_helper_patterns=org.apache.catalina.connector.RequestFacade.*
```

- `TraceHelperStart.x` 파일에 플러그인 코드를 작성하세요.

```
$WHATAP_HOME/plugin/TraceHelperStart.x
```

```
// String prefix에 클래스명, 메소드명을 대입하세요.
String prefix = $point.class1 + "." + $point.method;
// 와탭 트레이스 속성 명칭을 `prefix` + "st" 로 현재 시간을 설정하세요.
$ctx.setAttribute(prefix + "st", new Long(System.currentTimeMillis()));
```

```
// 트레이스 정보에 현재 시간을 추가하세요.
ctx.profile(prefix + " Start", new java.util.Date().toString());
```

트레이스 내역에 메소드 시작시간을 다음과 같이 표시합니다.

```
io/home/test/TestData.getTestId Start
(Tue Aug 02 12:09:57 KST 2022)
```

## 설정된 메소드 종료시간과 수행시간을 트레이스에 기록하기

1. Java 에이전트 `hook_trace_helper_patterns` 옵션에 적용할 메소드를 `whatap.conf` 파일에 작성하세요.

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

```
hook_trace_helper_patterns=org.apache.catalina.connector.RequestFacade.*
```

2. `TraceHelperEnd.x` 파일에 플러그인 코드를 작성하세요.

```
$WHATAP_HOME/plugin/TraceHelperEnd.x
```

```
1 String prefix = $point.class1 + "." + $point.method;
2 // TraceHelperStart.x 파일에서 추가한 트레이스 속성을 가져오세요.
3 long st = ((Long) ctx.getAttribute(prefix + "st")).longValue();
4 long gap = System.currentTimeMillis() - st;
5 StringBuilder sb = new StringBuilder();
6 sb.append(new java.util.Date().toString() + " (Gap:" + gap + " milliseconds)");
7 // 트레이스 정보에 시작 시간과의 현재 시간과의 Gap(수행시간)을 추가하세요.
8 ctx.profile(prefix + " End", sb.toString());
```

트레이스 내역에 메소드 종료시간과 수행시간을 다음과 같이 표시합니다.



```
io/home/test/TestData.getTestId Start
(Tue Aug 02 12:09:57 KST 2022)
```

## 플러그인 API

플러그인 API는 공통 항목인 `$ctx`, `$point`, `$req`, `$res` 로 구분합니다. 트랜잭션 시작/종료시, HTTPC 구간, 특정 메소드 실행 구간 등에 적용할 수 있습니다.

**!** 플러그인 API를 사용해 개인식별정보 저장을 금지하며, 개인정보 수집 시 해당 내용을 개인정보처리방침에 명시해야 합니다.

### 파라미터 제공 - 공통 플러그인 파일

파라미터	메소드/변수	설명
\$ctx	속성관리	
	void setAttribute(String key, Object value)	Attribute를 설정합니다.
	Object getAttribute(String key)	Attribute의 값을 가져옵니다.
	트레이스 데이터	
	void profile(String desc)	트레이스 메시지 기록
	void profile(String name, String msg)	트레이스 Step명, 메시지 기록
	void profile(String name, String msg, int elapsed)	트레이스 Step명, 메시지, 수행시간 기록
	void profile(String name, String msg, int elapsed, int value)	트레이스 Step명, 메시지, 수행시간, 순서 기록

파라미터	메소드/변수	설명
	HTTP Request	
	String service()	서비스 이름 반환
	void service(String name)	서비스 이름 설정
	int serviceHash()	서비스이름의 Hash값 반환
	String remotep()	서비스 요청 IP 반환
	void remotep(String ip)	서비스 요청 IP 설정
	boolean isError()	에러 여부 반환
	void login(String id)	로그인 ID 설정
	String login()	로그인 ID 반환
	String httpMethod()	HTTP Method 반환
	String httpQuery()	HTTP Query 반환
	String httpContentType()	HTTP ContentType 반환
	String userAgent()	User-Agent 반환
	int status()	HTTP status 반환
\$point	String class1	클래스명
	String method	메소드명
	Object this1	Hooking 대상 클래스/메소드
	Object[] args	인자

파라미터	메소드/변수	설명
	Object return1	리턴

## 파라미터 미제공 - 공통 플러그인 파일

파라미터	메소드/변수	설명
없음	void log(Object c)	Logger를 통한 Logging
	void println(Object c)	System.out.println() 을 통한 출력
	Object field(Object o, String field)	Field 값 반환
	Object method(Object o, String method)	Invoke 메소드
	Object method(Object o, String method, String param)	Invoke 메소드
	String toString(Object o)	Object를 toString() 반환
	String toString(Object o, String def)	Object를 toString() 반환, null 인경우 def 반환
	int syshash(Object o)	hash 값 반환
	int syshash(HookArgs hook, int x)	x 번째 argument의 hash 값 반환
	int syshash(HookArgs hook)	argument의 hash 값 반환
	int cint(Object o)	int로 반환
	float cfloat(Object o)	float 으로 반환
	String cString(Object o)	String 으로 반환
	long clong(Object o)	long으로 반환

파라미터	메소드/변수	설명
	double cdouble(Object o)	double로 반환
	String desc(Object o)	Class signature 반환
	String toJson(Object o)	json으로 반환
	void shell(final String cmd, final String env)	shell 실행

## 데몬 및 배치

데몬, 배치와 같은 애플리케이션 서버를 모니터링할 때 시작점으로 설정한 메소드에 적용하는 API입니다. 다음 파일에 코드를 작성하세요.

- 서비스 시작 부분 적용: `$WHATAP_HOME/plugin/AppServiceStart.x`
- 서비스 종료 부분 적용: `$WHATAP_HOME/plugin/AppServiceEnd.x`

플러그인 파일	파라미터	메소드/변수	설명
AppServiceStart.x	\$ctx	공통 속성 참조	
	\$point	공통 속성 참조	
AppServiceEnd.x	\$ctx	공통 속성 참조	

## HTTP 서비스

`javax.servlet.http.HttpServlet` 클래스를 사용하는 일반적인 웹 애플리케이션에 적용할 수 있는 API입니다. 다음 파일에 코드를 작성하세요.

- 서비스 시작 부분 적용: `$WHATAP_HOME/plugin/HttpServiceStart.x`
- 서비스 종료 부분 적용: `$WHATAP_HOME/plugin/HttpServiceEnd.x`

플러그인 파일	파라미터	메소드/변수	설명
HttpServiceStart.x/ HttpServiceEnd.x	\$ctx	공통 속성 참조	
	\$req	String getCookie(String key)	Cookie 값 반환
		String getRequestURI()	RequestURI 반환
		String getRemoteAddr()	RemoteAddr 반환
		String getMethod()	HTTP Method 반환
		String getQueryString()	HTTP QueryString 반환
		String getParameter(String key)	Parameter 값 반환
		Object getAttribute(String key)	Attribute 값 반환
		String getHeader(String key)	Header 값 반환
		Enumeration getParameterNames()	getParameterNames 반환
		Enumeration getHeaderNames()	getHeaderNames 반환
		WrSession getSession()	Session Wrapper 반환
		Set getSessionNames()	getSessionNames 반환
		Object getSessionAttribute(String key)	getSessionAttribute 반환
		boolean isOk()	Plugin 상태 반환
		Throwable error()	Error 반환
	\$res	String getContentType()	ContentType 반환
String getCharacterEncoding()		CharacterEncoding 반환	

플러그인 파일	파라미터	메소드/변수	설명
		boolean isOk()	Plugin 상태 반환
		Throwable error()	Error 반환

## HTTP Outbound

`HttpClient` 와 같은 라이브러리를 이용해 HTTP Outbound Call 수행 할 때 적용할 수 있는 API입니다. 다음 파일에 코드를 작성하세요.

- 서비스 시작 부분 적용: `$_WHATAP_HOME/plugin/HttpCallStart.x`
- 서비스 종료 부분 적용: `$_WHATAP_HOME/plugin/HttpCallEnd.x`

플러그인 파일	파라미터	메소드/변수	설명
HttpCallStart.x	\$ctx	공통 속성 참조	
	\$req	String url()	URL 반환
		String host()	Hostname 반환
		int port()	Port 반환
		boolean isOk()	Plugin 상태 반환
		Throwable error()	Error 반환
HttpCallEnd.x	\$ctx	공통 속성 참조	
	\$res	String getContentType()	ContentType 반환
		String getCharacterEncoding()	CharacterEncoding 반환
		boolean isOk()	Plugin 상태 반환

플러그인 파일	파라미터	메소드/변수	설명
		Throwable error()	Error 반환

## 특정 메소드

자바 에이전트 옵션(`hook_trace_helper_*`)을 통해 설정한 메소드에 대해 적용하는 API 입니다.

플러그인 파일	파라미터	메소드/변수	설명
TraceHelperStart.x	\$ctx	공통 속성 참조	
	\$point	공통 속성 참조	
TraceHelperEnd.x	\$ctx	공통 속성 참조	
	\$point	공통 속성 참조	

## 사용자 정의 Pool

사용자 정의 Pool을 모니터링 하기 위한 API입니다. Pool 사용량 정보를 가진 메소드를 설정하면 와탭 모니터링 서비스에서 관찰하고 통계 정보를 확인할 수 있습니다.

플러그인 파일	파라미터	메소드/변수	설명	
CustomPool.x	\$id	custom_pool_classes에 설정한 id 값		
	\$pool	custom_pool_classes에 설정한 class		
	\$result	active(Object o)	Active Pool Count 설정	
		int idle(Object o)	Idle Pool Count 설정	

## 플러그인 적용 사례

다음은 실제 플러그인 적용 사례입니다.

### Elasticsearch 엔진의 검색 요청 모니터링

Elasticsearch(이하 ES) 엔진은 서블릿 엔진이 아니기 때문에 비정형 모니터링을 해야 합니다. 타상품이 지표 중심의 모니터링을 한다면 와탭의 플러그인을 활용해 ES의 요청 및 처리시간, 검색 키워드를 트레이싱할 수 있습니다.

whatap.conf

```
# 트랜잭션 EndPoint로 org.elasticsearch.search.SearchService.executeQueryPhase를 설정합니다.
hook_service_patterns=org.elasticsearch.search.SearchService.executeQueryPhase

# hook_trace_helper_start_patterns에 등록된 메소드가 실행될 때마다 TraceHelperStart.x 파일의 코드를 실행합니다.
hook_trace_helper_start_patterns=org.elasticsearch.search.query.QueryPhase.execute
```

\${WHATAP\_HOME}/plugin/TraceHelperStart.x

```
1  if($ctx.inner() == null) {
2    return;
3  }
4
5  try{
6    String tclass = "org.elasticsearch.search.query.QueryPhase";
7    String tmethod = "execute";
8
9    if(tclass.equals($point.class1) && tmethod.equals($point.method)) {
10     // 첫번째 argument의 query 메소드 호출 결과를 String query에 저장,
11     // org.elasticsearch.search.query.QueryPhase.execute(SearchContext searchContext)입니다.
12     // 즉, SearchContext.query() 메소드를 invoke 한 결과를 String query에 저장합니다.
13     String query = " " + method($point.args[0], "query");
14     // String query를 트레이스 정보로 출력
15     $ctx.profile(query);
16   }
17 } catch(Exception e) {
18   $ctx.profile(e.toString());
19 }
```



## 특정 IP로 유입되는 트랜잭션 수집 제외

Health Check, 내부 사용자들이 유발한 트랜잭션을 수집 제외하려면 플러그인을 활용할 수 있습니다.

```

${WHATAP_HOME}/plugin/HttpServiceStart.x

```

```

1 // ignoreIP 값을 수집 제외할 IP로 설정하세요.
2 String ignoreIP = "123.234.123.234";
3 String remoteIP = $req.getRemoteAddr();
4
5 if($ctx.ok() && remoteIP != null) {
6     if(remoteIP.equals(ignoreIP)) {
7         $ctx.ignore();
8     }
9 }

```

## Atomikos Pool 모니터링

[API 문서](#)를 참고해 적용하세요.

```

whatap.conf

```

```

custom_pool_classes=atomikos@com.atomikos.jdbc.internal.AbstractDataSourceBean

```

```

${WHATAP_HOME}/plugin/CustomPool.x

```

```

int total = cint(method($pool,"poolTotalSize"));
int avail = cint(method($pool,"poolAvailableSize"));

$result.active(total - avail);
$result.idle(avail);

```

## 트레이스에 쿠키값 표현

쿠키값을 트레이스에서 확인할 수 있습니다.

```

${WHATAP_HOME}/plugin/HttpServiceStart.x

```

```
1 // $req.getCookie() 파라미터로 확인하려는 쿠키를 설정하세요.  
2 String cookie = $req.getCookie("AWSALB");  
3  
4 if($ctx.ok()) {  
5     if(cookie != null) {  
6         $ctx.profile(cookie);  
7     }  
8 }
```

# 사용자 수

자바(Java) 웹 애플리케이션 서버에 연결된 사용자 수를 에이전트 설정을 통해 집계하는 방법을 안내합니다. 실시간 사용자 집계 활성화 여부 설정, 사용자 수를 집계하기 위한 쿠키 제한 설정, IP 주소 또는 HTTP 헤더 특정 값을 기준으로 한 사용자 집계 방법, 클라이언트 IP 정보 추적을 위한 HTTP 헤더 설정 등을 포함합니다.

- **trace\_user\_enabled** Boolean

기본값 `true`

실시간 사용자 집계 여부를 설정합니다. 사용자 추적 옵션이 중복 설정된 경우 동작 우선 순위는 다음과 같습니다.

1. `trace_user_using_ip` / `wclient_using_ip`
2. `trace_user_using_session`
3. `user_header_ticket` / `wclient_header_ticket`

ⓘ Java 에이전트 2.2.0 버전 이상의 경우 `wclient_using_ip`, `wclient_header_ticket` 옵션을 이용하세요.

- **trace\_user\_cookie\_limit** / **wclient\_cookie\_limit** Int

기본값 `2048`

사용자 집계를 위해 쿠키를 발행하는 경우 기존 쿠키가 너무 많다면 쿠키 오버플로어가 발생할 수 있습니다. 이를 방지하기 위해 limit을 설정합니다.

ⓘ Java 에이전트 2.2.0 버전 이상의 경우 `wclient_cookie_limit` 옵션을 이용하세요.

- **trace\_user\_using\_ip** / **wclient\_using\_ip** Boolean

기본값 `true`

실시간 사용자를 IP 주소 기반으로 집계합니다. IP 주소가 아닌 쿠키를 기반으로 사용자를 구분하고 싶으면 값을 `false` 로 변경하세요.

ⓘ Java 에이전트 2.2.0 버전 이상의 경우 `wclient_using_ip` 옵션을 이용하세요.

- **user\_header\_ticket** / **wclient\_header\_ticket** String

HTTP 헤더의 특정 값으로 사용자 수를 집계하고자 하는 경우 해당 Key 값을 설정합니다. 모바일 클라이언트인 경우 사용자 구분을

위해 header를 사용한다면 다음 옵션을 이용할 수 있습니다.

```
whatap.conf
```

```
user_header_ticket=login
```

! Java 에이전트 2.2.0 버전 이상의 경우 `wclient_header_ticket` 옵션을 이용하세요.

- **trace\_http\_client\_ip\_header\_key** *String*

클라이언트 IP(Remote IP)의 정보를 특정 HTTP 헤더의 값으로 변경해 설정하는 기능입니다. 프록시(Proxy) 환경에서 `X-Forwarded-For` 헤더 값을 클라이언트 IP로 설정할 수 있습니다.

WEB/WAS 앞에 L4와 같은 로드밸런서가 위치한 경우 클라이언트의 IP 주소가 아닌 L4의 IP 주소가 Remote Address가 되는 경우가 있습니다. 이 상황에서 실제 클라이언트 IP 정보가 http 헤더에 특정 키 값으로 기록되는 경우라면 해당 키 값으로 대체할 수 있습니다.

- **trace\_user\_using\_jsession** *Boolean*

기본값 `false`

실시간 사용자를 SESSIONID 기반으로 집계합니다.

# 부하량 제어

자바(Java) 애플리케이션 서버의 트래픽 스로틀링 옵션을 설정하는 에이전트 옵션을 안내합니다. 애플리케이션의 최대 동시 요청 수를 제한하고, 특정 사용자나 URL을 기준으로 서비스 접근을 제어하는 기능을 포함합니다. 사용자 정의 메시지 전송, URL 리다이렉션 설정, 이벤트 알림 활성화 등 다양한 옵션을 제공하여 과부하 상황에서 애플리케이션의 안정성을 보장할 수 있습니다.

- **throttle\_enabled** Boolean

기본값 `false`

애플리케이션의 최대 동시 처리 수를 제한하는 스로틀링 기능을 활성화합니다. `throttle_`으로 시작하는 모든 옵션은 `throttle_enabled=true` 상태에서만 동작합니다.

스로틀링 제어 관련 정책의 우선 순위는 다음과 같습니다.

1. Block: URL, 사용자 IP 주소 기준으로 서비스를 차단하며 가장 우선 적용합니다.
2. Passing: Passing에 적용된 URL들은 Reject 정책보다 우선합니다.
3. Reject: Block, Passing 정책 이후에 Reject 정책을 적용합니다.

- **throttle\_limit** Int

기본값 `10000`

에이전트 별 동시 처리하는 요청(트랜잭션) 수가 설정한 값을 초과하면 추가로 도달하는 요청은 reject 합니다.

- **throttle\_rejected\_message** String

기본값 `too many request!!`

스로틀링을 제한할 경우 사용자에게 전달할 메시지를 설정합니다.

- **throttle\_rejected\_forward** String

사용자 요청이 limit 값을 초과해 reject 될 때 사용자에게 전달하는 안내 페이지 URL을 설정합니다. `throttle_rejected_message` 를 동시 설정한 경우 `throttle_rejected_forward` 를 우선 적용합니다.

**!** 안내 페이지를 동일한 컨테이너에서 서비스한다면, 이 역시 동시 처리 수로 산정하므로 재귀 호출로 인한 장애 요소가 될 수 있습니다. 그러므로 안내 페이지는 static html 페이지로 만들거나 외부에 있어야 합니다.

- **reject\_event\_enabled** Boolean

기본값 `false`

사용자 요청을 Reject할 경우 이벤트 알림 발생 여부를 설정합니다.

- **reject\_event\_interval** *MiliSecond*

기본값 `30000`

Reject에 대한 이벤트 알림 발생 후 설정한 시간 동안 중복 이벤트에 대한 알림은 발생하지 않습니다.

- **throttle\_blocking\_url** *String*

`throttle_limit` 을 초과하지 않는 경우라도 블록킹(처리 거부)할 URL을 설정합니다. 시스템 장애를 유발하는 URL을 긴급하게 블록킹하기 위해 사용할 수 있습니다.

- **throttle\_blocking\_ip** *ip\_address*

사용자 IP 주소를 기준으로 블록킹하고자 할 때 설정합니다. 디도스 공격이나 잘못된 사용자를 IP 주소 기반으로 차단할 때 사용할 수 있습니다. 여러 개인 경우 쉼표(,)를 구분자로 사용하세요.

- **throttle\_target\_urls** *String*

등록한 URL을 대상으로 쓰로틀링 기능을 적용합니다. 여러 개인 경우 쉼표(,)를 구분자로 사용하세요.

- **throttle\_passing\_url** *String*

`throttle_limit` 을 초과하는 경우라도 처리해야 하는 URL을 설정합니다. 여러 개인 경우 쉼표(,)를 구분자로 사용하세요.

- **throttle\_passing\_url\_prefix** *String*

`throttle_limit` 을 초과하는 경우라도 처리해야 하는 URL들을 prefix로 설정합니다. 여러 개인 경우 쉼표(,)를 구분자로 사용하세요.

- **throttle\_blocked\_message** *String*

기본값 `request blocked!!`

요청을 블록킹한 사용자에게 전달할 메시지를 설정합니다.

- **throttle\_blocked\_forward** *String*

요청을 블록킹한 사용자에게 전달할 URL을 설정합니다. `throttle_blocked_message` 를 동시 설정한 경우

`throttle_blocked_forward` 를 우선 적용합니다.

# 에이전트 알림

자바(Java) 에이전트를 통해 애플리케이션 서버에서 발생하는 다양한 이벤트에 대한 알림 설정 방법을 제공합니다. 트랜잭션 재귀 호출, 서비스 거절, HTTPC 연결 오류, 힙 및 디스크 사용량 초과, CPU 사용량 임계치 도달, DB 커넥션 중복 할당 및 예외 발생 시 이벤트 알림을 설정하는 옵션을 포함합니다. 각 이벤트별로 발행 간격, 발행 여부, 임계치 설정 등 세밀한 조정을 할 수 있습니다.

- **recursive\_event\_interval** MiliSeconds

기본값 300000

트랜잭션의 재귀 호출에 대한 이벤트 알림 발행 간격을 설정합니다.

- **reject\_event\_enabled** Boolean

기본값 false

서비스 거절(호출 부하 제한/거절) 시 이벤트 알림 발행 여부를 설정합니다.

- **reject\_event\_interval** MiliSeconds

기본값 300000

서비스 거절(호출 부하 제한/거절) 시 이벤트 알림 발행 간격을 설정합니다.

- **httpc\_event\_enabled** Boolean

기본값 false

HTTPC 연결 오류 발생 시 이벤트 알림 발행 여부를 설정합니다.

- **httpc\_event\_interval** MiliSeconds

기본값 300000

HTTPC 연결 오류 발생 시 이벤트 알림 발행 간격을 설정합니다.

- **heap\_event\_enabled** Boolean

기본값 false

힙 사용량 임계 도달 시 이벤트 알림 발행 여부를 설정합니다.

- **heap\_event\_percent** Percentage

기본값 90

힙 사용량 이벤트 알림 발행 기준 임계치를 설정합니다.

- **heap\_event\_duration** MiliSeconds

기본값 30000

힙 사용량 이벤트 알림 발행 기준 지속 시간을 설정합니다.

- **heap\_event\_interval** MiliSeconds

기본값 300000

힙 사용량 이벤트 알림 발행 간격을 설정합니다.

- **heap\_event\_action** String

기본값 NONE

힙 사용량 이벤트 발생 시 실행할 동적 로딩 코드를 설정합니다.

! \$WHATAP\_HOME /plugin/ActionScript.x 파일에 작성한 Java 코드에 전달할 ID ( \$id 로 전달함)

- **disk\_event\_enabled** Boolean

기본값 false

디스크 사용량 임계치 도달 시 이벤트 알림 발행 여부를 설정합니다.

- **disk\_event\_percent** Percentage

기본값 90

디스크 사용량 이벤트 알림 발행 기준 임계치를 설정합니다.

- **disk\_event\_interval** MiliSeconds

기본값 300000

디스크 사용량 이벤트 알림 발행 간격을 설정합니다.

- **disk\_event\_action** String

디스크 사용량 이벤트 발생 시 실행할 동적 로딩 코드를 설정합니다.

! \$WHATAP\_HOME /plugin/ActionScript.x 파일에 작성한 Java 코드에 전달할 ID ( \$id 로 전달함)

- **cpu\_event\_enabled** Boolean

기본값 false

CPU 사용량 임계 도달 시 이벤트 알림 발행 여부를 설정합니다.

- **cpu\_event\_percent** Percentage



기본값

CPU 사용량 이벤트 알림 발행 기준 임계치를 설정합니다.

- **cpu\_event\_duration** MiliSeconds

기본값

CPU 사용량 이벤트 알림 발행 기준 지속시간을 설정합니다.

- **cpu\_event\_interval** MiliSeconds

기본값

CPU 사용량 이벤트 알림 발행 간격을 설정합니다.

- **cpu\_event\_action** String

CPU 사용량 이벤트 발생 시, 실행할 동적 로딩 코드에 전달할 ID를 설정합니다.

❗  /plugin/ActionScript.x 파일에 작성한 Java 코드에 전달할 ID (  로 전달함)

- **dbc\_dup\_event\_enabled** Boolean

기본값

DB Connection이 중복 할당될 경우 이벤트 알림 발행 여부를 설정합니다.

- **dbc\_dup\_event\_fullstack\_enabled** Boolean

기본값

DB Connection이 중복 할당될 때 Stack 확보 여부를 설정합니다.

- **exception\_event\_enabled** Boolean

기본값

Exception 발생 시 이벤트 알림 발행 여부를 설정합니다.

- **exception\_event\_interval** MiliSeconds

기본값

Exception 발생 시 이벤트 알림 발행 간격을 설정합니다.

- **exception\_event\_set** String

대상 Exception을 설정합니다. 여러 개인 경우 쉼표(,)를 구분자로 사용하세요.

- **exception\_event\_action** String

이벤트 발생 시 실행할 동적 로딩 코드를 설정합니다.

! \$WHATAP\_HOME /plugin/ActionScript.x 파일에 작성한 Java 코드에 전달할 ID ( \$id 로 전달함)

# Apdex

사용자 만족도를 측정하는 Apdex 점수를 설정하는 자바(Java) 에이전트 옵션을 안내합니다. Apdex 지표를 통해 애플리케이션의 성능을 객관적으로 평가하고 사용자 경험을 개선할 수 있습니다.

**whatap.apdex\_time** millisecond

기본값 1200

만족 S 기본값은 에이전트 설정 메뉴에서 변경할 수 있습니다.

[Application Performance Index\(Apdex\)](#)는 애플리케이션 성능 지표입니다. 웹 애플리케이션의 고객 만족도를 측정합니다. 사용자 만족도에 대한 지표로 활용할 수 있으며, 0 ~ 1 사이의 값을 갖습니다.

$(\text{만족 횟수} + (\text{허용 횟수} * 0.5)) / \text{전체 요청 수}$

- 만족 (Satisfied, S): 업무처리에 전혀 문제 없음  $\leq 1.2$ 초 (만족 S 기본값)
- 허용 (Tolerating, T): 사용자가 지연을 느끼나 업무처리는 가능  $\leq 4.8$ 초 (만족 S \* 4)
- 불만 (Frustrated, F): 업무처리가 불가능  $> 4.8$ 초 (허용 T 초과 및 오류)

ⓘ 에이전트 2.0 버전 이상부터 지원합니다.

# 통계

자바(Java) 애플리케이션 모니터링을 위한 다양한 통계 수집 기능 관련 에이전트 옵션을 안내합니다. 성능 카운터 확장, 도메인별 트랜잭션 수집, 멀티 서버 트랜잭션 의존성 분석, 로그인 유형별 및 Referer 별 통계 수집 등을 포함합니다. 또한 SQL, HTTP Call, 오류 통계와 사용자 에이전트 정보 수집에 대한 최대 레코드 수 제한 설정도 확인할 수 있습니다.

- **addin\_monitor\_num** *Int*

기본값 `0`

성능 카운터 확장을 위해 추가 선언할 클래스 수를 설정합니다. `addin_monitor_num` 에 설정한 값만큼 `addin_monitor_x` 를 선언합니다.

- **addin\_monitor\_0** *Int*

기본값 `0`

```
whatap.conf
```

```
addin_monitor_num=2
addin_monitor_0=a.a.Class1
addin_monitor_1=a.a.Class2
```

- **stat\_domain\_enabled** *Boolean*

기본값 `false`

클라이언트의 접속 도메인별 트랜잭션 통계 수집 기능을 활성화합니다.

- **stat\_domain\_max\_count** *Int*

기본값 `7000`

하나의 JVM에서 5분 동안 수집할 도메인별 트랜잭션 통계의 최대 레코드 수입니다.

- **stat\_mtrace\_enabled** *Boolean*

기본값 `false`

멀티 서버 트랜잭션에서 Caller & Callee 간에 버전 별 의존성 통계를 수집합니다.

- **mtrace\_spec** *String*

기본값 `v1`

현재 인스턴스의 애플리케이션 버전을 설정합니다. 임의의 문자열을 설정할 수 있습니다. 이 데이터는 호출 통계를 위해 사용됩니다.

- **stat\_mtrace\_max\_count** `Int`

기본값 `7000`

멀티 서버 트랜잭션에서 Caller & Callee 간에 버전별 의존성 통계의 최대 레코드 수입니다.

- **stat\_login\_enabled** `Boolean`

기본값 `false`

로그인별 트랜잭션 통계를 수집합니다.

- **stat\_login\_max\_count** `Int`

기본값 `7000`

하나의 JVM에서 5분 동안 수집할 로그인 별 트랜잭션 통계의 최대 레코드 수입니다.

- **stat\_referer\_enabled** `Boolean`

기본값 `false`

Referer별 트랜잭션 통계를 수집합니다.

- **stat\_referer\_max\_count** `Int`

기본값 `7000`

하나의 JVM에서 5분 동안 수집할 Referer 별 트랜잭션 통계의 최대 레코드 수입니다.

- **stat\_tx\_max\_count** `Int`

기본값 `5000`

트랜잭션 통계 정보의 개수를 제한합니다. 5분 동안 수집해 서버에 전송하는 통계 정보의 최대 레코드 수를 제한합니다.

- **stat\_sql\_max\_count** `Int`

기본값 `5000`

SQL 통계 정보의 개수를 제한합니다. 5분 동안 수집해 서버에 전송하는 통계 정보의 최대 레코드 수를 제한합니다.

- **stat\_httpc\_max\_count** `Int`

기본값 `5000`

Http Call 통계 정보의 개수를 제한합니다. 5분 동안 수집해 서버에 전송하는 통계 정보의 최대 레코드 수를 제한합니다.

- **stat\_error\_max\_count** `Int`

기본값 1000

Error 통계 정보의 개수를 제한합니다. 5분 동안 수집해 서버에 전송하는 통계 정보의 최대 레코드 수를 제한합니다.

- **stat\_useragent\_max\_count** `int`

기본값 500

User Agent 통계 정보의 개수를 제한합니다. 5분 동안 수집해 서버에 전송하는 통계 정보의 최대 레코드 수를 제한합니다.

# 토폴로지 맵

자바(Java) 에이전트가 수집한 데이터를 사용하여 애플리케이션의 토폴로지 맵을 생성하는 에이전트 설정 옵션을 안내합니다. 트랜잭션 호출자, 데이터베이스 연결 정보, HTTPC 아웃바운드 정보, 액티브 트랜잭션 상태 등의 다양한 지표를 통해 시스템의 토폴로지를 시각화하고 분석할 수 있습니다.

- **tx\_caller\_meter\_enabled** Boolean

기본값 `false`

트랜잭션 정보로 토폴로지 통계를 생성합니다. `mtrace_enabled` 옵션의 값이 `true` 이면 동작합니다.

- **sql\_dbc\_meter\_enabled** Boolean

기본값 `false`

데이터 베이스 연결 정보로 토폴로지 통계를 생성합니다.

- **httpc\_host\_meter\_enabled** Boolean

기본값 `false`

토폴로지 맵에서 httpc outbound 정보를 표현합니다.

- **actx\_meter\_enabled** Boolean

기본값 `false`

토폴로지 맵에서 액티브 트랜잭션 상태를 표현합니다.

# 트랜잭션 에러 스택

자바(Java) 애플리케이션에서 트랜잭션 중 발생하는 에러를 감지하고 이에 대한 스택 트레이스를 표시하는 에이전트 설정 옵션을 안내합니다. 트랜잭션, 메소드, SQL, HTTP 호출, SOCKET 호출 등 다양한 영역에서 에러 정보를 수집하고, 에러 스택의 길이, 제목 길이 등을 설정하는 방법을 포함합니다. 에이전트 설정을 통해 에러 관리를 개선하고, 애플리케이션의 오류 진단을 보다 효율적으로 수행할 수 있도록 지원합니다.

호출 서비스의 `status_code`와는 무관하며, 여러 개의 에러 발생 시 에러 스택은 첫번째 에러 스택만 표시합니다.

❗ 이 문서에서 안내하는 에이전트 설정은 **Java 에이전트 2.2.15** 버전 이상에서 지원합니다.

## 에이전트 설정

### 전체 설정

whatap.conf

```
# 에러 스택 표시 여부
trace_exception_stack=false

# 에러 스택 길이의 기본값
trace_exception_depth=50

# 에러 스택 메시지 제목의 길이
trace_exception_title_len=128
```

### 메소드

whatap.conf

```
# 메소드 에러 스택 표시 여부
# trace_exception_stack 값을 따라가며 별도로 true/false 값으로 기능을 켜거나 끌 수 있습니다.
trace_method_error_stack=trace_exception_stack
```



```
# 메소드 에러 스택 길이
# trace_exception_depth 값을 따라가며 별도로 설정할 수 있습니다.
trace_method_error_depth=trace_exception_depth
```

## SQL

```
whatap.conf
```

```
# SQL 에러 스택 표시 여부
# trace_exception_stack 값을 따라가며 별도로 true/false 값으로 기능을 켜거나 끌 수 있습니다.
trace_sql_error_stack=trace_exception_stack

# sql 에러 스택 길이
# trace_exception_depth 값을 따라가며 별도로 설정할 수 있습니다.
trace_sql_error_depth=trace_exception_depth
```

## HTTP Call

```
whatap.conf
```

```
# http call 에러 스택 표시 여부
# trace_exception_stack 값을 따라가며 별도로 true/false 값으로 기능을 켜거나 끌 수 있습니다.
trace_httpc_error_stack=trace_exception_stack

# http call 에러 스택 길이
# trace_exception_depth 값을 따라가며 별도로 설정할 수 있습니다.
trace_httpc_error_depth=trace_exception_depth
```

## SOCKET

```
whatap.conf
```

```
# socket 에러 스택 표시 여부
# trace_exception_stack 값을 따라가며 별도로 true/false 값으로 기능을 켜거나 끌 수 있습니다.
trace_socket_error_stack=trace_exception_stack
```

```
# socket 에러 스택 길이  
# trace_exception_depth 값을 따라가며 별도로 설정할 수 있습니다.  
trace_socket_error_depth=trace_exception_depth
```

## 트랜잭션

```
whatap.conf
```

```
# 트랜잭션 에러 스택 표시 여부  
# trace_exception_stack 값을 따라가며 별도로 true/false값으로 기능을 켜거나 끌 수 있습니다.  
trace_endtx_error_stack=trace_exception_stack  
  
# 트랜잭션 에러 스택 길이  
# trace_exception_depth 값을 따라가며 별도로 설정할 수 있습니다.  
trace_endtx_error_depth=trace_exception_depth
```

# 부가 기능

자바(Java) 애플리케이션의 모니터링 수준을 사용자의 필요에 맞게 조정할 수 있는 추가적인 에이전트 설정 옵션을 안내합니다. SQL 파라미터 수집, HTTP 파라미터 및 헤더 정보 수집, 사용자 IP 정보 추출, 사용자 수 추적 방법 선택, SAP Function 추적 등을 포함합니다. 이 옵션들을 통해 보다 세밀한 데이터 수집 및 분석이 가능하며, 특히 보안이 중요한 환경에서 필요에 따라 옵션을 조정할 수 있습니다.

다음은 자주 사용하지만 보안 등의 이유로 에이전트 기본 설정에서 제외한 옵션입니다. 필요하다면 아래 옵션을 추가해 적용할 수 있습니다.

- **profile\_sql\_param\_enabled** Boolean

기본값 `true`

SQL 파라미터를 수집합니다.

- **profile\_http\_parameter\_enabled** Boolean

기본값 `true`

HTTP 파라미터를 수집합니다.

- **profile\_http\_header\_enabled** Boolean

기본값 `false`

HTTP 헤더 정보를 수집합니다.

- **trace\_http\_client\_ip\_header\_key** String

기본값 `X-Forwarded-For`

사용자 IP 정보를 HTTP 헤더의 키에서 추출합니다.

- **trace\_user\_method / wclient\_cookie\_limit** Boolean

기본값 `cookie`

사용자 수 추적을 쿠키 기반으로 합니다.

ⓘ Java 에이전트 2.2.0 버전 이상의 경우 `wclient_trace_type` 옵션을 이용하세요.

- **hook\_httpc\_sap3\_enabled** Boolean

기본값 `true`

SAP JCO를 사용하는 경우 SAP Function을 추적합니다.

# 비동기 추적

자바(Java) 에이전트 설정을 통해 비동기 애플리케이션 활동을 추적하는 방법을 안내합니다. `CompletableFuture` 메소드 수집 방법과 사용자 애플리케이션에 해당 메소드를 hooking하는 방식을 포함합니다. 설정 예시와 함께 [whatap.conf](https://whatap.conf) 파일에 추가해야 할 옵션을 제공하여 비동기 호출의 성능 모니터링을 강화할 수 있도록 지원합니다.

## CompletableFuture

Java 에이전트 설정을 통해 CompletableFuture 메소드를 수집할 수 있는 방법을 안내합니다.

### CompletableFuture 메서드 추가

사용자의 Java 애플리케이션 프로젝트에 에이전트로 hooking할 CompletableFuture 메소드를 추가하세요.

- 메서드 이름: `trace` 로 통일
- ReturnType: `Supplier`, `Consumer`, `Runnable`, `Future`

```
package io.home.test.util;

import java.util.concurrent.Future;
import java.util.function.Consumer;
import java.util.function.Supplier;

public class W {
    public static <T> Supplier<T> trace(Supplier<T> f) {
        return f;
    }
    public static <T> Consumer<T> trace(Consumer<T> f) {
        return f;
    }
    public static <T> Runnable trace(Runnable f) {
        return f;
    }
    public static <T> Future<T> trace(Future<T> f) {
        return f;
    }
}
```

## 에이전트 설정 추가

에이전트 설정을 위해 [whatap.conf](#) 파일에 다음 옵션을 추가하세요.

```
whatap.conf
```

```
hook_completablefuture_patterns=io.home.test.util.W.*
```

- ⓘ • Context가 있는 경우 `trace` 메소드를 추가해 트랜잭션을 연결하세요.
- Context가 없는 경우 `hook_service_patterns` 옵션으로 서비스를 시작해야 합니다.

## 사용 예시

### • 원본

```
1 public CompletableFuture<String> serviceATimeout() {
2     return CompletableFuture.supplyAsync(() -> {
3         RestTemplate restTemplate = new RestTemplate();
4         return restTemplate.getForObject("http://localhost:8081/api/serviceB/timeout", String.class);
5     });
6 }
```

### • `io.home.test.util.W.trace()` 적용

```
1 // io.home.test.util.W.trace() 적용
2 public CompletableFuture<String> serviceATimeout() {
3     return CompletableFuture.supplyAsync(W.trace(() -> {
4         RestTemplate restTemplate = new RestTemplate();
5         return restTemplate.getForObject("http://localhost:8081/api/serviceB/timeout", String.class);
6     }));
7 }
```

# 오픈소스 추적

자바(Java) 애플리케이션에서 사용하는 프레임워크나 오픈소스 라이브러리를 에이전트를 통해 추적하는 설정 방법을 제공합니다. 이는 Java 에이전트 설정 파일([whatap.conf](#)) 파일에 `weaving` 옵션을 추가하여 설정하며, 다양한 프레임워크 및 라이브러리 버전에 대응하는 방법을 안내합니다.

```

${WHATAP_HOME}/whatap.conf

```

```

weaving=

```

Java 에이전트를 통해 추적하고 있는 프레임워크 또는 오픈소스에 대한 설정 방법은 다음을 참조하세요.

프레임워크/ 오픈소스	버전	설정	에이전트 최소 버전	비고
apache-camel-cxf	camel-cxf-3.15 이상	weaving=camel-cxf-3.15	<a href="#">v2.2.27</a>	-
apache-camel-seda	camel-seda-2.22 이상	weaving=camel-seda-2.22	<a href="#">v2.2.20</a>	-
	camel-seda-3.2 이상	weaving=camel-seda-3.2	<a href="#">v2.2.20</a>	-
db2	db2-11.5 이상	weaving=db2-11.5	<a href="#">v2.2.18</a>	-
feign-client	feign-11 이상	weaving=feign-11	<a href="#">v2.2.6</a>	-
hystrix	hystrix-1.5 이상	weaving=hystrix-1.5	<a href="#">v2.0_21</a>	-
kafka	kafka-clients-2.4.0 이상	weaving=kafka-clients-2.4.0	<a href="#">v2.2.15</a>	-
	reactor-kafka-1.3 이상	weaving=reactor-kafka-1.3	<a href="#">v2.2.5</a>	-
redis(jedis)	jedis-2.9.3 이상	weaving=jedis-2.9.3	<a href="#">v2.0_33</a>	-
	jedis-3.2 이상	weaving=jedis-3.2	<a href="#">v2.0_09</a>	-

프레임워크/ 오픈소스	버전	설정	에이전트 최소 버전	비고
redis(lettuce)	lettuce-5.1 이상	weaving=lettuce-5.1	<a href="#">v2.2.7</a>	-
	lettuce-6.2 이상	weaving=lettuce-6.2	<a href="#">v2.2.16</a>	-
mongodb, mongodb- reactive	mongodb-3.8.2 이상	weaving=mongodb-3.8.2	<a href="#">v2.2.11</a>	-
	mongodb-4.0.3 이상	weaving=mongodb-4.0.3	<a href="#">v2.2.11</a>	-
	mongodb-4.4 이상	weaving=mongodb-4.4	<a href="#">v2.2.11</a>	-
	mongodb-4.8 이상	weaving=mongodb-4.8	<a href="#">v2.2.11</a>	-
mule framework	mule-3.9.5 이상	weaving=mule-3.9.5	<a href="#">v2.2.23</a>	-
	mule-4.5 이상	weaving=mule-4.5	<a href="#">v2.2.23</a>	-
okhttp	okhttp-2.7 이상	weaving=okhttp-2.7	<a href="#">v2.0.15</a>	-
	okhttp3 이상	weaving=okhttp3	<a href="#">v2.0.15</a>	-
	okhttp3-4.4	weaving=okhttp3-4.4	<a href="#">v2.2.9</a>	-
quarkus, quarkus- reactive	quarkus-reactive-1.13 이상	weaving=quarkus- reactive-1.13	<a href="#">v2.2.19</a>	-
	quarkus-reactive-2.10 이상	weaving=quarkus- reactive-2.10	<a href="#">v2.2.19</a>	-
rabbitmq	reactor-rabbitmq-1.2 이상	weaving=reactor- rabbitmq-1.2	<a href="#">v2.0.06</a>	-
retrofit	retrofit2-2.5 이상	weaving=retrofit-2.5	-	(배포 예정)

프레임워크/ 오픈소스	버전	설정	에이전트 최소 버전	비고
ribbon	ribbon	weaving=ribbon	<a href="#">v2.2.10</a>	-
spring-boot	spring-boot-2.1 이상	weaving=spring-boot-2.1	<a href="#">v2.2.23</a>	kafka-clients, r2dbc-mysql, spring-cloud- gateway, spring- webflux, tomcat9, undertow 포함
	spring-boot-2.5 이상	weaving=spring-boot-2.5	<a href="#">v2.2.9</a>	kafka-clients, r2dbc-mysql, redis(lettuce), spring-cloud- gateway, spring- webflux, tomcat9, undertow 포함
	spring-boot-2.7 이상	weaving=spring-boot-2.7	<a href="#">v2.2.9</a>	jasync-r2dbc- mysql, kafka- clients, r2dbc- mysql, redis(lettuce), spring-cloud- gateway, spring- webflux, tomcat9, undertow 포함
	spring-boot-3.0 이상	weaving=spring-boot-3.0	<a href="#">v2.2.9</a>	jasync-r2dbc- mysql, kafka- clients, r2dbc- mysql, redis(lettuce), spring-cloud- gateway, spring- webflux,



프레임워크/ 오픈소스	버전	설정	에이전트 최소 버전	비고
				tomcat10, undertow 포함
tomcat	tomcat9	weaving=tomcat9	<a href="#">v2.2.5</a>	-
	tomcat10	weaving=tomcat10	<a href="#">v2.2.5</a>	-
undertow	undertow-2.3 이상	weaving=undertow-2.3	<a href="#">v2.2.14</a>	-

프레임워크나 오픈소스로 spring-boot-3.x, feign-client-11, okhttp3-4.4 사용 시 다음과 같이 옵션을 설정하세요.

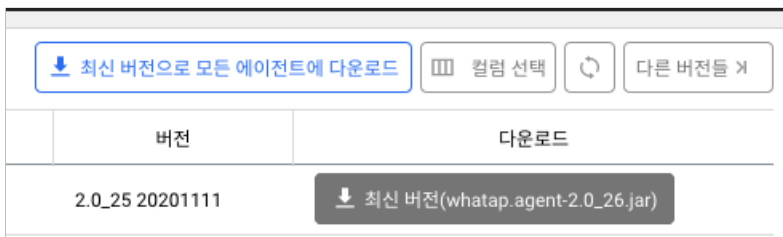
```
whatap.conf
```

```
weaving=spring-boot-3.0,feign-11,okhttp3-4.4
```

# 에이전트 업데이트

## 최신 버전 에이전트 다운로드

1. [와탭 모니터링 서비스](#)로 이동해 로그인하세요.
2. 에이전트를 업데이트할 프로젝트를 선택하세요.
3. 왼쪽 메뉴에서 [관리](#) > [에이전트 업데이트](#)를 선택하세요.
4. [최신 버전](#) 또는 [최신 버전으로 모든 에이전트에 다운로드](#) 버튼을 클릭하세요.



에이전트를 다운로드한 다음 애플리케이션 서버를 다시 시작하세요. 업데이트를 자동 적용합니다.

## 에이전트 업데이트 자동 설정

자바 에이전트를 업데이트하려면 서버를 재시작해야 합니다. 재시작 이전에 [whatap.agent.jar](#) 파일이 변경되면 애플리케이션 서버가 재시작되는 과정에서 안정적인 시스템 종료(shutdown)를 실패할 수 있습니다.

원칙적으로 자바 에이전트를 업데이트 하기 위해서는 다음의 순서로 진행해야 합니다.

1. app shutdown
2. agent jar 변경
3. app start

위 과정을 보다 쉽게 처리하기 위해 와탭의 에이전트 jar 파일명에는 버전 번호를 포함합니다. 업데이트 과정에서 최신 버전을 자동 적용하도록 설정하는 것이 중요합니다.

```

1 ##### WHATAP #####
2 WHATAP_HOME=/whatap
3 WHATAP_JAR=`ls ${WHATAP_HOME}/whatap.agent-*.jar | sort -V | tail -1`

```

```
4 JAVA_OPTS="${JAVA_OPTS} -javaagent:${WHATAP_JAR}"  
5 ##### WHATAP #####
```

# 에이전트 삭제 및 롤백

## 에이전트 삭제

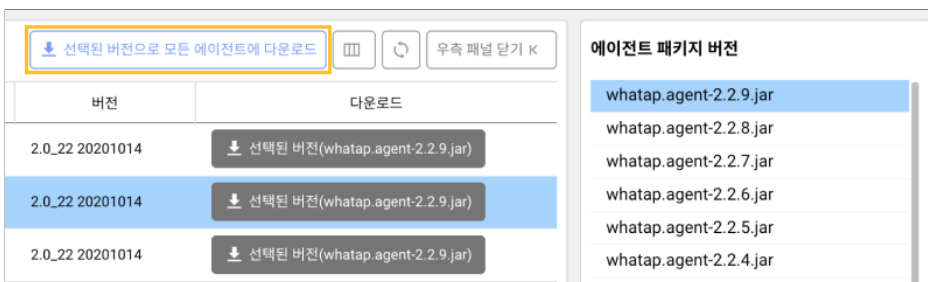
애플리케이션 서버에서 JVM 옵션의 `-javaagent` 설정을 삭제한 다음 서버를 다시 시작하세요. 이후 `$WHATAP_HOME` 환경 변수를 삭제하세요.

## 에이전트 롤백

1. [와탭 모니터링 서비스](#)에서 롤백을 원하는 프로젝트를 선택하세요.
2. 왼쪽 메뉴에서 [관리](#) > [에이전트 업데이트](#)를 선택하세요.
3. [다른 버전들](#) 버튼을 선택하세요.



4. 롤백을 원하는 버전을 선택한 다음 [선택된 버전](#) 또는 [선택된 버전으로 모든 에이전트에 다운로드](#) 버튼을 클릭하세요.



5. 에이전트를 설치할 때 수정한 스크립트에서 `WHATAP_JAR` 환경 변수를 롤백하려는 버전으로 변경하세요.

```
##### WHATAP START #####
WHATAP_HOME=/Users/gyuseokjung/javatest/whatap
WHATAP_JAR=`ls ${WHATAP_HOME}/whatap.agent-2.0_25.jar`
JAVA_OPTS="${JAVA_OPTS} -javaagent:${WHATAP_JAR} "
##### WHATAP END #####
```

# MSA 업무 그룹 구분

Auto Scale Out 환경에서 자동 등록된 에이전트의 IP / Port / PID 정보만을 가지고 업무 단위의 정보를 인지하기 어려운 경우가 있습니다. 이 경우 `okind` 속성을 JVM 옵션으로 부여해 쉽게 식별할 수 있습니다.

다음은 전형적인 Docker 환경에 대한 설명입니다.

- 각 에이전트는 잘게 쪼개어 놓은 MSA (Micro Service Architecture) 구성 요소입니다.
- 에이전트들은 자동으로 Scale In/Out 됩니다.
- 어떤 IP를 가진 서버에서 어떤 PID로 애플리케이션이 시작할지 예측할 수 없습니다.

위와 같은 Docker 환경에서 에이전트 이름과 더불어 업무를 식별할 수 있는 JVM 옵션으로 `whatap.okind` 를 추가해 `okind` 속성을 추가할 수 있습니다.

```
JAVA_OPTS
```

```
-Dwhatap.okind=mobile_ui
```

`whatap.conf` 파일에 설정을 추가할 수도 있습니다.

```
whatap.conf
```

```
whatap.okind=common_api
```

- ✔ MSA 환경에서 와탭이 제공하는 다양한 기능을 활용하려면 `okind` 속성을 추가하세요.

# AES 256 암호화

와탭 애플리케이션 에이전트는 수집한 데이터를 암호화해 서버로 전송합니다. 이를 위해 XOR(Exclusive-OR) 연산과 고급 암호화 표준(Advanced Encryption Standard, 이하 AES) 암호화를 사용하며, 평문을 128비트 단위로 나누어 암복호화를 수행합니다. 256비트까지 확장해 보안을 강화할 수 있습니다.

기본적으로 Java Cryptography Extension(이하 JCE)은 128비트를 지원합니다. AES 256비트를 적용하려면 JCE를 업데이트해야 합니다.

기본 환경에서 AES 256 적용 시 다음과 같이 오류가 발생합니다.

```
Unsupported keysize or algorithm parameters.
```

```
##혹은,
```

```
Illegal key size or default parameters.
```

1. 다음 링크에서 버전에 맞는 파일을 다운로드하세요.

## ❗ 자바 버전 별 JCE 다운로드 링크

- [JAVA 8](#)
- [JAVA 7](#)
- [JAVA 6](#)
- [JAVA 5](#)
- [JAVA 1.42](#)

2. ``${JAVA_HOME}/jre/lib/security` 경로에 파일을 덮어쓰기하세요.
3. JCE를 적용했다면 ``${WHATAP_HOME}/whatap.conf` 파일에 다음 설정을 추가하세요.

```
whatap.conf
```

```
cypher_level=256
```

4. 애플리케이션 서버를 다시 시작하면 ASE 256 암호화를 적용합니다.

# Scale In에서 에이전트 자동 삭제

Auto Scale 환경에서 에이전트 이름을 자동 부여하면 Scale Out이 발생할 경우 운영자 개입이 필요 없습니다. 시스템을 자동으로 확장합니다. 반면 Scale In이 발생하면 시스템이 정상적인 Shut down으로 인식해야 합니다.

에이전트 자동 삭제를 위해 와탭 서버는 자바 에이전트로부터 메시지를 받아야 합니다. 에이전트는 `${WHATAP_HOME}` 에 Java 프로세스의 `whatap_{java-process-pid}.shutdown` 파일이 나타나면 **SILENT\_SHUTDOWN** 이벤트를 와탭 서버에 전송합니다. 전송은 5초 이내로 완료합니다.

**SILENT\_SHUTDOWN** 이벤트를 와탭 서버에 전송하고 1분 이내에 에이전트가 셧다운 되면 와탭 서버는 목록에서 해당 Java 에이전트를 바로 제거합니다. inactive 이벤트도 발생시키지 않습니다.

ⓘ Tomcat의 경우 `shutdown.sh` 파일에 설정할 수 있습니다.

```
$TOMCAT_HOME/bin/shutdown.sh
```

```
touch $WHATAP_HOME/whatap_{java-process-pid}.shutdown
```

# 주요 메뉴 알아보기

와탭 에이전트 설치를 완료했다면 [와탭 모니터링 서비스](#)에 로그인하세요. 와탭 모니터링 서비스는 원활한 서비스 제공과 안정적 운영, 이상 징후 감시 및 파악을 위한 정보를 제공합니다.

애플리케이션 모니터링은 실시간으로 트랜잭션을 모니터링하고 특정 시점을 설정한 분석 기능 및 운영의 편리성을 높이기 위한 리포트 기능, 알림 수단을 제공합니다.

와탭 모니터링 서비스는 다음의 주요 기능을 제공합니다.

- **실시간 모니터링**

실시간 트랜잭션 현황 / 응답시간 분포도(Hitmap) / 서버 현황 / 실시간 사용자(Realtime User) 현황 / 트랜잭션 현황 / 리소스 현황

- **트랜잭션 분석**

트랜잭션 트레이싱 / 스택 분석 / 멀티 트랜잭션 트레이스(연계 추적) / 응답시간 분포도(Hitmap) / SQL 추적 / HTTP Call 추적

- **사후 분석**

Cube 분석 / 스택 분석 / Performance Trend 분석 / 응답시간 분포도(Hitmap) / 에러 통계

- **통계**

Transaction / Error / SQL / Remote HTTP Call / Client IP / 브라우저 통계

- **관리**

Agent 실행 방식 / 개발 및 운영 환경 분리 시 지원 / 알람 설정 / 알림 전달 수단 설정

- **리포트**

Daily Report / Daily Application Check / Weekly Report / Monthly Report



# 대시보드

와탭 에이전트 설치를 완료했다면 [와탭 모니터링 서비스](#)에 로그인하세요. 초기 화면은 [프로젝트 목록](#)입니다. [프로젝트 목록](#)에서 에이전트를 설치한 애플리케이션 서버를 확인할 수 있습니다. 에이전트를 설치한 하나의 애플리케이션 서버는 프로젝트이며, 모니터링 단위입니다. 프로젝트 목록의 프로젝트를 선택하면 [애플리케이션 대시보드](#)로 진입할 수 있습니다.

## 대시보드란?

와탭 모니터링 서비스에서 대시보드는 프로젝트의 전체 현황을 한눈에 파악할 수 있는 기능입니다. 모니터링 중인 전체 자원 규모를 확인할 수 있고 실시간 대시보드에서 모니터링 대상 자원을 필터링할 수 있습니다. 진행 중인 트랜잭션과 종료된 트랜잭션 정보를 실시간으로 업데이트하고 시각화된 차트를 제공합니다. 운영자는 서비스 및 시스템의 정확한 현재 상태를 직관적으로 파악할 수 있습니다. 또한 즉각적인 장애 인지 및 잠재적 문제 요소를 빠르게 식별할 수 있습니다.

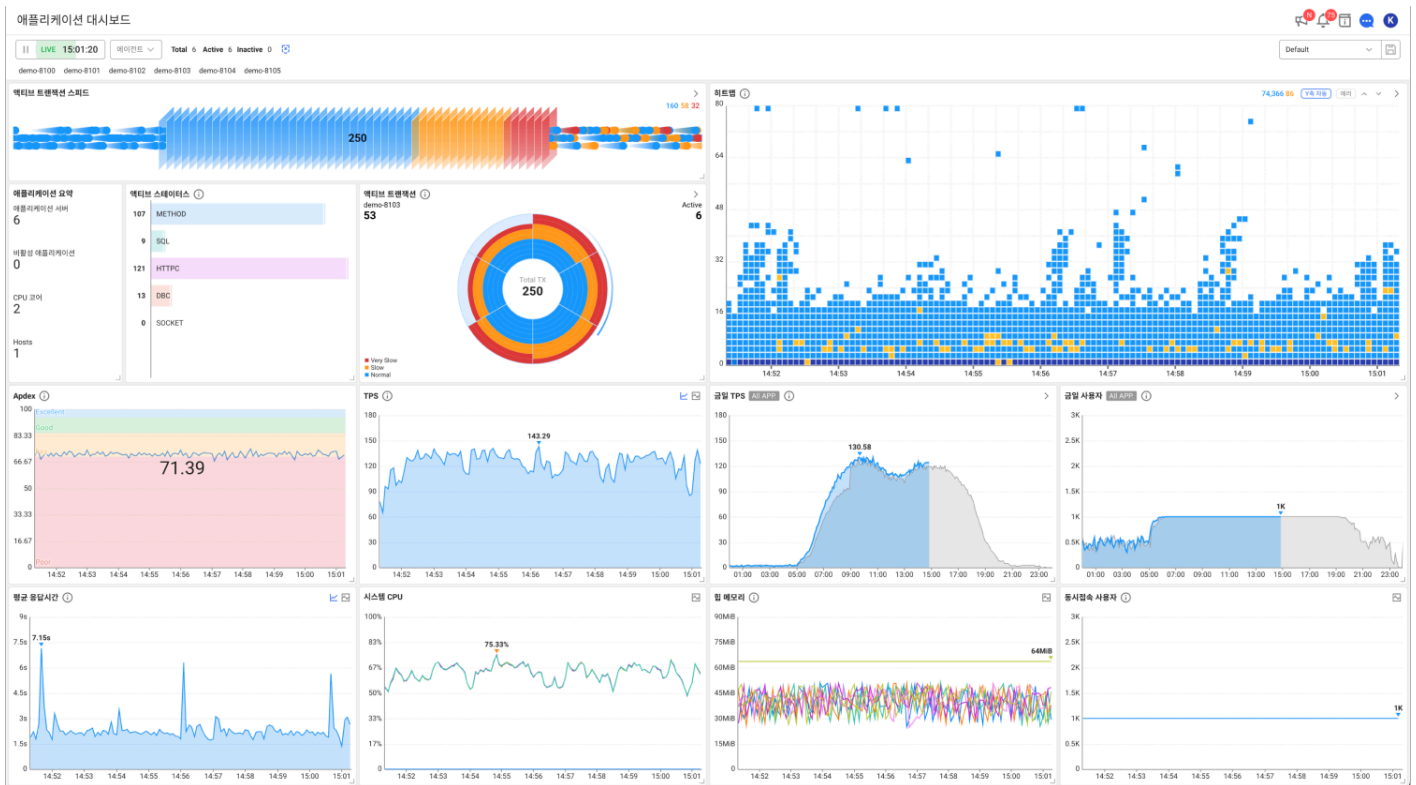
대시보드를 통해 파악할 수 있는 정보는 다음과 같습니다.

- 응답시간 분포 차트를 통해 응답시간 범위를 조정하거나 에러 데이터를 필터링할 수 있습니다.
- 진행 중 트랜잭션을 분석하고 종료한 트랜잭션을 실시간으로 업데이트합니다.
- TPS, 응답시간, CPU, Memory, 실시간 사용 등의 리소스 현황을 파악할 수 있습니다.
- 최신의 데이터를 자동 업데이트해 실시간 모니터링 서비스를 제공합니다.

다음 동영상을 통해 애플리케이션 대시보드에 대해 알아보세요.

# 애플리케이션 대시보드

홈 화면 > 프로젝트 선택 > 대시보드 > 애플리케이션 대시보드



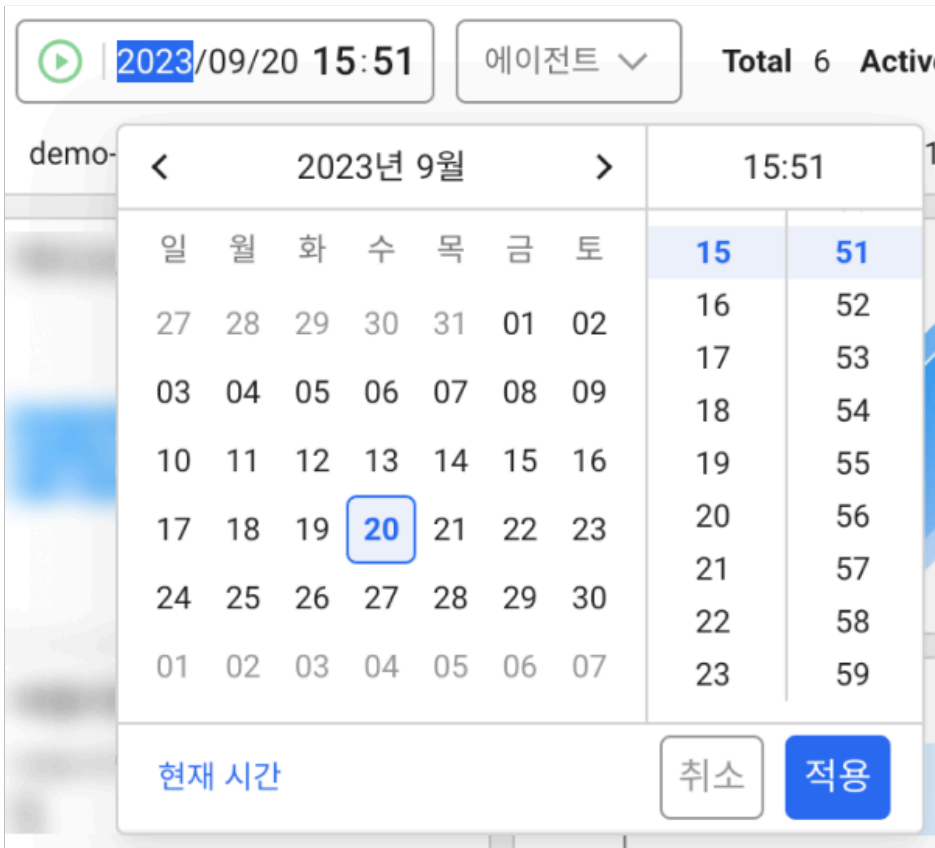
애플리케이션 대시보드에서는 애플리케이션의 주요 지표들을 에이전트를 통해 수집하고 차트를 구성해 실시간으로 모니터링하고 과거의 데이터를 조회할 수 있습니다. 대시보드에 배치된 위젯들을 통해 애플리케이션의 문제를 빠르게 파악하고 분석할 수 있습니다. CPU, Memory를 제외한 일반적인 차트의 경우 안정적인 데이터는 파란색 계열로 표현되고, 문제로 식별되는 요소들은 붉은색 계열로 표시되어 현황을 쉽게 인지할 수 있습니다.

## 과거 데이터 조회하기

대시보드에서는 실시간 모니터링 기능을 기본 제공하지만 과거 시간의 데이터를 조회할 수도 있습니다.



1. 화면 왼쪽 위에 시간 선택자에서 || 버튼을 선택하세요.
2. 날짜 및 시간 영역을 클릭해 원하는 시간대를 설정하세요.

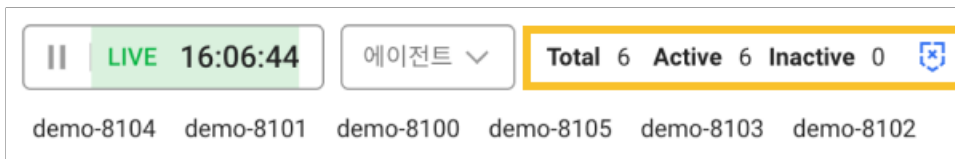


3. 적용 버튼을 선택하세요.

사용자가 설정한 시간을 기준으로 대시보드에 배치한 위젯의 데이터를 갱신합니다. 실시간 모드로 전환하려면 ▶ 버튼을 선택하세요.

# 에이전트 확인하기

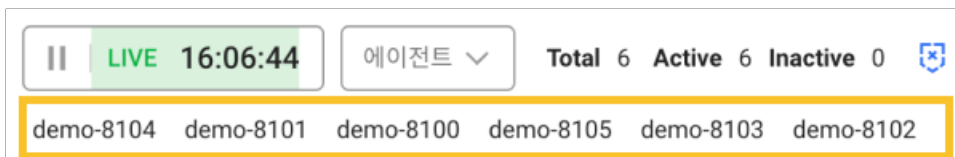
## 에이전트 연결 상태 확인하기



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.

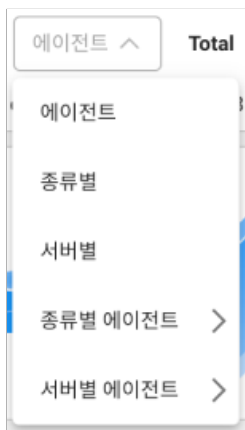
## 에이전트별 모니터링



기본적으로 대시보드에는 모든 에이전트로부터 수집한 지표들을 차트에 표시하지만 에이전트별로 데이터를 조회할 수도 있습니다. 시간 선택자 아래에 위치한 에이전트를 하나 또는 둘 이상을 선택하세요. 선택한 에이전트의 지표들로 대시보드에 배치된 위젯의 데이터를 갱신합니다.

- ✔ 에이전트를 하나 또는 둘 이상을 선택한 상태에서 다시 모든 에이전트를 선택하려면 선택을 해제하거나 **Total**을 선택하세요.
- ① 프로젝트에 연결된 에이전트의 수가 많을 경우 에이전트의 이름을 짧게 설정하는 것이 효율적입니다. 에이전트 이름 설정에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

## 분류별 에이전트 모니터링



에이전트 설정을 통해 분류한 그룹 단위로 에이전트를 선택해 모니터링할 수 있습니다.

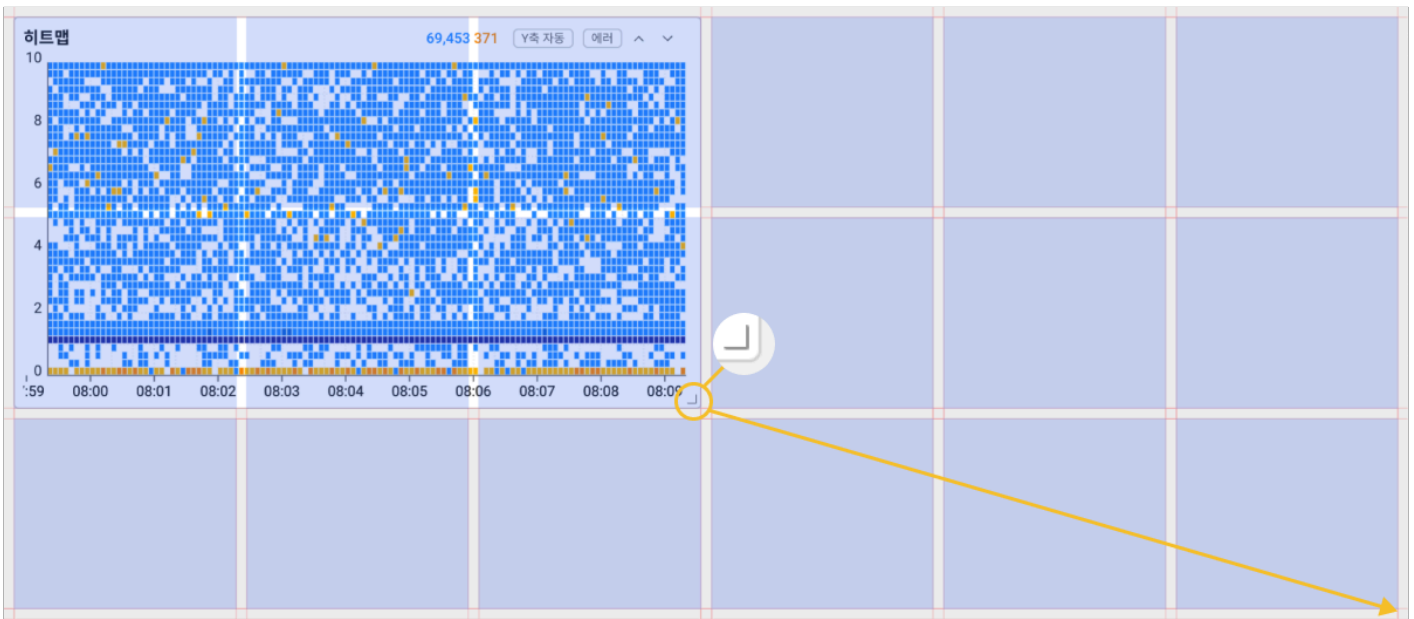
- **에이전트**: 개별 에이전트를 선택하거나 모두 선택할 수 있습니다.
- **종류별**: 에이전트 설정에서 `whatap.okind` 로 분류된 그룹 단위로 에이전트를 모니터링합니다.
- **서버별**: 에이전트 설정에서 `whatap.onode` 로 분류된 그룹 단위로 에이전트를 모니터링합니다.
- **종류별 에이전트**: 에이전트 설정에서 `whatap.okind` 로 분류된 에이전트 목록만을 모니터링합니다.
- **서버별 에이전트**: 에이전트 설정에서 `whatap.onode` 로 분류된 에이전트 목록만을 모니터링합니다.


① 에이전트 설정에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

## 대시보드 위젯 편집하기

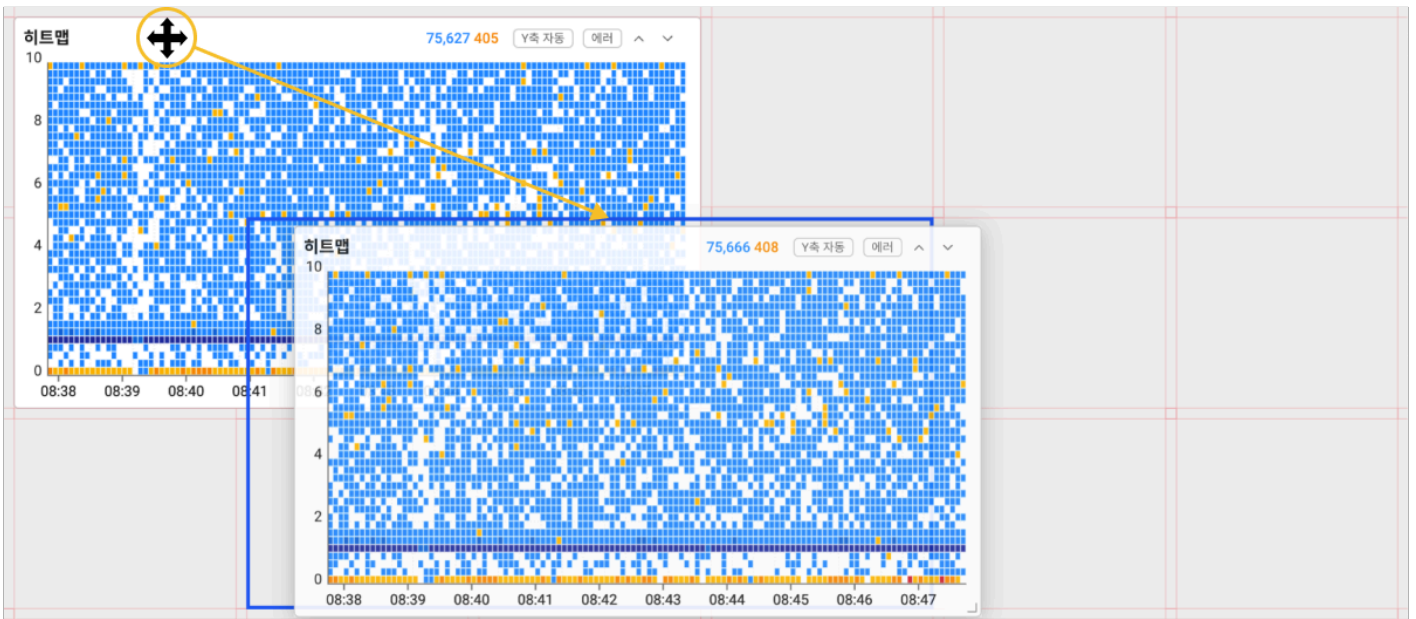
대시보드에 배치한 위젯은 사용자가 원하는 크기로 조절할 수 있고, 원하는 위치에 배치할 수 있습니다. 불필요하다고 생각되는 위젯은 삭제하고 다시 추가할 수도 있습니다.

## 위젯 크기 조절하기



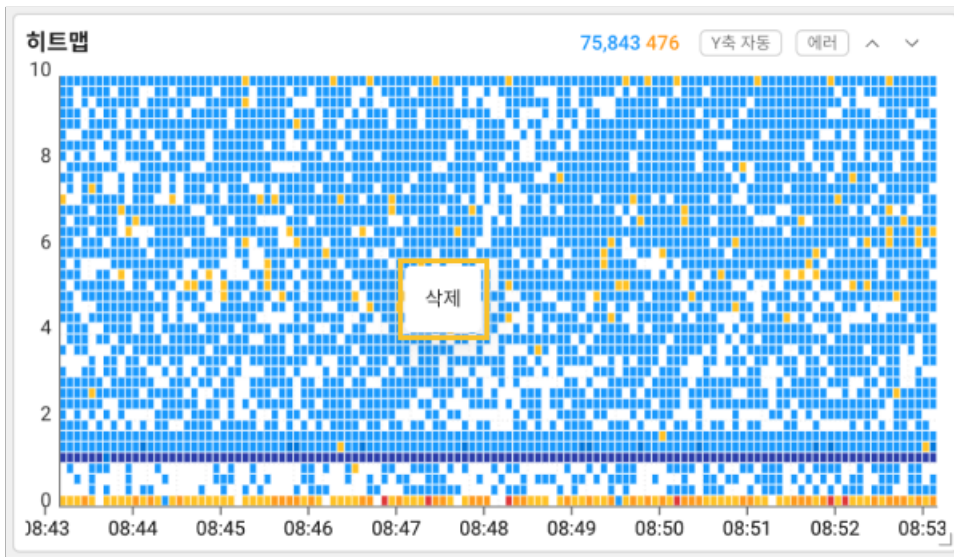
위젯의 오른쪽 아래에  요소를 마우스로 클릭한 상태에서 원하는 크기로 드래그하세요. 균일한 가로, 세로 비율의 격자가 표시되고, 격자 단위로 위젯의 크기를 조절할 수 있습니다.

## 위젯 이동하기



위젯의 위 부분으로 마우스 커서를 이동하면 커서 모양이 **+** 모양으로 변경됩니다. 이때 마우스 왼쪽 버튼을 클릭한 상태로 원하는 위치로 드래그하여 위젯을 이동할 수 있습니다.

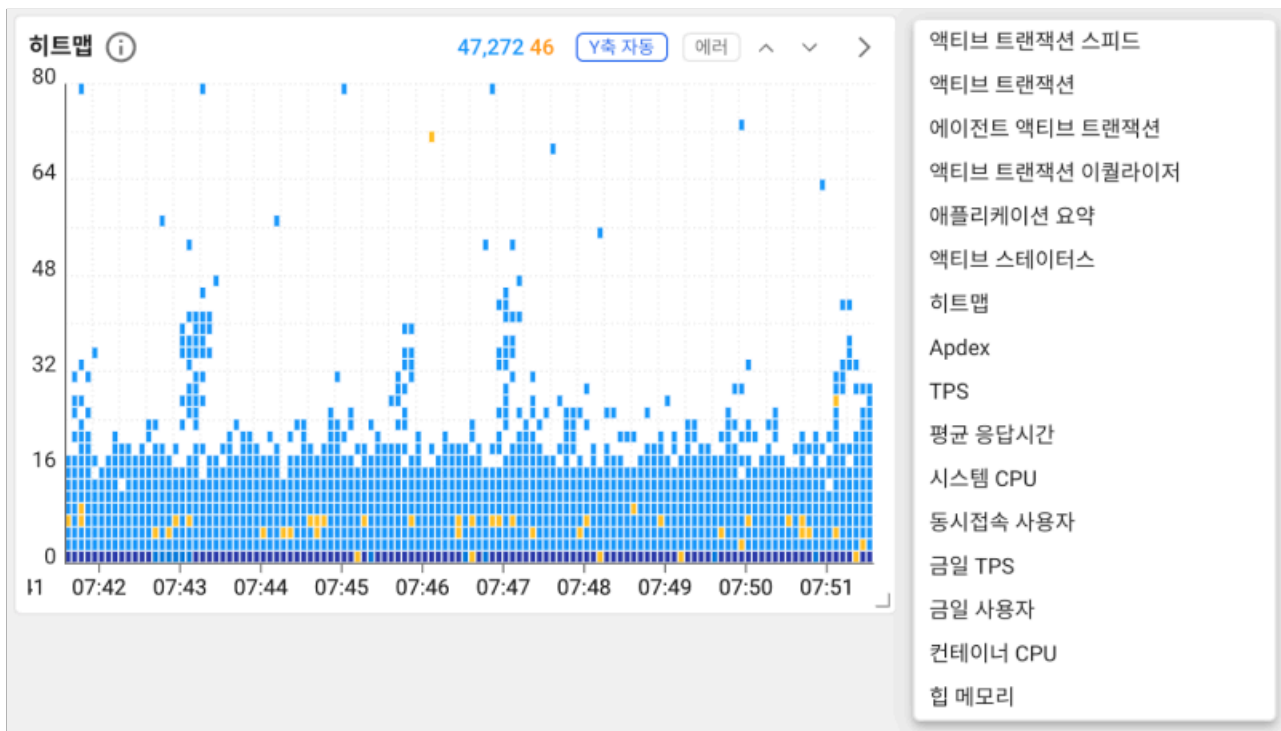
## 위젯 삭제하기



삭제하길 원하는 위젯에서 마우스 오른쪽 버튼을 클릭하세요. [삭제](#) 버튼을 선택하면 해당 위젯이 대시보드에서 삭제됩니다.



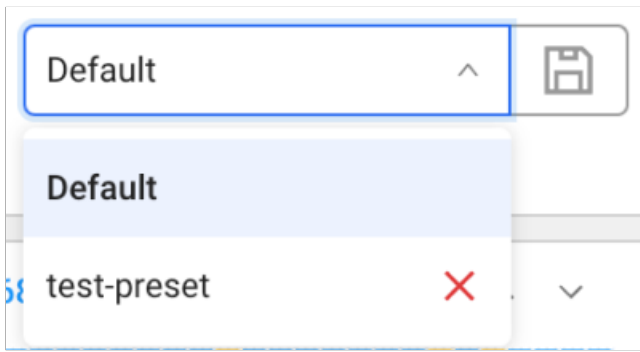
## 위젯 추가하기



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


- ⓘ • 대시보드에 배치할 수 있는 위젯에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.
- 현재 추가할 수 있는 위젯은 고정적이지만 향후 업데이트를 통해 위젯 지원을 늘려갈 계획입니다.

## 프리셋



대시보드에서 사용자가 설정한 위젯의 설정과 레이아웃 상태를 저장하고 불러올 수 있습니다. 기본 프리셋(Default)은 변경할 수 없으며, 원하는 형식으로 위젯의 크기를 조절하고, 배치해 새로운 프리셋을 만들 수 있습니다.

### 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.

 A screenshot of a dialog box titled '프리셋 저장하기' (Save Preset). At the top right, there is a 'Default' dropdown menu and a save icon. The main text says '현재 설정된 지표와 레이아웃을 저장합니다.' (Save the currently set indicators and layout). Below this is a text input field containing 'Default'. There is a checkbox labeled '에이전트 선택 내역 포함' (Include agent selection history) which is currently unchecked. At the bottom right, there are two buttons: '취소' (Cancel) and '저장' (Save).


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

4. **저장** 버튼을 선택하세요.

프리셋 목록에서 새로 저장한 프리셋을 확인할 수 있습니다.



- ① 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 사용자 계정으로 저장되며 현재는 다른 사용자와 공유할 수 없습니다. 더 나은 기능을 제공하도록 업데이트 예정입니다.

## 프리셋 삭제하기

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

## 위젯 옵션 이용하기

위젯에 표시된 아이콘 버튼의 기능은 다음과 같습니다.

- ① : 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.

- ① 위젯에 따라 제공되는 옵션은 다를 수 있습니다.

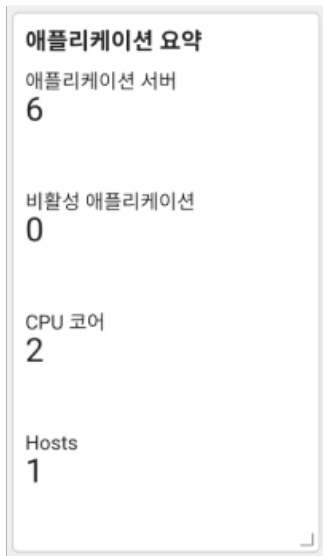
## 위젯 알아보기

- ① 이 문서에서는 [애플리케이션 대시보드](#)에 배치된 위젯에 대한 설명을 주로 다룹니다. [애플리케이션 대시보드](#)를 통해 트랜잭션 및 사용자, 서비스, 리소스 분석 방법에 대한 자세한 내용은 다음 문서를 참조하세요.
  - [애플리케이션 대시보드 살펴보기](#)
  - [히트맵 트랜잭션](#)
  - [DB 연결 지연과 커넥션 풀](#)

## 에이전트 관련 위젯

애플리케이션에 설치한 에이전트를 통해서 애플리케이션의 정보를 확인할 수 있습니다.

### 애플리케이션 요약

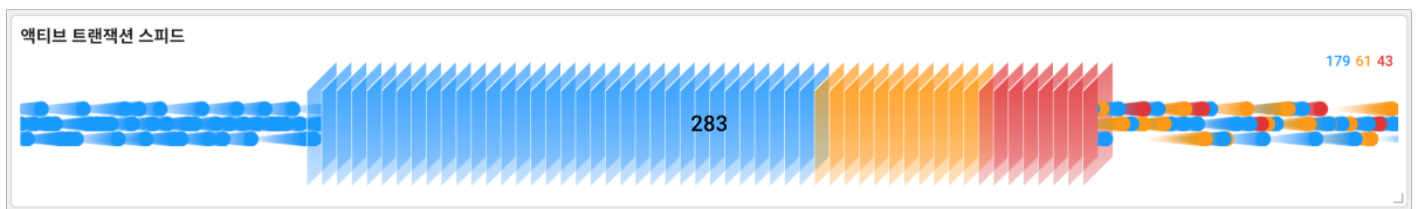


프로젝트에 등록된 애플리케이션의 실행 상태 및 개수, CPU 코어 수, Hosts 수를 표시합니다.

## 트랜잭션 관련 위젯

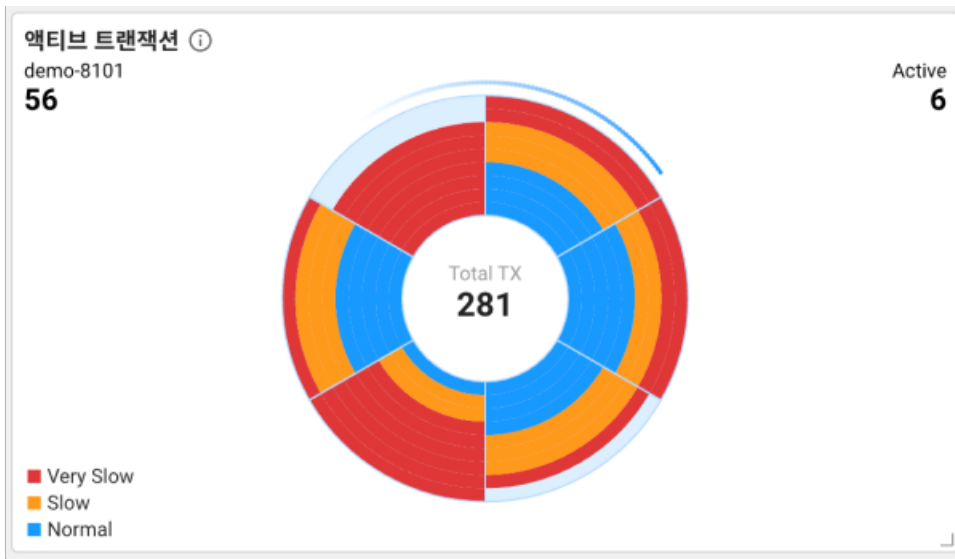
트랜잭션을 통해 확인할 수 있는 장애의 현황은 우선 응답시간을 통해 알 수 있습니다. 또한 진행 중인 트랜잭션이 종료되지 않는다면 이 또한 장애로 인식해야 합니다. 와탭은 진행 중인 상태의 시간에 따라 구간을 나누어 표시합니다. **파랑색(Normal)**은 응답 시간이 정상인 트랜잭션, **주황색(Slow)**은 응답 시간이 8초 정도의 느린 트랜잭션, **빨간색(Very slow)**은 응답 시간이 보통의 2배 이상으로 느린 트랜잭션을 의미합니다. 이를 통해 사용자는 직관적으로 가장 빨리 장애를 인지할 수 있습니다.

### 액티브 트랜잭션 스피드



현재 진행 중인 트랜잭션(가운데 영역)과 종료된 트랜잭션(오른쪽 영역) 현황을 확인할 수 있습니다. 왼쪽에서 오른쪽으로 지나가는 물방울 모양의 아이콘은 트랜잭션을 의미합니다. 액티브 트랜잭션의 상태를 속도(Normal, Slow, Very Slow)로 색상으로 분류하여 우측 상단에 개수로 표시합니다.

### 액티브 트랜잭션

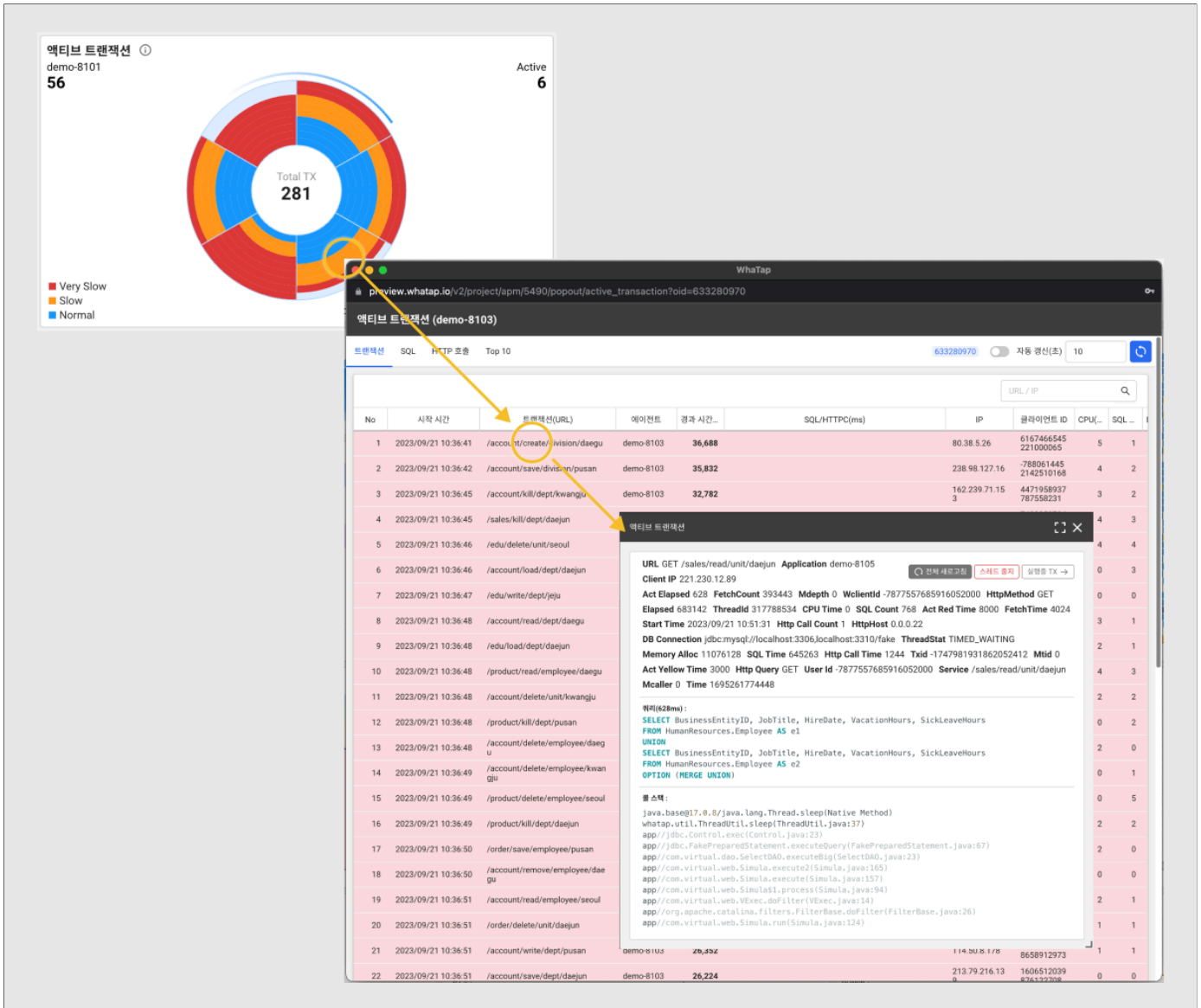


액티브 트랜잭션 위젯에서는 진행 중인 트랜잭션을 **아크 이퀄라이저** 차트로 표현합니다. 지연 발생 현황을 에이전트별로 확인할 수 있습니다. 에이전트 수만큼 아크가 분할됩니다. 5초마다 현재 서버에서 처리 중인 요청의 수를 표현해서 해당 요청이 각각 어느 정도의 시간 동안 처리 중인지 알 수 있습니다. 5초 간격의 시간에 감지된 요청들은 위험 여부를 파악할 수 있도록 다음과 같이 색으로 분류합니다.

- 빨간색(Very Slow): 8초 이상
- 주황색(Slow): 3초 ~ 8초
- 파란색(Normal): 0초 ~ 3초

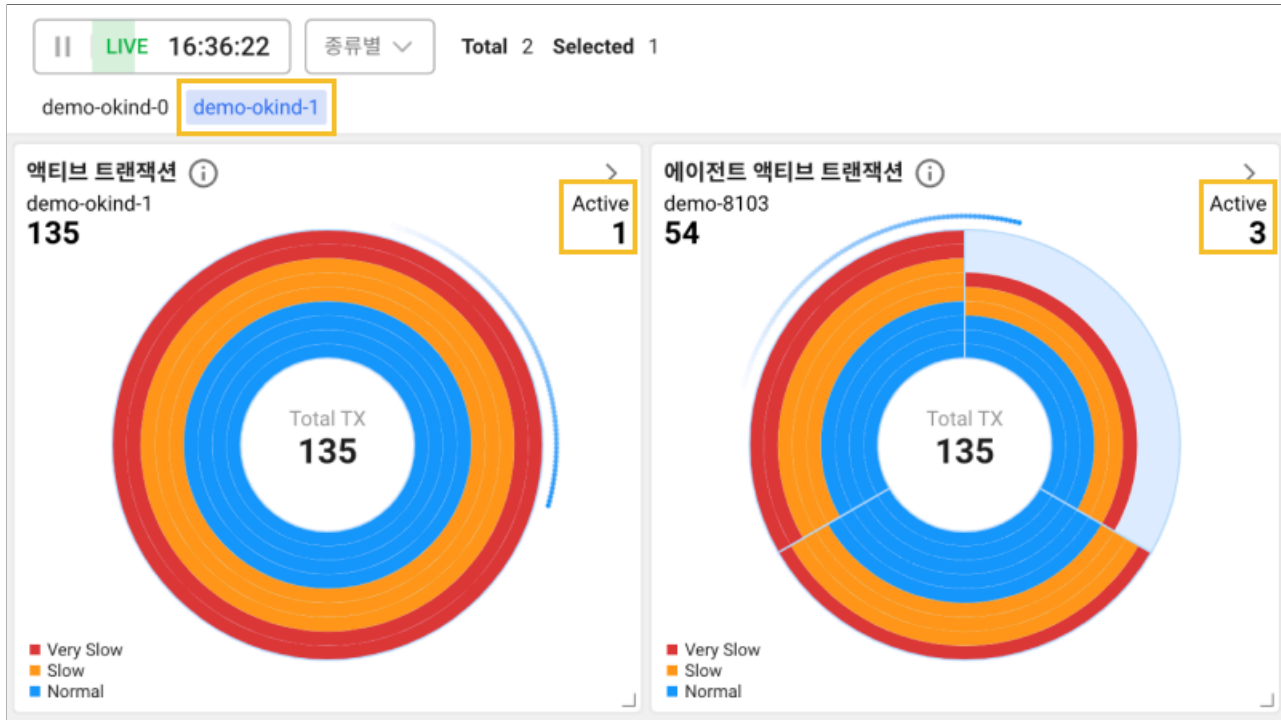
파란색이 많이 표현되는 상황은 문제가 되지 않습니다. 이 중 일부가 주황색이나 빨간색으로 변경되는지 추이를 지켜봐야 합니다. 장애 상황에는 진행 중 트랜잭션 수가 증가하고 빨간색 비율이 높아집니다. 반면, 응답이 매우 빠른 시스템의 경우 처리하는 트랜잭션 수(TPS)는 높아도 진행 중인 트랜잭션 수는 낮을 수 있습니다.

- 왼쪽 위에는 진행 중인 트랜잭션이 가장 많은 에이전트 이름을 표시합니다.
- 가운데 숫자는 전체 에이전트의 액티브 트랜잭션 수의 합계입니다.
- 액티브 트랜잭션 둘레에 두 개의 바는 처리량(TPS)에 따라 3단계 속도로 차트 주변을 회전합니다.
- 분할된 아크 영역을 선택하면 해당 에이전트에서 진행 중인 트랜잭션 목록 팝업창이 나타납니다. 상세한 트랜잭션 정보를 확인하려면 트랜잭션 목록에서 원하는 항목을 하나 선택하세요.



- ① ○ 트랜잭션 목록에서 선택한 항목이 이미 종료된 트랜잭션일 경우 **트랜잭션 보기** 버튼을 선택하세요. **트랜잭션 정보** 창이 나타납니다. **트랜잭션 정보** 창에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.
- 트랜잭션에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

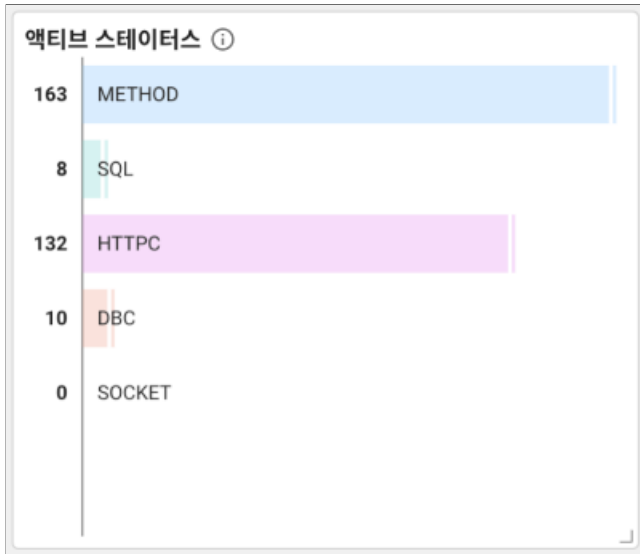
## 에이전트 액티브 트랜잭션



액티브 트랜잭션 위젯은 에이전트 선택 옵션에서 선택한 기준에 따라 병합된 데이터를 차트에 표시합니다. 반면 에이전트 액티브 트랜잭션 위젯은 선택한 에이전트 분류 기준에 포함된 에이전트의 현황을 개별로 확인할 수 있습니다.

대시보드에서 마우스 오른쪽 버튼을 클릭한 다음 위젯 목록에서 에이전트 액티브 트랜잭션 위젯을 선택하세요.

## 액티브 스테이터스



프로젝트 내의 모든 에이전트는 5초마다 액티브 트랜잭션 수를 수집함과 동시에 진행 상태를 수집합니다. 진행 상태는 **METHOD**, **SQL**, **HTTPC**, **DBC**, **SOCKET**으로 분류할 수 있습니다.

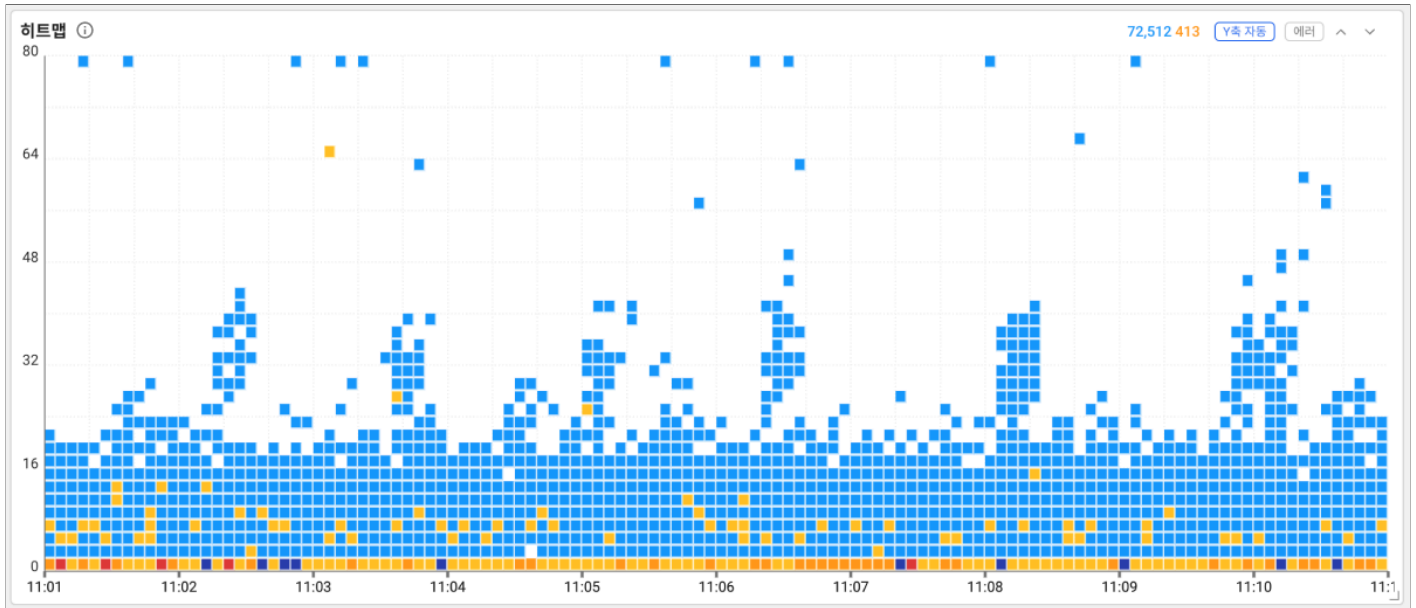
- **METHOD**: 메소드를 수행 중인 상태입니다. 일반적인 상황으로 파란색으로 표시됩니다.
- **SQL**: SQL을 수행 중인 상태입니다. **SQL** 수치가 증가한다면 데이터베이스 서버와의 연결에 문제를 확인해 보세요.
- **HTTPC**: 외부 API를 호출 중인 상태입니다. **HTTPC** 수치가 증가한다면 외부와 연결된 서버의 응답이 제대로 이루어지지 않는다고 봐야 합니다.
- **DBC**: 트랜잭션이 Connection Pool로부터 새로운 Connection을 획득(get)하려는 상태입니다. DB Connection Pool의 개수가 부족하면 새로운 연결 요청이 발생할 때마다 지연이 되면서 성능 장애의 원인이 됩니다. 이 경우 **DBC** 수치가 증가합니다.

ⓘ DB 연결 지연과 관련한 자세한 내용은 [다음 문서](#)를 참조하세요.

- **SOCKET**: 외부 시스템과의 TCP 연결 시도를 의미합니다. **SOCKET** 수치가 지속적으로 증가한다는 것은 외부 시스템과의 연결이 되지 않아 장애가 발생 중일 가능성이 높습니다.



## 히트맵



히트맵 차트는 종료된 트랜잭션 응답시간을 분포도로 표현합니다. 가로축은 트랜잭션 종료 시간입니다. 세로축은 수행 시간입니다. 수행 시간이 긴 트랜잭션은 분포도 상단에 위치합니다. 히트맵 내의 차트 영역을 드래그하면 세부 트랜잭션 정보를 확인할 수 있는 [트레이스 분석](#) 창이 나타납니다. [트레이스 분석](#)에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

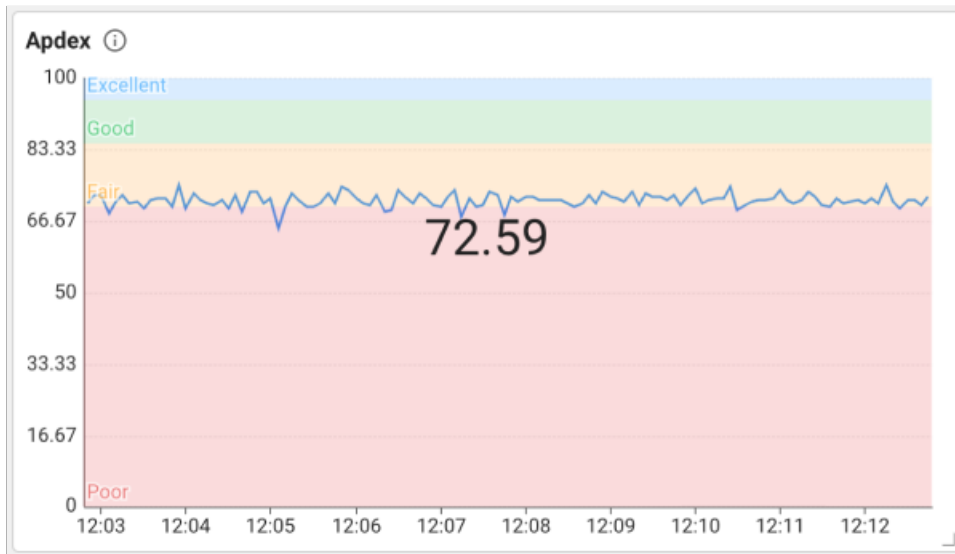
- 정상 트랜잭션은 하늘색 → 파란색 → 남색 순으로 표현됩니다.
- 에러 트랜잭션은 노란색 → 주황색 → 빨간색 순으로 표현됩니다.
- 오른쪽 위에 위치한 숫자는 차트 내 **트랜잭션 건 수/에러 건수**를 의미합니다.
- **Y축 자동** 버튼을 선택하면 수행 시간을 나타내는 Y축을 자동 조정합니다.
- **에러** 버튼을 선택하면 에러 트랜잭션만 표시합니다.
- **^** 또는 **v** 버튼을 선택해 5초 ~ 80초까지 차트를 확대/축소할 수 있습니다.

- ⓘ • 오류가 발생하지 않았고 수행 시간 500ms 이하 트랜잭션의 세부 정보는 url 당 5분에 1건만 수집됩니다.
  - TPS나 평균 응답시간 같은 통계정보에는 영향을 미치지 않습니다.
  - 에이전트 설정을 통해 정책을 설정할 수 있습니다. 자세한 내용은 [다음 문서](#)를 참조하세요.
  - 히트맵 위젯을 통해 트랜잭션을 분석하는 과정은 [다음 문서](#)를 참조하세요.

## 서비스 관련 위젯

서비스 관련 위젯을 통해 애플리케이션 성능 튜닝을 위한 지표로 활용할 수 있습니다.

### Apdex



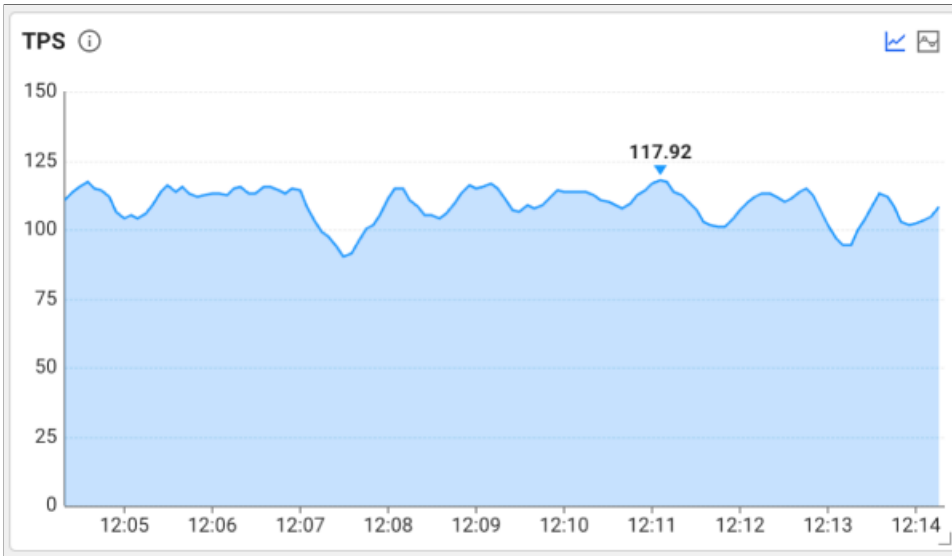
[Application Performance Index\(Apdex\)](#)는 애플리케이션 성능 지표입니다. 웹 애플리케이션의 고객 만족도를 측정합니다. 사용자 만족도에 대한 지표로 활용할 수 있으며, 0 ~ 1 사이의 값을 갖습니다.

$$(만족 횟수 + (허용 횟수 * 0.5)) / 전체 요청 수$$

- 만족 (Satisfied, S): 업무처리에 전혀 문제 없음 ≤ 1.2초 (만족 S 기본값)
- 허용 (Tolerating, T): 사용자가 지연을 느끼나 업무처리는 가능 ≤ 4.8초 (만족 S \* 4)
- 불만 (Frustrated, F): 업무처리가 불가능 > 4.8초 (허용 T 초과 및 오류)

ⓘ 만족 S 기본값은 [관리 > 에이전트 설정](#) 메뉴에서 변경할 수 있습니다. 설정 키 값은 `apdex_time` 입니다. 에이전트 버전 2.0 이상만 지원합니다.

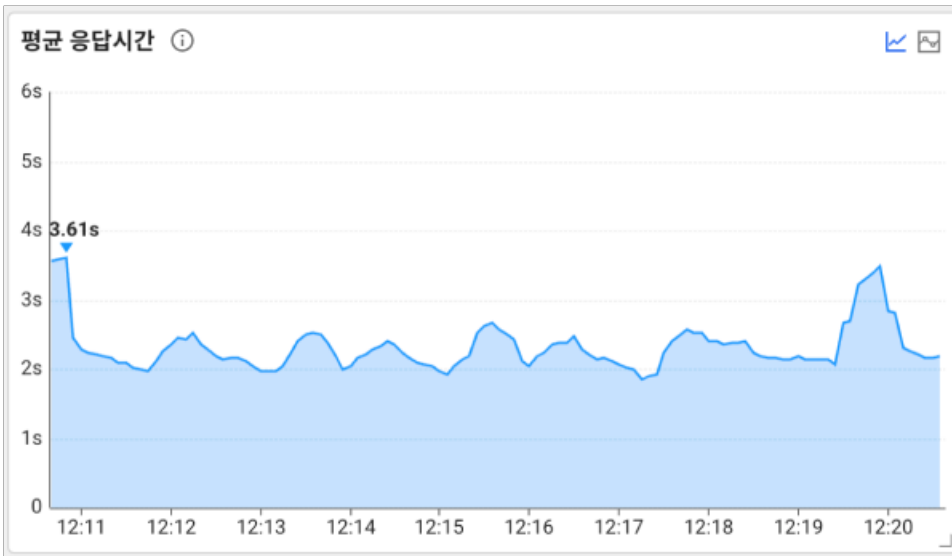
## TPS



**Transaction Per Second(TPS)**는 초당 처리한 트랜잭션 건수를 의미하며 서비스 성능 지표의 기준이 됩니다. 5초마다 처리한 트랜잭션의 수를 초당 건수로 환산해 차트로 표현합니다. 최근 10분간의 TPS를 보여줍니다.

ⓘ TPS, 응답시간, CPU 사용률 분석에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

## 평균 응답시간



애플리케이션 서버가 사용자에게 요청 결과를 반환하는 데 걸리는 시간을 실시간으로 모니터링합니다. **평균 응답시간**은 튜닝 지표로서 의미를 가집니다. 와탭의 모니터링 서비스는 5초 간격으로 트랜잭션의 평균 응답 시간을 계산합니다.

ⓘ TPS, 응답시간, CPU 사용률 분석에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

## 리소스 관련 위젯

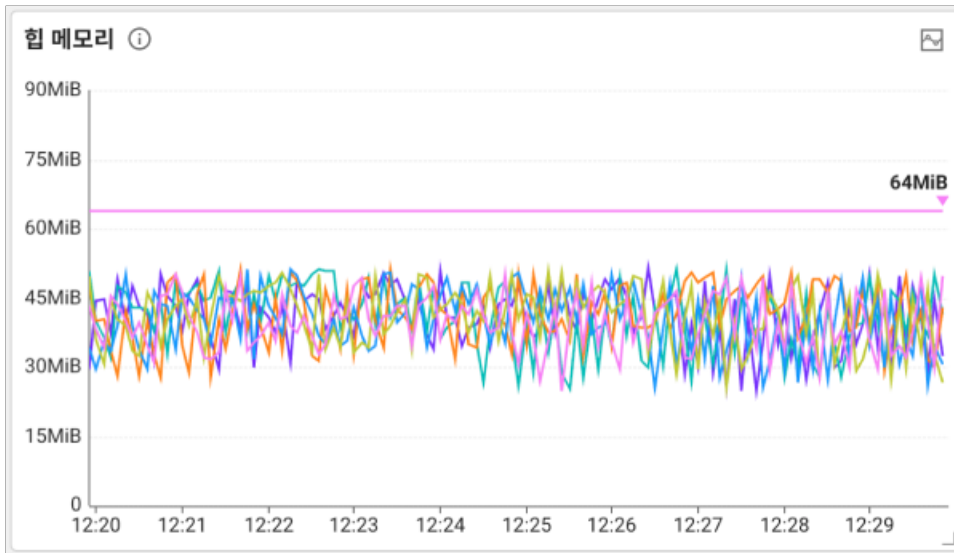
### 시스템 CPU



애플리케이션 서버의 CPU 사용량입니다. 실시간으로 CPU 사용량 변화 추이를 파악할 수 있습니다.

ⓘ TPS, 응답시간, CPU 사용률 분석에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

## 힙 메모리



각 서버당 사용할 수 있는 최대 메모리와 현재 메모리를 표시해 위험 수치에 있는 서버를 확인할 수 있습니다. 시간에 따른 메모리 사용량 변화를 실시간으로 볼 수 있습니다.

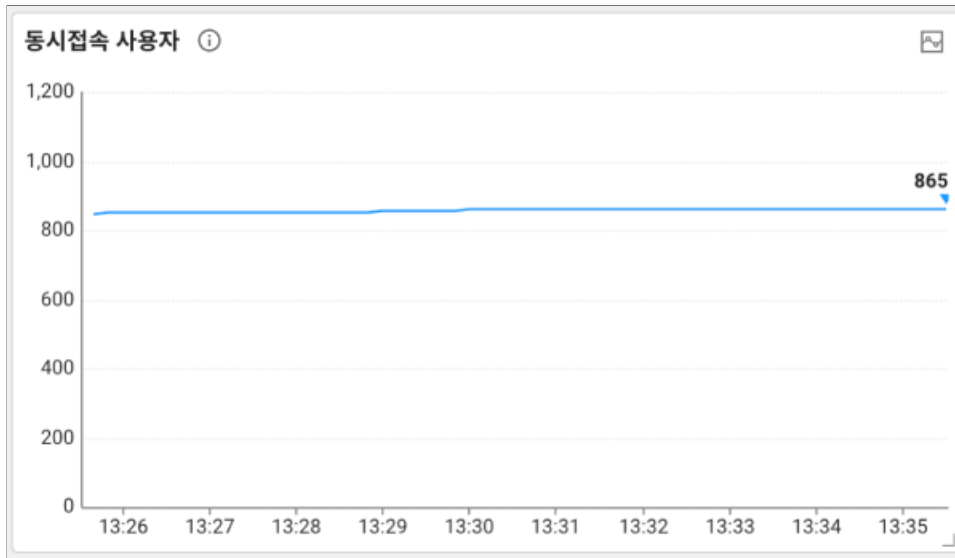
메모리 라인 차트는 보통 계속해서 물결칩니다. 애플리케이션 서버가 요청을 처리하기 위해 메모리를 사용할 때 증가합니다. GC(Garbage Collection)를 통해서 메모리를 정리할 경우에는 감소합니다. **힙 메모리** 지표는 평균값을 사용해 표기합니다.

✔ 힙 메모리 차트 분석에 대한 자세한 내용은 다음 링크를 참조하세요.

- [월간 와탭 : 모니터링에 주목해야할 지표](#)
- [Java 힙 메모리 차트 분석 : Ch.1 힙차트 관찰하기](#)
- [JAVA 힙메모리 차트 분석 : Ch.2 메모리 릭, 그리고 힙덤프 분석](#)

## 사용자 관련 위젯

### 동시접속 사용자



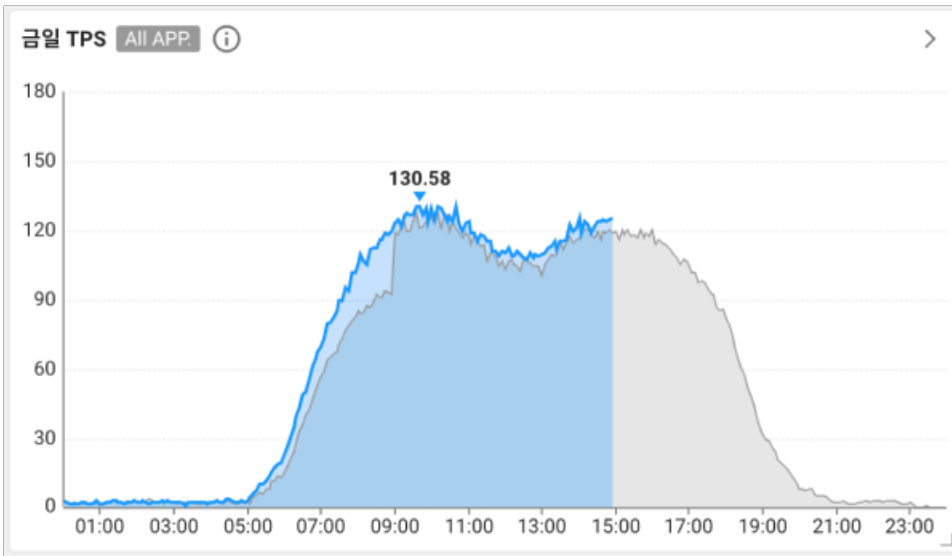
실시간 브라우저 사용자 수를 보여줍니다. 5초마다 최근 5분 이내에 트랜잭션을 일으킨 사용자를 중복없이 카운팅해 표시합니다. 사용자 브라우저의 IP를 기반으로 카운팅합니다.

- ① • 에이전트 설정에서 사용자 구분을 위해 IP를 사용하거나 쿠키를 사용할 수 있습니다. 자세한 내용은 [다음 문서](#)를 참조하세요.
- 일반적으로 접속 사용자는 현재 네트워크로 접속해 연결된 사용자를 의미합니다. 하지만 웹 시스템은 비 연결 네트워크를 사용하기 때문에 다릅니다. 접속되어 있다는 의미보다는 최근 요청을 보낸 사용자가 측정 대상이 될 수 있습니다.
- 실시간(동시접속) 사용자는 같은 시간대에 시스템을 사용하고 있는 사용자를 의미합니다. 실시간(동시접속) 사용자를 산정하거나 측정하는 방식은 다양합니다.
- 사용자 수는 단순히 합산하는 것이 아니라 **HyperLogLog**로 산출합니다. **HyperLogLog**는 매우 적은 메모리로 집합의 원소 개수를 추정하는 확률적 자료 구조를 의미합니다.

## 1일 기준 비교 위젯

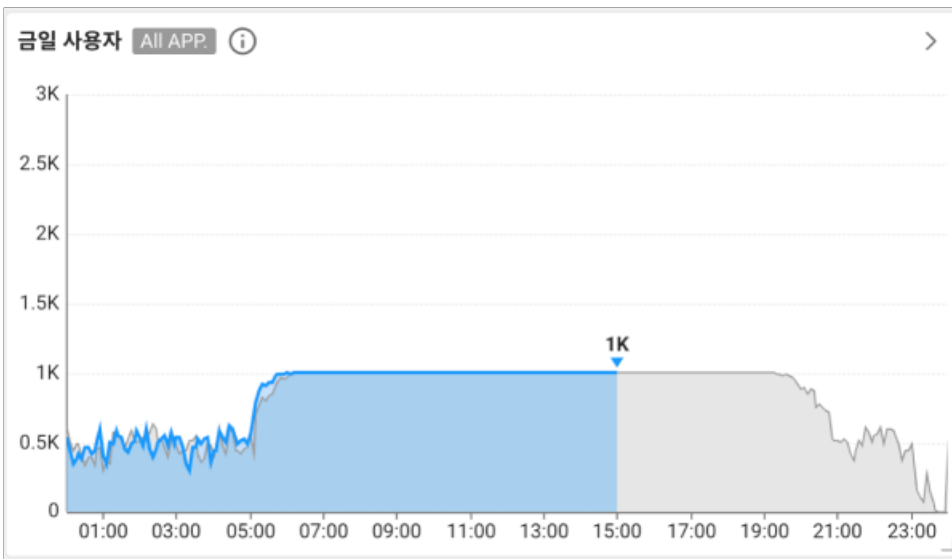
- ① All APPS. 태그가 표시된 위젯은 항상 모든 에이전트에 대한 데이터를 표시합니다.

## 금일 TPS



금일 하루 동안의 TPS를 시간대별로 표시합니다. 어제 동시간대의 TPS(회색)와 비교해 확인할 수 있습니다. > 버튼을 클릭하면 [통계/보고서 > 일자별 애플리케이션 현황](#) 메뉴로 이동합니다.


## 금일 사용자



하루 동안의 동시접속 사용자 수를 시간대별로 나타낸 지표입니다. 어제 동시간대의 사용자 수(회색)와 비교해 확인할 수 있습니다. > 버튼을 클릭하면 [통계/보고서 > 일자별 애플리케이션 현황](#) 메뉴로 이동합니다.



## 실시간 알림 확인하기

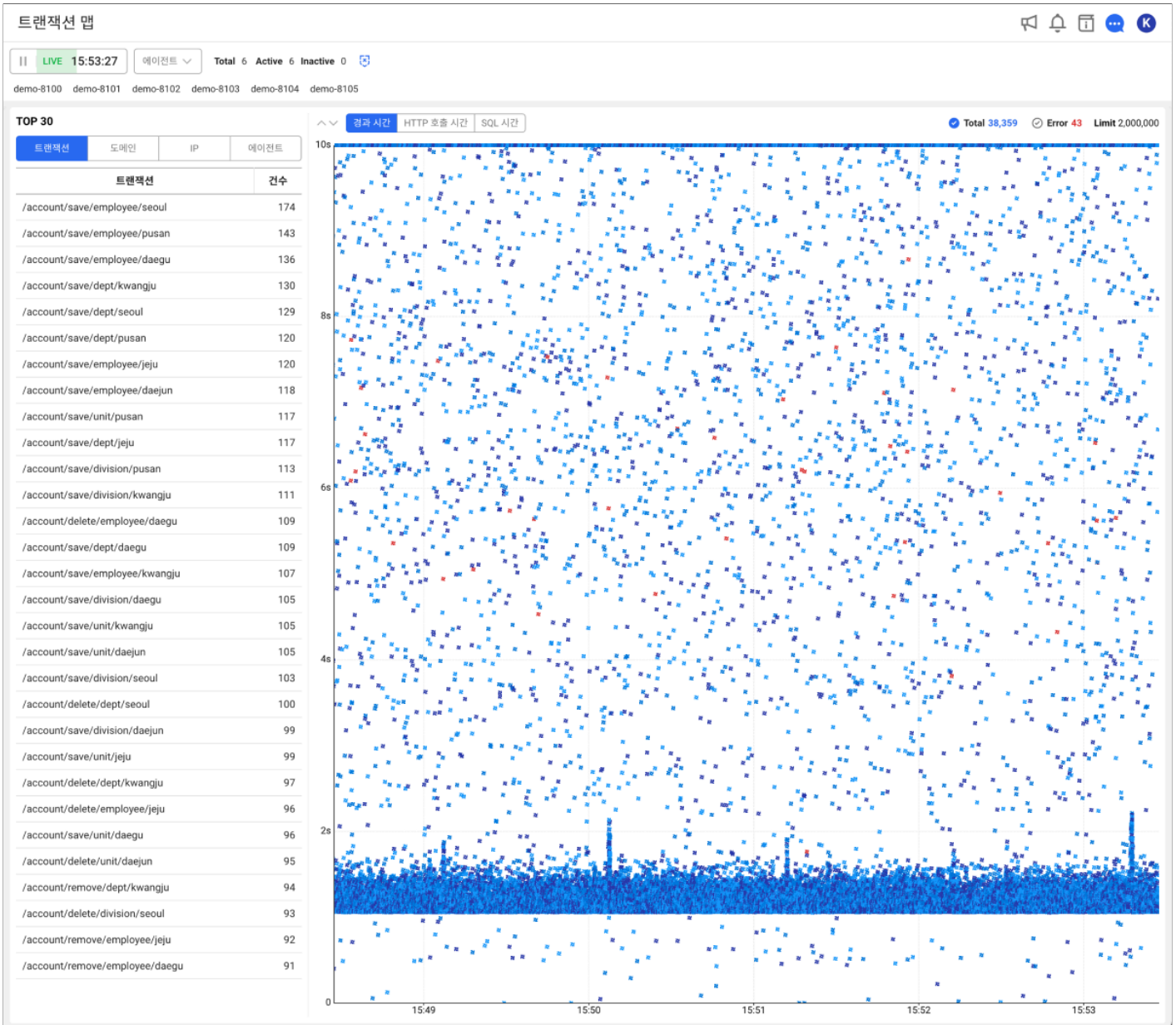
화면 오른쪽 위에  실시간 알림 버튼을 선택하면 최근 발생한 이벤트를 확인할 수 있습니다. 토글 메뉴를 클릭해 브라우저 알림을 켜거나 끌 수 있습니다.

❗ 화면 가장 위에 고정 메뉴 영역의 기본 요소들에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

# 트랜잭션 맵

홈 화면 > 프로젝트 선택 > 대시보드 > 트랜잭션 맵

트랜잭션 맵 차트는 종료된 개별 트랜잭션의 응답시간을 분포도 형태로 표현한 차트입니다. 히트맵과 동일하게 분포 패턴에 따른 문제점을 발견하고 분석할 수 있습니다. 히트맵은 5초 단위로 트랜잭션을 그룹화해서 보여주지만 트랜잭션 맵은 트랜잭션을 개별로 표시합니다.



① [Service 2.2.0 릴리스](#)를 통해 변경된 사항은 다음과 같습니다.

- 실시간 최대 5분, 과거 시점 최대 10분까지 데이터를 조회할 수 있습니다. (기존: 실시간 최대 5분 한정)
- 과거 시점의 데이터를 조회할 경우 서버로부터 받은 데이터를 필터링할 수 있습니다.
- [트레이스 분석](#) 창을 통해 조회할 수 있는 데이터 개수가 100건에서 1000건으로 늘어났습니다.

- ① 트랜잭션 및 에러 개수를 실시간으로 표시합니다.
- TOP 30 목록 및 차트 영역에 키보드 이벤트를 추가했습니다. 키보드 방향 버튼을 선택하면 원하는 위치로 이동할 수 있습니다.  
트랜잭션 맵 메뉴의 개편과 관련한 자세한 내용은 [다음 링크](#)를 참조하세요.

### ✔ 히트맵 분석과 다른 점은?

히트맵 차트는 특정 구간 동안의 트랜잭션 발생 건을 합산하여 표시합니다. 발생 수가 많은 곳을 색상으로 차이를 표현하고 있습니다. 히트맵의 경우에는 구간별 합산 데이터를 따로 저장하고 있어 비교적 긴 시간의 추이도 확인할 수 있습니다. 하지만 트랜잭션을 합산해 표시하기 때문에 개별 트랜잭션 정보를 알기 어렵습니다.

- 장기간, 대용량 트랜잭션 분포도를 분석하려면 히트맵 분석([분석 > 히트맵](#))을 이용하세요.
- 성능 테스트를 하거나 장애 상황에서 1~5분 이내의 트랜잭션 분포도 분석, 수행 시간 제한 없이 트랜잭션 분포도를 확인하려면 [트랜잭션 맵](#)을 확인하세요.

- ① • [히트맵](#) 메뉴에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.
- [히트맵 트랜잭션](#) 차트를 분석 방법에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.
- 서버 시간보다 사용자의 PC 시간이 과거로 설정된 경우, 서버 시간과 사용자의 PC 시간이 5분 이상 차이가 날 경우 차트에 데이터가 출력되지 않을 수 있습니다.

## 트레이스 분석하기

트랜잭션 맵의 차트 영역을 드래그하면 세부 트랜잭션 정보를 확인할 수 있는 [트레이스 분석](#) 창이 나타납니다. 트랜잭션 목록과 각 트랜잭션 하위의 스텝 정보를 한 번에 확인할 수 있습니다.



차트 영역을 드래그한 다음 **트레이스 분석** 창에서 조회할 수 있는 데이터는 최대 1,000건입니다.

① 트레이스 분석에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

## 조회 기준 변경하기

트랜잭션 맵의 차트 영역, 왼쪽 위에 버튼을 선택해 데이터 조회 기준을 변경할 수 있습니다.

- **경과 시간**: 트랜잭션이 수행된 경과 시간 기준으로 데이터를 조회합니다.
- **HTTP 호출 시간**: HTTP 호출에 대한 응답 시간 기준으로 데이터를 조회합니다.
- **SQL 시간**: SQL 패치 시간을 기준으로 데이터를 조회합니다.

차트 영역, 오른쪽 위에 **Error** 버튼을 선택하면 에러가 발생한 트랜잭션만을 조회할 수 있습니다. 이 상태에서 차트의 트랜잭션을 드래그하면 에러 상태의 트랜잭션 목록이 표시된 **트레이스 분석** 창을 열 수 있습니다. 다시 전체 트랜잭션을 조회하려면 **Total** 버튼을 선택하세요.

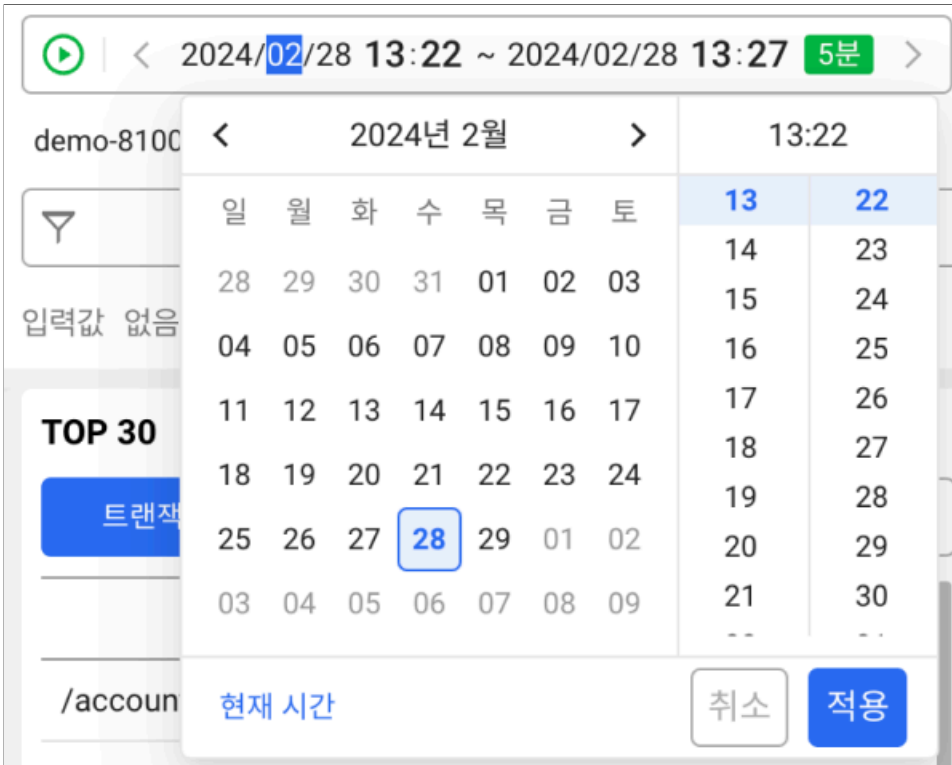
✔ 차트의 왼쪽 위에 ^ 또는 v 버튼을 선택하면 Y축의 최댓값을 쉽게 변경할 수 있습니다. 또한 키보드의 위 또는 아래 방향 버튼을 눌러 같은 기능을 이용할 수 있습니다.

ⓘ 트랜잭션 맵의 차트에 표시되는 트랜잭션의 최대 개수(TX Max)는 2,000,000개입니다.

## 과거 데이터 조회하기

**트랜잭션 맵**은 실시간 모니터링을 기본 제공합니다. 기본값으로 5분 동안의 데이터를 실시간으로 조회할 수 있습니다. 시간 선택자의 녹색 버튼을 클릭해 원하는 조회 시간을 선택하세요.

과거 데이터를 조회하려면 시간 선택자에서 || 버튼을 선택하세요. 비실시간 모드로 변경되며 최대 10분까지의 트랜잭션 데이터를 조회할 수 있습니다.



원하는 날짜와 시간을 설정한 다음 **적용** 버튼을 선택하세요. 사용자가 설정한 시간을 기준으로 **트랜잭션 맵**의 차트 데이터를 갱신합니다.

- ① 과거 조회 범위는 트랜잭션의 양에 따라 달라질 수 있습니다.
- 과거 데이터를 조회한 다음 실시간 모드로 전환하려면 **🕒** 버튼을 선택하세요.

## 필터 적용하기

과거 데이터를 조회하게 되면 필터를 적용해 원하는 데이터를 빠르게 필터링할 수 있습니다.

1. 시간 선택자에서 **||** 버튼을 선택하면 필터 입력 상자가 나타납니다.



2. 필터 입력 상자를 선택하면 선택할 수 있는 필터 항목이 나타납니다. 선택할 수 있는 항목은 다음과 같습니다.

- 경과 시간
- 트랜잭션
- 도메인
- IP

3. 필터링할 수 있는 값이 목록으로 표시되면 원하는 항목을 선택하세요.

❗ 필터 항목으로 **경과 시간**을 선택하면 사용자가 직접 시간을 입력해야 합니다. 단위는 밀리초(ms)입니다.

4. 🔍 버튼을 선택하세요.

Top 30 목록과 **트랜잭션 맵** 차트에 필터링한 데이터를 갱신합니다.

- ❗ 다중 조건을 설정해 필터링할 수 있습니다. 추가한 조건은 **&&** 기준으로 적용됩니다. 예를 들어, 트랜잭션이 `/account/save/employee/seoul` 이면서 경과 시간이 3초 이상인 경우 다음과 같이 적용할 수 있습니다.

트랜잭션 = /account/save/employee/seoul 경과 시간 (ms) >= 3

- **경과 시간**은 추가로 적용할 수 없습니다.

## 에이전트 확인하기


### 에이전트 연결 상태 확인하기

The screenshot shows a monitoring interface with a 'LIVE' status indicator and a timer at 15:56:00. A dropdown menu is set to '에이전트'. A summary box displays 'Total 6 Active 6 Inactive 0'. Below this, a list of demo servers is shown: demo-8100, demo-8101, demo-8102, demo-8103, demo-8104, and demo-8105.

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

||
LIVE
15:56:00

에이전트
▼

**Total** 6
**Active** 6
**Inactive** 0


demo-8100
demo-8101
demo-8102
demo-8103
demo-8104
demo-8105

기본적으로 차트에는 모든 에이전트로부터 수집한 지표들을 차트에 표시하지만 에이전트별로 데이터를 조회할 수도 있습니다. 시간 선택자 아래에 위치한 에이전트를 하나 또는 둘 이상을 선택하세요. 선택한 에이전트의 트랜잭션 데이터로 차트를 갱신합니다.

- ✓ 에이전트를 하나 또는 둘 이상을 선택한 상태에서 다시 모든 에이전트를 선택하려면 선택을 해제하거나 **Total**을 선택하세요.

- ⓘ 프로젝트에 연결된 에이전트의 수가 많을 경우 에이전트의 이름을 짧게 설정하는 것이 효율적입니다. 에이전트 이름 설정에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

## 상위 목록 확인하기

**Top 30** 섹션에서는 [트랜잭션](#) 및 [도메인](#), [IP](#), [에이전트](#) 기준으로 트랜잭션 수행 건수가 많은 상위 30개의 목록을 표시합니다.

- **트랜잭션**: 트랜잭션 URL 기준으로 집계된 상위 30개의 목록을 확인할 수 있습니다.
- **도메인**: 클라이언트가 접속한 IP 주소에 지정된 인터넷 주소를 기준으로 집계된 상위 30개의 목록을 확인할 수 있습니다.
- **IP**: 클라이언트의 IP 주소를 기준으로 집계된 상위 30개의 목록을 확인할 수 있습니다.
- **에이전트**: 해당 프로젝트에 포함된 에이전트를 기준으로 집계된 상위 30개의 목록을 확인할 수 있습니다.

목록에서 개별 항목을 선택하면 트랜잭션 맵 차트에 선택한 항목 기준으로 데이터를 반영합니다. 여러 개의 항목을 다중 선택하려면 **Ctrl** (Windows/Linux) 또는 **COMMAND** (Mac) 키를 누른 상태에서 원하는 항목을 클릭하세요.

**TOP 30**

트랜잭션	건수
/account/save/employee/seoul	277
/account/save/employee/pusan	235
/account/save/employee/daegu	230
/account/save/employee/kwangju	218
/account/save/dept/daegu	217
/account/save/employee/daejun	210
/account/save/employee/jeju	201
/account/save/division/daegu	197
/account/save/dept/pusan	196
/account/save/division/pusan	187
/account/save/dept/seoul	183
/account/save/dept/daejun	180
/account/save/dept/jeju	174

☑ Top 30 섹션의 목록에서 키보드의 위 또는 아래 방향 버튼을 눌러 항목을 이동할 수 있습니다.

## 사용자 IP 주소 추가하기

TOP 30 섹션의 IP 기준으로 트랜잭션 수행 건수를 조회할 때 특정 IP 주소를 추가해 상시 확인할 수 있는 기능을 제공합니다.

1. TOP 30 섹션에서 IP를 선택하세요.

**TOP 30**


트랜잭션	도메인	IP	에이전트
IP			건수
255.255.255.255			+
10.237.141.102			150

2. 목록의 가장 위에 입력란에 특정 IP 주소를 입력하세요.
3. + 버튼을 선택하세요.

**TOP 30**

트랜잭션	도메인	IP	에이전트
		IP	건수
		255.255.255.255	+
		 255.255.255.255	0
		127.115.101.81	125

목록의 가장 위에 입력한 IP 주소가 고정되어 표시됩니다.

ⓘ 추가한 IP 주소를 삭제하려면  버튼을 선택하세요.

# 액티브 트랜잭션

홈 화면 > 프로젝트 선택 > [대시보드](#) > [액티브 트랜잭션](#)

현재 진행 중인 트랜잭션의 URL, SQL, HTTP 호출 정보로 이상 트랜잭션을 신속하게 파악할 수 있으며, 상세 트레이스를 통해서 지연 구간을 상세하게 분석할 수 있습니다.

# 화면 구성

**1** 액티브 트랜잭션

**2** 전체 에이전트  활성화 에이전트만 보기

트랜잭션 SQL HTTP 호출 Top 10 자동 경신(초) 10

**3**

No	시작 시간	트랜잭션	에이전트 명 (...)	결과 시간...	SQL/HTTP(ms)	클라이언트 IP	WClientID	CPU ...	SQL ...	HTTP...
1	2023/10/06 09:02:58	/sales/pickup/unit/kwangju	demo-8102	1,393,236	[SQL] SELECT ProductModelID, AVG(ListPrice) AS [Average List Price] FROM Production.Product WHERE ListPrice > \$1000 GROUP BY ProductModelID ORDER BY ProductModelID (경과 시간: 473ms)	160.148.184.142	4323044368642708538	40	903	1
2	2023/10/06 09:15:42	/product/create/division/jeju	demo-8105	629,690	[SQL] SELECT ename, deptno, sal, job FROM emp (경과 시간: 415ms)	60.236.25.188	3697720415751884109	173	955	1
3	2023/10/06 09:22:33	/account/write/employee/jeju	demo-8103	218,361	[SQL] SELECT ordernumber, SUM(quantityOrdered) AS itemsCount, SUM(priceeach) AS total FROM orderdetails GROUP BY ordernumber (경과 시간: 326ms)	112.17.247.89	8111412030467165526	104	572	1
4	2023/10/06 09:24:34	/account/kill/dept/pusan	demo-8101	97,195		159.193.161.241	8660080583753574	35	187	1
5	2023/10/06 09:25:55	/account/pickup/employee/daeju	demo-8101	16,276	[HTTP] /remote/order/create/unit/daeju (경과 시간: 1,001ms)	252.111.140.61	623504528287294318	5	3	9
6	2023/10/06 09:25:56	/account/delete/dept/seoul	demo-8104	15,999	[HTTP] /remote/account/pickup/dept/daeju (경과 시간: 761ms)	70.47.221.49	-8699183568114967612	0	3	10
7	2023/10/06 09:25:56	/account/delete/dept/daeju	demo-8100	15,919	[HTTP] /remote/order/delete/dept/seoul (경과 시간: 426ms)	53.199.161.32	5986125078063307531	7	4	12
8	2023/10/06 09:25:56	/account/read/dept/jeju	demo-8103	15,321	[HTTP] /remote/account/write/division/kwangju (경과 시간: 117ms)	78.205.227.191	-285562735664598794	0	4	12
9	2023/10/06 09:25:57	/account/delete/dept/daeju	demo-8102	14,974	[HTTP] /remote/order/pickup/employee/daeju (경과 시간: 1,160ms)	34.17.89.209	-554002463522755325	6	1	12
10	2023/10/06 09:25:57	/account/delete/employee/kwangju	demo-8105	14,508	[HTTP] /remote/product/delete/division/daeju (경과 시간: 547ms)	131.246.229.118	-6849533954250466301	0	5	10
11	2023/10/06 09:25:57	/product/remove/division/kwangju	demo-8100	14,302	[HTTP] /remote/account/save/division/seoul (경과 시간: 746ms)	29.207.62.171	3850294828179258638	6	5	10
12	2023/10/06 09:25:57	/account/write/dept/kwangju	demo-8101	14,047	[HTTP] /remote/sales/pickup/employee/kwangju (경과 시간: 1,254ms)	215.125.31.10	-1575241641797234617	6	3	10
13	2023/10/06 09:25:58	/edu/read/division/daeju	demo-8103	13,665	[HTTP] /remote/account/read/employee/kwangju (경과 시간: 184ms)	28.178.8.254	1005136282425175133	0	4	11
14	2023/10/06 09:25:58	/sales/read/unit/pusan	demo-8102	13,590	[HTTP] /remote/account/create/employee/daeju (경과 시간: 875ms)	101.243.208.67	-4329996680045999463	6	5	10
15	2023/10/06 09:25:58	/account/read/dept/seoul	demo-8105	13,311		237.22.158.163	-3289139337146026640	6	2	10
16	2023/10/06 09:25:58	/product/read/employee/daeju	demo-8102	13,252	[HTTP] /remote/sales/pickup/employee/daeju (경과 시간: 1,265ms)	96.57.95.16	7687102955386973910	0	4	9
17	2023/10/06 09:25:59	/product/save/division/pusan	demo-8101	12,522	[HTTP] /remote/account/write/division/kwangju (경과 시간: 1,036ms)	254.22.63.94	-5886187930318572769	0	3	9
18	2023/10/06 09:25:59	/account/read/unit/daeju	demo-8103	12,452	[HTTP] /remote/edu/save/dept/kwangju (경과 시간: 905ms)	25.14.74.7	-1176198797927439561	4	2	8
19	2023/10/06 09:26:00	/sales/pickup/division/daeju	demo-8104	11,994	[HTTP] /remote/edu/write/dept/daeju (경과 시간: 823ms)	64.5.10.93	-4237505319141287580	0	2	9
20	2023/10/06 09:26:00	/account/read/dept/seoul	demo-8101	11,520	[HTTP] /remote/order/write/unit/pusan (경과 시간: 701ms)	9.1.231.78	1280619473792807937	0	6	9
21	2023/10/06 09:26:00	/account/write/dept/pusan	demo-8103	11,380	[HTTP] /remote/product/write/dept/jeju (경과 시간: 1,102ms)	114.250.69.89	4540194209964704802	5	5	8
22	2023/10/06 09:26:00	/order/load/employee/seoul	demo-8104	11,268	[HTTP] /remote/account/kill/employee/daeju (경과 시간: 1,238ms)	228.76.247.169	-4887447133175400083	4	4	8
23	2023/10/06 09:26:01	/account/read/unit/jeju	demo-8100	10,925	[HTTP] /remote/account/remove/dept/kwangju (경과 시간: 989ms)	186.242.152.126	6555637030278391290	0	4	8
24	2023/10/06 09:26:01	/account/write/dept/kwangju	demo-8100	10,804	[HTTP] /remote/sales/kill/division/kwangju (경과 시간: 880ms)	205.96.146.41	1586163290816700232	4	4	7
25	2023/10/06 09:26:01	/product/create/unit/daeju	demo-8101	10,774	[HTTP] /remote/product/delete/dept/kwangju (경과 시간: 395ms)	195.57.135.171	5676527949830896043	4	3	8

## 1 에이전트 별 액티브 트랜잭션

전체 애플리케이션 서버의 목록과 진행 중인 트랜잭션의 총 개수를 표시합니다. 차트의 각 그래프에 마우스를 오버하면 트랜잭션 상태 속도를 표시합니다. 트랜잭션 상태의 속도는 Normal, Slow, Very Slow로 분류합니다.

## 2 에이전트 목록

애플리케이션 서버의 목록과 진행 중인 트랜잭션의 총 개수를 표시합니다. 해당 애플리케이션 서버를 선택하면 3 트랜잭션 목록에 세부 정보가 나타납니다.

## 3 트랜잭션 목록

진행 중인 트랜잭션의 URL, SQL, HTTP 호출 정보의 상세 내용을 표시합니다. Top 10 항목 탭을 선택하면 각 항목별 경과 시간이 높은 순으로 정렬된 표를 확인할 수 있습니다. 마찬가지로 트랜잭션 상태의 속도는 **Normal**, **Slow**, **Very Slow**로 분류합니다.

트랜잭션 목록에서 항목을 선택하면 선택한 트랜잭션에 대한 상세 정보를 확인할 수 있는 **액티브 트랜잭션** 창이 나타납니다.

액티브 트랜잭션
🗄️ ✕

**URL** GET /account/read/employee/daejun **Application** demo-8103 🔍 전체 새로고침 스크린 중지

**Client IP** 146.79.105.60

**Act Elapsed** 456 **FetchCount** 321546 **Mdepth** 0 **WclientId** -1840289476 **HttpMethod** GET

**Elapsed** 349089 **ThreadId** 648949541 **CPU Time** 100 **SQL Count** 621 **Act Red Time** 8000

**FetchTime** 3459 **Start Time** 2024/03/13 11:35:55 **Http Call Count** 1 **HttpHost** 0.0.0.79

**DB Connection** jdbc:mysql://localhost:3306,localhost:3310/fake **ThreadStat** TIMED\_WAITING

**Memory Alloc** 8982840 **SQL Time** 317632 **Http Call Time** 1064 **Txid** 8979113710728931233 **Mtid** 0

**Act Yellow Time** 3000 **Http Query** GET **User Id** -1840289476 **Service** /account/read/employee/daejun

**Mcaller** 0 **Time** 1710297705002

---

**쿼리(456ms):**

```
SELECT DISTINCT p.LastName, p.FirstName
FROM Person.Person AS p
JOIN HumanResources.Employee AS e
ON e.BusinessEntityID = p.BusinessEntityID WHERE $ IN
(SELECT Bonus
FROM Sales.SalesPerson AS sp
WHERE e.BusinessEntityID = sp.BusinessEntityID)
```

---

**콜 스택:**


```
java.base@17.0.9/java.lang.Thread.sleep(Native Method)
whatap.util.ThreadUtil.sleep(ThreadUtil.java:37)
app//jdbc.Control.exec(Control.java:23)
app//jdbc.FakePreparedStatement.executeQuery(FakePreparedStatement.java:67)
app//com.virtual.dao.SelectDAO.executeBig(SelectDAO.java:23)
app//com.virtual.web.Simula.execute2(Simula.java:165)
app//com.virtual.web.Simula.execute(Simula.java:157)
app//com.virtual.web.Simula$1.process(Simula.java:94)
app//com.virtual.web.VExec.doFilter(VExec.java:14)
app//org.apache.catalina.filters.FilterBase.doFilter(FilterBase.java:26)
app//com.virtual.web.Simula.run(Simula.java:124)
```

- 실행 중인 트랜잭션에 대한 상세 정보 및 외부 호출, 쿼리, 콜 스택 등을 확인할 수 있습니다.
  - **전체 새로고침**: 진행 중인 트랜잭션의 정보를 갱신할 수 있습니다.

- **스레드 중지**: 진행 중인 트랜잭션을 중지할 수 있습니다. 스레드 중지를 위해서는 **시크릿키**가 필요합니다. **시크릿키**에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.
- 이미 종료된 트랜잭션이라면 **트랜잭션 보기** 버튼을 표시됩니다. 버튼을 선택하면 **트랜잭션 정보** 창이 나타납니다. **트랜잭션 정보**에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

## 에이전트 목록 필터링


2 에이전트 목록에서 필터링 대상을 선택하거나 검색해 트랜잭션 목록에서 개별로 확인할 수 있습니다.

- 현재 활성화된 에이전트를 확인하려면 **활성화 에이전트만 보기** 체크박스를 선택하세요.
- **전체 에이전트** 목록 상자를 클릭하세요. 에이전트를 카테고리 별로 분류해 확인할 수 있습니다.
- 진행 중인 트랜잭션 정보를 에이전트 별로 확인하려면 에이전트를 선택한 다음 아래에 **적용** 버튼을 클릭하세요. 선택을 취소하려면  버튼을 선택하세요.


## 자동 갱신 설정


진행 중인 트랜잭션을 설정한 간격마다 자동 갱신할 수 있습니다. **자동 갱신(초)** 토글 버튼을 선택해 옵션을 활성화한 다음 오른쪽 텍스트 입력 상자에 갱신 간격을 입력하세요.

자동 갱신 설정을 비활성화하려면 **자동 갱신(초)** 토글 버튼을 다시 선택하세요.

- ⓘ • 시간은 초 단위로 갱신합니다.
- 자동 갱신 설정을 비활성화한 상태에서 트랜잭션 목록을 갱신하려면  버튼을 선택하세요.

## 트랜잭션 목록 필터링

3 트랜잭션 목록에서 URL 또는 IP 값을 검색해 필터링할 수 있습니다. 오른쪽 위에 **URL / IP** 텍스트 입력 상자에 필터링할 대상을 입력한 다음 엔터를 입력하거나  버튼을 선택하세요. 입력한 텍스트 값을 포함하는 트랜잭션 목록으로 필터링합니다.

필터링한 목록을 초기화하려면 오른쪽 위에 **URL / IP** 텍스트 입력 상자의 텍스트를 삭제한 다음 엔터를 입력하거나  버튼을 선택하세요.

ⓘ 액티브 트랜잭션 설정에 관한 자세한 내용은 [다음 문서](#)를 참조하세요.

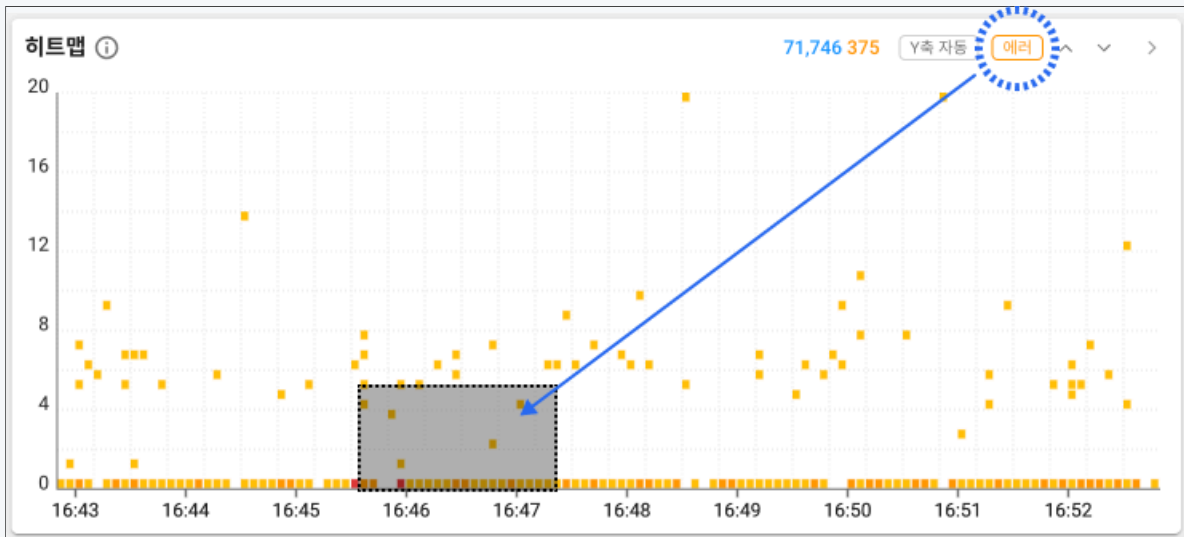


# 히트맵 - 트레이스 분석

애플리케이션 대시보드의 히트맵 위젯을 통해 트레이스 분석 창을 열어 트랜잭션 목록과 각 트랜잭션 하위의 스텝 정보를 한 번에 확인할 수 있습니다. 히트맵 위젯에서 원하는 영역을 드래그하세요. 선택한 영역의 트랜잭션 목록과 하위 스텝 정보를 포함하는 트레이스 분석 창이 나타납니다.

ⓘ 지난 2023년 7월 25일, 릴리스된 트레이스 분석 기능의 변경 사항을 확인하려면 [다음 문서](#)를 참조하세요.

✔ 에러가 발생한 트랜잭션 정보만 분류해 확인하려면 애플리케이션 대시보드의 히트맵 위젯의 오른쪽 위에 에러 버튼을 선택한 다음 차트 영역을 드래그하세요.



# 기본 화면 안내

트레이스 분석 1000건 조회되었습니다. ①

전체 ② 필터링 설정 TX 검색어를 입력하세요 ③

No	에이전트 명 (oname)	트랜잭션	경과 시간	시작 시간	종료 시간	HTTP 호출 ...	HTTP 호출 시간 (...)	SQL 시간	SQL 건수	DB 연결 시간 (...)	SQL 패지 건수	에러 메시지	클라이언트 IP	WClientID	멀티 트랜잭...
1	demo-8100	/edu/load/divis... <b>A</b>	13,919	23/10/06 08:27:15.989	23/10/06 08:27:29.908	9	13,154	714	6	20	1,918		246.159.226.226	6299760544220	
2	demo-8103	/order/write/de... <b>A</b>	13,271	23/10/06 08:27:18.842	23/10/06 08:27:32.113	10	12,285	790	5	192	0		53.128.58.29	-6966890863916	
3	demo-8105	/edu/kill/divisio... <b>A</b>	12,351	23/10/06 08:27:16.740	23/10/06 08:27:29.091	10	11,836	495	2	3	143		163.234.79.106	79916051556534	
4	demo-8100	/order/write/un... <b>A</b>	12,233	23/10/06 08:27:18.904	23/10/06 08:27:31.137	9	11,635	398	8	184	92		113.217.26.61	18123471058131	
5	demo-8104	/order/pickup/d... <b>A</b>	11,993	23/10/06 08:27:14.781	23/10/06 08:27:26.774	10	11,921	67	3	3	0		165.239.32.206	-8541053171176	
6	demo-8102	/order/remove/... <b>A</b>	11,648	23/10/06 08:27:22.493	23/10/06 08:27:34.141	9	11,365	263	4	4	610		101.243.208.67	1849935236386	
7	demo-8105	/sales/read/uni... <b>A</b>	11,505	23/10/06 08:27:19.740	23/10/06 08:27:31.245	9	11,034	265	4	203	0		130.135.32.247	6289584611395	
8	demo-8101	/order/pickup/e... <b>A</b>	11,495	23/10/06 08:27:18.073	23/10/06 08:27:29.568	9	10,858	426	4	196	1,266		160.106.28.10	-5449598593135	
9	demo-8104	/order/read/em... <b>A</b>	11,476	23/10/06 08:27:14.258	23/10/06 08:27:25.734	8	10,572	838	7	43	1,645		62.26.92.37	11623328420977	
10	demo-8100	/edu/write/divi... <b>A</b>	11,204	23/10/06 08:27:14.097	23/10/06 08:27:25.301	9	10,944	154	4	97	413		161.42.184.55	-6015131889945	

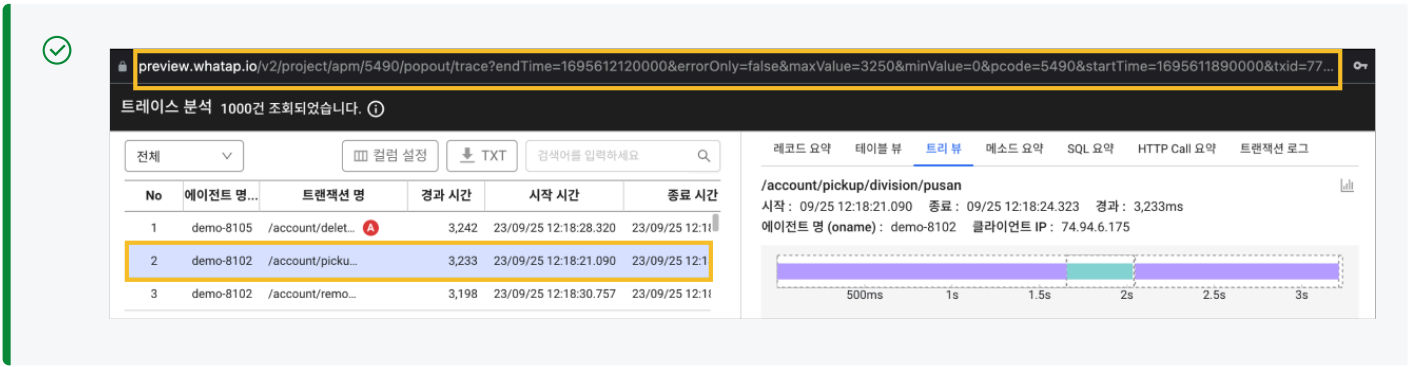
- ① **트랜잭션 조회 건수**: 히트맵 위젯에서 드래그한 영역의 트랜잭션 목록의 개수를 확인할 수 있습니다. 최대 1000건까지 조회할 수 있습니다.
- ② **트랜잭션 분류**: **A** **액티브 스택**과 **M** **멀티 트랜잭션**, 에러 항목을 구분하여 목록을 확인할 수 있습니다.
- ③ **트랜잭션 검색**: 트랜잭션 명과 에이전트 명, 에러 메시지를 기준으로 트랜잭션 목록을 검색할 수 있습니다.
- ④ **목록 정렬하기**: 테이블의 헤더 컬럼 항목을 클릭하면 선택한 항목을 기준으로 트랜잭션 목록을 정렬할 수 있습니다.
- 에러가 발생한 트랜잭션 항목은 목록에서 빨간색으로 표시됩니다.

트레이스 분석 92건 조회되었습니다. ①

전체 필터링 설정 TXT 검색어를 입력하세요

No	에이전트 명 (oname)	트랜잭션	경과 시간	시작 시간	종료 시간	HTTP 호출 ...	HTTP 호출 시간 (...)	SQL 시간	SQL 건수	DB 연결 시간 (...)	SQL 패지 건수	에러 메시지	클라이언트 IP	WClientID	멀티 트랜잭...
1	demo-8102	/account/delet... <b>A</b>	9,637	23/10/06 08:45:29.188	23/10/06 08:45:38.825	7	8,802	648	3	184	0	Sql Exception	11.105.116.110	-5540024635227	
2	demo-8104	/edu/read/unit/... <b>A</b>	8,918	23/10/06 08:44:21.258	23/10/06 08:44:30.176	7	8,867	45	1	1	0	Sql Exception	53.199.161.32	22360232179385	
3	demo-8101	/product/create... <b>A</b>	8,384	23/10/06 08:45:57.515	23/10/06 08:46:05.899	7	8,167	193	3	4	709	Sql Exception	5.217.123.24	7443623688153	
4	demo-8102	/account/remo... <b>A</b>	7,810	23/10/06 08:45:48.766	23/10/06 08:45:56.576	6	7,145	651	2	2	220	Sql Exception	246.254.84.54	1849935236386	
5	demo-8103	/order/delete/u... <b>A</b>	7,762	23/10/06 08:45:16.257	23/10/06 08:45:24.019	6	7,290	458	2	3	0	Sql Exception	131.246.229.18	7124226905278	
6	demo-8104	/product/remov... <b>A</b>	7,289	23/10/06 08:46:19.886	23/10/06 08:46:27.175	6	7,281	3	1	1	0	Sql Exception	68.188.222.114	-1064563861026	

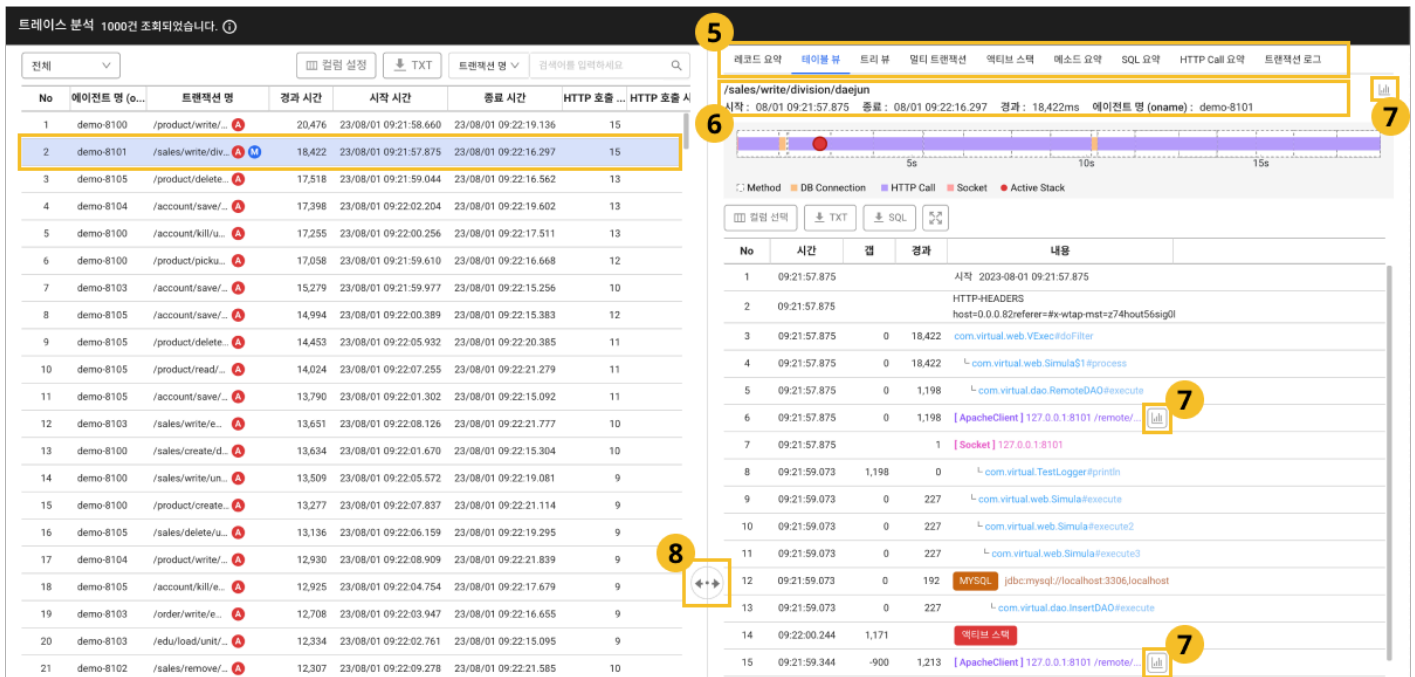
- ☑ 테이블에서 자주 보길 원하는 컬럼 항목을 앞으로 배치하거나 추가, 숨기기할 수 있습니다. 자세한 내용은 [다음 문서](#)를 참조하세요.
- 히트맵 위젯에서 다른 영역을 드래그하면 새로운 트레이스 분석 창을 불러올 수 있습니다. 기존에 열린 트레이스 분석 창과 비교해 확인할 수 있습니다.
- 현재 사용자가 보고 있는 트랜잭션 항목을 다른 사용자에게 공유할 수 있습니다. 트랜잭션 항목을 선택한 상태에서 트레이스 분석 창의 URL 주소를 복사해 다른 사용자에게 전달하세요. URL 주소를 전달 받은 사용자는 브라우저 주소 표시줄에 URL 주소를 입력하면 같은 항목을 빠르게 확인할 수 있습니다.





## 트레이스 분석하기




트랜잭션 목록에서 상세 내용을 확인할 항목을 선택하세요. 화면 오른쪽에 선택한 트랜잭션에 대한 상세 분석 내용을 표시합니다. 이 상태에서 다른 트랜잭션 항목을 선택하면 선택한 트랜잭션 항목에 대한 상세 분석 내용으로 변경됩니다.

와탭은 트랜잭션의 성능을 분석하기 위해 이름과 클라이언트 정보 등의 속성, 트랜잭션의 처리 성능 그리고 각 구간별 상세 수행 이력을 수집하고 보여줍니다.




- 5 영역에서는 스텝 정보를 다양한 형식으로 확인할 수 있습니다. 원하는 형식의 탭을 선택하세요.

- 6 영역에서는 트랜잭션에 대한 기본 정보를 확인할 수 있습니다. 왼쪽 목록의 컬럼 항목과 일치하는 정보입니다.
- 7 : 해당 URL 또는 각 수행 구간의 통계 데이터 창이 나타납니다. 통계 데이터 창에서 상세 버튼을 선택하면 통계 메뉴로 이동합니다.
- 8  왼쪽 또는 오른쪽으로 드래그해서 화면의 크기를 조절할 수 있습니다. 더 많은 정보를 보길 원할 때 이용하면 유용합니다.

-  통계 메뉴에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.
  -  또는  버튼을 클릭하면 트랜잭션 목록 또는 상세 분석 화면을 전체 크기로 변경할 수 있습니다.

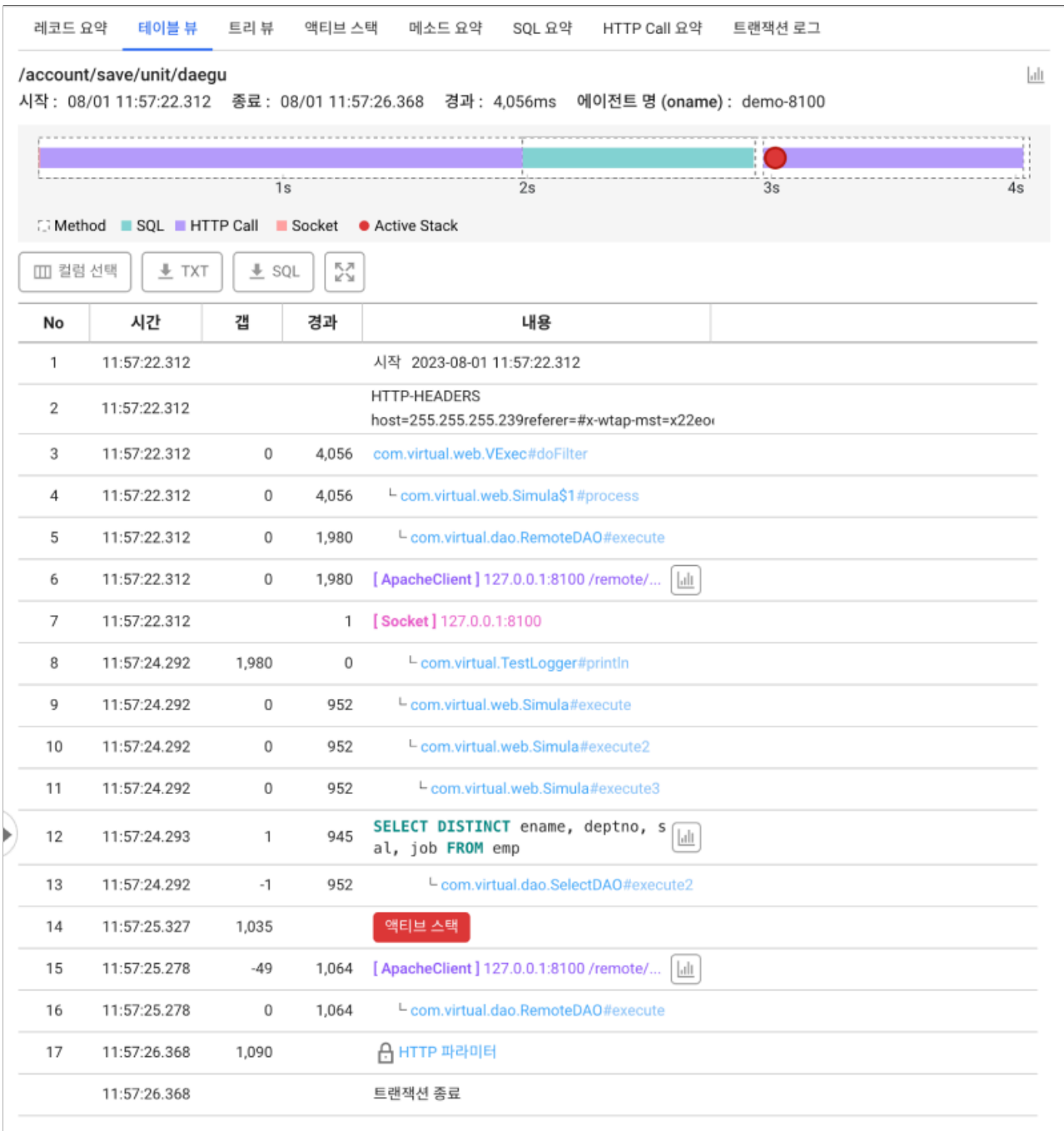
수집하는 스텝(Step)의 종류는 다음과 같습니다. 다이어그램과 텍스트 색상을 구분해 표시합니다.

- DB Connection 스텝 START-END  
RDB에 대한 연결에 대한 성능을 포함합니다. 스텝 정보에는 이름, 응답시간, 에러를 포함합니다.
- SQL 스텝 START-END  
JDBC SQL에 대한 성능을 포함합니다. 스텝 정보에는 연결 정보, SQL문, 에러가 포함되어 있습니다.
- HTTP Call 스텝 START-END  
외부 http 서비스 호출에 대한 성능을 포함합니다. 스텝 정보에는 url, host, port, 응답시간, 에러가 포함됩니다.
- Message 스텝 ADD  
트레이스를 수집하는 과정에서 비정형적인 모든 구간에 대한 이력을 수집할 때 메시지 스텝을 사용합니다.
- Socket 스텝 ADD  
Socket 오픈을 표현하는 스텝입니다.
- Method 스텝 START-END  
메소드 응답시간을 추적합니다.
- Active Stack 스텝 START-END  
액티브 스텝에 대한 정보를 포함합니다. 별도 스레드가 생성하여 트레이스에 추가하는 방식으로 수집합니다.

-  와탭 모니터링 서비스에서 스텝(Step)은 스패ن(Span)과 같은 뜻으로 사용됩니다.
  - 지난 2023년 7월 25일, 릴리스된 [트레이스 분석](#) 기능의 변경 사항을 확인하려면 [다음 문서](#)를 참조하세요.

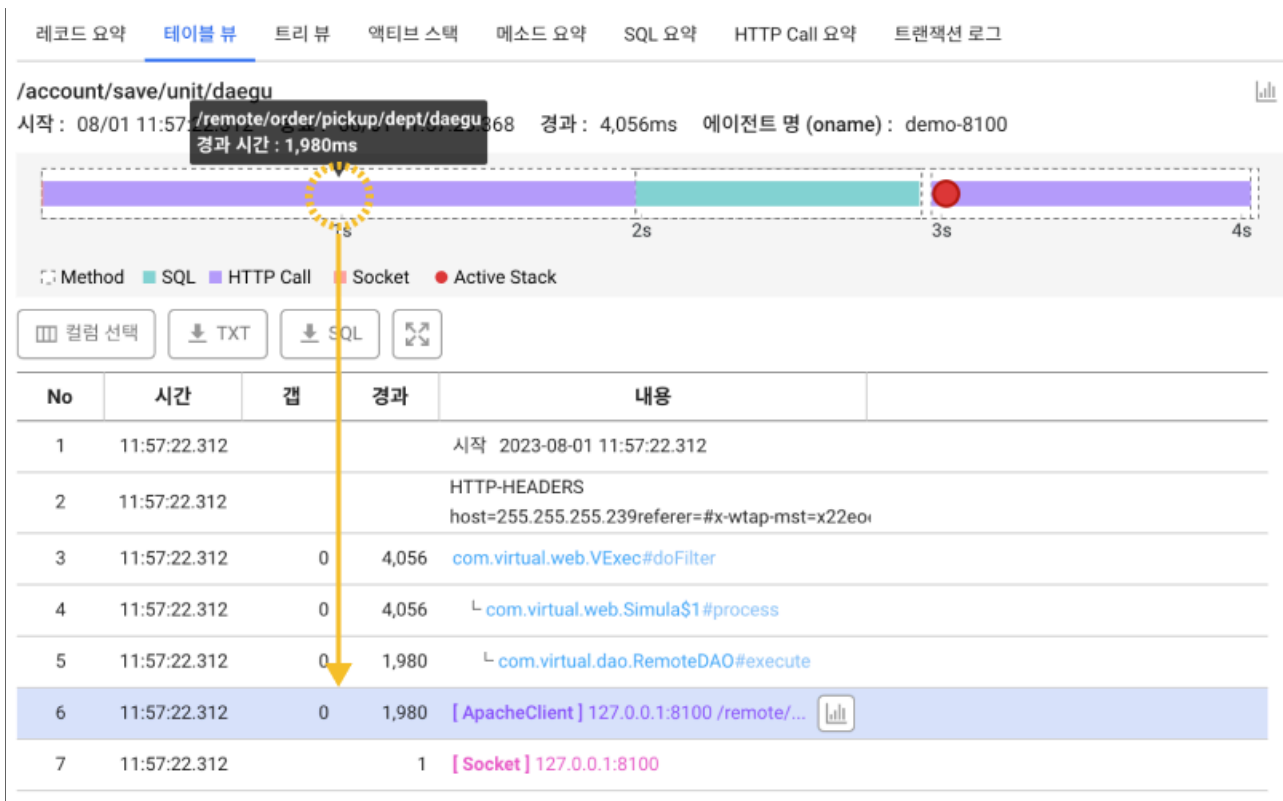
## 테이블 뷰

테이블 뷰 탭에서는 트랜잭션의 수행 과정을 시간의 순서대로 확인할 수 있습니다.



- 트랜잭션을 수행 구간별로 분류해 다이어그램을 통해 확인할 수 있습니다. 전체 경과 시간 중 각 구간별 소요된 시간, 가장 오래 소요된 구간을 빠르게 파악할 수 있습니다.

- 다이어그램의 각 구간을 선택하면 해당 스텝이 위치한 테이블 목록으로 이동합니다.



- 빨간색 원(●)이 위치한 영역은 액티브 스택이 수집된 순간입니다. 빨간색 원을 선택하면 **액티브 스택** 버튼이 위치한 테이블 목록으로 이동할 수 있습니다. **액티브 스택** 버튼을 선택하면 해당 구간 동안 수행된 스텝 정보를 확인할 수 있습니다.

ⓘ ○ 트랜잭션 목록에서 **A** 아이콘이 표시된 항목에서 확인할 수 있습니다.  
○ 액티브 스택에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

- **☰ 컬럼 선택**: 테이블 목록에 컬럼으로 메모리 누적 정보 및 CPU 누적 정보를 추가하거나 감출 수 있습니다. 컬럼 정보는 다음과 같습니다.
  - **No**: 스텝의 발생 순서
  - **시간**: 각 스텝의 시작 시각
  - **갭**: 해당 스텝의 시작 시각부터 다음 스텝으로 넘어가기까지 대기 시간, 외부 요인으로 지연될 경우 경과 시간과 차이가 있을 수 있습니다.
  - **경과**: 각 메소드 시작부터 종료까지 총 소요 시간
  - **내용**: 해당 스텝의 세부 수행 내용

- **↓ TXT**: 트랜잭션 기본 정보 및 구간별 수행 정보를 txt 형식의 파일로 다운로드할 수 있습니다.
- **↓ SQL**: 트랜잭션 기본 정보 및 SQL 수행 정보를 txt 형식의 파일로 다운로드할 수 있습니다.
- **🔍**: 트랜잭션 요약 정보와 다이어그램을 감추고 테이블 목록만 확인할 수 있습니다. **⌵** 버튼을 선택하면 감춰진 정보를 다시 표시합니다. 테이블 목록이 긴 경우 이 기능을 이용하면 유용합니다.
- SQL 스텝을 선택하면 파라미터를 조회할 수 있는 **SQL** 창이 나타납니다. 자세한 내용은 [다음 문서](#)를 참조하세요.

## 레코드 요약

[레코드 요약](#)
[테이블 뷰](#)
[트리 뷰](#)
[메소드 요약](#)
[SQL 요약](#)
[HTTP Call 요약](#)
[트랜잭션 로그](#)

**/account/remove/dept/pusan**

<b>에이전트 명 (oname)</b> : demo-8101	에이전트 ID (oid) : -877561626
에이전트 그룹 명 : demo-okind-1	에이전트 그룹 ID : -628198688
에이전트 서버 명 : node-1	에이전트 서버 ID : 334634079
프로젝트 코드 : 5490	

---

<b>트랜잭션 ID</b> : <u>-3538973756714763020</u>	시작 시간 : 23/10/06 08:48:11.706
종료 시간 : 23/10/06 08:48:14.585	<b>경과 시간</b> : 2,879ms
HTTP 메소드 : GET	HTTP 호출 시간 : 2,409ms
HTTP 호출 건수 : 2	DB 연결 시간 : 139ms
SQL 시간 : 324ms	SQL 건수 : 2
SQL 패치 시간 : 6ms	SQL 패치 건수 : 647

---

CPU 사용 시간 : 2ms	메모리 할당량 : 197,232byte
-----------------	-----------------------

---

클라이언트 IP : 168.196.231.239	도메인 : 255.255.255.159
WClientID : 4545076460797978934	국가 : BR
도시 : Acailandia	운영체제 : Windows
클라이언트 타입 : IE	클라이언트 명 : Other
상태 : 200	Referer : http://www.test.com/account/load/employee/seoul

---

원본 URL/트랜잭션 : /account/remove/dept/pusan

유저 에이전트 : Mozilla/5.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4325)

트랜잭션의 기본 정보를 확인하려면 **레코드 요약** 탭을 선택하세요. 트랜잭션의 기본 정보 및 해당 트랜잭션이 수행된 에이전트 관련



정보, 메모리 할당 사용량, 클라이언트 관련 정보를 확인할 수 있습니다.

## 트랜잭션 속성

분류	속성	설명
에이전트	<b>에이전트 명(oname)</b>	에이전트 오브젝트 이름
	에이전트 ID (oid)	에이전트 오브젝트 아이디
	에이전트 그룹 명	okind 옵션으로 설정된 에이전트 그룹 이름
	에이전트 그룹 ID	에이전트 그룹 아이디
	에이전트 서버 명	ondoe 옵션으로 설정한 에이전트 서버 이름
	에이전트 서버 ID	에이전트 서버 아이디
	프로젝트 코드	와탭 모니터링 서비스에 등록된 프로젝트의 식별 코드
트랜잭션	<b>트랜잭션 ID</b>	트랜잭션 아이디
	시작 시간	트랜잭션 시작 시각
	종료 시간	트랜잭션 종료 시각
	<b>경과 시간</b>	트랜잭션 수행 시간
	HTTP 메소드	HTTP 메소드 - GET, POST, PUT, HEAD 등
	HTTP 호출 시간	외부 Http Call 시간
	HTTP 호출 건수	외부 HTTP Call 건수
	DB 연결 시간	DB와 연결된 시간
	SQL 시간	SQL 수행 시간

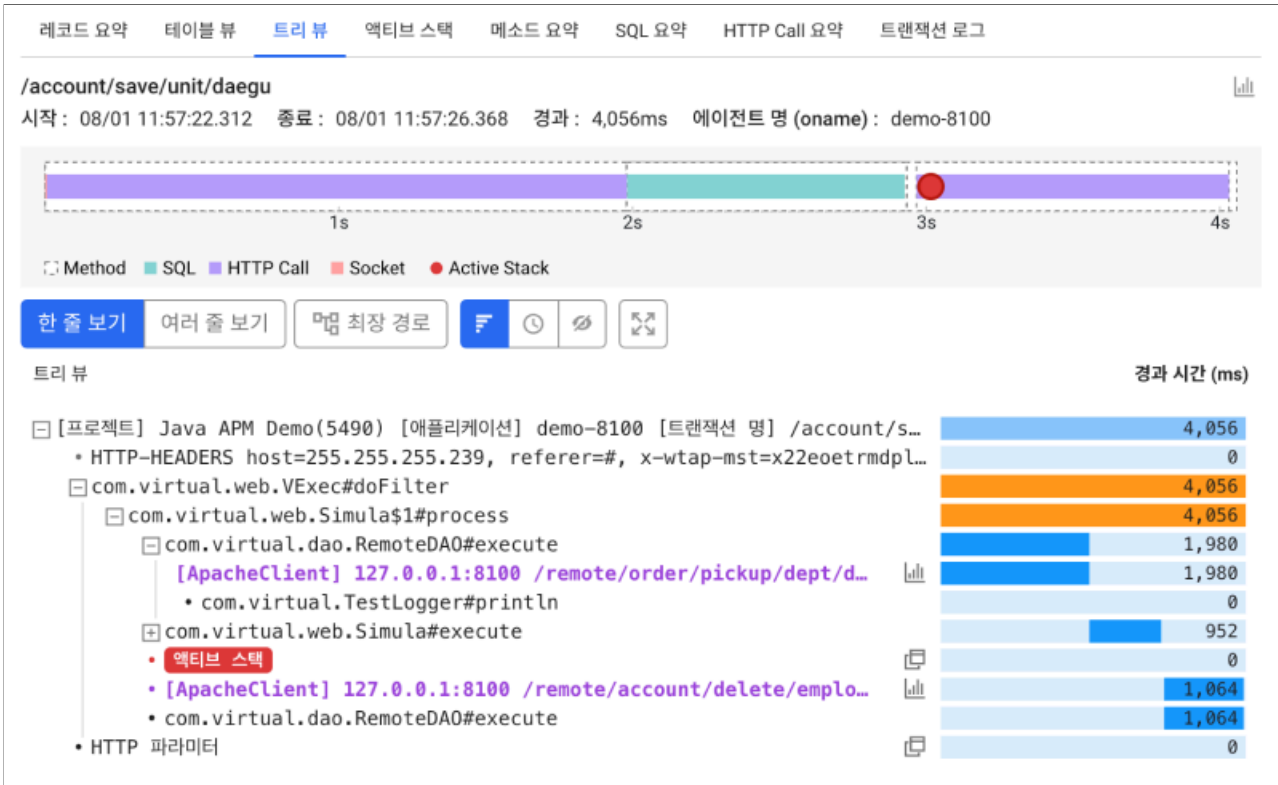
분류	속성	설명
	SQL 건수	SQL 수행 건수
	SQL 패치 시간	레코드를 조회하는 데 걸린 시간 (중첩이 발생하거나 타 로직이 포함될 수 있음)
	SQL 패치 건수	SQL에서 데이터를 조회한 레코드 건수
멀티 트랜잭션	CallerUrl	부모 트랜잭션의 트랜잭션 주소
	CallerPcode	부모 트랜잭션이 수집된 프로젝트 코드
	CallerOid	부모 트랜잭션의 에이전트 오브젝트 아이디
	CallerOName	부모 트랜잭션의 에이전트 이름
	CallerOkind	부모 트랜잭션의 에이전트 그룹 ID
	CallerOkindName	부모 트랜잭션의 에이전트 그룹 이름
자원	CPU 사용 시간	트랜잭션이 사용한 CPU 사용량
	메모리 할당량	트랜잭션이 사용한 메모리 사용량
클라이언트	클라이언트 IP	클라이언트의 IP 주소
	도메인	클라이언트가 접속한 IP 주소에 지정된 인터넷 주소
	WClientID	클라이언트의 ID
	국가	클라이언트의 국가 정보
	도시	클라이언트의 국가 내 도시 정보
	운영체제	브라우저가 실행되는 운영 체제 환경
	클라이언트 타입	클라이언트가 이용한 브라우저 종류

분류	속성	설명
	클라이언트 명	클라이언트가 이용한 기기 이름
	상태	클라이언트의 요청에 대한 HTTP 상태 코드
	Referer	클라이언트가 브라우저에서 이전에 이용한 페이지 주소 또는 유입 경로
	원본 URL/트랜잭션	트랜잭션 이름과 원본 URL의 조합 데이터 원본 URL이 있을 경우 원본 URL 표시, 원본 URL이 없을 경우 트랜잭션 표시
	유저 에이전트	클라이언트 관련 정보를 추출하는데 이용하는 브라우저 정보

- ⓘ 애플리케이션 종류나 설정, 스텝의 종류에 따라 수집하는 정보는 달라질 수 있습니다.
- 와탭은 클라이언트와 관련한 정보를 기본 저장합니다. 사용자 데이터 수집과 관련한 에이전트 설정에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

## 트리 뷰

트랜잭션 수행 과정을 트리 형식으로 확인하려면 **트리뷰** 탭을 선택하세요. 각 트랜잭션과 그에 속한 트race의 세부 정보, 트race의 시작 시간 및 소요 시간, 호출 관계를 확인할 수 있습니다. 다이어그램의 각 구간을 선택하면 해당 스텝이 위치한 리뷰로 이동합니다.




- **한 줄 보기:** 각 구간 별 수행 정보에 표시된 텍스트를 한 줄로 표시해 트리 형식을 간격하게 정리할 수 있습니다.
- **여러 줄 보기:** 각 구간 별 수행 정보에 표시된 텍스트를 줄바꿈해 모두 표시합니다.
- **최장 경로:** 가장 긴 경로로 이동할 수 있습니다.
- **F 시간바 표시:** 경과 시간을 막대 형식의 차트로 표시합니다.
- **🕒 시간 표시:** 각 구간별 타임 스탬프, 갭, 경과 시간을 텍스트 형식으로 표시합니다.
  - 8초 이상: **초과 지연** 상태로 **빨간색**으로 표현합니다.
  - 3초 이상 8초 미만: **지연** 상태로 **주황색**으로 표현합니다.
  - 3초 미만: **정상** 상태로 **파란색**으로 표현합니다.
- **🔍 시간 숨기기:** 시간 정보를 숨깁니다.
- **🔄 트랜잭션 요약 정보와 다이어그램을 감추고 트리뷰만 확인할 수 있습니다.** **🔍** 버튼을 선택하면 감춰진 정보를 다시 표시합니다. 트리뷰 목록이 긴 경우 이 기능을 이용하면 유용합니다.
- **🔍 SQL 변수와 HTTP 쿼리를 조회할 수 있는 창이 나타납니다.** 자세한 내용은 [다음 문서](#)를 참조하세요.

- ① 시작 및 소요 시간의 경우 트랜잭션 호출 환경에 따라 발생하는 시차를 상위 트랜잭션 내 트레이스와 매핑을 통해 보정하여 표현하기 때문에 실제 수집된 시간 데이터와 차이가 발생할 수 있습니다.

## 액티브 스택

레코드 요약   테이블 뷰   트리 뷰   **액티브 스택**   메소드 요약   SQL 요약   HTTP Call 요약   트랜잭션 로그

/account/save/unit/daegu 

시작: 08/01 11:57:22.312   종료: 08/01 11:57:26.368   경과: 4,056ms   에이전트 명 (oname): demo-8100

액티브 스택 (1)

No	시간	
14	11:57:25.327	<pre> java.base@17.0.7/sun.nio.ch.SocketDispatcher.read0(Native Method) java.base@17.0.7/sun.nio.ch.SocketDispatcher.read(SocketDispatcher.java:47) java.base@17.0.7/sun.nio.ch.NioSocketImpl.tryRead(NioSocketImpl.java:261) java.base@17.0.7/sun.nio.ch.NioSocketImpl.implRead(NioSocketImpl.java:312) java.base@17.0.7/sun.nio.ch.NioSocketImpl.read(NioSocketImpl.java:350) java.base@17.0.7/sun.nio.ch.NioSocketImpl\$1.read(NioSocketImpl.java:803) java.base@17.0.7/java.net.Socket\$SocketInputStream.read(Socket.java:966) app//org.apache.http.impl.io.SessionInputBufferImpl.streamRead(SessionInputBufferImpl.java:139) app//org.apache.http.impl.io.SessionInputBufferImpl.fillBuffer(SessionInputBufferImpl.java:155) app//org.apache.http.impl.io.SessionInputBufferImpl.readLine(SessionInputBufferImpl.java:284) app//org.apache.http.impl.conn.DefaultHttpResponseParser.parseHead(DefaultHttpResponseParser.java:140) app//org.apache.http.impl.conn.DefaultHttpResponseParser.parseHead(DefaultHttpResponseParser.java:57) app//org.apache.http.impl.io.AbstractMessageParser.parse(AbstractMessageParser.java:261) </pre>

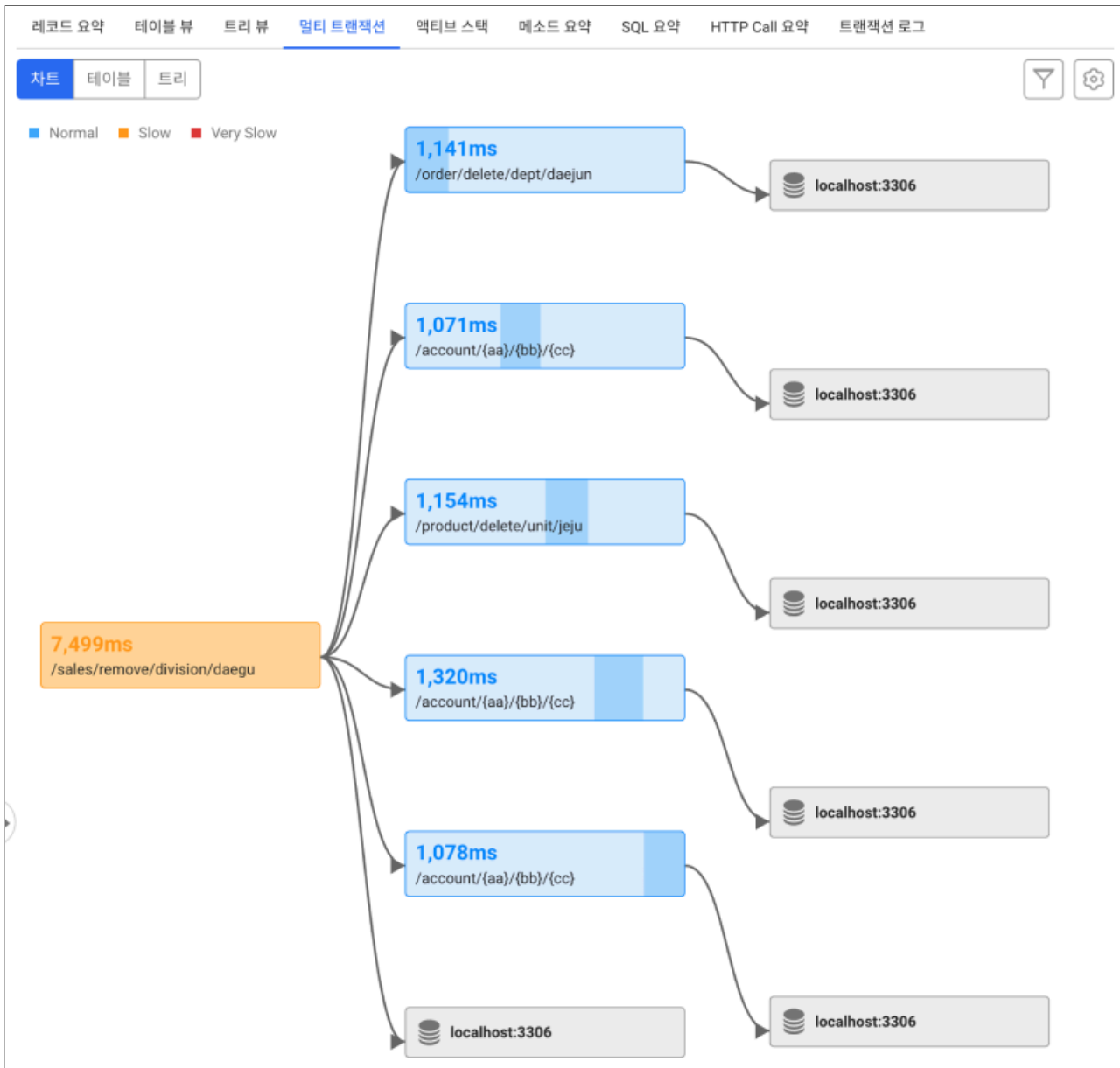
액티브 스택을 수집한 구간과 시간, 관련한 클래스, 메소드 정보를 확인하려면 **액티브 스택** 탭을 선택하세요.


- ① • 트랜잭션 목록에서 **A** 아이콘이 표시된 항목에서 확인할 수 있습니다.
  - 액티브 스택에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

## 멀티 트랜잭션

멀티 트랜잭션은 다른 에이전트나 프로젝트와의 연관된 트랜잭션을 의미합니다. [멀티 트랜잭션](#) 탭에서는 와탭 모니터링 서비스에 등록된 애플리케이션 간의 호출 관계를 확인할 수 있습니다.

- ✔ 멀티 트랜잭션을 추적하려면 [관리](#) > [에이전트 설정](#) 메뉴에서 `mtrace_enabled` 옵션을 `true` 로 설정하세요. 에이전트 설정에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.



- **차트**: 각 트랜잭션의 호출 관계를 플로우 차트 형식으로 제공합니다. 트랜잭션 노드를 선택하면 해당 트랜잭션 노드에 대한 트레이스 분석 정보를 확인할 수 있습니다. 차트 모드에서는 마우스를 이용해 원하는 위치로 이동하거나 스크롤을 통해서 확대, 축소할 수 있습니다.
  - **차트 뷰 설정**: 차트에 표시할 요소를 표시하거나 숨길 수 있습니다.
- **테이블**: 테이블 형식으로 멀티 트랜잭션 내에 포함된 각 트랜잭션 별 정보를 확인할 수 있습니다.  컬럼 선택 아이콘을 선택해

테이블 헤더 컬럼을 편집할 수 있습니다. 각 트랜잭션 항목을 선택하면 트레이스 분석 정보를 확인할 수 있습니다.

- **트리**: 트리 형식으로 트랜잭션 간의 호출 관계를 파악할 수 있습니다. 관련한 부가 기능은 **트리뷰** 탭의 기능과 같습니다.
- **프로젝트 선택**: 차트에 표시될 프로젝트를 선택하거나 해제할 수 있습니다.

- ⓘ • 트랜잭션 목록에서 **M** 아이콘이 표시된 항목에서 확인할 수 있습니다.
  - 멀티 트랜잭션에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

## 메소드 요약

레코드 요약    테이블 뷰    트리 뷰    액티브 스택 <b>메소드 요약</b> SQL 요약    HTTP Call 요약    트랜잭션 로그							
/account/save/unit/daegu <span style="float: right;">📊</span>							
No	클래스	메소드	파라미터/리턴	건수	합계 (ms)	최대 (ms)	
1	com.virtual.dao.Remot...	execute	()V	2	3,044	1,980	
2	com.virtual.web.VExec	doFilter	(Ljavax/servlet/Servlet...	1	4,056	4,056	
3	com.virtual.web.Simul...	process	()V	1	4,056	4,056	
4	com.virtual.TestLogger	println	(Ljava/lang/String;)V	1	0	0	
5	com.virtual.web.Simula	execute	()V	1	952	952	
6	com.virtual.web.Simula	execute2	()V	1	952	952	
7	com.virtual.web.Simula	execute3	()V	1	952	952	
8	com.virtual.dao.Select...	execute2	()V	1	952	952	

메소드 정보만을 확인하려면 **메소드 요약** 탭을 선택하세요. 에이전트에 추적이 설정된 메소드 이름과 소요 시간을 표시합니다. 불필요한 로직이 반복 실행되는 경우, 낮은 건수에 비해 실행 시간이 오래 걸리는 경우를 파악해 메소드 로직 개선을 위한 분석 정보로 활용할 수 있습니다.

- ⓘ 메소드(method)와 관련한 에이전트 설정 옵션에 대한 자세한 내용은 [다음 문서](#)를 참조하세요. 이용 중인 상품에 따라 메소드 추적 지원 여부는 다를 수 있습니다.



## SQL 요약

레코드 요약   테이블 뷰   트리 뷰   액티브 스택   메소드 요약 <b>SQL 요약</b> HTTP Call 요약   트랜잭션 로그						
/account/save/unit/daegu 						
No	데이터베이스	SQL	건수	합계 (ms)	최대 (ms)	
1	jdbc:mysql://localhost:...	SELECT DISTINCT ena...	1	945	945	

SQL문에 대한 정보를 확인하려면 [SQL 요약](#) 탭을 선택하세요. 불필요하게 반복 실행되는 경우, 낮은 건수에 비해 실행 시간이 오래 걸리는 경우를 파악해 쿼리 성능 개선을 위한 분석 정보로 활용할 수 있습니다.

## HTTP Call 요약

레코드 요약   테이블 뷰   트리 뷰   액티브 스택   메소드 요약   SQL 요약 <b>HTTP Call 요약</b> 트랜잭션 로그						
/account/save/unit/daegu 						
No	원격 서버	URL	건수	합계 (ms)	최대 (ms)	
1	127.0.0.1:8100	/remote/order/pickup/...	1	1,980	1,980	
2	127.0.0.1:8100	/remote/account/delet...	1	1,064	1,064	

HTTP 호출의 호출 건수, 합계 시간, 평균 시간 등을 확인하려면 [HTTP Call 요약](#) 탭을 선택하세요. 불필요한 외부 호출이 반복 실행되는 경우, 낮은 건수에 비해 실행 시간이 오래 걸리는 경우를 파악해 트랜잭션 지연 요인이 외부인지 내부인지를 파악하는 분석 정보로 활용할 수 있습니다.

## 트랜잭션 로그

트랜잭션과 관련한 로그 정보를 확인하려면 [트랜잭션 로그](#) 탭을 선택하세요.

레코드 요약   테이블 뷰   트리 뷰   액티브 스택   메소드 요약   SQL 요약   HTTP Call 요약   **트랜잭션 로그**

/account/save/unit/daegu

키워드(를) 입력해주세요 🔍 🏠 ⚙️

▶	oname	타임스탬프	로그
▶	demo-8100	2023-08-01 11:57:24.293	<a href="#">@txid</a> 5501194250256405697 <a href="#">pcode</a> 5490 <a href="#">oname</a> demo-8100 <a href="#">onodeName</a> node-0 http://127.0.0.1:8100/remote/order/pickup/dept/daegu status=200
▶	demo-8100	2023-08-01 11:57:25.245	<a href="#">@txid</a> 5501194250256405697 <a href="#">pcode</a> 5490 <a href="#">oname</a> demo-8100 <a href="#">onodeName</a> node-0 select distinct ename, deptno, sal, job from emp
▶	demo-8100	2023-08-01 11:57:25.279	<a href="#">@txid</a> 5501194250256405697 <a href="#">pcode</a> 5490 <a href="#">oname</a> demo-8100 <a href="#">onodeName</a> node-0 select productmodelid, name
▶	demo-8100	2023-08-01 11:57:26.342	<a href="#">@txid</a> 5501194250256405697 <a href="#">pcode</a> 5490 <a href="#">oname</a> demo-8100 <a href="#">onodeName</a> node-0 http://127.0.0.1:8100/remote/account/delete/employee/pusan status=20
▶	demo-8100	2023-08-01 11:57:26.369	<a href="#">@txid</a> 5501194250256405697 <a href="#">pcode</a> 5490 <a href="#">oname</a> demo-8100 <a href="#">onodeName</a> node-0 select ename, sal+1000 from emp

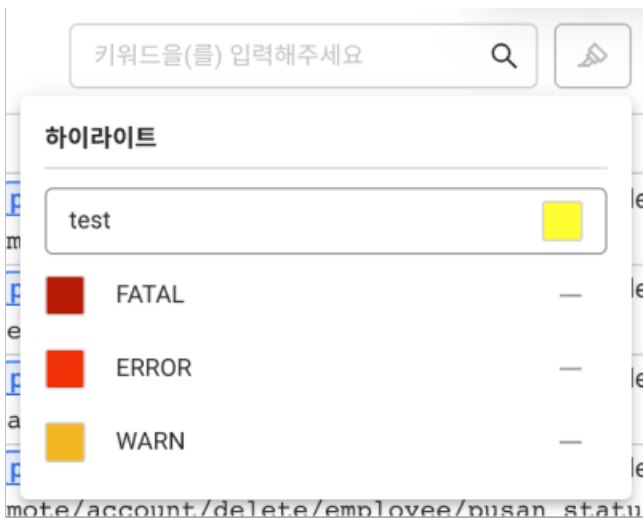
- **키워드 검색:** 키워드 검색란에 검색하려는 텍스트를 입력한 다음 엔터 키를 입력하거나 🔍 버튼을 선택하세요. 입력한 텍스트와 일치하는 키워드를 하이라이트 표시합니다.
- 🏠 **키워드 하이라이트:** 키워드와 색상을 설정하면 자동으로 로그 목록에서 키워드와 일치하는 텍스트를 하이라이트 표시합니다.
- ⚙️ **테이블 설정:** 로그 목록 테이블의 헤더 컬럼을 설정할 수 있습니다.

ⓘ • **트랜잭션 로그** 탭을 활성화하려면 로그와 관련한 에이전트 설정 옵션을 적용해야 합니다. 로그 설정에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

- **트랜잭션 로그** 탭은 **로그 조회** 권한이 가진 멤버만 진입할 수 있습니다. 멤버 권한에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

## 키워드 하이라이트 설정하기

1. 하이라이트 표시할 키워드를 추가하려면 🏠 버튼을 선택하세요.



2. 원하는 키워드를 입력하세요.
3. 색상 아이콘을 선택한 다음 원하는 색상을 선택하세요.
4. 엔터를 입력하세요.

추가한 키워드가 목록에 생성됩니다.

## 키워드 하이라이트 삭제하기

하이라이트 목록에서 이용하지 않는 항목을 삭제하려면 — 버튼을 선택하세요.

## 테이블 컬럼 표시하거나 숨기기

1. 테이블 오른쪽 위에 ⚙️ 버튼을 선택하세요.
2. 드롭다운 메뉴에서 **컬럼 설정**을 선택하세요.
3. **컬럼 설정** 창이 나타나면 **컬럼 추가** 섹션에서 원하는 항목을 선택하거나 선택을 해제하세요.
4. 모든 설정을 완료한 다음 **확인** 버튼을 선택하세요.

설정된 내용이 테이블에 적용됩니다.

## 테이블 컬럼 순서 바꾸기

1. 테이블 오른쪽 위에 ⚙️ 버튼을 선택하세요.
2. 드롭다운 메뉴에서 **컬럼 설정**을 선택하세요.

3. 컬럼 설정 창이 나타나면 컬럼 순서 설정 섹션에서 원하는 항목을 드래그해서 순서를 변경하세요.
4. 모든 설정을 완료한 다음 확인 버튼을 선택하세요.

설정된 내용이 테이블에 적용됩니다.

## 테이블 콘텐츠 표시 설정

1. 테이블 오른쪽 위에 ⚙️ 버튼을 선택하세요.
2. 드롭다운 메뉴에서 테이블 콘텐츠 설정을 선택하세요.
3. 다음 옵션을 설정하세요.
  - 콘텐츠 표시 여부: 테이블 목록에 표시하는 콘텐츠 표시 여부를 선택하세요.
  - 태그 관리: 테이블 목록에 표시하고 싶은 태그를 선택할 수 있습니다. 보이지 않도록 설정하거나 순서를 변경할 수 있습니다.
4. 모든 설정을 완료한 다음 확인 버튼을 선택하세요.

설정된 내용이 테이블에 적용됩니다.

# 부가 기능

## HTTP 파라미터 조회

테이블 뷰 탭에서 해당 트랜잭션의 HTTP 파라미터를 조회할 수 있습니다.

1. 페이지 아래로 스크롤해 트랜잭션 수행의 가장 마지막 단계로 이동하세요.
2. ☰ HTTP 파라미터 항목을 선택하세요.
3. HTTP-PARAMETERS 창이 나타나면 비밀번호 버튼을 선택하세요.
4. 설정한 Param Key를 입력하세요.

가려진 매개 변수를 확인할 수 있습니다.

- ① • HTTP 파라미터와 관련한 에이전트 설정은 [다음 문서](#)를 참조하세요.
  - 비밀번호: 복호화된 파라미터 값을 확인할 수 있습니다. 비밀번호는 `WHATAP_HOME/security.conf(paramkey.txt)` 파일 내 6자리 문자열입니다. 다른 문자열로 변경 가능합니다. SQL 변수 조회, HTTP 쿼리 조회, Thread 중지 필요합니다. 자세한 내용은 [다음 문서](#)를 참조하세요.

## SQL 파라미터 조회

테이블 뷰 탭에서 SQL 스텝을 선택하거나 트리 뷰 탭에서  버튼을 선택하세요. 파라미터를 조회할 수 있는 SQL 창이 나타납니다.

SQL 📄 🗑️ ✖️

연결 URL jdbc:mysql://localhost:3306,localhost:3310/fake@BasicDataSource

CPU 누적 3ms

메모리 누적 329,752byte

쿼리 (Hash: -2057375474) 🔒 비밀번호 SQL 포매팅 📄 통계 >

SID 107


```
insert into
  dept
values(
  #, '#', '#')
```

**치환 변수**

파라메타가 존재하지 않거나 복호화에 실패하였습니다.  
복호화를 위하여 에이전트 설치 경로의 [paramkey.txt]에서 비밀번호를 확인해주세요.

- **비밀번호:** 복호화된 파라미터 값을 확인할 수 있습니다. 비밀번호는 `WHATAP_HOME/security.conf(paramkey.txt)` 파일 내 6자리 문자열입니다. 다른 문자열로 변경 가능합니다.

ⓘ **비밀번호**는 SQL 변수 조회, HTTP 쿼리 조회, Thread 중지에도 필요합니다. 자세한 내용은 [다음 문서](#)를 참조하세요.

- **SQL 포매팅:** SQL 문장을 들여쓰기 및 포매팅하여 가독성을 높일 수 있습니다.
- : SQL 문장을 클립보드에 복사할 수 있습니다.
- **통계:** [통계/보고서](#) > **통계** 메뉴의 **SQL** 탭으로 이동합니다.

✅ SQL 변수와 HTTP 쿼리를 조회하려면 다음 옵션을 에이전트 설정에 추가하세요.

- SQL 파라미터 정보 기록과 관련한 에이전트 설정은 [다음 문서](#)를 참조하세요.
- HTTP 파라미터 정보 기록과 관련한 에이전트 설정은 [다음 문서](#)를 참조하세요.

```
whatap.conf

# SQL 파라미터 조회 옵션: 옵션이 적용되면 SQL 파라미터를 암호화하여 수집합니다.
profile_sql_param_enabled=true
```



# HTTP 파라미터 조회 옵션: 옵션이 적용되면 HTTP 쿼리 파라미터를 암호화하여 수집합니다.  
 profile\_http\_parameter\_enabled=true

## 테이블 컬럼 설정하기

트랜잭션 목록에서 테이블 헤더 컬럼을 감추거나 원하는 항목을 추가할 수 있습니다. 컬럼 순서를 변경할 수도 있습니다. **▣ 컬럼 설정** 버튼을 선택하세요.

**컬럼 설정**

표시할 컬럼을 선택해주세요

3 컬럼명 검색

1

- 전체 선택
- 경과 시간
- 국가
- 도메인
- 도시
- 로그인 ID
- 멀티 트랜잭션 ID
- 메모리 할당량
- 시작 시간
- 에러 단계
- 에러 메시지
- 에이전트 그룹 명
- 에이전트 그룹 ID
- 에이전트 명 (oname)
- 에이전트 서버 명
- 에이전트 서버 ID
- 에이전트 ID (oid)
- 운영 체제
- 원본 URL

초기화

2 표시 항목(18)

No	
에이전트 명 (oname)	X
트랜잭션	X
경과 시간	X
시작 시간 YY/MM/DD 00:00:00	X
종료 시간 YY/MM/DD 00:00:00	X
HTTP 호출 건수	X
HTTP 호출 시간	X
SQL 시간	X
SQL 건수	X
DB 연결 시간	X
SQL 패치 건수	X
에러 메시지	X
클라이언트 IP	X

취소 확인

- ① • 설정을 완료한 다음에는 **확인** 버튼을 선택해야 설정 사항이 테이블에 반영됩니다.
- ③ 검색란에 텍스트를 입력해 원하는 컬럼 항목을 검색할 수 있습니다. 입력한 텍스트와 매칭되는 컬럼 항목만 표시됩니다.

## 컬럼 추가하기

- ① 목록에서 테이블 헤더 컬럼으로 추가할 항목을 선택하세요. 모든 항목을 추가하려면 **전체 선택**을 선택하세요.

## 컬럼 삭제하기

- ① 목록에서 삭제할 컬럼 항목의 체크 박스를 선택 해제하세요. 또는 ② 목록에서 삭제할 항목의 오른쪽에 **X** 버튼을 선택하세요.

## 컬럼 순서 변경하기

- ② 목록에 순서를 변경할 항목을 드래그해서 원하는 위치로 이동할 수 있습니다.

## 설정 사항 초기화하기

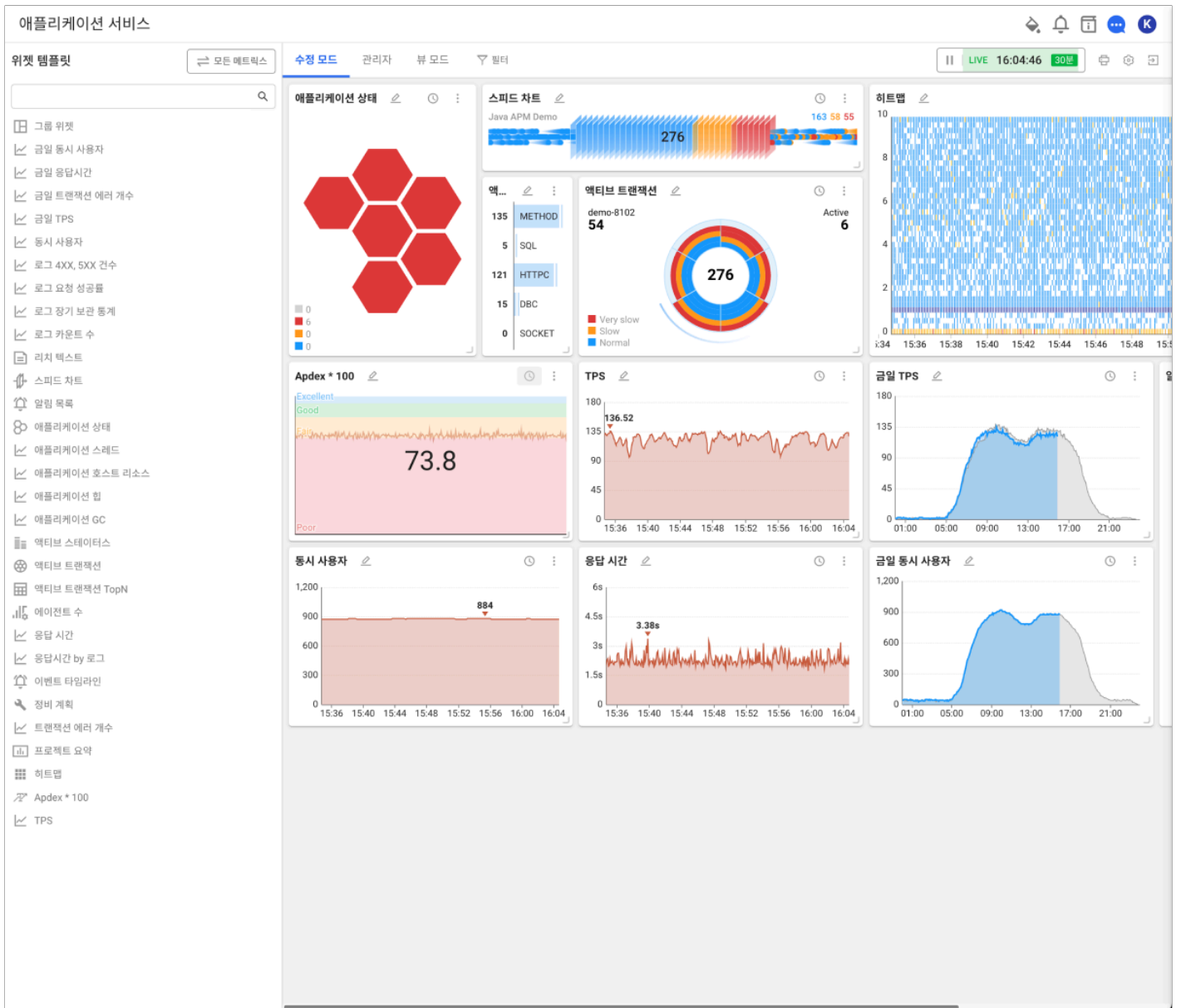
변경 사항은 모두 취소하고 초기화하려면 **초기화** 버튼을 선택하세요.

## 조회 데이터 다운로드

테이블에 출력된 데이터를 txt 파일 형식으로 다운로드하려면 **TXT** 버튼을 선택하세요.

# Flex 보드

Flex 보드는 사용자 정의형 통합 대시보드입니다. 애플리케이션, 서버, 데이터베이스, 컨테이너 등 프로젝트의 데이터로 실시간 대시보드를 만들 수 있습니다.



- 사전 구성된 템플릿을 제공해 초기 설정을 손쉽게 진행할 수 있습니다. 이를 통해 사용자가 원하는 다양한 대시보드를 구성할 수



있습니다.

- 다양한 형태의 데이터 위젯을 추가할 수 있고, 개별 위젯의 속성을 수정해 사용자가 원하는 데이터 형식을 표시할 수 있습니다.
- 데이터 필터링을 통해 사용자가 원하는 모니터링 대상을 간추릴 수 있습니다.
- 시간 범위를 설정해 중요 시간의 데이터를 확인할 수 있습니다.
- 보조 차트를 활용해 다양한 방법으로 데이터를 확인할 수 있습니다.
- 대시보드를 즐겨 찾는 메뉴로 등록할 수 있습니다.
- 개인화된 대시보드를 타 계정에 복사해 활용할 수 있습니다.

Flex 보드는 다음의 진입 경로에 따라 활용 범위가 달라집니다.

- 홈 화면 > [Flex 보드](#)
  - 위젯 생성 시 조회 가능한 모든 프로젝트를 선택 옵션으로 제공합니다.
  - 사용자 계정에 대시보드가 저장되며 다른 사용자에게 복사하기 기능을 이용해 공유할 수 있습니다.
  - 개인 계정 대시보드로 권한에 따른 영향은 없으나 읽기 전용으로 공유된 대시보드의 경우 수정할 수 없습니다.
- 홈 화면 > 프로젝트 선택 > [대시보드](#) > [Flex 보드](#)
  - 위젯 생성 시 해당 프로젝트 정보를 자동 입력합니다.
  - 프로젝트 멤버들에게 생성한 Flex 보드가 자동 공유됩니다.
  - 프로젝트 수정 권한 또는 **프로젝트 플렉스보드 편집** 권한, **Site Admin** 권한을 가진 사용자가 수정할 수 있습니다.

✔ Flex 보드를 수정할 수 있는 사용자는 다음의 주요 기능을 이용할 수 있습니다.

- 대시보드를 json 파일 형식으로 내보내거나 가져올 수 있습니다.
- 대시보드 내의 데이터 요청 및 응답 내용을 확인할 수 있습니다.
- 위젯의 설정 옵션을 json 파일 형식으로 조회하고 수정할 수 있습니다.

ⓘ 프로젝트 내 [Flex 보드](#) 메뉴에서는 대시보드 수정 권한이 있는 사용자만이 [수정 모드](#) 및 [관리자 모드](#), [필터](#) 기능에 접근할 수 있습니다. 접근할 수 있는 멤버 권한은 다음과 같습니다.

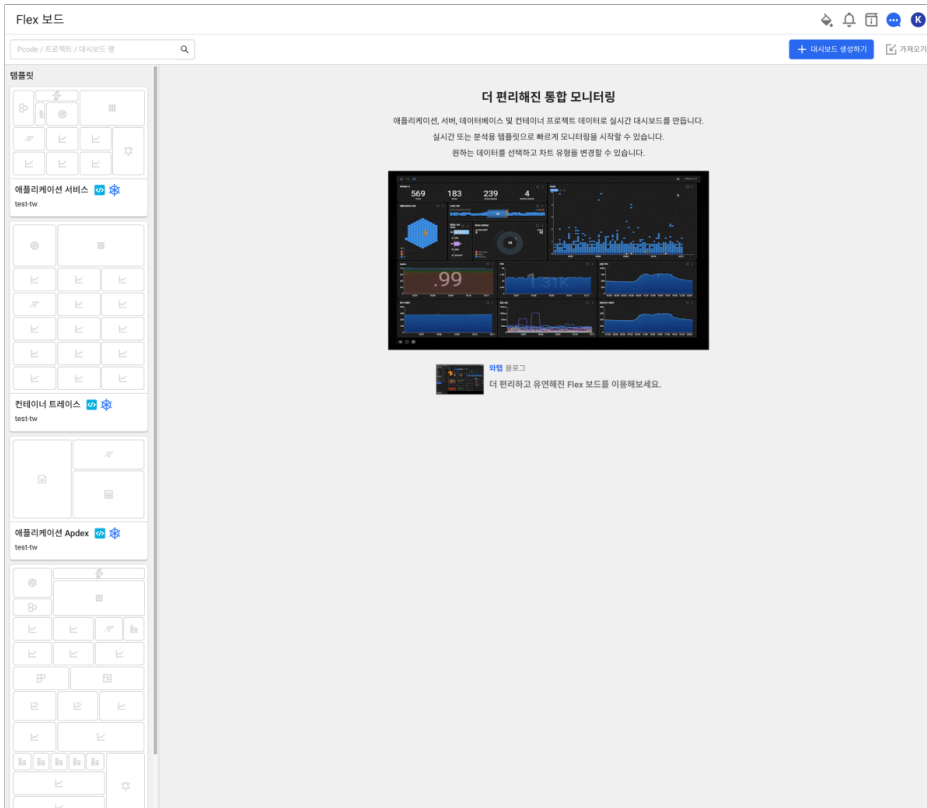
- 프로젝트 수정 권한
- **프로젝트 플렉스보드 편집** 권한
- **Site Admin** 권한
- **화면 모드** 및 [필터](#) 기능에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

- ⓘ • 멤버 권한에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

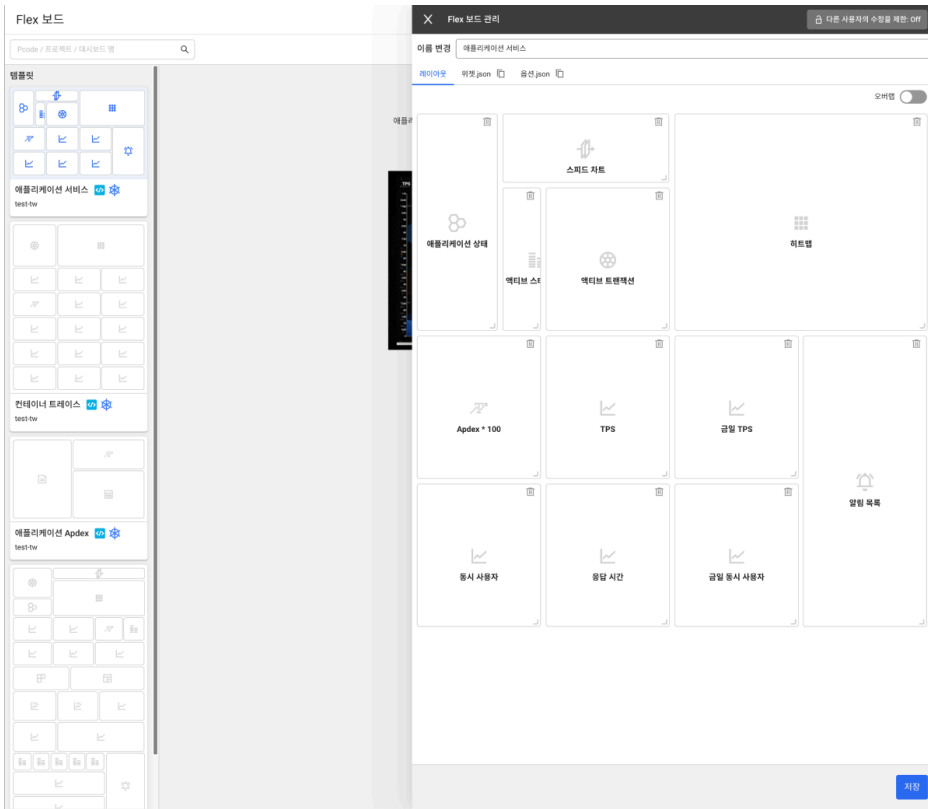
# 사전 템플릿 적용

제공된 사전 구성 템플릿을 이용하세요. 초기 설정을 손쉽게 진행하고 설정을 변경해 활용할 수 있습니다.

1. Flex 보드 메뉴를 선택하세요.
2. 왼쪽 템플릿 목록에서 원하는 템플릿을 선택하세요.



3. Flex 보드 관리 창이 나타나면 이름 변경에 원하는 이름을 입력하세요.



- 레이아웃 항목에 위젯을 마우스로 선택한 다음 원하는 위치로 드래그해 이동할 수 있습니다.
- 오버랩 토글 버튼을 선택해 옵션을 활성화하면 위젯을 서로 겹치게 배치할 수 있습니다.

4. 저장 버튼을 선택하세요.

선택한 템플릿 기준의 Flex 보드를 생성합니다. Flex 보드 메뉴에서 생성한 보드 목록을 확인할 수 있습니다.

# Flex 보드 만들기

## 대시보드 생성

Flex 보드를 생성하고 위젯을 배치해 자신만의 대시보드를 만들 수 있습니다.

1. Flex 보드 메뉴에서 오른쪽 위에 **+ 대시보드 생성하기** 버튼을 선택하세요.
2. **대시보드 생성하기** 창이 나타나면 **이름** 입력 텍스트 상자에 대시 보드 이름을 입력하세요.
3. 위젯 배치 방법을 선택해 **대시보드 생성하기** 버튼을 클릭하세요.
  - **고정 레이아웃 플렉스 보드**: 픽셀 기반으로 자유롭게 위젯을 배치할 수 있습니다.
  - **반응형 플렉스 보드**: 브라우저 사이즈 기준으로 그리드 반응형 레이아웃을 제공합니다. **반응형 플렉스 보드**를 선택하면 그리드를 설정한 다음 **대시보드 생성하기** 버튼을 선택하세요.
4. **위젯 템플릿** 선택 화면에서 대시보드에 배치할 위젯을 선택하세요. 매트릭스 위젯을 추가하려면 **위젯 템플릿**의 오른쪽에 **모든 매트릭스** 버튼을 선택하세요.

- ❗ **위젯 템플릿**: 일반적인 모니터링 상황에서 중요하게 다뤄지는 지표를 간추려 사전 정의된 위젯 목록을 선택할 수 있습니다.
- **모든 매트릭스**: 사용자의 프로젝트에서 수집 중인 모든 매트릭스 데이터를 기준으로 위젯을 생성할 때 사용합니다. 매트릭스 위젯에 대한 자세한 설명은 [다음 문서](#)를 참조하세요.

5. 배치한 위젯의 위치를 이동하거나 크기를 조절하세요.
  - 배치한 위젯의 위쪽으로 마우스 커서를 이동하세요. 커서 모양이 십자 형태로 변경되면 위젯을 마우스로 클릭한 상태에서 드래그하세요. 위치를 이동할 수 있습니다.
  - 배치한 위젯의 오른쪽 아래로 마우스 커서를 이동하세요. 커서 모양이 화살표 형태로 변경되면 위젯을 마우스로 클릭한 상태로 드래그하세요. 원하는 크기로 조절할 수 있습니다.
6. 추가할 위젯을 모두 배치를 완료한 다음 **뷰 모드**를 선택해 레이아웃을 확인하세요. 레이아웃을 다시 수정하려면 **수정 모드**를 선택해 위젯의 배치를 변경하세요.


모든 과정을 완료했다면 오른쪽 위에 **☰ (목록으로 가기)** 버튼을 선택하세요. Flex 보드 메뉴의 **보드** 목록에서 생성한 보드를 확인할 수 있습니다. 생성한 보드를 선택해 새로 생성한 Flex 보드를 확인할 수 있습니다.

- ❗ 고객의 의견을 반영해 대시보드 템플릿 또는 위젯을 추가하려고 합니다. 필요한 대시보드 템플릿 또는 위젯이 있다면

ⓘ [support@whatap.io](mailto:support@whatap.io)로 문의해 주세요.

# Flex 보드 관리하기

## Flex 보드 편집하기

Flex 보드 메뉴의 [대시보드 목록](#)에서  버튼을 선택하세요. 화면 오른쪽에 Flex 보드 관리 창이 나타납니다.

### ⓘ Flex 보드 편집 기능 권한 안내

- 홈 화면 > [Flex 보드](#)  
개인 계정 대시보드로 권한에 따른 영향은 없으나 읽기 전용으로 공유된 대시보드의 경우 수정할 수 없습니다.
- 홈 화면 > 프로젝트 선택 > [대시보드](#) > [Flex 보드](#)  
프로젝트 수정 권한 또는 [프로젝트 플렉스보드 편집](#) 권한, **Site Admin** 권한을 가진 사용자가 수정할 수 있습니다.
- 멤버 권한에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

The image shows two side-by-side screenshots of the Flex Board management interface. The left screenshot displays the 'Flex 보드' (Flex Board) configuration page, which includes a search bar for 'Pcode / 프로젝트 / 대시보드 명', a '템플릿' (Template) section with various widget icons, and a '대시보드 목록' (Dashboard List) section with a list of dashboards. The right screenshot shows the 'Flex 보드 관리' (Flex Board Management) window, which includes a search bar for '이름 변경', a '레이아웃' (Layout) section with a grid of widgets, and a '대시보드 목록' (Dashboard List) section. The layout section contains widgets such as '스피드 차트', '애플리케이션 상태', '액티브 스레드', '액티브 트랜잭션', '히트맵', 'Apdex \* 100', 'TPS', '금일 TPS', '알림 목록', '동시 사용자', '응답 시간', and '금일 동시 사용자'. A '저장' (Save) button is visible in the bottom right corner of the right screenshot.


설정할 수 있는 속성은 다음과 같습니다.

- **이름 변경:** 대시보드의 이름을 수정할 수 있습니다.
- **프로젝트:** 대시보드 위젯에 데이터를 가져올 프로젝트를 선택할 수 있습니다.





❗ 이 옵션은 홈 화면 > Flex 보드 메뉴에서만 설정할 수 있습니다.

#### • 레이아웃

- 위젯을 선택하고 드래그하여 위치를 이동할 수 있습니다.
- 위젯 오른쪽 아래를 마우스로 선택한 다음 드래그해 위젯의 크기를 조절할 수 있습니다.
- 위젯을 삭제하려면 위젯 오른쪽 위에  버튼을 선택하세요.
- **오버랩** 토글 버튼을 활성화하면 위젯의 배치를 겹쳐진 상태로 배치할 수 있습니다.

❗ 홈 화면 > Flex 보드 메뉴로 진입한 경우, 각 위젯마다 데이터를 가져올 프로젝트를 선택할 수 있습니다. 위젯 오른쪽 위에 + 버튼을 클릭한 다음 원하는 프로젝트를 선택하세요. 프로젝트를 검색해 선택할 수 있고, 다중 선택할 수 있습니다. 선택을 완료한 다음 **적용** 버튼을 선택하세요.

- **위젯.json**: 대시보드의 위젯 설정을 json 형식으로 불러올 수 있습니다.  버튼을 선택하면 json 내용을 복사합니다.
- **옵션.json**: 대시보드에 설정된 옵션을 json 형식으로 불러올 수 있습니다.  버튼을 선택하면 json 내용을 복사합니다.

## Flex 보드 삭제하기

Flex 보드 > 대시보드 목록에서 삭제하려는 항목의 오른쪽에  버튼을 선택하세요. 확인 메시지가 나타나면 **삭제** 버튼을 선택하세요.

- ❗ • 대시보드 소유자가 대시보드를 삭제하는 경우 대시보드를 공유한 모든 사용자 계정에서도 해당 대시보드를 삭제합니다.
- 대시보드를 공유 받은 사용자가 대시보드를 삭제하면, 해당 사용자의 **대시보드 목록**에서만 삭제합니다. 원본 대시보드는 그대로 유지합니다.

# 메트릭스 위젯

## 메트릭스란?

프로젝트에서 수집 중인 숫자로된 성능 지표를 **메트릭스**라고 부릅니다. 메트릭스는 '카테고리'로 분류되며, 카테고리별 태그를 이용해 데이터를 탐색할 수 있습니다.

- **카테고리(Category)**: 관련한 지표들을 묶는 단위
- **태그(Tags)**: 수집 대상을 구분할 수 있는 고유 정보가 포함된 데이터
- **필드(Fields)**: 에이전트로부터 수집한 지표

메트릭스에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

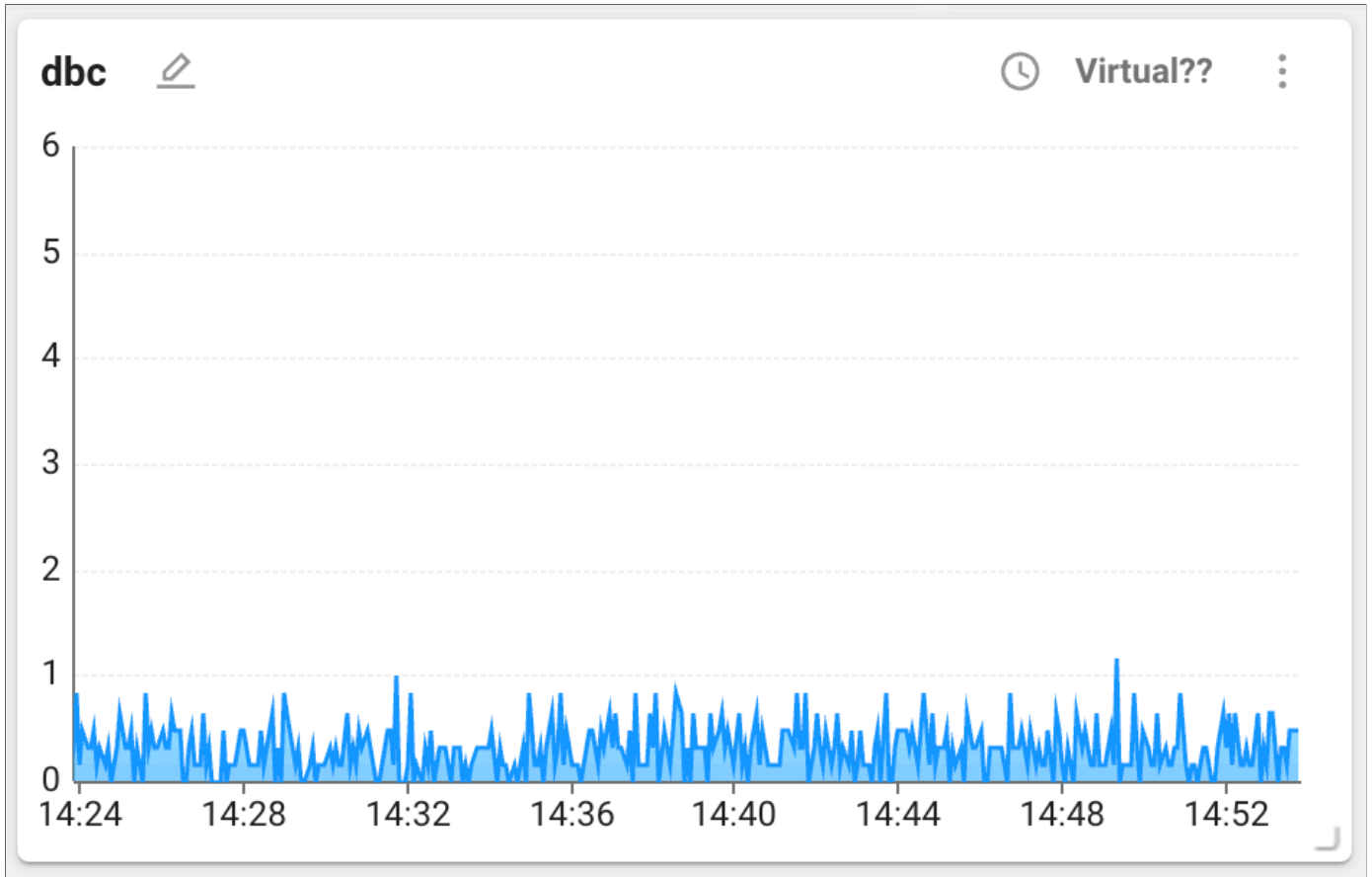
## 메트릭스 위젯 종류

모든 **메트릭스**에서 추가할 수 있는 위젯 유형은 카테고리에 대한 태그와 필드 정보를 나열한 테이블 형식의 위젯, 단일 필드에 대한 추이 그래프로 나뉩니다.

- **테이블 위젯**: 선택한 메트릭스 카테고리에 대한 데이터를 표 형식으로 제공합니다.

container	containerKey	host_ip	okindName	oname	onodeName	pid	Project	type	dbc	httpc	method	socket	sql	oid
prod-agent.service	1204611323	10.21.2.34	demo-okind-0	demo-8102	node-0	3788129	W_JAVA_DEMO	java	0.36	9.6	11	-	3.05	1387800924
prod-agent.service	1204611323	10.21.2.34	demo-okind-1	demo-8103	node-1	3788128	W_JAVA_DEMO	java	0.33	9.66	10.55	-	3.14	633280970
prod-agent.service	1204611323	10.21.2.34	demo-okind-1	demo-8105	node-1	3788134	W_JAVA_DEMO	java	0.3	9.19	11.37	-	2.91	-857948929
prod-agent.service	1204611323	10.21.2.34	demo-okind-1	demo-8101	node-1	3788130	W_JAVA_DEMO	java	0.28	9.93	10.82	-	3.07	-877561626
prod-agent.service	1204611323	10.21.2.34	demo-okind-0	demo-8100	node-0	3788127	W_JAVA_DEMO	java	0.34	9.71	11.38	-	2.88	-1128904592
prod-agent.service	1204611323	10.21.2.34	demo-okind-0	demo-8104	node-0	3788126	W_JAVA_DEMO	java	0.29	9.69	10.96	-	3.11	-1143239575

- **시리즈 위젯**: 선택한 메트릭스 카테고리의 선택 필드에 대한 데이터를 시리즈 차트 형식으로 제공합니다.



ⓘ 메트릭스 위젯 설정에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.


# 위젯 관리하기

Flex 보드에 배치한 위젯 관리를 통해 대시보드를 사용자가 원하는 대로 꾸밀 수 있습니다.

## 위젯 추가

이미 만들어진 Flex 보드에 위젯을 추가할 수 있습니다.


1. Flex 보드 메뉴의 **보드** 목록에서 위젯을 추가할 대시보드를 선택하세요.
2. 화면 왼쪽 위에 **수정 모드**를 선택하세요.
3. 왼쪽에 **위젯 템플릿** 목록에서 원하는 위젯을 선택하세요. 매트릭스 위젯을 추가하려면 **모든 매트릭스**를 선택한 다음 원하는 위젯을 클릭하세요.
  - 홈 화면의 Flex 보드 메뉴로 진입한 경우 위젯을 추가했다면 **+ 프로젝트 선택** 버튼을 클릭해 수집 대상 프로젝트를 선택하세요.
  - 추가할 위젯이 있다면 같은 과정을 반복하세요.


모든 과정을 완료했다면 화면 위에 **뷰 모드**를 선택해 레이아웃을 확인한 다음  (**목록으로 가기**) 버튼을 선택하세요.

## 위젯 속성 변경


Flex 보드 메뉴의 **보드** 목록에서 위젯 속성을 변경할 보드를 선택하세요. **수정 모드**로 진입합니다.

### 위젯 이름 변경

이름을 변경할 위젯의 이름 오른쪽에  버튼을 선택하세요. 위젯의 **제목**, **폰트 크기**, **색상**을 설정한 다음 **저장** 버튼을 선택하세요.

- ✔ • 저장 시 **전체 위젯에 적용** 토글 버튼의 기능을 활성화하면 폰트 크기와 색상을 다른 위젯에도 적용합니다.
  - 위젯의 제목을 잠금 설정하려면  버튼을 선택한 다음 **저장** 버튼을 클릭하세요.

### 시간 설정

시간을 설정하려는 위젯의 오른쪽 위에  버튼을 선택하세요. **시간 선택** 창이 나타납니다.

- 대시보드 화면에 설정한 시간을 적용하려면 **대시보드 설정을 따름**을 체크하세요. 대시보드의 시간 설정은 오른쪽 상단에 위치합니다.
- 사용자가 원하는 시간을 설정을 적용하려면 **대시보드 설정을 따름**을 체크 해제하세요. < 또는 > 버튼을 이용해 원하는 시간을 설정하세요.

시간 설정을 완료했다면 ✕ 버튼을 선택하세요.

ⓘ 위젯의 데이터 유형에 따라서 시간 설정 기능을 지원하지 않을 수 있습니다.

## 출력 데이터 옵션 설정

차트형 위젯의 출력 데이터 옵션을 설정해 사용자가 원하는 데이터를 선택할 수 있습니다. 위젯의 오른쪽 위에 ⋮ 버튼을 선택하세요.

- 프로젝트, 에이전트, 애플리케이션 등의 종류를 선택해 데이터를 필터링할 수 있습니다.
- 개별, 병합 데이터를 선택해 차트에 표시되는 그래프의 유형을 변경할 수 있습니다.
- ⋮ > ⚙️을 선택하세요. **차트 설정** 창을 통해 세부 옵션을 변경할 수 있습니다. 설정할 수 있는 옵션은 다음과 같습니다.
  - **차트 유형**: 집계하는 현재 수치를 차트에 표시할 수 있는 옵션입니다.
  - **y축 차트 최대값**: y축에 차트 최대값을 **자동** 또는 **고정**으로 선택할 수 있습니다.
  - **수평선 설정**: **+ 추가** 버튼을 선택해 임계치 또는 범위를 수평선으로 표시할 수 있습니다. 여러 개를 추가해 적용할 수 있습니다. 수평선을 추가한 다음에는 **적용** 버튼을 선택해야 차트에 반영됩니다.
  - **차트 최대값 표시**: 차트에 표시되는 그래프의 최대값을 표시합니다.
  - **보조 차트**: 위젯의 주요 차트에 보조 차트를 추가 설정할 수 있습니다. 보조 차트의 종류를 선택한 다음 레이아웃, 크기, 데이터 병합 여부를 선택하세요. 다른 위젯에도 공통 적용하려면 **전체 위젯에 적용** 버튼을 선택하세요. 보조 차트는 데이터를 다양한 방법으로 보고 싶을 때 유용합니다.

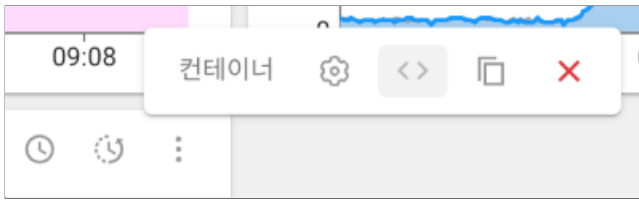
옵션 설정을 모두 완료했다면 **차트 설정** 창의 왼쪽 위에 ✕ 버튼을 클릭하거나 **차트 설정** 창 영역 밖을 클릭하세요.

ⓘ 위젯의 데이터 유형에 따라서 선택할 수 있는 옵션은 다를 수 있습니다.

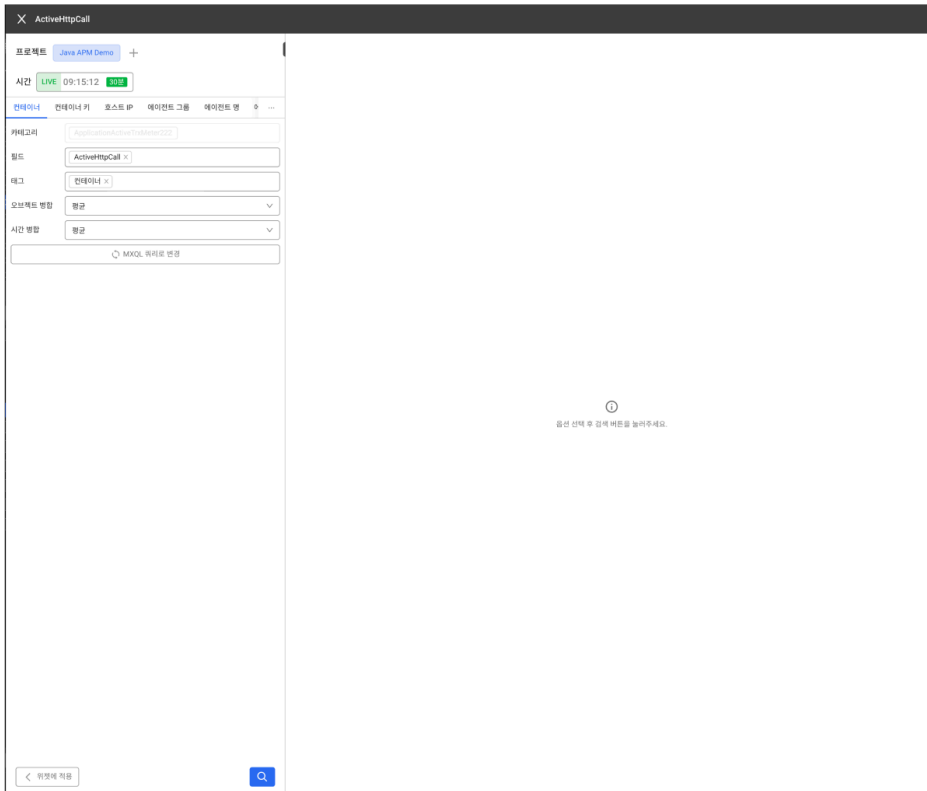
## 메트릭스 위젯 데이터 변경

메트릭스 위젯의 데이터 조회 조건을 변경할 수 있습니다.

1. Flex 보드 > 수정 모드에서 변경할 위젯의 오른쪽 위에 ⋮ 버튼을 선택하세요.




2. <> 버튼을 선택하세요. 데이터 설정 편집 창이 나타납니다.



3. 다음을 참조해 설정을 변경한 다음 < 위젯에 적용 > 버튼을 선택하세요.
  - 위젯 생성 시 기본 생성되는 메트릭스 옵션이 탭에 표시됩니다.
  - 메트릭스 옵션에는 **카테고리**와 **필드**, **태그**가 있습니다.
  - **카테고리** 항목은 현재 변경 기능을 제한합니다.
  - **필드**, **태그** 항목은 복수 선택할 수 있습니다. 선택한 태그 옵션으로 차트 데이터를 그룹화합니다.
  - 🔍 버튼을 선택해 원본데이터를 조회할 수 있습니다.
  - **오브젝트 병합** 및 **시간 병합** 항목에 대한 자세한 내용은 [데이터 병합 옵션](#)을 참조하세요.

## 데이터 병합 옵션

Flex 보드의 [모든 메트릭스](#) 목록에서 배치한 메트릭스 위젯은 데이터 병합 옵션을 제공합니다.

1. 수정 모드에서 [모든 메트릭스](#) 버튼을 선택해 [위젯 템플릿](#) 목록을 [모든 메트릭스](#) 목록으로 변경하세요.
2. 메트릭스 위젯을 레이아웃에 배치하세요.
3. 메트릭스 위젯 오른쪽 위에  버튼을 선택하세요.
4. 태그(예: 에이전트 명) 옵션을 선택하면 데이터 병합 옵션이 나타납니다. 위젯의 크기에 따라 메트릭스 옵션의 위치가 다를 수 있습니다.



데이터 병합은 [오브젝트 병합](#)과 [시간 병합](#) 방법을 제공합니다.

- **오브젝트 병합**

좌측에서 선택한 태그(예: 에이전트 명)를 기준으로 차트 데이터를 그룹화할 때 사용합니다. 이때 서로 다른 필드(예: CPU)값을 가진 데이터들 중에서 태그가 일치하는 경우 해당 데이터를 병합하는 방법입니다.

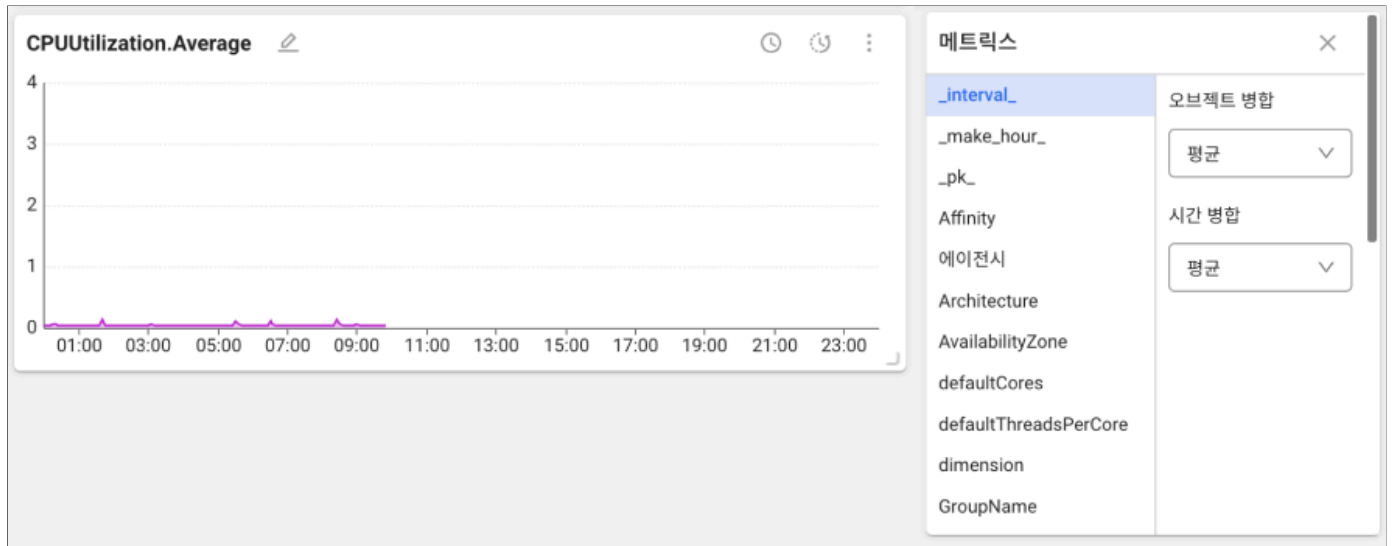
- **시간 병합**

조회 시간이 긴 경우 주로 사용합니다. 원본 데이터에서 필드 값이 같은 데이터끼리 시간 범위에 따라 일정한 시간 간격(5분 또는 1시간)으로 데이터를 병합합니다. 이 병합 방법을 시간 병합이라고 합니다.

- **AWS CloudWatch 메트릭스**

AWS 모니터링 지표의 경우 통계 방식이 필드명 뒤에 suffix(접미사) 형식으로 추가됩니다. 데이터 병합 기본값은 이 suffix를

참조하고 있습니다.



## 위젯 복사

1. 위젯의 오른쪽 위에 버튼을 선택하세요.
2. 버튼을 선택하세요.
3. 복사 창이 나타나면 **+ 위젯 추가**를 선택하세요.
4. 목록에 위젯의 이름을 변경하세요.
5. **적용** 버튼을 선택하세요.

수정 모드 화면에서 복사한 위젯을 확인할 수 있습니다.

## 위젯 삭제

1. 위젯의 오른쪽 위에 버튼을 선택하세요.
2. 버튼을 선택하세요.
3. 확인 메시지 창이 나타나면 **삭제** 버튼을 선택하세요. 삭제를 취소하려면 **취소** 버튼을 선택하세요.




# Flex 보드 화면 모드

Flex 보드의 화면 모드에 따라 제공하는 기능을 설명합니다.

- ❗ 프로젝트 내 **Flex 보드** 메뉴에서는 대시보드 수정 권한이 있는 사용자만이 **수정 모드** 및 **관리자** 모드, **필터** 기능에 접근할 수 있습니다. 접근할 수 있는 멤버 권한은 다음과 같습니다.
  - 프로젝트 수정 권한
  - 프로젝트 플렉스보드 편집 권한
  - **Site Admin** 권한
- 멤버 권한에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

## 화면 모드

**Flex 보드**는 다음과 같은 화면 모드를 제공합니다. 화면 모드에 따라 설정할 수 있는 기능이 다릅니다.

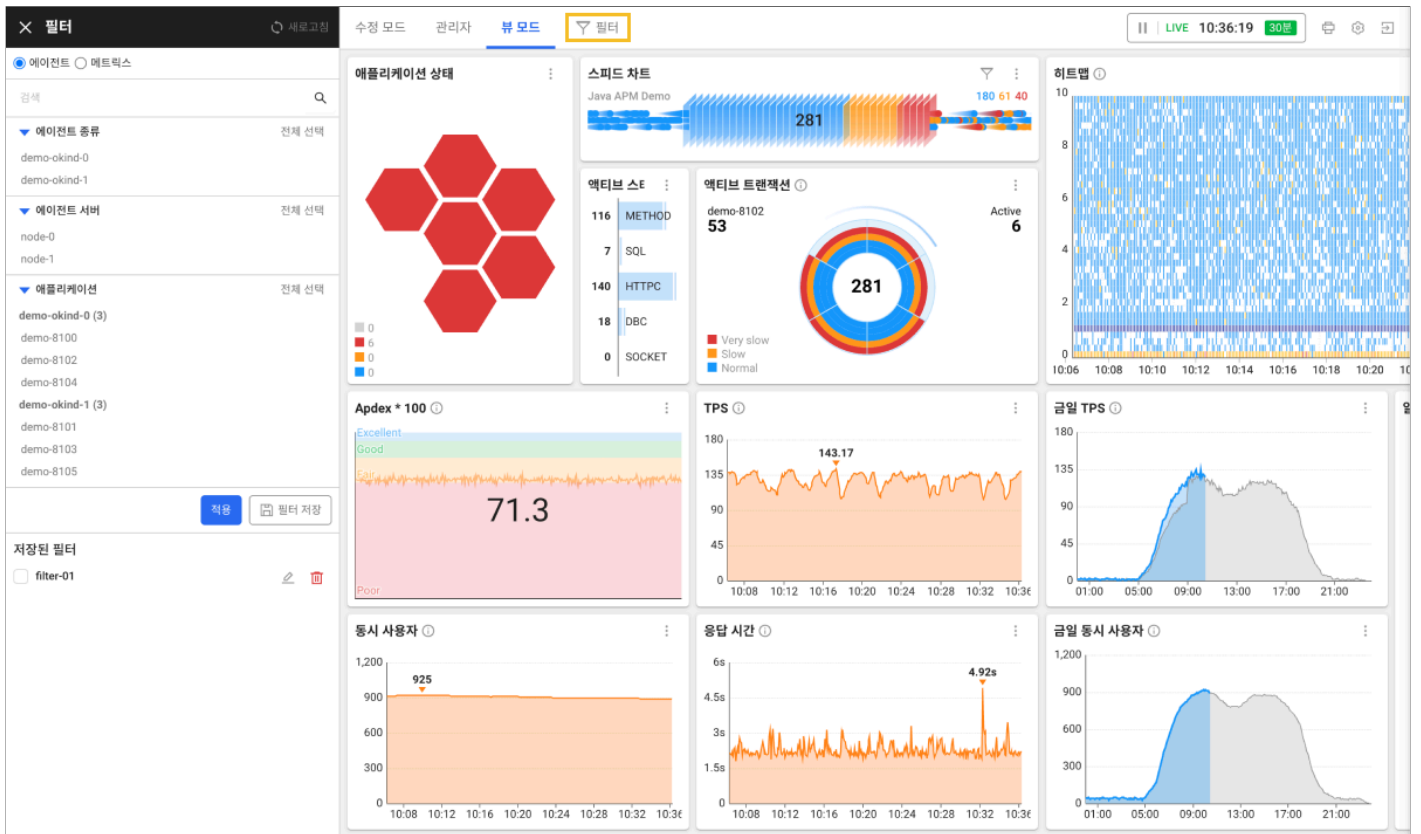
- **수정 모드**: 위젯 템플릿 목록에서 Flex 보드의 레이아웃에 위젯을 배치하거나 위젯을 수정, 삭제, 위치 조정할 수 있습니다.
- **관리자**: 위젯의 세부 속성을 json 형태로 관리할 수 있습니다. 레이아웃에 배치한 위젯의 오른쪽 위에 위치한  버튼을 선택하세요.
- **뷰 모드**: 위젯 배치, 크기 조절, 세부 옵션 등의 설정을 완료한 다음 대시보드와 같이 각 위젯의 데이터를 실시간으로 모니터링할 수 있습니다.

**Flex 보드**에서는 다음의 공통 기능을 사용할 수 있습니다.

- 데이터 필터링
- 실시간 데이터 조회 범위 설정
- 인쇄 모드
- Flex 보드 관리
- 데이터 병합 옵션

## 데이터 필터링

Flex 보드는 수백 가지의 종류의 데이터를 다룹니다. 데이터 필터링 기능을 이용해 모니터링하려는 대상을 간추려 Flex 보드를 구성하세요.



- 화면 위에 ① 필터를 선택하세요.
- 화면 왼쪽으로 ② 필터 창이 나타납니다.
  - 에이전트** 옵션을 선택했다면 에이전트 종류 또는 애플리케이션 목록 기준으로 필터링 대상을 선택할 수 있습니다. 필터링 대상을 모두 선택하려면 **전체 선택**을 선택하세요. 필요한 경우 **검색** 기능을 이용해 선택할 수 있습니다.
  - 메트릭스** 옵션을 선택했다면 **카테고리** 옵션에서 프로젝트, 에이전트 유형에 따라 필터링 대상을 선택할 수 있습니다. **카테고리** 옵션은 + **And**를 선택해 추가할 수 있습니다. **포함**, **제외** 옵션을 선택하면 문자열로 필터링 대상을 자동 선택합니다.
- 필터링 대상 선택을 완료했다면 **적용** 버튼을 선택하세요.

사용자가 설정한 필터링 설정을 저장해 이용하려면 **필터 저장** 버튼을 선택하세요. 필터 이름을 입력하고 **저장**을 선택하세요. ③ **저장된 필터**는 **필터** 창 하단에 나타납니다.

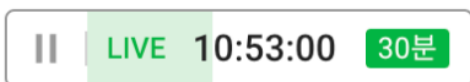
필터링 조건은 에이전트 타입과 메트릭스 타입으로 구분합니다.

타입	에이전트	메트릭스
선택 옵션	와탭 프로젝트 모니터링 대상 (애플리케이션, 서버, 데이터베이스, 컨테이너 및 쿠버네티스 POD와 NODE)	에이전트를 포함한 모든 메트릭스 태그
적용 범위	모든 위젯	메트릭스 데이터를 사용하는 모든 위젯
필터링 조건	체크박스	일치, 포함, 제외 방식

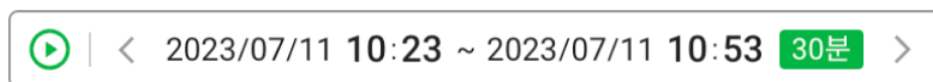
- ① • 에이전트 타입은 기본적인 모니터링 대상만을 선택 옵션으로 제공합니다.
  - 메트릭스 타입은 프로젝트에서 수집 중인 모든 메트릭스 데이터의 태그를 선택 옵션으로 제공합니다. 메트릭스 타입은 클라우드 모니터링과 같이 에이전트가 아닌 형태로 데이터를 수집한 후 필터링을 적용할 때 유용합니다.
  - 메트릭스에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

## 실시간 데이터 조회 범위 설정

Flex 보드 메뉴의 화면 오른쪽 위에 UI 요소를 이용해 위젯에 데이터를 출력하는 시간을 설정할 수 있습니다.




II 버튼을 선택하면 실시간 데이터 조회를 일시 정지하고 다음과 같이 조회 시간을 설정할 수 있습니다.



- 가장 오른쪽의 녹색 배경의 시간 버튼을 선택하세요. 다양한 시간 설정을 선택할 수 있는 옵션이 나타납니다. 원하는 시간 설정을 선택하세요.
- < 또는 > 버튼을 클릭하면 설정된 시간 범위 만큼 조절할 수 있습니다.
- 날짜와 시간 텍스트 영역을 클릭하면 날짜와 시간을 선택할 수 있는 옵션이 나타납니다.
- 조회 시간 설정을 모두 완료했다면 ▶ 버튼을 선택하세요. 실시간 데이터 조회를 시작합니다.


## 인쇄하기


Flex 보드 화면을 PDF로 저장해 인쇄할 수 있습니다. Flex 보드 화면의 오른쪽 위에  버튼을 선택하세요. 화면을 인쇄 모드로 전환합니다. 다음의 옵션을 설정한 다음 [PDF 다운로드](#)를 선택하세요.

- **필터 옵션 보기**: 화면에 필터 옵션을 켜거나 끌 수 있습니다.
- **가로/세로 방향으로 변환**: 화면을 가로 또는 세로로 변환할 수 있습니다.


인쇄 모드를 종료하려면 오른쪽 위에 **×** [인쇄모드 종료](#)를 선택하세요.

## Flex 보드 관리

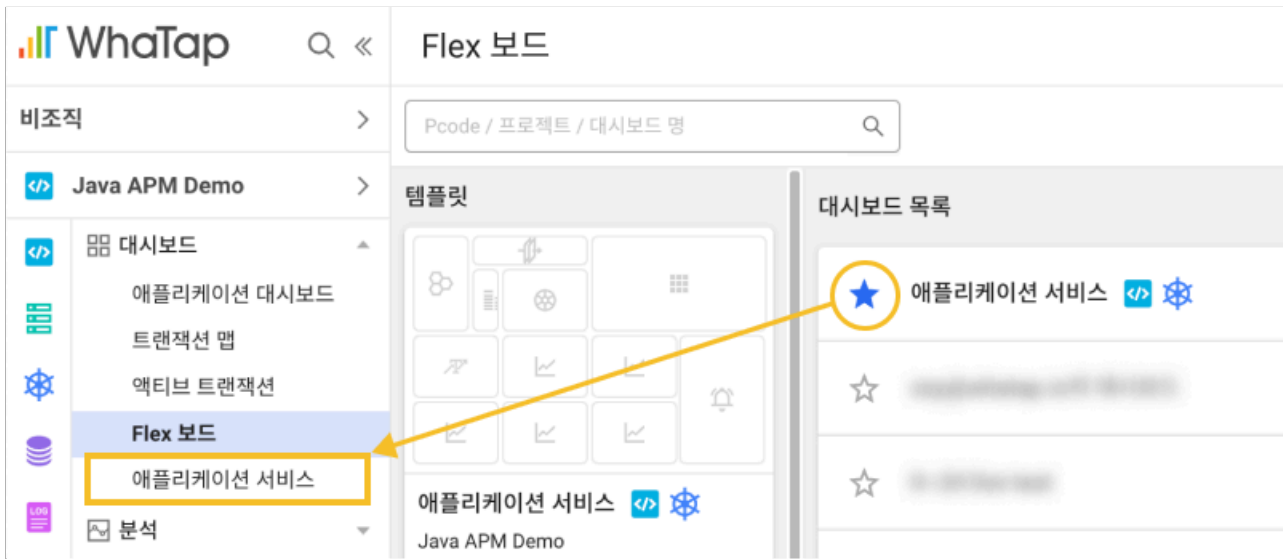
Flex 보드의 레이아웃을 변경하거나 위젯의 위치 변경, 크기 조절, 이름 변경 등의 관리 기능을 제공합니다. Flex 보드 화면의 오른쪽 위에  버튼을 선택하세요. [Flex 보드 관리](#) 창이 나타납니다.

- **이름 변경**: Flex 보드의 이름을 변경할 수 있습니다. 변경하려는 이름을 입력하세요.
- **레이아웃**: 위젯의 배치를 변경하거나 위치 및 크기를 조절할 수 있습니다. 배치한 위젯을 삭제하려면  버튼을 클릭하세요. [오버랩](#) 토글 버튼을 선택해 옵션을 활성화하면 위젯을 서로 겹치게 배치할 수 있습니다.
- **다른 사용자의 수정을 제한**: 다른 사용자가 해당 Flex 보드의 수정을 할 수 없도록 설정합니다. 수정 권한이 있는 멤버만 수정 또는 삭제, 위젯을 편집할 수 있습니다. 버튼을 클릭하면 [다른 사용자의 수정을 제한](#): on으로 변경됩니다.
- **위젯 json**: Flex 보드에 포함된 위젯을 json 형태로 편집해 관리할 수 있습니다.
- **옵션 json**: Flex 보드의 설정을 json 형태로 편집해 관리할 수 있습니다. 프로젝트 정보, Flex 보드 필터, 시간 선택 등의 설정값을 편집할 수 있습니다.

모든 설정을 완료한 다음에는 화면 오른쪽 아래에 [저장](#) 버튼을 클릭하세요.

- ① • **위젯 json, 옵션 json** 기능은 대시보드 관리자 권한의 사용자에게만 제공합니다.
  - [Flex 보드 관리](#) 기능은 [대시보드](#) > [Flex 보드](#) 메뉴에서 [보드 목록](#)의  버튼을 선택해 진입할 수도 있습니다.

## 대시보드 즐겨찾기로 등록

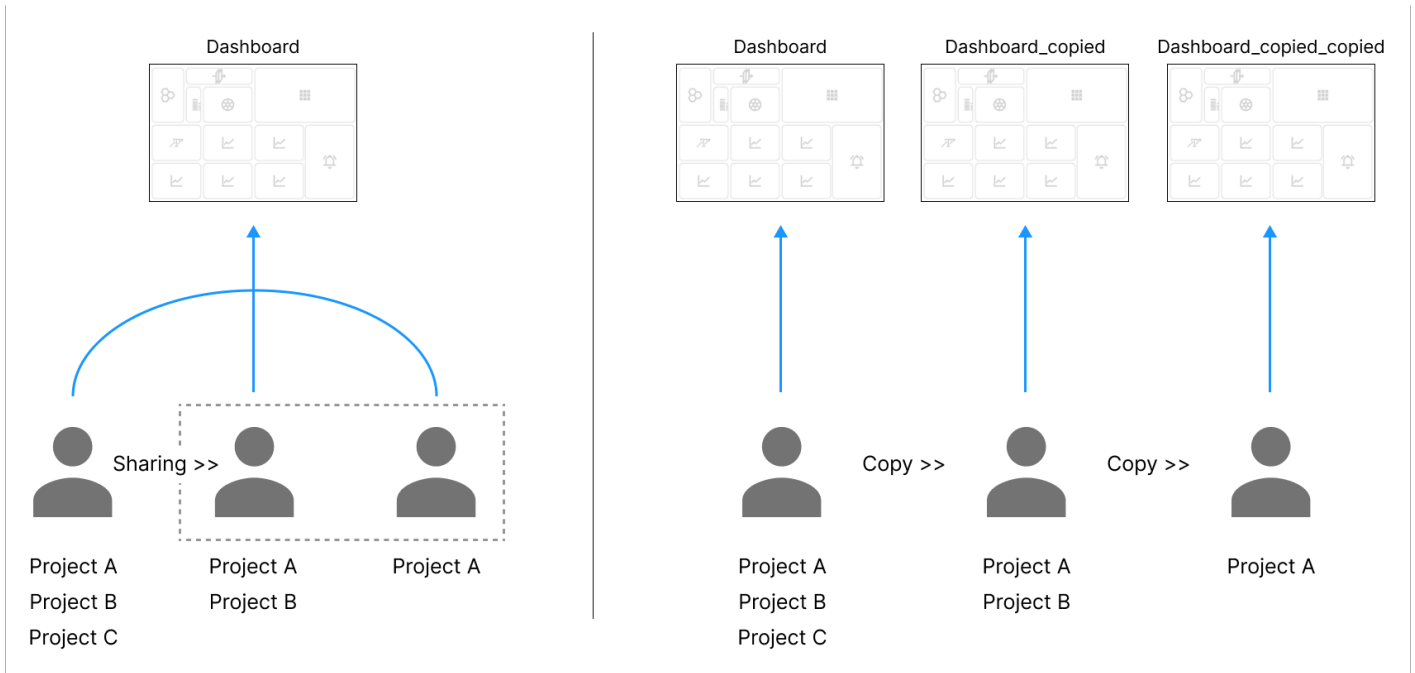


사용자가 생성한 Flex 보드는 [대시보드](#) 메뉴 하위에 즐겨찾는 메뉴로 등록할 수 있습니다. [대시보드](#) > [Flex 보드](#) 메뉴에서 [대시보드 목록](#)의 ☆ 버튼을 선택하세요. 즐겨찾기로 설정된 항목은 ★으로 변경됩니다.

# Flex 보드 공유하기

통합 Flex 보드 메뉴에서 생성한 대시보드를 다른 멤버와 공유하거나 복사해 재사용할 수 있습니다.

## 공유와 복사의 차이



- 대시보드를 공유한 경우 공유한 멤버는 모두 같은 대시보드를 볼 수 있습니다.
- 대시보드를 다른 멤버 또는 나에게 복사한 경우 복사된 별도의 대시보드를 볼 수 있습니다.
- **읽기 전용**으로 대시보드를 공유 또는 복사 받은 멤버는 대시보드를 수정할 수 없지만 **수정 모드**로 공유 또는 복사된 대시보드는 수정할 수 있습니다.
- **읽기 전용**으로 대시보드를 공유받은 멤버는 읽기 전용으로 대시보드를 공유할 수 있으며 복사할 수 있습니다.
- 대시보드에 포함된 프로젝트 중 최소 1개 프로젝트에 초대된 멤버에게 대시보드를 공유할 수 있습니다. 공유 또는 복사 받을 멤버가 프로젝트에 소속되어 있는지 확인하세요.




### ❗ 대시보드 공유 및 복사 기능이 업데이트됐습니다

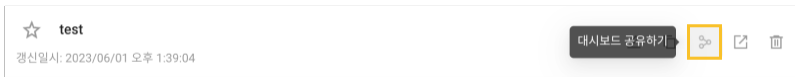
이전에는 공유 또는 복사 받을 멤버에게 수정 권한이 있어야 했지만, [Service 2.3.0 릴리스](#) 이후 권한에 상관없이 프로젝트에 소속되어 있기만 한다면 복사 또는 공유할 수 있습니다. 단, 공유 받은 대시보드의 프로젝트 중 **조회 분석** 권한이 없는 프로젝트는 데이터를 조회할 수 없습니다. 멤버 권한에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

## Flex 보드 공유하기

사용자가 생성한 Flex 보드를 다른 멤버와 공유해 동일한 대시보드를 볼 수 있습니다. 이를 통해 업무를 공유할 수 있습니다. 공유한 대시보드를 수정한 경우 공유받은 모든 멤버는 동일한 대시보드를 확인할 수 있습니다.

❗ 이 기능은 **통합 Flex 보드** 메뉴에서만 이용할 수 있습니다. 프로젝트의 **Flex 보드** 메뉴에서는 이용할 수 없습니다.

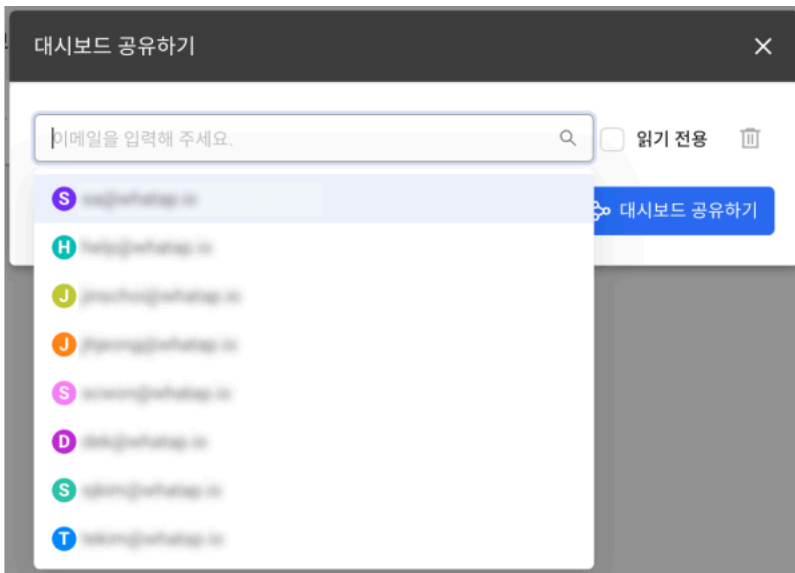
1. **통합 Flex 보드** 메뉴의 **대시보드 목록**에서 공유하려는 대시보드의  버튼을 선택하세요.



2. **대시보드 공유하기** 창이 나타나면 **+ 계정** 버튼을 선택하세요.



3. 대시보드를 공유할 멤버를 선택하세요.



- 공유 대상을 추가하려면 + **계정** 버튼을 선택하세요. 멤버를 추가로 선택할 수 있습니다.
- 대시보드에 포함된 프로젝트 중 최소 1개 프로젝트에 초대된 멤버에게 대시보드를 공유할 수 있습니다. 공유 또는 복사 받을 멤버가 프로젝트에 소속되어 있는지 확인하세요.

#### 4. **대시보드 공유하기** 버튼을 선택하세요.



- **읽기 전용**을 선택하면 공유받은 멤버는 대시보드를 수정할 수 없습니다.
- 공유받을 멤버 모두에게 읽기 전용으로 공유하려면 **읽기 전용(전체)**를 선택하세요.

대시보드를 공유한 항목은 **대시보드 목록**에서 **공유** 태그가 표시된 것을 확인할 수 있습니다. **공유** 태그에 마우스를 오버하면 대시보드를 공유한 멤버에 대한 정보를 확인할 수 있습니다. 읽기 전용으로 공유한 멤버에게는 **읽기 전용** 태그가 표시됩니다.




대시보드를 공유받은 경우	대시보드를 공유한 경우(대시보드 소유자)
	

### ✔ 대시보드 공유 조건

- 대시보드 공유 기능은 홈 화면 > [통합 Flex 보드](#) 메뉴에서만 이용할 수 있습니다.
- 누구나 자신의 대시보드를 공유할 수 있으며, 공유받은 대시보드를 다시 공유하거나 복사할 수 있습니다.
- [읽기 전용](#)으로 대시보드를 공유받은 멤버는 대시보드를 수정할 수 없지만 [수정 모드](#)로 공유받은 대시보드는 수정할 수 있습니다.
- 대시보드 소유자가 대시보드를 삭제하면 대시보드를 공유받은 모든 멤버 계정에서도 해당 대시보드가 삭제됩니다.
- 대시보드를 공유받은 멤버가 대시보드를 삭제하면, 해당 멤버의 [대시보드 목록](#)에서만 삭제됩니다. 원본 대시보드는 그대로 유지합니다.
- 공유 받은 대시보드의 프로젝트 중 [조회 분석](#) 권한이 없는 프로젝트는 데이터를 조회할 수 없습니다.

## Flex 보드 복사하기

사용자가 생성한 Flex 보드를 복사해 다른 멤버에게 전달할 수 있습니다.

[통합 Flex 보드](#) 메뉴의 [대시보드 목록](#)에서  버튼을 선택하세요. [나에게 복사](#) 또는 [다른 사람에게 복사](#) 옵션을 선택하세요.

- [나에게 복사](#) 옵션을 선택하면 [대시보드 목록](#)에 '\_copied' 접미어가 붙은 항목이 추가됩니다.
- [다른 사람에게 복사](#) 옵션을 선택하면 + [계정](#)을 선택하세요.




다른 멤버의 이메일을 선택한 다음 [대시보드 복사하기](#) 버튼을 클릭하세요. 복사 받은 멤버의 [대시보드 목록](#) 목록에 '\_copied' 접미어가 붙은 항목이 추가됩니다.

## JSON 파일로 공유하기


Flex 보드 설정을 JSON 파일로 저장하고 다른 멤버에서 전달하거나 다른 멤버의 설정을 가져올 수 있습니다.

### 내보내기

1. [통합 Flex 보드](#) 또는 프로젝트의 [Flex 보드](#) 메뉴로 이동하세요.
2. [대시보드 목록](#)에서  버튼을 선택하세요.
3. JSON 파일이 다운로드되면 공유할 다른 멤버에게 전달하세요.

### 가져오기

1. [통합 Flex 보드](#) 또는 프로젝트의 [Flex 보드](#) 메뉴로 이동하세요.

2. 화면 오른쪽 위에  가져오기 버튼을 선택하세요.
3. 다운로드한 JSON 파일을 선택하세요.

ⓘ 이 기능은 프로젝트의 수정 권한을 소유한 멤버만 이용할 수 있습니다.

# 분석

애플리케이션 환경을 다양한 관점에서 분석할 수 있는 메뉴를 제공합니다. 부하가 높거나 성능이 저하된 구간을 파악할 수 있으며, 다양한 성능 지표를 제공해 일별 현황을 확인할 수 있습니다. 또한 성능 측정을 위한 트랜잭션을 추적하고 분석할 수 있습니다.

와탭에서 제공하는 분석 기능을 통해 주요 성능을 분석하고, 문제를 파악해 발생할 수 있는 장애에 대처할 수 있습니다.

## 일자별 애플리케이션 현황

애플리케이션의 주요 성능 지표들의 하루 동안 추이를 시간 단위 차트를 통해 확인할 수 있습니다. 자세한 내용은 [다음 문서](#)를 참조하세요.

## 성능 추이

특정 기간 동안 애플리케이션 주요 성능 지표들의 추이를 차트를 통해서 부하가 높거나 성능이 저하된 구간을 빠르게 파악할 수 있고, 문제가 된 시점을 특정할 수 있습니다. 자세한 내용은 [다음 문서](#)를 참조하세요.

## 메트릭스 차트

시각화한 차트를 통해 메트릭스 데이터를 조회할 수 있습니다. 자세한 내용은 [다음 문서](#)를 참조하세요.

## 메트릭스 조회

카테고리화된 애플리케이션의 성능 지표를 태그와 필드 기반으로 구성된 데이터로 조회할 수 있습니다. 자세한 내용은 [다음 문서](#)를 참조하세요.

## 스택

트랜잭션의 스택 정보를 수집해 실행 중인 메소드의 사용량 통계와 많이 사용되는 스택에 대한 통계를 제공합니다. 실행 중인 트랜잭션의 스택 정보를 통해 장시간 실행되거나 짧은 시간 실행되지만 자주 실행되는 메소드를 파악할 수 있습니다. 자세한 내용은 [다음 문서](#)를 참조하세요.

## 큐브

5분 단위로 만든 성능 통계를 큐브(Cube)라고 부릅니다. 사후 분석을 위한 통계 분석 도구로 응답 시간 및 에러 건수, 처리량 등이 높은 시간대를 특정할 수 있기 때문에 동 시간대의 문제 요소를 파악하거나 서비스 현황을 모니터링할 수 있습니다. 트래픽의 지리적 분포, 리소스 사용량을 파악할 수 있습니다. 자세한 내용은 [다음 문서](#)를 참조하세요.

## 히트맵

시간의 흐름에 따라 사용자 의 요청에 대한 응답시간을 분포도 형태로 표현한 차트를 제공합니다. 분포도 차트의 형태에 따라 어떤 장애가 발생했는지 특정하는데 도움을 받을 수 있습니다. 자세한 내용은 [다음 문서](#)를 참조하세요.

## 멀티 트랜잭션 추적

다른 에이전트나 프로젝트와 연관된 트랜잭션 간의 호출을 추적합니다. 시스템 내 또는 시스템 간에 발생하는 다양한 호출 관계를 한 눈에 파악하고 어느 부분에서 문제가 발생했는지 식별하여 개선할 수 있도록 트랜잭션과 트레이스 정보를 제공합니다. 자세한 내용은 [다음 문서](#)를 참조하세요.

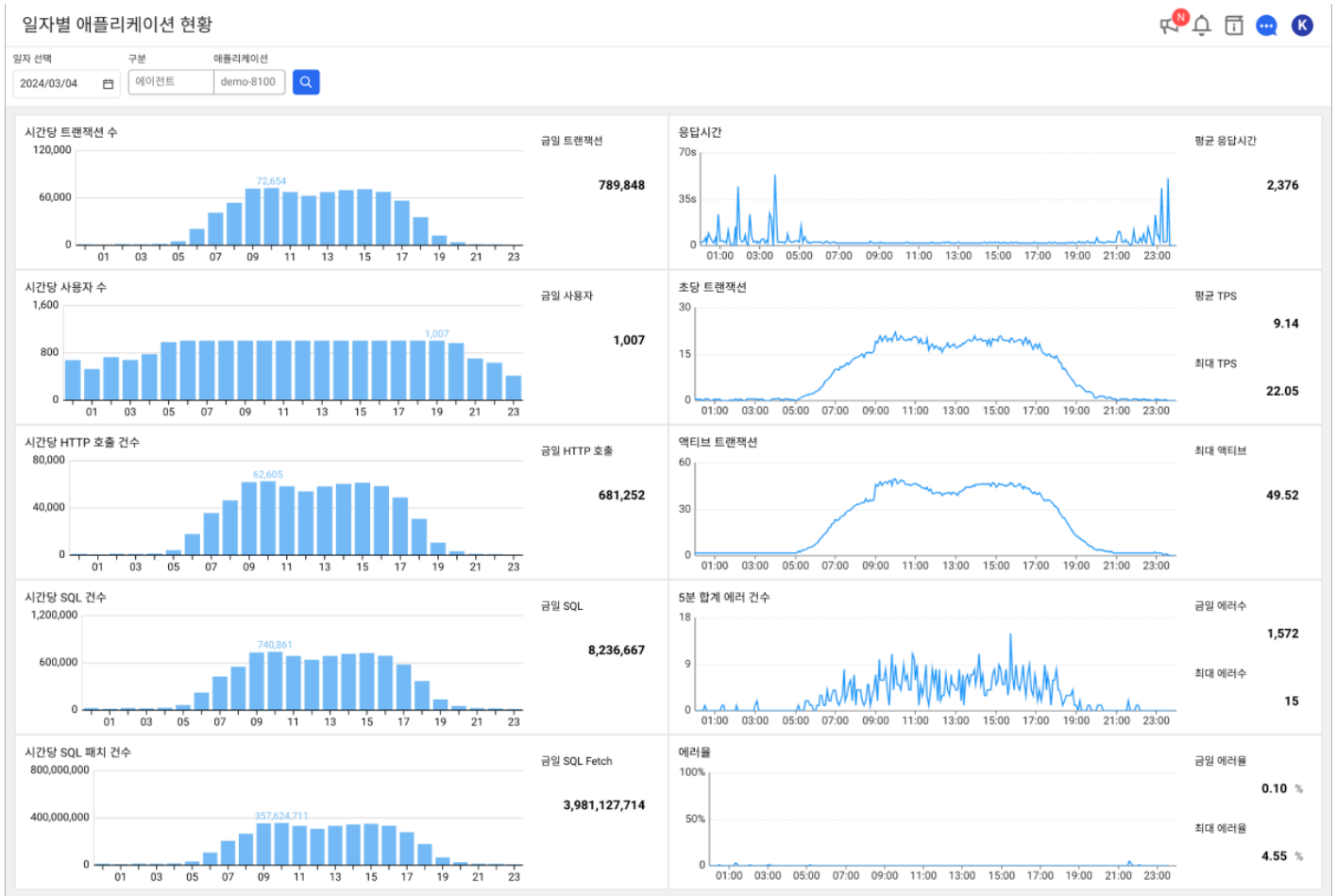
## 트랜잭션 검색

특정 기간 동안 각 트랜잭션의 성능 속성에 대한 통계와 실행 상태를 검색할 수 있습니다. 모든 트랜잭션을 확인하고 정상 및 에러 상태를 빠르게 확인할 수 있습니다. 제공되는 정보를 통해 어떤 트랜잭션의 사용량이 많고 리소스를 많이 소비하는지 알 수 있습니다. 트랜잭션의 속성에 따라 다양하게 필터링할 수 있어 특정 조건의 트랜잭션을 찾는데 유용합니다. 자세한 내용은 [다음 문서](#)를 참조하세요.

# 일자별 애플리케이션 현황

홈 화면 > 프로젝트 선택 > 분석 > 일자별 애플리케이션 현황

애플리케이션의 주요 성능 지표들의 하루 동안 추이를 시간 단위 차트를 통해 확인할 수 있습니다.



하루 동안의 애플리케이션 상태를 확인할 수 있습니다. 검색 조건(일자 선택, 구분, 애플리케이션)을 설정하고 🔍 버튼을 선택하세요. 구분 옵션의 조건은 다음과 같습니다.

분류	설명
에이전트	프로젝트에 포함된 애플리케이션 에이전트 이름
에이전트 종류	에이전트 설정에서 <code>whatap.okind</code> 옵션으로 분류된 그룹 단위
에이전트 서버	에이전트 설정에서 <code>whatap.onode</code> 옵션으로 분류된 그룹 단위

제공하는 차트는 다음과 같습니다. 차트의 오른쪽에 위치한 수치는 하루 동안 누적된 값 또는 평균, 최댓값을 확인할 수 있습니다.

- 시간당 트랜잭션 수
- 시간당 사용자 수

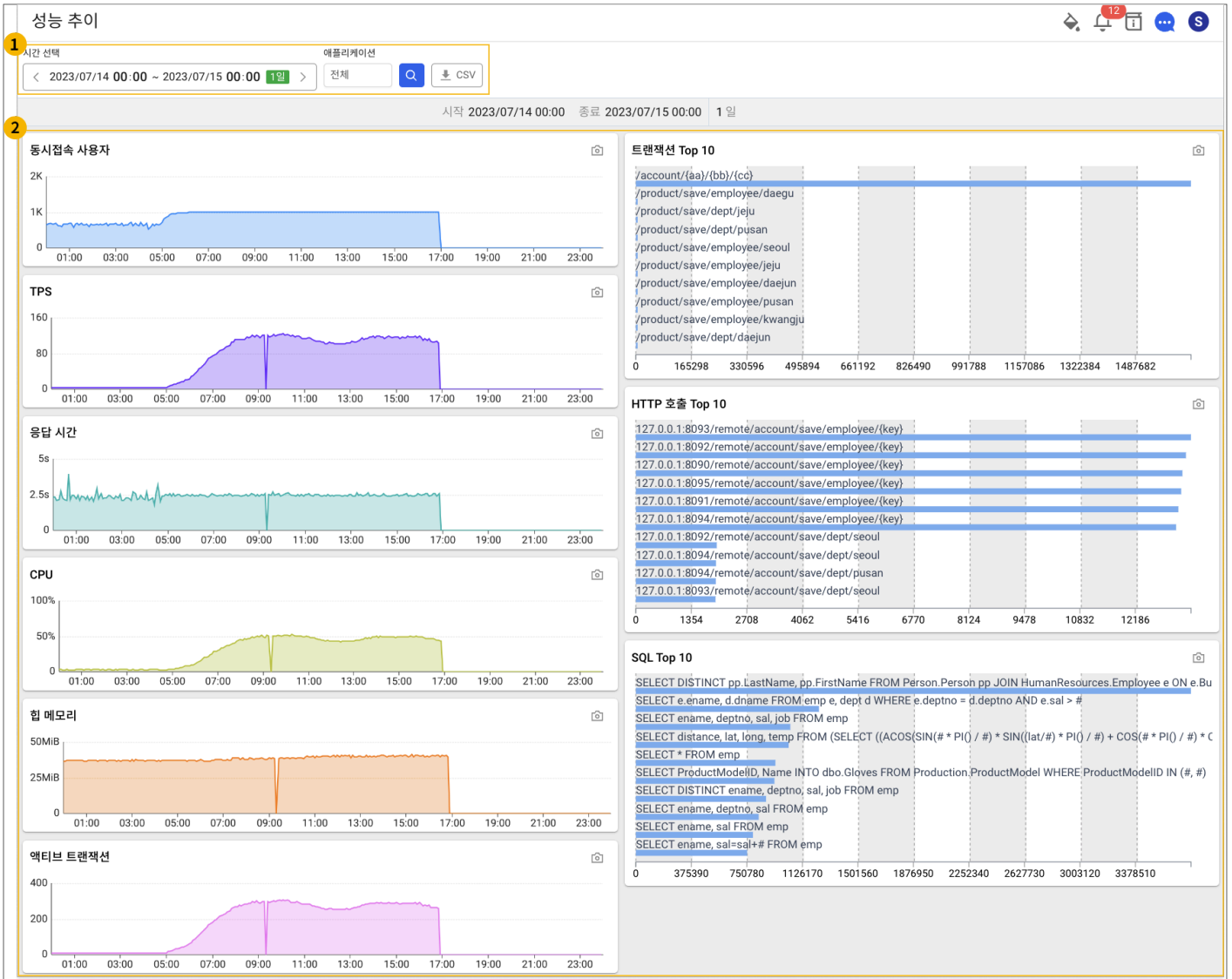
ⓘ **금일 사용자**는 하루 동안 누적된 사용자를 중복없이 합산한 값입니다. 단순히 합산하는 것이 아니라 **HyperLogLog**로 산출합니다. **HyperLogLog**는 매우 적은 메모리로 집합의 원소 개수를 추정하는 확률적 자료 구조를 의미합니다.

- 시간당 HTTP 호출 건수
- 시간당 SQL 건수
- 시간당 SQL 패치 건수
- 응답시간
- 초당 트랜잭션
- 액티브 트랜잭션
- 5분 합계 에러 건수
- 에러율

# 성능 추이


홈 화면 > 프로젝트 선택 > 분석 > 성능 추이

와탭 모니터링 서비스 초기 화면에서 프로젝트를 선택한 다음 **프로젝트 메뉴** 하위에 **분석 > 성능 추이** 메뉴를 선택하세요. 조회를 원하는 특정 시간 범위 내 수행된 성능에 대한 여러 지표를 **성능 추이** 메뉴에서 확인할 수 있습니다.






## 1 성능 추이 조회 조건

- 시간 선택자를 통해 조회를 원하는 시간 범위를 선택할 수 있습니다. 기본 조회 기간은 **1일**입니다.
  - 조회 범위가 3시간 이하인 경우 5초 Raw 데이터를 사용해 5초 추이를 확인할 수 있습니다.
  - 조회 범위가 하루 이하인 경우 5분 통계를 사용해 5분 추이를 확인할 수 있습니다.
  - 조회 범위가 하루 이상인 경우 1시간 통계를 사용해 1시간 추이를 확인할 수 있습니다.
- 애플리케이션 선택자를 통해 전체 또는 조회를 원하는 특정 애플리케이션을 선택할 수 있습니다.
-  CSV 다운로드 아이콘을 선택해 화면에 표시된 목록 데이터를 CSV 파일로 다운로드할 수 있습니다.

## 2 성능 추이 차트

- 지정한 조회 시간 범위를 차트 상단에서 확인할 수 있습니다.
- 차트 영역에서 **동시 접속 사용자**, **응답 시간**, **CPU**, **힙 메모리**, **액티브 트랜잭션**, **트랜잭션 Top 10**, **HTTP 호출 Top 10**, **SQL Top 10** 등의 정보를 확인할 수 있습니다.
-  **스냅샷** 아이콘을 선택해 위젯의 옵션을 제외한 차트를 스냅샷 할 수 있습니다.

### 동시 접속 사용자

실시간 브라우저 사용자 수를 보여줍니다. 사용자는 브라우저 IP를 기반으로 카운팅합니다.

- 5분 통계 사용 시 5분 동안 요청을 호출한 유니크한 사용자의 수를 표시합니다.
- 1시간 통계 사용 시 1시간 동안 요청을 호출한 유니크한 사용자의 수를 표시합니다.

### TPS

**TPS**(Transaction Per Second)는 초당 처리된 트랜잭션 건수를 의미합니다.

### 응답 시간

요청에 대한 응답 시간의 평균을 의미합니다. 평균이 높은 경우 특정 시점에 느린 트랜잭션이 많이 수행되었거나 또는 장애 발생 가능성이 높습니다.

### CPU

애플리케이션 서버의 CPU 사용량을 의미합니다. 사용량 변화의 추이를 확인할 수 있습니다.

## 힙 메모리

모니터링 대상 프로세스의 [힙 메모리](#) 사용량을 의미합니다. 시간에 따른 메모리 사용량 변화를 확인할 수 있습니다.

## 액티브 트랜잭션

액티브 트랜잭션은 진행 중인 트랜잭션을 의미합니다. 액티브 트랜잭션에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

## 트랜잭션 Top 10

트랜잭션 호출 통계를 사용해 트랜잭션 호출이 많은 상위 10개 목록을 제공합니다.

## HTTP 호출 Top 10

HTTP 통계통계를 사용해 HTTP Call URL의 건수가 많은 상위 10개 목록을 제공합니다.

## SQL Top 10

SQL 통계를 사용해 SQL 호출 건수가 많은 상위 10개 목록을 제공합니다.

# 스택

홈 화면 > 프로젝트 선택 > 분석 > 스택

와탭 모니터링 서비스 초기 화면에서 프로젝트를 선택한 다음 [프로젝트 메뉴](#) 하위에 [분석 > 스택](#)을 선택하세요. [탑 스택](#)과 [유니크 스택](#), [액티브 스택](#)을 확인할 수 있습니다.

① 스택 분석 기능을 사용할 수 있는 애플리케이션은 **Java**와 **Python**, **.NET**입니다.

와탭은 10초(기본값) 간격으로 수집한 스레드 스택을 활용하여 메소드 레벨의 성능 지연 구간을 분석합니다.

```
java.net.SocketInputStream.socketRead0(Native Method)
java.net.SocketInputStream.socketRead0(Native Method)
java.net.SocketInputStream.read(SocketInputStream.java:152)
java.net.SocketInputStream.read(SocketInputStream.java:122)
oracle.net.ns.Packet.receive(Packet.java:311)
oracle.net.ns.DataPacket.receive(DataPacket.java:105)
oracle.net.ns.NetInputStream.getNextPacket(NetInputStream.java:305)
oracle.net.ns.NetInputStream.read(NetInputStream.java:249)
oracle.net.ns.NetInputStream.read(NetInputStream.java:171)
oracle.net.ns.NetInputStream.read(NetInputStream.java:89)
oracle.jdbc.driver.T4CSocketInputStreamWrapper.readNextPacket(T4CSocketInputStreamWrapper.java:123)
oracle.jdbc.driver.T4CSocketInputStreamWrapper.read(T4CSocketInputStreamWrapper.java:79)
oracle.jdbc.driver.T4CMAREngineStream.unmarshaling(LUBI(T4CMAREngineStream.java:426)
oracle.jdbc.driver.T4CTTIFun.receive(T4CTTIFun.java:390)
oracle.jdbc.driver.T4CTTIFun.doRPC(T4CTTIFun.java:249)
oracle.jdbc.driver.T4C8oall.doALL(T4C8oall.java:566)
oracle.jdbc.driver.T4CPreparedStatement.doAll8(T4CPreparedStatement.java:215)
oracle.jdbc.driver.T4CPreparedStatement.doAll8(T4CPreparedStatement.java:58)
oracle.jdbc.driver.T4CPreparedStatement.executeForDescribe(T4CPreparedStatement.java:776)
oracle.jdbc.driver.OracleStatement.executeMaybeDescribe(OracleStatement.java:897)
oracle.jdbc.driver.OracleStatement.doExecuteWithTimeout(OracleStatement.java:1026)
oracle.jdbc.driver.OraclePreparedStatement.executeInternal(OraclePreparedStatement.java:3820)
oracle.jdbc.driver.OraclePreparedStatement.execute(OraclePreparedStatement.java:3923)
oracle.jdbc.driver.OraclePreparedStatementWrapper.execute(OraclePreparedStatementWrapper.java:1385)
org.apache.ibatis.executor.statement.PreparedStatementHandler.query(PreparedStatementHandler.java:56)
org.apache.ibatis.executor.statement.RoutingStatementHandler.query(RoutingStatementHandler.java:70)
org.apache.ibatis.executor.SimpleExecutor.doQuery(SimpleExecutor.java:57)
org.apache.ibatis.executor.BaseExecutor.queryFromDatabase(BaseExecutor.java:259)
org.apache.ibatis.executor.BaseExecutor.query(BaseExecutor.java:132)
org.apache.ibatis.executor.CachingExecutor.query(CachingExecutor.java:105)
org.apache.ibatis.executor.CachingExecutor.query(CachingExecutor.java:81)
org.apache.ibatis.session.defaults.DefaultSqlSession.selectList(DefaultSqlSession.java:104)
org.apache.ibatis.session.defaults.DefaultSqlSession.selectList(DefaultSqlSession.java:98)
sun.reflect.GeneratedMethodAccessor108.invoke(Unknown Source)
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
java.lang.reflect.Method.invoke(Method.java:606)
org.mybatis.spring.SqlSessionTemplate$SqlSessionInterceptor.invoke(SqlSessionTemplate.java:354)
com.sun.proxy.$Proxy62.selectList(Unknown Source)
org.mybatis.spring.SqlSessionTemplate.selectList(SqlSessionTemplate.java:194)
```

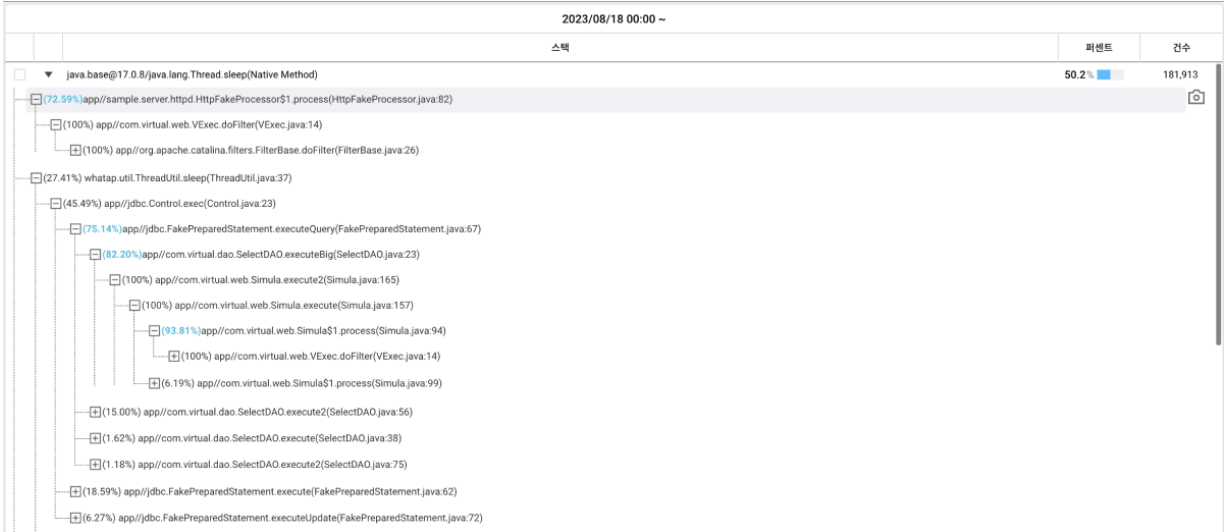
예시 스택에서 **탑 라인**은 `socketRead0` 입니다.

```
java.net.SocketInputStream.socketRead0(Native Method)
```

**탑 라인**은 덤퍼를 수행할 스레드가 해당 메소드를 수행 중이라는 것을 의미합니다. 순간적으로 잡혔을 가능성도 있지만 확률적으로 해당 모듈 처리 시간의 합의 비율만큼 스택에 나타납니다. 이 **탑 라인** 메소드의 빈도를 계산하여 메소드 레벨의 성능을 판단할 수 있습니다. 와탭은 **탑 라인 빈도 통계**를 **탑 스택**(Top Stack)이라고 합니다.

2023/08/18 00:00 ~			
스택		퍼센트	건수
<input type="checkbox"/>	▶ java.base@17.0.8/java.lang.Thread.sleep(Native Method)	50.2%	181,913
<input type="checkbox"/>	▶ java.base@17.0.8/sun.nio.ch.SocketDispatcher.read0(Native Method)	45.1%	163,369
<input type="checkbox"/>	▶ app//com.virtual.web.Simula.execute3(Simula.java:175)	4.64%	16,825
<input type="checkbox"/>	▶ java.base@17.0.8/java.lang.Thread.start0(Native Method)	0.01%	25
<input type="checkbox"/>	▶ java.management@17.0.8/sun.management.ThreadImpl.getThreadTotalCpuTime0(Native Method)	0%	18
<input type="checkbox"/>	▶ app//com.virtual.dao.SelectDAO.execute2(SelectDAO.java:79)	0%	17

**탑 스택** 분석을 통해 도출된 메소드를 어떤 메소드가 호출했는지에 대한 빈도를 분석할 수 있습니다.



탑 스택 계층 분석에서는 원래 액티브 스택을 확인하기 어려웠습니다. 따라서 와탭은 **액티브 스택(Active Stack)**을 조회할 수 있도록 동일 스택을 모아서 **유니크 스택(Unique Stack)**이라는 조회 기능을 제공합니다.

## 탑 스택

Stack Trace 상의 각 Step 기준으로, Step과 Step 간의 호출 비율을 백분율로 분석한 정보를 제공합니다. 최상위 Step의 적체 빈도를 백분율로 산출하여 내림차순으로 정렬한 결과를 제시합니다.

각 Step을 클릭하면 해당 Step을 호출하는 상위 Step을 호출 빈도 기준 백분율로 산출하여 제공합니다.

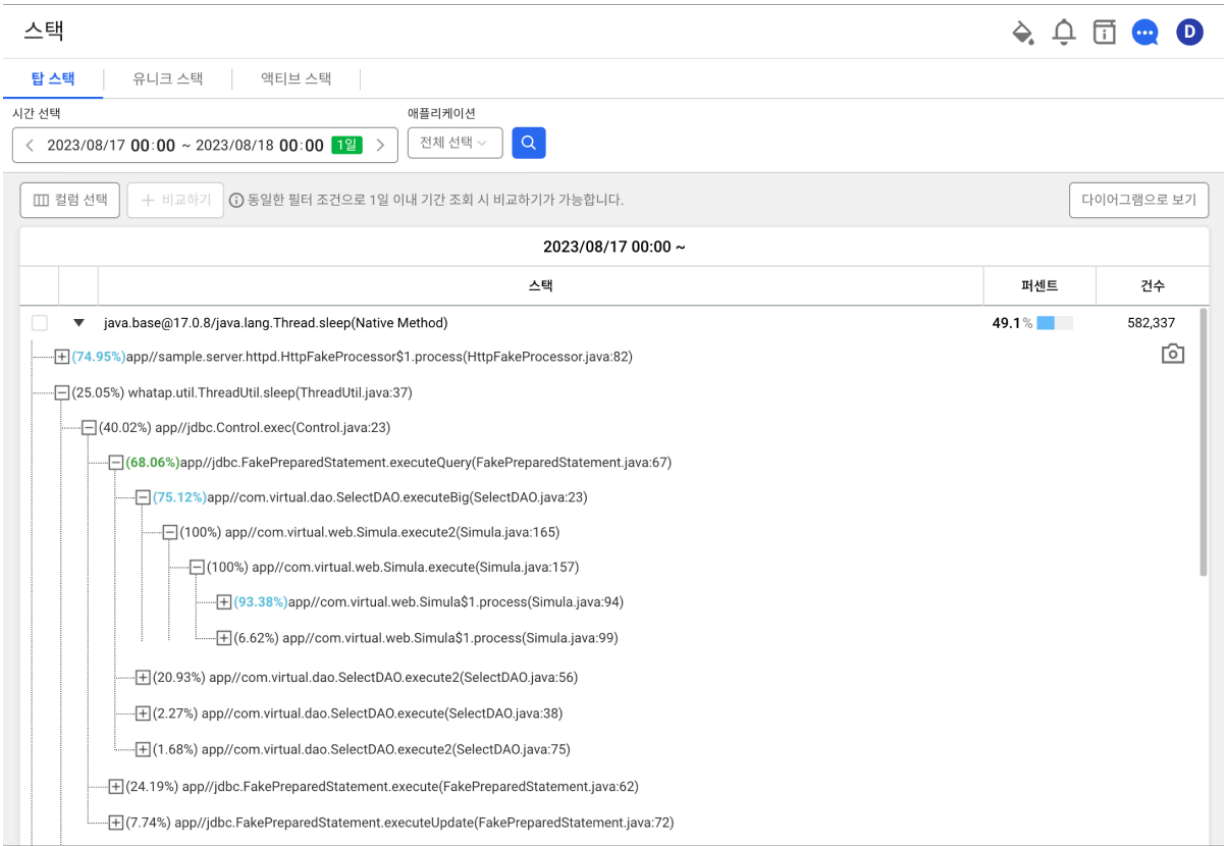
탑 스택 통계는 충분히 많은 데이터를 가지고 판단하는 것이 좋습니다. 수집한 스택의 개수가 10개 미만의 소수인 경우 통계 의미를 갖기에 부족합니다.

탑 스택은 튜닝 시 인지하기 힘들었던 부분의 튜닝 포인트를 찾아내는 데 유용합니다. 가장 빈번하게 나타난 스택은 현재 애플리케이션 서버에서 가장 많은 응답 지연을 발생하는 것으로 판단할 수 있습니다. 가장 왼쪽의 나타나는 비율은 애플리케이션 서버 성능에 영향을 미치는 정도입니다.

안정적인 애플리케이션 서버일지라도 빈번하게 나타난 스택은 성능 저하를 일으킬 가능성이 있으므로 해당 클래스는 유심히 보는 것이 좋습니다.

탑 스택 클릭 시 해당 최상위 스택에 대한 호출 빈도를 확인할 수 있습니다. 탑 스택의 호출 관계는 1 대 1 관계이므로 탑 스택의 depth가 밑으로 내려갈수록 정보의 정확성이 떨어질 수 있습니다. 하위 depth에 대한 정보는 참고 용도로 사용하며 튜닝을 진행하시기 바랍니다.

애플리케이션 성능 개선을 위해 최상위 Step의 적체 비율이 높은 모듈의 병목 가능성을 검토해야 합니다. 적체 비율이 높은 모듈의 경우 작은 성능 개선도 애플리케이션 전체에 상당한 개선 효과를 가져올 수 있습니다.



```

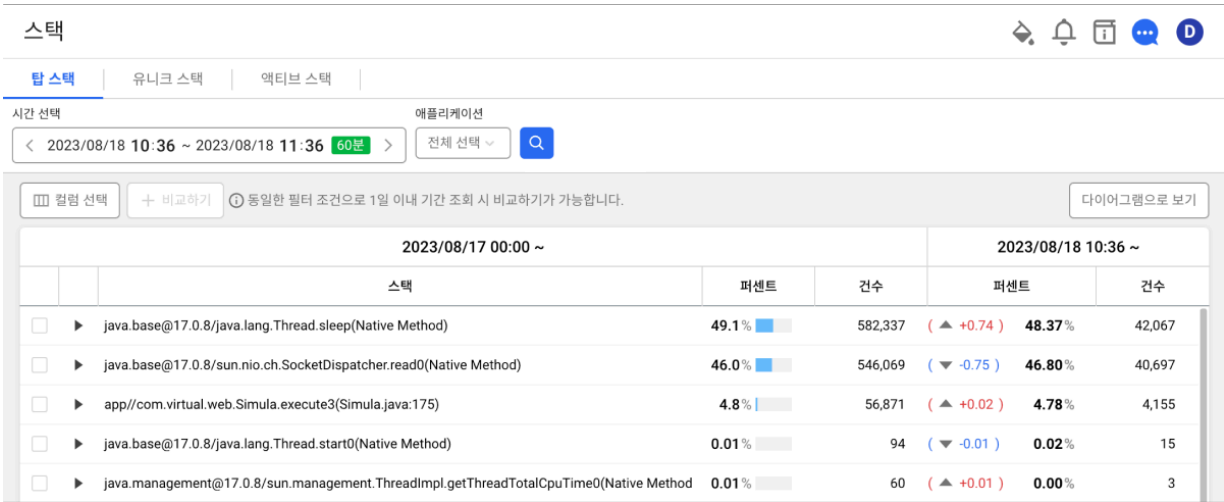
whatap.util.ThreadUtil.sleep
// jdbc.Control.exec의 호출 비율은 40.02%
jdbc.Control.exec
// jdbc.FakePreparedStatement.executeQuery의 호출 비율은 68.06%

```

whatap.util.ThreadUtil.sleep ← jdbc.Control.exec ← jdbc.FakePreparedStatement.executeQuery 의 호출 비율이 40.02% \* 68.06%를 의미하지는 않습니다. jdbc.Control.exec 에서 타 모듈의 호출 가능성이 존재하기 때문입니다.

탐 스택을 통해 호출 비율을 판단할 경우, 각 Step 간 호출 비율을 곱하여 전체 호출 관계 비율을 산출해서는 안 됩니다. Top Stack의 호출 비율은 Stack Trace 상에 노출된 정보의 Step 간 호출 비율의 산출 결과이기 때문에, Step 간 호출 비율로 전체 호출 비율을 도출할 경우 왜곡된 결과를 도출하게 됩니다.

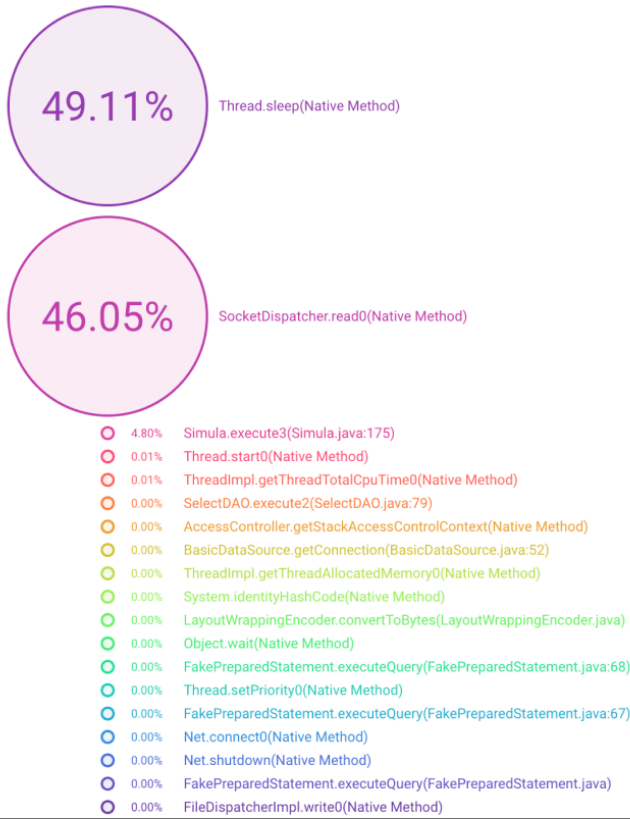
탐 스택 통계에서는 일정 기간을 기준으로 시간에 따른 비율 변화와 수집 건수에 대한 히스토리를 제공합니다.



- 퍼센트
  - 조회 기간 선택된 탑 스택의 비율 변화를 나타냅니다.
  - 장애 시점 현황 파악, 개선 전/후 비교에 유용하게 사용됩니다.
- 건수
  - 수집되는 스택의 수는 액티브 트랜잭션 수에 비례합니다.
  - 특정 구간에서 수집량이 증가했다면 서비스 지연이나 급격한 유입량 증가가 있었음을 알 수 있습니다.

다음과 같은 다이어그램으로도 확인 가능합니다.

\*마우스 스크롤과 드래그로 움직일 수 있습니다



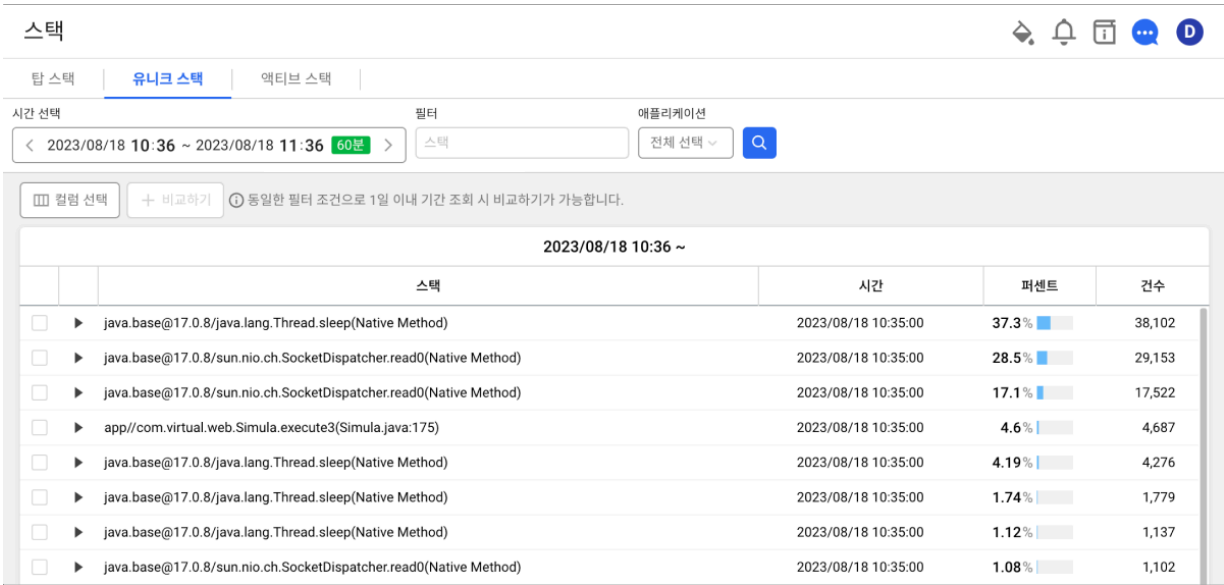
Stack Chart

개별 탑 스택의 비율을 차트로 나타냅니다.

### 유니크 스택

Stack Trace 전체의 Hash 값 기준의 산출 결과로 전체 Step이 동일한 호출 비율을 백분율로 분석한 정보를 제공합니다. 탑 스택은 Step 간의 호출 비율에 대한 정보를 제공합니다.

유니크 스택은 Stack Trace 전체의 정확한 호출 정보를 기반으로 한 데이터를 제공합니다. 상세 호출 관계를 파악하는 데 유용한 정보를 제공합니다. 예를 들면 적체 비율이 높은 Stack Trace를 식별할 수 있습니다.



상세 호출 Step 검토를 통한 호출 경로 상에 이상 모듈의 존재 여부를 파악할 수 있습니다.

### 액티브 스택

진행 중인 트랜잭션을 액티브 트랜잭션이라고 합니다. 액티브 트랜잭션에서 정기적으로 덤프한 스택을 액티브 스택이라 합니다.

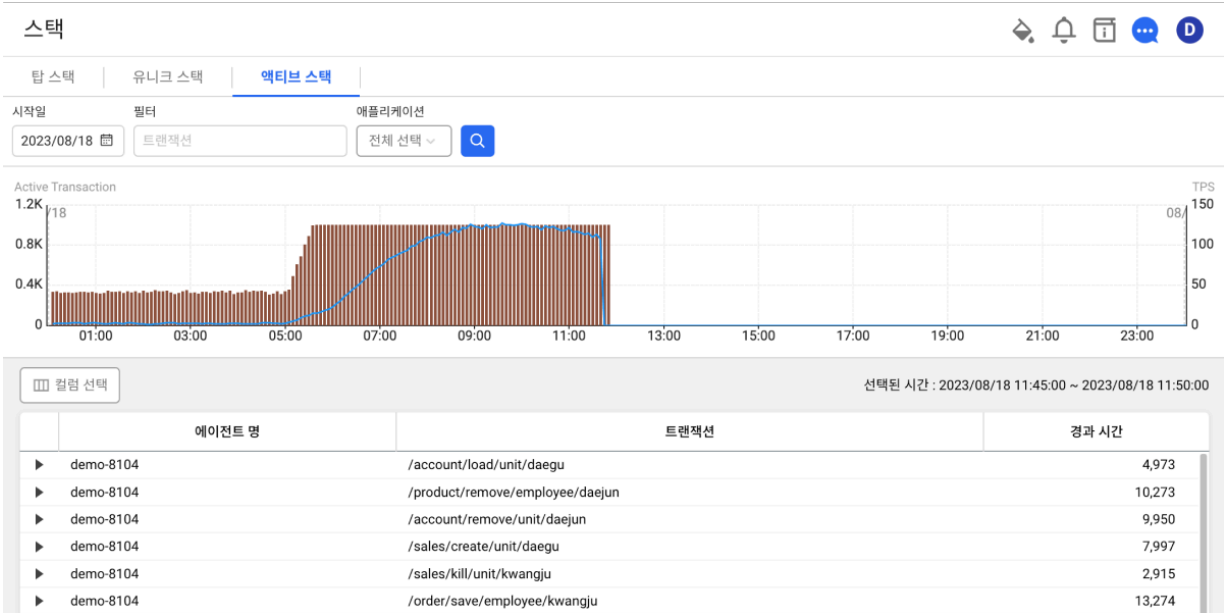
① 와탭 에이전트는 매 10초(옵션 가능)마다 액티브 트랜잭션에서 액티브 스택을 덤프하고 이것을 서버에 전송합니다.

```
active_stack_second=10
```

액티브 스택 메뉴를 선택하면 수집된 ActiveStack을 차트로 확인할 수 있습니다. 차트는 5분간의 단위 통계 데이터를 Active Transaction의 수를 막대로, TPS를 꺾은 선으로 표시합니다.

막대를 클릭하면, 클릭한 시간대의 Active Transaction의 정보가 나오며, 그 정보를 클릭하면 해당 Transaction의 Active Stack을 볼 수 있습니다.





① 액티브 트랜잭션에 관한 자세한 내용은 [다음 문서](#)를 참조하세요.

## 컴팩트한 액티브 스택 수집

액티브 스택은 스레드 덤프를 정기적으로 수행하기 때문에 잘못 구현되면 에이전트에 오버헤드가 커질 수 있습니다. 와탭은 에이전트 부하를 최소화하면서 액티브 스택을 수집하기 위해 다양한 옵션들을 가지고 있습니다.

✔ [사이트맵](#) > [스레드 목록/덤프](#) 메뉴에서 스레드 목록 중에 [WhaTap-ActiveStackDump](#) 스레드의 CPU Time을 확인하면 오버헤드를 판단할 수 있습니다.

### 액티브 스택 예시

```

JAVA

java.lang.StringBuffer.append(StringBuffer.java:309)
java.util.regex.Matcher.appendReplacement(Matcher.java:839)
java.util.regex.Matcher.replaceAll(Matcher.java:906)
java.lang.String.replaceAll(String.java:2162)
core.log.triggers.TriggerRegister.changeNotify(TriggerRegister.java:114)
core.log.aop.handler.DaoInfo.log(DaoInfo.java:141)
core.log.aop.handler.DaoInfo.doAround(DaoInfo.java:102)
core.log.aop.reflection.profiler.AroundProfiler.invoke(AroundProfiler.java:19)
com.sun.proxy.$Proxy39.getUpdateCount(Unknown Source)
org.apache.ibatis.executor.resultset.DefaultResultSetHandler.getNextResultSet(DefaultResultSetHandler.java:256)
org.apache.ibatis.executor.resultset.DefaultResultSetHandler.handleResultSets(DefaultResultSetHandler.java:193)
org.apache.ibatis.executor.statement.PreparedStatementHandler.query(PreparedStatementHandler.java:64)

***

sun.reflect.GeneratedMethodAccessor140.invoke(Unknown Source)
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
java.lang.reflect.Method.invoke(Method.java:606)
org.springframework.web.method.support.InvocableHandlerMethod.doInvoke(InvocableHandlerMethod.java:221)
org.springframework.web.method.support.InvocableHandlerMethod.invokeForRequest(InvocableHandlerMethod.java:136)
org.springframework.web.servlet.mvc.method.annotation.ServletInvocableHandlerMethod.invokeAndHandle(ServletInvocableHandlerMethod.java:114)
org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter.invokeHandlerMethod(RequestMappingHandlerAdapter.java:827)
    
```

### 최적화된 데이터 수집

- 트랜잭션을 수행 중인 스레드에 대해서만 스택을 덤프합니다.
- 액티브 스택 덤프 시간 간격을 조정할 수 있습니다.

```
active_stack_second=10
```

- 액티브 스택의 최대 라인에 제한되어 있습니다. Top 라인에서부터 기본 50라인을 수집합니다.

```
trace_active_callstack_depth=50
```

- 액티브 스택의 각 라인은 해시 처리되어 수집됩니다. text는 한 번만 수집됩니다.
- 한 타임에 수집되는 최대 액티브 스택 개수도 제한되어 있습니다.

```
active_stack_count=100
```

## Background Thread에 대한 액티브 스택

기본적으로 액티브 스택은 트랜잭션이 수행되고 있는 스레드의 스택을 말합니다. 하지만 일부 백그라운드 스레드에 대해서도 스택을 분석할 필요가 있을 수 있습니다. 이때 옵션을 통해서 백그라운드 스레드에 대한 액티브 스택을 확보할 수 있습니다. 1.6.2 버전 이후부터 가능합니다.

- `async_stack_enabled`의 값을 `true`로 설정하면 활성화됩니다.

```
async_stack_enabled=false
```

- 스택 덤프 간격은 포그라운드 액티브 스택 설정에 따라갑니다.

```
active_stack_second=10
```

- 대상 스레드 이름을 지정할 때는 `*`를 사용하여 문자열 패턴을 지정합니다.

```
async_thread_match=http*.abc*
```

스레드 이름으로 스택 덤프 대상을 식별합니다. `,`를 사용하여 match를 여러 개 지정할 수 있습니다.

- 스택의 Top 메소드가 `async_thread_parking`에 등록된 클래스 / 메소드일 때 스레드가 파킹 상태에 있다고 판단하고 덤프를 생성하지 않습니다.

```
async_thread_parking_class=sun.misc.Unsafe
async_thread_parking_method=park
```

# 큐브

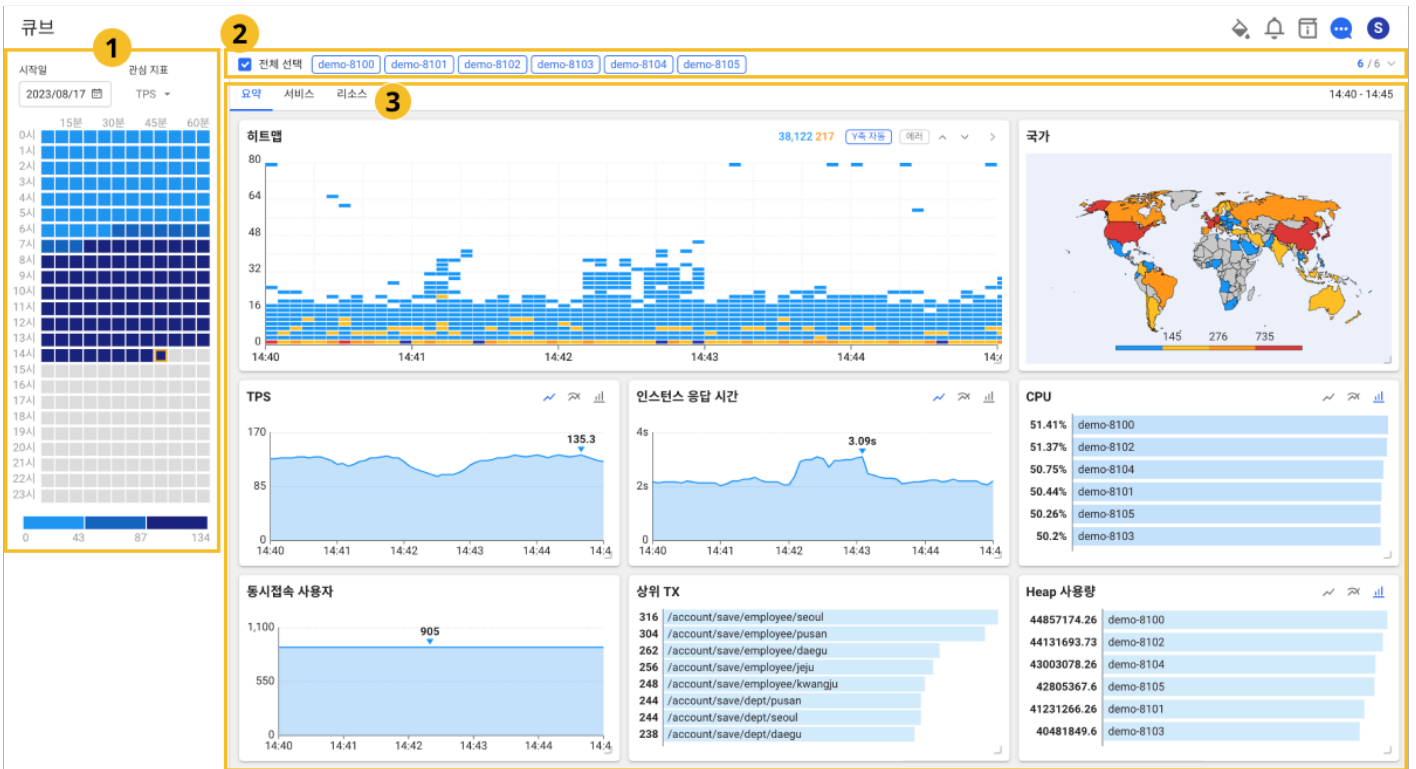
홈 화면 > 프로젝트 선택 > 분석 > 큐브

국가별 접속, 히트맵, 트랜잭션, TOP 트랜잭션, TPS, 응답 시간, 자원 사용 정보를 제공합니다.

- 시간 경과에 따른 애플리케이션 성능 추세를 파악할 수 있습니다. 5분 간격으로 처리된 통계 데이터를 사용해 애플리케이션 성능의 변화를 빠르게 파악할 수 있습니다.
- 트래픽의 지리적 분포를 파악할 수 있습니다. 애플리케이션에 대한 트래픽 데이터를 통해 트래픽의 지리적 분포를 파악하고, 지역별 성능을 비교할 수 있습니다.
- 리소스 사용량을 파악할 수 있습니다. 애플리케이션이 사용하는 리소스 사용량을 파악하고 성능 저하의 원인을 분석할 수 있습니다.

## 큐브란?

와탭은 5분 단위로 만든 성능 통계를 **큐브**라고 부릅니다. 큐브 분석은 큐브에 저장된 5분 단위 성능 데이터를 활용한 분석 기능입니다.



## 1 큐브 셀렉트 패널

왼쪽 ① 영역의 큐브 셀렉트 패널에서 특정 시간 기준으로 큐브를 선택하세요. 큐브는 5분 단위로 저장되어 있기 때문에 특정 시간을 선택해야 합니다.

- 패널 관심 지표

- TPS
- 응답시간
- TX Error 건수
- 액티브 TX 건수
- 동시접속 사용자

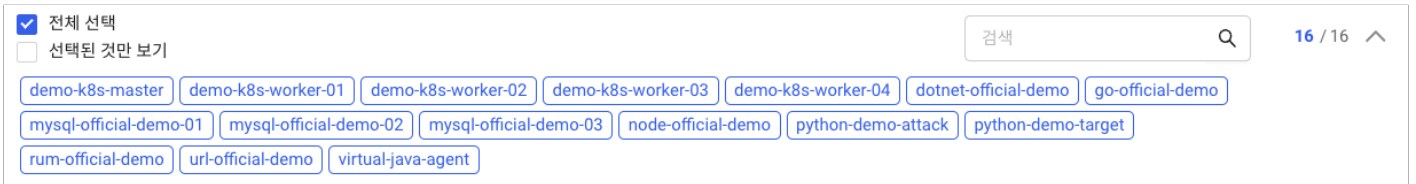
- 큐브 표시색

큐브는 선택된 지표의 수치에 따라 다른 색으로 표현됩니다.

- 남색: 높은 수치
- 하늘색: 낮은 수치

## 2 애플리케이션 셀렉트

상단 ② 영역의 애플리케이션 셀렉트 옵션을 통해 특정 애플리케이션의 데이터를 선택할 수 있습니다. 오른쪽의 ∨ 버튼을 선택하면 다음과 같이 선택 가능한 전체 애플리케이션을 확인할 수 있습니다.



- 전체 선택:** 모든 애플리케이션을 조회합니다. 기본은 전체 선택과 활성 애플리케이션이 선택되어 있습니다.
- 활성 애플리케이션:** 애플리케이션 중 활성화된 애플리케이션을 지정해 조회합니다.
- 선택된 것만 보기:** 애플리케이션 중 사용자가 지정한 해당 애플리케이션만 조회합니다.
- 검색:** 애플리케이션이 많은 경우 입력창을 통해 해당 애플리케이션을 검색할 수 있습니다.

- 숫자/숫자: 지정한 애플리케이션 개수/전체 애플리케이션 개수를 표시합니다.

### 3 큐브 데이터

3 영역에서 해당 큐브 데이터를 [히트맵](#), [국가](#), [TPS](#), [인스턴스 응답 시간](#), [CPU](#), [동시접속 사용자](#), [상위 TX](#), [Heap 사용량](#) 등의 차트를 통해 확인할 수 있습니다.

- [히트맵](#)

큐브 시간 동안 트랜잭션 분포를 보여줍니다. 트레이스 상세 분석이 가능합니다.

> 버튼을 선택하면 큐브 시간 동안 [탭 스택](#)과 [유니크 스택](#)을 조회할 수 있습니다.

❗ 스택 분석 기능을 사용할 수 있는 애플리케이션은 **Java**와 **Python**입니다.

- [국가](#)

클라이언트 아이피를 기준으로 국가를 매핑하여 어느 국가에서 트랜잭션이 들어오는지 보여줍니다. 지도의 색은 트랜잭션의 양을 상대적으로 표시합니다.

❗ 와탭은 클라이언트와 관련한 정보를 기본 저장합니다. 사용자 데이터 수집과 관련한 에이전트 설정에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

- [TPS](#)

Transaction Per Second(TPS)는 초당 처리한 트랜잭션 건수를 의미합니다.

- [인스턴스 응답 시간](#)

트랜잭션 평균 응답시간입니다.

- [CPU](#)

프로젝트 내 애플리케이션(에이전트)를 IP로 정렬하여 사용량이 많은 서버의 CPU를 바 차트로 보여줍니다.

- [동시접속 사용자](#)




실시간 브라우저 사용자 수를 보여줍니다.

- [상위 TX](#)

호출 건수가 많은 트랜잭션 5건을 보여줍니다. 트랜잭션 통계에서는 좀 더 상세한 데이터를 확인할 수 있습니다.

- [Heap 사용량](#)

애플리케이션(인스턴스) 중 힙 메모리 사용량이 많은 순서대로 인스턴스를 정렬해 표시합니다.

- ① • 차트 상단 오른쪽의  아이콘을 선택하면 병합 선형 차트를 제공합니다.
- 차트 상단 오른쪽의  아이콘을 선택하면 개별 선형 차트를 제공합니다.
- 차트 상단 오른쪽의  아이콘을 선택하면 바형 차트를 제공합니다.

- ① WhaTap uses the IP2Location LITE database for [IP geolocation](#).

# 멀티 트랜잭션 추적

멀티 트랜잭션은 다른 에이전트나 프로젝트와 연관된 트랜잭션을 의미합니다. 와탭 프로젝트에 등록된 애플리케이션 서비스 간의 호출을 추적하는 것이 멀티 트랜잭션 추적입니다.

❗ Java 에이전트는 세 개의 HTTP 헤더 키값( `x-wtap-po` , `x-wtap-mst` , `x-wtap-sp1` )으로 멀티 트랜잭션을 추적합니다. 게이트웨이를 통과하는 HTTP 트랜잭션이 연계 추적이 안 된다면 HTTP 헤더 조건을 확인하세요.

## ❗ 멀티 트랜잭션 활성화

멀티 트랜잭션을 추적하려면 **관리 > 에이전트 설정** 메뉴에서 `mtrace_enabled` 옵션을 `true` 로 설정하세요. 에이전트 설정에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

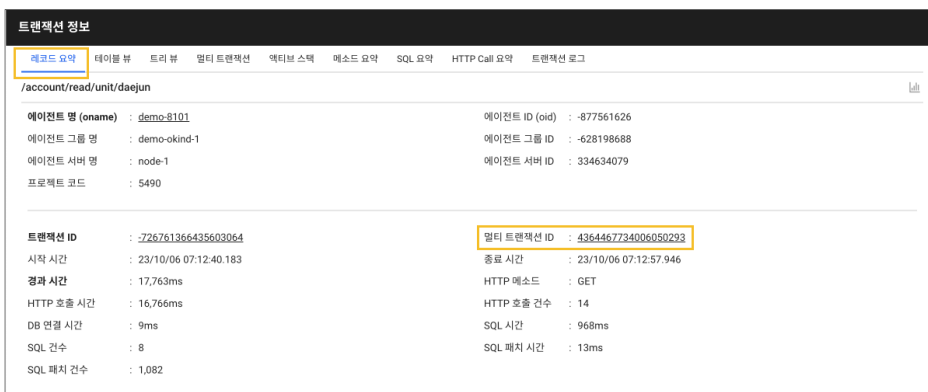
# 멀티 트랜잭션 ID 확인하기

멀티 트랜잭션 추적 메뉴를 이용하려면 MTID(Multi Transaction ID)가 필요합니다. 다음 과정을 통해 MTID 값을 확인할 수 있습니다.

1. **분석 > 히트맵** 메뉴에서 차트 영역을 드래그하세요.
2. 드래그한 차트 영역의 트랜잭션 정보가 다음과 같이 하단 **TX 트레이스** 목록에 나타납니다.

시작 시간	종료 시간	트랜잭션	HTTP 호출	SQL 건수	SQL 시간	SQL 페지 건	DB 연결 시간	에러 클래스	에러 메시지
2023/10/06 09 54 50 327	40,072	/product/delete/unit/seoul	13,359	3	1,554	401	3		
2023/10/06 09 54 42 319	40,212	/product/kill/division/jeju	12,512	6	883	93	259		
2023/10/06 09 56 10 918	40,275	/account/read/unit/daejeon	17,686	2	86	421	40		
2023/10/06 09 54 39 622	40,338	/account/kill/division/seoul	16,513	3	425	1,231	160		
2023/10/06 09 57 15 090	40,481	/account/remove/division/pusan	4,609	4	35,854	1,693	4	SLOW_SQL	SLOW_SQL
2023/10/06 09 54 42 941	40,554	/product/remove/employee/daejeon	12,413	8	444	713	175		
2023/10/06 09 59 29 442	40,649	/account/delete/dept/daejeon	20,626	4	53	426	32		
2023/10/06 10 02 54 198	40,763	/edu/delete/unit/kwangju	15,399	8	156	1,036	10		
2023/10/06 10 02 52 457	41,140	/account/create/employee/kwangju	15,916	4	102	1,483	10		
2023/10/06 10 02 52 815	41,179	/order/remove/division/jeju	15,472	6	323	276	9		
2023/10/06 09 58 59 804	41,278	/sales/read/dept/seoul	18,451	1	0	0	2		
2023/10/06 09 58 38 422	41,664	/account/read/unit/daejeon	13,241	7	7,308	1,817	12		
2023/10/06 09 56 02 316	41,820	/account/remove/dept/pusan	18,508	1	16	631	3		
2023/10/06 09 59 03 822	41,949	/account/kill/employee/kwangju	19,086	2	56	0	2		
2023/10/06 09 54 53 750	41,953	/account/pickup/employee/kwangju	15,812	3	105	865	149		

3. **M** 아이콘이 표시된 트레이스를 선택하면 **트랜잭션 정보** 창이 나타납니다.
4. **레코드 요약** 탭에서 **멀티 트랜잭션 ID** 값을 확인할 수 있습니다.



- ① 트랜잭션에서 외부 호출을 하는 경우에도 동일한 **멀티 트랜잭션 ID**가 생성됩니다. 서비스별로 프로젝트가 분리되어 있더라도 처음 발급한 **멀티 트랜잭션 ID**를 통해 애플리케이션 간의 모든 트랜잭션을 확인할 수 있습니다. **트랜잭션 정보** 창을 활용한 트랜잭션 트레이스 상세 분석에 관한 자세한 내용은 [다음 문서](#)를 참조하세요.
- 트랜잭션 정보** 창에서 **멀티 트랜잭션 ID**를 선택하면 **멀티 트랜잭션** 탭으로 이동합니다. 시스템 내 또는 시스템 간에 발생하는 다양한 호출 관계를 한 눈에 파악하고 어느 부분에서 문제가 발생했는지 식별하여 개선할 수 있도록 트랜잭션과 트레이스 정보를 제공합니다.

## 멀티 트랜잭션 추적 기능 이용하기

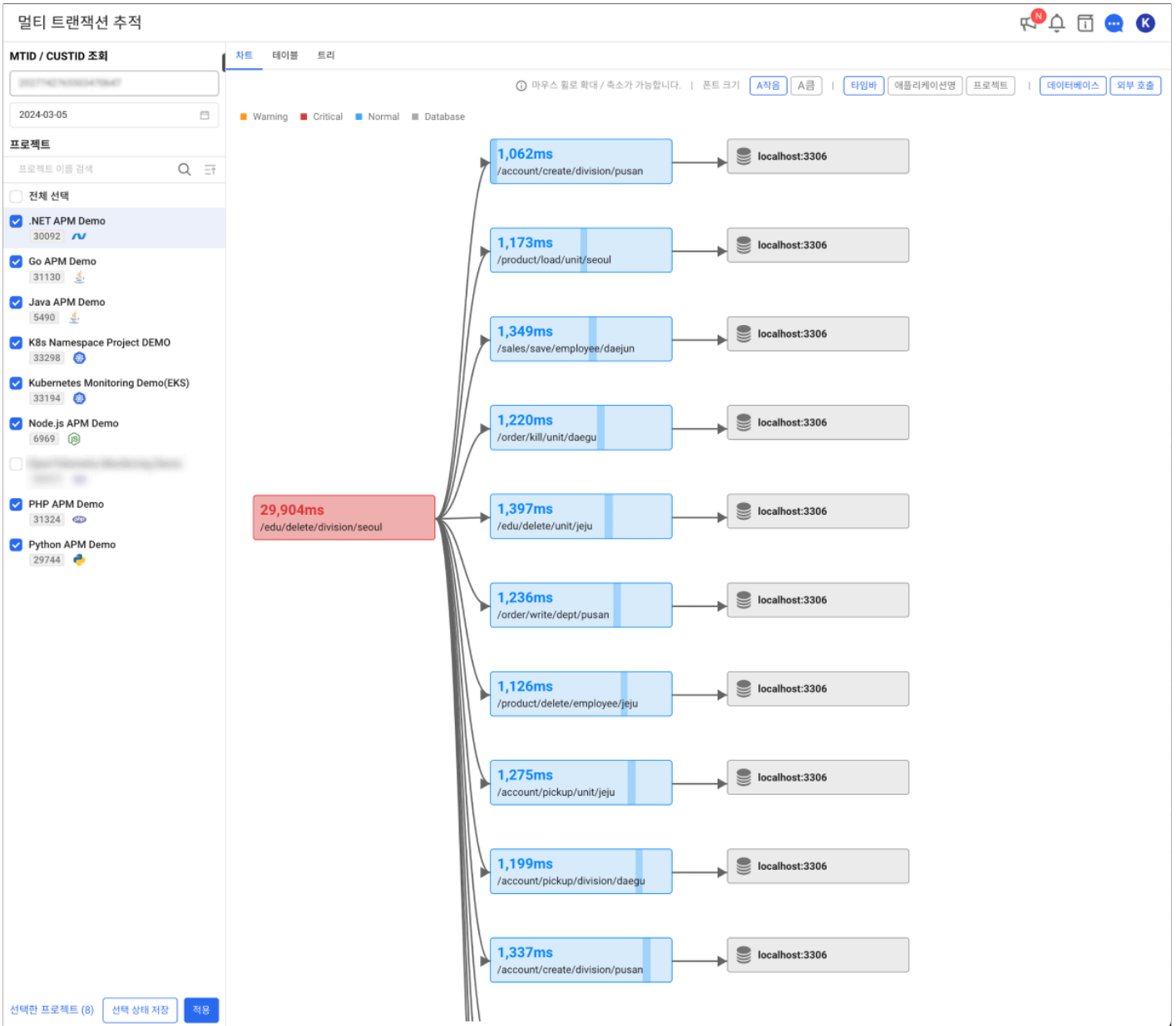
1. **분석 > 멀티 트랜잭션 추적** 메뉴로 이동하세요.
2. **트랜잭션 정보** 창에서 확인한 **MTID** 값을 **MTID / CUSTID 조회** 항목에 입력하세요.



3. 조회할 날짜와 프로젝트를 선택하세요.
4. 화면 아래에 **적용** 버튼을 선택하세요.

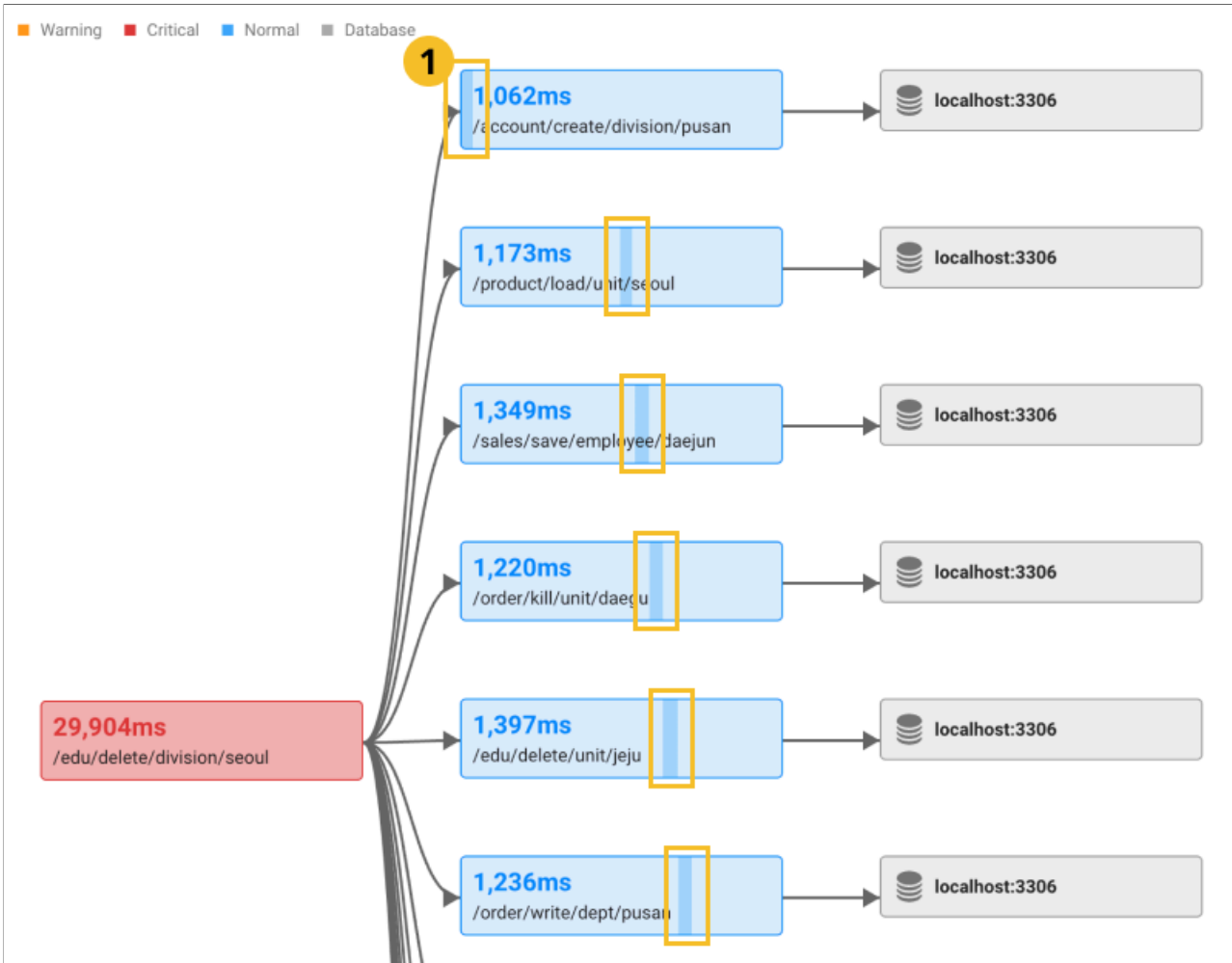
오른쪽 **차트** 탭에 각 트랜잭션의 호출 관계를 파악할 수 있는 다이어그램이 표시됩니다.





## 차트

차트는 각 트랜잭션의 호출 관계를 빠르고 명확하게 사용자에게 제공합니다. 동일한 멀티 트랜잭션 ID를 갖는 트랜잭션 서비스들의 개별 수행 시간을 확인할 수 있습니다. 트랜잭션 노드의 배경색으로 표현되어 있는 소요 시간(1 타임바)를 통해 트랜잭션 간 호출 관계를 확인할 수 있습니다. 차트에서는 마우스를 이용해 원하는 위치로 이동하거나 스크롤을 통해서 확대, 축소할 수 있습니다.



트랜잭션 노드를 선택하면 **트랜잭션 정보** 창이 추가로 나타납니다. 트랜잭션 트레이스를 통해 해당 트랜잭션의 상세 내역을 확인할 수 있습니다. **트랜잭션 정보** 창을 활용한 트랜잭션 트레이스 상세 분석에 관한 자세한 내용은 [다음 문서](#)를 참조하세요

- **A작음 / A큼**: 노드에 표시된 폰트 크기를 조절할 수 있습니다.
- **타임바**: 각 노드에 소요 시간(타임바)을 표시하거나 숨길 수 있습니다.
- **애플리케이션명**: 각 노드에 에이전트 이름(oname)을 표시하거나 숨길 수 있습니다.
- **프로젝트**: 각 노드에 프로젝트 이름을 표시하거나 숨길 수 있습니다.
- **데이터베이스 / 외부 호출 / 내부 호출**: 해당 트랜잭션에서 발생한 다른 데이터베이스 커넥션 요청이나 HTTP Call의 정보 또한 차트의 노드로 확인할 수 있습니다.

## 테이블

테이블 탭에서는 멀티 트랜잭션 내에 포함된 각 트랜잭션 별 정보를 테이블 형식으로 확인할 수 있습니다.

트랜잭션	경과 시간 (ms)	시작 시간	에이전트 명	프로젝트	Caller	트랜잭션 ID	종
/product/create/dept/kwangju	5,874	2024-03-06 03:46:39	demo-8102	5490	5490	-8398204456457444331	2024-0
/edu/write/unit/daegu	1,057	2024-03-06 03:46:44	demo-8102	5490	5490	-8398204456457444331 4491114531979950672	2024-0
/sales/write/dept/daejun	1,058	2024-03-06 03:46:43	demo-8102	5490	5490	-8398204456457444331 1182616852843989315	2024-0
/order/delete/division/daegu	1,229	2024-03-06 03:46:42	demo-8102	5490	5490	-8398204456457444331 -6958668892518915742	2024-0
/account/delete/unit/daejun	1,058	2024-03-06 03:46:41	demo-8102	5490	5490	-8398204456457444331 -7274645409069425203	2024-0
/account/read/division/daegu	1,393	2024-03-06 03:46:39	demo-8102	5490	5490	-8398204456457444331 8054316813226932913	2024-0

각 트랜잭션 항목을 선택하면 차트 탭의 노드 선택과 마찬가지로 선택한 트랜잭션에 대한 트랜잭션 정보 창이 나타납니다. 트랜잭션 트레이스를 통해 해당 트랜잭션의 상세 내역을 확인할 수 있습니다. 트랜잭션 정보 창을 활용한 트랜잭션 트레이스 상세 분석에 관한 자세한 내용은 [다음 문서](#)를 참조하세요.



☐ 컬럼 설정: 테이블 내 컬럼을 편집할 수 있습니다.

## 트리


트리 탭에서는 각 트랜잭션과 그에 속해 있는 트레이스의 세부 정보를 확인할 수 있습니다. 전체 트랜잭션 소요 시간 내의 각 하위 트랜잭션이나 트레이스의 시작 및 소요 시간을 시각화해 트랜잭션 호출 관계를 트리 형식으로 제공합니다.




- ☐ 한 줄 보기: 각 구간 별 수행 정보에 표시된 텍스트를 한 줄로 표시해 트리 형식을 간격하게 정리할 수 있습니다.
- ☑ 여러 줄 보기: 각 구간 별 수행 정보에 표시된 텍스트를 줄바꿈해 모두 표시합니다.
- 🕒 가장 긴 경로: 가장 긴 경로로 이동할 수 있습니다.

-  **시간바 표시**: 경과 시간을 막대 형식의 차트로 표시합니다.
-  **시간 표시**: 각 구간별 타임 스탬프, 갭, 경과 시간을 텍스트 형식으로 표시합니다.
  - 8초 이상: **초과 지연** 상태로 **빨간색**으로 표현합니다.
  - 3초 이상 8초 미만: **지연** 상태로 **주황색**으로 표현합니다.
  - 3초 미만: **정상** 상태로 **파란색**으로 표현합니다.

ⓘ 시작 및 소요 시간의 경우 트랜잭션 호출 환경에 따라 발생하는 시차를 상위 트랜잭션 내 트레이스와의 매핑을 통해 보정하여 표현하기 때문에 실제 수집된 시간 데이터와 차이가 발생할 수 있습니다.

-  **시간 숨기기**: 시간 정보를 숨깁니다.

ⓘ 해당 트레이스의  버튼 또는  버튼을 선택하면 **HTTP 호출 통계**, **액티브 스택** 등의 요약 창을 확인할 수 있습니다.

-  SQL 변수와 HTTP 쿼리를 조회하려면 다음 옵션을 에이전트 설정에 추가하세요.
  - SQL 파라미터 정보 기록과 관련한 에이전트 설정은 [다음 문서](#)를 참조하세요.
  - HTTP 파라미터 정보 기록과 관련한 에이전트 설정은 [다음 문서](#)를 참조하세요.

```
whatap.conf
```

```
# SQL 파라미터 조회 옵션: 옵션이 적용되면 SQL 파라미터를 암호화하여 수집합니다.
```

```
profile_sql_param_enabled=true
```

```
# HTTP 파라미터 조회 옵션: 옵션이 적용되면 HTTP 쿼리 파라미터를 암호화하여 수집합니다.
```

```
profile_http_parameter_enabled=true
```

# 트랜잭션 검색

홈 화면 > 프로젝트 선택 > 분석 > 트랜잭션 검색

특정 기간 동안 각 트랜잭션의 성능 속성에 대한 통계와 실행 상태를 검색할 수 있습니다. 모든 트랜잭션을 확인하고 정상 및 에러 상태를 빠르게 확인할 수 있습니다. 제공되는 정보를 통해 어떤 트랜잭션의 사용량이 많고 리소스를 많이 소비하는지 알 수 있습니다. 트랜잭션의 속성에 따라 다양하게 필터링할 수 있어 특정 조건의 트랜잭션을 찾는 데 유용합니다.

트랜잭션 검색
🔊 🔔 📄 ⋮ 👤

시간 선택 애플리케이션

< 2024/03/06 00:00 ~ 2024/03/06 23:59 1일 > 전체 선택 ~

필터

+ 경과 시간 + 에러 메시지 + HTTP 호출 건수 + SQL 건수 + SQL 패치 건수 + 트랜잭션 + 클라이언트 IP + 도메인 + 유저 에이전트 + Referer + 트랜잭션 ID + 멀티 트랜잭션 ID + WClientID + 로그인 ID  에러만 보기 🔍

📄 CSV

☰ 컬럼 선택

에이전트명 (oname)	시작 시간	종료 시간	↓	경과 시간	트랜잭션	CPU 사용 ...	HTTP 호출 ...	SQL 건수	SQL 패치 건...
<div style="border: 1px solid #007bff; background-color: #e6f2ff; padding: 10px; border-radius: 10px; width: 60%; margin: 0 auto;"> <p style="text-align: center; margin: 0;">🔍</p> <p style="text-align: center; margin: 0;"><b>상세 검색을 위해 필터를 이용해주세요</b></p> <p style="font-size: 0.8em; margin: 0;">여러개의 검색 조건에 의한 데이터 탐색 방식을 지원하지 않습니다. 검색 범위가 지나치게 큰 경우 데이터 추출에 제한이 있습니다.</p> </div>									

조회 시간을 설정한 다음 **애플리케이션** 옵션에서 조회 대상을 선택하세요. 🔍 버튼을 선택하면 검색 결과가 테이블에 표시됩니다. 결과 목록에서 원하는 항목을 선택하면 **트랜잭션 정보** 창이 나타납니다. **트랜잭션 정보** 창을 활용한 트랜잭션 트레이스 상세 분석에 관한 자세한 내용은 [다음 문서](#)를 참조하세요.

- ① • 에러 상태의 트랜잭션만 검색하려면 **에러만 보기**를 클릭한 다음 🔍 버튼을 선택하세요.

ⓘ • 검색 조건에 따라 응답 지연이 발생할 수 있기 때문에 검색 시간을 가급적 짧게 설정할 것을 권고합니다.

## 컬럼 편집하기

전체 선택   전체 해제   초기화(기본값)
×

<input type="checkbox"/> 에이전트	<input type="checkbox"/> 클라이언트	<input type="checkbox"/> 에러	<input type="checkbox"/> HTTP 호출	<input type="checkbox"/> 리소스	<input type="checkbox"/> SQL	<input type="checkbox"/> 트랜잭션
<input type="checkbox"/> 컨테이너 ID	<input type="checkbox"/> 도시	<input type="checkbox"/> 에러 클래스	<input checked="" type="checkbox"/> HTTP 호출 건수	<input checked="" type="checkbox"/> CPU 시간	<input type="checkbox"/> DB 접속시간	<input type="checkbox"/> 활성화
<input type="checkbox"/> 에이전트 ID	<input type="checkbox"/> Client OS Pro	<input type="checkbox"/> 에러 단계	<input type="checkbox"/> HTTP 호출 시간	<input type="checkbox"/> 메모리 할당	<input checked="" type="checkbox"/> SQL 패치 건수	<input type="checkbox"/> 커스텀 TID
<input type="checkbox"/> 에이전트 그룹 ID	<input type="checkbox"/> 국가	<input type="checkbox"/> 에러 메시지			<input type="checkbox"/> 패치 시간	<input type="checkbox"/> 도메인
<input type="checkbox"/> 에이전트 그룹	<input type="checkbox"/> IP				<input type="checkbox"/> URL 해시	<input checked="" type="checkbox"/> 경과 시간
<input checked="" type="checkbox"/> 에이전트 명	<input type="checkbox"/> 로그인				<input checked="" type="checkbox"/> SQL 건수	<input checked="" type="checkbox"/> 종료 시간
<input type="checkbox"/> 에이전트 노드 ID	<input type="checkbox"/> 유저 에이전트				<input type="checkbox"/> SQL 시간	<input type="checkbox"/> Multi Caller
<input type="checkbox"/> 에이전트 노드	<input type="checkbox"/> User Agent Hash					<input type="checkbox"/> Call Depth
<input type="checkbox"/> 프로젝트 코드	<input type="checkbox"/> 사용자 종류					<input type="checkbox"/> HTTP 메소드
<input type="checkbox"/> Pod 명	<input type="checkbox"/> 클라이언트 ID					<input type="checkbox"/> 멀티 트랜잭션 ID
<input type="checkbox"/> Sub Type						<input type="checkbox"/> referer
						<input checked="" type="checkbox"/> URL
						<input checked="" type="checkbox"/> 시작 시간

결과 조회 시 컬럼 선택 버튼을 클릭하면 확인이 필요한 컬럼을 적절하게 선택할 수 있습니다.

## 필터 적용하기

상세 검색을 위해 필터를 이용하세요. 여러 개의 검색 조건에 의한 데이터 탐색 방식을 지원합니다. 필터를 설정한 다음 버튼을 선택하세요.

ⓘ 일부 필터 항목은 상품에 따라 지원하지 않을 수 있습니다.

## 경과 시간

트랜잭션 검색

시간 선택 < 2024/03/06 00:00 ~ 2024/03/06 23:59 **1일** > 전체 선택

필터

경과 시간 1000 5000 + 에러 메시지 + HTTP 호출 건수 + SQL 건수 + SQL 패치 건수 + 트랜잭션 + 클라이언트 IP + 도메인 + 유저 에이전트 + Referer + 트랜잭션 ID + 멀티 트랜잭션 ID + WClientID

+ 로그인 ID  에러만 보기

🔍 CSV

📄 컬럼 선택

	에이전트 명 (oname)	시작 시간	종료 시간	경과 시간	트랜잭션	CPU 사용 ...	HTTP 호출 ...	SQL 건수	SQL 패치 건...
1	demo-8101	2024/03/06 00:00:00.034	2024/03/06 00:00:01.101	1,067	/account/save/dept/daejun	1	0	11	6,789
2	demo-8104	2024/03/06 00:00:00.176	2024/03/06 00:00:01.333	1,157	/product/remove/dept/pusan	2	0	11	7,112
3	demo-8100	2024/03/06 00:00:00.326	2024/03/06 00:00:01.466	1,140	/account/load/employee/daejun	1	0	11	6,406
4	demo-8103	2024/03/06 00:00:00.419	2024/03/06 00:00:01.485	1,066	/account/delete/dept/kwangju	1	0	11	5,260

트랜잭션 응답시간 기준으로 검색합니다. 기준 이상과 미만의 2개 값을 입력해 검색합니다.

## 에러 메시지

트랜잭션 검색

시간 선택 < 2024/03/06 00:00 ~ 2024/03/06 23:59 **1일** > 전체 선택

필터

+ 경과 시간 + 에러 메시지 Exception + HTTP 호출 건수 + SQL 건수 + SQL 패치 건수 + 트랜잭션 + 클라이언트 IP + 도메인 + 유저 에이전트 + Referer + 트랜잭션 ID + 멀티 트랜잭션 ID + WClientID

+ 로그인 ID  에러만 보기

🔍 CSV

📄 컬럼 선택

	에이전트 명 (oname)	시작 시간	종료 시간	경과 시간	트랜잭션	CPU 사용 ...	HTTP 호출 ...	SQL 건수	SQL 패치 건...	에러 메시지
81	demo-8102	2024/03/06 06:15:13.130	2024/03/06 06:15:18.417	5,287	/edu/remove/dept/daegu	2	4	3	767	Sql Exception
82	demo-8104	2024/03/06 06:15:45.081	2024/03/06 06:15:49.991	4,910	/sales/save/unit/daegu	0	4	3	0	Internal RuntimeExcepti on
83	demo-8100	2024/03/06 06:15:51.329	2024/03/06 06:15:57.407	6,078	/account/save/dept/daejun	2	3	4	0	Sql Exception
84	demo-8104	2024/03/06 06:18:00.875	2024/03/06 06:18:11.402	10,527	/account/pickup/division/jeju	3	8	2	0	Internal RuntimeExcepti on

트랜잭션 에러 메시지를 기반으로 검색합니다. 디폴트 상태에서 에러 관련 컬럼이 보이지 않을 수 있습니다. 컬럼을 선택하세요.

❗ 에러 상태의 트랜잭션은 빨간색으로 표시됩니다.

## HTTP 호출 건수

트랜잭션 검색

시간 선택 < 2024/03/06 00:00 ~ 2024/03/06 23:59 1일 > 전체 선택

필터

+ 경과 시간 + 에러 메시지 - HTTP 호출 건수 3 + SQL 건수 + SQL 패치 건수 + 트랜잭션 + 클라이언트 IP + 도메인 + 유저 에이전트 + Referer + 트랜잭션 ID + 멀티 트랜잭션 ID + WClientID

+ 로그인 ID  에러만 보기

🔍 CSV

📄 컬럼 선택

	에이전트명 (oname)	시작 시간	종료 시간 ↓	경과 시간	트랜잭션	CPU 사용 ...	HTTP 호출 건수	SQL 건수	SQL 패치 건...
1	demo-8104	2024/03/06 00:00:00.176	2024/03/06 00:00:07.006	6,830	/account/read/unit/kwangju	0	4	3	1,139
2	demo-8105	2024/03/06 00:00:01.949	2024/03/06 00:00:07.857	5,908	/product/read/dept/pusan	0	4	4	760
3	demo-8101	2024/03/06 00:00:01.034	2024/03/06 00:00:09.403	8,369	/product/kill/dept/pusan	0	6	8	3,343
4	demo-8102	2024/03/06 00:00:01.650	2024/03/06 00:00:09.475	7,825	/account/delete/division/kwangju	0	5	3	0

트랜잭션 중 외부 HTTP 서비스를 호출한 건수를 기준으로 입력한 자연수 이상에 해당하는 트랜잭션을 검색합니다.

## SQL 건수

트랜잭션 검색

시간 선택 < 2024/03/06 00:00 ~ 2024/03/06 23:59 1일 > 전체 선택

필터

+ 경과 시간 + 에러 메시지 + HTTP 호출 건수 - SQL 건수 1200 + SQL 패치 건수 + 트랜잭션 + 클라이언트 IP + 도메인 + 유저 에이전트 + Referer + 트랜잭션 ID + 멀티 트랜잭션 ID + WClientID

+ 로그인 ID  에러만 보기

🔍 CSV

📄 컬럼 선택

	에이전트명 (oname)	시작 시간	종료 시간 ↓	경과 시간	트랜잭션	CPU 사용 ...	HTTP 호출 ...	SQL 건수	SQL 패치 건수
1	demo-8101	2024/03/06 00:00:00.034	2024/03/06 00:04:50.282	290,248	/sales/read/unit/kwangju	0	9	2,002	992,790
2	demo-8101	2024/03/06 00:24:06.096	2024/03/06 00:29:00.821	294,725	/product/create/division/daejun	327	2	2,002	993,322
3	demo-8103	2024/03/06 00:20:19.152	2024/03/06 00:29:31.244	552,092	/account/write/employee/kwangju	0	2	2,002	1,001,229
4	demo-8103	2024/03/06 00:24:30.544	2024/03/06 00:46:15.140	1,304,596	/edu/read/unit/pusan	352	1	2,000	1,000,654

트랜잭션별 SQL 수행 건수가 기준으로 입력한 자연수 이상에 해당하는 트랜잭션을 검색합니다. 예외적으로 과도한 수의 SQL을 호출하는 트랜잭션을 검색할 때 이용합니다.



## SQL 패치 건수

트랜잭션 검색 🔊 🔔 🗑️ ⋮ 👤

시간 선택 애플리케이션

< 2024/03/06 00:00 ~ 2024/03/06 23:59 ▶ 전체 선택 >

필터

+ 경과 시간 + 에러 메시지 + HTTP 호출 건수 + SQL 건수 **- SQL 패치 건수 500000** + 트랜잭션 + 클라이언트 IP + 도메인 + 유저 에이전트 + Referer + 트랜잭션 ID + 멀티 트랜잭션 ID + WClientID

+ 로그인 ID  에러만 보기

	에이전트 명 (oname)	시작 시간	종료 시간 ↓	경과 시간	트랜잭션	CPU 사용 ...	HTTP 호출 ...	SQL 건수	SQL 패치 건수
1	demo-8103	2024/03/06 00:00:00.419	2024/03/06 00:02:27.984	147,565	/order/pickup/division/jeju 📄	0	5	1,006	505,725
2	demo-8101	2024/03/06 00:00:00.034	2024/03/06 00:04:50.282	290,248	/sales/read/unit/kwangju 📄	0	9	2,002	992,790
3	demo-8102	2024/03/06 00:03:48.478	2024/03/06 00:05:01.167	72,689	/account/load/dept/kwangju 📄	165	15	1,006	507,850
4	demo-8104	2024/03/06 00:03:55.851	2024/03/06 00:05:07.832	71,981	/account/write/unit/jeju 📄	167	15	1,005	502,953

SQL 수행 후 DB 패치 건수를 기준으로 입력한 자연수 이상에 해당하는 트랜잭션을 검색합니다. 과도한 데이터 패치를 수행하는 트랜잭션을 검색할 때 이용합니다.

## 트랜잭션

트랜잭션 검색 🔊 🔔 🗑️ ⋮ 👤

시간 선택 애플리케이션

< 2024/03/06 00:00 ~ 2024/03/06 23:59 ▶ 전체 선택 >

필터

+ 경과 시간 + 에러 메시지 + HTTP 호출 건수 + SQL 건수 **- 트랜잭션 account** + 클라이언트 IP + 도메인 + 유저 에이전트 + Referer + 트랜잭션 ID + 멀티 트랜잭션 ID + WClientID

+ 로그인 ID  에러만 보기

	에이전트 명 (oname)	시작 시간	종료 시간 ↓	경과 시간	트랜잭션	CPU 사용 ...	HTTP 호출 ...	SQL 건수	SQL 패치 건...
1	demo-8101	2024/03/06 00:00:00.034	2024/03/06 00:00:01.101	1,067	/account/save/dept/daejeon 📄	1	0	11	6,789
2	demo-8100	2024/03/06 00:00:00.326	2024/03/06 00:00:01.466	1,140	/account/load/employee/daejeon 📄	1	0	11	6,406
3	demo-8103	2024/03/06 00:00:00.419	2024/03/06 00:00:01.485	1,066	/account/delete/dept/kwangju 📄	1	0	11	5,260
4	demo-8100	2024/03/06 00:00:01.326	2024/03/06 00:00:02.566	1,240	/account/write/unit/daejeon 📄	2	0	11	6,106

트랜잭션을 기준으로 검색합니다. 입력한 검색 문자열은 포함 여부(부분 일치)로 비교 검색합니다.

## 클라이언트 IP

트랜잭션 검색

시간 선택 < 2024/03/06 00:00 ~ 2024/03/06 23:59 1일 > 전제 선택

필터 + 경과 시간 + 에러 메시지 + HTTP 호출 건수 + SQL 건수 + SQL 패치 건수 + 트랜잭션 - 클라이언트 IP 50.182 + 도메인 + 유저 에이전트 + Referer + 트랜잭션 ID + 멀티 트랜잭션 ID + WClientID

+ 로그인 ID  에러만 보기

🔍 CSV

필름 선택

	에이전트명 (oname)	시작 시간	종료 시간 ↓	경과 시간	트랜잭션	클라이언트 IP	CPU 사용 ...	HTTP 호출 ...	SQL 건수	SQL 패치 건...
1	demo-8104	2024/03/06 00:00:05.938	2024/03/06 00:00:06.996	1,058	/account/delete/unit/jeju	50.182.206.151	1	0	11	6,898
2	demo-8105	2024/03/06 00:01:09.381	2024/03/06 00:01:10.530	1,149	/account/remove/division/daegu	50.182.206.151	1	0	11	5,140
3	demo-8101	2024/03/06 00:05:46.335	2024/03/06 00:05:47.728	1,393	/edu/kill/dept/kwangju	50.182.206.151	2	0	11	4,359
4	demo-8101	2024/03/06 00:10:10.715	2024/03/06 00:10:11.958	1,243	/order/read/dept/jeju	50.182.206.151	1	0	11	4,997

클라이언트 IP 기반으로 트랜잭션을 검색합니다. 단, IP 문자열은 PREFIX 비교(전방 일치)로 검색합니다.

## 도메인

트랜잭션 검색

시간 선택 < 2024/03/06 00:00 ~ 2024/03/06 23:59 1일 > 전제 선택

필터 + 경과 시간 + 에러 메시지 + HTTP 호출 건수 + SQL 건수 + SQL 패치 건수 + 트랜잭션 + 클라이언트 IP - 도메인 255.255.255.255 + 유저 에이전트 + Referer + 트랜잭션 ID + 멀티 트랜잭션 ID + WClientID

+ 로그인 ID  에러만 보기

🔍 CSV

필름 선택

	에이전트명 (oname)	시작 시간	종료 시간 ↓	경과 시간	트랜잭션	도메인	CPU 사용 ...	HTTP 호출 ...	SQL 건수	SQL 패치 건...
1	demo-8102	2024/03/06 00:01:37.696	2024/03/06 00:01:47.736	10,040	/edu/kill/employee/kwangju	255.255.255.255	4	8	2	509
2	demo-8102	2024/03/06 00:03:27.111	2024/03/06 00:03:30.926	3,815	/edu/remove/unit/daegu	255.255.255.255	0	2	1	471
3	demo-8104	2024/03/06 00:14:19.100	2024/03/06 00:14:22.568	3,468	/sales/kill/dept/seoul	255.255.255.255	2	3	0	0
4	demo-8101	2024/03/06 00:24:06.096	2024/03/06 00:29:00.821	294,725	/product/create/division/daejun	255.255.255.255	327	2	2,002	993,322

HTTP 서비스 도메인(host:port) 기반으로 트랜잭션을 검색합니다. 검색 조건은 도메인이 완전히 일치(전체 일치)해야 합니다.

## 유저 에이전트

트랜잭션 검색
🔊 🔔 🗑️ ⋮ 👤

시간 선택 애플리케이션

< 2024/03/06 00:00 ~ 2024/03/06 23:59 1일 > 전체 선택

필터

+ 경과 시간 + 에러 메시지 + HTTP 호출 건수 + SQL 건수 + SQL 패치 건수 + 트랜잭션 + 클라이언트 IP + 도메인 - 유저 에이전트 Chrome + Referer + 트랜잭션 ID + 멀티 트랜잭션 ID + WClientID

+ 로그인 ID  예러만 보기

🔍

📄 컬럼 선택

에이전트 명 (oname)	시작 시간	종료 시간 ↓	경과 시간	트랜잭션	CPU 사용 ...	HTTP 호출 ...	SQL 건수	SQL 패치 건수	유저 에이전트	
1 demo-8103	2024/03/06 00:00:00.419	2024/03/06 00:00:01.485	1,066	/account/delete/dept/kwangju 🗑️		1	0	11	5,260	Mozilla/5.0 (X11; U; Linux x86_64; en-US; AppleWebKit/532.2 (KHTML, like Gecko) Chrome/4.0.222.5 Safari/532.2
2 demo-8105	2024/03/06 00:00:01.950	2024/03/06 00:00:03.379	1,429	/account/save/dept/daegu 🗑️		2	0	11	5,293	Mozilla/5.0 (Macintosh; AMD Mac OS X 10.8_2) AppleWebKit/535.22 (KHTML, like Gecko) Chrome/18.6.872

유저 에이전트 기반으로 트랜잭션을 검색(부분 일치)합니다.

## Referer

Referer 문자열 기반으로 트랜잭션을 검색합니다.

- ⓘ 기본 설정 상태의 에이전트는 Referer 정보를 수집하지 않습니다. Referer 설정이 필요한 경우 [다음 문서](#)를 참조해 관련 설정을 추가하세요.

## 트랜잭션 ID

트랜잭션 검색

시간 선택: < 2024/02/28 00:00 ~ 2024/02/29 00:00 **1일** > 전체 선택 ~

필터: + 경과 시간 + 에러 메시지 + HTTP 호출 건수 + SQL 건수 + SQL 패치 건수 + 트랜잭션 + 클라이언트 IP + 도메인 + 유저 에이전트 + Referer **- 트랜잭션 ID 6703915381101427655** + 멀티 트랜잭션 ID + WClientID

+ 로그인 ID  에러만 보기

**Q** **CSV**

**필름 선택**

	에이전트 명 (oname)	시작 시간	종료 시간 ↓	경과 시간	트랜잭션	CPU 사용 ...	HTTP 호출 ...	SQL 건수	SQL 패치 건...	트랜잭션 ID	
1	demo-8101	2024/02/28 00:00:00.419	2024/02/28 00:00:01.798	1,379	/product/kill/employee/daejun		2	0	11	5,748	6703915381101427655

트랜잭션 ID 기반으로 트랜잭션을 검색(전체 일치)합니다. 트랜잭션 ID의 경우 검색 인덱스가 생성되어 있기 때문에 빠른 속도로 검색할 수 있습니다.

- ① • 트랜잭션 ID는 [트랜잭션 정보](#) 창의 [레코드 요약](#) 탭에서 확인할 수 있습니다. 자세한 내용은 [다음 문서](#)를 참조하세요.
- 최신 버전의 에이전트가 아닌 경우 인덱스 사용이 아닌 풀스캔이 발생할 수 있습니다.

## 멀티 트랜잭션 ID

트랜잭션 검색

시간 선택: < 2024/03/06 00:00 ~ 2024/03/06 23:59 **1일** > 전체 선택 ~

필터: + 경과 시간 + 에러 메시지 + HTTP 호출 건수 + SQL 건수 + SQL 패치 건수 + 트랜잭션 + 클라이언트 IP + 도메인 + 유저 에이전트 + Referer + 트랜잭션 ID **- 멀티 트랜잭션 ID 846085190067870005** + WClientID

+ 로그인 ID  에러만 보기

**Q** **CSV**

**필름 선택**

	에이전트 명 (oname)	시작 시간	종료 시간 ↓	경과 시간	트랜잭션	CPU 사용 ...	HTTP 호출 ...	SQL 건수	SQL 패치 건...	멀티 트랜잭션 ID	
1	demo-8104	2024/03/06 00:00:00.176	2024/03/06 00:00:01.333	1,157	/product/remove/dept/pusan		2	0	11	7,112	846085190067870005

멀티 트랜잭션 ID 기반으로 트랜잭션을 검색(전체 일치)합니다.

- ① 멀티 트랜잭션 ID는 [트랜잭션 정보](#) 창의 [레코드 요약](#) 탭에서 확인할 수 있습니다. 자세한 내용은 [다음 문서](#)를 참조하세요.
- 최신 버전의 에이전트가 아닌 경우 인덱스 사용이 아닌 풀스캔이 발생할 수 있습니다.
- 멀티 트랜잭션 ID 필터를 적용하여 검색할 경우 [트랜잭션 ID](#) 필터만 추가 적용할 수 있습니다.

## WClientID

트랜잭션 검색

시간 선택: 2024/03/06 00:00 ~ 2024/03/06 23:59 1일 | 애플리케이션: 전체 선택

필터: + 경과 시간 | + 에러 메시지 | + HTTP 호출 건수 | + SQL 건수 | + SQL 패치 건수 | + 트랜잭션 | + 클라이언트 IP | + 도메인 | + 유저 에이전트 | + Referer | + 트랜잭션 ID | + 멀티 트랜잭션 ID | **- WClientID** |

+ 로그인 ID |  에러만 보기

🔍 CSV

필름 선택

	에이전트명 (oname)	시작 시간	종료 시간 ↓	경과 시간	트랜잭션	CPU 사용 ...	HTTP 호출 ...	SQL 건수	SQL 패치 건...	WClientID
1	demo-8104	2024/03/06 00:00:00.176	2024/03/06 00:00:01.333	1,157	/product/remove/dept/pusan	2	0	11	7,112	
2	demo-8101	2024/03/06 00:06:45.921	2024/03/06 00:06:46.979	1,058	/product/pickup/employee/daejun	1	0	11	3,811	
3	demo-8102	2024/03/06 00:07:10.522	2024/03/06 00:07:16.853	6,331	/account/save/employee/daejun	2	4	2	0	
4	demo-8103	2024/03/06 00:07:27.810	2024/03/06 00:07:28.867	1,057	/product/create/employee/kwangju	1	0	11	5,251	

클라이언트 ID(`wClientID`)는 와탭 에이전트가 사용자를 구분하기 위해 부여하는 ID입니다. 이 ID 값 기반으로 트랜잭션을 검색(전체 일치)합니다. 검색을 위한 별도 인덱스는 없습니다.

- ① 와탭은 클라이언트와 관련한 정보를 기본 저장합니다.

## 로그인 ID

로그인 아이디 기반으로 트랜잭션을 검색합니다.

- ① 기본 설정 상태의 에이전트는 로그인 ID 정보를 수집하지 않습니다. 로그인 ID 설정이 필요한 경우 [다음 문서](#)를 참조해 설정을 추가하세요.

## 다중 필터 적용하기

여러 개의 필터 항목을 적용해 원하는 트랜잭션 목록을 확인할 수 있습니다. 다음 사례는 경과 시간이 5 ~ 8초 사이이면서, HTTP 호출 건수가 5건 이상, 트랜잭션 경로에 'account'가 포함된 경우입니다.

트랜잭션 검색 🔊 🔔 📄 ⋮ 🇰

시간 선택 애플리케이션

< 2024/03/06 00:00 ~ 2024/03/06 23:59 ▶ 전체 선택 <

필터

경과 시간  
 에러 메시지  HTTP 호출 건수 
 SQL 건수  SQL 패치 건수  트랜잭션 
 클라이언트 IP  도메인  유저 에이전트

Referer  트랜잭션 ID  멀티 트랜잭션 ID  WClientID  로그인 ID  에러만 보기

컬럼 선택

	에이전트명 (oname)	시작 시간	종료 시간 ↓	경과 시간	트랜잭션	CPU 사용 ...	HTTP 호출 ...	SQL 건수	SQL 패치 건...
1	demo-8102	2024/03/06 00:00:01.650	2024/03/06 00:00:09.475	7,825	/account/delete/division/kwangju 📄	0	5	3	0
2	demo-8100	2024/03/06 00:00:19.877	2024/03/06 00:00:26.727	6,850	/account/pickup/dept/kwangju 📄	3	6	1	849
3	demo-8102	2024/03/06 00:01:07.095	2024/03/06 00:01:14.188	7,093	/account/save/dept/kwangju 📄	0	6	3	346
4	demo-8104	2024/03/06 00:01:37.539	2024/03/06 00:01:44.531	6,992	/account/delete/unit/kwangju 📄	3	5	3	940
5	demo-8105	2024/03/06 00:02:00.400	2024/03/06 00:02:07.181	6,781	/account/pickup/dept/kwangju 📄	3	5	4	1,627

❗ 멀티 트랜잭션 ID 필터를 적용하여 검색할 경우 트랜잭션 ID 필터만 추가 적용할 수 있습니다.

## 검색 결과 다운로드하기

사용자가 설정한 필터 조건에 따라 검색된 결과를 CSV 파일로 다운로드할 수 있습니다.

- 버튼을 클릭해 검색 결과를 조회하세요.
- 버튼을 선택하세요.
- 최대 CSV 라인 수를 입력하세요. 입력한 라인 수만큼 조회된 결과를 저장할 수 있습니다.
- [다운로드](#) 버튼을 선택하면 CSV 파일로 다운로드합니다.

❗ CSV 파일 이름은 `profile_{pcode}_YYMMDD.csv` 형식입니다.

# 메트릭스

## 메트릭스란?

와탭은 모니터링 대상으로부터 데이터를 수집해서 사용자에게 제공합니다. 에이전트로부터 수집되는 데이터를 **메트릭스**라고 표기합니다.

**메트릭스**는 사용자 환경을 한 눈에 살펴볼 수 있는 기준 요소를 제공합니다. 예를 들어 서버별 메모리 사용률 평균, DB 평균 연결 시간 등을 원본 데이터 목록이나 시각화한 차트 뷰를 통해 간편하게 확인할 수 있습니다. 문제 요소를 찾은 후에는 로그와 트레이스 등을 통해 상세 분석을 확인할 수 있습니다.

**메트릭스**는 또한 사용자 환경의 스케일을 조절하는 것에도 도움을 줍니다. 자원 사용량 통계를 통해 필요 자원량을 확정하는 것은 성능 향상과 비용 효율성 측면에서 중요한 기준입니다.

## 와탭의 메트릭스 수집 방식



와탭 에이전트는 모니터링 대상으로부터 모니터링 지표를 수집해 메트릭 데이터의 형태로 와탭 수집 서버에 전송합니다. 와탭 수집 서버는 관련 데이터를 카테고리별로 저장하고 관리합니다.

와탭의 수집 서버는 다양한 모니터링 대상에서 메트릭스를 수집합니다. 사용자는 원하는 메트릭스에 접근하기 위해 해당 상품별 안내 화면으로 이동해 기술된 과정을 따라야 합니다.

예를 들어 [Java 애플리케이션](#)을 모니터링하고 싶다면 먼저 와탭 에이전트를 설치해야 합니다. [다음 문서](#)를 참조하세요. 관련 메트릭스 지표 안내 또한 [다음의 상품별 페이지](#)에서 확인할 수 있습니다.

## 와탭의 메트릭스 구성 요소

와탭의 **메트릭스**는 다음의 정보들로 구성되어 있습니다.

- **Category:** 관련된 지표들을 묶는 단위로 메트릭스를 구분하는 Key를 의미합니다.
- **Tags:** 수집 대상을 구분할 수 있는 고유 정보를 포함하는 데이터입니다. 변경이 드문 IP, Oname, Host 정보 등의 항목을 저장합니다. Map 형태로 Multi Tag가 존재합니다.
- **Fields:** 에이전트로부터 수집된 모든 지표 값을 저장합니다. Map 형태로 Multi Field가 존재합니다.
- **Time:** 메트릭스가 수집된 시간입니다.
- **Oid:** 메트릭스를 수집한 에이전트의 고유 번호입니다.
- **Oname:** 메트릭스를 수집한 에이전트의 명칭입니다.

## 메트릭스 데이터 조회 및 시각화

와탭은 사용자가 지정한 조건에 따라 수집한 원본 데이터 목록과 편의성을 위해 다양하게 시각화한 차트를 다음과 같이 제공합니다. 메트릭스의 원본 데이터를 조회할 수 있는 **메트릭스 조회**, 시각화한 차트를 통해 메트릭스 데이터를 조회할 수 있는 **메트릭스 차트**, 시가 학습한 메트릭스 지표의 패턴과 비교해 예상 패턴을 벗어난 이상을 탐지할 수 있는 **메트릭스 이상 탐지** 메뉴를 확인해 보세요.

The screenshot displays the Metrics tool interface, divided into several sections:

- 메트릭스 조회 (Metrics Search):** A table listing metrics with columns for Time, Oid, Tags (alias, container, containerKey, host\_ip, okindName, oname, onodeName, pid, type), and Fields (dbc, http, method, socket, sql, s/s). It includes search filters and a CSV download button.
- 메트릭스 이상 탐지 (Metrics Anomaly Detection):** A panel for detecting anomalies in metrics like ApplicationActiveStat, DbConnection, HttpCall, and Socket.
- 메트릭스 차트 (Metrics Chart):** A central area with an 'Overlay Chart' and several sub-charts:
  - TPS:** Transaction Per Second chart showing a peak of 22.05.
  - 인스턴스 응답 시간 (Instance Response Time):** Chart showing a peak of 15.8s.
  - Think Time:** Chart showing a peak of 92.87s.
  - 서비스 시간 (Service Time):** Chart showing a peak of 57.82s.
  - HttpCall, DbConnection, Call, and Sql:** Time-series line charts for various metrics.
- Left Sidebar:** A navigation menu with categories like 트랜잭션 (Transaction), Think Time, 서비스 시간 (Service Time), TX 건수 (TX Count), and 트랜잭션별 SQL 시간 (Transaction SQL Time).



# 애플리케이션 지표

홈 화면 > 프로젝트 선택 > 분석 > [메트릭스 조회](#)

[메트릭스 조회](#) 메뉴에서 검색할 수 있는 메트릭스 지표입니다.

## app\_active\_stat

액티브 트랜잭션 구간 지표입니다.

- 수집 간격 : 5초
- 통계 데이터 : 5분, 1시간

## Tags

태그명	설명	비고
container	컨테이너 이름	고유값
containerKey	컨테이너 키값	고유값
host_ip	Host IP	고유값
okindName	애플리케이션 종류명	-
oname	에이전트 이름	고유값
onodeName	애플리케이션 노드명	-
pid	애플리케이션 PID	-
type	애플리케이션 유형	언어 이름

## Fields

필드명	단위	설명	비고
dbc	건수	Database Connection을 수행 중인 트랜잭션 건수	-
httpc	건수	HTTP Call을 수행 중인 트랜잭션 건수	-
method	건수	Method 로직 수행 중인 트랜잭션 건수	-
socket	건수	Socket 연결 수행 중인 트랜잭션 건수	-
sql	건수	SQL 쿼리를 수행 중인 트랜잭션 건수	-

## app\_counter

트랜잭션 지표입니다.

- 수집 간격 : 5초
- 통계 데이터 : 5분, 1시간

## Tags

태그명	설명	비고
alias	애플리케이션 별칭	고유값
container	컨테이너 이름	고유값
containerKey	컨테이너 키 값	고유값
host_ip	Host IP	고유값
okindName	애플리케이션 종류명	-

태그명	설명	비고
oname	에이전트 이름	고유값
onodeName	애플리케이션 노드명	-
pid	애플리케이션 PID	-
type	애플리케이션 유형	언어 이름

## Fields

필드명	단위	설명	비고
active_tx_0	건수	3초 이하 구간 트랜잭션 수	-
active_tx_3	건수	3초 초과 8초 이하 구간 트랜잭션 수	-
active_tx_8	건수	8초 초과 구간 트랜잭션 수	-
active_tx_count	건수	수행 중인 전체 트랜잭션 수	-
apdex_satisfied	건수	APDEX 만족 수	-
apdex_tolerated	건수	APDEX 허용 수	-
apdex_total	건수	APDEX 트랜잭션 총 수	-
arrival_rate	퍼센트	트랜잭션 요청률	-
httpc_count	건수	HTTP 외부 호출 수	-
httpc_error	건수	HTTP 외부 호출 에러 수	-
httpc_time	밀리 세컨드	HTTP 외부 호출 평균 시간	-

필드명	단위	설명	비고
metering	코어	애플리케이션이 작동 중인 호스트의 코어 수	컨테이너의 경우 limit cpu
resp_time	밀리 세컨드	평균 응답 시간	-
sql_count	건수	실행 완료된 SQL 건수	-
sql_error	건수	SQL 에러 건수	-
sql_fetch_count	건수	SQL Fetch 건수	-
sql_fetch_time	밀리 세컨드	SQL Fetch 수행 시간	-
sql_time	밀리 세컨드	SQL 평균 수행 시간	-
tps	건수	초당 트랜잭션 처리 수	-
tx_count	건수	트랜잭션 처리 건 수	-
tx_dbc_time	밀리 세컨드	DB 평균 연결 시간	-
tx_error	건수	트랜잭션 에러 건수	-
tx_httpc_time	밀리 세컨드	HTTP 호출 평균 시간	-
tx_sql_time	밀리 세컨드	트랜잭션별 SQL 수행시간의 합에 대한 평균	-
tx_time	밀리 세컨드	트랜잭션 수행 시간	-

## app\_host\_resource

애플리케이션 서버 자원 지표입니다.

- 수집 간격 : 5초

- 통계 데이터 : 5분, 1시간

## Tags

태그명	설명	비고
alias	애플리케이션 별칭	고유값
container	컨테이너 이름	고유값
containerKey	컨테이너 키값	고유값
host_ip	Host IP	고유값
okindName	애플리케이션 종류명	-
oname	에이전트 이름	고유값
onodeName	애플리케이션 노드명	-
pid	애플리케이션 PID	-
type	애플리케이션 유형	언어 이름

## Fields

필드명	단위	설명	비고
cpu	퍼센트	호스트 CPU 사용률	-
cpu_cores	정수	호스트 CPU 코어 수	-
cpu_irq	퍼센트	호스트 CPU IRQ 사용률	-
cpu_proc	퍼센트	자바 프로세스 CPU 사용률	-

필드명	단위	설명	비고
cpu_steal	퍼센트	호스트 CPU Steal 사용률	-
cpu_sys	퍼센트	호스트 CPU SYS 사용률	-
cpu_usr	퍼센트	호스트 CPU USER 사용률	-
cpu_wait	퍼센트	호스트 CPU WAIT 사용률	-
disk	퍼센트	루트 파일시스템 디스크 사용률	JAVA는 설정으로 지정한 파일시스템 경로의 디스크 사용률
mem	퍼센트	호스트 메모리 사용률	-
swap	퍼센트	호스트 SWAP 사용률	-

## app\_proc\_counter

애플리케이션 프로세스 지표입니다.

- 수집 간격 : 5초
- 통계 데이터 : 5분, 1시간

## Tags

태그명	설명	비고
container	컨테이너 이름	고유값
containerKey	컨테이너 키 값	고유값
host_ip	Host IP	고유값

태그명	설명	비고
okindName	애플리케이션 종류명	-
oname	에이전트 이름	고유값
onodeName	애플리케이션 노드명	-
pid	애플리케이션 PID	-
type	애플리케이션 유형	언어 이름

### Fields

필드명	단위	설명	비고
cputime	밀리 세컨드	트랜잭션 cpu 시간	-
db_num_active	정수	DB Connection Pool Active 수	-
db_num_idle	정수	DB Connection Pool Idel 수	-
gc_count	회수	GC 발생 회수	-
gc_oldgen_count	건수	Old Generation GC 건수	-
gc_time	밀리 세컨드	GC 수행 시간	-
heap_max	바이트	힙 최대량	-
heap_perm	바이트	Perm 사용량	-
heap_tot	바이트	힙 총량	-
heap_use	바이트	힙 사용량	-
pending_finalization	정수	GC 중 fianlize 수행을 위해 대기 중인 객체 수	-

필드명	단위	설명	비고
proc_fd	정수	프로세스 fd 사용수	-
proc_fd_max	정수	프로세스 fd 최대수	-
thread_count	정수	JVM 실행중인 스레드 수	-
thread_daemon	정수	JVM 데몬 스레드 수	-
thread_peak_count	정수	JVM 최대 스레드 수	-
thread_total_started	정수	JVM 실행 이후 총 start된 스레드 수	-



# 애플리케이션 성능 카운터

와탭 에이전트는 애플리케이션 성능과 관련된 다양한 정보를 수집합니다. 크게 3가지로 분류할 수 있습니다.

- **User** : 실시간 사용자 혹은 방문 사용자
- **Service** : 트랜잭션, SQL, 외부 호출 건수 및 응답, 에러율 등
- **Resource** : 시스템, 프로세스 자원 사용량

## User Counter

사용자는 모니터링 대상 시스템을 사용하는 클라이언트를 말합니다. 클라이언트에서는 일반적으로는 브라우저를 기준으로 사용자 수를 계산합니다.

웹 시스템 성능에서 사용자는 부하를 발생시키는 시작이기 때문에 중요합니다. 사용자 추적을 위해서는 사용자는 어떤 기준으로 구분하고, 어떻게 카운팅 할지에 대한 고려가 필요합니다.

## 사용자 구분

와탭 에이전트 사용자를 구분하기 위해 다양한 옵션을 제공합니다.

- **Remote IP**

기본값은 remote ip를 사용하여 사용자를 구분합니다. remote ip는 실제 사용자를 구분하는 데 한계가 있습니다.

- **Cookie**

쿠키를 사용하여 사용자를 구분합니다. 모든 접속 클라이언트를 대상으로 **WHATAP**이라는 쿠키에 UUID를 저장합니다.

```
whatap.conf
```

```
trace_user_using_ip=false
```

```
# Java agent v2.2.0 or later
```

```
wclient_using_ip=false
```

- **Header Key**

HTTP 헤더에 전달되는 값으로 사용자를 구분할 수 있습니다.

```
whatap.conf
```

```
user_header_ticket=USER

# Java agent v2.2.0 or later
wclient_header_ticket=USER
```

## 사용자 카운팅

사용자를 카운팅 하는 방법에 따라서 다르게 사용합니다. 실시간 사용자는 현재 시스템을 사용하는 사용자의 수를 알기 위해서 측정합니다. 일일 방문 사용자는 하루 동안 해당 서비스에 관심을 갖는 사용자가 몇 명인지에 대한 비즈니스적인 관리를 위해 측정합니다.

- **실시간 사용자**

최근 5분 동안 사용자 수를 카운팅 합니다. 5초마다 shifting 하면 사용자를 카운팅 합니다. 각 서버에서 카운팅 된 숫자는 HyperLogLog 알고리즘을 통해서 머지 됩니다.

- **일일 방문자(DAU, Daily Active User)**

하루 동안 시스템에 접속한 사용자를 카운팅 합니다. 24시간 동안 접속한 사용자를 HyperLogLog를 통해서 계산합니다.

✔ 와탭에서는 장기간 사용자를 카운팅 하기 위해 사용자 데이터에 대한 byte block을 서버로 수집합니다. 이 데이터를 Hyperloglog로 머지하면 이론적으로 한 달 이상의 방문 사용자를 계산할 수 있습니다.

## Service Counter

트랜잭션과 트랜잭션이 사용하는 SQL 혹은 외부 호출 등에 대한 건수, 응답시간 에러 건수 등에 대한 성능지표가 포함됩니다.

- **Transaction Counter**

트랜잭션을 수행하면 측정하는 카운터입니다.

- 건수
- 응답 시간
- 에러 건수

- **Active Transaction Counter**

진행 중인 트랜잭션의 수를 카운팅 합니다.

- **건수**

- **Active Status**

진행 상태는 METHOD, SQL, HTTPC, DBC, SOCKET 5가지 상태로 고정되어 있습니다.

- METHOD - 일반 함수를 호출하는 상태
- SQL - db sql을 수행 중인 상태
- HTTPC - 외부 Http Api(서비스)를 호출 중인 상태
- DBC - DB 연결을 요청한 상태, 일반적으로 Pool에서 가져옴
- SOCKET - TCP 세션을 Connecting 중인 상태

- **SQL**

SQL 수행 현황을 카운팅 합니다.

- **건수**
- **응답 시간**
- **에러 건수**
- **패치 건수**

- **HTTP Call**

외부 Http 호출에 대한 현황을 카운팅 합니다.

- **건수**
- **응답 시간**
- **에러 건수**

## Resource Counter

서버 자원 혹은 node 프로세스 내부의 자원 사용량을 카운팅 합니다.

- **CPU** (sys, usr, wait, steal, irq, cores)

CPU 사용량 %입니다. 각 종류별로 수집됩니다. 가상환경에서만 Steal이 의미가 있습니다. Cpu Core 개수를 같이 수집하고 있습니다.

- **Process CPU**

자바 프로세스가 사용하는 CPU%입니다.

- **Memory**

시스템 메모리 사용률(%)입니다.

- **Swap**

Swap 메모리 사용률(%)입니다.

- **Disk**

Disk는 Java Process의 Current 디렉터리의 사용률(%)입니다.

- **Heap (Total, Used, Perm)**

Java Heap 메모리의 Total, Used, Perm 양입니다. 데이터 단위는 KBytes 입니다.

- **GC (Count, Time)**

5초 동안 발생한 GC 건수와 시간의 합계입니다.

- **ObjectPendingFinalizationCount**

GC 호출되는 도중에 finalize() 수행하기 위해 대기 중인 객체 숫자입니다. 이 값이 커지면 GC time이 지연될 수 있습니다.

- **Thread (start Count, Count, Daemon, Peak Count)**

JVM이 실행된 이후부터 시작된 스레드 수, 현재 돌고 있는 스레드 수, 데몬, 최대 스레드 수를 수집합니다. 스레드 풀이 필요한지에 대한 판단 등을 할 수 있습니다.

- **DB Connection Count (active,idle)**

Connection Pool의 idle와 active 카운트를 수집합니다.

- **Tomcat Thread Pool (active, queueSize)**

톰캣의 Executor 스레드 풀이 설정되었을 때 성능 정보를 수집합니다.

XML

```
<Executor name="tomcatThreadPool" namePrefix="catalina-exec-"
maxThreads="50" minSpareThreads="2" maxQueueSize="78"/>
```

## DB Connection Pool 카운터

DB Connection Pool은 DataSource라고 합니다. WAS가 제공하는 경우가 일반적이지만 일부에서는 오픈소스 Pool을 사용하기도 합니다. 사용량 정보가 JMX로 노출되기도 하지만 그렇지 않은 경우도 많습니다.

와탭은 2가지 방식을 제공하고 있습니다. 하나의 BCI에 의한 직접 조회 방식과 JMX를 이용한 방식입니다. 기본은 BCI 방식을 사용하고 보조로 JMX를 사용할 수 있도록 제공하고 있습니다.

다음은 개별 옵션별 디폴트 값과 연관되어 ByteCode Injection되는 클래스입니다.

```
hikari_pool_enabled=false
```

```
com.zaxxer.hikari.pool.HikariPool
```

```
dbcp_pool_enabled=true
```

```
org.apache.commons.dbcp.BasicDataSource
org.apache.commons.dbcp2.BasicDataSource
org.apache.tomcat.dbcp.dbcp.BasicDataSource
org.apache.tomcat.dbcp.dbcp2.BasicDataSource
```

```
tomcat_pool_enabled=true
```

```
org.apache.tomcat.jdbc.pool.ConnectionPool
```

```
weblogic_pool_enabled=true
```

```
weblogic.jdbc.common.internal.ConnectionPool
```

```
jeus_pool_enabled=true
```

```
jeus.jdbc.connectionpool.ConnectionPoolImpl
```

```
jboss_pool_enabled=true
```

```
org.jboss.jca.core.connectionmanager.pool.PoolStatisticsImpl
```

⚠ 옵션을 변경한다면 BCI 관련 옵션이기 때문에 **다시 시작**해야 합니다. 클래스가 instrument된 후에도 추적 테이블에 등록되는 과정이 필요합니다. 따라서 실행 중에 redefine을 해도 동작하지 않습니다.

## DataSource 현황 JMX로 가져오기

```
dbcp_pool_enabled==true 이면 tomcat_ds_enabled=false
weblogic_pool_enabled== true 이면 weblogic_ds_enabled=false
```

1. 옵션 자동 결정 내용을 확인하세요.
2. 관련된 pool 쪽 옵션 값을 `false` 로 선언하세요.
3. DataSource 관련 옵션 값을 `true` 로 변경하세요. 기본값은 `false` 입니다.

whatap.conf

```
tomcat_ds_enabled=false
weblogic_ds_enabled=false
```

## ThreadPool 카운터

ThreadPool 카운터는 JMX Mbean 정보를 통해 수집됩니다. Tomcat과 같이 사용 사례가 많은 경우라면 문제없지만 특정 환경에서는 이 지표가 수집되지 않을 수 있습니다.

수집을 위해서 추가 설정이 필요합니다. 관련 설정은 다음과 같습니다.

whatap.conf

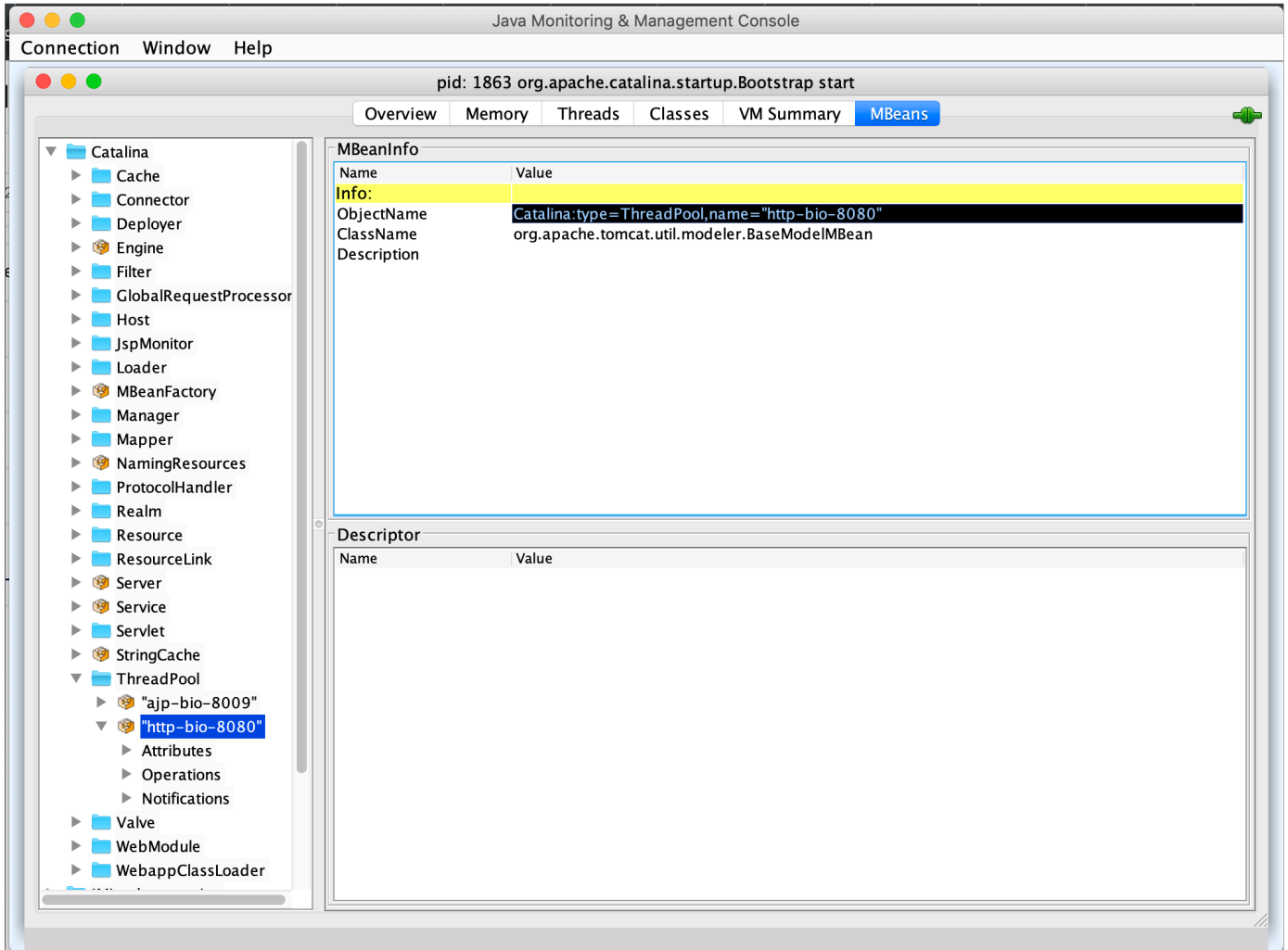
```
1 jmx_threadpool_objectname=Catalina:type=ThreadPool,name="http-bio-8080"
2 // JMX ThreadPool MBean 이름을 설정합니다.
3 jmx_threadpool_atter_activecount=connectionCount
4 // activecount 수치를 나타내는 attribute를 설정합니다.
5 jmx_threadpool_atter_queuesize=maxThreads
6 // maxThreads 수치를 나타내는 attribute를 설정합니다.
```

JMX ThreadPool ObjectName 및 AttributeName 과 같은 Mbean 명은 JMX 모니터링 도구를 통해 확인하거나 각 WAS 별 Mbean 문서를 참고합니다.

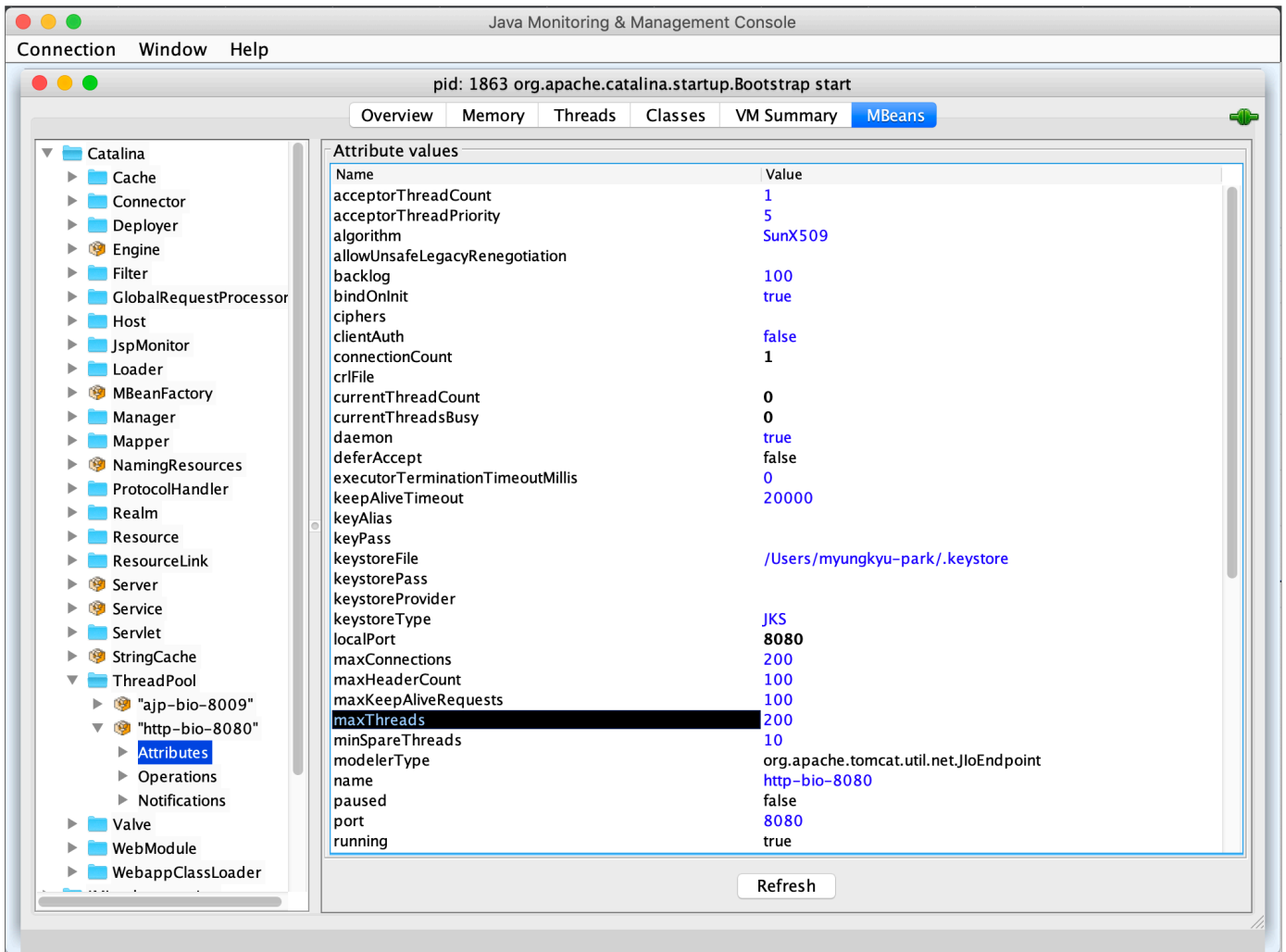
❗ 사용 중인 WAS, 프레임워크 관련 [문서](#)를 참고해 Mbean 정보를 확인하세요.

## Jconsole 사용 시

- **ObjectName** : Catalina  
**type** : ThreadPool  
**name** : "http-bio-8080"



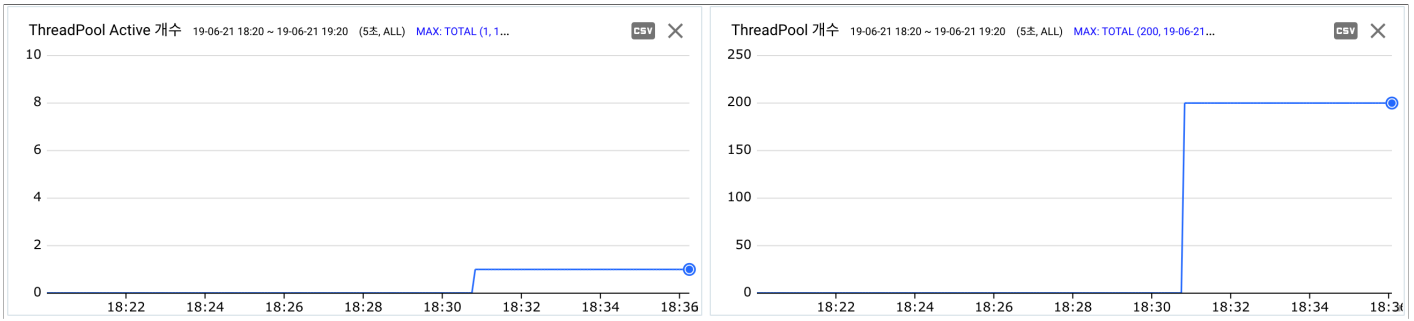
- **ActiveName** : connectionCount  
**QueueName** : maxThreads



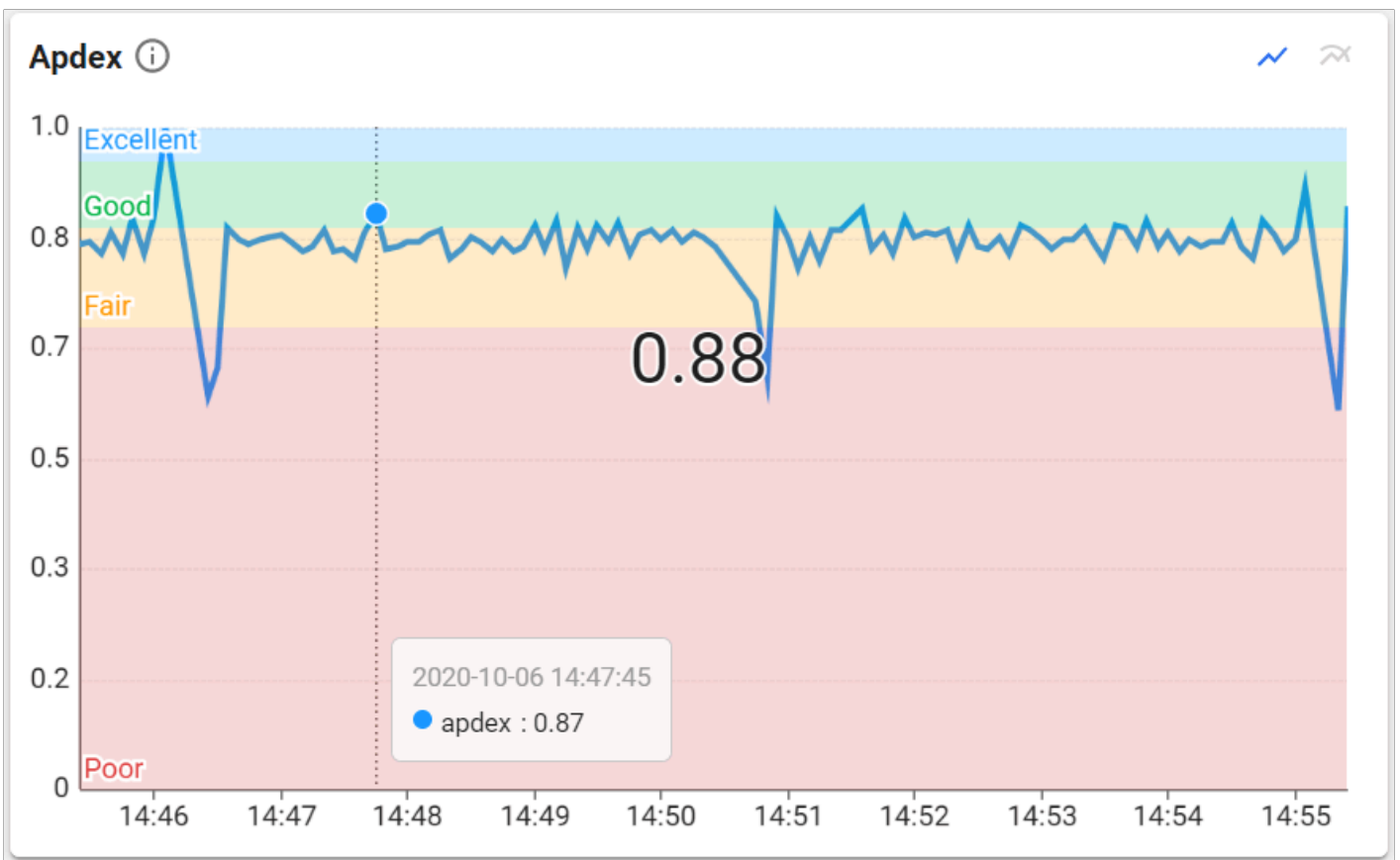
## 결과 확인

설정이 정상적으로 되면 카운터에서 다음과 같이 데이터를 조회할 수 있습니다.





## Apdex



Apdex(Application Performance Index)는 개방형 표준을 따르는 애플리케이션 성능지표입니다. Apdex는 응답시간에 기반하며 전체 요청 중 만족과 허용 건 비율로 수치화합니다. 대시보드에 Apdex 그래프가 추가되었습니다.

Apdex는 사용자 만족도에 대한 지표로 활용할 수 있으며, 0 ~ 1 사이의 값을 갖습니다.

$$(만족 횟수 + (허용 횟수 * 0.5)) / 전체 요청 수$$

상태	설명
만족 (Satisfied, <b>S</b> )	업무처리에 전혀 문제가 없음 ≤ 1.2초 (만족 <b>S</b> 기본값)
허용 (Tolerating, <b>T</b> )	사용자가 지연을 느끼나 업무처리는 가능함 ≤ 4.8초 (만족 <b>S</b> * 4)
불만 (Frustrated, <b>F</b> )	업무처리가 불가능함 > 4.8초 (허용 <b>T</b> 초과 및 오류)

• **whatap.apdex\_time** millisecond

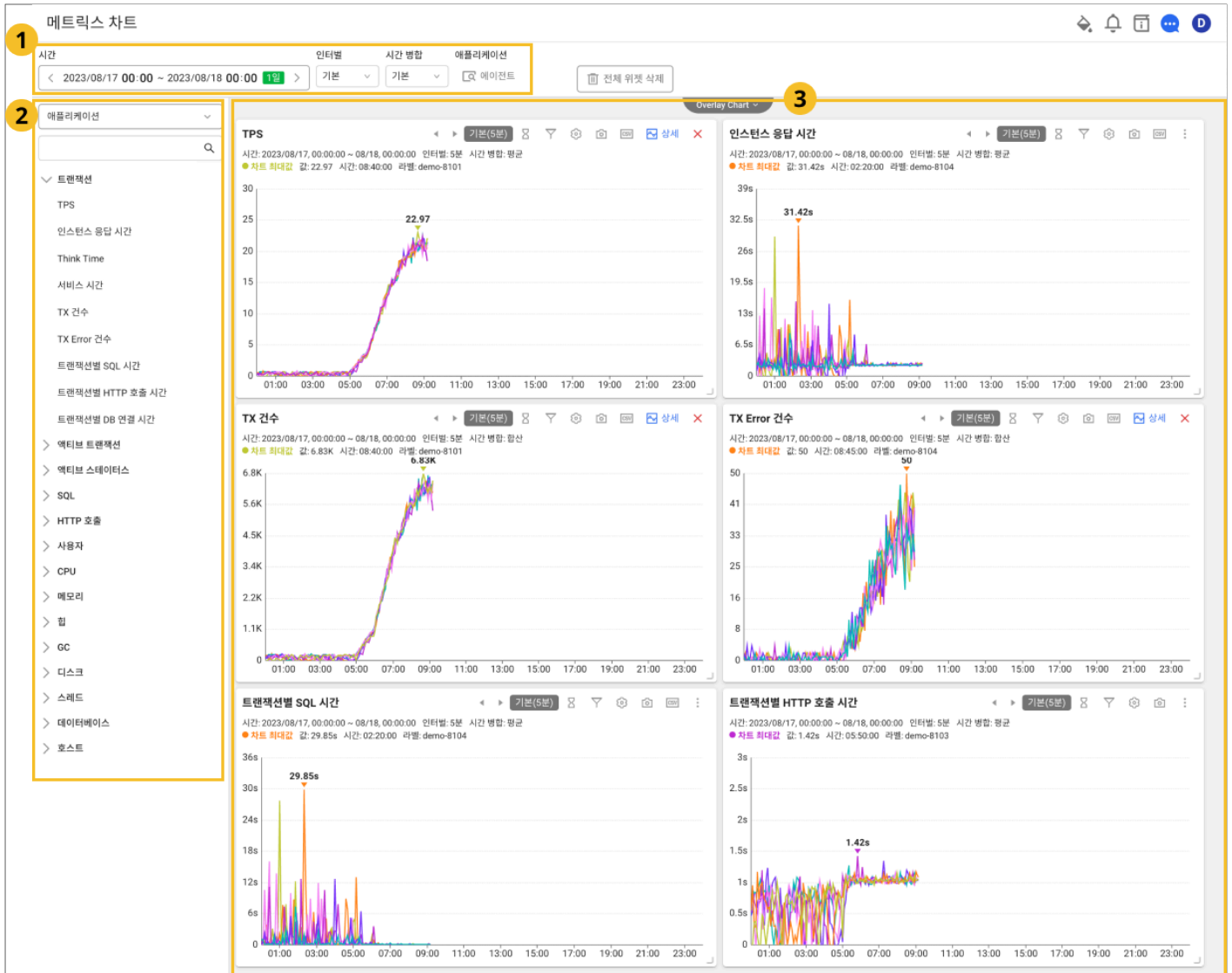
기본값

만족 **S** 기본값은 에이전트 설정 메뉴에서 변경할 수 있습니다.

# 메트릭스 차트

홈 화면 > 프로젝트 선택 > 분석 > 메트릭스 차트

메트릭스 차트 메뉴에서 모니터링 대상에서 수집된 메트릭스 데이터를 다음과 같이 차트로 조회할 수 있습니다. 시간과 지표를 지정하는 것은 필수입니다.



## 상단 옵션

1 영역의 **메트릭스 차트**의 상단 옵션을 통해 차트의 시간 범위와 모니터링 대상 에이전트를 지정할 수 있습니다.

- **시간**: 시간의 총 범위로 X축의 시작과 끝을 지정할 수 있습니다.
- **인터벌**: 시간 간격으로 X축 데이터 간격을 지정할 수 있습니다.
- **시간 병합**: 데이터 병합 방식 중 하나로 인터벌로 지정한 시간 내의 데이터를 병합할 수 있습니다.  
예, **평균**은 1시간 안 데이터의 평균값을 말합니다.
- **에이전트**: 조회할 에이전트를 지정할 수 있습니다. 지정하지 않으면 전체가 조회됩니다.

## 지표 목록

2 영역은 옵션을 조회할 지표 목록입니다. 먼저 **카테고리**를 선택하세요. 선택한 **카테고리** 하위의 지표를 조회한 후 원하는 지표를 선택하세요. **카테고리**와 지표를 선택하면 1 영역의 상단 메뉴에서 지정한 시간 범위의 데이터를 바탕으로 3 영역에서 차트 위젯을 조회할 수 있습니다.

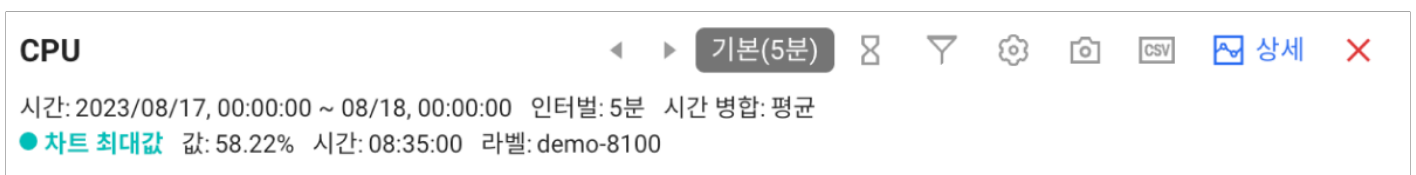
### ! 데이터 병합

데이터 병합은 **시간 병합**과 **오브젝트 병합**을 제공합니다.

- **시간 병합**은 원본 데이터에서 필드 값이 같은 데이터끼리 일정한 간격으로 데이터를 병합합니다.
- **오브젝트 병합**은 서로 다른 필드 값을 가진 데이터들 중에서 태그가 일치하는 경우 해당 데이터를 병합합니다.

## 차트 위젯

3 영역 차트 위젯의 좌측 상단에서 지표명을 확인할 수 있습니다. 차트 위젯의 우측 상단에서 다음과 같은 옵션을 확인할 수 있습니다.



- **시간 이동**: ◀ ▶ 왼쪽 화살표 또는 오른쪽 화살표 버튼을 통해 선택한 시간 범위만큼 -1, +1 씩 이동 가능합니다.

예, 시간 범위가 2월 13일 00:00~2월 14일 00:00일 때, ◀ 왼쪽 화살표를 선택하면 2월 12일 00:00~2월 13일 00:00 데이터를 조회할 수 있습니다.

- 인터벌/시간 병합: ① 상단 메뉴에서 지정한 인터벌과 시간 병합을 수정할 수 있습니다.
- 모니터링 대상: ▾ 아이콘을 선택해 모니터링 대상을 지정할 수 있습니다. 선택하지 않으면 전체를 대상으로 조회합니다.
- 시간 비교: ⏮ 아이콘을 선택하면 동일한 지표의 이전 시간대의 추이를 비교할 수 있습니다.
- 스냅샷: 📷 아이콘을 선택해 위젯의 옵션을 제외한 차트를 스냅샷 할 수 있습니다.
- CSV: 📄 아이콘을 선택해 차트를 그리는 데이터를 CSV 파일로 다운로드할 수 있습니다.
- 상세 보기: 🔍 아이콘을 선택해 상세 조회가 가능합니다. 모니터링 대상이 많은 경우 유용하며, 모니터링 대상의 지표 추이를 개별로 확인 할 수 있습니다.

ⓘ 메트릭스 차트 위젯 상단에서 옵션이 보이지 않을 경우 ⋮ 아이콘을 선택하세요.

## 위젯 삭제하기

화면에 배치한 모든 위젯을 삭제하려면 화면 위에 위치한 ☰ **전체 위젯 삭제** 버튼을 선택하세요.

# 메트릭스 조회

홈 화면 > 프로젝트 선택 > 분석 > 메트릭스 조회

메트릭스 조회 메뉴에서 태그 기반으로 특정 메트릭스를 조회할 수 있습니다.

The screenshot shows the '메트릭스 조회' (Metrics Query) interface. It includes a search bar with time and category filters, a tag selection section, and a table of results.

	Time	Oid	Tags							Fields					
			container	containerKey	hostIp	oKindName	oname	onodeName	pid	type	active_tx_3	arrival_rate	sql_time	tps	tx_error
1	2023-08-17 07:56:45	-877561626	agent.service	1395943798	10.21.1.26	demo-okind-1	demo-8101	node-1	3284967	java	11	24.390244	5.7834587	20.68982	1
2	2023-08-17 07:56:45	-1128904592	agent.service	1395943798	10.21.1.26	demo-okind-0	demo-8100	node-0	3284934	java	5	15.993603	9.613368	18.623732	0

## 1 시간과 카테고리 선택

1 영역에서 메트릭스가 수집된 시간과 최대 개수 및 카테고리를 지정할 수 있습니다. 시간과 카테고리는 반드시 지정해야 합니다.

- **시간**: 메트릭스가 수집된 시간을 지정해 조회할 수 있습니다. 기본값은 1시간 입니다. 기본 옵션으로 제공하는 조회 시간 외 사용자가 직접 시간 탭을 선택해 날짜와 시간을 지정할 수 있습니다.
- **최대 개수**: 3 목록에 조회할 메트릭스 최대 개수를 지정할 수 있습니다. 10, 50, 100, 200, 300, 1000, 2000, 3000 개까지 설정할 수 있습니다.
- **카테고리**: 유관 지표들의 분류 단위입니다. 카테고리 탭을 선택해 원하는 카테고리를 지정할 수 있습니다.
- **새로 고침**: 새로 고침 아이콘을 선택하면 카테고리, 태그 및 필드 옵션을 다시 불러올 수 있습니다.

## 2 태그와 필드 선택

2 영역에서 태그와 필드를 선택합니다. 사용자가 개별적으로 지정하지 않는다면 기본 설정은 전체 선택입니다.

- **태그**: 수집된 대상을 구분할 수 있는 고유 정보 데이터입니다.

- **필드**: 모니터링 대상으로부터 수집된 지표입니다.
- **필터**: ▾ **태그값으로 필터링** 버튼을 선택하고 태그 값을 설정해 필터링할 수 있습니다.  
예시, `oname` 의 값을 `demo-8101` 로 설정해 필터링한 데이터를 조회할 수 있습니다.
- **🔍 검색**: 조건을 설정 후 **검색** 아이콘을 선택하면 ③ 영역에서 해당 메트릭스의 원본 데이터를 조회할 수 있습니다.
- **📄 CSV** **CSV 다운로드**: **CSV 다운로드** 버튼을 선택해 해당 메트릭스 원본 데이터를 CSV 파일로 다운로드할 수 있습니다.

### 3 메트릭스 테이블

수집되는 메트릭스를 사전에 특정할 수 없기에 수집 중인 모든 메트릭스의 원본 데이터를 확인하는 것이 중요합니다. 위의 조건 영역에서 원하는 조건을 설정 후 ③ 영역에서 해당 메트릭스의 원본 데이터를 테이블 형식으로 조회할 수 있습니다. 사용자가 **태그**와 **필드** 각 조건을 지정함에 따라 테이블의 컬럼이 변경됩니다.

- ① • 메트릭스 조회 시 **시간**과 **카테고리**는 반드시 지정해야 합니다.
- 메트릭스 조회 시 **태그**와 **필드** 지정은 선택 사항입니다.

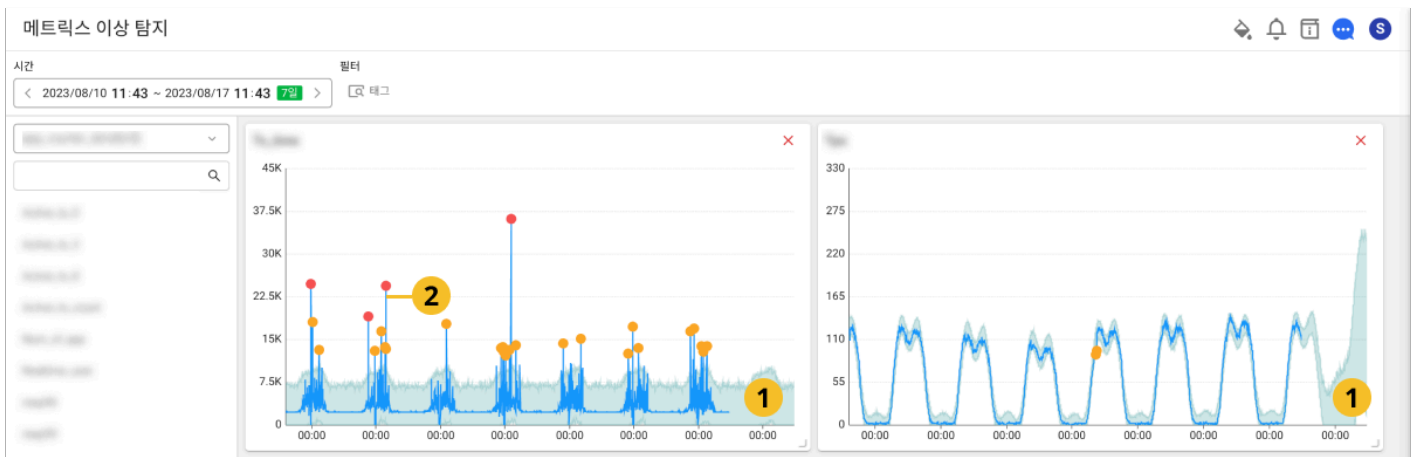
# 메트릭스 이상 탐지

홈 화면 > 프로젝트 선택 > [사이트맵](#) > [메트릭스 이상 탐지](#)

다양한 메트릭스 지표의 패턴을 시가 학습한 예상 패턴과 비교해 볼 수 있습니다. 예상 패턴을 벗어난 이상 탐지를 그래프 차트를 통해 확인할 수 있습니다. 과거 데이터를 바탕으로 반복되는 패턴을 확인하고 향후 지표 값 예측에 활용할 수 있습니다.

- ❗ • 화면에 배치한 위젯은 다른 메뉴로 이동할 경우 저장되지 않고 초기화합니다.
- 패턴 표시와 이상치 표시 기능을 제외하면 [분석 > 메트릭스 차트](#) 메뉴와 유사합니다.
- **이상치 탐지(Anomaly Detection)** 경고 알림 기능의 기술 근간은 **메트릭스 이상 탐지**입니다. **이상치 탐지** 경고 알림 설정에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

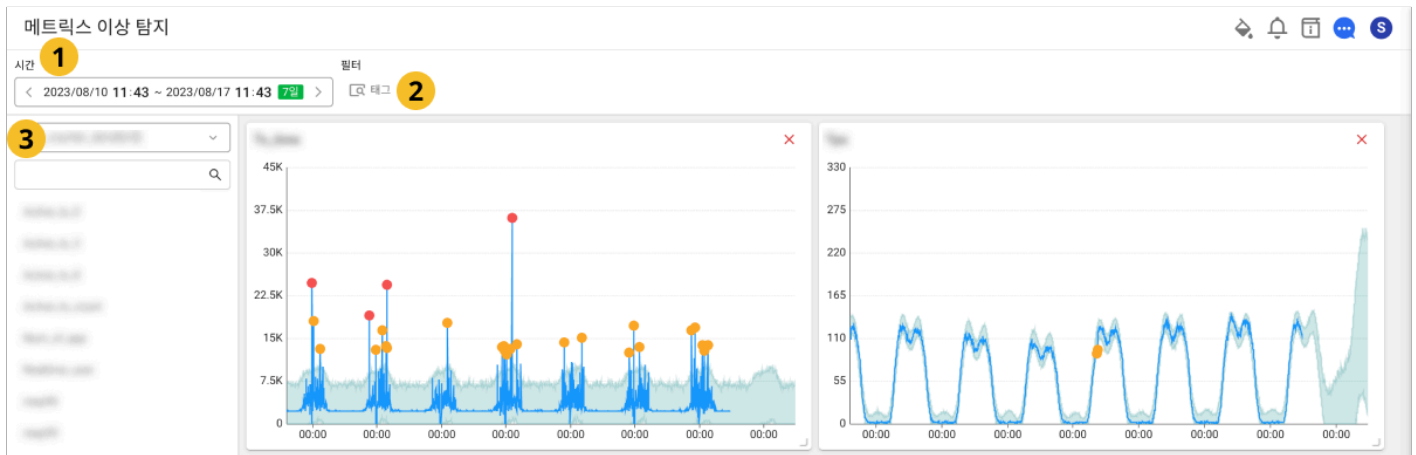
## 위젯 확인하기



- **1** 밝은 색상의 그래프 영역은 시가 분석한 예상 패턴입니다.
- **2 파랑색** 그래프는 프로젝트의 메트릭스 지표 추이입니다.
- 시가 분석한 예상 패턴을 벗어나면 **주황색**, **빨간색**의 단계로 그래프에 점을 표시합니다. 예상 패턴 범위를 크게 벗어난 값을 **빨간색**으로 표시합니다.



# 위젯 배치하기

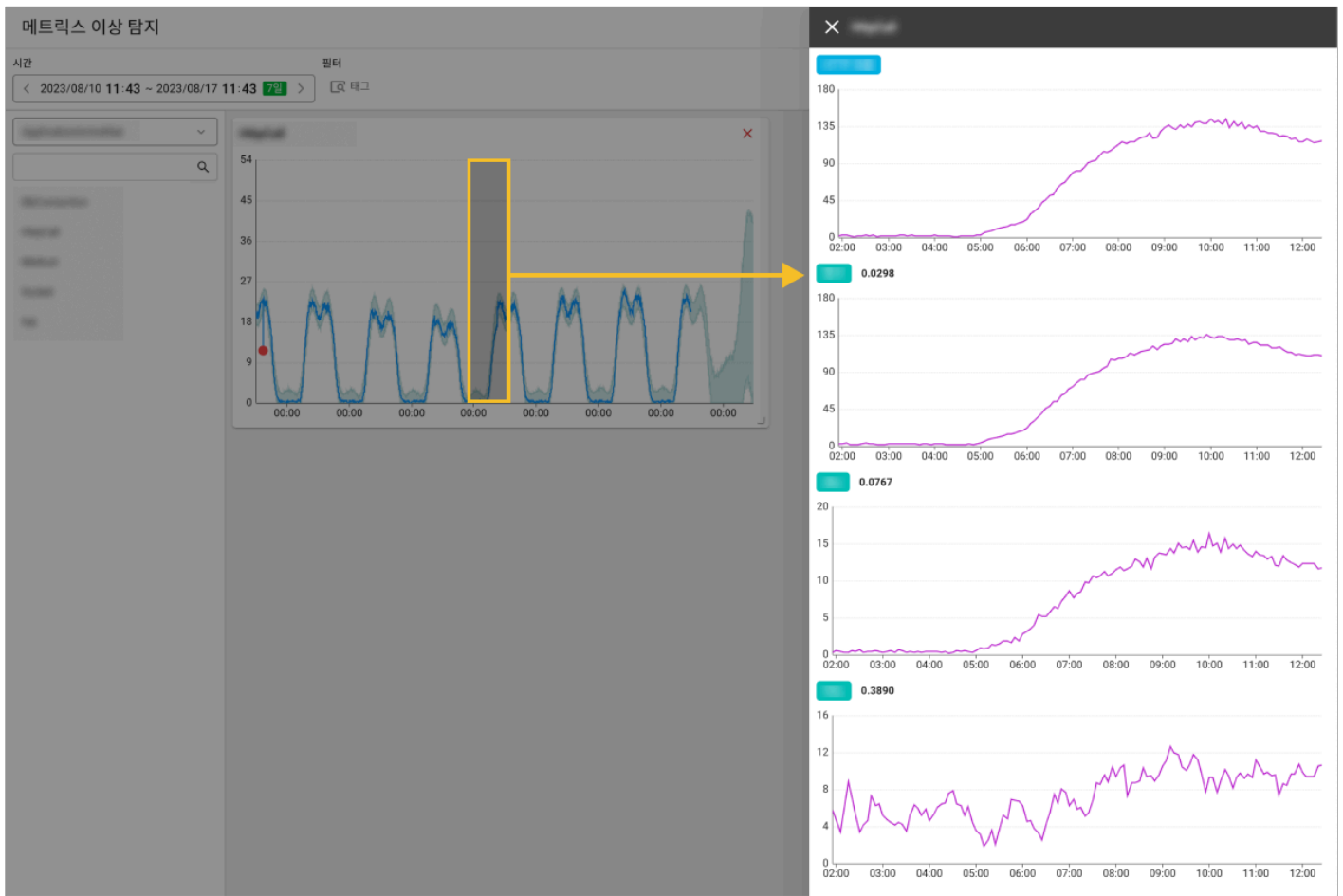


1. ① **시간**에서 원하는 시간 간격을 설정하세요. 최대 1개월 간격까지 설정할 수 있습니다.
2. ② **필터**에서 메트릭스 지표의 범위를 선택하세요.
3. 아래 목록에서 모니터링하길 원하는 메트릭스 지표를 선택하세요.

선택한 메트릭스 지표를 화면 오른쪽에 배치합니다.

- ① • 모니터링 대상을 선택해 메트릭스 지표를 구분해서 확인하려면 ② **태그**를 선택하세요. 화면 오른쪽에 태그 선택 목록이 나타납니다. 원하는 항목을 선택한 다음 위젯을 추가하세요.
  - 화면에 배치한 위젯은 **시간** 또는 **태그** 값을 변경해도 차트에 반영되지 않습니다.
  - 화면에 배치한 위젯을 삭제하려면 위젯 오른쪽 위에 **×** 버튼을 선택하세요.
  - 위젯의 왼쪽 위를 선택한 상태에서 드래그해 위젯 위치를 변경할 수 있습니다.
  - 위젯 오른쪽 아래를 선택한 상태에서 드래그해 위젯 크기를 조절할 수 있습니다.

# 연관 지표 확인하기



위젯에서 차트의 일부 영역을 드래그하세요. 선택한 영역에서 해당 메트릭스 지표와 관련한 지표를 같이 확인할 수 있습니다.

# 트랜잭션

## 트랜잭션이란?

사용자 브라우저의 요청을 처리하기 위한 서버 사이드의 LUW(Logical Unit of Work)를 말합니다. 개별 웹서비스(URL) 요청에 대한 처리 과정이 바로 트랜잭션입니다. 웹 애플리케이션에서 트랜잭션은 웹서비스(URL)에 대한 HTTP Request를 받아 Response를 반환하는 과정입니다.

애플리케이션의 성능은 이 트랜잭션들의 성능으로 요약할 수 있습니다. 트랜잭션 성능은 트랜잭션 시작에서부터 종료 시점, 응답시간 및 자원 사용량 혹은 트랜잭션 호출자 속성 등의 정보를 포함합니다.

기본적으로 트랜잭션 응답 분포와 트랜잭션 통계를 통해서 트랜잭션 성능을 분석할 수 있습니다.

## 트랜잭션의 이름

트랜잭션의 이름은 URL입니다. 단 Get 파라미터(Query String)는 제외됩니다.

### 브라우저 요청

```
http://www.whatap.io/hr/apply.do?name='kim'
```

### 트랜잭션 이름

```
/hr/apply.do
```

- ⓘ 와탭에서는 **웹서비스 이름**과 **트랜잭션 이름**을 혼용하고 있습니다. 서비스 특정 URL과 그에 대한 요청을 처리하기 위한 모듈로 볼 수 있습니다. 트랜잭션 요청에 대한 처리 하나를 의미하기 때문에 둘의 이름은 동일하게 URL이라고 할 수 있습니다.

## 트랜잭션 이름 정규화

MSA 기반의 시스템이 발전하면서 URL + ? 인자 파라미터 형식보다 URL 패스에 파라미터를 넣는 방식을 많이 사용하게 됩니다.

```
http://www.whatap.io/hr/kim/apply.do
```

이렇게 패스 파라미터를 그대로 트랜잭션 이름으로 사용하게 되면 통계적 관점의 성능 분석이 어렵습니다. 이를 정규화할 필요가 있습니다. 와탭은 정규화를 위한 옵션과 기능을 제공합니다.

```
whatap.conf
```

```
trace_normalize_urls=/hr/{name}/apply.do
```

위와 같이 설정하면 트랜잭션 이름이 `/hr/kim/apply.do` → `/hello/:name/apply.do`로 치환되어 수집됩니다. 만약 대상 URL 설정은 그대로 두고 기능만 off 하고자 한다면 다음과 같이 옵션을 지정할 수 있습니다. 기본값은 `true` 입니다.

```
whatap.conf
```

```
trace_normalize_enabled=false
```

# 트랜잭션 분석하기

## 히트맵

홈 화면 > 프로젝트 선택 > 분석 > 히트맵

와탭 모니터링 서비스 초기 화면에서 프로젝트를 선택한 다음 프로젝트 메뉴 하위에 [분석](#) > [히트맵](#) 메뉴를 선택하세요.

① 히트맵은 [대시보드](#) > [애플리케이션 대시보드](#)의 [히트맵](#) 위젯을 통해 접근할 수도 있습니다.

# 상세 분석

**1 히트맵**

시간 선택 < 2023/10/05 00:00 ~ 2023/10/06 00:00 1인 >

구분 에이전트 전체

애플리케이션

**2 히트맵 트랜잭션** 5,202,658 28,415

**4 애플리케이션** All

에이전트 명	히트 ↓	예러
All	1,669,159	9,165
demo-8105	280,764	1,568
demo-8102	280,148	1,535
demo-8101	278,052	1,549
demo-8100	277,758	1,577
demo-8103	276,727	1,452
demo-8104	275,710	1,484

**3** (Zoomed-in heatmap area)

**5 TX 트레이스 (All applications)**

시작 시간	경과 시간↑	트랜잭션	HTTP 호출...	SQL 건수	SQL 시간	SQL 패치 건...	DB 연결 시간	에러 클래스	에러 메시지
2023/10/05 10:59:50.892	11,624	/account/delete/employee/daejun	11,077	6	524	28	11		
2023/10/05 10:59:47.862	11,749	/account/save/unit/daegu	10,887	4	835	1,443	4		
2023/10/05 10:59:47.473	12,179	/edu/save/dept/pusan	10,272	6	1,871	1,593	6		
2023/10/05 10:59:44.481	12,443	/product/create/unit/seoul	11,660	3	776	0	4		
2023/10/05 10:59:46.258	12,649	/sales/pickup/division/kwangju	11,591	8	968	1,499	74		
2023/10/05 10:59:45.406	12,939	/product/read/dept/seoul	12,864	2	4	0	66		
2023/10/05 10:59:47.310	13,297	/edu/write/dept/kwangju	13,216	3	58	1,354	4		
2023/10/05 10:59:43.980	13,325	/account/save/division/jeju	12,989	7	98	681	208		
2023/10/05 10:59:48.240	13,417	/account/create/dept/kwangju	12,902	4	496	374	8		
2023/10/05 10:59:43.956	13,624	/product/create/division/jeju	13,588	2	30	0	2		
2023/10/05 10:59:46.414	13,851	/product/remove/division/daejun	13,039	6	609	1,215	180		
2023/10/05 10:59:44.541	13,925	/edu/delete/employee/daegu	13,534	6	281	1,137	87		
2023/10/05 10:59:38.400	19,094	/edu/delete/division/daegu	18,791	2	285	678	2		
2023/10/05 10:59:32.760	24,622	/account/save/division/pusan	13,184	8	171	0	77		
2023/10/05 10:59:32.867	24,646	/account/create/employee/kwangju	13,025	2	133	120	143		

## 1 에이전트 영역

현재 프로젝트와 연결된 에이전트를 선택해 필터링할 수 있습니다. 돋보기 아이콘을 선택하면 2 영역에서 히트맵 트랜잭션 차트를 확인할 수 있습니다.

## 2 히트맵 트랜잭션 차트

지연 문제가 있는 트랜잭션이 포함된 경우 **히트맵 트랜잭션** 차트에서 **주황색**이나 **붉은색**으로 표시됩니다. 정상 트랜잭션만 포함된 경우 **파란색**으로 표시됩니다.

## 3 선택 영역

실시간 히트맵 차트를 드래그하면 선택 영역의 **4 에이전트 목록**과 트랜잭션 정보를 확인할 수 있는 **5 TX 트레이스** 목록이 나타납니다.

## 4 에이전트 목록

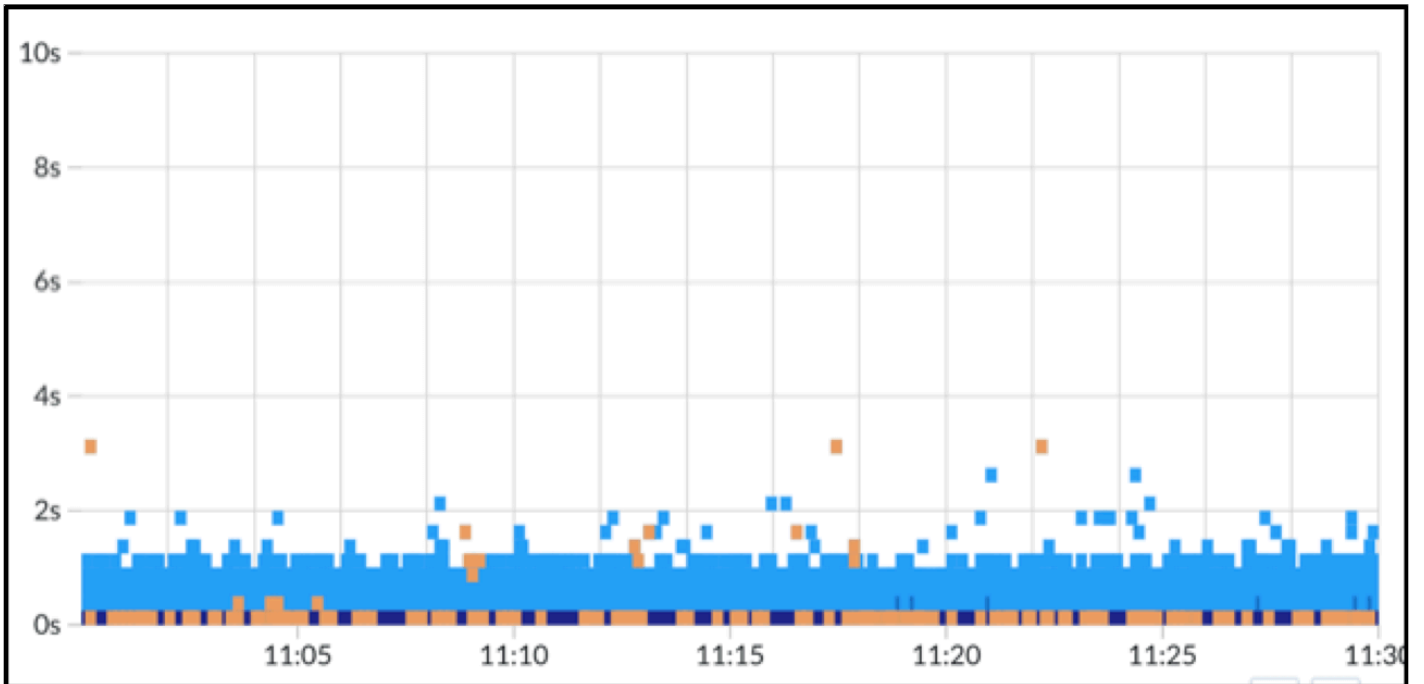
차트에서 선택된 영역에 포함되는 트랜잭션 수와 에러 수를 확인할 수 있습니다. 목록 중에서 원하는 애플리케이션을 선택하면 그 애플리케이션의 **5 TX 트레이스** 목록에 세부 정보가 나타납니다.

## 5 TX 트레이스 목록

**TX 트레이스** 목록은 세부 정보를 가지고 있습니다. 목록에서 원하는 트랜잭션을 선택하면 **통계** 정보 창이 나타납니다. **통계** 창에서 해당 트랜잭션의 트레이스 상세 분석을 확인할 수 있습니다. 자세한 내용은 [다음 문서](#)를 참조하세요.

## 히트맵 패턴의 이해

히트맵은 트랜잭션의 종료시간은 X축, 응답시간은 Y축으로 한 분포 차트입니다. 정상적인 웹 애플리케이션이라면 수 초 이하 구간에 집중된 분포를 보입니다.

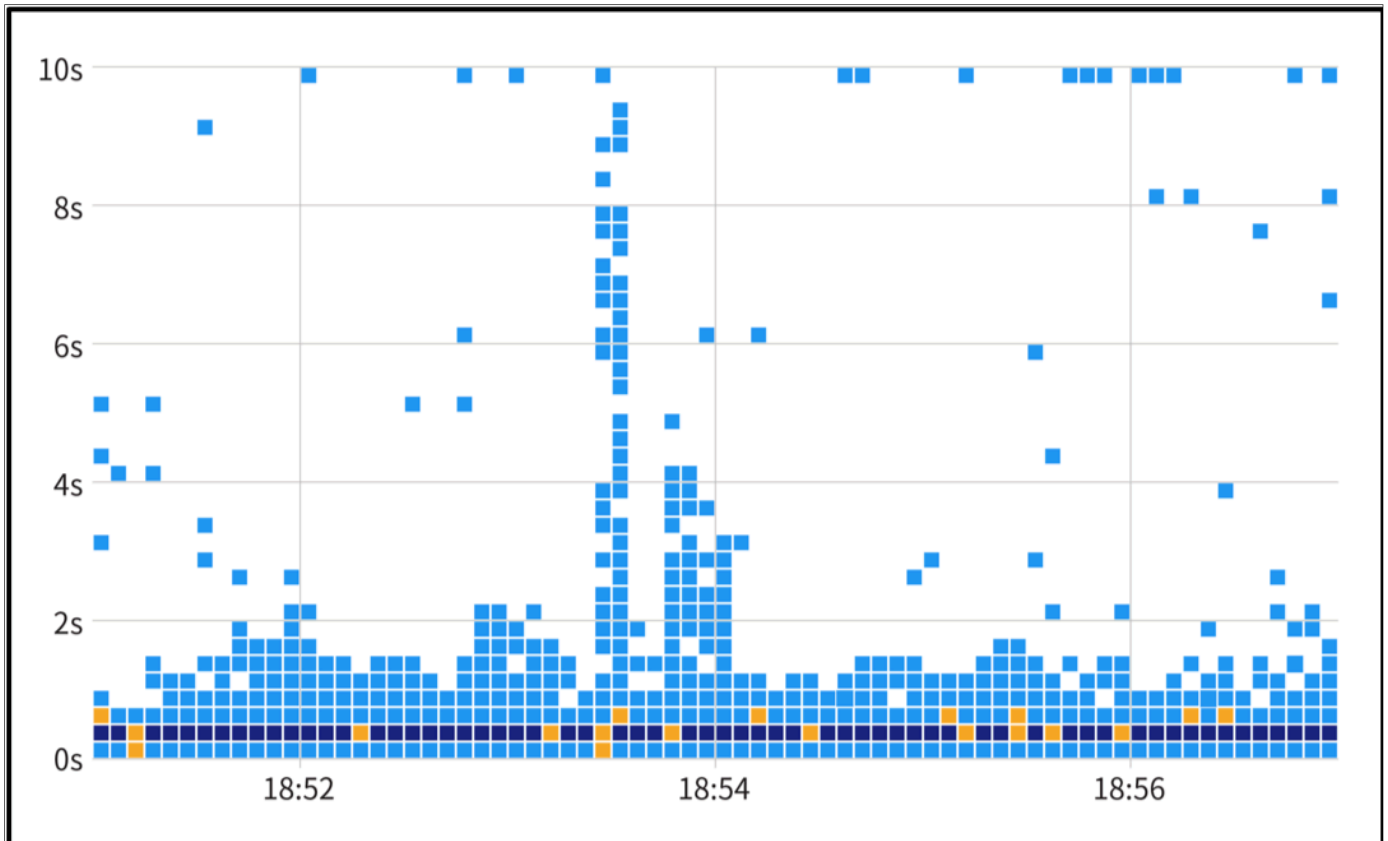


## 히트맵 라인 분석하기

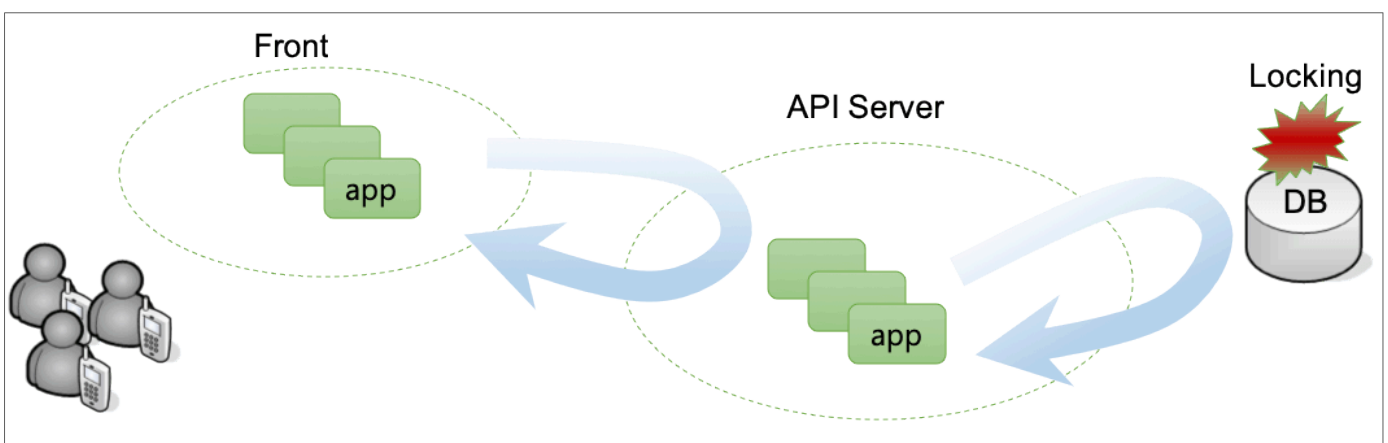
- 세로줄(LOCK 현상) 패턴

트랜잭션 처리 중 일시적인 락(Not only DB Lock)이 발생하면 이로 인해 처리를 대기합니다. 락이 해소되면 처리 대기 중 트랜잭션들은 비슷한 시간대에 함께 종료됩니다. 그러면 다음과 같이 세로로 줄이 만들어집니다.





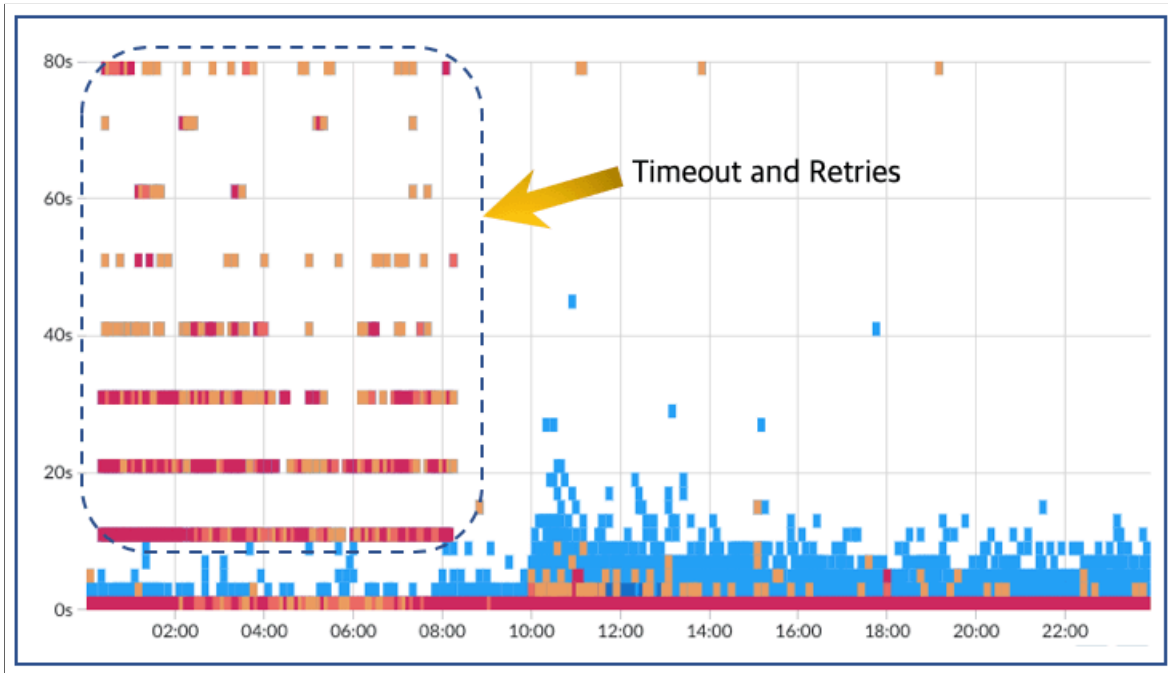
세로줄 패턴으로 락을 감지하는 것은 매우 강력한 개념입니다. 특히 마이크로 서비스 아키텍처에서는 백엔드 시스템에서 발생하는 LOCK도 동일하게 감지될 수 있습니다.



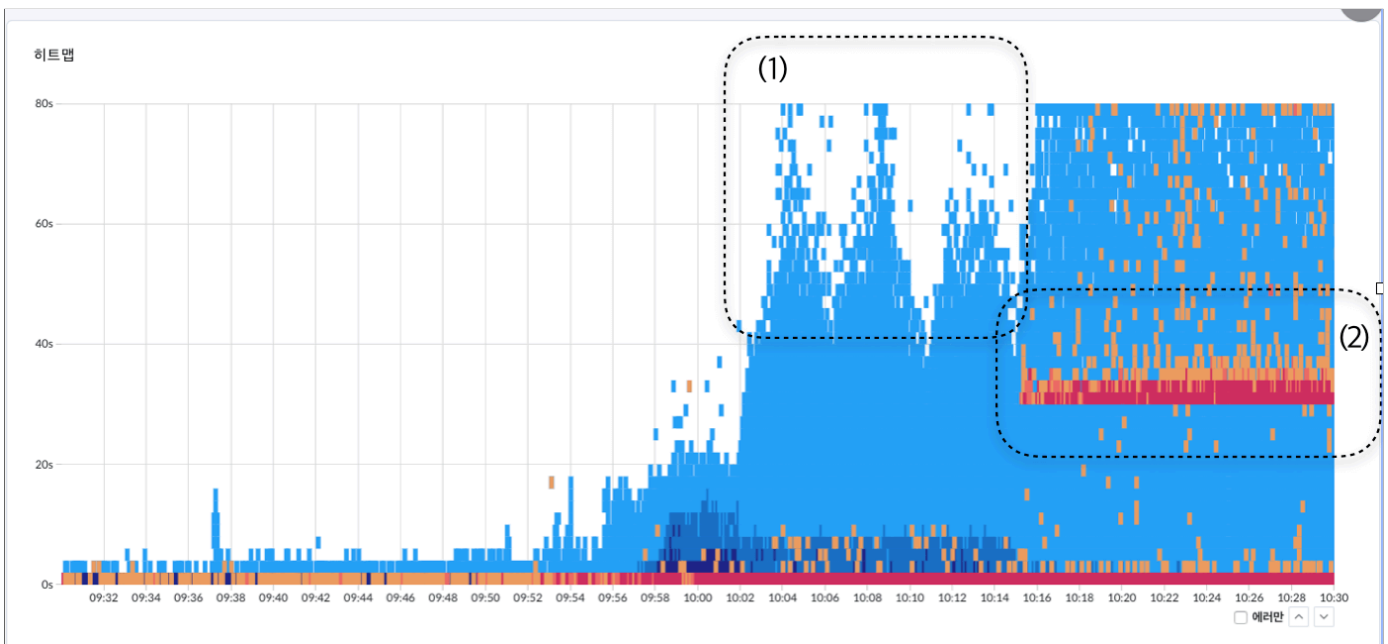
Front 애플리케이션의 응답 패턴 세로줄은 Back-End 시스템이 사용하는 DB에서 락이 발생한 경우에도 감지됩니다.

- 가로줄(타임아웃) 패턴

10초 타임아웃 조건에서 해당 자원이 부족하면 트랜잭션들은 10초 대기 후 타임아웃 에러가 발생합니다. 이때 히트맵 10초 부근에 다음과 같이 가로줄이 생깁니다.



타임아웃 이후 재시도하는 로직이 있다면 그림처럼 가로라인이 10초 단위로 반복됩니다. 다음은 실제 장애 상황의 히트맵입니다.

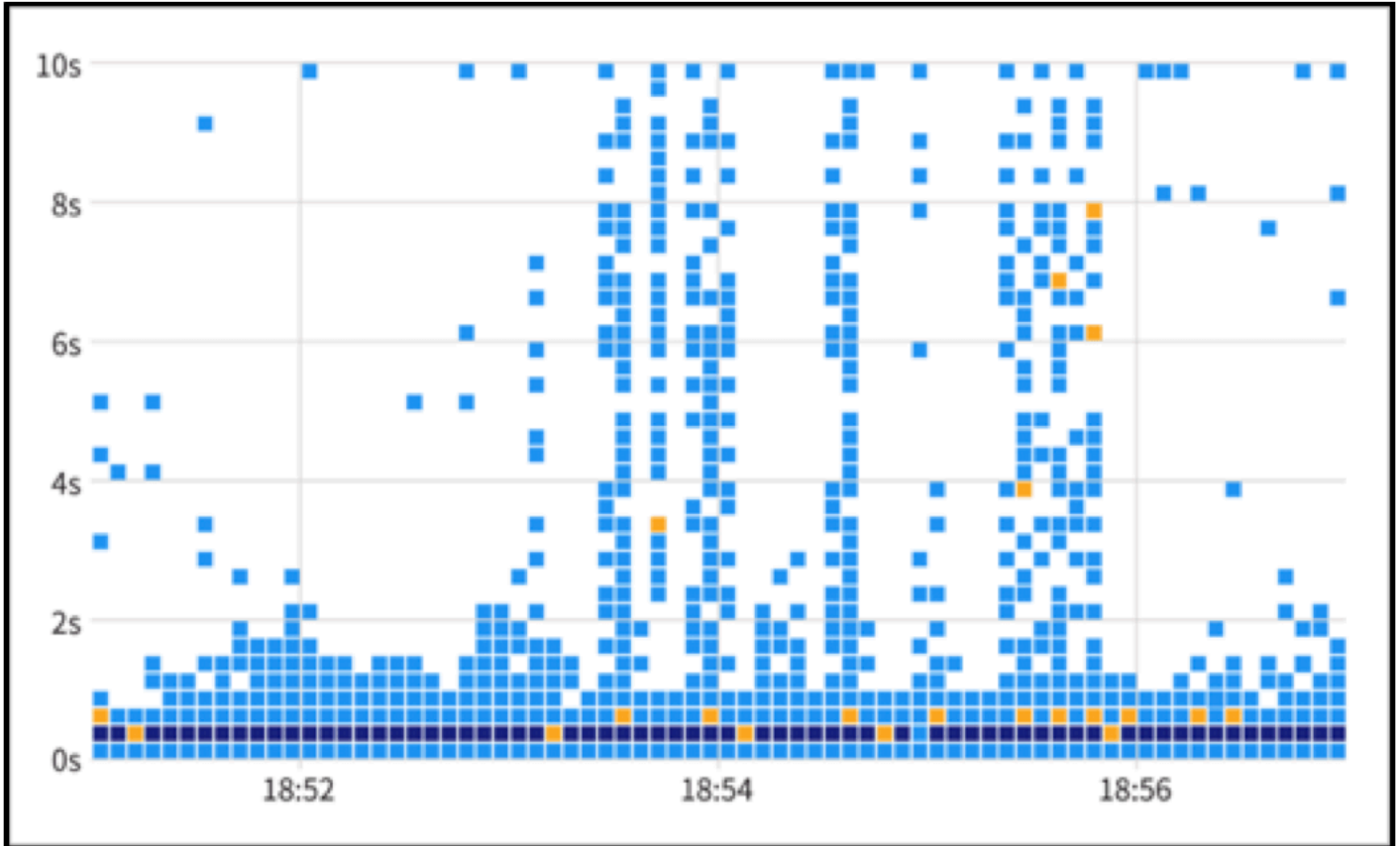


(1) 구간에서 응답시간이 급증했고 (2) 구간의 빨간 라인은 전형적인 가로 라인 패턴입니다. (1) 구간 부하 발생으로

ConnectionPool이 소진되고 (2) 구간은 ConnectionPool 부족으로 2차 타임아웃 장애가 발생한 상황입니다.

## 패턴 분석 활용

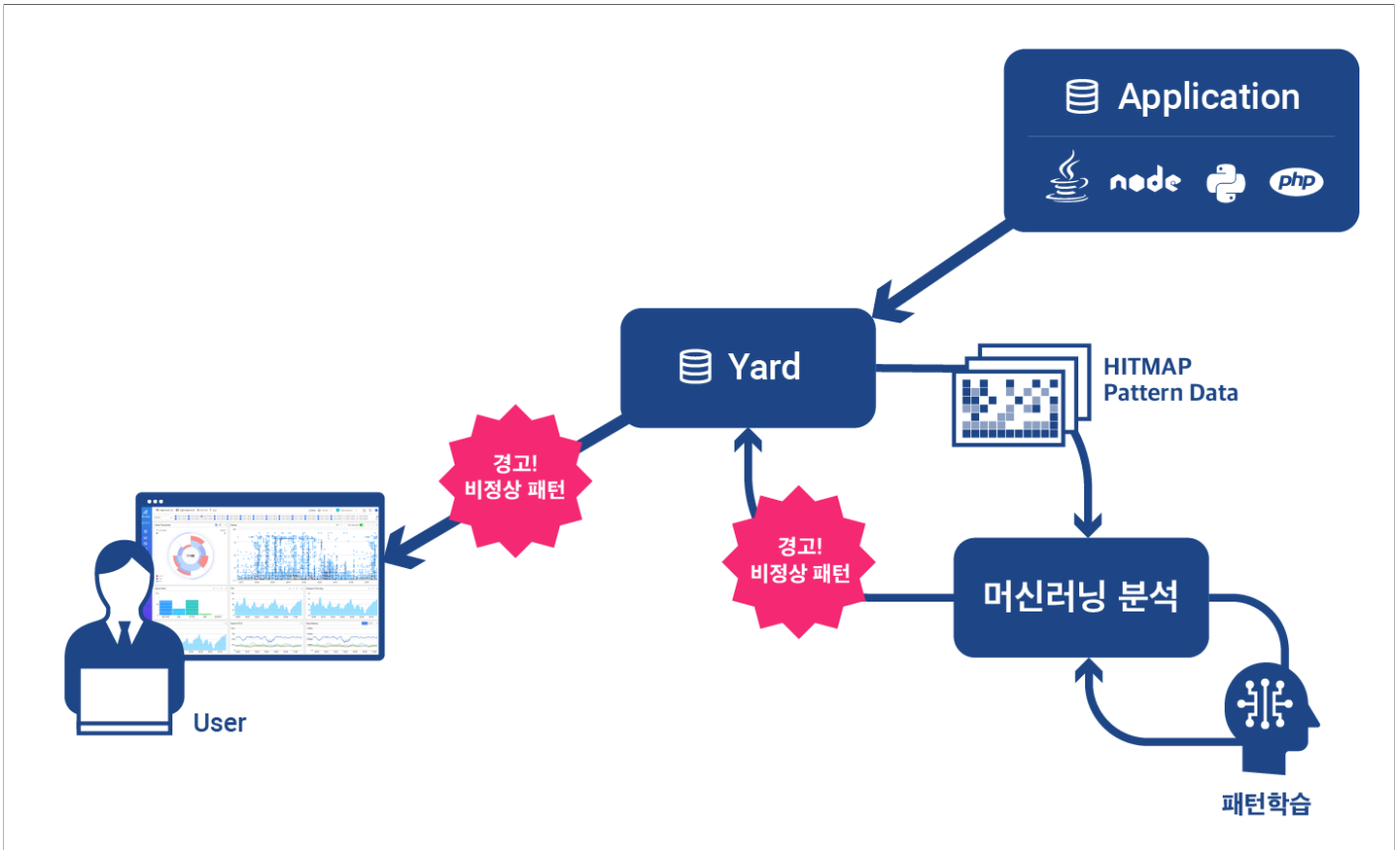
트랜잭션 응답분포에 줄이 보인다는 것은 병목이 있다는 것입니다. 일시적인 락킹이면 세로줄이 그 병목이 타임아웃으로 빠지면 가로 줄이 만들어집니다.



문제를 분석할 때 라인에 포함된 트랜잭션만을 선택적으로 분석해서 문제를 빠르게 찾아낼 수 있습니다.

## 머신러닝 기반 응답 패턴 분석

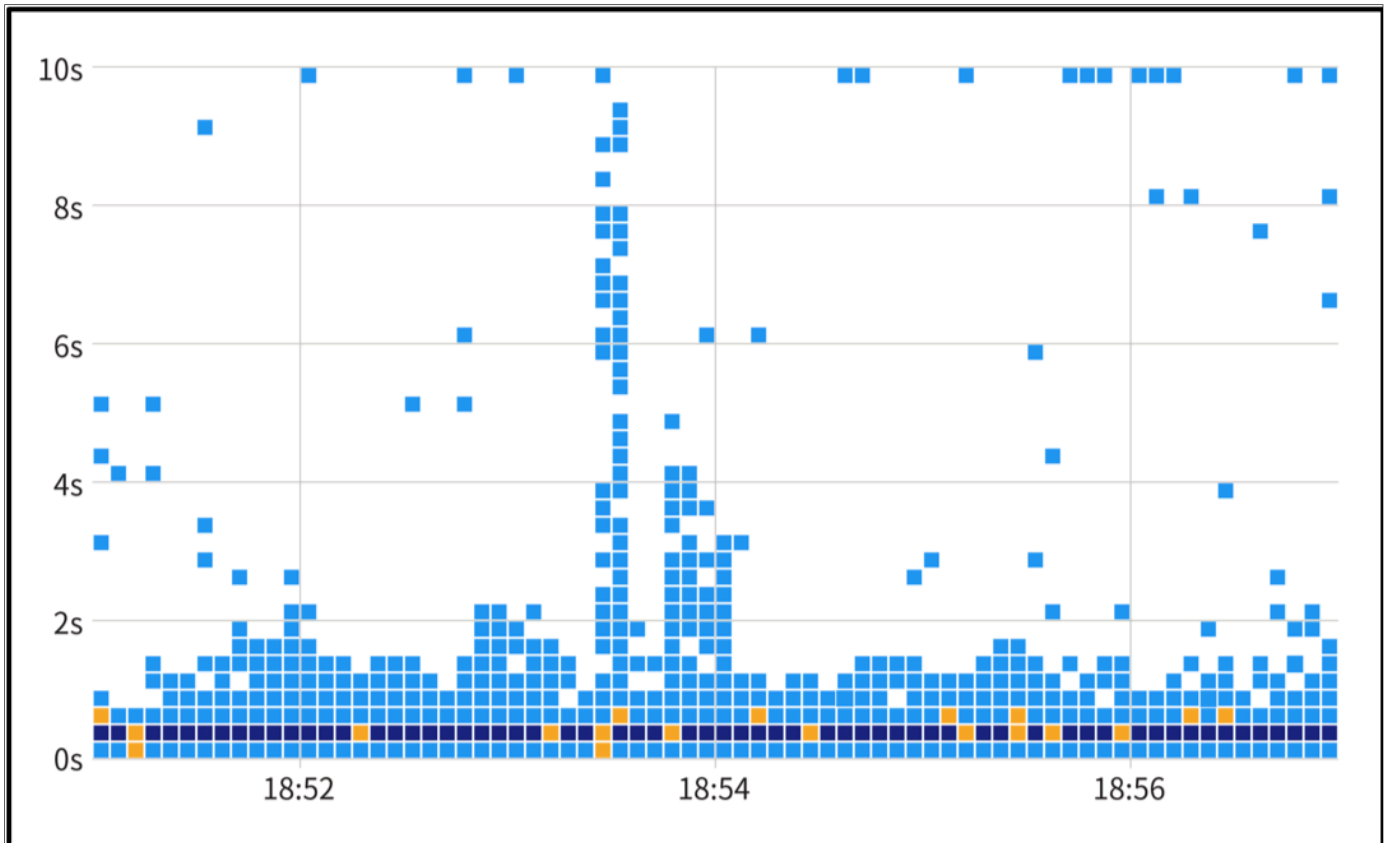
머신러닝 기술을 통해 히트맵 패턴을 분석 후 비정상 여부를 자동 감지해 경고를 발행하는 기능입니다.



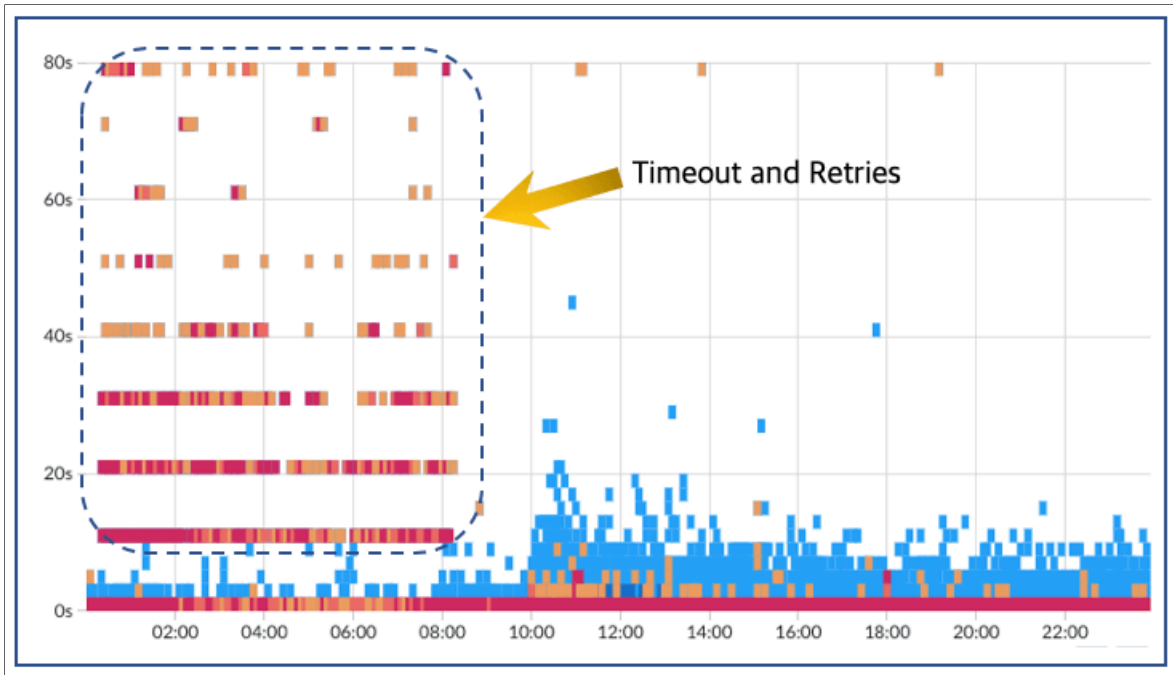
## 비정상 패턴 예시

월 수백 TB의 성능 데이터로부터 비정상 패턴을 학습하고 학습된 비정상 패턴과 유사한 패턴이 발생하는 경우 이에 대한 알람을 발행합니다.

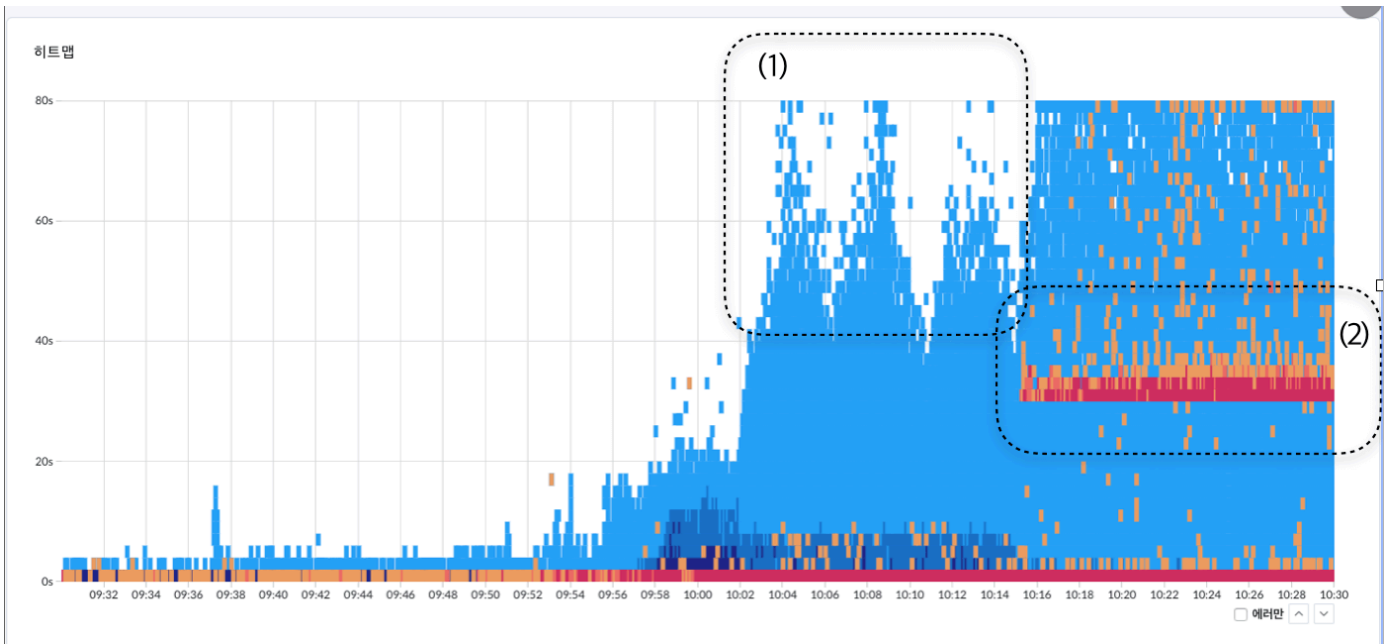
- 세로줄 패턴



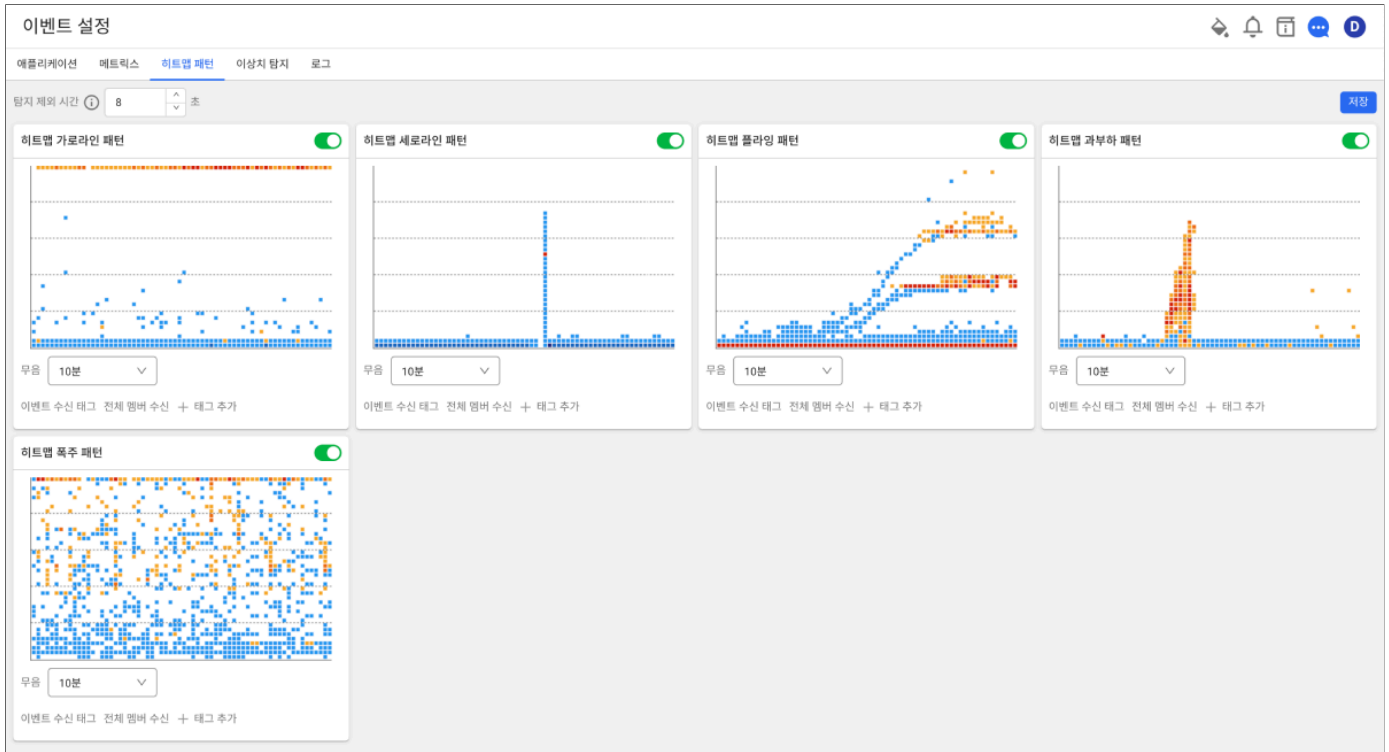
- 가로줄 패턴



• 복합 패턴



• 히트맵 알람

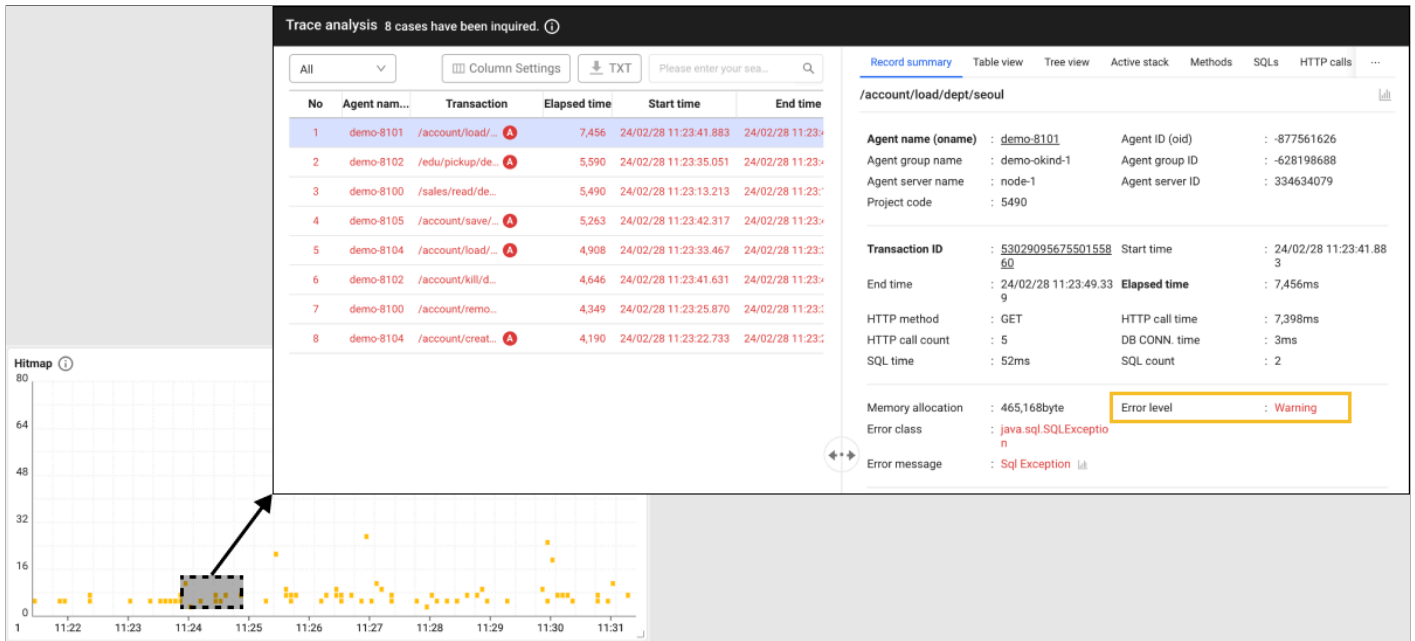


ⓘ 히트맵 트랜잭션 차트를 분석하는 방법에 대한 자세한 설명은 [다음 문서](#)를 참조하세요.

## Java 에이전트의 예외 처리(WARNING) 기준

Java 언어를 사용하는 애플리케이션 환경에서 발생하는 에러를 Java 에이전트가 예외 처리하는 기준에 대해서 안내합니다. 다음 내용은 [Java 에이전트 v2.2.27](#) 기준으로 작성되었습니다.

## 서비스 화면에서 에러 표시



히트맵 위젯, 트레이스 분석 창을 통해서 확인할 수 있는 에러 단계는 대부분 WARNING 수준에 해당합니다.

## Spring ExceptionHandler 통과 시 에러 처리

Spring 프레임워크에서는 `org.springframework.web.servlet.DispatcherServlet.processHandlerException` 메소드를 통과하는 경우 에러로 처리합니다. 다만 개발자가 Try-Catch 문을 사용하는 경우 `ExceptionHandler` 를 통과하지 않으므로 와탭에서는 에러 처리하지 않습니다.

## Status Code 400 이상 에러 처리

HTTP 응답 코드가 400 이상인 경우에는 예외 클래스(Exception Class)가 발생하지 않더라도 에러로 처리합니다.

## JDBC driver 호출 시 에러 처리

JDBC 연결 호출이나 SQL 실행 호출 시에 발생하는 에러는 에러로 처리됩니다.

## HTTP call 에러 처리

HTTP 호출 중 발생하는 에러는 `SLOW_HTTPC` , `httplib error` 등의 형태로 표시됩니다.



## 와탭 에러 처리

와탭 모니터링 서비스에서는 특정한 조건에 따라 에러로 처리합니다.

- **TOO\_MANY\_RECORDS**: fetch 건수 10000건 이상
- **SLOW\_SQL**: SQL 시간 30000ms 이상
- **SLOW\_HTTPC**: HTTP call 경과 시간이 10000ms 이상
- **httpc error**: HTTP call status code 400 이상인 경우
- **status error**: 트랜잭션 status code 400 이상인 경우

## 트랜잭션 에러 단계 INFO 처리

에이전트 설정을 통해서 일부 에러를 정상 상태로 표시하거나 무시할 수 있습니다. [히트맵](#) 위젯에서 표시되는 레벨은 **INFO**(파란색)입니다.

ⓘ 에이전트 설정은 [whatap.conf](#) 파일을 편집하거나 [관리 > 에이전트 설정](#) 메뉴에서 확인할 수 있습니다. 자세한 내용은 [다음 문서](#)를 참조하세요.

- `biz_exceptions` : 레코드 에러 표시

whatap.conf

```
biz_exceptions=io.home.test.baseapp.base.exception.custom.CustomParameterizedException
```

- `ignore_exceptions` : 레코드 에러 미표시

whatap.conf

```
ignore_exceptions=org.springframework.web.util.NestedServletException
```

- `status_ignore`

whatap.conf

```
status_ignore=408,500,200
```

- `httpc_status_ignore`

```
whatap.conf
```

```
httpc_status_ignore=403,401
```

- status\_ignore\_set

```
whatap.conf
```

```
status_ignore_set=/api/posts/test/timeout:200,/error:500
```

- httpc\_status\_url\_ignore\_set

```
whatap.conf
```

```
httpc_status_url_ignore_set=/api/posts/test/timeout:408,/api/apache/httpclient/posts/timeout:500
```

# 트랜잭션 트레이싱

## 트랜잭션 트레이싱이란?

트랜잭션 성능이 트랜잭션 시작과 종료 사이의 요약 지표들이나 속성들을 의미한다면 트랜잭션 트레이스는 트랜잭션이 수행되는 과정 중인 스텝들을 추적하는 것입니다. 트랜잭션이 느리거나 오류가 있다면 그 원인을 추적하기 위해서 수행 이력을 스텝별로 추적해야 합니다. 이것을 **트랜잭션 트레이싱**이라고 합니다.

트랜잭션 성능 추적을 위해 수집하는 스텝의 종류는 다음과 같습니다.

- **DB 연결 스텝** **START-END**  
RDB에 대한 연결에 대한 성능을 포함합니다. 스텝 정보에는 이름, 응답시간, 에러를 포함합니다.
- **SQL 스텝** **START-END**  
JDBC SQL에 대한 성능을 포함합니다. 스텝 정보에는 연결 정보, SQL문, 에러가 포함되어 있습니다.
- **HTTP Call 스텝** **START-END**  
외부 http 서비스 호출에 대한 성능을 포함합니다. 스텝 정보에는 url, host, port, 응답시간, 에러가 포함됩니다.
- **Message 스텝** **ADD**  
트레이스를 수집하는 과정에서 비정형적인 모든 구간에 대한 이력을 수집할 때 메시지 스텝을 사용합니다.
- **SOCKET 스텝** **ADD**  
Socket 오픈을 표현하는 스텝입니다.
- **METHOD 스텝** **START-END**  
메소드 응답시간을 추적합니다.
- **ACTIVE STACK 스텝** **START-END**  
액티브 스택에 대한 정보를 포함합니다. 별도 스레드가 생성하여 트레이스에 추가하는 방식으로 수집합니다.

❗ 와탭 모니터링 서비스에서 **스텝(Step)**은 **스팬(Span)**과 같은 뜻으로 사용됩니다.

# 상세 분석

홈 화면 > 프로젝트 선택 > 분석 > 히트맵

**1** 히트맵

시간 선택 < 2023/10/05 00:00 ~ 2023/10/06 00:00 1인 > 구분 에이전트 전체 에이플리케이션 전체

**2** 히트맵 트랜잭션 5,202,658 28,415 Y축 지움 예러 DB 추적

**3**

**4** 에이플리케이션 All

에이전트 명	히트 ↓	예러
All	1,669,159	9,165
demo-8105	280,764	1,568
demo-8102	280,148	1,535
demo-8101	278,052	1,549
demo-8100	277,758	1,577
demo-8103	276,727	1,452
demo-8104	275,710	1,484

이전 1 다음

**5** TX 트레이스 (All applications) 열람 선택 CSV 모두 보기 액티브 스택 멀티 트랜잭션 Search URL

시작 시간	경과 시간↑	트랜잭션	HTTP 호출...	SQL 건수	SQL 시간	SQL 패치 건...	DB 연결 시간	예러 클래스	예러 메시지
2023/10/05 10:59:50.892	11,624	/account/delete/employee/daejun	11,077	6	524	28	11		
2023/10/05 10:59:47.862	11,749	/account/save/unit/daegu	10,887	4	835	1,443	4		
2023/10/05 10:59:47.473	12,179	/edu/save/dept/pusan	10,272	6	1,871	1,593	6		
2023/10/05 10:59:44.481	12,443	/product/create/unit/seoul	11,660	3	776	0	4		
2023/10/05 10:59:46.258	12,649	/sales/pickup/division/kwangju	11,591	8	968	1,499	74		
2023/10/05 10:59:45.406	12,939	/product/read/dept/seoul	12,864	2	4	0	66		
2023/10/05 10:59:47.310	13,297	/edu/write/dept/kwangju	13,216	3	58	1,354	4		
2023/10/05 10:59:43.980	13,325	/account/save/division/jeju	12,989	7	98	681	208		
2023/10/05 10:59:48.240	13,417	/account/create/dept/kwangju	12,902	4	496	374	8		
2023/10/05 10:59:43.956	13,624	/product/create/division/jeju	13,588	2	30	0	2		
2023/10/05 10:59:46.414	13,851	/product/remove/division/daejun	13,039	6	609	1,215	180		
2023/10/05 10:59:44.541	13,925	/edu/delete/employee/daegu	13,534	6	281	1,137	87		
2023/10/05 10:59:38.400	19,094	/edu/delete/division/daegu	18,791	2	285	678	2		
2023/10/05 10:59:32.760	24,622	/account/save/division/pusan	13,184	8	171	0	77		
2023/10/05 10:59:32.867	24,646	/account/create/employee/kwangju	13,025	2	133	120	143		

1. **1** 시간 선택에서 히트맵을 조회할 기간을 선택하세요.
2. **1** 구분, 애플리케이션에서 조회 대상을 선택하세요.
3. **Q** 버튼을 선택하세요.
4. **2** 히트맵 트랜잭션에서 원하는 영역을 **3** 과 같이 드래그하세요.

5. **5** TX 트레이스에 목록이 나타나면 분석하기 원하는 트랜잭션 항목을 선택하세요.

트랜잭션의 성능 분석을 위한 클라이언트 정보 등의 속성, 트랜잭션의 처리 성능, 각 구간별 상세 수행 이력 등을 확인할 수 있는 [트랜잭션 정보](#) 창이 나타납니다.

- ⓘ • 히트맵 메뉴의 기능에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.
- 애플리케이션 대시보드의 히트맵 위젯에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

## 트랜잭션 정보 창 안내

트랜잭션 정보

레코드 요약   **테이블 뷰**   트리 뷰   메소드 요약   SQL 요약   HTTP Call 요약   트랜잭션 로그
1

/account/create/employee/daejun
📄

시작: 11/06 03:11:02.994   종료: 11/06 03:11:10.650   경과: 7,656ms   에이전트 명 (oname): demo-8104   클라이언트 IP: 77.131.96.144
2

Method   SQL   HTTP Call

📄 컬럼 선택   📄 TXT   📄 SQL   📄 📄

No	시간	갭	결과	내용
1	03:11:02.994			시작 2023-11-06 03:11:02.994
2	03:11:02.994			<b>HTTP-HEADERS</b> host=255.255.255.159 x-wtap-mst=x4716m89sq3jmt,1,zmeu5ongdb33 7,x5rs
3	03:11:02.994	0	7,656	com.virtual.web.VExec#doFilter
4	03:11:02.994	0	7,656	↳ com.virtual.web.Simula\$1#process
5	03:11:02.994	0	1,356	↳ com.virtual.dao.RemoteDAO#execute
6	03:11:02.994	0	1,356	[ApacheClient] 127.0.0.1:8104 /remote/a. 📄 <span style="border: 1px solid orange; border-radius: 50%; padding: 2px 5px;">3</span>
7	03:11:02.994			[Socket] 127.0.0.1:8104
8	03:11:04.350	1,356	0	↳ com.virtual.TestLogger#println
9	03:11:04.350	0	138	↳ com.virtual.web.Simula#execute
10	03:11:04.350	0	138	↳ com.virtual.web.Simula#execute2
11	03:11:04.350	0	138	↳ com.virtual.web.Simula#execute?
12	03:11:04.351	1	137	DELETE FROM Customers WHERE CustomerName='\$' AND ContactName='\$'
13	03:11:04.350	-1	138	↳ com.virtual.dao.DeleteDAO#execute
14	03:11:04.489	139	430	insert into emp values( \$, '\$', '\$', \$, to date('\$', '\$')...
15	03:11:04.488	-1	431	↳ com.virtual.dao.InsertDAO#execute2
16	03:11:04.488	0	431	↳ com.virtual.web.Simula#execute3
17	03:11:04.488	0	431	↳ com.virtual.web.Simula#execute2

↕

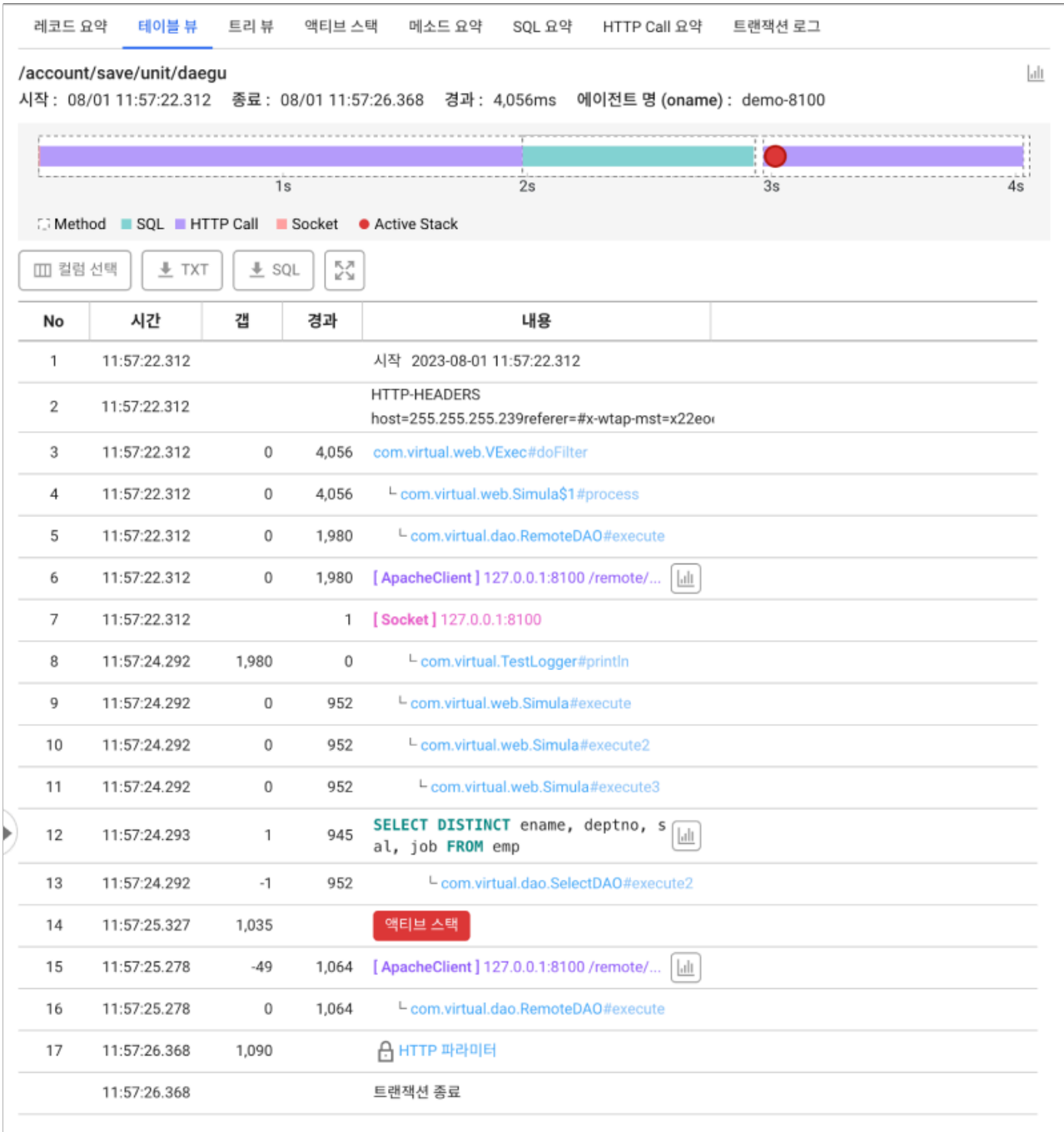
- **1** 영역에서는 스텝 정보를 다양한 형식으로 확인할 수 있습니다. 원하는 형식의 탭을 선택하세요.
- **2** 영역에서는 트랜잭션에 대한 기본 정보를 확인할 수 있습니다.
- **3** 📄: 해당 URL 또는 각 수행 구간의 통계 데이터 창이 나타납니다. 통계 데이터 창에서 **상세 >** 버튼을 선택하면 **통계/보고서 > 통계** 메뉴로 이동합니다.

ⓘ 통계 메뉴에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

- 4 SQL 스텝을 선택하면 SQL 변수와 HTTP 쿼리를 조회할 수 있습니다. 자세한 내용은 [다음 문서](#)를 참조하세요.

## 테이블 뷰

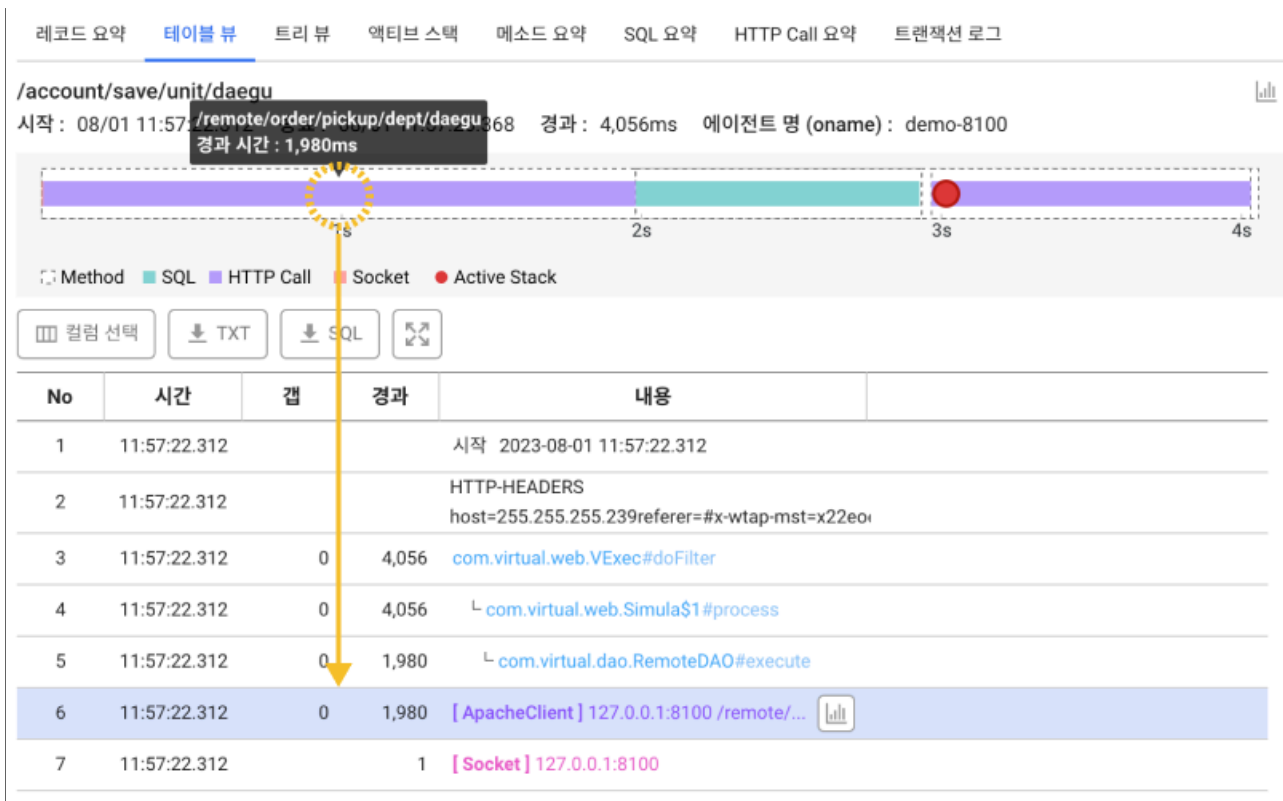
테이블 뷰 탭에서는 트랜잭션의 수행 과정을 시간의 순서대로 확인할 수 있습니다.



- 트랜잭션을 수행 구간별로 분류해 다이어그램을 통해 확인할 수 있습니다. 전체 경과 시간 중 각 구간별 소요된 시간, 가장 오래 소요된 구간을 빠르게 파악할 수 있습니다.



- 다이어그램의 각 구간을 선택하면 해당 스텝이 위치한 테이블 목록으로 이동합니다.



- 빨간색 원(●)이 위치한 영역은 액티브 스택이 수집된 순간입니다. 빨간색 원을 선택하면 **액티브 스택** 버튼이 위치한 테이블 목록으로 이동할 수 있습니다. **액티브 스택** 버튼을 선택하면 해당 구간 동안 수행된 스텝 정보를 확인할 수 있습니다.

ⓘ ○ 트랜잭션 목록에서 **A** 아이콘이 표시된 항목에서 확인할 수 있습니다.

○ 액티브 스택에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

- **☰ 컬럼 선택**: 테이블 목록에 컬럼으로 메모리 누적 정보 및 CPU 누적 정보를 추가하거나 감출 수 있습니다. 컬럼 정보는 다음과 같습니다.
  - **No**: 스텝의 발생 순서
  - **시간**: 각 스텝의 시작 시각
  - **갭**: 해당 스텝의 시작 시각부터 다음 스텝으로 넘어가기까지 대기 시간, 외부 요인으로 지연될 경우 경과 시간과 차이가 있을 수 있습니다.
  - **경과**: 각 메소드 시작부터 종료까지 총 소요 시간
  - **내용**: 해당 스텝의 세부 수행 내용

- **↓ TXT**: 트랜잭션 기본 정보 및 구간별 수행 정보를 txt 형식의 파일로 다운로드할 수 있습니다.
- **↓ SQL**: 트랜잭션 기본 정보 및 SQL 수행 정보를 txt 형식의 파일로 다운로드할 수 있습니다.
- **🔍**: 트랜잭션 요약 정보와 다이어그램을 감추고 테이블 목록만 확인할 수 있습니다. **⌵** 버튼을 선택하면 감춰진 정보를 다시 표시합니다. 테이블 목록이 긴 경우 이 기능을 이용하면 유용합니다.
- SQL 스텝을 선택하면 파라미터를 조회할 수 있는 **SQL** 창이 나타납니다. 자세한 내용은 [다음 문서](#)를 참조하세요.

## 레코드 요약

<a href="#">레코드 요약</a>	<a href="#">테이블 뷰</a>	<a href="#">트리 뷰</a>	<a href="#">메소드 요약</a>	<a href="#">SQL 요약</a>	<a href="#">HTTP Call 요약</a>	<a href="#">트랜잭션 로그</a>
<b>/account/remove/dept/pusan</b> <span style="float: right;">📄</span>						
<b>에이전트 명 (oname)</b>	: demo-8101		<b>에이전트 ID (oid)</b>	: -877561626		
에이전트 그룹 명	: demo-okind-1		<b>에이전트 그룹 ID</b>	: -628198688		
에이전트 서버 명	: node-1		<b>에이전트 서버 ID</b>	: 334634079		
프로젝트 코드	: 5490					
<hr/>						
<b>트랜잭션 ID</b>	: <u>-3538973756714763020</u>		<b>시작 시간</b>	: 23/10/06 08:48:11.706		
종료 시간	: 23/10/06 08:48:14.585		<b>경과 시간</b>	: 2,879ms		
HTTP 메소드	: GET		<b>HTTP 호출 시간</b>	: 2,409ms		
HTTP 호출 건수	: 2		<b>DB 연결 시간</b>	: 139ms		
SQL 시간	: 324ms		<b>SQL 건수</b>	: 2		
SQL 패치 시간	: 6ms		<b>SQL 패치 건수</b>	: 647		
<hr/>						
<b>CPU 사용 시간</b>	: 2ms		<b>메모리 할당량</b>	: 197,232byte		
<hr/>						
<b>클라이언트 IP</b>	: 168.196.231.239		<b>도메인</b>	: 255.255.255.159		
WClientID	: 4545076460797978934		<b>국가</b>	: BR		
도시	: Acailandia		<b>운영체제</b>	: Windows		
클라이언트 타입	: IE		<b>클라이언트 명</b>	: Other		
<b>상태</b>	: 200		<b>Referer</b>	: http://www.test.com/account/load/employee/seoul		
<b>원본 URL/트랜잭션</b>	: /account/remove/dept/pusan					
<b>유저 에이전트</b>	: Mozilla/5.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4325) <span style="float: right;">📄</span>					

트랜잭션의 기본 정보를 확인하려면 **레코드 요약** 탭을 선택하세요. 트랜잭션의 기본 정보 및 해당 트랜잭션이 수행된 에이전트 관련

정보, 메모리 할당 사용량, 클라이언트 관련 정보를 확인할 수 있습니다.

## 트랜잭션 속성

분류	속성	설명
에이전트	<b>에이전트 명(oname)</b>	에이전트 오브젝트 이름
	에이전트 ID (oid)	에이전트 오브젝트 아이디
	에이전트 그룹 명	okind 옵션으로 설정된 에이전트 그룹 이름
	에이전트 그룹 ID	에이전트 그룹 아이디
	에이전트 서버 명	ondoe 옵션으로 설정한 에이전트 서버 이름
	에이전트 서버 ID	에이전트 서버 아이디
	프로젝트 코드	와탭 모니터링 서비스에 등록된 프로젝트의 식별 코드
트랜잭션	<b>트랜잭션 ID</b>	트랜잭션 아이디
	시작 시간	트랜잭션 시작 시각
	종료 시간	트랜잭션 종료 시각
	<b>경과 시간</b>	트랜잭션 수행 시간
	HTTP 메소드	HTTP 메소드 - GET, POST, PUT, HEAD 등
	HTTP 호출 시간	외부 Http Call 시간
	HTTP 호출 건수	외부 HTTP Call 건수
	DB 연결 시간	DB와 연결된 시간
	SQL 시간	SQL 수행 시간

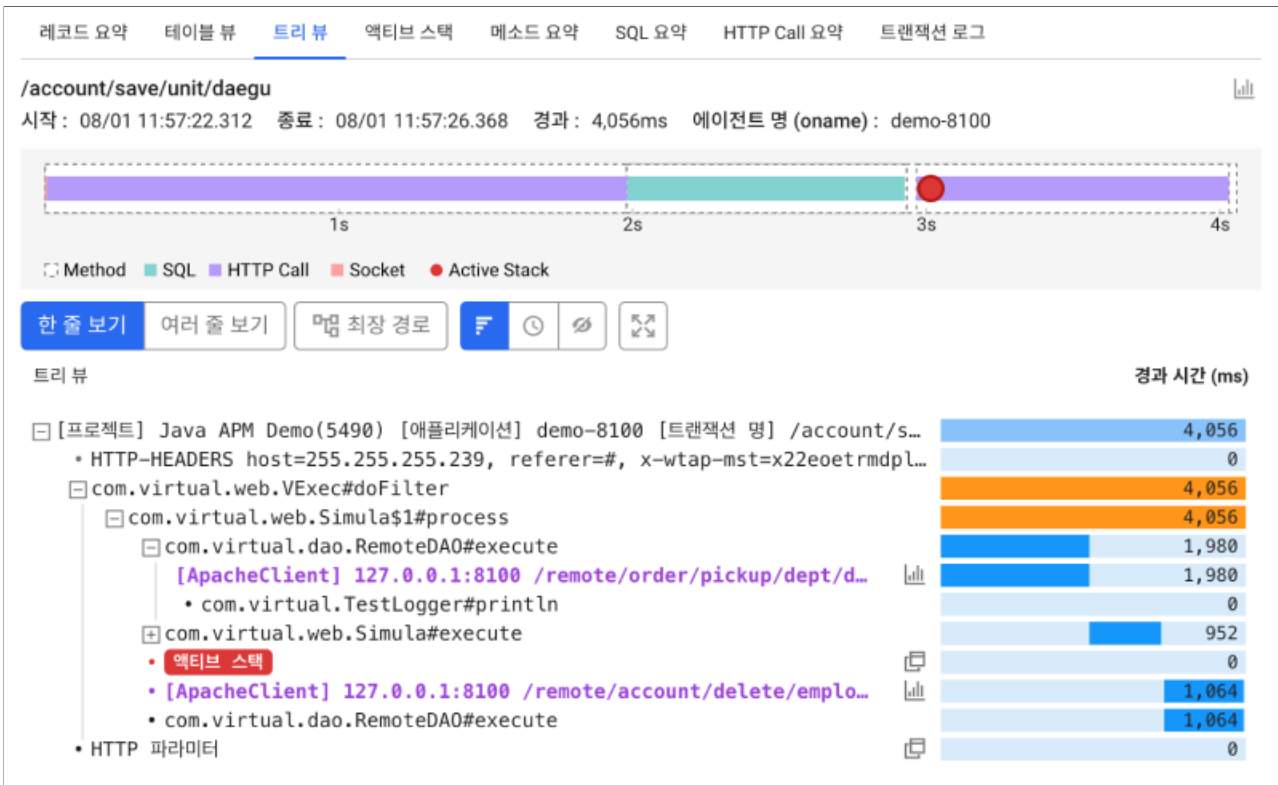
분류	속성	설명
	SQL 건수	SQL 수행 건수
	SQL 패치 시간	레코드를 조회하는 데 걸린 시간 (중첩이 발생하거나 타 로직이 포함될 수 있음)
	SQL 패치 건수	SQL에서 데이터를 조회한 레코드 건수
멀티 트랜잭션	CallerUrl	부모 트랜잭션의 트랜잭션 주소
	CallerPcode	부모 트랜잭션이 수집된 프로젝트 코드
	CallerOid	부모 트랜잭션의 에이전트 오브젝트 아이디
	CallerOName	부모 트랜잭션의 에이전트 이름
	CallerOkind	부모 트랜잭션의 에이전트 그룹 ID
	CallerOkindName	부모 트랜잭션의 에이전트 그룹 이름
자원	CPU 사용 시간	트랜잭션이 사용한 CPU 사용량
	메모리 할당량	트랜잭션이 사용한 메모리 사용량
클라이언트	클라이언트 IP	클라이언트의 IP 주소
	도메인	클라이언트가 접속한 IP 주소에 지정된 인터넷 주소
	WClientID	클라이언트의 ID
	국가	클라이언트의 국가 정보
	도시	클라이언트의 국가 내 도시 정보
	운영체제	브라우저가 실행되는 운영 체제 환경
	클라이언트 타입	클라이언트가 이용한 브라우저 종류

분류	속성	설명
	클라이언트 명	클라이언트가 이용한 기기 이름
	상태	클라이언트의 요청에 대한 HTTP 상태 코드
	Referer	클라이언트가 브라우저에서 이전에 이용한 페이지 주소 또는 유입 경로
	원본 URL/트랜잭션	트랜잭션 이름과 원본 URL의 조합 데이터 원본 URL이 있을 경우 원본 URL 표시, 원본 URL이 없을 경우 트랜잭션 표시
	유저 에이전트	클라이언트 관련 정보를 추출하는데 이용하는 브라우저 정보

- ⓘ 애플리케이션 종류나 설정, 스텝의 종류에 따라 수집하는 정보는 달라질 수 있습니다.
- 와탭은 클라이언트와 관련한 정보를 기본 저장합니다. 사용자 데이터 수집과 관련한 에이전트 설정에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

## 트리 뷰

트랜잭션 수행 과정을 트리 형식으로 확인하려면 **트리뷰** 탭을 선택하세요. 각 트랜잭션과 그에 속한 트레이스의 세부 정보, 트레이스의 시작 시간 및 소요 시간, 호출 관계를 확인할 수 있습니다. 다이어그램의 각 구간을 선택하면 해당 스텝이 위치한 트리로 이동합니다.




- **한 줄 보기**: 각 구간 별 수행 정보에 표시된 텍스트를 한 줄로 표시해 트리 형식을 간격하게 정리할 수 있습니다.
- **여러 줄 보기**: 각 구간 별 수행 정보에 표시된 텍스트를 줄바꿈해 모두 표시합니다.
- **최장 경로**: 가장 긴 경로로 이동할 수 있습니다.
- **시간바 표시**: 경과 시간을 막대 형식의 차트로 표시합니다.
- **시간 표시**: 각 구간별 타임 스탬프, 갭, 경과 시간을 텍스트 형식으로 표시합니다.
  - 8초 이상: **초과 지연** 상태로 **빨간색**으로 표현합니다.
  - 3초 이상 8초 미만: **지연** 상태로 **주황색**으로 표현합니다.
  - 3초 미만: **정상** 상태로 **파란색**으로 표현합니다.
- **시간 숨기기**: 시간 정보를 숨깁니다.
- **트랜잭션 요약 정보와 다이어그램을 감추고 트리뷰만 확인할 수 있습니다.** **버튼**을 선택하면 감춰진 정보를 다시 표시합니다. 트리뷰 목록이 긴 경우 이 기능을 이용하면 유용합니다.
- **SQL 변수와 HTTP 쿼리를 조회할 수 있는 창이 나타납니다.** 자세한 내용은 [다음 문서](#)를 참조하세요.

- ① 시작 및 소요 시간의 경우 트랜잭션 호출 환경에 따라 발생하는 시차를 상위 트랜잭션 내 트레이스와 매핑을 통해 보정하여 표현하기 때문에 실제 수집된 시간 데이터와 차이가 발생할 수 있습니다.

## 액티브 스택

레코드 요약   테이블 뷰   트리 뷰   **액티브 스택**   메소드 요약   SQL 요약   HTTP Call 요약   트랜잭션 로그

/account/save/unit/daegu 

시작: 08/01 11:57:22.312   종료: 08/01 11:57:26.368   경과: 4,056ms   에이전트 명 (oname): demo-8100

액티브 스택 (1)

No	시간	
14	11:57:25.327	<pre> java.base@17.0.7/sun.nio.ch.SocketDispatcher.read0(Native Method) java.base@17.0.7/sun.nio.ch.SocketDispatcher.read(SocketDispatcher.java:47) java.base@17.0.7/sun.nio.ch.NioSocketImpl.tryRead(NioSocketImpl.java:261) java.base@17.0.7/sun.nio.ch.NioSocketImpl.implRead(NioSocketImpl.java:312) java.base@17.0.7/sun.nio.ch.NioSocketImpl.read(NioSocketImpl.java:350) java.base@17.0.7/sun.nio.ch.NioSocketImpl\$1.read(NioSocketImpl.java:803) java.base@17.0.7/java.net.Socket\$SocketInputStream.read(Socket.java:966) app//org.apache.http.impl.io.SessionInputBufferImpl.streamRead(SessionInputBufferImpl.java:139) app//org.apache.http.impl.io.SessionInputBufferImpl.fillBuffer(SessionInputBufferImpl.java:155) app//org.apache.http.impl.io.SessionInputBufferImpl.readLine(SessionInputBufferImpl.java:284) app//org.apache.http.impl.conn.DefaultHttpResponseParser.parseHead(DefaultHttpResponseParser.java:140) app//org.apache.http.impl.conn.DefaultHttpResponseParser.parseHead(DefaultHttpResponseParser.java:57) app//org.apache.http.impl.io.AbstractMessageParser.parse(AbstractMessageParser.java:261) </pre>

액티브 스택을 수집한 구간과 시간, 관련한 클래스, 메소드 정보를 확인하려면 **액티브 스택** 탭을 선택하세요.

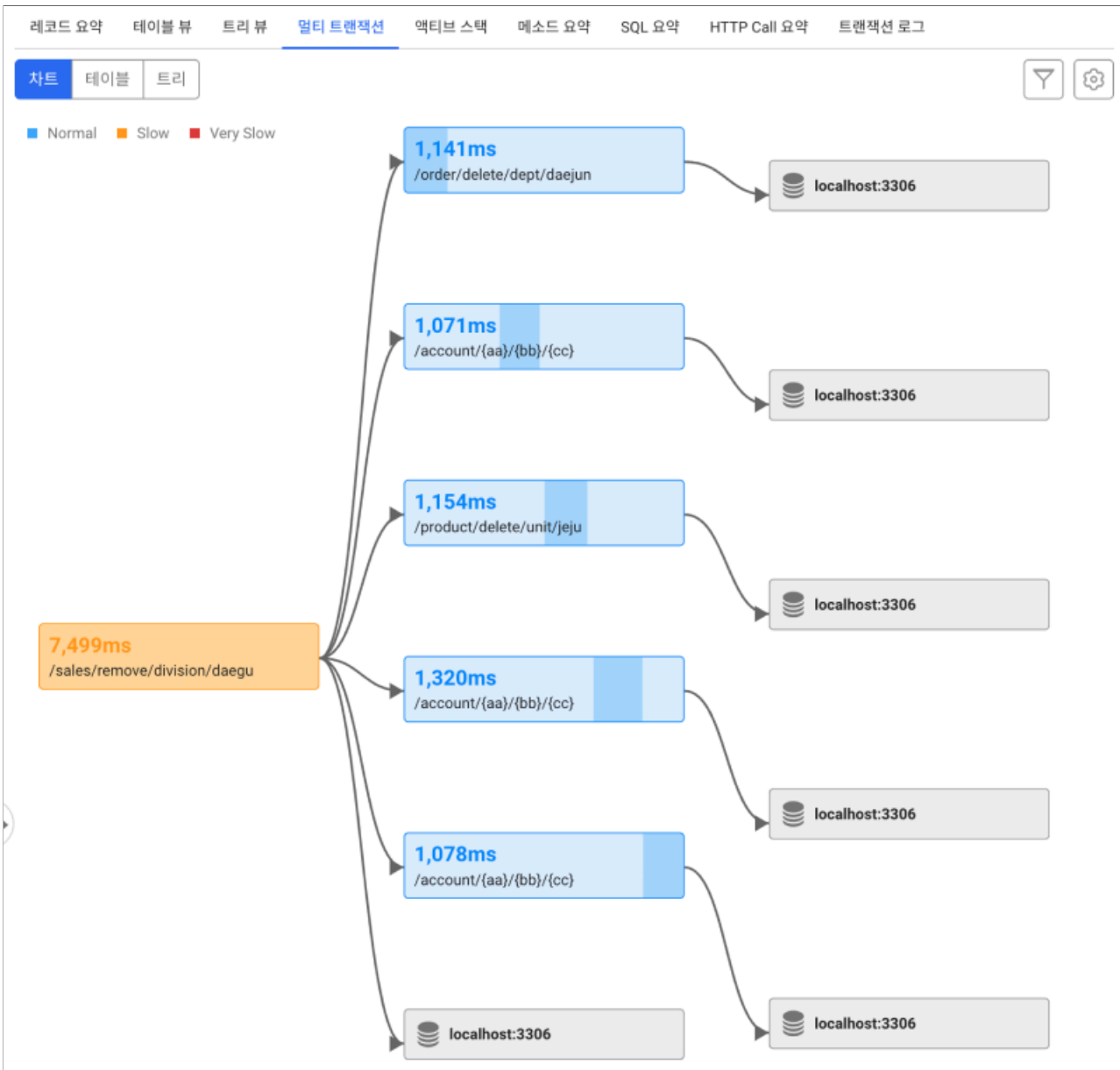
- ① • 트랜잭션 목록에서 **A** 아이콘이 표시된 항목에서 확인할 수 있습니다.
  - 액티브 스택에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.


## 멀티 트랜잭션

멀티 트랜잭션은 다른 에이전트나 프로젝트와의 연관된 트랜잭션을 의미합니다. [멀티 트랜잭션](#) 탭에서는 와탭 모니터링 서비스에 등록된 애플리케이션 간의 호출 관계를 확인할 수 있습니다.

- ✔ 멀티 트랜잭션을 추적하려면 [관리 > 에이전트 설정](#) 메뉴에서 `mtrace_enabled` 옵션을 `true` 로 설정하세요. 에이전트 설정에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.





- **차트**: 각 트랜잭션의 호출 관계를 플로우 차트 형식으로 제공합니다. 트랜잭션 노드를 선택하면 해당 트랜잭션 노드에 대한 트레이스 분석 정보를 확인할 수 있습니다. 차트 모드에서는 마우스를 이용해 원하는 위치로 이동하거나 스크롤을 통해서 확대, 축소할 수 있습니다.
  - **차트 뷰 설정**: 차트에 표시할 요소를 표시하거나 숨길 수 있습니다.
- **테이블**: 테이블 형식으로 멀티 트랜잭션 내에 포함된 각 트랜잭션 별 정보를 확인할 수 있습니다.  컬럼 선택 아이콘을 선택해

테이블 헤더 컬럼을 편집할 수 있습니다. 각 트랜잭션 항목을 선택하면 트레이스 분석 정보를 확인할 수 있습니다.

- **트리**: 트리 형식으로 트랜잭션 간의 호출 관계를 파악할 수 있습니다. 관련한 부가 기능은 **트리뷰** 탭의 기능과 같습니다.
- **프로젝트 선택**: 차트에 표시될 프로젝트를 선택하거나 해제할 수 있습니다.

- ⓘ • 트랜잭션 목록에서 **M** 아이콘이 표시된 항목에서 확인할 수 있습니다.
  - 멀티 트랜잭션에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

## 메소드 요약

레코드 요약	테이블 뷰	트리 뷰	액티브 스택	메소드 요약	SQL 요약	HTTP Call 요약	트랜잭션 로그
/account/save/unit/daegu 							
No	클래스	메소드	파라미터/리턴	건수	합계 (ms)	최대 (ms)	
1	com.virtual.dao.Remot...	execute	()V	2	3,044	1,980	
2	com.virtual.web.VExec	doFilter	(Ljavax/servlet/Servlet...	1	4,056	4,056	
3	com.virtual.web.Simul...	process	()V	1	4,056	4,056	
4	com.virtual.TestLogger	println	(Ljava/lang/String;)V	1	0	0	
5	com.virtual.web.Simula	execute	()V	1	952	952	
6	com.virtual.web.Simula	execute2	()V	1	952	952	
7	com.virtual.web.Simula	execute3	()V	1	952	952	
8	com.virtual.dao.Select...	execute2	()V	1	952	952	

메소드 정보만을 확인하려면 **메소드 요약** 탭을 선택하세요. 에이전트에 추적이 설정된 메소드 이름과 소요 시간을 표시합니다. 불필요한 로직이 반복 실행되는 경우, 낮은 건수에 비해 실행 시간이 오래 걸리는 경우를 파악해 메소드 로직 개선을 위한 분석 정보로 활용할 수 있습니다.

- ⓘ • 메소드(method)와 관련한 에이전트 설정 옵션에 대한 자세한 내용은 [다음 문서](#)를 참조하세요. 이용 중인 상품에 따라 메소드 추적 지원 여부는 다를 수 있습니다.

## SQL 요약

레코드 요약   테이블 뷰   트리 뷰   액티브 스택   메소드 요약 <b>SQL 요약</b> HTTP Call 요약   트랜잭션 로그						
/account/save/unit/daegu 						
No	데이터베이스	SQL	건수	합계 (ms)	최대 (ms)	
1	jdbc:mysql://localhost:...	SELECT DISTINCT ena...	1	945	945	

SQL문에 대한 정보를 확인하려면 [SQL 요약](#) 탭을 선택하세요. 불필요하게 반복 실행되는 경우, 낮은 건수에 비해 실행 시간이 오래 걸리는 경우를 파악해 쿼리 성능 개선을 위한 분석 정보로 활용할 수 있습니다.

## HTTP Call 요약

레코드 요약   테이블 뷰   트리 뷰   액티브 스택   메소드 요약   SQL 요약 <b>HTTP Call 요약</b> 트랜잭션 로그						
/account/save/unit/daegu 						
No	원격 서버	URL	건수	합계 (ms)	최대 (ms)	
1	127.0.0.1:8100	/remote/order/pickup/...	1	1,980	1,980	
2	127.0.0.1:8100	/remote/account/delet...	1	1,064	1,064	

HTTP 호출의 호출 건수, 합계 시간, 평균 시간 등을 확인하려면 [HTTP Call 요약](#) 탭을 선택하세요. 불필요한 외부 호출이 반복 실행되는 경우, 낮은 건수에 비해 실행 시간이 오래 걸리는 경우를 파악해 트랜잭션 지연 요인이 외부인지 내부인지를 파악하는 분석 정보로 활용할 수 있습니다.

## 트랜잭션 로그

트랜잭션과 관련한 로그 정보를 확인하려면 [트랜잭션 로그](#) 탭을 선택하세요.

레코드 요약   테이블 뷰   트리 뷰   액티브 스택   메소드 요약   SQL 요약   HTTP Call 요약   **트랜잭션 로그**

/account/save/unit/daegu

키워드(를) 입력해주세요 🔍 🏠 ⚙️

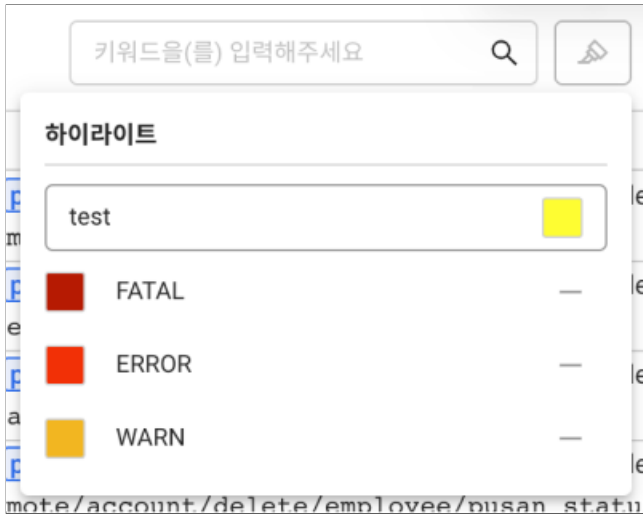
▶	oname	타임스탬프	로그
▶	demo-8100	2023-08-01 11:57:24.293	<a href="#">@txid</a> 5501194250256405697 <a href="#">pcode</a> 5490 <a href="#">oname</a> demo-8100 <a href="#">onodeName</a> node-0 http://127.0.0.1:8100/remote/order/pickup/dept/daegu status=200
▶	demo-8100	2023-08-01 11:57:25.245	<a href="#">@txid</a> 5501194250256405697 <a href="#">pcode</a> 5490 <a href="#">oname</a> demo-8100 <a href="#">onodeName</a> node-0 select distinct ename, deptno, sal, job from emp
▶	demo-8100	2023-08-01 11:57:25.279	<a href="#">@txid</a> 5501194250256405697 <a href="#">pcode</a> 5490 <a href="#">oname</a> demo-8100 <a href="#">onodeName</a> node-0 select productmodelid, name
▶	demo-8100	2023-08-01 11:57:26.342	<a href="#">@txid</a> 5501194250256405697 <a href="#">pcode</a> 5490 <a href="#">oname</a> demo-8100 <a href="#">onodeName</a> node-0 http://127.0.0.1:8100/remote/account/delete/employee/pusan status=20
▶	demo-8100	2023-08-01 11:57:26.369	<a href="#">@txid</a> 5501194250256405697 <a href="#">pcode</a> 5490 <a href="#">oname</a> demo-8100 <a href="#">onodeName</a> node-0 select ename, sal+1000 from emp

- **키워드 검색:** 키워드 검색란에 검색하려는 텍스트를 입력한 다음 엔터 키를 입력하거나 🔍 버튼을 선택하세요. 입력한 텍스트와 일치하는 키워드를 하이라이트 표시합니다.
- 🏠 **키워드 하이라이트:** 키워드와 색상을 설정하면 자동으로 로그 목록에서 키워드와 일치하는 텍스트를 하이라이트 표시합니다.
- ⚙️ **테이블 설정:** 로그 목록 테이블의 헤더 컬럼을 설정할 수 있습니다.

- ⓘ • **트랜잭션 로그** 탭을 활성화하려면 로그와 관련된 에이전트 설정 옵션을 적용해야 합니다. 로그 설정에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.
  - **트랜잭션 로그** 탭은 **로그 조회** 권한이 가진 멤버만 진입할 수 있습니다. 멤버 권한에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

## 키워드 하이라이트 설정하기

1. 하이라이트 표시할 키워드를 추가하려면 🏠 버튼을 선택하세요.



2. 원하는 키워드를 입력하세요.
3. 색상 아이콘을 선택한 다음 원하는 색상을 선택하세요.
4. 엔터를 입력하세요.

추가한 키워드가 목록에 생성됩니다.

## 키워드 하이라이트 삭제하기

하이라이트 목록에서 이용하지 않는 항목을 삭제하려면 — 버튼을 선택하세요.

# 부가 기능

## HTTP 파라미터 조회

**테이블 뷰** 탭에서 해당 트랜잭션의 HTTP 파라미터를 조회할 수 있습니다.

1. 페이지 아래로 스크롤해 트랜잭션 수행의 가장 마지막 단계로 이동하세요.
2. ☰ **HTTP 파라미터** 항목을 선택하세요.
3. **HTTP-PARAMETERS** 창이 나타나면 **비밀번호** 버튼을 선택하세요.
4. 설정한 Param Key를 입력하세요.

가려진 매개 변수를 확인할 수 있습니다.

ⓘ • HTTP 파라미터와 관련한 에이전트 설정은 [다음 문서](#)를 참조하세요.

- ❗ **비밀번호**: 복호화된 파라미터 값을 확인할 수 있습니다. 비밀번호는 `WHATAP_HOME/security.conf(paramkey.txt)` 파일 내 6자리 문자열입니다. 다른 문자열로 변경 가능합니다. SQL 변수 조회, HTTP 쿼리 조회, Thread 중지에도 필요합니다. 자세한 내용은 [다음 문서](#)를 참조하세요.

## SQL 파라미터 조회

테이블 뷰 탭에서 SQL 스텝을 선택하거나 트리 뷰 탭에서  버튼을 선택하세요. 파라미터를 조회할 수 있는 SQL 창이 나타납니다.



연결 URL jdbc:mysql://localhost:3306,localhost:3310/fake@BasicDataSource

CPU 누적 3ms

메모리 누적 329,752byte

쿼리 (Hash: -2057375474)

SID 107


```
insert into
  dept
values(
  #, '#', '#')
```

치환 변수

파라메타가 존재하지 않거나 복호화에 실패하였습니다.  
복호화를 위하여 에이전트 설치 경로의 [paramkey.txt]에서 비밀번호를 확인해주세요.

- **비밀번호**: 복호화된 파라미터 값을 확인할 수 있습니다. 비밀번호는 `WHATAP_HOME/security.conf(paramkey.txt)` 파일 내 6자리 문자열입니다. 다른 문자열로 변경 가능합니다.

❗ **비밀번호**는 SQL 변수 조회, HTTP 쿼리 조회, Thread 중지에도 필요합니다. 자세한 내용은 [다음 문서](#)를 참조하세요.

- **SQL 포매팅**: SQL 문장을 들여쓰기 및 포매팅하여 가독성을 높일 수 있습니다.
- : SQL 문장을 클립보드에 복사할 수 있습니다.
- **통계**: [통계/보고서](#) > [통계](#) 메뉴의 **SQL** 탭으로 이동합니다.

- ✔ SQL 변수와 HTTP 쿼리를 조회하려면 다음 옵션을 에이전트 설정에 추가하세요.
  - SQL 파라미터 정보 기록과 관련한 에이전트 설정은 [다음 문서](#)를 참조하세요.
  - HTTP 파라미터 정보 기록과 관련한 에이전트 설정은 [다음 문서](#)를 참조하세요.



whatap.conf

# SQL 파라미터 조회 옵션: 옵션이 적용되면 SQL 파라미터를 암호화하여 수집합니다.  
profile\_sql\_param\_enabled=true

# HTTP 파라미터 조회 옵션: 옵션이 적용되면 HTTP 쿼리 파라미터를 암호화하여 수집합니다.  
profile\_http\_parameter\_enabled=true

# 액티브 트랜잭션

진행 중인 트랜잭션을 액티브 트랜잭션이라고 합니다. **액티브 트랜잭션에서 정기적으로 덤프한 스택을 액티브 스택**이라 합니다.

① 와탭 에이전트는 매 10초(옵션 가능)마다 액티브 트랜잭션에 대해서 액티브 스택을 덤프하고 이것을 서버에 전송합니다. `active_stack_second=10`

## 컴팩트한 액티브 스택 수집

액티브 스택은 스레드 덤프를 정기적으로 수행하기 때문에 잘못 구현되면 에이전트에 오버헤드가 커질 수 있습니다. 와탭은 에이전트 부하를 최소화하면서 액티브 스택을 수집하기 위해 다양한 옵션들을 가지고 있습니다.

👉 [사이트맵](#) > [스레드 목록/덤프](#) 메뉴에서 스레드 목록 중에 [WhaTap-ActiveStackDump](#) 스레드의 **CPU Time**을 확인하면 오버헤드를 판단할 수 있습니다.

### 액티브 스택 예시

```
JAVA
java.lang.StringBuffer.append(StringBuffer.java:309)
java.util.regex.Matcher.appendReplacement(Matcher.java:839)
java.util.regex.Matcher.replaceAll(Matcher.java:906)
java.lang.String.replaceAll(String.java:2162)
core.log.triggers.TriggerRegister.changeNotify(TriggerRegister.java:114)
core.log.aop.handler.DaoInfo.log(DaoInfo.java:141)
core.log.aop.handler.DaoInfo.doAround(DaoInfo.java:102)
core.log.aop.reflection.profiler.AroundProfiler.invoke(AroundProfiler.java:19)
com.sun.proxy.$Proxy39.getUpdateCount(Unknown Source)
org.apache.ibatis.executor.resultset.DefaultResultSetHandler.getNextResultSet(DefaultResultSetHandler.java:256)
org.apache.ibatis.executor.resultset.DefaultResultSetHandler.handleResultSets(DefaultResultSetHandler.java:193)
org.apache.ibatis.executor.statement.PreparedStatementHandler.query(PreparedStatementHandler.java:64)
***
sun.reflect.GeneratedMethodAccessor140.invoke(Unknown Source)
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
java.lang.reflect.Method.invoke(Method.java:606)
org.springframework.web.method.support.InvocableHandlerMethod.doInvoke(InvocableHandlerMethod.java:221)
org.springframework.web.method.support.InvocableHandlerMethod.invokeForRequest(InvocableHandlerMethod.java:136)
org.springframework.web.servlet.mvc.method.annotation.ServletInvocableHandlerMethod.invokeAndHandle(ServletInvocableHandlerMethod.java:114)
org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter.invokeHandlerMethod(RequestMappingHandlerAdapter.java:827)
```

### 최적화된 데이터 수집

- 트랜잭션을 수행 중인 스레드에 대해서만 스택을 덤프합니다.
- 액티브 스택 덤프 시간 간격을 조정할 수 있습니다.
 

```
active_stack_second=10
```
- 액티브 스택의 최대 라인에 제한되어 있습니다. Top 라인에서부터 기본 50라인을 수집합니다.
 

```
trace_active_callstack_depth=50
```
- 액티브 스택의 각 라인은 해시 처리되어 수집됩니다. text는 한 번만 수집됩니다.
- 한 타임에 수집되는 최대 액티브 스택 개수도 제한되어 있습니다.
 

```
active_stack_count=100
```

## Background Thread에 대한 액티브 스택

기본적으로 액티브 스택은 트랜잭션이 수행되고 있는 스레드의 스택을 말합니다. 하지만 일부 백그라운드 스레드에 대해서도 스택을 분석할 필요가 있을 수 있습니다. 이때 옵션을 통해서 백그라운드 스레드에 대한 액티브 스택을 확보할 수 있습니다. 1.6.2 버전 이후부터 가능합니다.



- `async_stack_enabled` 의 값을 `true` 로 설정하면 활성화됩니다.

```
async_stack_enabled=false
```

- 스택 덤프 간격은 포그라운드 액티브 스택 설정에 따라갑니다.

```
active_stack_second=10
```

- 대상 스레드 이름을 지정할 때는 `*` 를 사용하여 문자열 패턴을 지정합니다.

```
async_thread_match=http*,abc*
```

스레드 이름으로 스택 덤프 대상을 식별합니다. `*` 를 사용하여 match를 여러 개 지정할 수 있습니다.

- 스택의 Top 메소드가 `async_thread_parking` 에 등록된 클래스 / 메소드일 때 스레드가 파킹 상태에 있다고 판단하고 덤프를 생성하지 않습니다.

```
async_thread_parking_class=sun.misc.Unsafe
async_thread_parking_method=park
```

## 참고 자료

- [액티브 트랜잭션](#)
- [장애를 가장 빠르게 알아내는 액티브 트랜잭션](#)

# 멀티 트랜잭션

멀티 트랜잭션은 다른 에이전트나 프로젝트와 연관된 트랜잭션을 의미합니다. 와탭 프로젝트에 등록된 애플리케이션 서비스 간의 호출을 추적하는 것이 멀티 트랜잭션 추적입니다.

① Java 에이전트는 세 개의 HTTP 헤더 키값( `x-wtap-po` , `x-wtap-mst` , `x-wtap-sp1` )으로 멀티 트랜잭션을 추적합니다. 게이트웨이를 통과하는 HTTP 트랜잭션이 연계 추적이 안 된다면 HTTP 헤더 조건을 확인하세요.

① **멀티 트랜잭션 활성화**  
 멀티 트랜잭션을 추적하려면 **관리 > 에이전트 설정** 메뉴에서 `mtrace_enabled` 옵션을 `true` 로 설정하세요. 에이전트 설정에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

## 멀티 트랜잭션 추적

홈 화면 > 프로젝트 선택 > 분석 > 히트맵 메뉴에서 차트 영역을 드래그하세요. 드래그한 차트 영역의 트랜잭션 정보가 다음과 같이 하단 TX 트레이스 목록에 나타납니다. 목록에서 멀티 트랜잭션을 포함한 트랜잭션은 **M** 아이콘이 표시됩니다.

시작 시간	경과 시간↑	트랜잭션	HTTP 호출 ...	SQL 건수	SQL 시간	SQL 패치 건...	DB 연결 시간	에러 클래스	에러 메시지
2023/10/06 09:54:50.327	40,072	/product/delete/unit/seoul	13,359	3	1,554	401	3		
2023/10/06 09:54:42.319	40,212	/product/kill/division/jeju	12,512	8	883	93	259		
2023/10/06 09:56:10.918	40,275	/account/read/unit/daejun	17,686	2	86	421	40		
2023/10/06 09:54:39.622	40,338	/account/kill/division/seoul	18,513	3	425	1,231	160		
2023/10/06 09:57:15.090	40,481	/account/remove/division/pusan	4,609	4	35,854	1,693	4	SLOW_SQL	SLOW_SQL
2023/10/06 09:54:42.941	40,554	/product/remove/employee/daegu	12,413	8	444	713	175		
2023/10/06 09:59:29.442	40,649	/account/delete/dept/daegu	20,626	4	53	426	32		
2023/10/06 10:02:54.198	40,763	/edu/delete/unit/kwangju	15,399	8	156	1,036	10		
2023/10/06 10:02:52.457	41,140	/account/create/employee/kwangju	15,916	4	102	1,483	10		
2023/10/06 10:02:52.815	41,179	/order/remove/division/jeju	15,472	6	323	276	9		
2023/10/06 09:58:59.804	41,278	/sales/read/dept/seoul	18,451	1	0	0	2		
2023/10/06 09:58:38.422	41,664	/account/read/unit/daegu	13,241	7	7,308	1,817	12		
2023/10/06 09:56:02.316	41,820	/account/remove/dept/pusan	18,508	1	16	631	3		
2023/10/06 09:59:03.822	41,949	/account/kill/employee/kwangju	19,086	2	56	0	2		
2023/10/06 09:54:53.750	41,953	/account/pickup/employee/kwangju	15,812	3	105	865	149		

해당 트레이스를 선택하면 **트랜잭션 정보** 창이 나타납니다. 멀티 트랜잭션이 추적되는 경우 **레코드 요약** 탭에서 **멀티 트랜잭션 ID** 값을 확인할 수 있습니다. 트랜잭션에서 외부 호출을 하는 경우에도 동일한 **멀티 트랜잭션 ID**가 생성됩니다. 서비스별로 프로젝트가 분리되어 있더라도 처음 발급한 **멀티 트랜잭션 ID**를 통해 애플리케이션 간의 모든 트랜잭션을 확인할 수 있습니다. **트랜잭션 정보** 창을 활용한 TX 트레이스 상세 분석에 관한 자세한 내용은 [다음 문서](#)를 참조하세요.

트랜잭션 정보

레코드 요약
테이블 뷰
트리 뷰
멀티 트랜잭션
액티브 스택
메소드 요약
SQL 요약
HTTP Call 요약
트랜잭션 로그

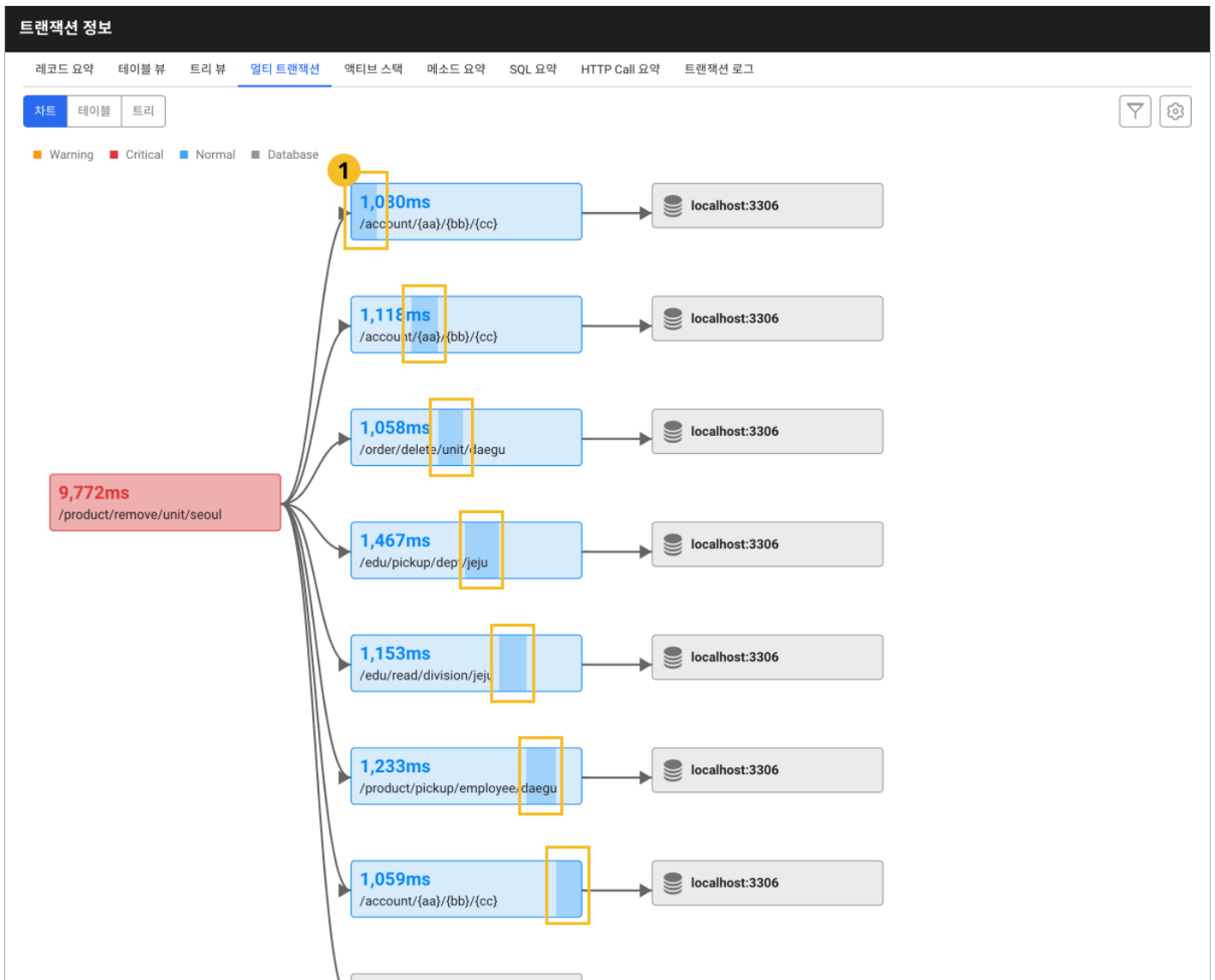
/account/read/unit/daejun
🔍

<b>에이전트명 (oname)</b> : demo-8101	<b>에이전트 ID (oid)</b> : -877561626
<b>에이전트 그룹명</b> : demo-okind-1	<b>에이전트 그룹 ID</b> : -628198688
<b>에이전트 서버명</b> : node-1	<b>에이전트 서버 ID</b> : 334634079
<b>프로젝트 코드</b> : 5490	

<b>트랜잭션 ID</b> : -726761366435603064	<b>멀티 트랜잭션 ID</b> : 4364467734006050293
<b>시작 시간</b> : 23/10/06 07:12:40.183	<b>종료 시간</b> : 23/10/06 07:12:57.946
<b>경과 시간</b> : 17,763ms	<b>HTTP 메소드</b> : GET
<b>HTTP 호출 시간</b> : 16,766ms	<b>HTTP 호출 건수</b> : 14
<b>DB 연결 시간</b> : 9ms	<b>SQL 시간</b> : 968ms
<b>SQL 건수</b> : 8	<b>SQL 패치 시간</b> : 13ms
<b>SQL 패치 건수</b> : 1,082	

멀티 트랜잭션 ID를 선택하면 멀티 트랜잭션 탭으로 이동합니다. 시스템 내 또는 시스템 간에 발생하는 다양한 호출 관계를 한 눈에 파악하고 어느 부분에서 문제가 발생했는지 식별하여 개선할 수 있도록 트랜잭션과 트레이스 정보를 제공합니다.

# 차트



멀티 트랜잭션 추적 메뉴의 기본 화면은 차트입니다. 차트는 각 트랜잭션의 호출 관계를 빠르고 명확하게 사용자에게 제공합니다. 동일한 멀티 트랜잭션 ID를 갖는 트랜잭션 서비스들의 개별 수행 시간을 확인할 수 있습니다. 트랜잭션 노드의 배경색으로 표현되어 있는 소요 시간(1 타임바)를 통해 트랜잭션 간 호출 관계를 확인할 수 있습니다. 차트에서는 마우스를 이용해 원하는 위치로 이동하거나 스크롤을 통해서 확대, 축소할 수 있습니다.

- **프로젝트 선택:** 차트에 표시될 프로젝트를 선택하거나 해제할 수 있습니다.

- **차트 뷰 설정:** 차트에 표시할 요소를 표시하거나 숨길 수 있습니다.
  - **타임바:** 각 노드에 소요 시간(타임바)를 표시하거나 숨길 수 있습니다.
  - **애플리케이션명:** 각 노드에 에이전트 이름(oname)을 표시하거나 숨길 수 있습니다.
  - **프로젝트:** 각 노드에 프로젝트 이름을 표시하거나 숨길 수 있습니다.
  - **데이터베이스 / 외부 호출 / 내부 호출:** 해당 트랜잭션에서 발생한 다른 데이터베이스 커넥션 요청이나 HTTP Call의 정보 또한 차트의 노드로 확인할 수 있습니다.

트랜잭션 노드를 선택하면 다음과 같이 **트랜잭션 정보** 창이 추가로 나타납니다. 트랜잭션 트레이스를 통해 해당 트랜잭션의 상세 내역을 확인할 수 있습니다. **트랜잭션 정보** 창을 활용한 트랜잭션 트레이스 상세 분석에 관한 자세한 내용은 [다음 문서](#)를 참조하세요

트랜잭션 정보 /account/{aa}/{bb}/{cc}

레코드 요약 테이블 뷰 트리 뷰 메소드 요약 SQL 요약 HTTP Call 요약 트랜잭션 로그

차트 테이블

/account/{aa}/{bb} /product/remove /account/{aa}/{bb} /order/delete/uni /edu/pickup/dept /edu/read/divisio /product/pickup/ /account/{aa}/{bb}

시작: 08/22 06:31:20.997 종료: 08/22 06:31:22.077 경과: 1,080ms 에이전트 명 (oname): dev2344677-8090

200ms 400ms 600ms 800ms 1s

DB Connection SQL

컬럼 선택 TXT SQL

No	시간	캡	경과	내용
1	06:31:20.997			시작 2023-08-22 06:31:20.997
2	06:31:20.997	0	1	MYSQL jdbc:mysql://localhost:3306,localhost
3	06:31:20.998	1	0	SELECT DISTINCT pp.LastName, pp. FirstName FROM Person.Person pp JOIN Human Resources.Employee e...
4	06:31:21.002	4	4	[결과 건수] 245
5	06:31:21.002	0	2	MYSQL jdbc:mysql://localhost:3306,localhost
6	06:31:21.004	2	0	SELECT first, COUNT(ngram) ngram_count FROM...
7	06:31:21.008	4	4	[결과 건수] 412
8	06:31:21.008	0	1	MYSQL jdbc:mysql://localhost:3306,localhost
9	06:31:21.009	1	0	SELECT ProductID, SpecialOfferID, AVG(UnitPrice) AS [Average Price], SUM(LineTotal) AS SubTotal...
10	06:31:21.013	4	4	[결과 건수] 580
11	06:31:21.013	0	1	MYSQL jdbc:mysql://localhost:3306,localhost
12	06:31:21.014	1	0	SELECT p.Name AS ProductName, NonDiscountSales = (OrderQty * UnitPrice), Discounts = ((OrderQty * UnitPr...
13	06:31:21.018	4	4	[결과 건수] 377
14	06:31:21.018	0	2	MYSQL jdbc:mysql://localhost:3306,localhost
15	06:31:21.020	2	0	SELECT ename, sal FROM emp

## 테이블

테이블 버튼을 선택하면 테이블 형식으로 멀티 트랜잭션 내에 포함된 각 트랜잭션 별 정보를 확인할 수 있습니다.

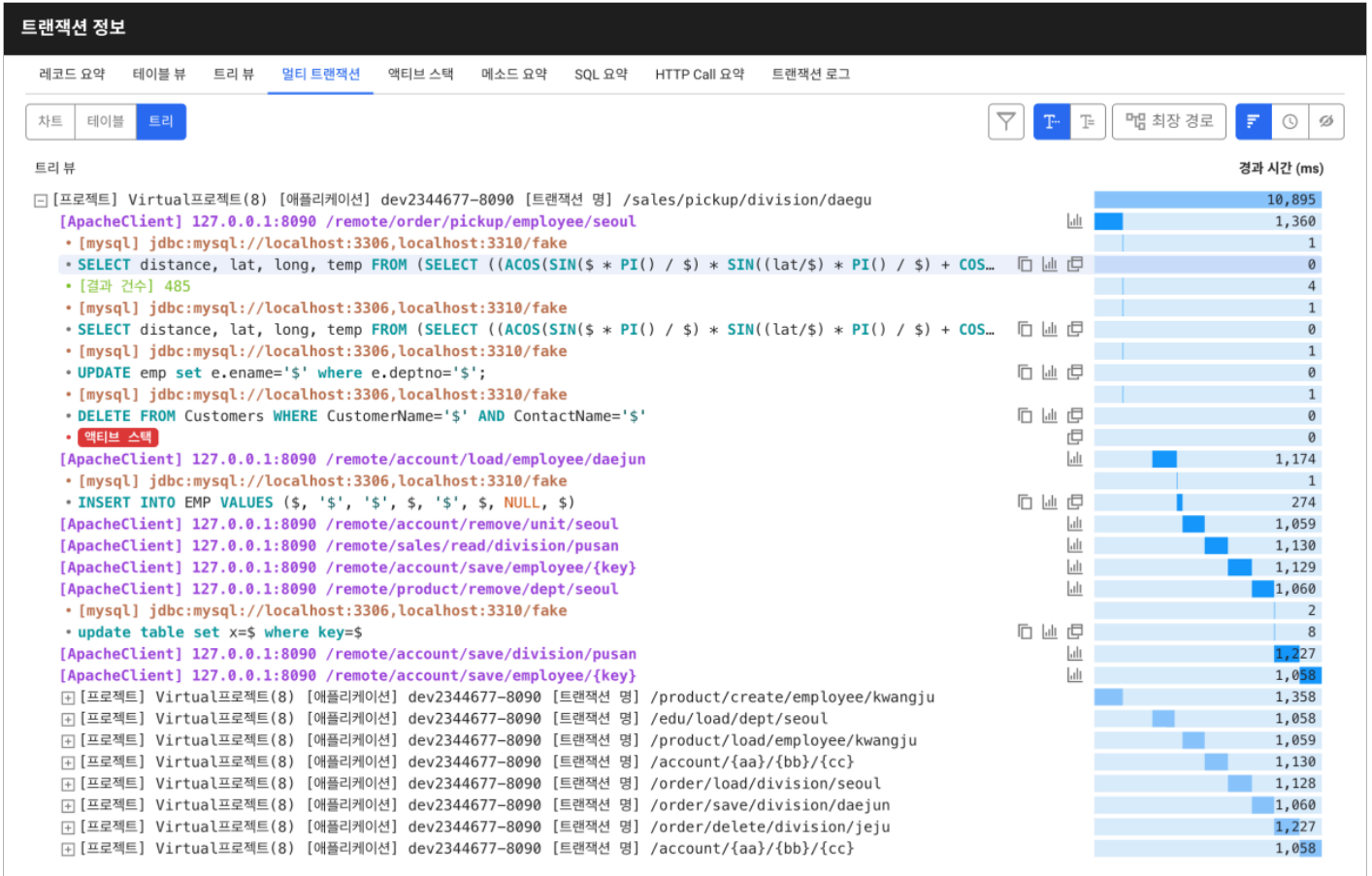
트랜잭션 정보							
레코드 요약	테이블 뷰	트리 뷰	멀티 트랜잭션	메소드 요약	SQL 요약	HTTP Call 요약	트랜잭션 로그
차트	테이블	트리					컬럼 설정
트랜잭션	경과 시간 (ms)	시작 시간	에이전트 명	프로젝트	Caller	트랜잭션 ID	종료
/account/load/employee/seoul	5,265	2023-10-10 06:20:16	demo-8105	Java APM Demo 5490	8782414760663308...	8782414760663308...	2023-10-
/account/remove/dept/jeju	1,092	2023-10-10 06:20:16	demo-8105	Java APM Demo 5490	8782414760663308...	-3735721994692158...	2023-10-
/account/kill/employee/daegu	1,221	2023-10-10 06:20:17	demo-8105	Java APM Demo 5490	8782414760663308...	-2099869066688328...	2023-10-
/order/kill/employee/jeju	1,311	2023-10-10 06:20:19	demo-8105	Java APM Demo 5490	8782414760663308...	480511625630425675	2023-10-
/order/pickup/employee/pusan	1,204	2023-10-10 06:20:20	demo-8105	Java APM Demo 5490	8782414760663308...	68039636449071290	2023-10-

- **프로젝트 선택**: 차트에 표시될 프로젝트를 선택하거나 해제할 수 있습니다.
- **컬럼 설정**: 테이블 내 컬럼을 편집할 수 있습니다.

각 트랜잭션 항목을 선택하면 **차트** 노드 선택과 마찬가지로 선택한 트랜잭션에 대한 **트랜잭션 정보** 창이 나타납니다. 트랜잭션 트레이스를 통해 해당 트랜잭션의 상세 내역을 확인할 수 있습니다. **트랜잭션 정보** 창을 활용한 트랜잭션 트레이스 상세 분석에 관한 자세한 내용은 [다음 문서](#)를 참조하세요.

## 트리


**트리** 버튼을 선택하면 각 트랜잭션과 그에 속해 있는 트레이스의 세부 정보를 확인할 수 있습니다. 전체 트랜잭션 소요 시간 내의 각 하위 트랜잭션이나 트레이스의 시작 및 소요 시간을 시각화해 트랜잭션 호출 관계를 트리 형식으로 제공합니다.



- 🔍 프로젝트 선택: 차트에 표시될 프로젝트를 선택하거나 해제할 수 있습니다.
- 📄 한 줄 보기: 각 구간 별 수행 정보에 표시된 텍스트를 한 줄로 표시해 트리 형식을 간격하게 정리할 수 있습니다.
- 📄 여러 줄 보기: 각 구간 별 수행 정보에 표시된 텍스트를 줄바꿈해 모두 표시합니다.
- 📏 최장 경로: 가장 긴 경로로 이동할 수 있습니다.
- 📊 시간바 표시: 경과 시간을 막대 형식의 차트로 표시합니다.
- 🕒 시간 표시: 각 구간별 타임 스탬프, 갭, 경과 시간을 텍스트 형식으로 표시합니다.
  - 8초 이상: **초과 지연** 상태로 **빨간색**으로 표현합니다.
  - 3초 이상 8초 미만: **지연** 상태로 **주황색**으로 표현합니다.
  - 3초 미만: **정상** 상태로 **파란색**으로 표현합니다.

ⓘ 시작 및 소요 시간의 경우 트랜잭션 호출 환경에 따라 발생하는 시차를 상위 트랜잭션 내 트레이스와 매핑을 통해 보정하여 표현하기 때문에 실제 수집된 시간 데이터와 차이가 발생할 수 있습니다.



-  시간 숨기기: 시간 정보를 숨깁니다.

ⓘ 해당 트레이스의  버튼 또는  버튼을 선택하면 HTTP 호출 통계, 액티브 스택 등의 요약 창을 확인할 수 있습니다.

# 트랜잭션 엔드포인트 설정

트랜잭션 엔드포인트는 트랜잭션의 시작 메소드입니다. HTTP 트랜잭션의 경우에는 `HttpServlet.service()` 혹은 `Filter.doFilter()` 가 트랜잭션의 시작점이고 이곳을 **트랜잭션 엔드 포인트**라고 부릅니다.

## NON HTTP 추적

트랜잭션 엔드포인트로 지정된 메소드가 시작해서 종료될 때까지의 성능을 트랜잭션 성능이라고 합니다. Non http 트랜잭션을 추적하기 위해서는 **엔드포인트**를 지정해야 합니다.

### ! 엔드포인트 찾기

1. 트랜잭션이 호출될 것으로 추정되는 모든 메소드에 대해 트레이스 추적을 설정하세요.
2. 다시 시작 후 트랜잭션을 발생시켜 모니터링하세요.
3. `back stack` 옵션을 켜서 진입 메소드를 정확히 파악하세요.

## 대상 선정

### 1. 메소드 트레이스 설정

메소드 트레이스를 설정하세요. 확실하게 트랜잭션에서 호출되는 클래스를 지정하세요. DB를 사용하는 프로그램에서는 **JDBC 드라이버**가 유용합니다.

```
whatap.conf
```

```
hook_method_patterns=jdbc.*.*
hook_method_access_public_enabled=true
hook_method_access_protected_enabled=true
hook_method_access_none_enabled=true
```

### 2. 트랜잭션 시작 옵션 설정

메소드가 호출되면 트랜잭션을 시작하세요. 트랜잭션 시작 옵션과 트랜잭션 시작 시 스택을 덤프하는 옵션을 켜세요.

```
whatap.conf
```

```
trace_auto_transaction_enabled=true
trace_auto_transaction_backstack_enabled=true
```

### 3. 재시작 후 트레이스 분석

다시 시작하세요. 이후 서비스를 호출하면 트랜잭션이 추적되는 것을 볼 수 있습니다.

트랜잭션을 조회하면 `jdbc.*`로 시작하는 모든 클래스의 메소드가 이 트랜잭션으로 나타나는 것을 알 수 있습니다. 트랜잭션 트레이스를 조회하면 **TRANSACTION BACKSTACK**라는 메시지 스템을 확인할 수 있습니다.

#### TRANSACTION BACKSTACK

```
jdbc.FakePreparedStatement.executeQuery(FakePreparedStatement.java),
com.virtual.dao.SelectDAO.execute2(SelectDAO.java:29),
com.virtual.web.SimulaNonHttp.execute(SimulaNonHttp.java:147),
com.virtual.web.SimulaNonHttp.process(SimulaNonHttp.java:76),
com.virtual.web.SimulaNonHttp.run(SimulaNonHttp.java:100)
```

### 4. 스택 내용 확인

스택 내용을 확인하면 어떤 메소드로부터 출발하고 있는지 추정할 수 있습니다.

#### Example

```
com.virtual.web.SimulaNonHttp.execute(SimulaNonHttp.java:147),
com.virtual.web.SimulaNonHttp.process(SimulaNonHttp.java:76),
com.virtual.web.SimulaNonHttp.run(SimulaNonHttp.java:100)
```

위 3개의 메소드 중에 하나를 트랜잭션 시작점으로 판단할 수 있습니다. 이 상황에서는 역 컴파일을 수행해서 적절한 **트랜잭션 엔드포인트**를 결정해야 합니다.

로직을 보면 `SimulaNonHttp.run` 내에서 `while()`가 돌면서 `SimulaNonHttp.process()`을 호출하고 `SimulaNonHttp.execute()`가 수행됩니다. `process()`가 적당하다는 것을 알 수 있습니다. 이 부분은 소스를 보고 판단해야 합니다.

- ✔ **엔드포인트의 가장 중요한 기준은 종료되어야 한다는 것입니다.** 정상적인 상황에서 지연되지 않고 곧바로 종료되어야 성능적인 판단을 할 수 있습니다.

## 트랜잭션 엔드포인트 지정

1. 트랜잭션 시작 지점을 다음과 같이 설정하세요.

```
whatap.conf
```

```
hook_service_patterns=com.virtual.web.SimulaNonHttp.process
```

2. 애플리케이션을 다시 시작하세요.

`process()` 메소드가 새로운 트랜잭션의 **엔드포인트**가 됩니다.

ⓘ `hook_service_patterns` 옵션은 와일드카드 문자를 사용할 수 없으며, 2개 이상의 값을 설정하려면 쉼표(,)를 구분자로 이용하세요.

## 트랜잭션 이름 정의

일반적으로 메소드 명칭으로 트랜잭션을 구분할 수 있습니다.

```
whatap.conf
```

```
service_name_mode=[full,class,method,string,arg]
service_name_index=0
```

### • service\_name\_mode

`full`, `class`, `method`, `string`, `arg` 5가지 옵션을 지정할 수 있습니다.

- `full`: Full Class 이름을 서비스 명으로 사용합니다.
- `class`: Class 이름을 서비스 명으로 사용합니다.
- `method`: Method 이름을 서비스명으로 사용합니다.
- `string`: 문자열 중에서 첫 번째 파라미터를 서비스 명으로 사용합니다.
- `arg`: 파라미터 중에서 `service_name_index` 옵션에 지정한 인덱스에 파라미터를 서비스 명으로 사용합니다.

## 플러그인 사용

⚠ 플러그인은 충분히 이해한 경우에만 사용하길 권장합니다.

WHATAP\_HOME 경로 아래에 `plugin` 폴더를 만드세요. 그리고 vi를 통해 `AppServiceStart.x` 파일을 만드세요.

```

avaproc.bat          watch_tomcat.bat
avaproc.sh           watch_tomcat.sh
lib1                 whatap-virtual-0.6.5.jar
logs                 whatap.agent.proxy-0.6.5.jar
paramkey.txt         whatap.agent.tracer-0.6.5.jar
esmon.sh             whatap.agent.watcher-0.6.5.jar
scripts             whatap.conf
.sh
aul@phost:/app/stage/whatap/agent$ mkdir plugin
aul@phost:/app/stage/whatap/agent$ cd plugin
aul@phost:/app/stage/whatap/agent/plugin$ vi AppServiceStart.x

```

`println("test");` 라고 타이핑하고 저장하면 화면에 "test"라는 문자열이 출력되는 것을 확인할 수 있습니다. 확인 후 파라미터에서 정보를 추출하세요.

예시에서는 파라미터가 HashMap에 전달되고 거기에 url 파라미터가 전달되고 있습니다.

### JAVA

```

Object url=((java.util.HashMap)$point.getArgs()[0]).get("url");
$cctx.service((String)url);
//println("url="+url);

```

이렇게 플러그인을 만들면 트랜잭션 이름이 변경됩니다.

# 트랜잭션 스텝 수집 방식

애플리케이션 모니터링 에이전트는 선형 수집 방식과 환형 수집 방식을 제공합니다.

## 선형 수집

개별 트랜잭션 트레이스는 무한정 스텝을 수집할 수 없습니다. 트랜잭션별로 길이가 제한된 버퍼에 트레이스 스텝을 저장합니다. 각 트랜잭션은 최대 수집할 수 있는 스텝의 수가 옵션으로 지정되어 있습니다.

- **profile\_step\_max\_count**

기본값 1024

최대로 수집 가능한 스텝 수입니다.

- **profile\_step\_normal\_count**

기본값 800

일반적으로 아무런 제약없이 수집되는 스텝 수입니다.

- **profile\_step\_heavy\_count**

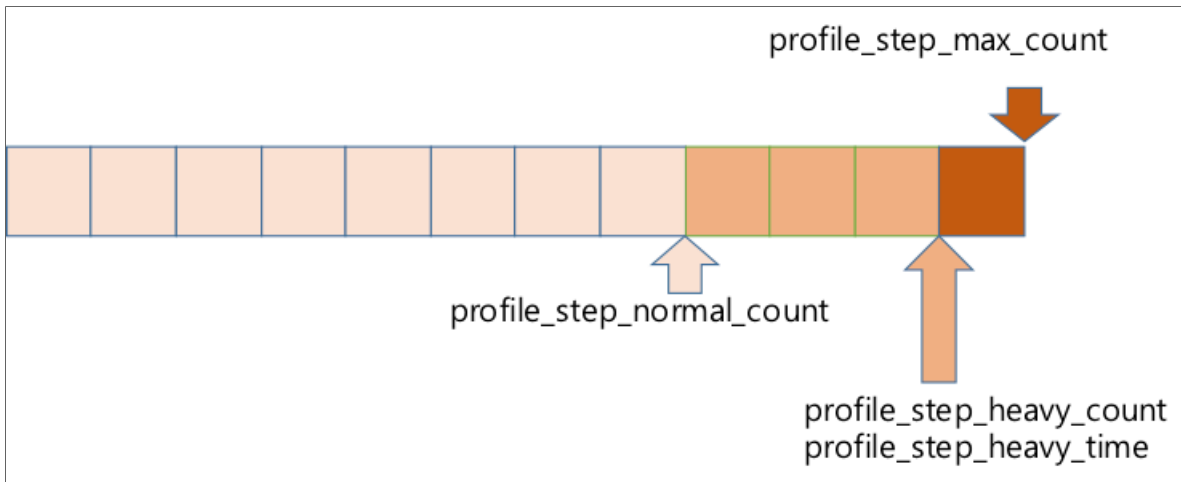
기본값 1000

normal count를 초과한 경우 스텝은 응답시간이 느린 스텝과 액티브 스택 스텝만 수집합니다.

- **profile\_step\_heavy\_time**

기본값 100

heavy count 이내에서의 수집되는 스텝의 기준 시간은 profile\_step\_heavy\_time 입니다.

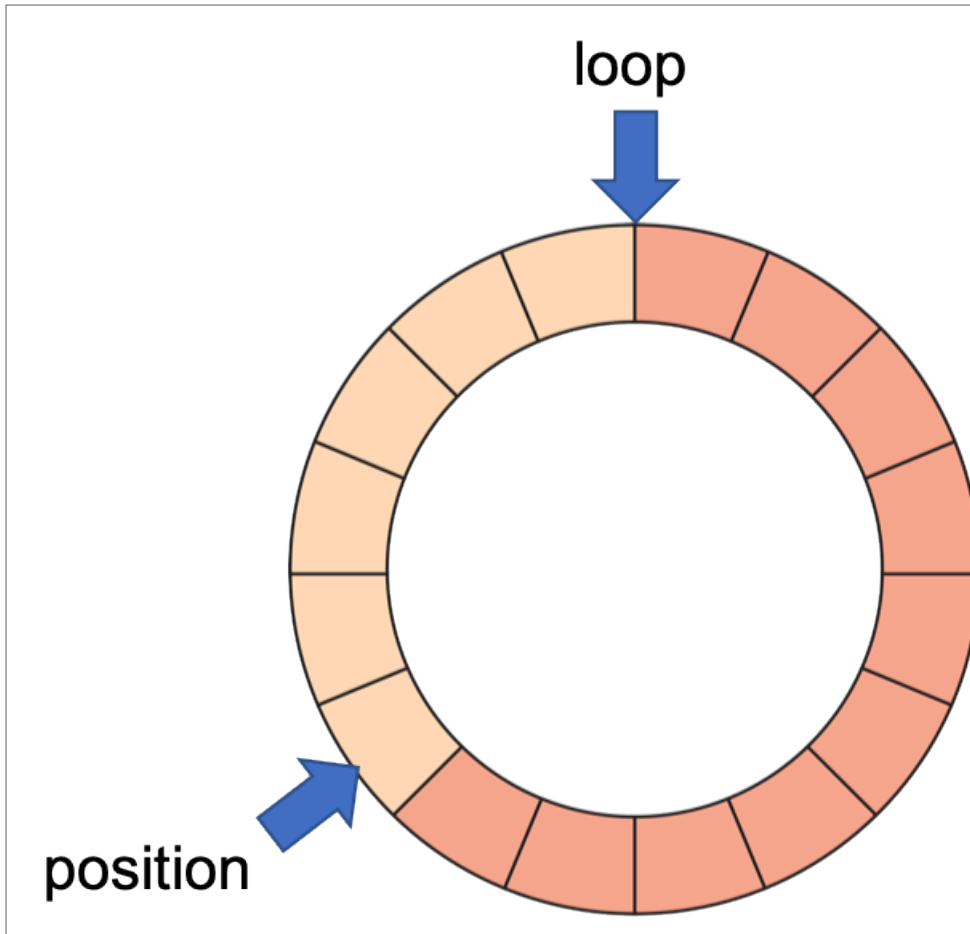


수집되는 트레이스의 스텝 수가 heavy count를 초과하는 경우에는 액티브 스택만이 수집됩니다. 이 경우에도 최대 스텝 수는 `profile_step_max_count` 를 넘지 않습니다.

## 환형 수집

선형 수집은 트레이스에서 앞부분을 수집하고 버퍼 사이즈를 넘으면 나중 내용을 버립니다. 반면 환형 수집은 앞부분의 스텝을 버리는 방식입니다.

```
circular_profile_enabled=true
```



버퍼 사이즈는 `profile_step_max_count` 에 설정합니다.

```
profile_step_max_count=1024
```



# 토폴로지

## 토폴로지란?

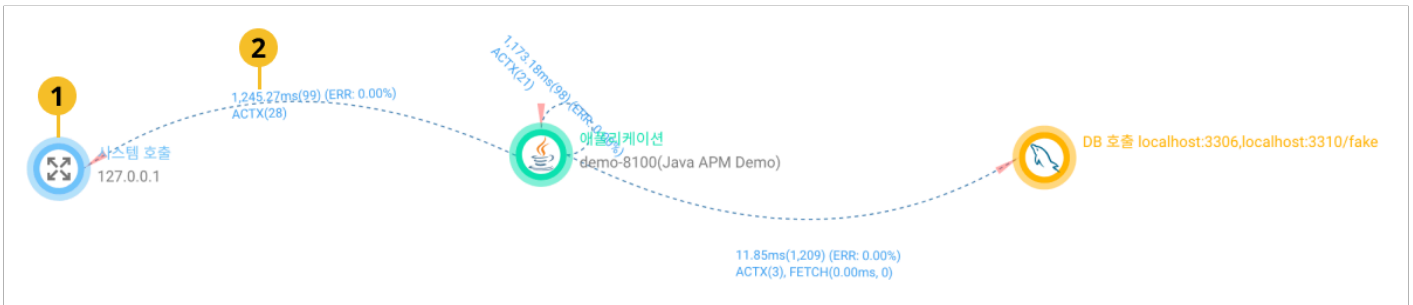
모니터링 대상 서버로부터 수집하는 실시간 정보를 통해 애플리케이션 서버 간의 연관 관계를 용이하게 파악할 수 있습니다.

다음 동영상을 통해 토폴로지에 대해 알아보세요.

❗ 일반적인 토폴로지 정의에서 링크 정보를 가지지 않은 노드를 포함하지 않습니다. 모니터링 관점에서의 토폴로지는 고립된 노드의 정보도 필요합니다.

## 토폴로지 기본 요소

토폴로지 화면에서 표현하는 정보는 노드와 링크, 각각이 포함하는 부가 정보로 구분합니다.



- **1 노드**

프로젝트, 애플리케이션 그룹, 애플리케이션, 데이터베이스, HTTP 호출 대상 외부 호스트, 외부 모듈(애플리케이션 호출자)을 표현합니다.

❗ 노드로부터 호출 정보가 존재하는 경우 노드 외곽에 에러 비중을 표현합니다.

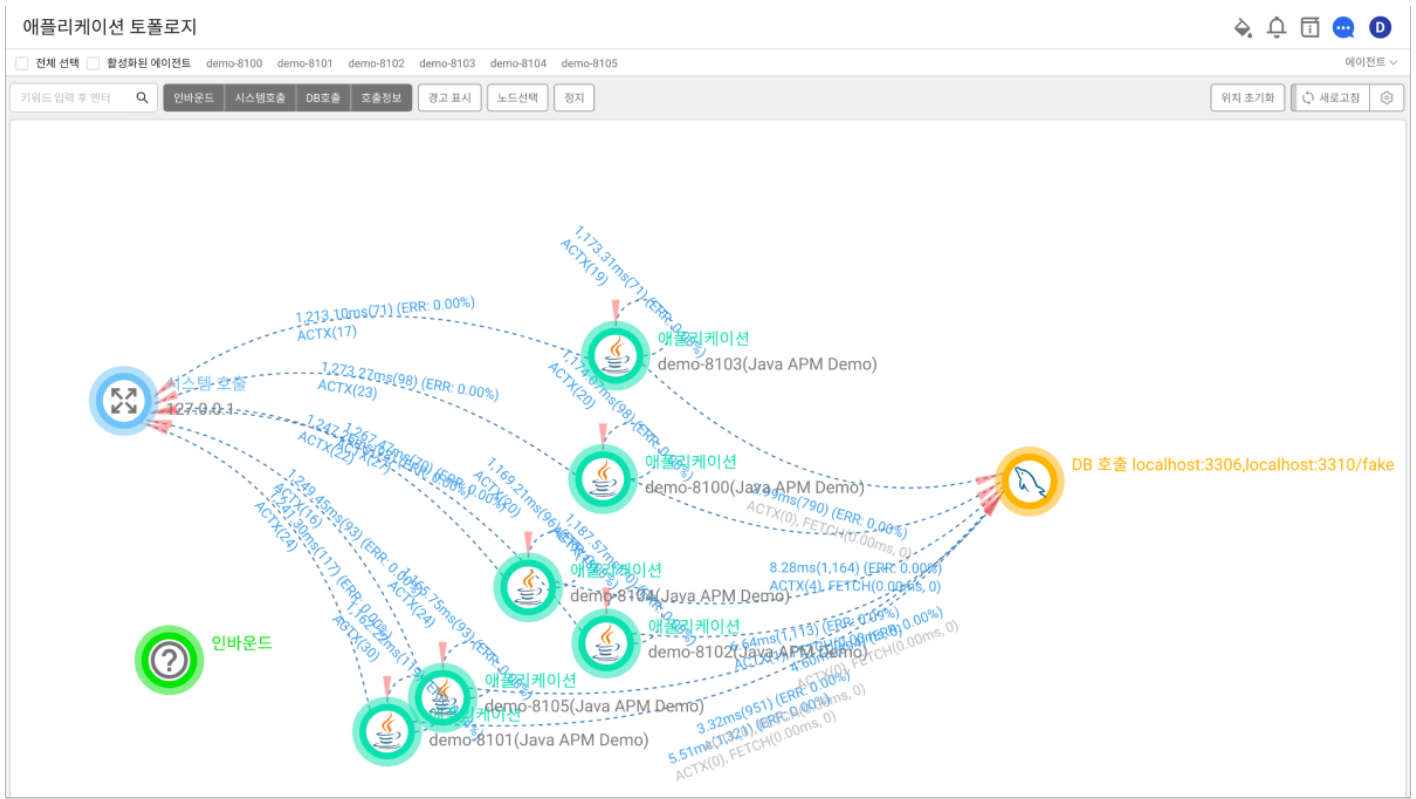
- **2 링크**

애플리케이션 또는 애플리케이션 그룹 간의 호출 정보, 데이터베이스 호출 정보, 외부 호스트 호출 정보, 외부 모듈로부터의 호출 정보를 표현합니다.

- **평균 응답시간**: 5초간 노드 사이에서 트랜잭션 또는 호출 내역의 응답 시간 합계를 총건수로 나눈 정보
- **건수**: 5초간 노드 사이에서 트랜잭션 또는 호출 내역의 (에러 건수를 포함한) 총건수
- **에러**: 5초간 노드 사이에서 트랜잭션 또는 호출 내역 중 총 에러 건수
- **액티브 TX**: 노드 사이에서 트랜잭션 또는 호출 중 현재 진행 상태의 건수
- **패치 시간**: 5초간 DB 호출 내역의 레코드 취득 시간 총합
- **패치 건수**: 5초간 DB 호출 내역의 레코드 취득 건수 총합

# 기본 사용법

토폴로지 메뉴의 기본 동작 방법에 대해 알아보세요.



토폴로지 화면의 기본 동작은 사용자 편의성을 위해 다음과 같은 기능을 제공합니다.

- 자동 갱신: 기본 30초 간격의 자동 갱신을 통해 토폴로지의 변경 사항을 자동으로 업데이트합니다.

ⓘ 갱신 간격을 변경하려면 오른쪽 위에 ⚙️을 선택한 다음 원하는 시간을 입력하세요. 5초에서 5분까지 설정할 수 있습니다. 브라우저에서 새로 고침하면 최신 정보로 업데이트할 수 있습니다.

- 줌인/줌아웃: 마우스 휠을 통해 토폴로지를 확대 또는 축소할 수 있습니다. 노드 수가 많은 경우 원하는 사이로 토폴로지를 표현할 수 있습니다. 줌인/줌아웃 레벨은 내부 설정에 유지됩니다.
- 노드 드래그: 노드를 마우스 왼쪽 버튼으로 클릭한 상태에서 드래그하여 원하는 위치에 고정할 수 있습니다.
- 프로젝트 노드 클릭

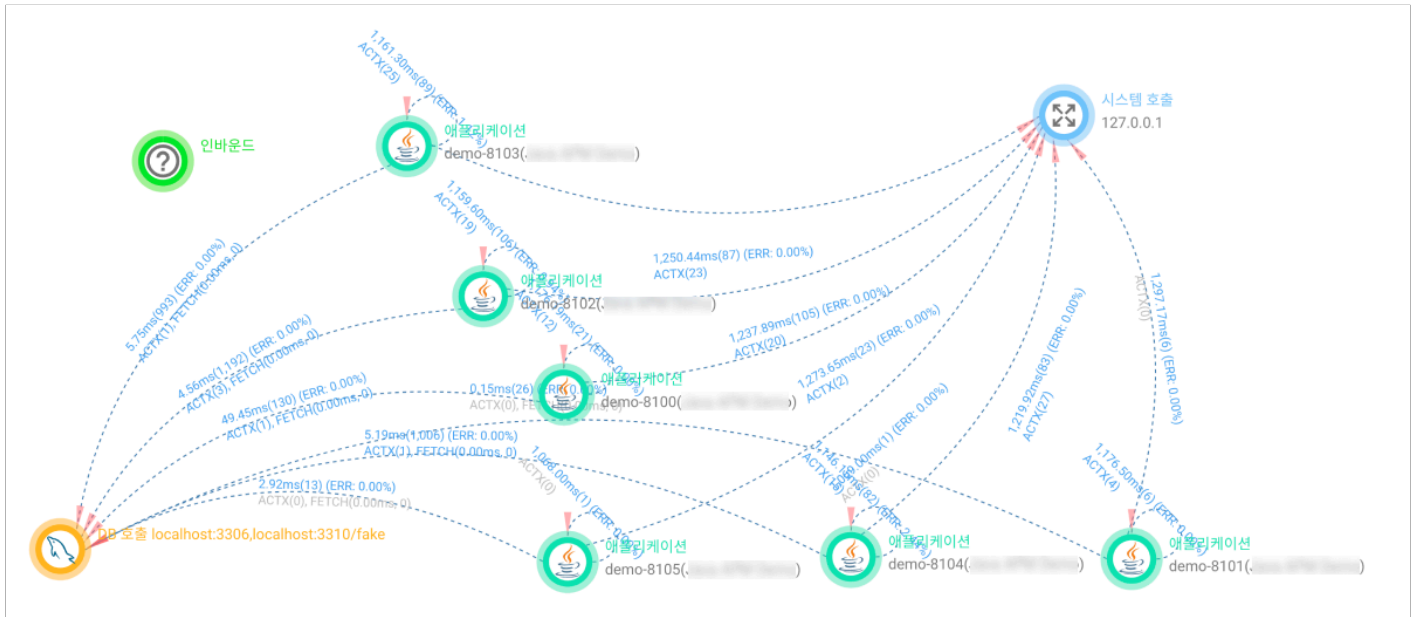
- [대시보드](#) > [애플리케이션 토폴로지](#) 메뉴에서 노드를 클릭하면 해당 프로젝트의 애플리케이션 토폴로지([인스턴스 토폴로지](#))로 이동합니다.
- [홈 화면](#) > [통합 APP.그룹 토폴로지](#) 메뉴에서 노드를 클릭하면 해당 프로젝트의 토폴로지 정보를 취득하고 현재 토폴로지에 표현합니다.
- 하이라이트 필터링: 노드 및 노드와 인정한 링크를 필터링해 하이라이트할 수 있습니다.

# 유형별 토폴로지

유형별 토폴로지에 대해 알아보고 각 토폴로지에서 제공하는 기능을 설명합니다.

## 애플리케이션 토폴로지

홈 화면 > 프로젝트 선택 > 대시보드 > 애플리케이션 토폴로지



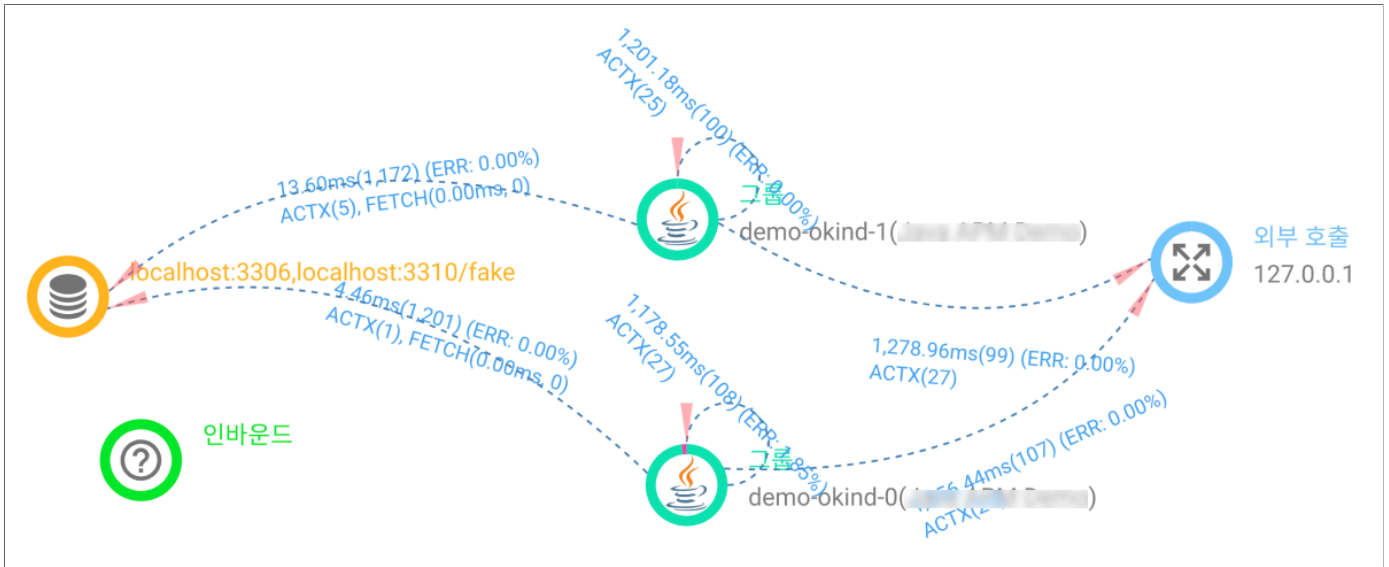
프로젝트 범위에 포함된 모든 애플리케이션의 호출 연관 정보를 표현합니다.

ⓘ 애플리케이션 노드를 더블 클릭하면 해당 애플리케이션의 [인스턴스 토폴로지](#) 화면으로 전환합니다. 선택한 애플리케이션 중심의 토폴로지를 표현합니다.

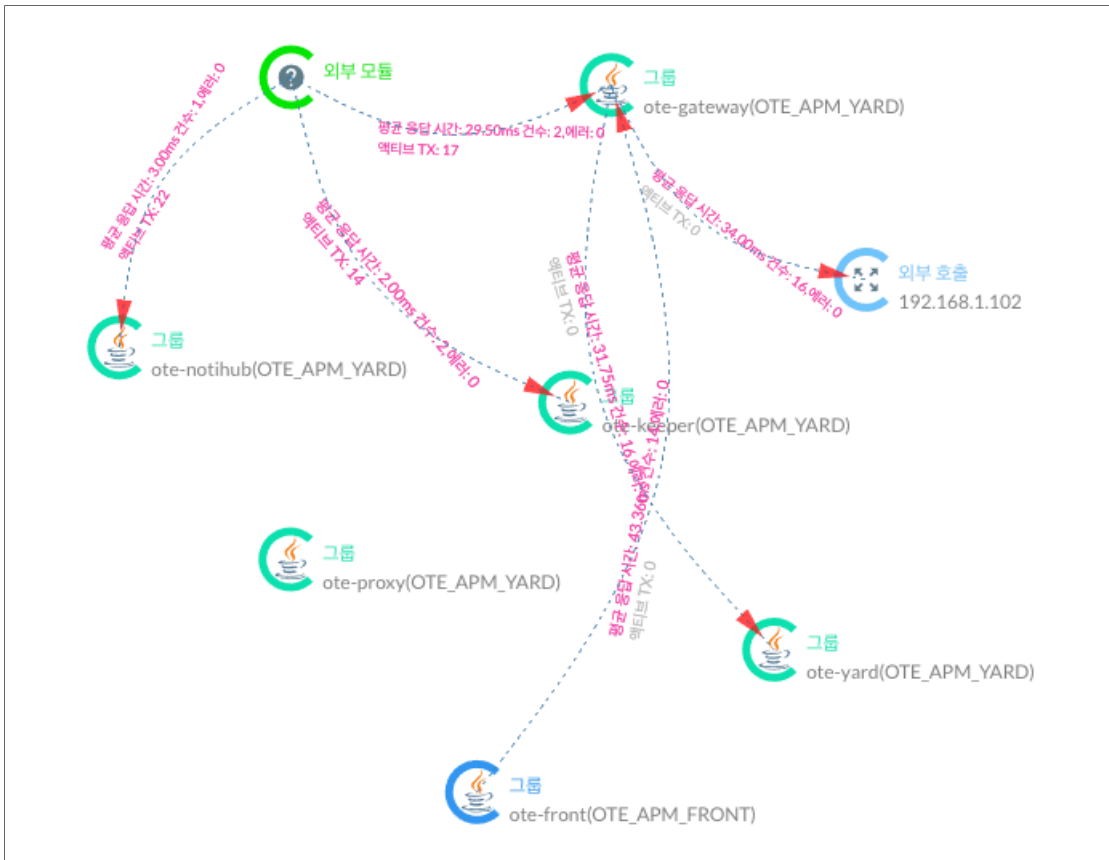
## 그룹 토폴로지

홈 화면 > 프로젝트 선택 > [사이트맵](#) > [그룹 통계 토폴로지](#)

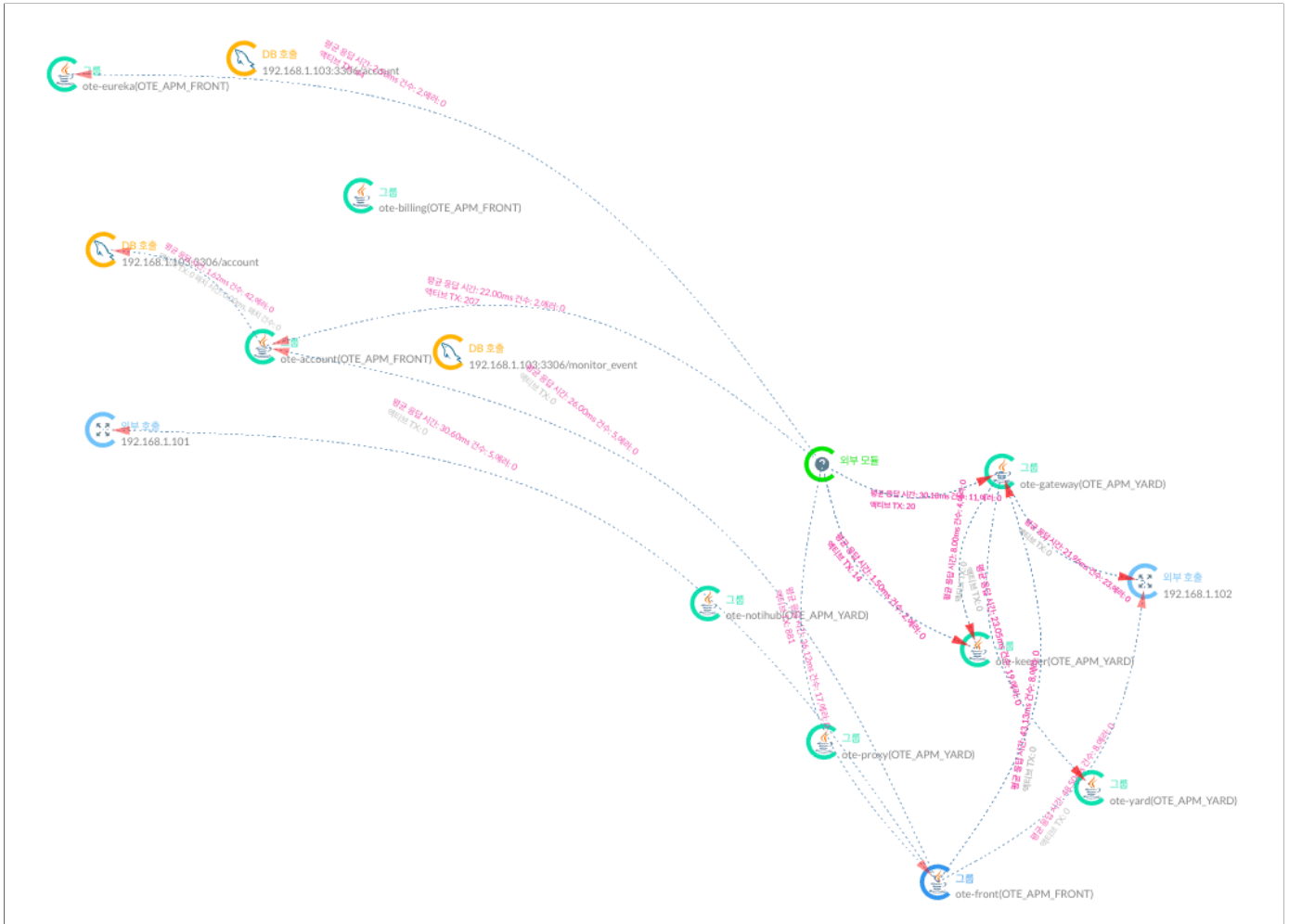
- 데이터베이스 호출 내역을 포함한 토폴로지의 경우 다음과 같이 데이터베이스 노드를 포함합니다.



- 타 프로젝트로부터 호출 내역이 존재하는 경우 타 프로젝트의 그룹 노드를 포함합니다.



- 타 프로젝트 그룹 노드를 클릭하면 타 프로젝트의 토폴로지 정보를 함께 표시합니다.



## 통합 토폴로지

사용자가 복수 프로젝트에 대한 권한을 보유한다면 프로젝트 단위 토폴로지로 전체 상황을 조망하기 어려운 제약이 존재합니다. 이와 같은 경우 복수의 프로젝트를 일괄 선택해 단일 토폴로지로 보기 위한 요구 사항을 충족하기 위한 기능입니다.

1. 와탭 모니터링 서비스 홈 화면에서 **통합 APP.그룹 토폴로지** 메뉴를 선택하세요.
2. 왼쪽 위에 **프로젝트**를 선택하세요.



### 3. 토폴로지에 표현할 그룹 또는 프로젝트를 선택하세요.

토폴로지 화면의 중앙을 기준으로 프로젝트 단위의 클러스터를 형성하고 각 프로젝트에 포함된 그룹을 인접 위치에 군집시켜 표현합니다. 인접한 노드 사이의 간격을 조정하려면 상단의 노드 간격 관련 확대/축소(++ + ■ - -) 버튼을 이용해 노드 간격을 조정할 수 있습니다.

- ❗ • 대규모 환경의 경우 일반적으로 멀티 프로젝트로 구성합니다. 이를 토폴로지로 표현할 경우 개별 노드를 화면 전체에 균등 분포시키기 보다는 프로젝트 단위로 모아서 배치하는 것이 전체 규모를 파악하는데 용이합니다. 따라서 군집 단위를 선택하여 화면에 자동 배치합니다.
- 프로젝트, 데이터베이스, 외부 호스트, 애플리케이션 및 그룹을 호출하는 외부 모듈을 군집 단위로 합니다.
- 클러스터 구성은 프로젝트, 외부 모듈, 외부 호출, DB 호출 단위로 구성합니다.

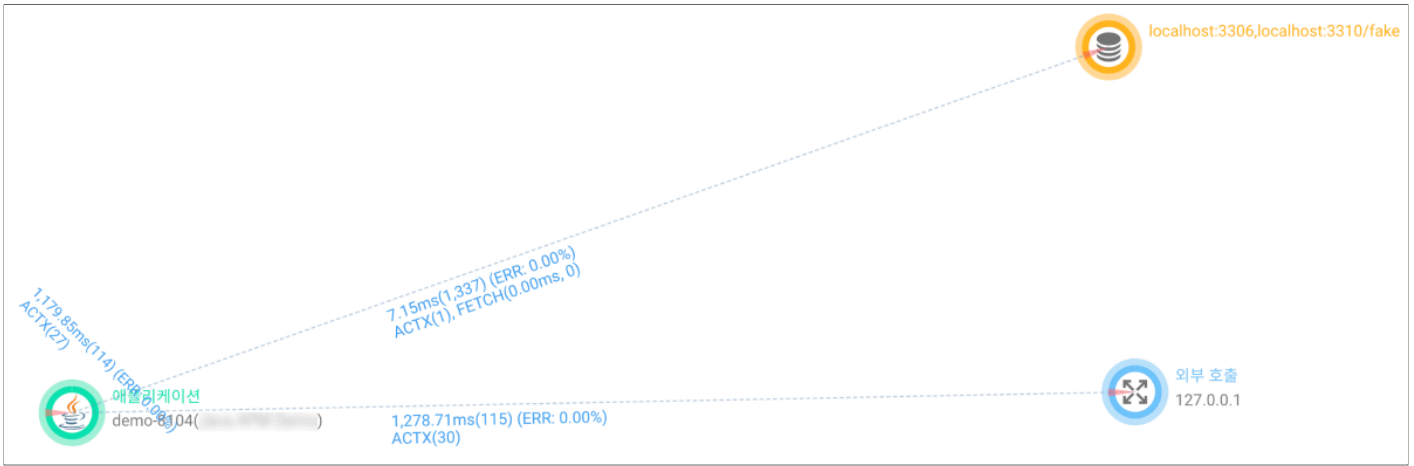
## 인스턴스 토폴로지

홈 화면 > 프로젝트 선택 > 사이트맵 > 인스턴스 토폴로지

**인스턴스 토폴로지**는 애플리케이션 토폴로지와 동일한 데이터를 표현합니다. **인스턴스 토폴로지**는 단일 애플리케이션을 대상으로 연관성을 지닌 애플리케이션, 외부 모듈, DB 및 외부 호출 노드와의 연관성을 표현합니다. 반면, **애플리케이션 토폴로지**는 프로젝트에 포함된 전체 애플리케이션을 대상으로 합니다.

**애플리케이션 토폴로지**에 표현되는 정보가 너무 많아 애플리케이션 단위로 확인할 경우 **인스턴스 토폴로지**를 이용하세요.

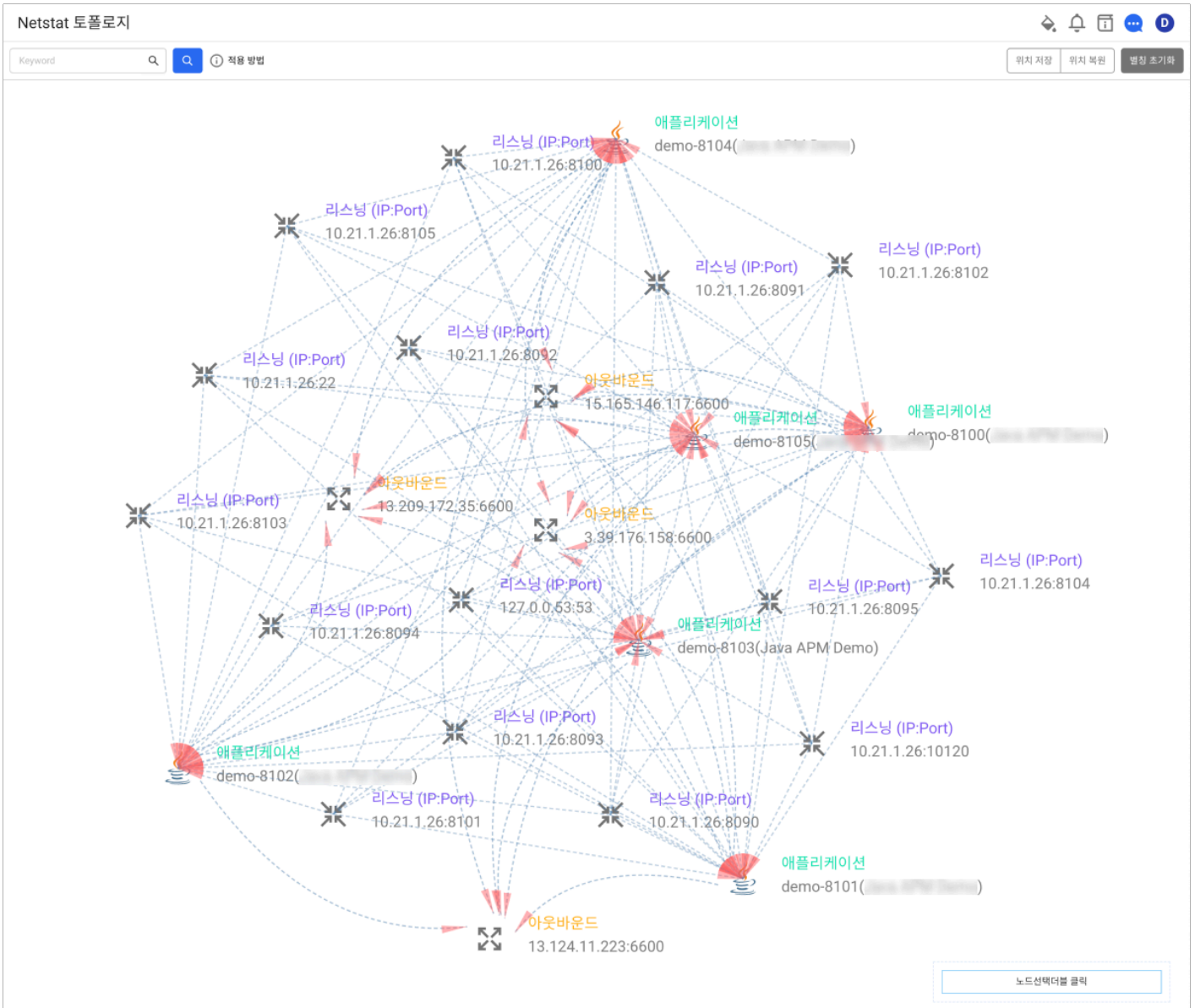




## Netstat 토폴로지

홈 화면 > 프로젝트 선택 > [사이트맵](#) > Netstat 토폴로지

프로젝트에 포함된 애플리케이션과 애플리케이션의 리스닝 정보, 아웃바운드 호출 정보를 노드로 표현합니다. 리스닝 포트와의 관계는 직선으로 아웃바운드 호출 정보와의 관계는 곡선으로 표현합니다.



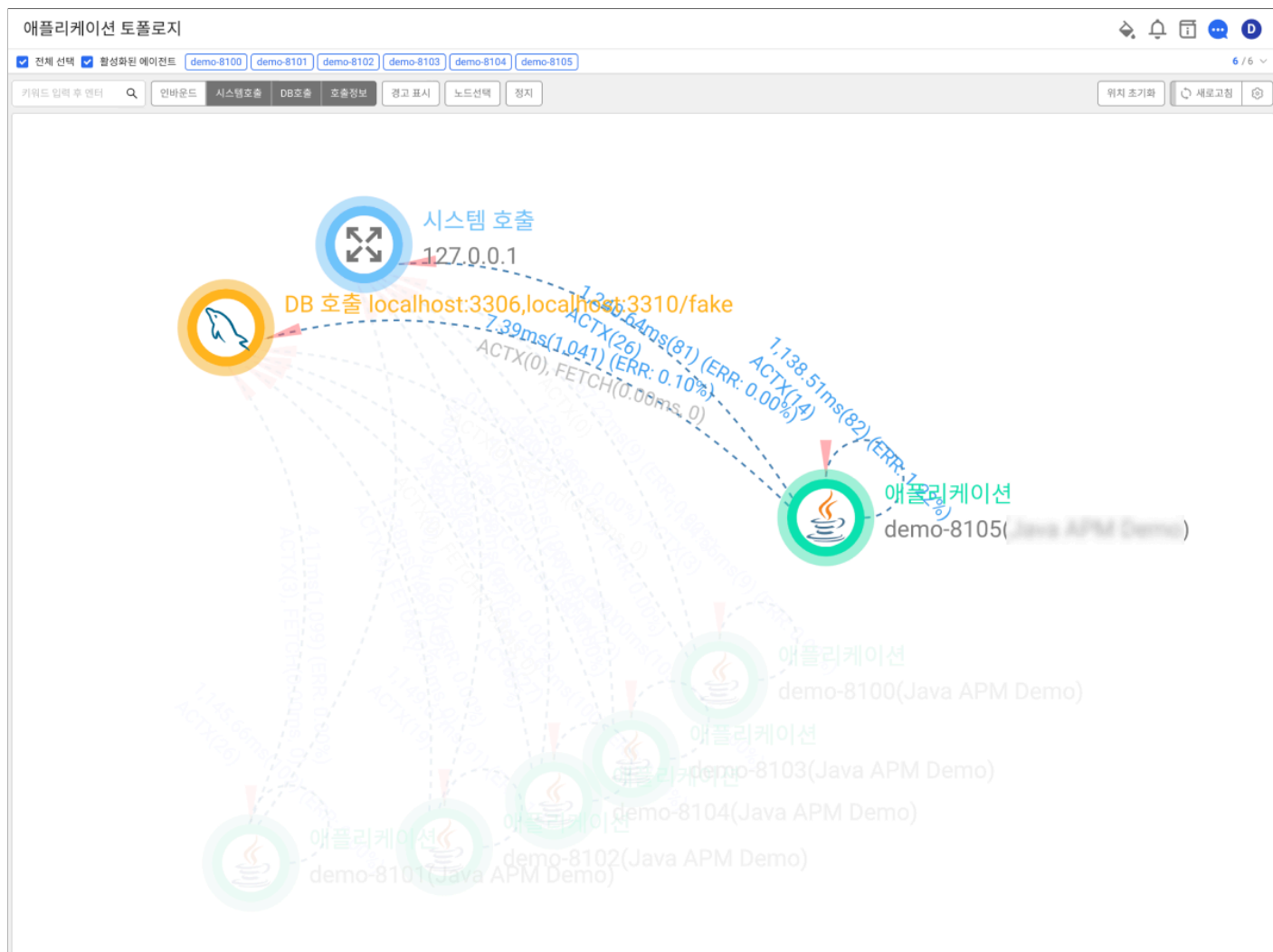
ⓘ Netstat 토폴로지를 취득하는 과정은 에이전트의 부하를 유발하므로 자동 갱신 기능을 제공하지 않습니다. 🔍 버튼을 통해 사용자의 요청이 있는 경우 데이터를 조회합니다.

# 부가 기능

토폴로지 메뉴에서 이용할 수 있는 부가 기능에 대한 설명을 제공합니다.

## 하이라이트 필터

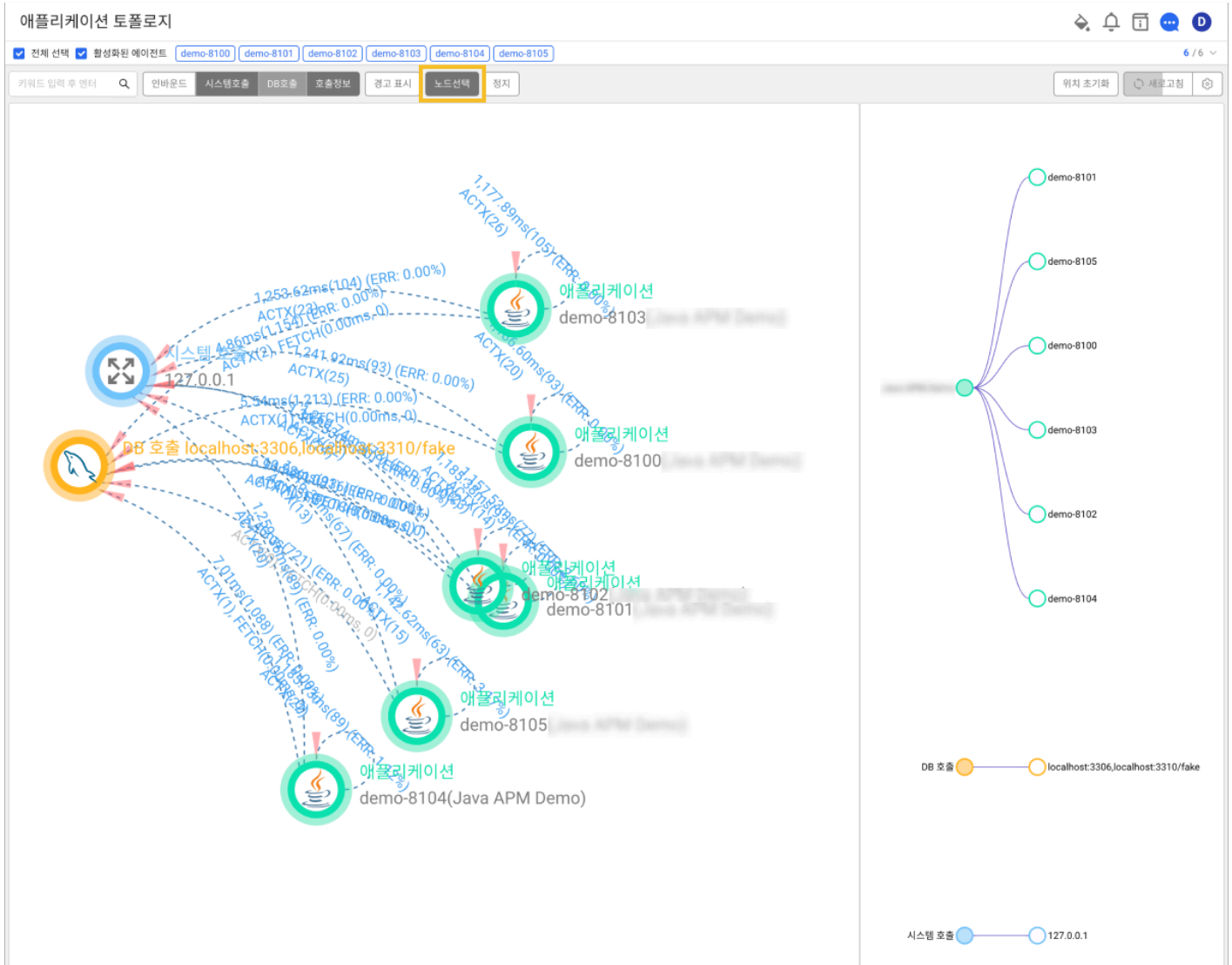
특정 노드를 선택해 연관 관계를 가진 노드를 강조하여 보여주기 위한 기능을 제공합니다. 대량 노드 간의 복잡한 연관 관계를 토폴로지로 용이하게 확인할 수 있습니다.



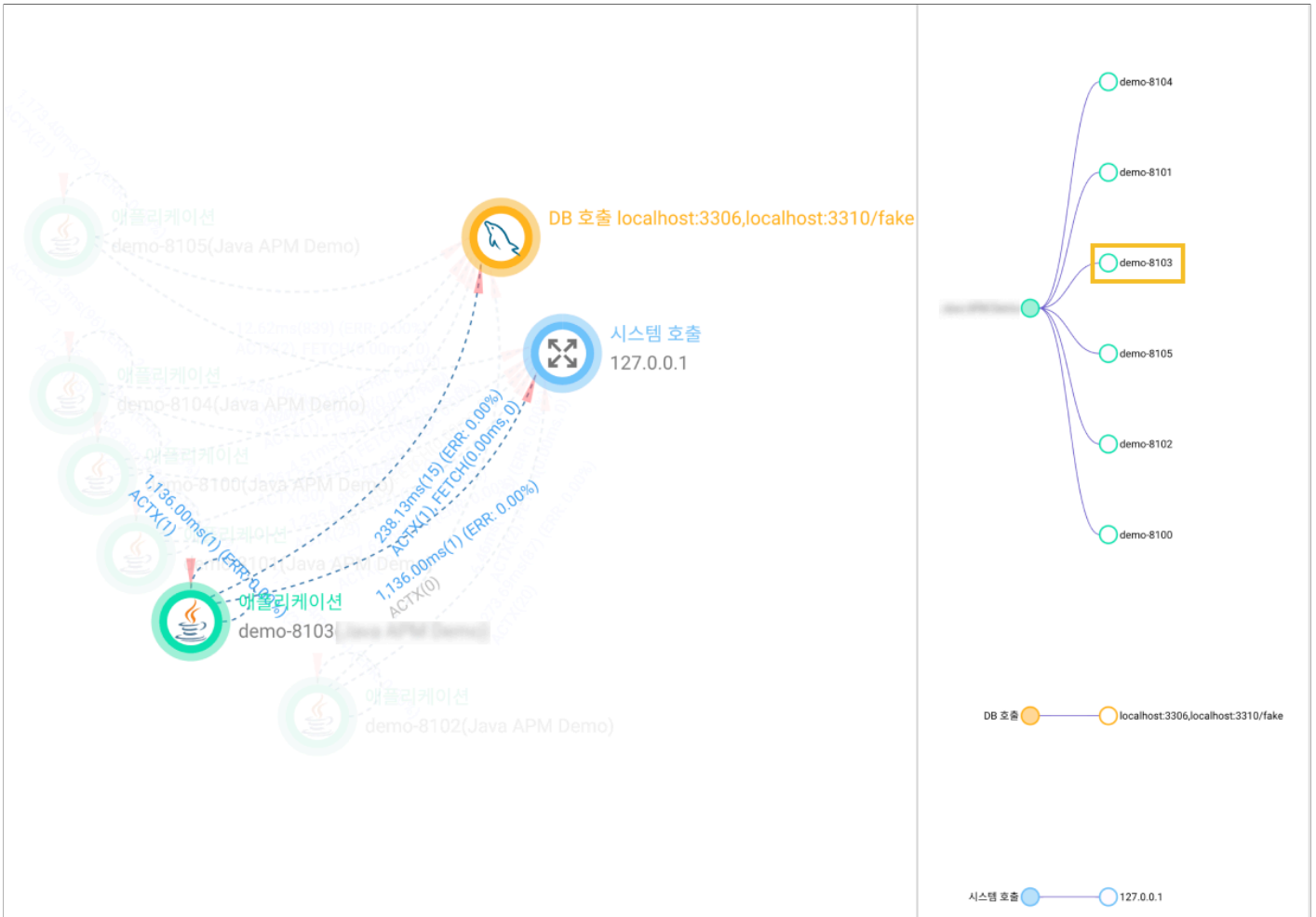
- 특정 노드에 마우스 오버하여 해당 노드 및 연관 노드, 링크 정보를 하이라이트로 표시합니다.
- 키워드 입력을 통해 원하는 노드의 선택을 유지할 수 있습니다. 토폴로지 갱신 시에도 하이라이트 상태는 유지합니다.

## 노드 선택 기능 이용하기

노드 선택 버튼을 선택하세요. 화면 오른쪽에 프로젝트, DB 호출, 외부 호출 노드를 그룹핑하여 다이어그램으로 제공합니다.



오른쪽 다이어그램에서 하위 노드를 선택하면 해당 노드를 왼쪽 토폴로지 화면에서 하이라이트 필터 적용합니다.



오른쪽 화면에서는 다음 기능을 이용할 수 있습니다.

- 줌인/줌아웃
- 드래그
- 하위 노드 클릭 시 하이라이트 필터 적용
- 상위 노드 클릭 시 하위 노드 숨김 또는 표시

하이라이트 필터 기능을 취소하려면 키워드 입력란에 에이전트 이름을 삭제하세요.

① 그룹 통계 토폴로지 메뉴에서는 화면 오른쪽 아래에 **노드선택(더블 클릭)** 버튼으로 해당 기능을 제공합니다.

## 컨텍스트 메뉴

토폴로지의 노드에서 마우스 오른쪽 버튼을 클릭하면 컨텍스트 메뉴가 나타납니다. 컨텍스트 메뉴에는 애플리케이션 모니터링 대시보드로 이동하기 위한 링크 등의 메뉴를 제공합니다.



① 컨테이너 맵, 노드 자원, 컨테이너 자원 메뉴는 컨테이너 모니터링에서만 제공됩니다.

## 노드 위치 저장 및 복원

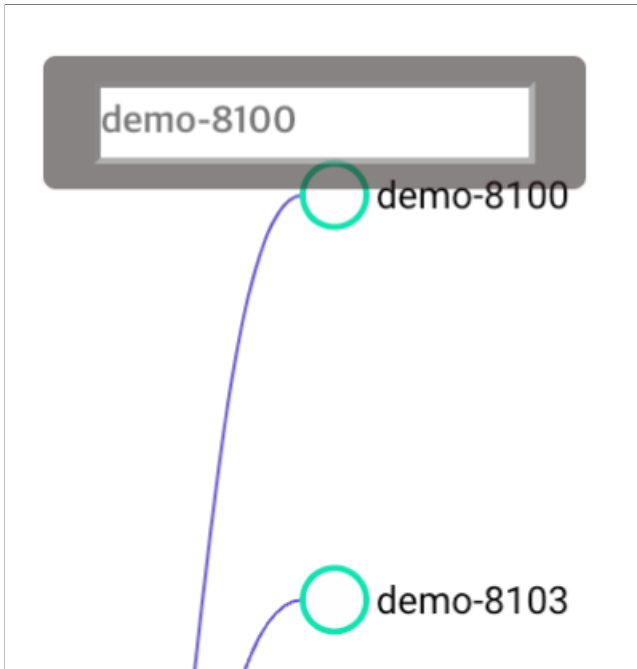
토폴로지 화면에서 차트의 노드 위치는 마우스 왼쪽 버튼을 클릭한 상태에서 드래그해 변경할 수 있습니다. 변경한 노드의 위치는 다른 메뉴로 진입한 후 토폴로지 화면으로 다시 돌아와도 유지됩니다.

- **위치 저장:** 사용자가 변경한 노드의 위치 정보를 저장할 수 있습니다.
- **위치 복원:** 사용자가 위치 저장을 실행한 시점으로 노드의 위치를 복원합니다.
- **위치 초기화:** 노드의 위치를 초기화합니다.

- ①
- 노드 위치 메뉴는 사용자가 진입한 토폴로지 메뉴에 따라 지원하지 않을 수 있습니다.
  - 변경한 노드의 위치 정보는 웹 브라우저의 로컬 스토리지에 저장됩니다. 다른 사용자의 화면에서는 변경된 노드 위치가 반영되지 않습니다.

## 별칭 부여 및 제거

1. **노드선택** 버튼을 선택하세요.
2. 화면 오른쪽에서 별칭을 설정할 노드로 이동해 마우스 오른쪽 버튼을 클릭하세요.



3. 별칭 입력 창이 나타나면 별칭을 입력 후 엔터를 입력하세요.

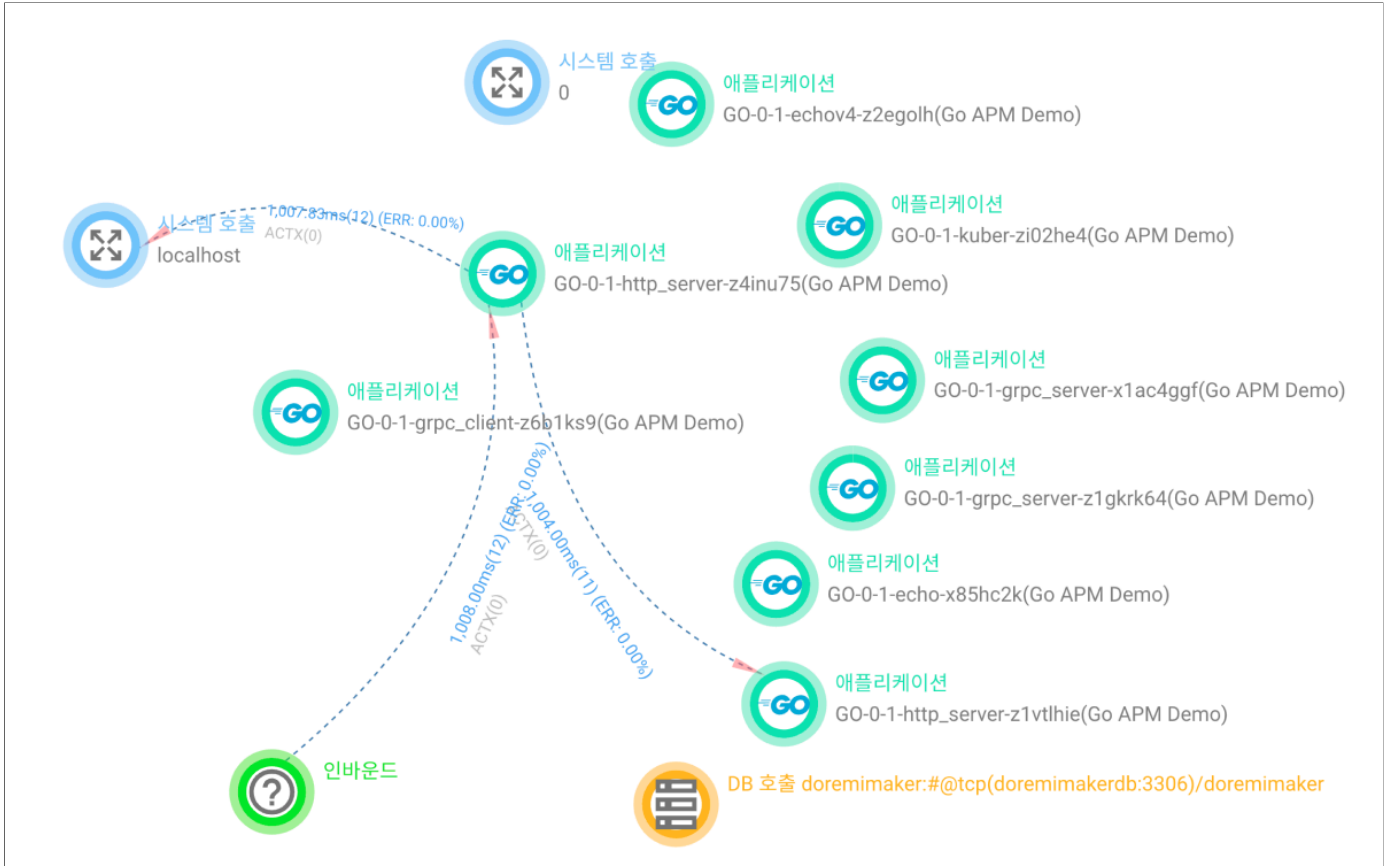
별칭 정보를 저장합니다. 차트의 데이터를 새로 조회하는 시점에서 저장한 별칭을 토폴로지 화면의 노드에 적용합니다.

ⓘ 노드 별칭 정보는 웹 브라우저의 로컬 스토리지에 저장됩니다.

## 인바운드/시스템호출 필터

**인바운드/시스템호출** 버튼을 이용해 토폴로지 화면에 외부 모듈 및 외부 호출 노드의 표시 여부를 변경할 수 있습니다.

- 인바운드/시스템호출 표시 상태



- 인바운드/시스템호출 숨김 상태





## 임계치 초과 정보 강조

노드 사이의 호출 응답 시간 및 에러 건수 임계치 설정 시 임계치를 초과한 노드 및 호출 관계를 강조해 표시합니다. 토폴로지 화면에서 **경고 표시** 버튼을 선택하세요. **응답 시간(ms)**과 **에러 건수** 텍스트 입력 상자에 원하는 설정값을 입력하세요.

키워드 입력 후 엔터 🔍    인바운드    시스템호출    DB호출    호출정보    경고 표시    노드선택    정지

응답시간 > 2000    밀리초  
에러 건수 > 10

애플리케이션    인바운드

sample-1-43-uwsgi(Python APM Demo)

ⓘ 애플리케이션 토폴로지 메뉴에서는 경고 표시를 선택한 다음 설정값을 입력할 수 있습니다.

# 에이전트 옵션 설정

토폴로지 기능을 이용하기 위해 에이전트에 옵션을 적용하는 방법을 제공합니다.

## 적용 가능 에이전트 및 버전

Java 에이전트 1.7.1 버전 이상

## whatap.conf 설정

다음은 토폴로지 표현을 위해 `whatap.conf` 파일에 설정할 수 있는 정보 수집 옵션입니다.

- `tx_caller_meter_enabled` Boolean

기본값 `false`

트랜잭션 정보로 토폴로지 통계를 생성합니다. `mtrace_enabled` 옵션의 값이 `true` 이면 동작합니다.

- `sql_dbc_meter_enabled` Boolean

기본값 `false`

데이터 베이스 연결 정보로 토폴로지 통계를 생성합니다.

- `httpc_host_meter_enabled` Boolean

기본값 `false`

토폴로지 맵에서 httpc outbound 정보를 표현합니다.

- `actx_meter_enabled` Boolean

기본값 `false`

토폴로지 맵에서 액티브 트랜잭션 상태를 표현합니다.

그룹 및 통합 토폴로지 사용을 위해 다음과 같이 옵션을 추가하세요.

```
Java: add JVM option
```

```
-Dwhatap.okind={그룹 식별자}
```

# 통계

홈 화면 > 프로젝트 선택 > 통계

와탭 모니터링 서비스 초기 화면에서 프로젝트를 선택한 다음 [통계](#)의 하위 메뉴를 선택하세요. 와탭 에이전트는 트랜잭션 정보, SQL과 같은 주요 트레이스 정보에 대해 통계 정보를 수집합니다. 5분마다 목록을 만들고 서버로 전송합니다.

- 각 [통계](#) 메뉴는 [합계](#), [최대](#) 및 [평균 시간](#) 및 [건수](#) 기준의 정렬 옵션을 제공해 원하는 관점으로 상위 목록을 조회할 수 있습니다.
- [컬럼 설정](#) 버튼을 통해 기본값 외에 목록 내 조회할 컬럼을 추가하거나 제외할 수 있습니다.
- 통계 목록에서 [상세 보기](#) 아이콘을 선택하면 건수와 평균 시간 차트를 확인할 수 있고, 에이전트 간 건수 및 에러 건수 비교, 응답 시간 비교 차트를 추가로 제공합니다. 해당 차트에서 조회를 원하는 시점을 선택하면 [트랜잭션 검색](#) 메뉴로 이동합니다.
- 통계 목록에서 [T](#) 버튼을 선택하면 [탭 스택 URL](#) 메뉴로 이동합니다.
- 통계 목록에서 [U](#) 버튼을 선택하면 [유니크 스택 URL](#) 메뉴로 이동합니다.

## 트랜잭션 통계

홈 화면 > 프로젝트 선택 > 통계 > 트랜잭션

트랜잭션 통계를 수집합니다. 5분마다 최대 5,000개의 URL 별 수행 통계를 수집하여 서버에 전송합니다. 만약 서로 다른 URL의 수가 5분 동안 5,000개가 넘으면 무시됩니다.

컬럼	설명
트랜잭션	트랜잭션 주소
건수	수행 건수
에러	수행된 트랜잭션 중 에러 발생 건수
평균 시간 (ms)	응답시간 합계에 대한 평균
합계 시간 (ms)	응답시간의 합
최대 시간 (ms)	최대 응답시간

컬럼	설명
평균 CPU 사용 시간	CPU 사용 시간에 대한 평균
HTTP 호출 건수	HTTP Call 건수
평균 HTTP 호출 시간 (ms)	HTTP Call 시간의 합계에 대한 평균
SQL 건수	SQL 수행 건수
평균 SQL 시간 (ms)	평균 SQL 수행 시간
SQL 패치	SQL 패치 건수
평균 SQL 패치 시간 (ms)	SQL 패치 시간의 합계에 대한 평균

ⓘ 평균 메모리 할당량은 에이전트 설정(\*whatap.conf\*)에서 `trace_malloc_enabled=true` 옵션을 설정해야 수집합니다. 기본값은 `false` 입니다.

## 트랜잭션 도메인 통계

홈 화면 > 프로젝트 선택 > 통계 > 도메인

와탭 에이전트는 도메인별 트랜잭션 통계를 수집할 수 있습니다. 하나의 서버에 여러 개의 도메인을 분리하여 서비스하는 시스템에서는 도메인 별 분석이 필요할 수 있습니다. 다음 옵션을 설정하세요.

```
whatap.conf
stat_domain_enabled=true
```

수집되는 데이터는 도메인별 URL의 처리 현황을 파악할 수 있습니다.

컬럼	설명
도메인	서비스 도메인
트랜잭션	트랜잭션 주소
건수	수행 건수
에러	수행된 트랜잭션 중 에러 발생 건수
평균 시간 (ms)	응답시간 합계에 대한 평균

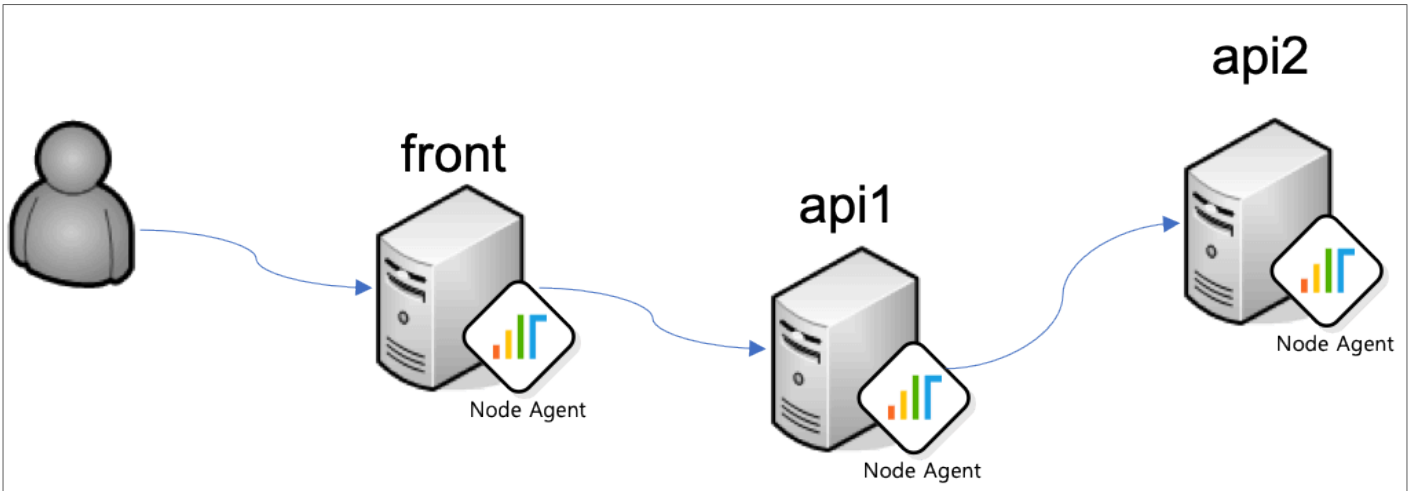
## 트랜잭션 Caller 통계

홈 화면 > 프로젝트 선택 > 통계 > Caller

멀티 서버가 rest 호출로 연결되면 **Caller**와 **Callee** 간의 연관 통계를 수집할 수 있습니다. 이 데이터를 수집하기 위해 다음의 옵션을 설정하세요.

```
whatap.conf
```

```
mtrace_rate=100
mtrace_spec=v1
stat_mtrace_enabled=true
```



위와 같은 아키텍처에서 **Caller & Callee** 통계는 api1, api2에서만 조회할 수 있습니다. 사용자 브라우저에서 호출되는 시스템에서는 **Caller** 통계를 조회할 수 없습니다.

하지만 **Caller** 쪽 서버에서 데이터를 전송해야 하므로 모든 서버에 적절한 설정이 들어가야 합니다.

```
whatap.conf of [front]

mtrace_rate=100
mtrace_spec=v1
stat_mtrace_enabled=true
```

```
whatap.conf of [api1] & [api2]

mtrace_spec=v1
stat_mtrace_enabled=true
```

수집되는 통계 데이터는 다음과 같습니다. **Callee** 쪽에서 조회되어야 합니다.

컬럼	설명
Caller Pcode	Caller에 해당하는 프로젝트 코드
Caller 에이전트 종류	Caller에 해당하는 에이전트가 속한 그룹 단위

컬럼	설명
Caller URL	Caller 트랜잭션의 트랜잭션 주소
트랜잭션	Callee 트랜잭션 주소
건수	수행 건수
에러	수행된 트랜잭션 중 에러 발생 건수
평균 시간 (ms)	응답시간 합계에 대한 평균

## 트랜잭션 Referer 통계

홈 화면 > 프로젝트 선택 > 통계 > 리퍼러

와탭 에이전트는 Referer 별 트랜잭션 통계를 수집할 수 있습니다. 다음 옵션을 설정하세요.

```
whatap.conf
stat_referer_enabled=true
```

수집되는 데이터는 Referer 정보를 통계로 확인할 수 있습니다.

컬럼	설명
Referer	Referer 주소
트랜잭션	트랜잭션 주소
건수	수행 건수
에러	수행된 트랜잭션 중 에러 발생 건수
평균 시간 (ms)	응답시간 합계에 대한 평균



## 에러 통계

홈 화면 > 프로젝트 선택 > 통계 > 에러

5분 동안 발생한 서비스 에러에 대한 통계입니다. 서로 다른 에러와 트랜잭션 이름을 키로 발생 건수를 수집합니다. 에러 발생 시점의 스택을 제공합니다. 5분 당 최대 1,000 가지 서로 다른 에러를 통계화합니다.

컬럼	설명
클래스	에러 클래스
트랜잭션	트랜잭션 주소
메시지	에러 메시지
건수	에러 발생 건수

## SQL 수행 통계

홈 화면 > 프로젝트 선택 > 통계 > SQL

5분 동안의 SQL 수행 통계를 수집합니다. 5분 동안 서로 다른 SQL 문장이 최대 5,000까지만 허용됩니다. 만약 하나의 Java 프로세스에서 한계를 넘는 SQL이 발생하면 통계 데이터에서는 버려집니다.

컬럼	설명
데이터베이스	데이터베이스 종류
SQL	SQL 쿼리문
트랜잭션	해당 SQL을 호출한 URL 중 하나를 임의로 수집한 정보
CRUD	Create, Read, Update, Delete
건수	쿼리 실행 건수

컬럼	설명
에러 건수	실행된 쿼리 중 에러 발생 건수
평균 시간 (ms)	쿼리 실행 시간 합계에 대한 평균
합계 시간 (ms)	쿼리 실행 시간의 합
표준 편차	쿼리 실행 시간에 대한 표준 편차

ⓘ 트랜잭션은 5분 동안 해당 SQL을 호출한 URL 중 하나(마지막 호출 URL)를 분석 활용을 위해 수집합니다.

## HTTP Call 수행 통계

홈 화면 > 프로젝트 선택 > 통계 > HTTP 호출

HTTP 및 HTTPS 프로토콜을 통해 5분 동안의 애플리케이션 외부 호출 통계를 수집합니다. 5분 동안 서로 다른 Http Call 문장이 최대 5,000까지만 허용됩니다. 만약 하나의 애플리케이션 프로세스에서 한계를 넘는 외부 Http Call이 발생하면 통계 데이터에서는 버려집니다.

컬럼	설명
HTTP 호출 URL	HTTP 호출 URL 해시(hash)
트랜잭션	해당 HTTP API를 요청한 URL 중 하나를 임의로 수집한 정보
호스트	Host 또는 IP 주소
포트	TCP 포트
전체	전체 수행 건수
에러	에러 발생 건수
평균 시간 (ms)	응답시간 합계에 대한 평균

컬럼	설명
합계 시간 (ms)	응답시간의 합
최소 시간 (ms)	최소 응답시간
최대 시간 (ms)	최대 응답시간

## 클라이언트 IP 지역별 호출 건수

홈 화면 > 프로젝트 선택 > 통계 > 클라이언트 IP

서버와 클라이언트의 물리적 거리는 데이터 전송 시간과 비례합니다. 클라이언트 IP의 지역별 호출 건수를 통계적으로 수집합니다. 5분당 수집할 수 있는 서로 다른 IP 수는 인스턴스 당 최대 70,000개입니다.

컬럼	설명
클라이언트 IP	클라이언트의 IP 주소
국가	사용자 접속 국가
도시	사용자 접속 도시
건수	전체 호출 건수

ⓘ 와탭은 클라이언트와 관련한 정보를 기본 저장합니다. 사용자 데이터 수집과 관련한 에이전트 설정에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

## 지역별 점유 비중 차트

점유 비중 차트는 사용자의 접속 위치를 국가와 도시 단위로 수집해 제시합니다.

- 국가 영역 선택 시 해당 국가의 도시와 접속 유저의 IP 등을 확인할 수 있는 상세 차트로 이동합니다.
- 중심 원 선택 시 기존 국가 영역 선택 차트로 돌아갑니다.

## 유저 에이전트별 호출 건수

홈 화면 > 프로젝트 선택 > 통계 > [유저 에이전트](#)

유저 에이전트 문자열의 Hash별로 호출 건수를 수집합니다. 5분당 수집할 수 있는 서로 다른 User Agent Hash는 인스턴스 당 최대 500개입니다.

컬럼	설명
유저 에이전트	브라우저의 유저 에이전트 정보
운영체제	브라우저가 실행되는 운영 체제 환경
브라우저	브라우저 종류
건수	수집된 전체 건수

## 클라이언트 브라우저 통계

홈 화면 > 프로젝트 선택 > 통계 > [클라이언트 브라우저](#)

유저 에이전트의 문자열을 분석하여 최종 사용자가 사용하는 브라우저의 종류, 운영체제 기준으로 집계한 통계 정보를 제공합니다. 수집한 통계 정보를 기준으로 생성된 파이 차트를 통해 점유 비중을 확인할 수 있습니다.

컬럼	설명
유저 에이전트	브라우저의 유저 에이전트 정보
운영체제	브라우저가 실행되는 운영 체제 환경
브라우저	브라우저 종류
건수	수집된 전체 건수

# 보고서

홈 화면 > 프로젝트 선택 > 보고서

보고서는 개별 프로젝트에 대해 서비스 이용 현황과 장애 발생 기록을 보고하는 문서입니다. 서비스 모니터링 담당자는 보고서를 통해 관련 부서 담당자들과 현황을 공유합니다.

모니터링 데이터 분석은 서비스의 개선 방향을 정하는 지표가 되기 때문에 중요합니다. 하지만 여러 대시보드의 데이터를 취합해서 문서화하는 일은 번거롭습니다.

와탭의 [보고서](#) 메뉴는 **보고서 작성 업무 자동화**를 지원합니다. 매주 보고서를 작성해야 하는 일, 정해진 시간에 보고서를 공유하는 일, 여러 가지 서식을 관리하는 일 모두 [보고서](#) 메뉴에서 할 수 있습니다.

✔ [보고서](#)를 다운로드하거나 인쇄, 메일 발송 예약을 원하면 해당 버튼을 클릭하세요. HTML 형식으로 다운로드할 수 있습니다.

## 보고서와 통합 보고서의 차이

- [보고서](#)는 하나의 프로젝트에 대한 보고서를 생성하는 메뉴입니다. 따라서 프로젝트를 선택한 다음 사이드 메뉴의 [보고서](#) 메뉴를 통해 진입할 수 있습니다.
- [통합 보고서](#)는 여러 프로젝트에 대한 보고서를 생성하는 메뉴입니다. 프로젝트를 선택하지 않고 홈 화면의 사이드 메뉴에서 진입할 수 있습니다. [통합 보고서](#)에 대한 자세한 내용은 [다음 문서](integrated-report)에서 확인할 수 있습니다.

## 보고서의 종류

기본 보고서는 대기업, 공공기관 및 IT 서비스 기업에서 실제로 사용하고 있는 양식입니다. 원하는 양식이 있으면 [support@whatap.io](mailto:support@whatap.io)로 요청해 주세요. 요청한 보고서는 [보고서](#) 목록에 추가됩니다.

### 보고서 양식 추가

- 요청 양식이 범용성이 있을 경우 [보고서](#) 목록에 추가로 제공합니다.
- 요청 시 HTML 및 CSS 양식이 함께 제공될 경우 추가가 용이합니다.

> [요청 양식 예시](#)

보고서 양식 추가 요청 시 예시와 같이 원하는 HTML 및 CSS 양식 첨부를 권장합니다. 세부 사항은 다음 예시 파일을 참조하세요.

- [예시 파일](#): 다운로드 후 ZIP 형식의 압축을 풀어 HTML 형식의 예시 파일을 확인하세요. 파일 내 주석에 커스터마이징을 위한 CSS 영역과 HTML 영역 안내가 기술되어있습니다.

## 메일 발송 예약

정기적으로 보고서를 이메일로 받으려면 [보고서 메일 발송 예약](#)을 선택하세요. 출근 직후 수행하던 여러 가지 서비스 점검 절차를 메일 확인으로 대체할 수 있습니다.

보고서 메일 발송 예약
×

① 등록 다음날 부터 04~06시에 리포트가 발송됩니다.

② 최대 5건의 리포트 메일 설정이 가능합니다.

**보고서 종류** 리포트

**발송 요일**  Mon  Tue  Wed  Thu  Fri  Sat  Sun

**보고 시간** 00:00 ~ 23:59

**메일**  + 저장

# 애플리케이션 보고서

홈 화면 > 프로젝트 선택 > 보고서

보고서 화면에서  $\surd$  버튼을 클릭하면 일간, 주간, 월간으로 분류된 각종 애플리케이션 분석 보고서를 확인할 수 있습니다.

## 일간 애플리케이션 보고서

**일간 애플리케이션 보고서**는 하루 동안의 애플리케이션 성능을 수치와 차트로 확인할 수 있는 보고서입니다. 시작일, 시작 시간, 종료 시간을 직접 설정하여 조회할 수 있습니다.

### 개요

■ 개요						
방문자(DAU)	요청	에러비율(%)	이벤트 건수	애플리케이션 서버	CPU코어	APDEX*100
940	4,103,039	0.55	12,460	6	12	72.234

■ 피크타임 성능요약(10:10)						
처리량	동시접속 사용자	요청	평균응답시간(ms)	에러비율(%)	애플리케이션 서버	CPU코어
102.68	932	30,791	2,269	0.49	6	12

**개요**는 선택한 조회 기간에 대한 애플리케이션 정보를 개략적으로 볼 수 있습니다. **피크타임 성능 요약**은 하루 동안 처리량이 가장 높았던 시간과 그 시간대의 지표들을 확인할 수 있습니다.

- **방문자(DAU, Daily Active User)**  
선택한 조회 기간 동안 접속한 사용자 수입니다.
- **요청**  
조회 기간 동안 들어온 트랜잭션의 수입니다.
- **에러 비율**

에러 건수에서 트랜잭션 건수를 나눈 값입니다.

- **치명적 이벤트**

조회 기간동안 발생한 매트릭스 이벤트 중 레벨이 critical인 이벤트의 수입니다.

- **애플리케이션 서버**

조회 기간동안 모니터링한 애플리케이션 수(에이전트 수)입니다.

- **CPU 코어**

조회 기간동안의 CPU 코어 수입니다.

- **APDEX\*100(Application Performance Index)**

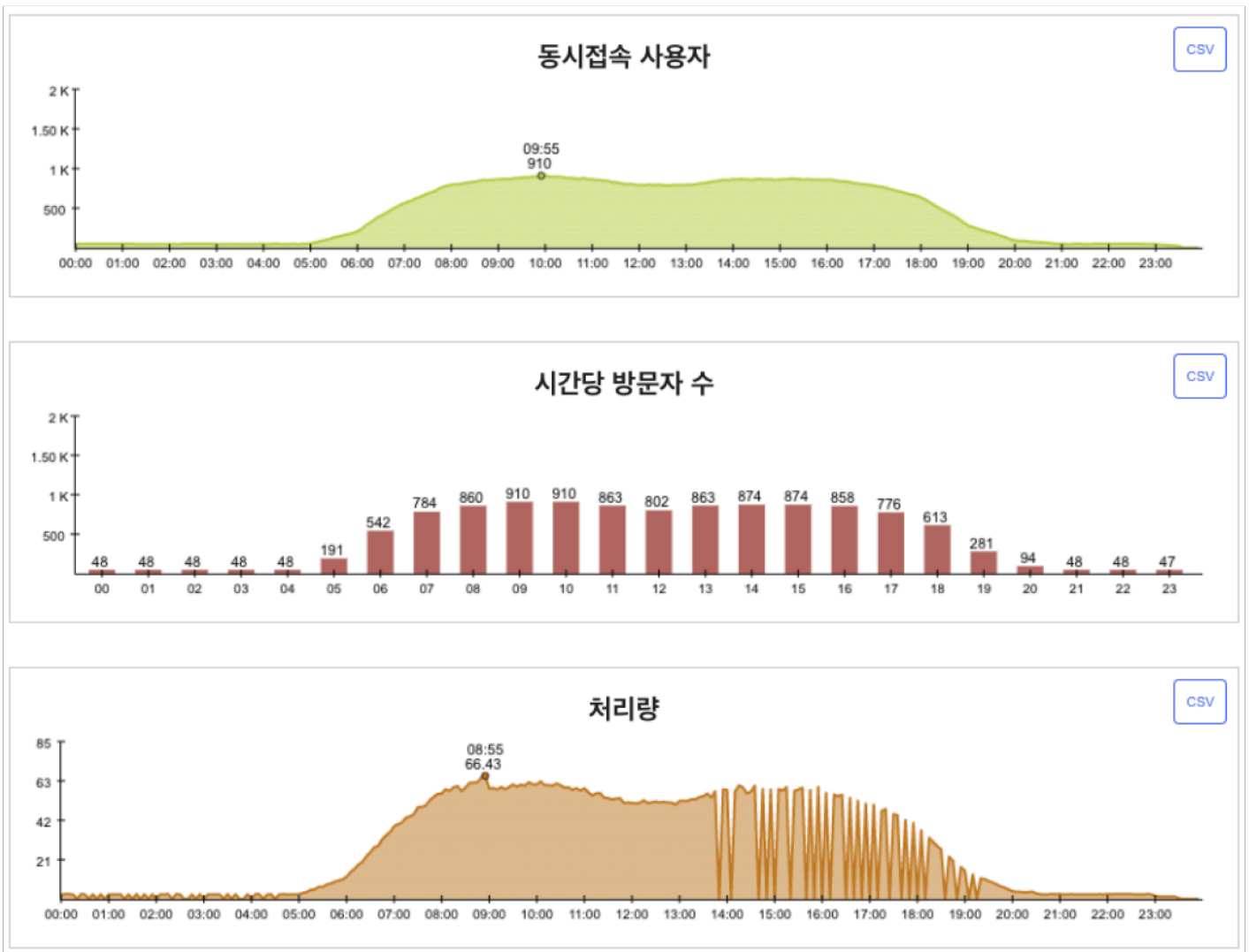
조회 기간 동안 애플리케이션에 대한 고객 성능 만족도입니다.

- **처리량(TPS, Transaction Per Second)**

초당 처리된 트랜잭션의 수를 의미하며, 서비스 성능 지표의 기준이 됩니다.



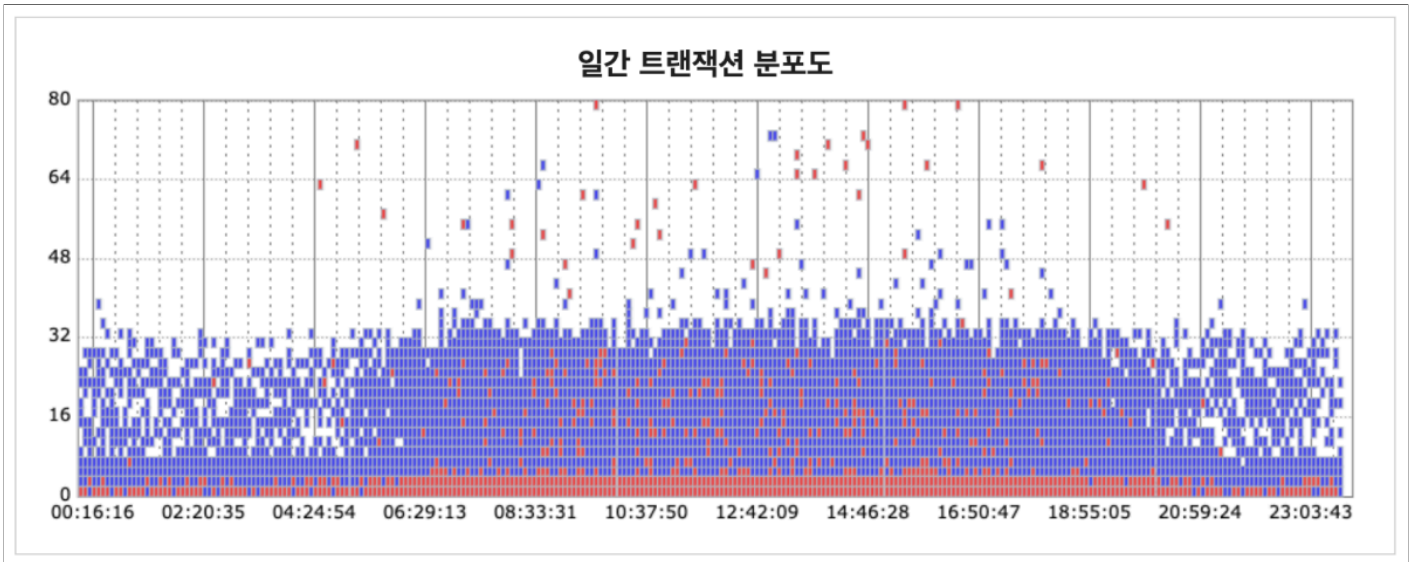
## 성능 추이 차트



성능 추이 차트를 통해 주요 지표들을 시각화하여 볼 수 있습니다.

성능 추이 차트에서 볼 수 있는 지표로는 동시접속 사용자, 시간당 방문자 수, 처리량(TPS), 평균 응답시간, CPU 사용률, 힙 사용량, 에러 건수, 일간 트랜잭션 분포도가 있습니다. 각 그래프에서는 해당 지표가 가장 높았던 시간과 수치를 함께 확인할 수 있습니다.

차트 우측 상단의 CSV 버튼을 이용하여 데이터를 엑셀 파일로 다운로드 할 수 있습니다.



일간 트랜잭션 분포도에서는 시간대별 트랜잭션 분포를 한눈에 확인할 수 있습니다. 그래프의 x축은 각 트랜잭션의 종료 시간, y축은 응답시간(초)을 의미합니다. 정상 트랜잭션의 경우 파란색으로 표시되고, 에러가 발생한 경우 빨간색으로 표시됩니다.

## 애플리케이션별 현황

애플리케이션 서버	IP	요청건수	에러비율 (%)	피크타임성능(10:10)					일일평균성능				
				방문자 수	처리량	평균응답시간 (ms)	액티브 건수	APDEX *100	방문자 수	처리량	평균응답시간 (ms)	액티브 건수	APDEX *100
demo-8100	192.168.1.100	676,948	0.54	162	15.84	2436.37	36.63	72.198	76.7	7.83	6789.29	18.47	72.285
demo-8101	192.168.1.101	690,836	0.55	153	16.97	2216.39	37.2	71.753	74.7	7.99	5401.21	18.56	72.249
demo-8102	192.168.1.102	683,859	0.54	154	16.40	2436.71	38.33	71.956	74.27	7.91	4202.0	18.55	72.248
demo-8103	192.168.1.103	685,558	0.54	156	17.85	2157.21	38.15	72.195	75.11	7.93	6709.14	18.53	72.164
demo-8104	192.168.1.104	682,709	0.54	151	18.71	2134.25	40.66	72.833	74.14	7.9	3926.35	18.53	72.23
demo-8105	192.168.1.105	683,129	0.54	155	16.90	2230.4	37.54	72.059	73.67	7.9	4014.86	18.5	72.222

모니터링 대상별로 피크타임 성능과 일일 평균 성능을 확인할 수 있습니다. 액티브 건수는 진행 중인 트랜잭션 건수를 말합니다.

## 트랜잭션 TOP 10

1. 트랜잭션 수

번호	서비스(URL)	건수	에러건수(증감%)
1	/account/save/employee/seoul	17,373	18 (▼-22.22 %)
2	<b>/account/save/employee/pusan</b>	15,622	<b>23 (▲26.09 %)</b>
3	/account/save/employee/daegu	14,607	12 (▼-25.0 %)
4	/account/save/employee/kwangju	13,939	16 (▼-25.0 %)
5	/account/save/employee/jeju	13,380	10 (▼-110.0 %)
6	<b>/account/save/employee/daejun</b>	13,045	<b>16 (▲12.5 %)</b>
7	<b>/account/save/dept/seoul</b>	12,741	<b>19 (▲31.58 %)</b>
8	/account/save/dept/pusan	12,250	13 (▼-38.46 %)
9	/account/save/dept/daegu	12,241	14 (▼-35.71 %)
10	/account/save/dept/kwangju	12,051	11 (▼-18.18 %)

2. 에러건수

번호	서비스(URL)	건수	에러건수(증감%)
1	/account/save/division/kwangju	11,074	11,068 (▼-19.03 %)
2	/product/remove/dept/daegu	4,495	4,490 (▼-20.91 %)
3	/order/create/dept/seoul	2,774	2,772 (▼-16.05 %)
4	<b>/account/save/employee/pusan</b>	15,622	<b>23 (▲26.09 %)</b>
5	<b>/account/save/dept/seoul</b>	12,741	<b>19 (▲31.58 %)</b>
6	/account/save/employee/seoul	17,373	18 (▼-22.22 %)
7	<b>/account/save/division/jeju</b>	10,822	<b>16 (▲25.0 %)</b>
8	/account/save/employee/kwangju	13,939	16 (▼-25.0 %)
9	<b>/account/save/employee/daejun</b>	13,045	<b>16 (▲12.5 %)</b>
10	<b>/account/load/employee/jeju</b>	7,604	<b>16 (▲25.0 %)</b>

3. 평균응답시간

번호	서비스(URL)	건수	에러건수	평균응답시간(증감%)
1	<b>/product/read/employee/daejun</b>	3,786	2	<b>4,257ms (▲42.92%)</b>
2	<b>/edu/save/unit/seoul</b>	2,210	3	<b>3,315ms (▲30.23%)</b>
3	<b>/order/pickup/employee/seoul</b>	2,870	3	<b>3,241ms (▲26.04%)</b>
4	<b>/account/kill/employee/daegu</b>	6,935	9	<b>3,143ms (▲26.92%)</b>
5	<b>/order/create/dept/daejun</b>	2,743	0	<b>3,117ms (▲25.79%)</b>
6	<b>/edu/delete/employee/kwangju</b>	2,277	2	<b>3,100ms (▲22.52%)</b>
7	<b>/edu/create/dept/jeju</b>	1,957	1	<b>3,100ms (▲23.87%)</b>
8	<b>/product/write/dept/jeju</b>	3,579	4	<b>3,021ms (▲24.3%)</b>
9	<b>/account/create/unit/kwangju</b>	5,111	6	<b>2,947ms (▲21.68%)</b>
10	<b>/edu/write/dept/pusan</b>	1,957	2	<b>2,947ms (▲24.02%)</b>

하루 동안 트랜잭션 건수, 에러 건수, 평균 응답시간이 높았던 트랜잭션을 상위 10개 항목을 뽑아서 보여줍니다.

## 탑 스택 TOP 10

번호	퍼센트	건수(증감%)	스택
1	49.71%	476,098 (▼-71.5 %)	java.base@17.0.8/java.lang.Thread.sleep(Native Method)
2	45.74%	438,108 (▼-71.93 %)	java.base@17.0.8/sun.nio.ch.SocketDispatcher.read0(Native Method)
3	4.51%	43,196 (▼-73.31 %)	app/com.virtual.web.Simula.execute3(Simula.java:175)
4	0.00%	45 (▼-78.87 %)	java.management@17.0.8/sun.management.ThreadImpl.getThreadTotalCpuTime0(Native Method)
5	0.00%	42 (▼-74.55 %)	app/com.virtual.dao.SelectDAO.execute2(SelectDAO.java:79)
6	0.00%	34 (▼-74.05 %)	java.base@17.0.8/java.lang.Thread.start0(Native Method)
7	0.00%	17 (▼-72.13 %)	app/org.apache.commons.dbcp.BasicDataSource.getConnection(BasicDataSource.java:52)
8	0.00%	16 (▼-69.23 %)	java.base@17.0.8/java.security.AccessController.getStackAccessControlContext(Native Method)
9	0.00%	14 (▼-69.57 %)	java.base@17.0.8/java.lang.System.identityHashCode(Native Method)
10	0.00%	12 (▼-70.73 %)	java.base@17.0.8/sun.nio.ch.Net.connect0(Native Method)

Stack Trace 상의 Step을 기준으로 Step 간의 호출 빈도를 비율로 계산한 정보를 간소화하여 볼 수 있습니다. [분석](#) > [스택](#) > [탑 스택](#) 페이지에서 Stack 상세 내용을 조회할 수 있습니다.

## SQL 평균응답시간 TOP 10

번호	SQL	건수	응답시간(증감%)
1	insert into emp values( #, '#', '#', #, to_date('#, '#) - #, #, null, # )	9,074	110ms (▲ 1.85%)
2	INSERT INTO EMP VALUES (#, '#', '#, NULL, '#, #, NULL, #)	22,194	109ms (▲ 0.93%)
3	INSERT INTO SALGRADE VALUES (#, #, #)	64,401	108ms (▲ 0.93%)
4	INSERT INTO DEPT VALUES (#, '#, '#)	62,334	108ms (▲ 1.89%)
5	insert into emp values( #, '#', '#, #, to_date('#, '#) - #, #, null, # )	10,467	108ms (▲ 0.93%)
6	update table set x=# where key=#	344,742	107ms (▲ 0.0%)
7	DELETE FROM posts where id=(SELECT id FROM posts GROUP BY id HAVING ( COUNT(id) > # ))	76,320	107ms (▲ 0.94%)
8	INSERT INTO EMP VALUES (#, '#, '#, #, '#, #, NULL, #)	229,320	106ms (▼-0.93 %)
9	DELETE FROM Customers WHERE CustomerName='#' AND ContactName='#'	171,333	106ms (▼-1.85 %)
10	INSERT INTO EMP VALUES (#, '#, '#, #, '#, #, #, #)	124,338	106ms (▼-1.85 %)

조회 기간 동안 사용된 SQL 쿼리문에 대해 평균 응답시간을 기준으로 정렬한 통계 정보입니다. [통계/보고서](#) > [통계](#) > [SQL](#) 페이지에서 각 쿼리의 발생 시간 및 성능을 상세 조회할 수 있습니다.

### ⓘ 증가%

조회 전일까지 3일간의 데이터를 평균 낸 값과 조회 당일 데이터를 비교하여 증가한 비율을 계산한 수치입니다.

#### • 예시

4월 25일 00시 00분 ~ 23시 59분 데이터 조회 → 4월 22일 00시 00분 ~ 4월 24일 23시 59분 데이터와 비교

해당 수치가 10% 이상인 경우 **붉은색**으로, 5%~10% 사이인 경우 **주황색**으로 표시됩니다. 5% 미만의 증가율은 표시하지 않습니다.

## 주간 애플리케이션 보고서

[주간 애플리케이션 보고서](#)는 일주일간 애플리케이션 성능을 수치와 차트로 확인할 수 있는 보고서입니다. 시작일을 직접 설정할 수 있으며, 조회 당일 기준 7일 전부터 조회 가능합니다.

## 개요

주간 총 방문자수	주간 트랜잭션건수	APDEX*100
927	21,472,203	82.992

- 주간 총 방문자 수(Weekly Active User)**  
 일주일 간 접속한 사용자 수입니다.
- 주간 트랜잭션 건수**  
 일주일 간 들어온 요청이 처리되어 응답이 반환된 수입니다.
- APDEX\*100**  
 조회 기간 동안 애플리케이션에 대한 고객 성능 만족도를 나타냅니다.

## 가동률

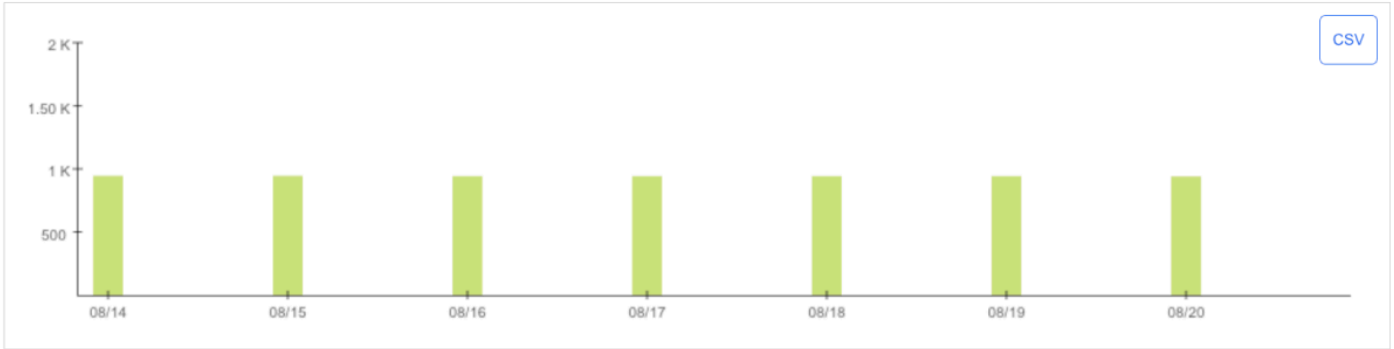
애플리케이션	14(Mon)	15(Tue)	16(Wed)	17(Thu)	18(Fri)	19(Sat)	20(Sun)	14(Mon) ~ 20(Sun)
demo-8100	100.0 %	100.0 %	100.0 %	100.0 %	100.0 %	100.0 %	100.0 %	100.0 %
demo-8101	100.0 %	100.0 %	100.0 %	100.0 %	100.0 %	100.0 %	100.0 %	100.0 %
demo-8102	100.0 %	100.0 %	100.0 %	100.0 %	100.0 %	100.0 %	100.0 %	100.0 %
demo-8103	100.0 %	100.0 %	100.0 %	100.0 %	100.0 %	100.0 %	100.0 %	100.0 %
demo-8104	100.0 %	100.0 %	100.0 %	100.0 %	100.0 %	100.0 %	100.0 %	100.0 %
demo-8105	100.0 %	100.0 %	100.0 %	100.0 %	100.0 %	100.0 %	100.0 %	100.0 %

모니터링 대상별 가동률을 일자별로 확인할 수 있으며, 주간 평균을 확인할 수 있습니다. 데이터가 없으면 - 로 표시되며, 데이터는 있으나 그 값이 작은 경우 0 으로 표시될 수 있습니다.

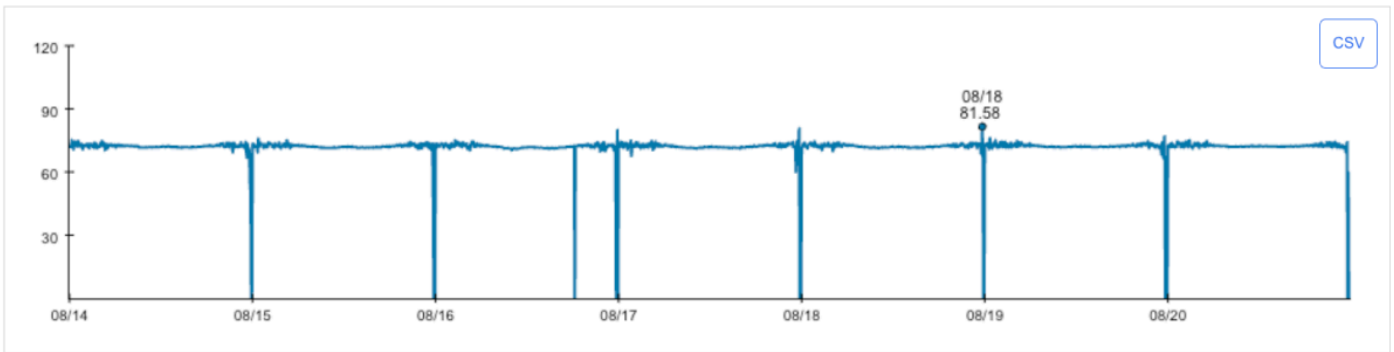
## 성능 추이 차트



방문자 수

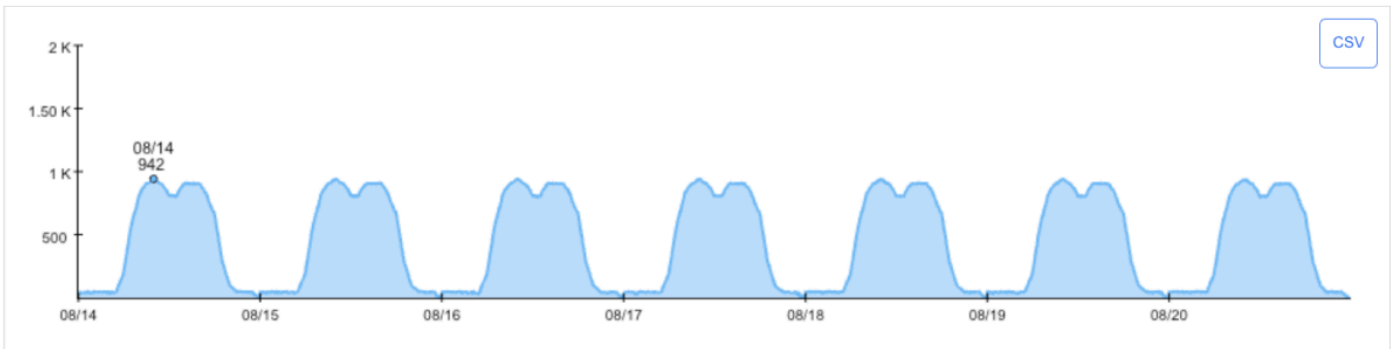


APDEX\*100



◎	14(Mon)	15(Tue)	16(Wed)	17(Thu)	18(Fri)	19(Sat)	20(Sun)	14(Mon) ~ 20(Sun)
평균	72.421	71.946	71.856	71.640	71.793	71.891	72.097	71.910

실시간 사용자



◎	14(Mon)	15(Tue)	16(Wed)	17(Thu)	18(Fri)	19(Sat)	20(Sun)	14(Mon) ~ 20(Sun)
평균	456	460	455	455	458	454	458	456
최대	942 (09:55)	942 (10:05)	942 (10:00)	940 (09:50)	940 (09:55)	940 (09:55)	937 (09:50)	942

주간 애플리케이션 보고서의 차트에는 **방문자 수, APDEX\*100, 실시간 사용자, 트랜잭션 수, 응답시간, 처리량, 액티브 트랜잭션, CPU 사용률, Heap 사용량**이 있습니다.

차트의 x 축은 날짜, y 축은 지표의 수치를 나타내며, 주간 데이터 중 지표 값이 가장 높았던 날짜와 그 값을 그래프 위의 텍스트로 확인할 수 있습니다. 그래프 아래의 표에서는 일자별 평균값, 최댓값, 주간 평균값 등을 확인할 수 있습니다.

차트 우측 상단의 **CSV** 버튼을 이용하여 데이터를 엑셀 파일로 다운로드 할 수 있습니다.

## 트랜잭션 TOP 10

트랜잭션 TOP 10 비교에서는 시작일을 기준으로 일주일간의 데이터와 그 전 주의 데이터를 비교해볼 수 있습니다.

각 트랜잭션에 대한 상세 정보 조회를 원하는 경우 [통계/보고서](#) > [통계](#) > [트랜잭션](#) 메뉴에서 조회할 수 있습니다.

### ⓘ 예시 시작일 4월 22일

- 조회 주: 4월 22일 00시 00분 ~ 4월 28일 23시 59분
- 일주일 전: 4월 15일 00시 00분 ~ 4월 21일 23시 59분

## 트랜잭션 수

번호	서비스(URL)	일주일 전 건수	조회 주 건수	증감건 (증감율)
1	/account/save/employee/seoul	144,266	144,969	703 (▲0.48%)
2	/account/save/employee/pusan	127,850	128,491	641 (▲0.5%)
3	/account/save/employee/daegu	120,461	120,413	-48 (▼0.04%)
4	/account/save/employee/kwangju	115,987	115,114	-873 (▼0.76%)
5	/account/save/employee/jeju	111,561	111,021	-540 (▼0.49%)
6	/account/save/employee/daejun	108,123	107,714	-409 (▼0.38%)
7	/account/save/dept/seoul	104,849	105,325	476 (▲0.45%)
8	/account/save/dept/pusan	102,541	102,590	49 (▲0.05%)
9	/account/save/dept/daegu	101,302	100,608	-694 (▼0.69%)
10	/account/save/dept/kwangju	98,400	98,901	501 (▲0.51%)

일주일 간 들어온 트랜잭션을 트랜잭션 수를 기준으로 내림차순 정렬한 결과입니다.

- 트랜잭션 수

해당 애플리케이션에 들어온 요청이 처리되어 응답이 반환된 수를 의미합니다. 일주일 간 어떤 서비스(URL)가 가장 활발했는지를 확인할 수 있는 지표가 됩니다.

- 증감건

조회 주와 조회 전주의 트랜잭션 수를 비교한 수치이며, 증감률은 이를 백분율로 계산한 수치입니다.

## 에러 건수

번호	서비스(URL)	일주일 전 에러건수	조회 주 에러건수	증감건 (증감율)
1	/account/save/division/kwangju	90,477	91,205	728 (▲0.8%)
2	/product/remove/dept/daegu	37,025	37,501	476 (▲1.27%)
3	/order/create/dept/seoul	22,305	22,295	-10 (▼0.04%)
4	/account/save/employee/seoul	140	145	5 (▲3.45%)
5	/account/save/employee/pusan	139	136	-3 (▼2.21%)
6	/account/save/employee/jeju	93	131	38 (▲29.01%)
7	/account/save/employee/kwangju	115	128	13 (▲10.16%)
8	/account/save/dept/pusan	94	117	23 (▲19.66%)
9	/account/save/dept/daegu	87	117	30 (▲25.64%)
10	/account/save/employee/daejeon	119	115	-4 (▼3.48%)

일주일 간 들어온 트랜잭션을 에러 건수를 기준으로 내림차순 정렬한 결과입니다.

- **에러 건수**

어떤 서비스(URL)에서 에러가 많이 발생했는지를 확인할 수 있는 통계 데이터입니다.

- **증감건**

조회 주와 조회 전주의 에러 건수를 비교한 수치이며, 증감률은 이를 백분율로 계산한 수치입니다.

## 평균 응답 시간

번호	서비스(URL)	일주일 전 평균 시간	조회 주 평균 시간	증감시간 (증감율)
1	/order/load/unit/pusan	2,440ms	3,672ms	1,232ms (▲33.55%)
2	/order/read/dept/seoul	2,339ms	2,927ms	588ms (▲20.09%)
3	/edu/write/division/pusan	2,459ms	2,850ms	391ms (▲13.72%)
4	/sales/save/dept/jeju	2,309ms	2,760ms	451ms (▲16.34%)
5	/sales/delete/unit/seoul	2,317ms	2,755ms	438ms (▲15.9%)
6	/edu/pickup/division/kwangju	2,377ms	2,718ms	341ms (▲12.55%)
7	/order/pickup/dept/daegu	2,357ms	2,630ms	273ms (▲10.38%)
8	/product/read/employee/daejun	2,396ms	2,610ms	214ms (▲8.2%)
9	/order/kill/dept/daegu	2,460ms	2,603ms	143ms (▲5.49%)
10	/product/write/employee/daejun	2,350ms	2,581ms	231ms (▲8.95%)

일주일 간 들어온 트랜잭션을 평균 응답시간을 기준으로 내림차순 정렬한 결과입니다.

- **평균 시간**

애플리케이션 서버가 사용자에게 요청 결과를 반환하는데 걸린 시간을 의미하며, 단위는 밀리 세컨드(ms)로 표시됩니다.

- **증감 시간**

조회 주와 조회 전주의 응답 시간을 비교한 수치이며, 증감률은 이를 백분율로 계산한 수치입니다.

## 탑 스택 TOP 10

번호	스택	일주일 전 퍼센트	조회 주 퍼센트	일주일 전 건수	조회 주 건수	증감건 (증감율)
1	java.base@17.0.8/java.lang.Thread.sleep(Native Method)	26.87%	49.26%	2,101,010	3,853,146	1,752,136 (▲83.39%)
2	java.base@17.0.8/sun.nio.ch.SocketDispatcher.read0(Native Method)	25.13%	45.91%	1,964,818	3,591,227	1,626,409 (▲82.78%)
3	app//com.virtual.web.Simula.execute3(Simula.java:175)	22.38%	4.79%	1,749,961	374,595	-1,375,366 (▼-78.59%)
4	java.base@17.0.8/java.lang.Thread.start0(Native Method)	20.79%	0.01%	1,625,529	537	-1,624,992 (▼-99.97%)
5	java.management@17.0.8/sun.management.ThreadImpl.getThreadTotalCpuTime0(Native Method)	4.79%	0.00%	374,864	346	-374,518 (▼-99.91%)
6	app//com.virtual.dao.SelectDAO.execute2(SelectDAO.java:79)	0.01%	0.00%	492	273	-219 (▼-44.51%)
7	java.base@17.0.8/java.security.AccessController.getStackAccessControlContext(Native Method)	0.00%	0.00%	260	150	-110 (▼-42.31%)
8	app//org.apache.commons.dbcp.BasicDataSource.getConnection(BasicDataSource.java:52)	0.00%	0.00%	212	119	-93 (▼-43.87%)
9	java.base@17.0.8/sun.nio.ch.Net.connect0(Native Method)	0.00%	0.00%	191	111	-80 (▼-41.88%)
10	app//jdbc.FakePreparedStatement.executeQuery(FakePreparedStatement.java:67)	0.00%	0.00%	162	104	-58 (▼-35.8%)

Stack Trace 상의 Step을 기준으로 Step 간의 호출 빈도를 비율로 계산한 결과입니다. 해당 스택에 대한 일주일 전 데이터와 조회 주 데이터를 비교해 볼 수 있습니다.

- 증감건

조회 주와 조회 전 주의 스택 호출 건수를 비교한 수치이며, 증감률은 이를 백분율로 계산한 수치입니다.

## SQL 평균 응답시간 TOP 10

번호	SQL	일주일 전 건수	조회 주 건수	일주일 전 평균 시간	조회 주 평균 시간	증감시간 (증감율)
1	DELETE FROM suppliers WHERE supplier_name = '#'	798,667	797,242	108ms	108ms	-
2	INSERT INTO SALGRADE VALUES (#, #, #)	539,777	538,818	107ms	108ms	1ms (▲0.93%)
3	INSERT INTO EMP VALUES (#, '#', '#', NULL, '#', #, NULL, #)	183,192	184,551	106ms	108ms	2ms (▲1.89%)
4	update table set x=# where key=#	2,864,916	2,868,480	107ms	107ms	-
5	INSERT INTO EMP VALUES (#, '#', '#', '#', #, NULL, #)	1,910,668	1,911,591	107ms	107ms	-
6	DELETE FROM Customers WHERE CustomerName='#' AND ContactName='#'	1,431,282	1,434,971	107ms	107ms	-
7	insert into emp values( #, '#', '#', #, to_date('#,##), #, null, # )	584,222	586,430	108ms	107ms	-1ms (▼0.93%)
8	INSERT INTO EMP VALUES (#, '#', '#', '#', #, #, #)	1,033,185	1,034,649	108ms	107ms	-1ms (▼0.93%)
9	DELETE FROM posts where id=(SELECT id FROM posts GROUP BY id HAVING ( COUNT(id) > # ))	636,714	637,584	107ms	107ms	-
10	INSERT INTO DEPT VALUES (#, '#', '#')	514,112	516,942	107ms	107ms	-

SQL 쿼리문에 대해 평균 응답시간을 기준으로 정렬한 결과입니다. 각 쿼리문에 대한 주간 건수 및 평균 시간을 비교해 볼 수 있습니다.

### • 증감 시간

조회 주와 조회 전 주의 SQL 처리 평균 시간을 비교한 수치이며, 증감률은 이를 백분율로 계산한 수치입니다.

## 프로젝트 에러 현황

프로젝트명	에러 클래스	에러 메시지	일주일 전	조회 주	증감건 (증감율)
			08/07 ~ 08/13	08/14 ~ 08/20	
www.nhn.com	REQUEST_REJECT	rejected request	149,807	151,000	1,193 (▲0.8%)
	java.sql.SQLException	Sql Exception	20,388	20,809	421 (▲2.06%)
	java.lang.RuntimeException	Internal RuntimeException	6,671	6,772	101 (▲1.51%)
	java.lang.Error	Unknown Error!!	6,735	6,760	25 (▲0.37%)
	SLOW_SQL	SLOW_SQL	500	698	198 (▲39.6%)

프로젝트 에러 현황에서는 에러 데이터를 주간별로 비교해 볼 수 있습니다. 에러가 발생한 클래스와 에러 메시지를 확인할 수 있습니다.

### • 에러 발생 건수

에러 클래스와 에러 메시지가 같은 것끼리 합산하여 내림차순 정렬된 결과입니다.

### • 증감건

조회 주와 조회 전 주의 에러 발생 건수를 비교한 수치이며, 증감률은 이를 백분율로 계산한 수치입니다.

#### ⓘ 증감률 색상별 의미

- **파란색**: 증감율이 음수인 경우로 전 주에 비해 감소했음을 의미합니다.
- **주황색**: 증감율이 0% 이상 10% 미만인 경우를 의미합니다.
- **빨간색**: 증감율이 10% 이상이거나, 전 주에는 없었던 새로운 데이터인 경우를 의미합니다.

## 월간 애플리케이션 보고서

월간 애플리케이션 보고서는 한 달 동안의 애플리케이션 성능을 수치와 차트로 확인할 수 있는 보고서입니다. 시작일을 직접 설정할 수 있으며, 조회 당일 기준 30일 전부터 조회 가능합니다.



## 개요

월간 트랜잭션 건수	월간 사용자수	가동률	애플리케이션 수	전체 코어	Apdex*100
150,283,340	1,839	100 %	6	2 Core	71.948

개요에서 한 달간의 전체 트랜잭션 건수, 사용자 수, 가동률, 애플리케이션 수, 코어 수, Apdex\*100 수치를 한 번에 확인할 수 있습니다.

- **월간 트랜잭션 건수**

한 달 동안 들어온 전체 트랜잭션의 수입입니다.

- **월간 사용자 수(MAU, Monthly Active User)**

한 달 동안 접속한 사용자 수입입니다.

- **가동률**

한 달 동안의 각 애플리케이션 가동률을 평균 낸 값입니다.

- **애플리케이션 수**

한 달 동안 모니터링한 애플리케이션의 전체 수입입니다.

- **전체 코어**

한 달 동안의 CPU 코어 전체 개수입니다.

- **APDEX\*100(Application Performance Index)**

조회 기간 동안 애플리케이션에 대한 고객 성능 만족도입니다.

## 애플리케이션별 현황

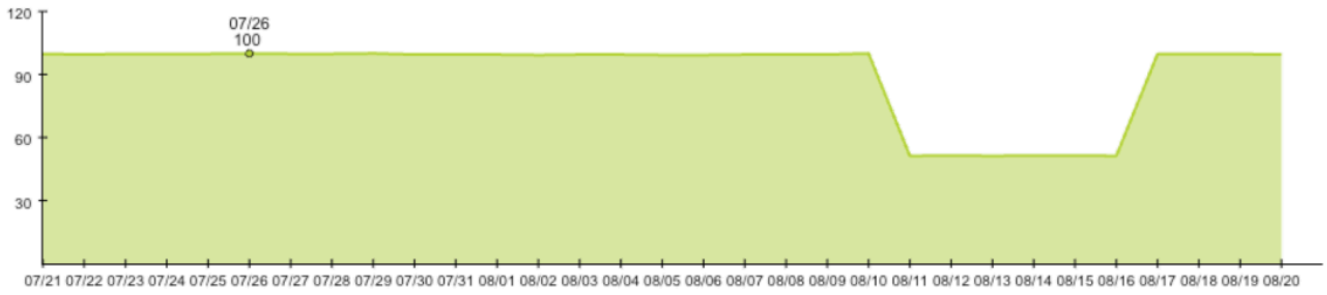
애플리케이션	IP 주소	OS 정보	가동률
demo-8100	10.21.1.20	Linux	100.0 %
demo-8101	10.21.1.20	Linux	100.0 %
demo-8102	10.21.1.20	Linux	100.0 %
demo-8103	10.21.1.20	Linux	100.0 %
demo-8104	10.21.1.20	Linux	100.0 %
demo-8105	10.21.1.20	Linux	100.0 %

모니터링 대상별로 IP 주소, OS 정보, 한 달간 가동률 평균을 확인할 수 있습니다.

## 성능 추이 차트

### 고객 충성도

CSV



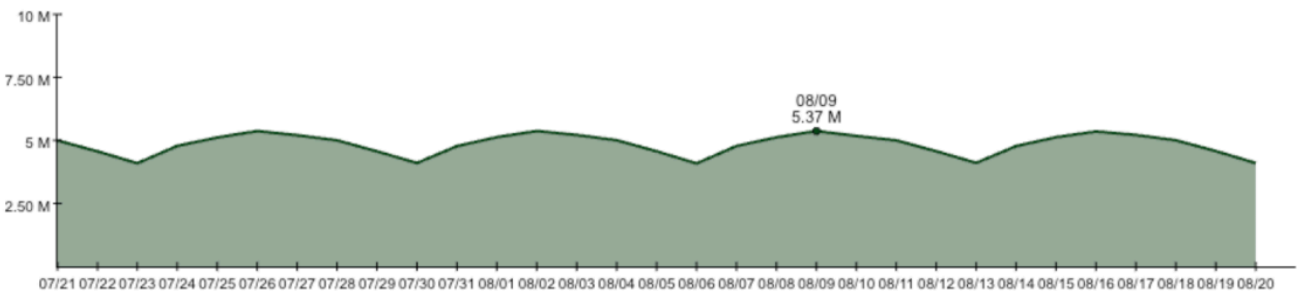
### 일간 방문자 수

CSV



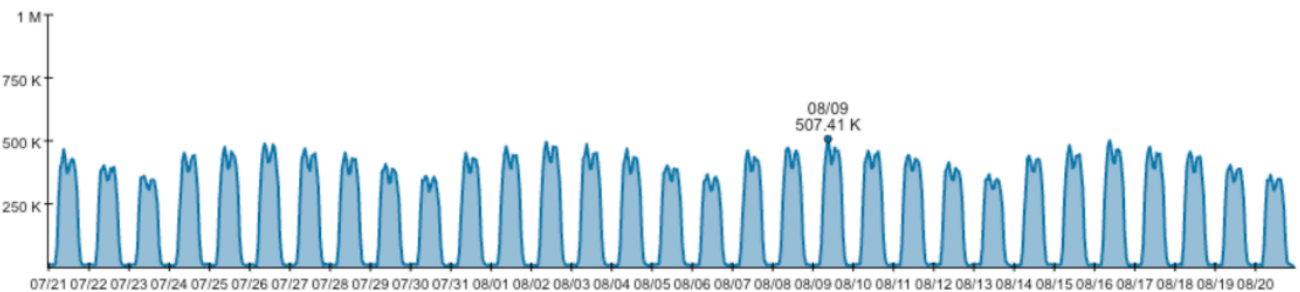
### 일간 트랜잭션 수

CSV



### 시간당 트랜잭션

CSV



한 달간 주요 지표들을 차트로 조회해 볼 수 있습니다. 월간 애플리케이션 보고서에서 차트로 그려지는 지표는 다음과 같습니다.

- **고객 충성도**

매주 사용하는 유저 중에서 얼마나 많은 유저가 매일 사용하고 있는지를 수치화 시킨 값입니다.

- **일간 방문자 수**

하루 동안 접속한 사용자 수를 나타낸 그래프입니다.

- **일간 트랜잭션 수**

하루 동안 들어온 트랜잭션 수를 나타낸 그래프입니다.

- **시간당 트랜잭션**

시간당 들어온 트랜잭션 수를 일별로 나타낸 그래프입니다.

- **시간당 에러 건수**

시간당 들어온 트랜잭션 에러 수를 일자별로 나타낸 그래프입니다.

- **액티브 트랜잭션**

시간당 들어온 액티브 트랜잭션 수를 일자별로 나타낸 그래프입니다.

- **처리량**

초당 트랜잭션 처리량을 시간별로 계산하여 일자별로 나타낸 그래프입니다.

- **시간당 평균 응답시간**

시간당 들어온 트랜잭션의 평균 응답시간을 일자별로 나타낸 그래프입니다.

- **힙 사용량**

시간당 들어온 자바 Heap 메모리 사용량을 일자별로 나타낸 그래프입니다.

- **CPU SYS 사용률**

하루 동안의 CPU SYS 사용률 %를 일자별로 나타낸 그래프입니다.

- **프로세스 CPU 사용률**

하루 동안 자바 프로세스가 사용하는 CPU %를 일자별로 나타낸 그래프입니다.

- **APDEX\*100**

하루 동안의 고객 만족도를 일자별로 나타낸 그래프입니다.

❗ **APDEX(Application Performance Index) 계산 방식**

응답 속도에 따라 **만족**, **허용**, **불만**의 3단계로 분리하여 다음과 같은 계산을 수행합니다.

ⓘ |  $APDEX*100 = (\text{만족한 사용자 수} + (\text{허용 수} * 0.5 / \text{전체 사용자 수})) * 100$

차트 우측 상단의 CSV 버튼을 이용하여 데이터를 엑셀 파일로 다운로드 할 수 있습니다.

## 트랜잭션 TOP 10

### 트랜잭션 수

번호	서비스(URL)	전월 건수	당월 건수	증감건 (증감율)
1	/account/save/employee/seoul	623,214	636,730	13,516 (▲2.12%)
2	/account/save/employee/pusan	555,604	564,675	9,071 (▲1.61%)
3	/account/save/employee/daegu	517,624	529,171	11,547 (▲2.18%)
4	/account/save/employee/kwangju	496,622	507,241	10,619 (▲2.09%)
5	/account/save/employee/jeju	479,571	489,632	10,061 (▲2.05%)
6	/account/save/employee/daejun	465,177	475,631	10,454 (▲2.2%)
7	/account/save/dept/seoul	453,229	462,275	9,046 (▲1.96%)
8	/account/save/dept/pusan	444,173	453,107	8,934 (▲1.97%)
9	/account/save/dept/daegu	435,613	443,196	7,583 (▲1.71%)
10	/account/save/dept/kwangju	426,184	435,085	8,901 (▲2.05%)

한 달 동안 들어온 트랜잭션을 트랜잭션 수를 기준으로 내림차순 정렬한 결과입니다.

- 트랜잭션 수

애플리케이션에 들어온 요청이 처리되어 응답이 반환된 수를 의미합니다. 한 달간 어떤 서비스(URL)가 가장 활발했는지를 확인할 수 있는 지표가 됩니다.

- 증감건

조회 당월과 조회 전월의 트랜잭션 수를 비교한 수치이며, 증감률은 이를 백분율로 계산한 수치입니다.

## 에러 건수

번호	서비스(URL)	전월 에러건수	당월 에러건수	증감건 (증감율)
1	/account/save/division/kwangju	392,085	399,395	7,310 (▲1.83%)
2	/product/remove/dept/daegu	161,719	164,423	2,704 (▲1.64%)
3	/order/create/dept/seoul	96,579	98,334	1,755 (▲1.78%)
4	/account/save/employee/seoul	621	644	23 (▲3.57%)
5	/account/save/employee/pusan	545	600	55 (▲9.17%)
6	/account/save/employee/daegu	503	532	29 (▲5.45%)
7	/account/save/employee/daejun	478	531	53 (▲9.98%)
8	/account/save/employee/kwangju	491	524	33 (▲6.3%)
9	/account/save/employee/jeju	492	498	6 (▲1.2%)
10	/edu/create/employee/{region}	433	454	21 (▲4.63%)

한 달 동안 들어온 트랜잭션을 에러 건수를 기준으로 내림차순 정렬한 결과입니다.

- **에러 건수**

어떤 서비스(URL)에서 에러가 많이 발생했는지를 확인할 수 있는 통계 데이터입니다.

- **증감건**

당월과 조회 전월의 에러 건수를 비교한 수치이며, 증감률은 이를 백분율로 계산한 수치입니다.

## 평균 응답 시간

번호	서비스(URL)	전월 평균 시간	당월 평균 시간	증감시간 (증감율)
1	/order/load/unit/pusan	2,317ms	2,679ms	362ms (▲13.51%)
2	/sales/save/division/jeju	2,372ms	2,529ms	157ms (▲6.21%)
3	/order/read/dept/seoul	2,284ms	2,491ms	207ms (▲8.31%)
4	/edu/write/division/pusan	2,311ms	2,487ms	176ms (▲7.08%)
5	/product/save/unit/daegu	2,350ms	2,469ms	119ms (▲4.82%)
6	/sales/save/dept/jeju	2,367ms	2,465ms	98ms (▲3.98%)
7	/product/create/employee/daegu	2,342ms	2,464ms	122ms (▲4.95%)
8	/product/create/employee/kwangju	2,338ms	2,446ms	108ms (▲4.42%)
9	/order/read/division/jeju	2,329ms	2,444ms	115ms (▲4.71%)
10	/order/pickup/employee/seoul	2,310ms	2,443ms	133ms (▲5.44%)

한 달 동안 들어온 트랜잭션을 평균 응답시간을 기준으로 내림차순 정렬한 결과입니다.

- **평균 시간**

애플리케이션 서버가 사용자에게 요청 결과를 반환하는데 걸린 시간을 의미하며, 단위는 밀리 세컨드(ms)로 표시됩니다.

- **증감 시간**

조회 당월과 조회 전월의 응답 시간을 비교한 수치이며, 증감률은 이를 백분율로 계산한 수치입니다.



## 탑 스택 TOP 10

번호	스택	전월 퍼센트	당월 퍼센트	전월 건수	당월 건수	증감건 (증감율)
1	java.base@17.0.7/java.lang.Thread.sleep(Native Method)	49.23%	33.02%	16,583,713	11,376,855	-5,206,858 (▼-31.4%)
2	java.base@17.0.7/sun.nio.ch.SocketDispatcher.read0(Native Method)	45.94%	30.75%	15,474,785	10,597,286	-4,877,499 (▼-31.52%)
3	java.base@17.0.8/java.lang.Thread.sleep(Native Method)	4.79%	16.26%	1,614,734	5,603,107	3,988,373 (▲247.0%)
4	java.base@17.0.8/sun.nio.ch.SocketDispatcher.read0(Native Method)	0.01%	15.14%	2,558	5,216,756	5,214,198 (▲203,838.8 6%)
5	app/com.virtual.web.Simula.execute3(Simula.java:175)	0.00%	4.79%	1,615	1,651,338	1,649,723 (▲102,150.0 3%)
6	java.base@17.0.7/java.lang.Thread.start0(Native Method)	0.00%	0.00%	1,258	1,587	329 (▲26.15%)
7	app/com.virtual.dao.SelectDAO.execute2(SelectDAO.java:79)	0.00%	0.00%	671	1,087	416 (▲62.0%)
8	java.management@17.0.7/sun.management.ThreadImpl.getThreadTotalCpuTime0(Native Method)	0.00%	0.00%	588	1,011	423 (▲71.94%)
9	java.base@17.0.8/java.lang.Thread.start0(Native Method)	0.00%	0.00%	479	749	270 (▲56.37%)
10	app/org.apache.commons.dbcp.BasicDataSource.getConnection(BasicDataSource.java:52)	0.00%	0.00%	426	536	110 (▲25.82%)

Stack Trace 상의 Step을 기준으로 Step 간의 호출 빈도를 비율로 계산한 결과입니다. 해당 스택에 대한 한 달 전 데이터와 조회 당월 데이터를 비교해 볼 수 있습니다.

### • 증감건

조회 당월과 조회 전월의 스택 호출 건수를 비교한 수치이며, 증감률은 이를 백분율로 계산한 수치입니다.

## SQL 평균응답시간 TOP 10

번호	SQL	전월 건수	당월 건수	전월 평균 시간	당월 평균 시간	증감시간 (증감율)
1	DELETE FROM suppliers WHERE supplier_name = '#'	3,434,407	3,506,790	107ms	108ms	1ms (▲0.93%)
2	INSERT INTO EMP VALUES (#, '#', '#', NULL, '#', #, NULL, #)	792,799	810,213	107ms	108ms	1ms (▲0.93%)
3	update table set x=# where key=#	12,368,049	12,611,925	107ms	107ms	-
4	INSERT INTO EMP VALUES (#, '#', '#', '#', '#', #, NULL, #)	8,252,804	8,410,819	107ms	107ms	-
5	DELETE FROM Customers WHERE CustomerName='#' AND ContactName = '#'	6,188,555	6,309,360	107ms	107ms	-
6	INSERT INTO EMP VALUES (#, '#', '#', '#', '#', #, #, #)	4,461,224	4,547,329	107ms	107ms	-
7	DELETE FROM posts where id=(SELECT id FROM posts GROUP BY id HAVING ( COUNT(id) > # ))	2,748,571	2,802,062	107ms	107ms	-
8	insert into emp values( #, '#', '#', #, to_date('#,##', #), null, # )	2,525,721	2,576,827	107ms	107ms	-
9	INSERT INTO DEPT VALUES (#, '#', '#')	2,221,793	2,266,991	107ms	107ms	-
10	INSERT INTO SALGRADE VALUES (#, #, #)	2,323,100	2,372,954	108ms	107ms	-1ms (▼0.93%)

SQL 쿼리문에 대해 평균 응답시간을 기준으로 정렬한 결과입니다. 각 쿼리문에 대한 월간 건수 및 평균 시간을 비교해 볼 수 있습니다.

### • 증감 시간

조회 당월과 조회 전월의 SQL 처리 평균 시간을 비교한 수치이며, 증감률은 이를 백분율로 계산한 수치입니다.

#### ! 증감률 색상별 의미

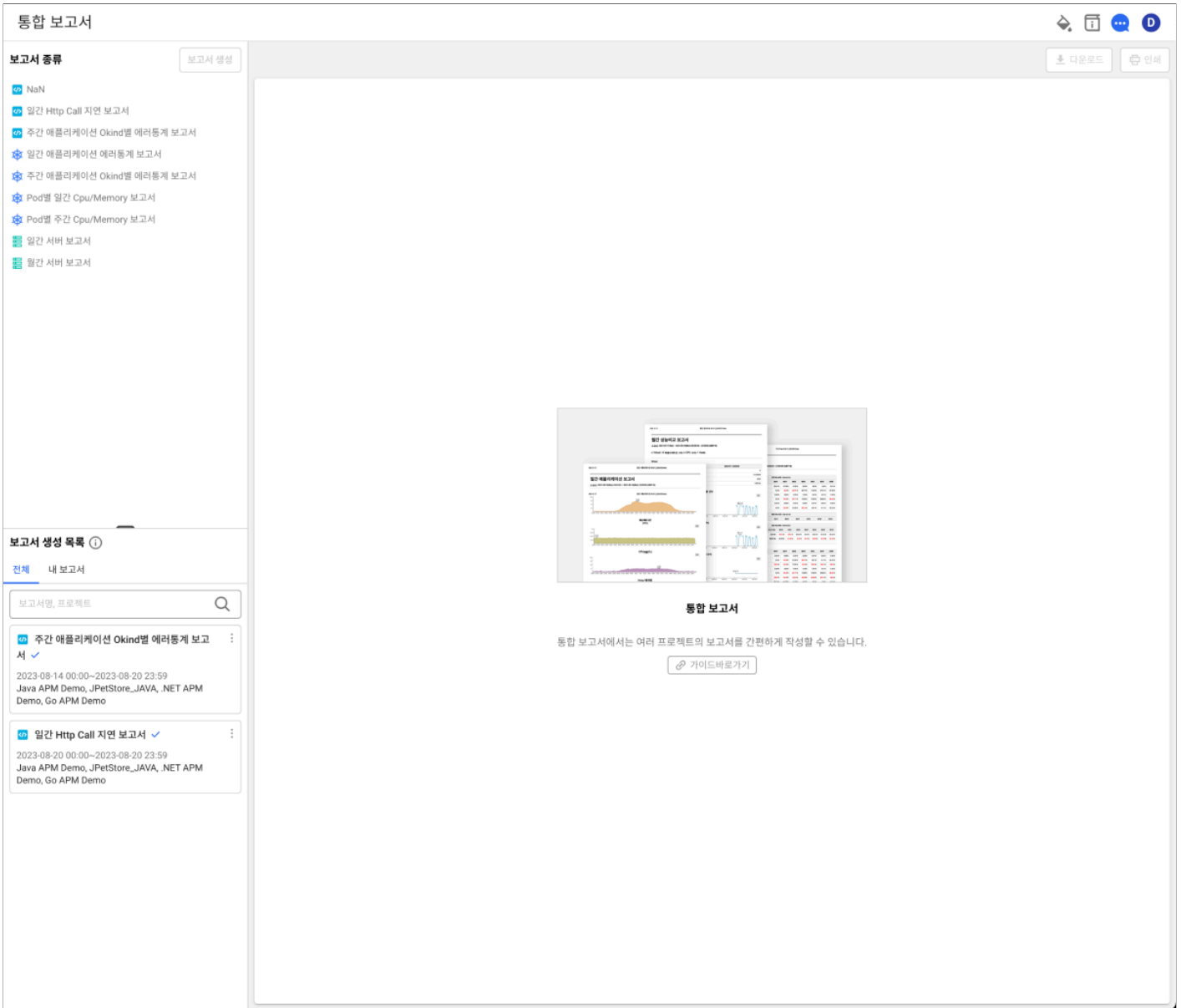
- **파란색**: 증감율이 음수인 경우로 전 주에 비해 감소했음을 의미합니다.
- **주황색**: 증감율이 0% 이상 10% 미만인 경우를 의미합니다.
- **빨간색**: 증감율이 10% 이상이거나, 전 주에는 없었던 새로운 데이터인 경우를 의미합니다.

# 통합 보고서

홈 화면 > [통합 보고서](#)

와탭 모니터링 서비스 초기 화면에서 프로젝트를 선택하지 않고 기본 [메뉴](#) 하위에 [통합 보고서](#)를 선택하세요. [통합 보고서](#)는 여러 프로젝트에 대한 보고서를 간편하게 생성할 수 있습니다.

보고서 종류와 원하는 프로젝트를 선택하여 보고서를 생성할 수 있고, 보고서 생성 목록에서 작업 진행 상태를 확인할 수 있습니다. 보고서 생성 중 특정 프로젝트에서 에러가 발생한 경우 에러 내용을 확인할 수 있습니다. 작업이 완료된 보고서를 바로 조회하거나, 다운로드 및 인쇄 기능을 통해 다른 사람에게 보고서를 공유할 수 있습니다.



통합 보고서의 기능은 다음과 같습니다.

- 멀티 프로젝트 지원

통합 보고서 메뉴에서 여러 프로젝트를 선택해 한 번에 보고서를 생성할 수 있어 편리합니다. 기존 보고서 메뉴는 프로젝트마다 보고서를 생성해야 했습니다.

- 업무 단위 보고서 생성

**보고서 생성** 메뉴에서 업무 단위로 프로젝트를 선택해 템플릿으로 저장할 수 있습니다. 보통 업무 단위로 프로젝트를 나누어 사용하는 마이크로 서비스 아키텍처(MSA, Micro Service Architecture) 환경에서 유용한 기능입니다.

- **대용량 데이터 최적화**

**통합 보고서** 메뉴는 보고서 생성 시간이 획기적으로 줄어들었습니다. 그리고 일부 프로젝트 보고서 작성 과정에서 오류가 발생하더라도 보고서 결과를 조회할 수 있습니다.

- **보고서 생성 중 동시 작업 가능**

**통합 보고서** 메뉴에서 보고서를 생성하는 동시에 다른 업무를 보는 것이 가능합니다. 기존 **보고서** 메뉴는 생성 작업이 완료될 때까지 기다려야만 했습니다.

**통합 보고서** 메뉴에서 보고서 생성을 시작한 후 다른 페이지로 이동할 수 있습니다. 또한 보고서 결과 목록에서 진행 상태와 완료된 보고서를 조회할 수 있습니다.

- **보고서 작업 공유**

동일 프로젝트 권한을 가진 사용자라면 누구든 생성한 보고서를 조회할 수 있습니다. 관련 보고서가 이미 생성됐는지 확인할 수 있어 중복으로 작성하지 않을 수 있습니다.

## 보고서와 통합 보고서의 차이

**보고서**는 하나의 프로젝트에 대한 보고서를 생성하는 메뉴입니다. 따라서 프로젝트를 선택 후 좌측의 **통계/보고서** 메뉴를 통해 접근이 가능합니다.

**통합 보고서**는 여러 프로젝트에 대한 보고서를 생성하는 메뉴입니다. 프로젝트를 선택하지 않고 기본 메뉴에서 접근이 가능합니다. **보고서**의 상세 내용은 [다음 문서](#)에서 확인할 수 있습니다.

## 보고서 생성 방법

1. 보고서 종류 선택

## 보고서 종류

보고서 생성

 **일간 Http Call 지연 보고서**

 주간 애플리케이션 Okind별 에러통계 보고서

 일간 애플리케이션 에러통계 보고서

 주간 애플리케이션 Okind별 에러통계 보고서




 Pod별 일간 Cpu/Memory 보고서

 Pod별 주간 Cpu/Memory 보고서

 일간 서버 보고서

 월간 서버 보고서

보고서 종류 메뉴에서 원하는 보고서 유형을 선택하면 **보고서 생성** 버튼이 활성화됩니다. 보고서 이름 앞의 **아이콘**은 보고서의 양식이 지원하는 프로젝트 플랫폼을 의미합니다.

-  애플리케이션 프로젝트 플랫폼을 나타냅니다.
-  쿠버네티스 프로젝트 플랫폼을 나타냅니다.
-  서버 프로젝트 플랫폼을 나타냅니다.

## 2. 상세 설정

✕
보고서 생성

타입

⌵
일간 Http Call 지연 보고서
⌵

제목

일간 Http Call 지연 보고서

시간

<
2023/08/22 00:00 ~ 2023/08/22 23:59
1일
>

---

프로젝트

🔍

▼ 프로젝트
전체 선택

⌵
[30092] .NET APM Demo

⌵
[31130] Go APM Demo

⌵
[5490] Java APM Demo

⌵
[6969] Node.js APM Demo

⌵
[31324] PHP APM Demo

⌵
[29744] Python APM Demo

보고서 양식 저장

보고서 생성

보고서 종류 선택 후 [보고서 생성](#) 버튼을 클릭하세요. 다음과 같이 상세 설정을 할 수 있습니다.

- **보고서 제목**

같은 타입의 보고서가 여러 개인 경우 이름을 지정하면 보고서 결과를 구분하기 쉽습니다.

- **시간**

보고서에 사용될 데이터의 기간을 선택하세요. 기간은 보고서 타입(일, 주, 월)에 따라 달라집니다.

- 프로젝트 선택

보고서를 작성할 프로젝트를 선택하세요. 그룹 및 개별 프로젝트 단위로 선택이 가능합니다.

- 보고서 양식 저장과 불러오기

지금 입력된 보고서의 설정을 템플릿으로 저장하고 다음 생성 시 타입 목록에서 불러올 수 있습니다. 업무 단위로 프로젝트를 자주 사용하시는 분들에게 유용한 기능입니다.

### 3. 보고서 생성

설정을 완료했다면 [보고서 생성](#) 버튼을 눌러 보고서를 생성하세요.

## 작업 목록 확인

### 보고서 생성 목록 (i)

[전체](#)    [내 보고서](#)

---

🔍

↔
**일간 Http Call 지연 보고서** ✓
⋮

2023-08-22 00:00~2023-08-22 23:59

Java APM Demo, .NET APM Demo, Go APM Demo

[보고서 생성 목록](#)에서 이전에 만들었던 보고서와 현재 작업 중인 보고서를 확인할 수 있습니다.

[전체](#) 목록에서는 내가 생성한 보고서 외에도 나와 같은 권한을 가진 사람이 생성한 보고서를 함께 볼 수 있습니다. [내 보고서](#) 목록에서는 내가 생성한 보고서만 필터링하여 볼 수 있습니다.



에러 없이 생성된 보고서는 ✓ **체크** 아이콘으로 표시되며, 에러가 포함된 경우는 △ **경고** 아이콘이 함께 표시됩니다. △ **경고** 아이콘을 클릭하시면 어떤 프로젝트에서 에러가 발생했는지 확인할 수 있습니다.

## 조회 및 공유

↓ 다운로드
☰ 인쇄

---

### 일간 Http Call 지연 보고서

조회조건 :2023-08-22(Tue) 00:00:00 ~ 2023-08-22(Tue) 23:59:59 (GMT+9)

[5490] Java APM Demo

---

### 일간 Http Call 지연 보고서

조회조건 2023-08-22(Tue) 00:00:00 ~ 2023-08-22(Tue) 23:59:59 (GMT+9)

**Http Call 지연 Top 5 (1초 이상)**

번호	URL	호출 URL	호스트	포트	전체	에러	평균 시간(ms)	합계 시간(ms)	최대 시
1	/remote/account/save/dept/kwangj	/account/load/dept/daejun	127.0.0.1	8105	2264	0	1247	2823259	
	u								

보고서는 3가지 방식으로 제공됩니다.

- **화면 조회**

작업 목록에서 생성한 보고서를 선택하면 화면에서 바로 조회하여 볼 수 있습니다.

- **다운로드**

[다운로드](#) 버튼을 클릭하면 바로 보기의 보고서와 같은 파일이 html 형식으로 다운로드 됩니다. 모니터링 관리 권한이 없는 사람에게 보고서를 공유할 수 있습니다.

- **인쇄 및 PDF 저장**

[인쇄](#) 버튼을 누르면 컴퓨터에 연결된 프린터로 인쇄하거나 또는 PDF 형식으로 저장할 수 있습니다.

# 알림 설정하기

이벤트 조건을 설정하고 이메일, SMS, 메신저, App Push 등 다양한 경로로 알림을 수신할 수 있습니다.

- 에이전트와 수집 서버의 통신 두절 및 애플리케이션 자원 사용률, 액티브 트랜잭션, 에러, 응답 지연, 응답 분포 이상에 대한 알림 설정이 가능합니다.
- 애플리케이션 알림에서 수용하기 어려운 조건의 알림 설정은 태그 카운트 알림 기능을 통해 제공합니다.
- 머신 러닝 기반 AI 알림 설정을 통해 히트맵 패턴을 감지해 알림 기능을 제공합니다. AI 모듈이 애플리케이션 실행 분포 패턴을 판단해 알림을 보냅니다. 복잡한 조건 없이 이슈 상황을 빠르고 정확하게 인지할 수 있습니다.
- [이벤트 설정](#) 메뉴를 통해 임계치 기반 알림 조건 및 수신 설정을 할 수 있습니다.
- [이벤트 수신 설정](#) 메뉴에서는 개인별 알림 수신 채널 및 시간대를 설정할 수 있습니다.
- [이벤트 기록](#) 메뉴에서는 알림 발생 기록을 조회할 수 있습니다.

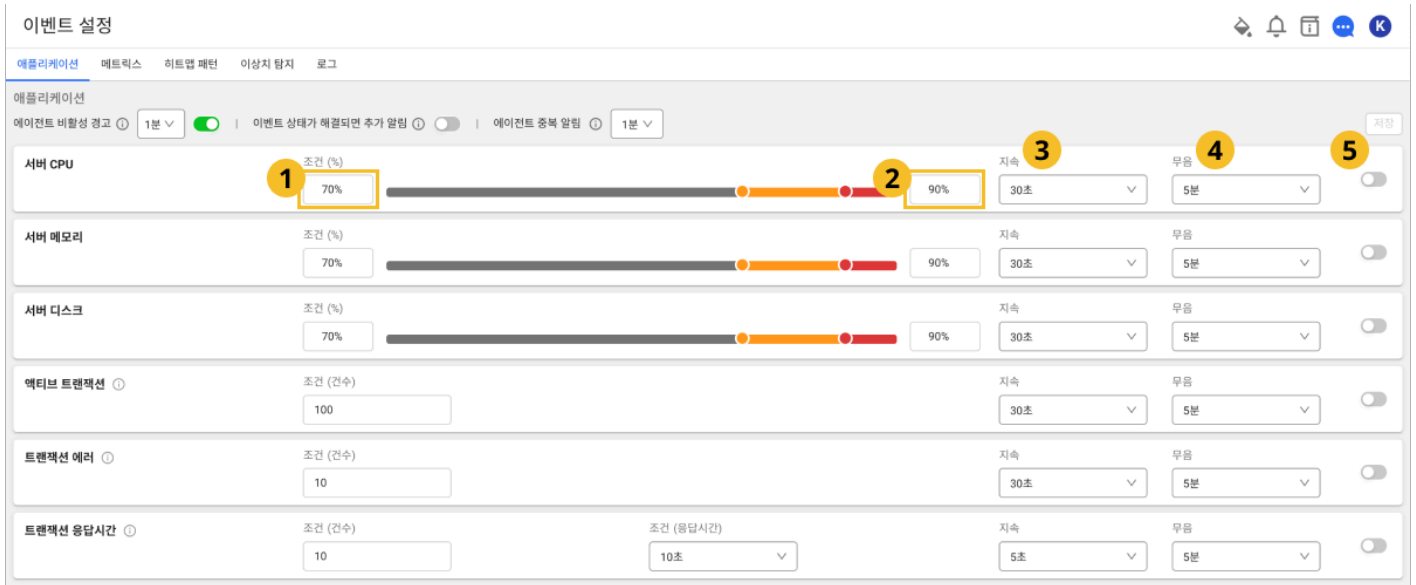
다음은 애플리케이션 모니터링에서 경고 알림에 대한 동영상 소개입니다.

- ① • 다른 계정에 대한 알림을 설정하려면 [수정](#), [알림 설정](#), [멤버 관리](#) 권한 중 하나가 있어야 합니다.
  - 이벤트를 추가하거나 설정하려면 알림 설정 권한이 있어야 합니다. 권한 체계에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.
  - 경고 알림과 관련해 모니터링 플랫폼별 지원되는 이벤트 종류를 확인하려면 [다음 문서](#)를 참조하세요.

# 애플리케이션 경고 알림

홈 화면 > 프로젝트 선택 > 경고 알림 > 이벤트 설정 > 애플리케이션 탭 선택

애플리케이션의 자원 사용량과 진행 중 트랜잭션, 에러 상황의 이벤트 조건을 통해 경고 알림을 보냅니다.



## 기본 옵션

다음은 각 항목별 설정할 수 있는 기본 옵션입니다.

- 에이전트 비활성화 경고

설정된 시간 동안 에이전트로부터 데이터를 수신할 수 없을 경우 경고 알림을 보냅니다. 토글 버튼을 선택해 기능을 켜거나 끌 수 있습니다.

- 이벤트 상태가 해결되면 추가 알림

애플리케이션 이벤트 항목 중 발생한 이벤트 상태가 해결되면 추가 알림 송신 여부를 선택할 수 있습니다. 토글 버튼을 선택해 기능을 켜거나 끌 수 있습니다.

- 에이전트 중복 알림

모니터링 중인 에이전트 가운데 동일한 ID를 설정한 에이전트가 2개 이상일 때 지표를 중복으로 수집하면서 발생하는 경고 알림입니다. 에이전트 이름을 중복으로 설정하지 않았는지 에이전트 설정에서 관련 옵션을 확인하세요. 블루/그린과 같은 배포

상황에서 에이전트가 중복되어 경고 알림이 발생할 경우 이 기능을 이용하길 권장합니다. 같은 이름의 에이전트가 동시에 2개 이상 기동된 경우 선택한 시간 동안 알림을 받지 않습니다.

- **조건**
  - ① 조건은 **경고(Warning)** 레벨, ② 조건은 **위험(Critical)** 레벨입니다.
- **③ 지속**

선택한 시간 동안 이벤트 조건이 지속될 경우 알림을 보냅니다.
- **④ 무음**

알림을 보낸 후 선택한 시간 동안 같은 알림을 보내지 않습니다.
- **⑤ On / Off**

토글 버튼을 선택해 발생하는 이벤트를 켜거나 끌 수 있습니다.

## 항목별 이벤트 설정

다음의 이벤트 항목을 설정할 수 있습니다.

- **서버 CPU, 서버 메모리, 서버 디스크**

각 항목의 사용량이 설정한 **조건(%)**에 설정한 기준치를 도달하면 알림을 보냅니다.
- **액티브 트랜잭션**

액티브 트랜잭션 발생 건수가 **조건(건수)**에 설정한 기준치를 넘은 상태를 지속하면 알림을 보냅니다. 발생 건수 기본값은 100개 기준입니다.
- **트랜잭션 에러**

정상적으로 수행하지 않은 트랜잭션을 의미합니다. 에러 건수가 **조건(건수)**에 설정한 기준치를 넘은 상태를 지속하면 알림을 보냅니다. 수집 주기는 5초입니다.
- **트랜잭션 응답시간**

트랜잭션이 끝난 시간으로 확인합니다. 예를 들어 **10초의 응답시간** 기준을 넘는 트랜잭션의 **건수가 10개**를 초과하고, 이 상태가 **5초간 지속**되면 알림을 보냅니다.

# 이벤트 설정

홈 화면 > 프로젝트 선택 >  사이트맵 > 경고 알림 > 이벤트 설정 New

새로운 **이벤트 설정** 메뉴는 카테고리(Category)가 아닌 필드(Field) 중심의 사용자 경험을 강화한 메뉴입니다. 매트릭스 이벤트 설정과 같은 강력한 기능을 모두 필요치 않거나 단순한 알림 조건을 반복해서 사용하는 사용자에게 빠르게 경고 알림을 적용할 수 있는 이벤트 기능입니다. 생성한 프로젝트의 상품에 적합한 기본 이벤트 템플릿을 제공하여 빠르고 쉽게 원하는 경고 알림 이벤트를 설정할 수 있습니다.

카테고리가 아니라 필드를 먼저 선택해 사용자가 인지하는 지표와 이벤트 설정의 불일치를 해소합니다. 시뮬레이션이 주는 시각적 직관성을 결합하여 빠르고 정확하게 이벤트 설정을 완료할 수 있습니다.

- ① 신규 프로젝트는 별도의 설정 없이 이 기능을 이용할 수 있으나 이미 생성된 프로젝트에서는 **이벤트 템플릿 생성** 버튼을 선택하세요. 새로운 기능에 최적화된 이벤트 목록을 자동으로 생성합니다.
- 상품에 따라 제공하는 이벤트 템플릿은 다를 수 있습니다.
- 이 기능은 알림 설정 권한이 있는 멤버만 이용할 수 있습니다. 멤버 권한에 대한 자세한 설명은 [다음 문서](#)를 참조하세요.

## 기본 이벤트 템플릿

생성한 프로젝트의 상품에 따라 기본 이벤트 템플릿을 제공합니다. 이벤트 목록의 가장 왼쪽에 **활성화** 버튼을 선택해 원하는 이벤트를 활성화할 수 있습니다.

활성화	수정	이벤트 이름 ↑	지표 이름	지표	대상	연속 ①	일시 중지	해결된 알림	이벤트 수신
<input type="checkbox"/>		Active Transaction	Active Transaction (count)	<span style="color: orange;">●</span> > 100	전체	사용 안함	5분	사용	전체 수신
<input type="checkbox"/>		Server CPU	Server CPU (%)	<span style="color: orange;">●</span> > 70 <span style="color: red;">●</span> > 90	전체	사용 안함	5분	사용	전체 수신
<input type="checkbox"/>		Server Disk	Server Disk (%)	<span style="color: orange;">●</span> > 70 <span style="color: red;">●</span> > 90	전체	사용 안함	5분	사용	전체 수신
<input type="checkbox"/>		Server Memory	Server Memory (%)	<span style="color: orange;">●</span> > 70 <span style="color: red;">●</span> > 90	전체	사용 안함	5분	사용	전체 수신
<input type="checkbox"/>		Transaction Error Count	Transaction Error Count (count)	<span style="color: orange;">●</span> > 10	전체	사용 안함	5분	사용	전체 수신
<input type="checkbox"/>		Transaction Response Time	Transaction Response Time (ms)	<span style="color: orange;">●</span> > 2,000	전체	10회	5분	사용	전체 수신

제공하는 이벤트 템플릿은 다음과 같습니다. **지표** 항목에 설정된 기본값은 사용자가 원하는 값으로 수정할 수 있습니다.

- **Active Transaction**

카테고리:

액티브 트랜잭션 발생 건수( `active_tx_count` )가 100개를 초과한 상태하면 경고(Warning) 수준의 알림을 보냅니다.

- Server CPU

카테고리: `app_host_resource`

호스트의 CPU 사용률( `cpu` )이 70%를 초과하면 경고(Warning) 수준의 알림을, 90%를 초과하면 위험(Critical) 수준의 알림을 보냅니다.

- Server Disk

카테고리: `app_host_resource`

호스트의 디스크 사용률( `disk` )이 70%를 초과하면 경고(Warning) 수준의 알림을, 90%를 초과하면 위험(Critical) 수준의 알림을 보냅니다.

- Server Memory

카테고리: `app_host_resource`

호스트의 메모리 사용률( `mem` )이 70%를 초과하면 경고(Warning) 수준의 알림을, 90%를 초과하면 위험(Critical) 수준의 알림을 보냅니다.

- Transaction Error Count

카테고리: `app_counter`

트랜잭션 에러 건수( `tx_error` )가 10개를 초과하면 경고(Warning) 수준의 알림을 보냅니다.


- Transaction Response Time

카테고리: `app_counter`

트랜잭션의 평균 응답 시간( `resp_time` )이 10회 연속, 2,000ms(2초)를 초과한 상태로 지속되면 경고(Warning) 수준의 알림을 보냅니다.

❗ 기본 이벤트 템플릿에 적용된 필드에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

## 이벤트 수정하기

기본 이벤트 템플릿에 적용된 지표값을 수정해 경고 알림이 발생 기준을 변경할 수 있습니다. 그 외에도 다양한 옵션 설정을 통해 알림 발생 기준을 변경할 수 있습니다. 수정하려는 이벤트 항목에서  버튼을 선택하면 이벤트 수정 화면으로 이동합니다. 다음의 항목을 차례로 수정한 다음 [저장](#) 버튼을 선택하세요.

## 이벤트 조건 정의

이벤트 발생 조건 기준을 설정할 수 있습니다.

**이벤트 조건 정의**

< 2024/02/06 10:46 ~ 2024/02/06 11:46 60분 > 실행

①

조건 설정 후 실행 버튼을 누르면 이벤트를 시뮬레이션할 수 있습니다.

**지표 설정 \***  Warning  Critical

Warning > 70 / Critical > 90

**연속**  사용 안함  연속 1 회 발생 (1분에 최대 N회 발생)

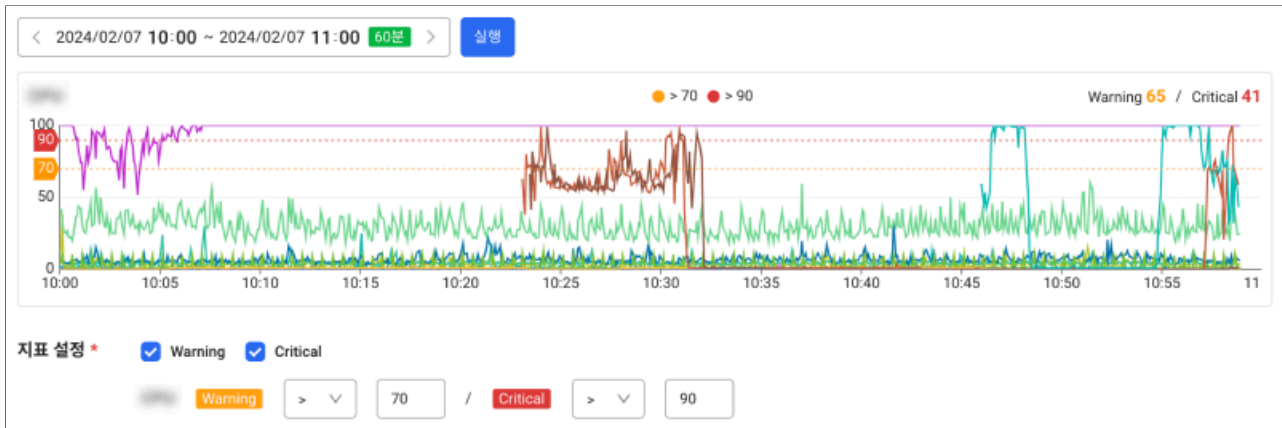
**일시 중지** 1분

알림 수신 후 선택한 시간 동안 이벤트가 발생하지 않습니다.  
단, "이벤트 상태가 해결되면 추가 알림" 기능을 활성화한 경우에는 RECOVERED 알림 수신 후 선택한 시간 동안 이벤트가 발생하지 않습니다.

**해결된 알림**

Critical과 Warning 레벨의 이벤트가 해결되면 RECOVERED 상태의 알림을 수신합니다.

- **실행:** 이벤트 조건을 설정한 다음 버튼을 선택하면 설정한 이벤트를 시뮬레이션할 수 있습니다.



- **지표 설정:** Warning 또는 Critical 알림에 대한 임계값을 설정할 수 있습니다. 연산자를 설정하고 임계값을 입력하세요.
  - 연산자 종류: > (보다 큼), >= (보다 크거나 같음), < (보다 작음), <= (보다 작거나 같음), == (같음)

- 임꺽값으로 양수, 음수, 0을 입력할 수 있지만, 선택한 이벤트 템플릿에 따라 음수 입력 여부는 다를 수 있습니다.

ⓘ ◦ 제공되는 이벤트 템플릿에 따라 설정할 수 있는 알림 수준(Level)은 다를 수 있습니다.

- **Warning** 또는 **Critical** 수준 하나만 선택할 수 있지만, 두 항목을 모두 비활성화할 수는 없습니다.

- **연속**: 설정한 이벤트 조건을 충족하는 횟수에 따라 알림을 보낼 수 있습니다.
  - **사용 안함**: 이벤트가 발생할 때마다 알림을 보냅니다.
  - **연속**: 입력한 횟수만큼 이벤트가 발생하면 알림을 보냅니다.
- **일시 중지**: 알림 수신 후 선택한 시간 동안 이벤트가 발생하지 않습니다.
- **해결된 알림**: 이벤트가 해결되면 **RECOVERED** 상태의 알림을 보냅니다.

✔ 다음 조건에 따라 알림을 보내는 기준이 달라집니다.

연속	해결된 알림	동작
사용 안함	<b>On</b>	설정된 이벤트 조건을 충족하거나 해결되면 알림을 보냅니다.
	<b>Off</b>	설정된 조건을 충족할 때마다 알림을 보냅니다.
연속 N 회 발생	<b>On</b>	N 회 이상 조건을 충족하거나 N 회 이상 이벤트가 해결되면 알림을 보냅니다.
	<b>Off</b>	N 회 이상 설정한 이벤트 조건을 충족하면 알림을 보냅니다. 알림을 보낸 후 횟수를 초기화해 횟수를 다시 집계합니다.

## 이벤트 대상 선택

특정 에이전트에서만 발생하는 이벤트를 알림으로 보내도록 설정할 수 있습니다. **에이전트 (인스턴스)** 항목을 클릭한 다음 원하는 에이전트를 선택하세요.



▼ 2 이벤트 대상 선택

대상 선택 전체 대상이 포함됩니다.

에이전트 (인스턴스)

위 항목들을 설정하지 않으면 프로젝트에 포함된 전체 에이전트를 대상으로 이벤트가 실행되며 많은 알림이 발생할 수 있습니다.

이벤트 대상에 대한 자세한 설명은 가이드 문서를 참고하세요. [가이드 문서 >](#)

이벤트를 시뮬레이션하려면 화면 최상단에 '이벤트 조건 정의' 항목에서 [실행 버튼](#)을 선택하세요.

- ① • 제공되는 템플릿에 따라 선택할 수 있는 대상은 다를 수 있습니다.
- 선택한 대상에 대한 이벤트 발생 현황을 확인하려면 [이벤트 조건 정의](#)에서 [실행](#) 버튼을 선택하세요.
- 대상을 선택하지 않으면 프로젝트에 포함된 전체 에이전트를 대상으로 알림을 보냅니다. 전체 에이전트를 대상으로 이벤트가 실행되면 많은 알림이 발생할 수 있습니다.

## 기본 정보 및 수신 설정

알림으로 수신되는 이벤트의 제목과 메시지를 작성하고, 수신 대상을 선택할 수 있습니다.

▼ 3 기본 정보 및 수신 설정

이벤트 활성화

이벤트 이름 \*

메시지 \*

\$를 입력하면 변수를 입력할 수 있습니다.  
ex) \$(active\_tx\_8) \$(time)에 예러가 발생했습니다.

수신 테스트

이벤트 수신  전체 수신  태그 선택 수신

- **이벤트 활성화:** 현재 이벤트를 활성화할 수 있습니다.
- **이벤트 이름:** 기본 제공된 템플릿의 이벤트 이름이 입력되어 있습니다. 사용자가 원하는 이벤트 이름으로 편집할 수 있습니다.
- **메시지:** 기본 제공된 템플릿의 메시지가 입력되어 있습니다. 사용자가 원하는 메시지로 편집할 수 있습니다. 🔄 버튼을 클릭하면 이전에 입력한 메시지 기록을 확인할 수 있습니다.

✓ 메시지 입력 창에 `#{Tag}` 또는 `#{Field}` 변수를 입력해 메시지를 작성할 수 있습니다.  
 분석 > 매트릭스 조회 메뉴에서 **카테고리**를 선택한 다음 입력할 수 있는 `#{Tag}` 또는 `#{Field}` 변수를 확인하세요.  
 현재 이벤트 템플릿의 **카테고리** 이름은 [다음 문서](#)의 **카테고리** 항목을 참조하세요.

- **수신 테스트**: 현재 이벤트로 발생하는 알림 수신 시 입력한 이벤트 이름과 메시지를 사전 점검할 수 있습니다. 필수 항목([지표 설정](#), [이벤트 이름](#), [메시지](#))을 모두 입력해야 테스트할 수 있습니다.

ⓘ 테스트 중에는 실제 매트릭스 값이나 변수에 대한 치환 기능이 작동하지 않으며, 수신자 태그가 설정된 사용자에게만 알림을 보낼 수 없습니다.

- **이벤트 수신**: 현재 이벤트로 발생하는 알림을 수신할 멤버를 선택할 수 있습니다.
  - **전체 수신**: 프로젝트에 소속된 멤버 전원에게 알림을 보냅니다.
  - **태그 선택 수신**: 선택한 태그를 가진 프로젝트 멤버와 3rd-party 플러그인에 알림을 보냅니다. [태그 추가](#) 또는 + 버튼을 클릭해 태그 목록에서 원하는 태그를 선택하세요.

ⓘ [경고 알림](#) > [이벤트 수신 설정](#) 메뉴에서 프로젝트 멤버와 3rd-party 플러그인에 태그를 설정할 수 있습니다. 자세한 내용은 [다음 문서](#)를 참조하세요.

## 이벤트 추가하기

기본 제공되는 이벤트 템플릿을 편집해 사용자가 원하는 알림을 추가할 수 있습니다.

1. 화면 오른쪽 위에 [이벤트 추가](#) 버튼을 선택하세요.
2. 템플릿 목록에서 추가하려는 이벤트 항목을 선택하세요.
3. [이벤트 대상 선택](#) 및 [이벤트 대상 선택](#), [기본 정보 및 수신 설정](#) 섹션의 옵션을 차례로 설정하세요.
4. 모든 설정을 완료한 다음 화면 오른쪽 위에 [저장](#) 버튼을 선택하세요.


이벤트 목록에서 새로 추가한 이벤트를 확인할 수 있습니다.

ⓘ • 기본 제공되는 이벤트 템플릿과 구별하여 사용하려면 [이벤트 이름](#) 항목을 편집해 저장하세요.

- ① 기본 제공하는 이벤트 템플릿에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.
- 이벤트 조건 설정의 각 섹션에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

## 이벤트 삭제하기


기본 제공된 이벤트 또는 새로 추가한 이벤트를 삭제할 수 있습니다.

- 이벤트 목록에서 삭제하려는 이벤트 항목에서  버튼을 선택하세요.
- 이벤트 수정 화면으로 이동하면 화면 오른쪽 위에 **삭제** 버튼을 선택하세요.
- 삭제 확인 메시지가 나타나면 다시 한번 **삭제** 버튼을 선택하세요.

- ① 삭제한 이벤트는 복구할 수 없습니다.

## JSON 형식으로 수정하기

이벤트 설정을 JSON 형식으로 수정할 수 있습니다.

- 화면 오른쪽 위에 **JSON**  버튼을 선택하세요.
- 편집 창이 나타나면 JSON 형식에 맞춰 내용을 수정하세요.
- 수정을 완료하면 화면 오른쪽 위에 **저장** 버튼을 선택하세요.

- ① 수정한 내용이 JSON 형식에 맞지 않으면 화면 아래에 에러 메시지가 표시되며, 저장할 수 없습니다. 표시되는 에러 메시지는 형식에 따라 다를 수 있습니다.

Expected ', or }' after property value in JSON at position 1964 (line 80 column 42)

JSON 데이터의 구조는 다음과 같습니다.

```
{
```

```

"metaId": "infra001",
"displayName": "CPU",
"stateful": true,
"selectCondition": {},
"warningEnabled": true,
"criticalEnabled": true,
"receiver": [],
"warningThreshold": "cpu > 70",
"criticalThreshold": "cpu > 90",
"repeatCount": 1,
"silentPeriod": 60000,
"enabled": false,
"message": "CPU = ${cpu}"
},

```

JSON 데이터의 필드는 이벤트 설정에서 다음 옵션 항목과 연결됩니다.



JSON 필드	옵션
metaId	사용자가 선택한 템플릿의 고유 식별자 값
displayName	<a href="#">이벤트 이름</a>
stateful	<a href="#">해결된 알림</a>
selectCondition	<a href="#">대상 선택</a>
warningEnabled	<b>Warning</b> 이벤트 활성화 여부
criticalEnabled	<b>Critical</b> 이벤트 활성화 여부
receiver	<a href="#">이벤트 수신</a> > <a href="#">태그 선택 수신</a> 옵션의 수신 태그 키값 목록
warningThreshold	<b>Warning</b> 이벤트의 임계값 설정
criticalThreshold	<b>Critical</b> 이벤트의 임계값 설정

JSON 필드	옵션
repeatCount	연속 N 회 발생
silentPeriod	일시 중지
enabled	이벤트 활성화
message	메시지

## JSON 파일로 공유하기


이벤트 설정을 JSON 파일로 저장해 다른 사용자와 설정을 공유하거나 다른 사용자의 설정을 가져올 수 있습니다.


### 내보내기

1. 화면 오른쪽 위에 **JSON**  버튼을 선택하세요.
2. JSON 편집 창이 나타나면 **내보내기**  버튼을 선택하세요.
3. JSON 파일이 다운로드되면 공유할 다른 사용자에게 전달하세요.

 JSON 파일 이름은 event-rules---.json 형식입니다.

### 가져오기

1. 화면 오른쪽 위에  버튼을 선택하세요.
2. **내보내기** 기능을 통해 다운로드한 JSON 파일을 선택하세요.
3. JSON 편집 창이 나타나면 **목록에 추가하기** 또는 **덮어쓰기** 버튼을 선택하세요.

 이 기능은 같은 종류의 상품 간에 이용할 것을 권장합니다. 다른 상품의 프로젝트로부터 이벤트 설정을 가져올 수는 있지만 정상 작동하지 않습니다.

## 이벤트 검색하기

이벤트 목록에서 이벤트 이름 또는 지표를 기준으로 검색할 수 있습니다. 검색 입력란에 문자열을 입력한 다음 🔍 버튼을 선택하세요.

# 메트릭스 경고 알림

홈 화면 > 프로젝트 선택 > 경고 알림 > 이벤트 설정 > 메트릭스 탭 선택

## 메트릭스 이벤트란?

메트릭스 이벤트는 기본 이벤트(애플리케이션 이벤트, 서버 이벤트 등)보다 구체적이고 복잡한 이벤트를 설정할 때 사용합니다. 프로젝트에서 실시간으로 수집 중인 메트릭스 데이터를 기반으로 이벤트를 설정할 수 있습니다. 사용에 따라 두 가지 설정 방법 중 하나를 선택해 이벤트를 설정할 수 있습니다.

- 메트릭스 이벤트
- 복합 메트릭스 이벤트

ⓘ 메트릭스에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

## 메트릭스 이벤트

[경고 알림](#) > [이벤트 설정](#) 메뉴에서 화면 위에 [메트릭스](#)를 선택하세요. 화면 오른쪽 위에 [이벤트 추가](#)를 선택하세요. [메트릭스 이벤트](#) 창이 나타납니다.

× 메트릭스 이벤트

---

**이벤트명 \***

**이벤트 활성화**

---

**템플릿**

**카테고리 \***

**레벨 \***

**메시지 \***

**수신 테스트**

Critical

Warning

Info

이벤트 상태가 해결되면 추가 알림

🔄

ex. \${active\_tx\_8} \${time}에 에러가 발생했습니다.

수신 테스트를 이용하려면 필수 항목(이벤트 명, 카테고리, 레벨, 메시지)에 값을 입력하거나 선택하세요.

---

**발생 조건 & 대상 설정**

이벤트 발생조건과 대상 입력시 특수문자(~!@#%\*&\*()\_+=[])가 포함되거나 숫자로 시작하는 필드명을 입력하면 오류가 발생할 수 있습니다.  
직접 입력 옵션을 선택한 다음 예시와 같이 중괄호({})로 묶어서 입력하세요.

**이벤트 발생 조건 \***

선택 입력
  직접 입력

필드를 선택해 주세요.

>

값

+ 추가

**이벤트 대상 필터링**

선택 입력
  직접 입력

태그를 선택해 주세요.

>

값

+ 추가

▶ 이벤트 수신 설정

▶ 알림 규칙 테스트

## 기본 정보 입력

- **이벤트명**: 추가하려는 이벤트 이름을 입력하세요.



- **이벤트 활성화:** 이벤트를 활성화 여부를 선택하세요.
- **템플릿:** 만들어진 템플릿을 선택해 빠르고 쉽게 이벤트를 설정할 수 있습니다. 템플릿을 사용하지 않을 경우 **사용 안 함**을 선택하세요.

❗ 템플릿 목록에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

- **카테고리:** 메트릭스 데이터를 구분하는 단위입니다. 메트릭스 이벤트 설정 시 필수 선택 값입니다.

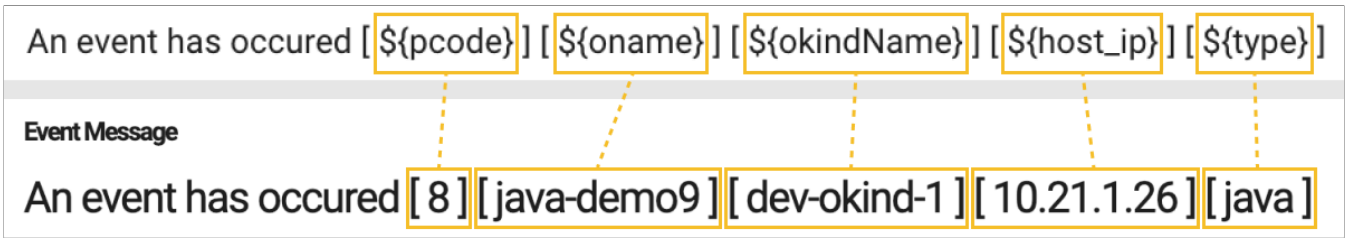
The screenshot shows a form with the following fields:
 

- 카테고리 \***: A dropdown menu with the placeholder text "카테고리를 선택해 주세요." and a search icon.
- 레벨 \***: A search bar with the text "검색" and a magnifying glass icon.
- 메시지 \***: A section with a "직접 입력하기" button.
- 수신 테스트**: A section with a "직접 입력하기" button.

 The dropdown menu is open, displaying a list of categories:
 

이름	키
AlertCount	alert_count
AlertCount 1시간	alert_count(h1)
AlertCount 5분	alert_count(m5)

- **카테고리** 선택 옵션에는 ① 이름과 ② 데이터 수집 간격, ③ 키 정보를 표시합니다. 이벤트 설정 시 해당 카테고리의 키 값을 사용합니다.
  - **카테고리**는 최근 3시간 범위 내 프로젝트에서 수집 중인 메트릭스 데이터를 조회해 목록에 표시합니다. **카테고리** 선택 옵션에 수집 간격이 표시되지 않는 경우 **직접 입력하기** 옵션을 선택해 카테고리 키를 입력할 수 있습니다.
- **레벨**
- 이벤트 발생 시 경고 수준을 나타냅니다. **Critical**, **Warning**, Info 수준으로 나눕니다. **Critical**, **Warning** 레벨 설정 시 **이벤트 상태가 해결되면 추가 알림** 선택 옵션이 활성화됩니다.
  - **이벤트 상태가 해결되면 추가 알림:** 이벤트 항목 중 발생한 이벤트 상태가 해결되면 추가 알림 송신 여부를 선택할 수 있습니다. 토글 버튼을 선택해 기능을 켜거나 끌 수 있습니다.
- **메시지**
- 이벤트 발생 시 출력하는 알림 메시지를 입력합니다. `{Tag}` 또는 `{Field}` 입력으로 메시지에 변수를 적용할 수 있습니다. 변수에 입력할 키는 선택한 메트릭스 데이터 **카테고리**에 포함된 값이어야 합니다. **메트릭스 조회** 메뉴에서 입력할 수 있는 태그 또는 필드키를 확인할 수 있습니다.



ⓘ 버튼을 클릭하면 이전에 입력한 메시지 기록을 확인할 수 있습니다.

수신 테스트

필수 항목인 이벤트명, 카테고리, 레벨, 메시지 정보를 기준으로 알림을 발생시켜 메시지를 점검하는 기능입니다.

ⓘ 수신 테스트를 이용하려면 필수 항목(이벤트명, 카테고리, 레벨, 메시지)에 값을 입력하거나 선택해야 합니다.

이벤트 발생 조건

이벤트 발생 조건 \*

선택 입력  직접 입력

4 cpu > 70 && mem > 80 5 6

cpu	Cpu (pct) ▾	>	70
&& ▾	mem Memory (pct) ▾	>	80

+ 추가

4 필드, 5 연산자 선택, 6 임계값을 입력해 이벤트 발생 조건을 설정하세요.

이벤트 대상 필터링

이벤트 대상 필터링

선택 입력  직접 입력

7 태그를 선택해 주세요. ▾ 8 > 9 값

+ 추가

7 태그, 8 연산자 선택, 9 필터링값을 입력해 대상을 필터링합니다. 입력값이 없을 경우 전체 에이전트를 대상으로 경고 알림을 보냅니다.

ⓘ

- 이벤트 발생 조건과 이벤트 대상 필터링에서 사용할 수 있는 기본 문법과 연산자 목록은 [다음 문서](#)를 참조하세요.
- 이벤트 발생 조건과 이벤트 대상 필터링 옵션은 **선택 입력** 또는 **직접 입력** 옵션을 선택할 수 있습니다.

- ① 이벤트 설정 내용이 저장된 이후에는 해당 옵션값은 **직접 입력** 옵션으로 관리합니다. 이후 **선택 입력** 옵션으로 전환하면 옵션값이 초기화될 수 있습니다.
- 이벤트 발생 조건과 대상 입력 시 특수 문자( `~!@#%&^&*( )_+=[ ]` )를 포함하거나 숫자로 시작하는 필드명을 입력하면 오류가 발생할 수 있습니다. 이런 경우 **직접 입력** 옵션을 선택한 다음 예시와 같이 중괄호( `{ }` )로 묶어서 입력하세요.

```

    ${4xxErrorType} == '401'
```

## 이벤트 수신 설정

**▼ 이벤트 수신 설정**

**발생 횟수**      최근  동안  회 발생

선택 시간 동안 설정한 이벤트가 입력 횟수만큼 발생할 때 알림을 수신합니다.  
 선택 시간이 "사용 안 함"인 경우에는 지정한 횟수만큼 연속적으로 발생할 때 알림을 수신합니다.  
 "이벤트 상태가 해결되면 추가 알림"을 사용하는 경우, 선택 시간은 "사용 안 함"을 권장합니다.  
 선택한 카테고리의 수집 주기는 5초입니다.

**이벤트 발생 일시 중지**     

알림 수신 후 선택한 시간 동안 이벤트가 발생하지 않습니다.  
 단, "이벤트 상태가 해결되면 추가 알림" 기능을 활성화한 경우에는 RECOVERED 알림 수신 후 선택한 시간 동안 이벤트가 발생하지 않습니다.

**이벤트 수신 태그**      전체 멤버 수신

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

[🔗 프로젝트 이벤트 수신설정 메뉴](#)

이벤트 설정 시 태그를 선택하지 않은 경우 프로젝트 이벤트 수신 설정 메뉴의 나머지 수신 조건(활성화 여부 등)에 따라 알림이 발생합니다.

- 발생 횟수:** 선택한 시간 동안 **이벤트 발생 조건**에서 설정한 이벤트가 입력 횟수만큼 발생하면 경고 알림을 보냅니다.

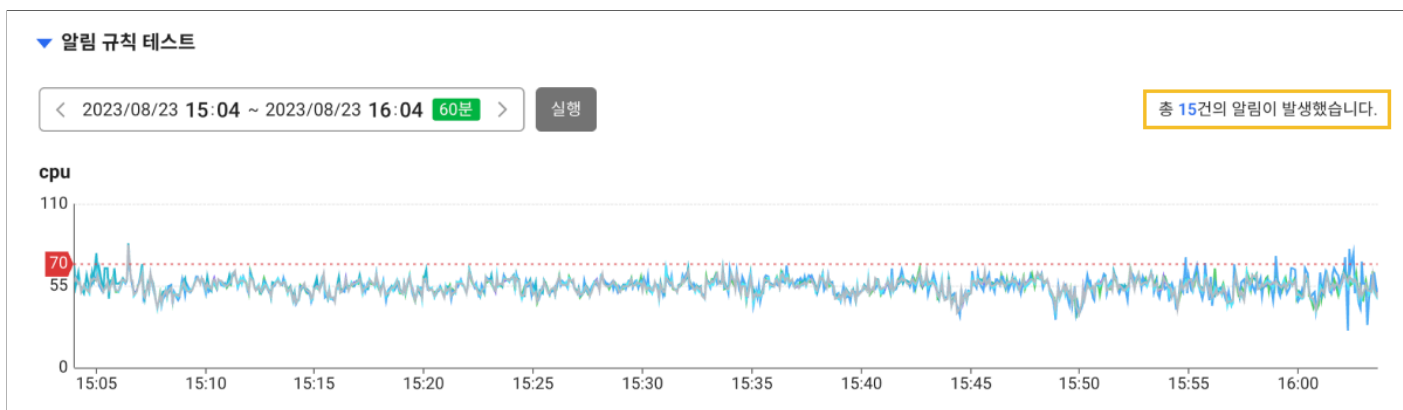
- ①
  - 선택 시간을 **사용 안 함**으로 설정하면 입력한 횟수만큼 연속 발생할 때 알림을 보냅니다.
  - 이벤트 상태가 해결되면 추가 알림** 옵션을 활성화한 경우 선택 시간은 **사용 안 함**으로 선택할 것을 권장합니다.
  - 카테고리** 옵션에서 선택한 항목의 수집 주기는 5초입니다.

- 이벤트 발생 일시 중지:** 과도한 경고 알림 발생을 방지할 수 있는 옵션입니다. 첫번째 경고 알림 이후 선택한 시간 동안 경고 알림을 보내지 않습니다. 또한 **이벤트 기록** 메뉴에 기록되지 않습니다.

- **관련 카테고리:** 관련 카테고리를 5개까지 설정하고 알림 조회 시 참조합니다.
- **이벤트 수신 태그:** 이벤트 수신 태그를 선택하면 해당 태그를 가진 프로젝트 멤버와 3rd-party 플러그인에 알림을 전송할 수 있습니다. 이벤트 수신 태그를 선택하지 않으면 프로젝트 전체 멤버에게 경고 알림을 보냅니다.

ⓘ 경고 알림 > 이벤트 수신 설정 메뉴에서 프로젝트 멤버와 3rd-party 플러그인에 태그를 설정할 수 있습니다.

## 알림 규칙 테스트



선택한 시간 동안 설정한 이벤트 조건을 실행해 몇 번의 경고 알림이 발생했는지 확인할 수 있습니다. 실행 버튼을 선택하면 알림 발생 건수 정보를 알 수 있으며, 이벤트 발생 조건에서 선택한 필드와 임계치를 차트상에 표시합니다.

## 복합 메트릭스 이벤트

복합 메트릭스 이벤트를 이용하려면 다음의 개념에 대한 이해가 필요합니다.

- [메트릭스란?](#)
- [MXQL](#)

복합 메트릭스 이벤트는 메트릭스 데이터에 보다 복잡한 규칙을 활용해 이벤트를 생성하고 경고 알림을 보낼 수 있습니다. 복합 메트릭스는 다음과 같은 상황에서 효과적으로 사용할 수 있습니다.

- 여러 에이전트에서 수신된 데이터에 대해 종합적인 이벤트 판정을 해야할 때
- 과거 데이터와 현재 데이터를 비교해 이벤트 판정을 해야할 때

메트릭스 이벤트는 에이전트로부터 메트릭스를 수신할 때마다 이벤트 판정을 합니다. 반면, 복합 메트릭스 이벤트는 각 에이전트에서 수집한 메트릭스들을 데이터베이스에 저장합니다. 그리고 다시 조회해서 이벤트 판정을 합니다. 이와 같은 특성 때문에 여러

에이전트의 데이터를 종합적으로 활용하거나 과거의 데이터를 활용할 수 있습니다. 하지만 **MXQL**이라는 와탭 고유의 데이터 조회 언어를 사용해야한다는 진입장벽이 존재합니다. 따라서 사용자들이 기초적인 **MXQL**만 이해하더라도 효과적으로 이벤트를 설정할 수 있도록 이벤트 템플릿을 제공합니다. MXQL 기초 사용자는 이벤트 대상 필터링과 이벤트 조건에 대한 쿼리만 수정해서 이벤트를 적용할 수 있습니다.

1. 경고 알림 > 이벤트 설정 메뉴에서 화면 위에 **메트릭스**를 선택하세요.
2. **복합 메트릭스** 섹션에서 오른쪽에 **이벤트 추가**를 선택하세요.
3. **복합 메트릭스** 창이 나타나면 **차트로 생성하기**를 선택하세요.

이벤트 설정 창이 나타납니다.

### 이벤트 설정

**알림**

이벤트 활성화

레벨 위험 경고 정보 이벤트 상태가 해결되면 추가 알림

\* 제목

\* 메시지

**이벤트 발생 조건**

데이터 조회 범위 1분

\* 조건  이벤트 데이터 조회에 포함된 필드만 사용 가능합니다

[> 추가정보](#)

**이벤트 규칙 테스트**

2023/08/22 실행

### 이벤트 데이터 조회

위젯 텍스트


< 2023/08/23 16:41 ~ 2023/08/23 16:51 10분 >

카테고리 [선택된 카테고리]

필터 + 필터 추가

그룹화 [0id (pk) X] 타임 유닛 5 초 ▼

필드 [HttpCallCount X]



name	sum	avg	max	recently
633280970	10285	85.70833333333333	122	81
1387800924	10363	86.35833333333333	121	101
-1128904592	10824	90.2	117	100
-1143239575	10038	83.65	124	107
-857948929	10388	86.56666666666666	115	102
-877561626	10278	85.65	127	102

저장

❗ 복합 메트릭스 이벤트를 설정하려면 **이벤트 설정** 권한이 있어야 합니다.

❗ 복합 메트릭스에 이벤트 템플릿에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

## 이벤트 데이터 조회

복합 메트릭스 이벤트는 메트릭스 데이터 질의 언어인 **MXQL**을 기반으로 이벤트 조건을 생성합니다. [차트로 생성하기](#) 기능은 **MXQL**의 자동완성을 위한 콤보박스 기능을 제공합니다. 이벤트 데이터를 조회하여 차트를 구성한 다음 이벤트 발행 조건을 직접 입력하기 위한 템플릿입니다. [위젯](#) 또는 [텍스트](#) 옵션을 선택해 이벤트를 설정하세요.

[위젯](#)

[텍스트](#)

시계열 차트를 구성하는 옵션을 통해 이벤트 설정 시 사용할 **MXQL**을 자동완성할 수 있습니다.

이벤트 데이터 조회

위젯 텍스트

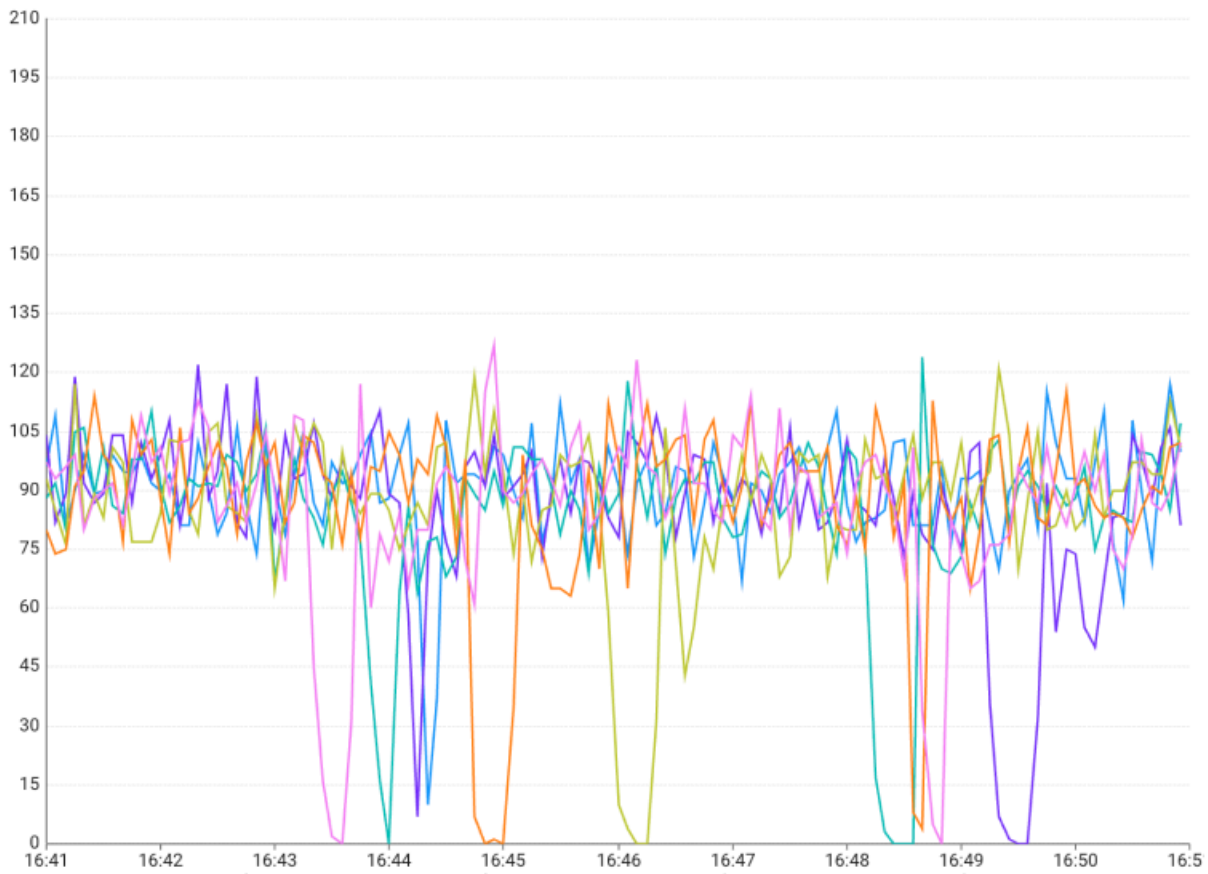
< 2023/08/23 16:41 ~ 2023/08/23 16:51 10분 >

카테고리

필터 + 필터 추가

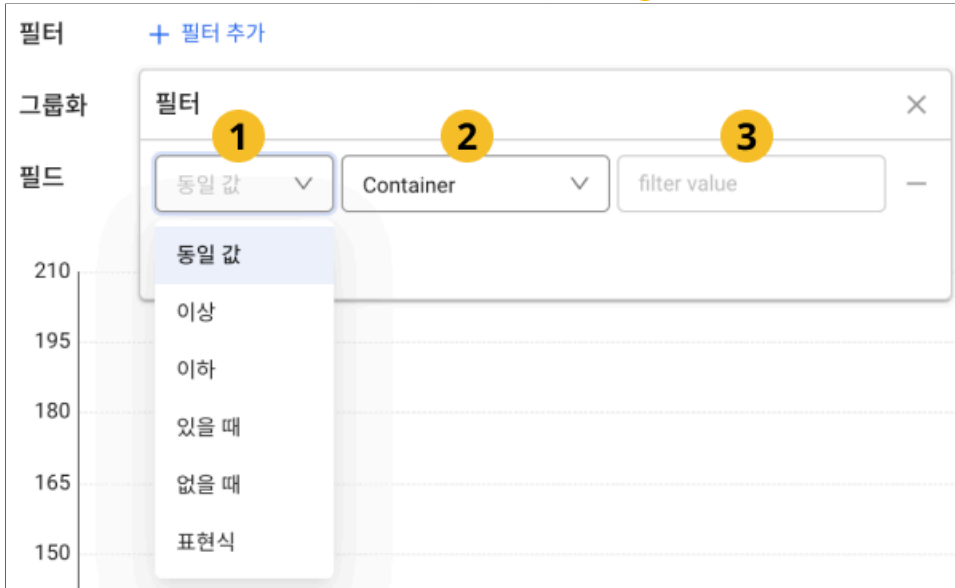
그룹화  타임 유닛 5 초

필드



name	sum	avg	max	recently
633280970	10285	85.70833333333333	122	81
1387800924	10363	86.35833333333333	121	101
-1128904592	10824	90.2	117	100

- **필터**: 이벤트 조건 대상을 선택합니다. ① 연산식, ② 태그, ③ 필터링값을 입력해 필터링 조건을 생성합니다.



- **그룹화**: 그룹화된 메트릭스 데이터를 선택합니다. 다중 선택할 수 있습니다.
- **타임 유닛**: 그룹화된 데이터를 나눌 시간 기준을 설정합니다. 초, 분, 시간 단위로 선택하고 설정할 수 있습니다.
- **필드**: 이벤트 발행 조건에 사용할 필드를 선택합니다. 다중 선택할 수 있습니다.

**MXQL**을 평문 그대로 수정할 수 있는 편집창이 나타납니다.



이벤트 데이터 조회

위젯 텍스트

```

INJECT timepast
HEADER { "httpc_count$":"COUNT" }
OIDSET { oid:$oid, okind:$okind, onode:$onode }
CATEGORY {"app_counter":6h, "app_counter{m5}":3d, "app_counter{h1}":unlimit }
TAGLOAD
INJECT default
UPDATE {key: httpc_count, value: avg}
FILTER {key: container, exist: true}
GROUP {pk:[oid], timeunit: 5000}
UPDATE {key: httpc_count, value: avg}
CREATE {key: _id_, expr:"oid"}
CREATE {key: _name_, expr:"oname"}
SELECT [_name_, _id_, time, oid, httpc_count]
    
```

조회

< 2023/08/23 16:51 ~ 2023/08/23 17:01 10분 >



httpc_count	_name_	_id_	time	oid
89	demo-8105	-857948929	2023/08/23 16:51:00	-857948929
82	demo-8100	-1128904592	2023/08/23 16:51:00	-1128904592
98	demo-8101	-877561626	2023/08/23 16:51:00	-877561626
81	demo-8102	1387800924	2023/08/23 16:51:00	1387800924
90	demo-8104	-1143239575	2023/08/23 16:51:00	-1143239575
107	demo-8103	633280970	2023/08/23 16:51:00	633280970
107	demo-8105	-857948929	2023/08/23 16:51:05	-857948929
85	demo-8100	-1128904592	2023/08/23 16:51:05	-1128904592
97	demo-8101	-877561626	2023/08/23 16:51:05	-877561626
94	demo-8102	1387800924	2023/08/23 16:51:05	1387800924
86	demo-8104	-1143239575	2023/08/23 16:51:05	-1143239575
85	demo-8103	633280970	2023/08/23 16:51:05	633280970
88	demo-8105	-857948929	2023/08/23 16:51:10	-857948929
6	demo-8100	-1128904592	2023/08/23 16:51:10	-1128904592
79	demo-8101	-877561626	2023/08/23 16:51:10	-877561626
84	demo-8102	1387800924	2023/08/23 16:51:10	1387800924
77	demo-8104	-1143239575	2023/08/23 16:51:10	-1143239575
70	demo-8103	633280970	2023/08/23 16:51:10	633280970
86	demo-8105	-857948929	2023/08/23 16:51:15	-857948929
5	demo-8100	-1128904592	2023/08/23 16:51:15	-1128904592

이전

페이지 1 of 4

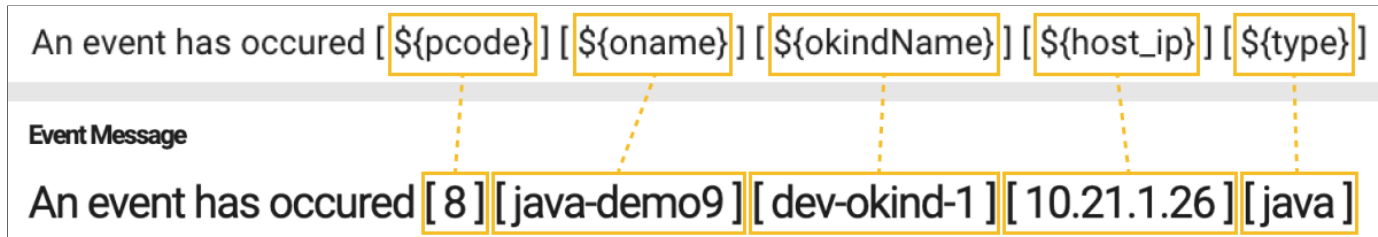
다음

저장

## 알림

경고 알림 설정의 기본 정보를 입력합니다.

- **이벤트 활성화:** 토글 버튼을 클릭해 이벤트를 활성화할 여부를 선택할 수 있습니다.
- **레벨:** 위험(Critical), 경고(Warning), 정보 수준 중 하나의 레벨을 선택하세요. **이벤트 상태가 해결되면 추가 알림:** 이벤트 항목 중 발생한 이벤트 상태가 해결되면 추가 알림 송신 여부를 선택할 수 있습니다. 토글 버튼을 선택해 기능을 켜거나 끌 수 있습니다.
- **제목:** 경고 알림의 제목을 입력하세요.
- **메시지:** 이벤트 발생 시 출력하는 알림 메시지를 입력합니다.  `${Tag}`  또는  `${Field}`  입력으로 메시지에 변수를 적용할 수 있습니다. 변수에 입력할 키는 선택한 메트릭스 데이터 **카테고리**에 포함된 값이어야 합니다. **메트릭스 조회** 메뉴에서 입력할 수 있는 태그 또는 필드키를 확인할 수 있습니다.



## 알림 정책

경고 알림을 보낼 조건을 입력합니다.

- **데이터 조회 범위:** 이벤트 조건에 사용할 **MXQL**의 실시간 데이터 조회 범위를 설정합니다. 이벤트 데이터 조회에 포함된 필드만 사용할 수 있습니다.  
복합 메트릭스 이벤트는 DB에 저장된 메트릭스를 조회해서 활용합니다. 따라서 데이터를 조회할 시간 범위를 먼저 지정해야 합니다. 데이터 조회 시간을 5분으로 선택하면 최근 5분동안 수집된 데이터를 조회해서 이벤트 발생 조건을 확인합니다. 최근 데이터에 대해서 이벤트를 설정할 때에는 짧게, 넓은 시간에 대해서 통계적으로 접근하고 싶을 때에는 길게 설정할 수 있습니다. 실제 사용 예시는 [다음 문서](#)를 참조하세요.
- **조건:** MXQL에 반영한 필드와 연산 규칙, 임계치를 입력합니다.

## 부가 정보

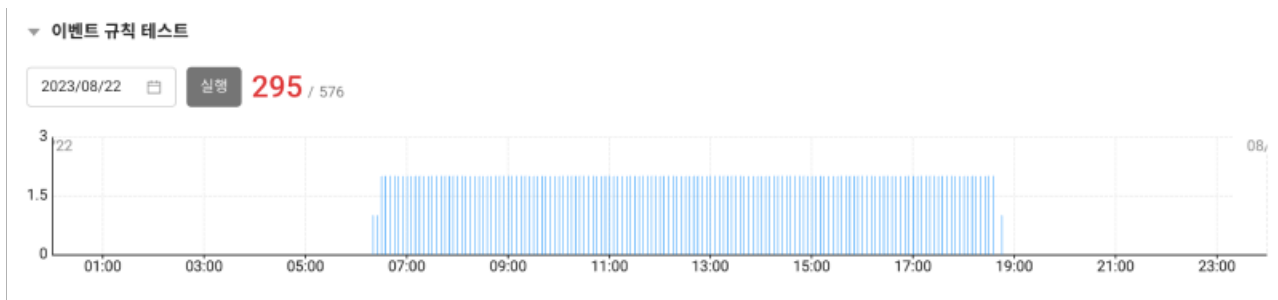
경고 알림 수신과 관련한 부가적인 옵션을 설정합니다.

- **인터벌:** 선택한 시간 간격으로 알림 조건을 확인합니다.
- **무음:** 과도한 경고 알림 발생을 방지할 수 있는 옵션입니다. 첫번째 경고 알림 이후 선택한 시간 동안 경고 알림을 보내지 않습니다. 또한 **이벤트 기록** 메뉴에 기록되지 않습니다.

- **이벤트 수신 태그:** 이벤트 수신 태그를 선택하면 해당 태그를 가진 프로젝트 멤버와 3rd-party 플러그인에 알림을 전송할 수 있습니다. 이벤트 수신 태그를 선택하지 않으면 프로젝트 전체 멤버에게 경고 알림을 보냅니다.

ⓘ 경고 알림 > 이벤트 수신 설정 메뉴에서 프로젝트 멤버와 3rd-party 플러그인에 태그를 설정할 수 있습니다.

## 이벤트 규칙 테스트



선택한 시간 동안 설정한 이벤트 조건을 실행해 몇 번의 경고 알림이 발생했는지 확인할 수 있습니다. 실행 버튼을 선택하면 알림 발생 건수 정보를 알 수 있으며, 이벤트 발생 조건에서 선택한 필드와 임계치를 차트상에 표시합니다.

이벤트 설정에 포함된 대부분의 내용들이 **MXQL**을 사용해서 지정됩니다. **MXQL**이 적절하게 작성되었는지 시뮬레이션할 수 있는 기능을 제공합니다. 시뮬레이션 기능은 과거의 24시간 데이터를 조회해서 이벤트 판정을 한 다음 몇 건의 메트릭스가 조회되었고 그 중 몇 건에서 이벤트 판정이 성공했는지 알려줍니다.

## 메트릭스 이벤트 수정 및 삭제

1. 경고 알림 > 이벤트 설정 메뉴로 이동한 다음 **메트릭스** 탭을 선택하세요.
2. 이벤트 목록에서 수정 또는 삭제하려는 항목의 가장 오른쪽에 버튼을 선택하세요.
3. 메트릭스 또는 복합 메트릭스 이벤트 설정 창이 나타나면 각 옵션을 수정한 다음 **저장** 버튼을 선택하세요.  
 선택한 이벤트를 삭제하려면 이벤트 설정 창의 오른쪽 위에 **삭제** 버튼을 선택하세요.

## 발생 조건, 대상 선택 가이드

메트릭스 경고 알림의 이벤트 발생 조건과 이벤트 대상 선택은 동일한 문법을 사용합니다. 단, 이벤트 발생 조건은 태그(Tag)의 Key를 변수로 사용하고, 이벤트 대상 선택은 필드(Field)의 Key를 변수로 사용합니다.

## 기본 문법

- 문자열을 그냥 입력하면 변수, 작은 따옴표(') 또는 큰 따옴표("")로 감싸면 text로 인식합니다.

```
oid == "oid"
```

- oid : 변수
- == : 함수
- "oid" : text

```
// oname가 ott-1235일 경우
```

```
// 정상적인 경우
```

```
onname = 'ott-1235' 또는 onname = "ott-1235"
```

```
// 비정상적인 경우, 알림이 동작하지 않습니다.
```

```
onname = ott-1235
```

- 숫자를 그냥 입력하면 number, 작은 따옴표(') 또는 큰 따옴표("")로 감싸면 text로 인식합니다.

```
oid == 123
```

- oid : 변수
- == : 함수
- 123 : number

```
// oid가 123일 경우
```

```
// 정상적인 경우
```

```
oid = 123
```

```
// 비정상적인 경우, 알림이 동작하지 않습니다.
```

```
id == '123' 또는 oid == "123"
```

## 사용 가능한 연산자 목록

연산자	사용법	설명
==	operand1 == operand2	operand1과 operand2의 값이 동일한지 확인합니다.
!=	operand1 != operand2	operand1과 operand2의 값이 다른지 확인합니다.
>	operand1 > operand2	operand1의 값이 operand2의 값보다 큰지 확인합니다.
>=	operand1 >= operand2	operand1의 값이 operand2의 값보다 크거나 같은지 확인합니다.
<	operand1 < operand2	operand1의 값이 operand2의 값보다 작은지 확인합니다.
<=	operand1 <= operand2	operand1의 값이 operand2의 값보다 작거나 같은지 확인합니다.
like	operand1 like operand2	operand1에 operand2가 포함되어 있는지를 패턴으로 검색합니다.
&&	expression1 && expression2	expression1과 expression2이 모두 true인지 확인합니다.
and	expression1 and expression2	expression1과 expression2이 모두 true인지 확인합니다. &&와 동일한 역할을 수행하는 연산자입니다.
	expression1    expression2	expression1 또는 expression2이 true인지 확인합니다.
or	expression1 or expression2	expression1 또는 expression2이 true인지 확인합니다.   와 동일한 역할을 수행하는 연산자입니다.

### like 사용법

와일드카드(**\***)를 통해 포함 문자열을 편리하게 검색할 수 있습니다.

- 특정 키워드로 시작하는 문자열 검색

Key like "Value\*"

- 특정 키워드로 끝나는 문자열 검색

```
Key like "*Value"
```

- 특정 키워드가 포함된 문자열 검색

```
Key like "*Value*"
```

- 키워드 중간에 와일드카드(\*)를 사용할 수 없습니다.

```
// 지원하지 않는 문법
Key like "Va*lue"
```

- like 연산자에서 와일드카드(\*)를 생략하는 경우 equals(==)로 동작합니다.

```
// 아래의 두 문장은 완전히 같은 결과를 가집니다.
Key like "Value"
Key == "Value"
```

## 사용 가능한 함수 목록

함수	사용법	설명
<a href="#">startsWith</a>	startsWith(param1, param2)	param1을 Key로 하는 Value가 param2로 시작하면 true, 반대의 경우 false

함수	사용법	설명
<a href="#">endsWith</a>	endsWith(param1, param2)	param1을 Key로 하는 Value가 param2로 끝나면 true, 반대의 경우 false
<a href="#">isNull</a>	isNull(param1)	param1이 null이면 true, 반대의 경우 false
<a href="#">isNotNull</a>	isNotNull(param1)	param1이 null이 아니면 true, 반대의 경우 false
<a href="#">isEmpty</a>	isEmpty(param1)	param1이 null 또는 EmptyString("") 이면 true, 반대의 경우 false
<a href="#">isNotEmpty</a>	isNotEmpty(param1)	param1이 null도 아니고 EmptyString("") 도 아니면 true, 반대의 경우 false

### startsWith

```
startsWith(Key, "Value")
```

### endsWith

```
endsWith(Key, "Value")
```

### isNull

```
isNull(Key)
```

### isNotNull

```
isNotNull(Key)
```

## isEmpty

```
isEmpty(Key)
```

## isNotEmpty

```
isNotEmpty(Key)
```

# 템플릿

## 메트릭스 이벤트

경고 알림 > 이벤트 설정 > 메트릭스 섹션에서 + 이벤트 추가 버튼을 선택하세요. 템플릿에서 원하는 항목을 선택하세요.

- [WARNING\\_OLD\\_GENERATION\\_GC](#)

힙 메모리(Heap Memory)의 Old Generation 영역에서 GC가 발생하면 경고 알림을 보냅니다.

- [TOO\\_SLOW\\_SQL](#)

수행 중인 쿼리가 5초 이상으로 너무 느린 쿼리가 발생할 경우 경고 알림을 보냅니다.

- [TOO\\_MANY\\_ACTX](#)

8초 초과 구간의 액티브 트랜잭션 수가 너무 많을 경우 경고 알림을 보냅니다.

- [APDEX](#)

APDEX 수치가 0.7보다 작을 경우 경고 알림을 보냅니다.

## 복합 메트릭스 이벤트

- [Inactive agents has been found.](#)

조건: `num_of_current_agents` < 6

프로젝트에 포함된 모든 에이전트 중 정상 상태의 에이전트 수가 6개 미만으로 떨어지는 이벤트가 발생하면 경고 알림을 보냅니다.

- [Very slow active transactions detected.](#)

조건: `very_slow_tx_cnt_m5_avg` > 10

프로젝트에 포함된 특정 `okind` 에 속한 에이전트에서 8초 이상 소요되는 트랜잭션의 수의 합이 10개 초과 발생하면 경고 알림을



보냅니다.

- [TPS has changed by more than 30% compared to the previous week.](#)

조건: `one_week_diff_abs` > `current_tps` \* 0.3

프로젝트에 포함된 특정 `okind` 에 속한 에이전트들의 TPS합이 7일 전과 비교했을 때 30% 초과 달라지면 경고 알림을 보냅니다.

여러 에이전트에서 수신된 데이터에 대해서 종합적 이벤트 판정해야할 때 다음의 복합 메트릭스 템플릿을 설정하세요.

- [Inactive agents has been found.](#)
- [Very slow active transactions detected.](#)

과거 데이터와 현재 데이터를 비교해서 이벤트를 판정해야할 때 다음의 복합 메트릭스 템플릿을 설정하세요.

- [TPS has changed by more than 30% compared to the previous week.](#)

# 알림 메시지 사용자 정의

홈 화면 > 프로젝트 선택 > 경고 알림 > 이벤트 설정 > 메트릭스 탭 선택

프로젝트 멤버에게 메일 또는 서드 파티 플러그인을 통해 전달하는 경고 알림 메시지를 수정할 수 있는 사용자 정의 기능을 제공합니다. 화면 오른쪽 위에 **알림 메시지 설정** 버튼을 선택하세요. 수신할 메시지에서 불필요하다고 생각되는 항목을 감추거나 사용자가 원하는 특별한 내용을 추가할 수 있습니다.

The screenshot shows the '이벤트 설정' (Event Settings) interface. At the top, there are navigation tabs: '애플리케이션', '메트릭스', '히트맵 패턴', '이상치 탐지', and '로그'. Below this, there are two main sections: '메트릭스' and '복합 메트릭스'. Each section contains a table of alert rules. The '메트릭스' section has two rows of alerts, and the '복합 메트릭스' section has one row. Each row includes columns for '카테고리' (Category), '대상' (Target), '규칙' (Rule), '이벤트 상태가 해결되면 추가 알림' (Additional alerts when event status is resolved), and '이벤트 수신 태그' (Event notification tags). The '알림 메시지 설정' button is highlighted in yellow in both sections.

- ① 이 기능은 **메트릭스**, **복합 메트릭스** 경고 알림에만 적용할 수 있습니다.
- 이 기능은 프로젝트 선택 > **실험실** > **알림 메시지** 설정 메뉴 경로로도 진입할 수 있습니다.
- 이 기능은 **알림 설정** 권한을 가진 멤버가 설정할 수 있습니다. 자세한 내용은 [다음 문서](#)를 참조하세요.

# 기본 화면 안내

알림 메시지
🔔 📄 ⋮ 👤

**알림 메시지**

수신할 메시지의 상세 항목을 선택하고, 항목 별 내용을 편집할 수 있습니다.

다른 프로젝트에 복사
전체 초기화
전체 저장

전체 펼치기
발생
해결
이벤트 발생 메시지
이벤트 해결 메시지

전체
발생
해결

검색

메트릭스	발생	해결
1 상태	ex) 상태:OCCURED/SOLVED	3 🔔 미리보기
알림 레벨	ex) 알림 레벨:Warning	4 ✎ 수정 <input checked="" type="checkbox"/>
프로젝트 번호	ex) 프로젝트 번호:101	✎ 수정 <input checked="" type="checkbox"/>
프로젝트 이름	ex) 프로젝트 이름:Example Project	✎ 수정 <input checked="" type="checkbox"/>
애플리케이션 이름	ex) 애플리케이션 이름:Whatap-Front-01	✎ 수정 <input checked="" type="checkbox"/>
메시지	ex) 메시지:Message	✎ 수정 <input checked="" type="checkbox"/>
이벤트 발생 시각	ex) 이벤트 발생 시각:2022-09-08 14:23:17 +0900	✎ 수정 <input checked="" type="checkbox"/>
이벤트 발생 조건	ex) 이벤트 발생 조건:tps > 20	✎ 수정 <input checked="" type="checkbox"/>
이벤트 대상 필터링	ex) 이벤트 대상 필터링:oid == 800488350	✎ 수정 <input checked="" type="checkbox"/>
해결된 이벤트 알림 기능	ex) 해결된 이벤트 알림 기능:ON/OFF	✎ 수정 <input checked="" type="checkbox"/>
알림 종류	ex) 알림 종류:METRICS	✎ 수정 <input checked="" type="checkbox"/>
알림 라벨	ex) 알림 라벨:oid(800488350)	✎ 수정 <input checked="" type="checkbox"/>
지표	ex) 지표:tps	✎ 수정 <input checked="" type="checkbox"/>
현재값	ex) 현재값:25	✎ 수정 <input checked="" type="checkbox"/>
임계값	ex) 임계값:20	✎ 수정 <input checked="" type="checkbox"/>
현재값(복수)	ex) 현재값(복수):tps=23.39, tx_error=7	✎ 수정 <input checked="" type="checkbox"/>
반복 횟수	ex) 반복 횟수:5	✎ 수정 <input checked="" type="checkbox"/>
반복 시간	ex) 반복 시간:5 minutes	✎ 수정 <input checked="" type="checkbox"/>
무음 시간	ex) 무음 시간:10 minutes	✎ 수정 <input checked="" type="checkbox"/>

▶ 메트릭스 해결

▶ 복합 메트릭스 발생

▶ 복합 메트릭스 해결

기본 설정 |  초기화  미리보기  저장  전체 활성화


- 알림 메시지의 종류는 **발생** 과 **해결** 로 분류할 수 있습니다. **발생** 은 이벤트가 발생했을 때 전달되는 알림이며, **해결** 은 이벤트가 해소되었을 때 전달되는 알림입니다.
- 1 영역은 수신할 메시지의 상세 항목입니다. 2 영역은 수신할 메시지의 상세 항목에 대한 내용입니다. 3 🔔 **미리보기** 버튼을

470

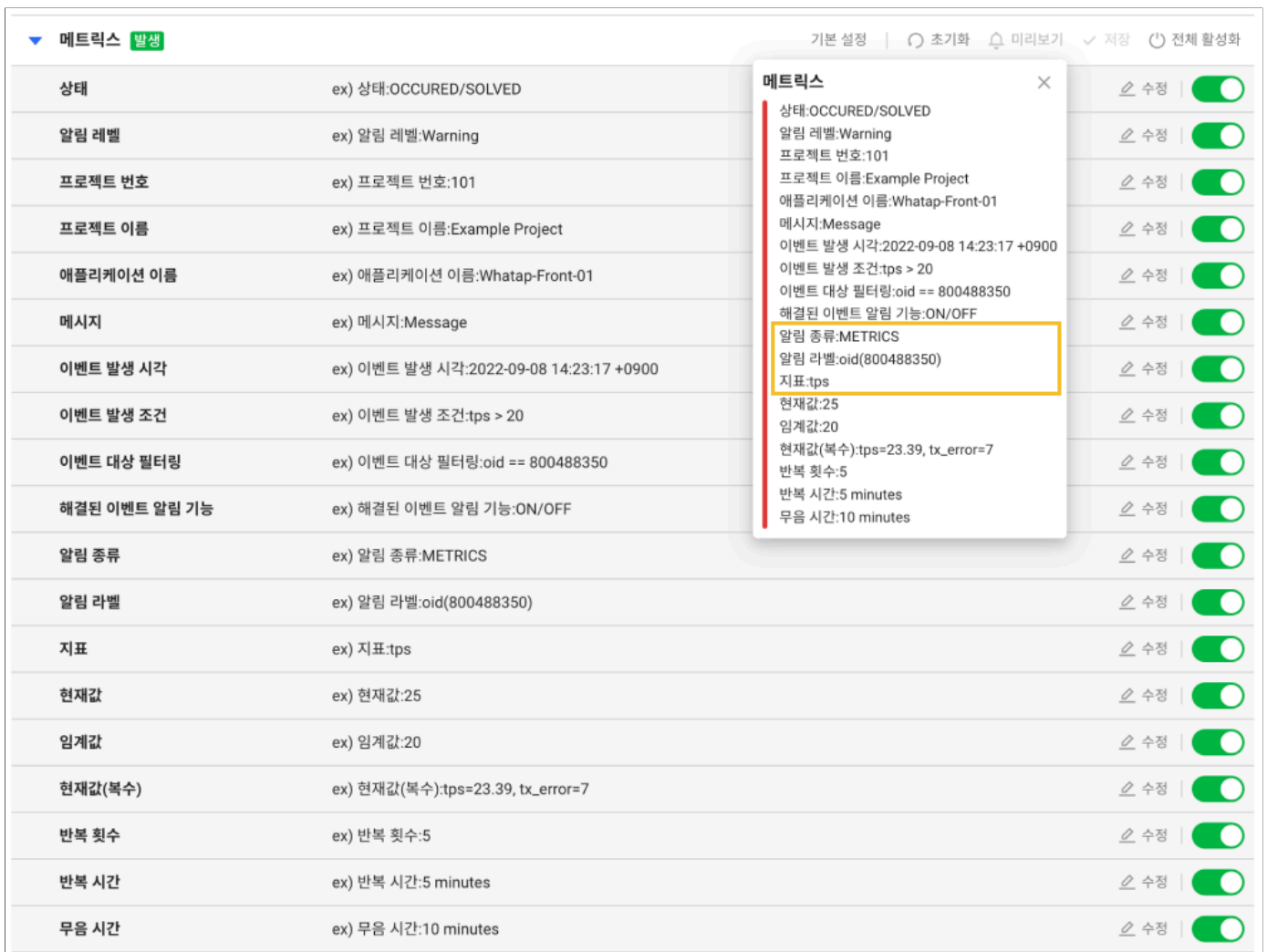
선택하면 수신할 메시지의 내용을 미리 확인할 수 있습니다.

- 수신한 메시지의 내용을 수정하려면 **4**  수정 버튼을 선택하세요.
- 수신한 메시지의 상세 항목을 감추려면 **5**  토글 버튼을 선택해 비활성화하세요.

## 수신 메시지 상세 항목 감추기

수신 메시지의 내용 중 불필요하다고 생각되는 항목을 감출 수 있습니다. 각 상세 항목의 가장 오른쪽의 토글 버튼을 선택해 해당 항목을 수신 메시지에서 감추거나 보이도록 설정할 수 있습니다.  **미리보기** 버튼을 선택해 수신할 메시지의 내용을 미리 확인할 수 있습니다.

- 상세 항목 활성화 예시



항목	예시	활성화
상태	ex) 상태:OCCURED/SOLVED	<input checked="" type="checkbox"/>
알림 레벨	ex) 알림 레벨:Warning	<input checked="" type="checkbox"/>
프로젝트 번호	ex) 프로젝트 번호:101	<input checked="" type="checkbox"/>
프로젝트 이름	ex) 프로젝트 이름:Example Project	<input checked="" type="checkbox"/>
애플리케이션 이름	ex) 애플리케이션 이름:Whatap-Front-01	<input checked="" type="checkbox"/>
메시지	ex) 메시지:Message	<input checked="" type="checkbox"/>
이벤트 발생 시간	ex) 이벤트 발생 시간:2022-09-08 14:23:17 +0900	<input checked="" type="checkbox"/>
이벤트 발생 조건	ex) 이벤트 발생 조건:tps > 20	<input checked="" type="checkbox"/>
이벤트 대상 필터링	ex) 이벤트 대상 필터링:oid == 800488350	<input checked="" type="checkbox"/>
해결된 이벤트 알림 기능	ex) 해결된 이벤트 알림 기능:ON/OFF	<input checked="" type="checkbox"/>
알림 종류	ex) 알림 종류:METRICS	<input checked="" type="checkbox"/>
알림 라벨	ex) 알림 라벨:oid(800488350)	<input checked="" type="checkbox"/>
지표	ex) 지표:tps	<input checked="" type="checkbox"/>
현재값	ex) 현재값:25	<input checked="" type="checkbox"/>
임계값	ex) 임계값:20	<input checked="" type="checkbox"/>
현재값(복수)	ex) 현재값(복수):tps=23.39, tx_error=7	<input checked="" type="checkbox"/>
반복 횟수	ex) 반복 횟수:5	<input checked="" type="checkbox"/>
반복 시간	ex) 반복 시간:5 minutes	<input checked="" type="checkbox"/>
무음 시간	ex) 무음 시간:10 minutes	<input checked="" type="checkbox"/>

**메트릭스** [X]

상태:OCCURED/SOLVED

알림 레벨:Warning

프로젝트 번호:101

프로젝트 이름:Example Project

애플리케이션 이름:Whatap-Front-01

메시지:Message

이벤트 발생 시간:2022-09-08 14:23:17 +0900

이벤트 발생 조건:tps > 20

이벤트 대상 필터링:oid == 800488350

해결된 이벤트 알림 기능:ON/OFF

알림 종류:METRICS

알림 라벨:oid(800488350)

지표:tps

현재값:25

임계값:20

현재값(복수):tps=23.39, tx\_error=7

반복 횟수:5

반복 시간:5 minutes

무음 시간:10 minutes

- 상세 항목 비활성화 예시

메트릭스 발생
기본 설정 | 초기화 | 미리보기 | 저장 | 전체 활성화

상태	ex) 상태:OCCURED/SOLVED	수정   <input checked="" type="checkbox"/>
알림 레벨	ex) 알림 레벨:Warning	수정   <input checked="" type="checkbox"/>
프로젝트 번호	ex) 프로젝트 번호:101	수정   <input checked="" type="checkbox"/>
프로젝트 이름	ex) 프로젝트 이름:Example Project	수정   <input checked="" type="checkbox"/>
애플리케이션 이름	ex) 애플리케이션 이름:Whatap-Front-01	수정   <input checked="" type="checkbox"/>
메시지	ex) 메시지:Message	수정   <input checked="" type="checkbox"/>
이벤트 발생 시각	ex) 이벤트 발생 시각:2022-09-08 14:23:17 +0900	수정   <input checked="" type="checkbox"/>
이벤트 발생 조건	ex) 이벤트 발생 조건:tps > 20	수정   <input checked="" type="checkbox"/>
이벤트 대상 필터링	ex) 이벤트 대상 필터링:oid == 800488350	수정   <input checked="" type="checkbox"/>
해결된 이벤트 알림 기능	ex) 해결된 이벤트 알림 기능:ON/OFF	수정   <input checked="" type="checkbox"/>
알림 종류	ex) 알림 종류:METRICS	수정   <input type="checkbox"/>
알림 라벨	ex) 알림 라벨:oid(800488350)	수정   <input type="checkbox"/>
지표	ex) 지표:tps	수정   <input type="checkbox"/>
현재값	ex) 현재값:25	수정   <input checked="" type="checkbox"/>
임계값	ex) 임계값:20	수정   <input checked="" type="checkbox"/>
현재값(복수)	ex) 현재값(복수):tps=23.39, tx_error=7	수정   <input checked="" type="checkbox"/>
반복 횟수	ex) 반복 횟수:5	수정   <input checked="" type="checkbox"/>
반복 시간	ex) 반복 시간:5 minutes	수정   <input checked="" type="checkbox"/>
무음 시간	ex) 무음 시간:10 minutes	수정   <input checked="" type="checkbox"/>

**메트릭스**


- 상태:OCCURED/SOLVED
- 알림 레벨:Warning
- 프로젝트 번호:101
- 프로젝트 이름:Example Project
- 애플리케이션 이름:Whatap-Front-01
- 메시지:Message
- 이벤트 발생 시각:2022-09-08 14:23:17 +0900
- 이벤트 발생 조건:tps > 20
- 이벤트 대상 필터링:oid == 800488350
- 해결된 이벤트 알림 기능:ON/OFF
- 현재값:25
- 임계값:20
- 현재값(복수):tps=23.39, tx\_error=7
- 반복 횟수:5
- 반복 시간:5 minutes
- 무음 시간:10 minutes

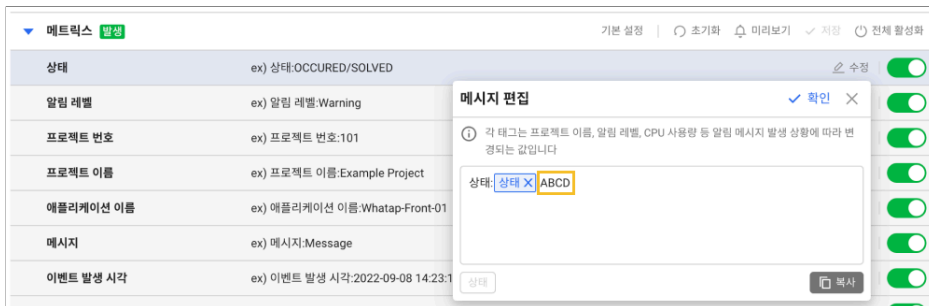
변경 사항을 반영하려면  저장 버튼을 선택하세요. 다른 카테고리의 변경 사항을 한번에 모두 반영하려면 화면 오른쪽 위에 전체 저장 버튼을 선택하세요.


- ⓘ • 감춘 상세 항목을 수신 메시지에 다시 추가하려면 해당 항목의 토글 버튼을 선택해 활성화하세요.
- 모든 상세 항목을 수신 메시지에 추가하려면  전체 활성화 버튼을 선택하세요.

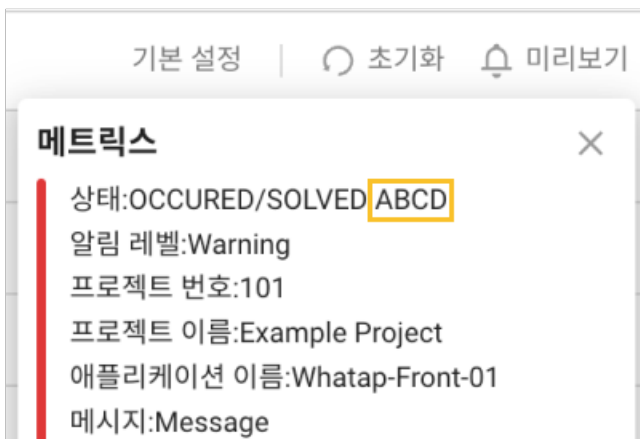
## 상세 항목 내용 수정하기


상세 항목의 내용에 사용자가 원하는 특별한 내용을 추가할 수 있습니다.

1. 각 상세 항목 가장 오른쪽에  수정 버튼을 선택하세요.
2. **메시지 편집** 창이 나타나면 추가할 내용을 입력하세요.

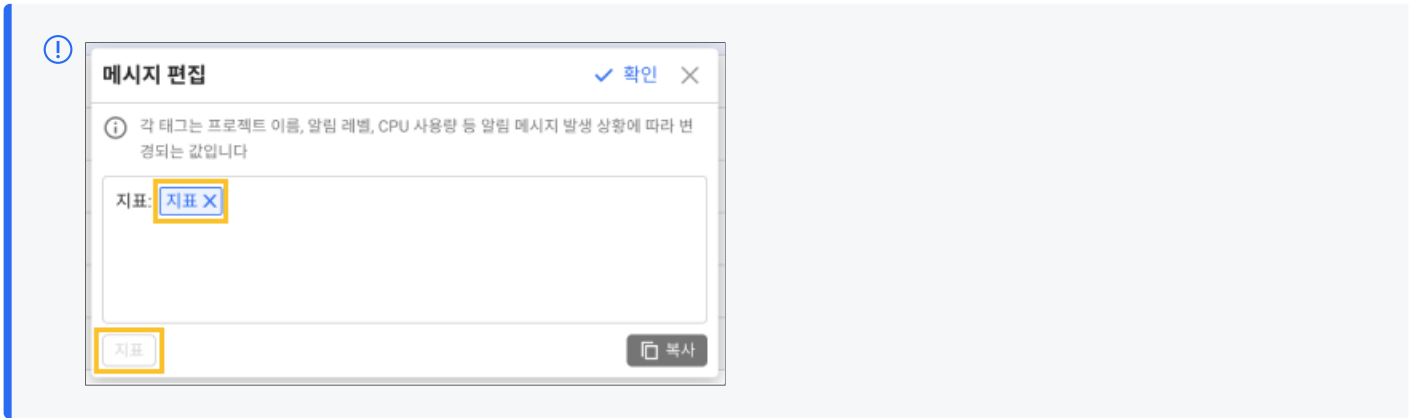


3. **확인** 버튼을 선택하세요.
4.  **미리보기** 버튼을 선택해 수신할 메시지의 내용을 미리 확인하세요.



변경 사항을 반영하려면  **저장** 버튼을 선택하세요. 다른 카테고리의 변경 사항을 한번에 모두 반영하려면 화면 오른쪽 위에 **전체 저장** 버튼을 선택하세요.

- ⓘ **메시지 편집** 창에서 텍스트 블록은 알림 발생 상황에 따라 변경되어 출력되는 변수입니다. 텍스트 블록을 드래그해서 위치를 변경할 수 있습니다. 텍스트 블록을 삭제했다면 창 아래에 텍스트 블록 버튼을 선택해 다시 추가할 수 있습니다.



- ⓘ 사용자에게 의해 내용을 수정하기 전의 메시지는 사용자의 언어 상태에 따라 번역되어 전송되나, 내용 수정 이후에는 언어 상태와 관계 없이 사용자 정의된 포맷으로만 알림을 전송합니다.

## 다른 프로젝트에 설정 복사하기

사용자가 설정한 내용을 다른 프로젝트에 복사할 수 있습니다.

1. 화면 오른쪽 위에 [다른 프로젝트에 복사](#) 버튼을 선택하세요.
2. [프로젝트 적용](#) 창이 나타나면 설정 내용을 복사할 프로젝트를 선택하세요. 모든 프로젝트를 선택하려면 [전체 선택](#)을 선택하세요.
3. [적용](#) 버튼을 선택하세요.

- ⓘ 현재 프로젝트와 동일한 제품 타입만 선택할 수 있습니다.

## 초기화하기

사용자가 설정한 모든 변경 사항을 초기화하려면 [초기화](#) 버튼을 선택하세요. 모든 카테고리의 변경 사항을 초기화하려면 화면 오른쪽 위에 [전체 초기화](#) 버튼을 선택하세요. 확인 메시지 창이 나타나면 [확인](#) 버튼을 선택하세요.

- ⓘ 초기화된 설정은 복구할 수 없습니다.

# AI 기반의 히트맵 패턴 경고 알림

홈 화면 > 프로젝트 선택 > 경고 알림 > 이벤트 설정 > 히트맵 패턴

애플리케이션 실행 분포 패턴을 머신 러닝을 통해 자동 인식하여 경고 알림을 보냅니다. 복잡한 조건 설정 없이도 이슈 상황을 빠르고 정확하게 인지할 수 있습니다. 수많은 히트맵 패턴들이 학습되어 있으며 실시간으로 수집되는 데이터에서 학습된 패턴과 유사한 패턴이 발생하면 경고 알림을 받아 볼 수 있습니다.

❗ 히트맵과 히트맵 패턴에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

## 기본 옵션

- **탐지 제외 시간**: 설정한 시간 미만의 패턴은 무시합니다.
- **무음**: 과도한 경고 알림 발생을 방지할 수 있는 옵션입니다. 첫번째 경고 알림 이후 선택한 시간 동안 경고 알림을 보내지 않습니다. 또한 **이벤트 기록** 메뉴에 기록되지 않습니다.

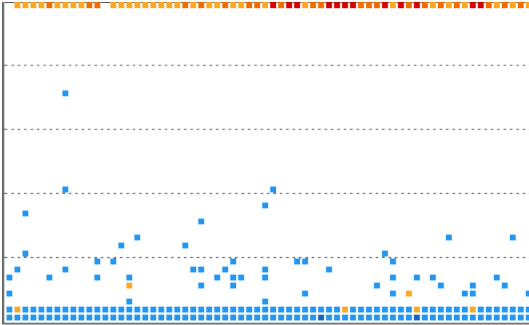


- **저장**: 설정한 내용을 저장하고 반영합니다.

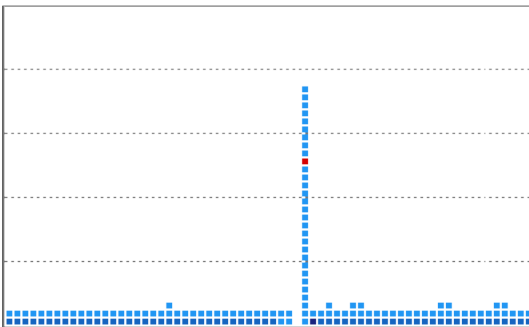
## 히트맵 패턴

다음의 히트맵 패턴이 발생하면 경고 알림을 보내도록 설정할 수 있습니다. 각 항목의 토글 버튼을 선택해 경고 알림을 켜거나 끌 수 있습니다.

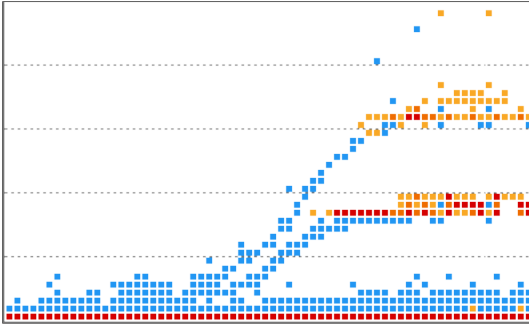
- **히트맵 가로라인 패턴**: 여러 트랜잭션이 일정한 시간 내에 종료되는 패턴입니다. 자원을 획득하거나 외부 HTTPS Call을 할 때 타임아웃 또는 지연이 발생합니다.



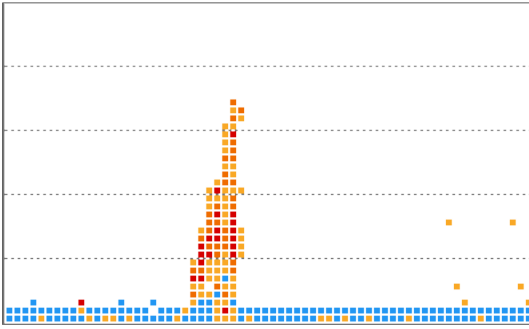
- **히트맵 세로라인 패턴**: 호출 시점이 다르지만 동일 시점에 트랜잭션이 종료되는 패턴입니다. 트랜잭션이 사용하는 공통의 자원에 일시적인 병목이 발생합니다.



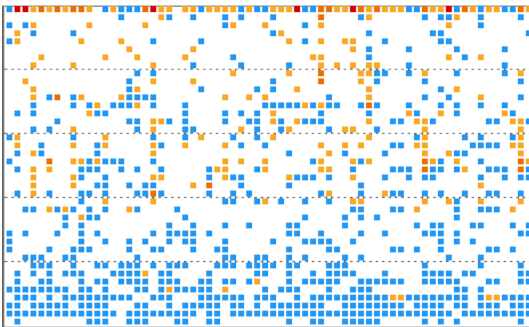
- **히트맵 플라잉 패턴**: 특정 리소스나 로그와 같은 공통 자원 부족현상으로 간격을 두고 파도 치는 것과 같은 현상의 패턴이 발생합니다.



- **히트맵 과부하 패턴:** 전체 또는 일부 응답에 일시적인 문제가 발생하면 일시적으로 트랜잭션이 밀집하는 패턴이 발생합니다.



- **히트맵 폭주 패턴:** 과도한 트랜잭션의 요청이나 부하가 발생하면 응답시간이 전체적으로 증가하는 패턴이 발생합니다.



ⓘ 히트맵과 히트맵 패턴에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

# 이상치 탐지 경고 알림

홈 화면 > 프로젝트 선택 > 경고 알림 > 이벤트 설정 > 이상치 탐지 탭 선택

이상치 탐지(Anomaly Detection)란 데이터 안에서 anomaly, outlier, abnormal과 같이 예상치 못한 패턴을 찾는 일련의 활동입니다. 사용자는 매트릭스의 데이터를 기반으로 상승 패턴과 하락 패턴의 움직임이 평소와 같지 않을 경우 경고 알림을 보내도록 설정할 수 있습니다.

## 이벤트 설정

애플리케이션
메트릭스
히트맵 패턴
이상치 탐지
로그

이상치 탐지
+ 이벤트 추가

	카테고리	필드	필터	상승/하락 패턴	
	app_counter(m5)	tx_error	없음	보통 / 보통	<input type="checkbox"/>
	java_memory(m5)	heap_used	없음	상승/하락 패턴 민감 / 민감	<input type="checkbox"/>
	app_host_resource(m5)	swap	없음	상승/하락 패턴 보통 / 보통	<input type="checkbox"/>
	app_counter	tps	없음	상승/하락 패턴 민감 / 민감	<input checked="" type="checkbox"/>
	app_counter(m5)	tx_count	demo-8100 demo-8101 demo-8102 demo-8103	비활성 / 보통	<input type="checkbox"/>

+ 이벤트 추가 버튼을 선택해 이상치 탐지 이벤트 설정을 진행할 수 있습니다. 모든 설정을 완료한 다음 저장 버튼을 선택하세요. 이상치 탐지 창의 제목 오른쪽에 토글 버튼을 선택해 활성화 여부를 선택할 수 있습니다.

# 이상치 탐지 이벤트 생성

### 이벤트 설정

애플리케이션    메트릭스    히트맵 패턴    **이상치 탐지**    로그

이상치 탐지

이벤트 이름	카테고리	필드
...	app_counter(m5)	tx_error
...	java_memory(m5)	heap_used
...	app_host_resource(m5)	swap
...	app_counter	tps
...	app_counter(m5)	tx_count

✕ 이상치 탐지
켜짐

▼ 이벤트 기본 정보

레벨 정보 경고 위험

\* 메시지   
알림 발생 시 알림 내용에 포함됩니다.

▼ 메트릭스

\* 카테고리

\* 필드

필터   
이벤트 대상을 필터링합니다.

오브젝트 병합 없음 합산 평균

▼ 이상치 탐지

상승 패턴  둔감 보통 민감

하락 패턴  둔감 보통 민감

저장

## 이벤트 기본 정보

이상치 탐지 이벤트의 기본 정보를 입력하세요.

- **레벨**: 정보, 경고, 위험 수준 중 하나의 레벨을 선택하세요.
- **메시지**: 이벤트 발생 시 출력하는 알림 메시지를 입력합니다. `${태그 또는 필드키}` 입력으로 메시지에 변수를 적용할 수 있습니다. 변수에 입력할 키는 선택한 메트릭스 데이터 **카테고리**에 포함된 값이어야 합니다.

## 메트릭스

이벤트 발생 대상을 선택하세요.

- **카테고리**: 메트릭스 데이터를 구분하는 단위입니다. 메트릭스 이벤트 설정 시 필수 선택 값입니다.
- **필드**: 이벤트 발행 조건에 사용할 필드를 선택합니다. 다중 선택할 수 있습니다.
- **필터**: 이벤트 조건 대상을 선택합니다. 다중 선택할 수 있습니다.
- **오브젝트 병합**: 오브젝트 병합 방법을 선택할 수 있습니다.

## 이상치 탐지

상승 패턴 또는 하락 패턴의 민감도를 선택할 수 있습니다. 각 항목의 토글 버튼을 선택해 기능을 켜거나 끌 수 있습니다.

## 이상치 탐지 이벤트 수정

1. 경고 알림 > 이벤트 설정 메뉴에서 이상치 탐지 탭을 선택하세요.
2. 이상치 탐지 목록에서 수정하려는 항목의 오른쪽에 ✎ 버튼을 선택하세요.
3. 이상치 탐지 설정 창이 나타납니다. 수정이 필요한 항목을 수정한 다음 저장 버튼을 선택하세요.

## 이상치 탐지 이벤트 삭제

경고 알림 > 이벤트 설정 메뉴에서 이상치 탐지 탭을 선택하세요. 목록에서 삭제하려는 항목의 오른쪽에 🗑 버튼을 선택하세요. 확인 팝업 메시지가 나타나면 삭제 버튼을 선택하세요.

# 로그 경고 알림

- ❗ 이 기능은 로그 모니터링을 활성화해야 이용할 수 있습니다. [로그 모니터링 활성화](#)에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.
- 로그 모니터링의 주요 메뉴에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

홈 화면 > 프로젝트 선택 > 경고 알림 > 이벤트 설정 > 로그 탭

수집한 로그 데이터를 조건에 맞춰 필터링해 경고 알림을 설정할 수 있습니다. + 이벤트 추가 버튼을 선택해 로그 이벤트 경고 알림을 설정하세요. 모든 설정을 완료한 다음 저장 버튼을 선택하세요.

### 이벤트 설정

실시간 로그 이벤트
JSON
+ 이벤트 추가

이벤트 이름	카테고리	검색 키	검색 값	이벤트 발생 일시 중지	이벤트 대상 필터링	이벤트 수신 태그	설정
실시간 로그 이벤트	AppLog	onodeName	node-0	20분	onode == '334634079' && okindName != 'demo-okind-1'	test-tag	🗑️ <input type="checkbox"/>
이벤트 테스트	AppLog	oid	-1128904592	사용 안 함	사용 안 함	전체 멤버 수신	🗑️ <input checked="" type="checkbox"/>
test	AppLog	oname	demo-8100	사용 안 함	oname && 'demo-8100'	전체 멤버 수신	🗑️ <input checked="" type="checkbox"/>

복합 로그 이벤트
JSON
+ 이벤트 추가

이벤트 이름	템플릿	카테고리	규칙	이벤트 상태가 해결되면 추가 알림	이벤트 발생 일시 중지	인터벌	이벤트 수신 태그	설정
이벤트 이름_테스트	2xx 상태코드 건수 <code>count</code>	AppLog	include_minus_exclude_count > 10	Off	1분	5분	전체 멤버 수신	🗑️ <input checked="" type="checkbox"/>
retreter	사용 안 함	#WhatapEvent	rows > 10	Off	1분	5분	전체 멤버 수신	🗑️ <input checked="" type="checkbox"/>
test	사용 안 함	*	rows > 10	Off	1분	5분	전체 멤버 수신	🗑️ <input checked="" type="checkbox"/>

추가할 수 있는 로그 이벤트 다음과 같습니다.

- 실시간 로그 이벤트** : 실시간으로 수집한 로그에서 검색 값이 등장하면 경고 알림을 보냅니다.
- 복합 로그 이벤트** : 최근에 수집한 로그 중 일정 조건을 만족하는 로그가 일정 개수 이상 수집한 경우에 경고 알림을 보냅니다.

- ① 이 이벤트를 추가하거나 설정하려면 **알림 설정** 권한이 있어야 합니다. 사용자별 권한에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.
- 경고 알림과 관련해 모니터링 플랫폼별 지원되는 이벤트 종류를 확인하려면 [다음 문서](#)를 참조하세요.

## 이벤트 추가 공통 옵션

다음은 이벤트 추가 시 공통으로 설정할 수 있는 옵션입니다.

- 이벤트 이름** : 추가하려는 이벤트 이름을 입력하세요.
- 이벤트 활성화** : 토글 버튼을 클릭해 경고 알림 활성화 여부를 선택할 수 있습니다.
- 레벨** : **위험**, **경고**, 정보 중 하나의 레벨을 선택하세요.
- 메시지** : 이벤트 발생 시 출력하는 알림 메시지를 입력합니다. `#{태그 또는 필드키}` 입력으로 메시지에 변수를 적용할 수 있습니다. 변수에 입력할 키는 선택한 매트릭스 데이터 **카테고리**에 포함된 값이어야 합니다.
- 카테고리** : 로그 구분 명칭(로그 폴더명)을 목록에서 선택하거나 직접 입력할 수 있습니다.
- 이벤트 발생 일시 중지** : 과도한 경고 알림 발생을 방지할 수 있는 옵션입니다. 첫 번째 경고 알림 이후 선택한 시간 동안 경고 알림을 보내지 않습니다. 또한 **이벤트 기록** 메뉴에 기록되지 않습니다.
- 이벤트 수신 태그** : 이벤트 수신 태그를 선택하면 해당 태그를 가진 프로젝트 멤버와 3rd-party 플러그인에 알림을 전송할 수 있습니다. 이벤트 수신 태그를 선택하지 않으면 프로젝트 전체 멤버에게 경고 알림을 보냅니다.

태그를 추가하지 않으면 전체 멤버에게 경고 알림을 보냅니다. **+ 태그 추가**를 클릭한 다음 **태그 목록**에서 경고 알림 수신 대상을 선택하세요. **+ 새 태그 생성**을 선택해 태그를 추가할 수도 있습니다.

- ① **경고 알림 > 이벤트 수신 설정** 메뉴에서 프로젝트 멤버와 3rd-party 플러그인에 태그를 설정할 수 있습니다. **이벤트 수신 설정** 메뉴에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

## 실시간 로그 이벤트 추가



## 이벤트 추가

이벤트 이름 \*

이벤트 활성화



레벨 \*

위험 경고 정보

메시지 \*

카테고리 ⓘ \*

검색 키 \*

검색 값 \*

입력된 단어가 일치하는 경우 알림이 발생합니다.

이벤트 대상 필터링 ⓘ

선택 입력  직접 입력

입력값이 없을 경우, 실시간으로 수정

선택 입력  직접 입력

검색 키 ▾

연산자 ▾

검색 값 ▾

+ 추가

ex. oid == '12345678' && level == 'ERROR'

이벤트 발생 일시 중지 \*

사용 안 함 ▾

알림 수신 후 선택한 시간 동안 이벤트가 발생하지 않습니다.

단, "이벤트 상태가 해결되면 추가 알림" 기능을 활성화한 경우에는 RECOVERED 알림 수신 후 선택한 시간 동안 이벤트가 발생하지 않습니다.

이벤트 수신 태그 ⓘ

전체 멤버 수신 + 태그 추가

[프로젝트 이벤트 수신설정 메뉴 바로가기](#)

저장

- **검색 키** : 로그 데이터 내에서 특정 값에 접근하기 위한 식별자를 의미합니다. 목록에서 선택하거나 직접 입력할 수 있습니다.

예시, HTTP 응답 상태 코드를 나타내는 값에 접근하고자 할 경우 **검색 키** `status`

- **검색 값** : **검색 키**에 해당하는 실제 데이터를 의미합니다. 로그에서 입력한 단어를 포함할 경우 경고 알림을 보냅니다. 목록에서 선택하거나 직접 입력할 수 있습니다.

예시, **검색 키** `status` **검색 값** `200`을 설정한 경우 HTTP 응답 상태 코드 200을 포함하는 로그 데이터 수집 시 경고 알림 발생

- **이벤트 대상 필터링** : **선택 입력** 옵션을 통해 **검색 키**와 **연산자**, **검색 값**을 선택해 대상을 필터링하거나 **직접 입력** 옵션을 선택할 수 있습니다. 입력값이 없을 경우 실시간으로 수집하는 데이터 전체에 대한 알림 발생 여부를 판단합니다.

예시, `AppLog` 카테고리의 로그 중 `level` 이 `ERROR`인 로그를 찾습니다. 일치하는 로그 중에서 `oid`가 `12345678`인 경우 경고 알림을 보냅니다.

# 복합 로그 이벤트 추가

✕
이벤트 추가

**\*이벤트 이름**

**이벤트 활성화**

**레벨** 위험 경고 정보 이벤트 상태가 해결되면 추가 알림 ⓘ

---

**템플릿** 사용 안 함

**\*메시지**

**\*카테고리 ⓘ**

**사용 안 함**

- 2xx 상태코드 건수 count
- 3xx 상태코드 건수 count
- 4xx 상태코드 건수 count
- 5xx 상태코드 건수 count
- 정상 상태코드(2xx,3xx) 건수 count
- 에러 상태코드(4xx,5xx) 건수 count
- 에러 수신 건수 count

**데이터 조회 범위** 최근에  ▼  
선택 시간동안 수집된 로그를 조회합니다.

**로그 검색 조건 ⓘ**  **선택 입력**  **직접 입력**

검색 키

▼

검색 값

▼

제외 —

+ 추가

이벤트 발행 조건을 입력하기 위해서 카테고리를 먼저 선택해 주세요.

**\*이벤트 발행 조건** 조건에 맞는 로그 > ▼

---

**인터벌 ⓘ**  ▼

**무음 ⓘ**  ▼

**이벤트 수신 태그 ⓘ** 전체 멤버 수신 + 태그 추가

🔗
프로젝트 이벤트 수신설정 메뉴 바로가기

저장

- **템플릿** : 복합 로그 템플릿을 제공합니다.
- **로그 검색 조건**
  - **검색 키**에서 이벤트 발생 조건 대상을 선택할 수 있습니다. 선택한 **검색 키**에 해당하는 검색 값을 선택할 수 있습니다.
  - **검색 키**에서 동일한 항목을 추가할 경우 'OR' 조건으로, 다른 항목을 추가할 경우 'AND' 조건으로 동작합니다.
  - **제외** 체크 박스를 선택해 선택한 검색 값을 이벤트 발행 조건에서 제외할 수 있습니다.
  - **+ 추가**를 선택해 여러개의 이벤트 발행 조건을 추가 또는 제외 설정할 수 있습니다.
- **데이터 조회 범위** : 선택한 시간동안 수집한 로그를 조회합니다. 데이터 조회 시간을 5분으로 선택하면 최근 5분 동안 수집한 데이터를 조회해서 이벤트 발생 조건을 확인합니다.
- **이벤트 발행 조건** : 이벤트가 입력한 횟수와 선택한 연산자의 조건과 같이 발생할 때 경고 알림을 보냅니다.

예시, AppLog 카테고리의 로그 중 조건 입력에 해당하는 로그를 필터링 합니다. 조건 입력에서 제외를 체크한 경우 해당 조건으로 찾은 로그를 제외하겠다는 의미입니다. 따라서 level 이 ERROR 인 로그는 제외합니다. 최근 10분 동안 수집한 로그 중 이벤트가 5 보다 작을 경우 경고 알림을 보냅니다.

## 로그 이벤트 설정 수정하기

1. 경고 알림 > 이벤트 설정 메뉴로 이동하세요.
2. 로그 탭을 선택하세요.
3. 로그 이벤트 목록 중 수정하려는 이벤트 항목에서 오른쪽에 ✎ 버튼을 선택하세요.
4. 이벤트 설정 창이 나타나면 옵션을 수정한 다음 저장 버튼을 선택하세요.

선택한 로그 이벤트를 삭제하려면 이벤트 설정 창에서 오른쪽 위에 삭제 버튼을 선택하세요.




## 로그 이벤트 끄기

1. 경고 알림 > 이벤트 설정 메뉴로 이동하세요.
2. 로그 탭을 선택하세요.
3. 로그 이벤트 목록 중 경고 알림을 끄려는 이벤트 항목의 가장 오른쪽에 토글 버튼을 선택하세요.

다시 토글 버튼을 선택하면 해당 경고 알림이 활성화됩니다.

## 로그 이벤트 내보내기/불러오기

로그 이벤트의 설정 내용을 json 파일로 저장한 다음 불러와 재사용할 수 있습니다.

1. 경고 알림 > 이벤트 설정 메뉴로 이동하세요.
2. 로그 탭을 선택하세요.
3. 로그 이벤트 목록 위에 JSON  버튼을 선택하세요. JSON 내보내기 창이 나타납니다.
4. 내보내기 할 항목을 수정 또는 편집하세요.
5. 오른쪽 위에  내보내기 버튼을 선택하세요. 브라우저에서 json 파일을 다운로드합니다.
6. 로그 이벤트 목록 위에  버튼을 선택하세요.
7. 파일 선택 창이 나타나면 앞서 다운로드 받은 json 파일을 선택하세요.
8. JSON 가져오기 창이 나타나면 내용을 수정한 다음 + 목록에 추가하기 버튼을 선택하세요.

❗ 이벤트에 id가 존재합니다. id를 제거한 뒤 다시 시도하세요.

- 메시지가 나타나면 JSON 가져오기 창에서 id 항목을 삭제한 다음 + 목록에 추가하기 버튼을 선택하세요.
- 기존의 이벤트 항목에 덮어쓰기를 하려면 id 항목을 삭제한 다음 덮어쓰기 버튼을 선택하세요.

# 경고 알림 수신 설정

홈 화면 > 프로젝트 선택 > 경고 알림 > 이벤트 수신 설정

프로젝트 멤버들의 경고 알림 수신과 관련한 다양한 기능을 설정할 수 있습니다.

이벤트 수신 설정



▼ 사용자별 이벤트 수신 설정 (64)

이벤트 알림의 일괄 수신설정 및 접근 설정을 위한 모바일 기기 관리는 [계정 정보 메뉴](#)에서 가능합니다. [계정 정보 >](#)

이름	이메일 알림	SMS 알림	WhatsApp 알림	모바일 알림	반복 알림 (에스컬레이션)	이벤트 수신 태그
JH	<input checked="" type="checkbox"/> 수신 레벨: 위임	<input type="checkbox"/> 수신 레벨: 위임	등록된 번호가 없습니다.	<input type="checkbox"/> 모바일 기기 6대 수신 레벨: 전체	0 저장	<input type="checkbox"/> 수신 태그 미설정 알림 받기 testtag 테스트 태그 +
JT	<input checked="" type="checkbox"/> 수신 레벨: 전체	<input type="checkbox"/> 수신 레벨: 전체	등록된 번호가 없습니다.	등록된 기기가 없습니다.	0 저장	<input checked="" type="checkbox"/> 수신 태그 미설정 알림 받기 + 태그 추가
JU	<input checked="" type="checkbox"/> 수신 레벨: 전체	등록된 번호가 없습니다.	등록된 번호가 없습니다.	등록된 기기가 없습니다.	0 저장	<input checked="" type="checkbox"/> 수신 태그 미설정 알림 받기 + 태그 추가
KJ	<input type="checkbox"/> 수신 레벨: 전체	<input type="checkbox"/> 수신 레벨: 전체	등록된 번호가 없습니다.	등록된 기기가 없습니다.	0 저장	<input checked="" type="checkbox"/> 수신 태그 미설정 알림 받기 + 태그 추가
KY	<input checked="" type="checkbox"/> 수신 레벨: 전체	<input type="checkbox"/> 수신 레벨: 전체	등록된 번호가 없습니다.	등록된 기기가 없습니다.	0 저장	<input checked="" type="checkbox"/> 수신 태그 미설정 알림 받기 + 태그 추가
	<input checked="" type="checkbox"/> 수신 레벨: 전체	<input type="checkbox"/> 수신 레벨: 전체	등록된 번호가 없습니다.	등록된 기기가 없습니다.	0 저장	<input checked="" type="checkbox"/> 수신 태그 미설정 알림 받기

3rd 파티 플러그인

Slack, Telegram, Teams, Jandi, Webhook 등을 이용하여 알림 메시지를 받으실 수 있습니다.

플러그인 이름	인증 키	인증 값	반복 알림 (에스컬레이션)	이벤트 수신 태그
SLACK			0 저장	<input checked="" type="checkbox"/> 수신 태그 미설정 알림 받기 + 태그 추가
SLACK			0 저장	<input checked="" type="checkbox"/> 수신 태그 미설정 알림 받기 + 태그 추가
SLACK			0 저장	<input checked="" type="checkbox"/> 수신 태그 미설정 알림 받기 + 태그 추가
SLACK			0 저장	<input checked="" type="checkbox"/> 수신 태그 미설정 알림 받기 + 태그 추가
TELEGRAM			0 저장	<input checked="" type="checkbox"/> 수신 태그 미설정 알림 받기 PREV_TEST +
WEBHOOK_JSON			0 저장	<input checked="" type="checkbox"/> 수신 태그 미설정 알림 받기 + 태그 추가

➕ 추가하기

대량 알림 발생 방지

알림이 대량으로 발생하면 지정된 시간 동안 알림이 일시적으로 중지됩니다.  
대량 알림 차단 기능을 해제하려면 (이메일 주소 옆의) '중단 해제' 버튼을 눌러주세요.

활성화

탐지 시간: 5분

탐지 횟수: 10

정지 시간: 3시간

저장

## 수신 수단 선택하기

이메일 알림 이외에 SMS, 모바일 알림을 선택할 수 있습니다. 원하는 알림 수신 수단의 체크 박스를 체크하면 경고 알림을 받을 수 있습니다. 알림 수신 수단의 체크 박스를 해제하면 경고 알림을 보내지 않습니다.

- ① • 이메일 알림은 회원 가입 시 입력한 이메일 주소로 알림을 보냅니다.
- 프로젝트 최고 관리자를 제외한 모든 사용자는 자신의 수신 설정만 변경할 수 있습니다.

## SMS 알림 수신 추가하기

SMS 알림 수신이 필요한 경우 [계정 관리](#)에서 사용자 전화번호를 설정하세요.

**사용자 전화번호**

전화번호는 경고 알림 문자에 사용됩니다. 전화번호를 변경하려면 SMS 인증이 필요합니다.

일반 휴대전화
WhatsApp

전화번호  인증 완료 번호 변경하기

일반 휴대전화는 **한국 휴대전화 번호만 등록** 가능합니다.

1. 화면 오른쪽 위에 프로필 아이콘을 선택하세요.
2. 팝업 메뉴가 나타나면 [계정 관리](#) 버튼을 선택하세요.
3. [사용자 전화번호](#) 섹션에서 [일반 휴대전화](#) 버튼을 선택하세요.
4. [전화번호](#)에 인증번호를 수신할 전화번호를 입력하세요.
5. [인증번호 전송](#) 버튼을 선택하세요.
6. 사용자의 휴대전화로 전송된 인증 번호를 입력하세요.
7. [인증하기](#) 버튼을 선택하세요.

- ① • 등록된 전화번호를 변경하려면 [번호 변경하기](#) 버튼을 선택한 다음 위의 과정을 반복하세요.
- SMS를 알림으로 수신할 수 있는 전화번호는 **한국 휴대전화 번호만** 등록할 수 있습니다.



## WhatsApp 알림 수신 추가하기

WhatsApp을 통해 알림을 수신할 수 있습니다.

**사용자 전화번호**

전화번호는 경고 알림 문자에 사용됩니다. 전화번호를 변경하려면 SMS 인증이 필요합니다.

일반 휴대전화   **WhatsApp**

전화번호   KR (+82)

1. 화면 오른쪽 위에 프로필 아이콘을 선택하세요.
2. 팝업 메뉴가 나타나면 **계정 관리** 버튼을 선택하세요.
3. **사용자 전화번호** 섹션에서 **WhatsApp** 버튼을 선택하세요.
4. **전화번호**에 인증번호를 수신할 전화번호를 입력하세요.
5. **인증번호 전송** 버튼을 선택하세요.
6. WhatsApp 애플리케이션으로 전송된 인증번호 6자리를 입력하세요.
7. **인증하기** 버튼을 선택하세요.

ⓘ 등록된 전화번호를 변경하려면 **번호 변경하기** 버튼을 선택한 다음 위의 과정을 반복하세요.

## 수신 레벨 선택하기

경고 알림 레벨에 따라 알림 수신 여부를 선택할 수 있습니다. **사용자별 이벤트 수신 설정** 섹션의 사용자 목록에서 **수신 레벨**을 **전체** 또는 **위험**을 선택하세요.

- **전체**: 모든 경고 알림을 수신할 수 있습니다.
- **위험**: 위험 레벨의 경고 알림만 수신할 수 있습니다.

## 요일 및 시간별 알람 설정하기

요일별, 시간별 알림 수신 여부를 선택할 수 있습니다. **사용자별 이벤트 수신 설정** 섹션의 사용자 목록에서 가장 왼쪽에 ▾ 버튼을 선택하세요. 경고 알림 수신을 원하는 요일을 선택하거나 시간을 입력하세요. 알림 수신 수단별로 설정할 수 있습니다.

▼ 사용자별 이벤트 수신 설정 (64)

이벤트 알림의 일괄 수신설정 및 접근 설정을 위한 모바일 기기 관리는 [계정 정보 메뉴에서](#) 가능합니다. [계정 정보 >](#)

이름	이메일 알림	SMS 알림	WhatsApp 알림	모바일 알림	반복 알림 (에스컬레이션)	이벤트 수신 태그
HS	<input checked="" type="checkbox"/> 수신 레벨: 전체	<input type="checkbox"/> 0***** 등록된 번호가 없습니다.	<input type="checkbox"/>	<input type="checkbox"/> 모바일 기기 2대 수신 레벨: 전체	0 저장	<input checked="" type="checkbox"/> 수신 태그 미설정 알림 받기 + 태그 추가

알림 수신 언어: 한국어

이메일 알림	SMS 알림	모바일 알림				
요일: 월, 화, 수, 목, 금, 토, 일 시간: 00:00 ~ 00:00	요일: 월, 화, 수, 목, 금, 토, 일 시간: 00:00 ~ 00:00	요일: 월, 화, 수, 목, 금, 토, 일 시간: 00:00 ~ 00:00 모바일 기기: <table border="1"> <tr><td>테스트</td><td>삭제</td></tr> <tr><td>테스트</td><td>삭제</td></tr> </table>	테스트	삭제	테스트	삭제
테스트	삭제					
테스트	삭제					

## 경고 알림 반복 설정하기

경고 알림 발생 시간으로부터 알림 발생 상황이 해소되지 않을 경우 최초 알림 발생 시각으로부터의 알림 반복 간격을 설정할 수 있습니다. 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.

**반복 알림 (에스컬레이션) ⓘ**

0,1H,1D 저장

- ⓘ • 이 기능은 **Critical** (또는 **위험**) 등급의 모든 이벤트를 대상으로 합니다. 이벤트 추가 시 설정한 **레벨** 항목을 참조하세요.
- **M**: 분, **H**: 시간, **D**: 일, 단위를 생략하면 분 단위로 시간을 설정합니다.
- **저장** 버튼을 선택하지 않으면 설정을 완료할 수 없습니다.
- 숫자 또는 숫자+단위(**M**, **H**, **D**)로 입력하세요. 입력이 올바르지 않으면 메시지가 표시됩니다.

# 팀별, 사용자별 알림 설정하기

프로젝트의 멤버 중 특정 멤버 또는 팀을 대상으로 알림 수신 여부를 설정합니다. [메트릭스](#) 및 [이상치 탐지](#), [로그](#) 이벤트 설정의 [이벤트 수신 태그](#) 항목에서 태그를 추가하세요. 이벤트별로 경고 알림을 수신하는 멤버 또는 팀을 선택할 수 있습니다.

반대로 이벤트 수신 태그를 설정하지 않으면 전체 멤버에게 경고 알림을 보낼 수 있습니다.

> 이벤트 수신 태그 사용 예시

이벤트 수신 태그에 대한 사용 예시를 통해 팀별 또는 사용자별로 경고 알림을 전송하는 설정 방법에 대해 알아봅니다. 다음과 같이 팀 별로 서로 다른 경고 알림을 전송하도록 설정합니다.

알림 종류	서버팀 수신 여부	데브옵스팀 수신 여부
메트릭스 경고 알림	O	X
이상치 탐지 경고 알림	O	O
로그 실시간 경고 알림	X	O

## 1. 이벤트 수신 설정하기 (경고 알림 > 이벤트 수신 설정)

▼ 사용자별 이벤트 수신 설정 (64)

이벤트 알림의 일괄 수신설정 및 접근 설정을 위한 모바일 기기 관리는 [계정 정보 메뉴에서](#) 가능합니다. [계정 정보 >](#)

이름	이메일 알림	SMS 알림	WhatsApp 알림	모바일 알림	반복 알림 (에스컬레이션)	이벤트 수신 태그
DE	<input checked="" type="checkbox"/> 수신 레벨: 전체	등록된 번호가 없습니다.	등록된 번호가 없습니다.	등록된 기기가 없습니다.	0 저장	<input checked="" type="checkbox"/> 수신 태그 미설정 알림 받기 [Save] +
DE	<input checked="" type="checkbox"/> 수신 레벨: 전체	등록된 번호가 없습니다.	등록된 번호가 없습니다.	등록된 기기가 없습니다.	0 저장	<input checked="" type="checkbox"/> 수신 태그 미설정 알림 받기 [DevOps] +

프로젝트에 속한 멤버들 중 서버팀 소속은 `서버팀`, 데브옵스팀 소속은 `데브옵스팀` 으로 이벤트 수신 태그를 설정하세요.

## 2. 메트릭스 경고 알림 설정하기 (경고 알림 > 이벤트 설정 > 메트릭스 > 이벤트 추가 > 이벤트 수신 설정)

## 이벤트 수신 태그

Server +

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

[프로젝트 이벤트 수신설정 메뉴](#)

이벤트 설정 시 태그를 선택하지 않은 경우 프로젝트 이벤트 수신 설정 메뉴의 나머지 수신 조건(활성화 여부 등)에 따라 알림이 발생합니다.

[태그 추가](#) 또는 + 버튼을 클릭하세요. 태그 목록에서 원하는 태그를 선택하거나 새 태그를 생성하세요. 메트릭스 이벤트에 대한 알림을 `서버팀`으로 설정한 경우입니다.

### 3. 로그 실시간 경고 알림 설정 (경고 알림 > 이벤트 설정 > 로그 > 이벤트 추가 > 이벤트 수신 태그)

## 이벤트 수신 태그 ⓘ

DevOps +

[프로젝트 이벤트 수신설정 메뉴 바로가기](#)

[태그 추가](#) 또는 + 버튼을 클릭하세요. 태그 목록에서 원하는 태그를 선택하거나 새 태그를 생성하세요. 로그 실시간 이벤트에 대한 알림을 `데브옵스팀`으로 설정한 경우입니다.

### 4. 이상치 탐지 이벤트는 전체 멤버에게 경고 알림을 전송하므로 이벤트 수신 태그를 설정하지 않습니다.

## 이벤트 수신 태그 추가하기

1. [사용자별 이벤트 수신 설정](#) 섹션의 사용자 목록에서 [태그 추가](#) 또는 + 버튼을 선택하세요.
2. [이벤트 수신 태그](#) 팝업 창이 나타나면 태그 입력란에 태그 이름을 입력한 다음 엔터를 입력하거나 [새 태그 생성](#)을 선택하세요.
3. 태그 색상을 선택하세요.
4. [태그 생성](#) 버튼을 선택하세요.

[태그 목록](#)에서 생성한 태그를 확인할 수 있습니다. 생성한 태그를 적용하려면 해당 태그를 선택하세요.

## 이벤트 수신 태그 해제하기

1. [사용자별 이벤트 수신 설정](#) 섹션의 사용자 목록에서 + 버튼을 선택하세요.
2. [이벤트 수신 태그](#) 팝업 창이 나타나면 적용된 태그 옆에 × 버튼을 선택하세요.
3. [이벤트 수신 태그](#) 팝업 창을 닫으세요.

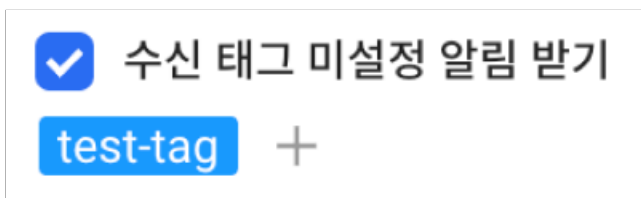
이벤트 수신 태그를 해제합니다.

## 이벤트 수신 태그 수정 및 삭제하기

1. **사용자별 이벤트 수신 설정** 섹션의 사용자 목록에서 **태그 추가** 또는 **+** 버튼을 선택하세요.
2. **이벤트 수신 태그** 팝업 창이 나타나면 **태그 목록**에서 수정 또는 삭제할 항목의 **✎** 버튼을 선택하세요.
3. 태그 이름을 수정하거나 색상을 변경한 다음 **적용** 버튼을 선택하세요.  
태그를 삭제하려면 **🗑️ 태그 삭제** 버튼을 선택하세요.

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

## 수신 태그 미설정 알림



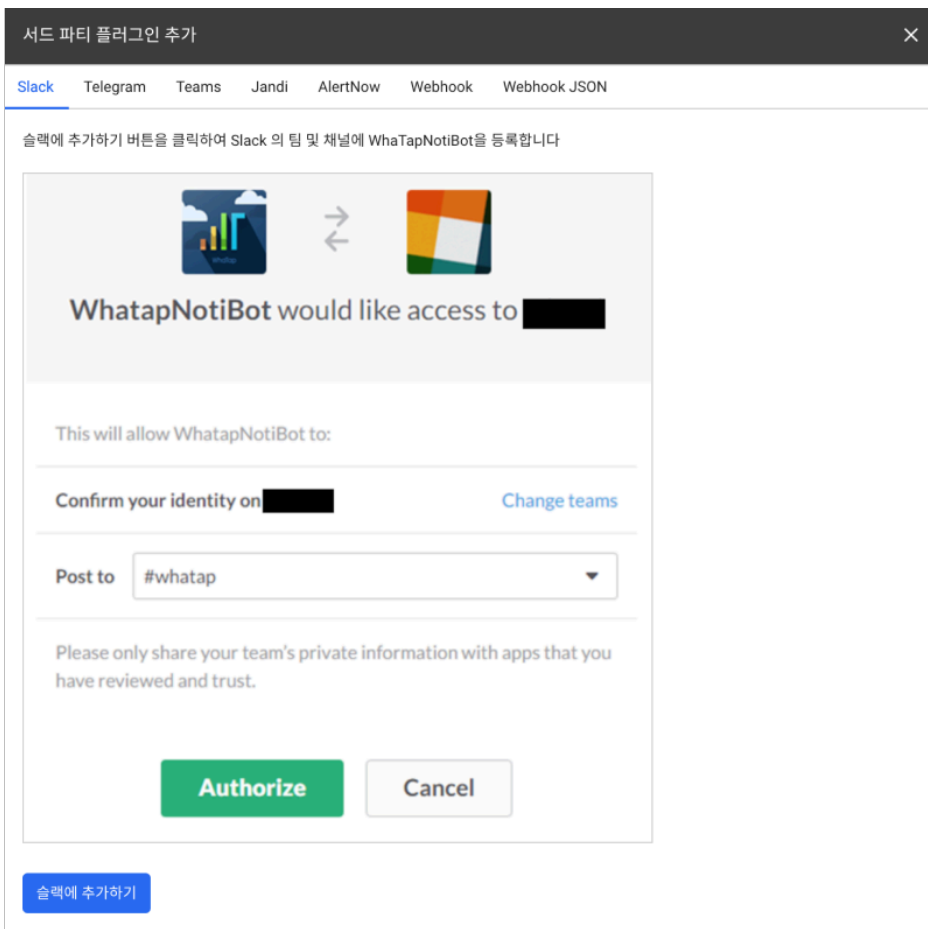
이벤트 수신 태그가 설정되지 않은 경고 알림을 받으려면 **수신 태그 미설정 알림 받기** 옵션을 선택하세요. **이벤트 수신 태그**가 설정된 경고 알림만 받고 싶다면 선택을 해제하세요.

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

## 3rd 파티 플러그인 알림 추가하기

Slack, Telegram, Teams, Jandi, Webhook 등의 외부 애플리케이션을 통해 경고 알림을 받을 수 있습니다.

1. **경고 알림 > 이벤트 수신 설정** 메뉴에서 **3rd 파티 플러그인** 섹션의 **추가하기** 버튼을 선택하세요.
2. 원하는 서비스를 선택하세요.



3. 선택한 서비스의 화면 안내에 따라 설정을 진행하세요.
4. 모든 과정을 완료했다면 추가 버튼을 선택하세요.

ⓘ 와탭랩스의 지원 범위에 포함하지 않는 사내 메신저는 표준 Webhook, webhook json을 통해 연동할 수 있습니다.

## 대량 알림 발생 방지

알림이 대량으로 발생하면 설정한 시간 동안 경고 알림을 일시적으로 중단합니다. [경고 알림](#) > [이벤트 수신 설정](#) 메뉴에서 [대량 알림 발생 방지](#) 섹션으로 이동하세요.

## 대량 알림 발생 방지

알림이 대량으로 발생하면 지정한 시간 동안 알림이 일시적으로 중지됩니다.  
대량 알림 차단 기능을 해제하려면 (이메일 주소 옆의) '중단 해제' 버튼을 눌러주세요.

활성화	<input checked="" type="checkbox"/>
탐지 시간	5분 <input type="button" value="v"/>
탐지 횟수	10
정지 시간	3시간 <input type="button" value="v"/>

저장

- **활성화** 토글 버튼을 선택해 기능을 켜거나 끌 수 있습니다.
- **탐지 시간** 동안 **탐지 횟수** 이상의 이벤트가 발생하면 **정지 시간** 동안 경고 알림을 중지합니다.

예를 들어, 5분 동안 20회의 이벤트가 발생하면 5분 동안 경고 알림을 중지합니다. 설정한 **정지 시간** 시간이 지나면 대량 알림 발생 방지 기능은 해제됩니다.

**!** 문자 메시지 알림이 하루 200건 이상 발생하면 일시 중지하며 다음 메시지를 표시합니다. 문자 알림 중단 기능을 해제하려면 **문자알림 중단 해제** 버튼을 선택하세요.

다량의 문자메시지가 전송되었습니다. (200건 / 일)

- 프로젝트에서 발생한 많은 양의 문자 메시지로 인해 문자 메시지 알림이 중지됩니다.
- 시작 시간 : 2024-02-02 18:40:01

문자알림 중단 해제

## 경고 알림 사용자 설정하기

**계정 관리** 메뉴에서 사용자 개인의 알림 수신 레벨, 수신 수단, 요일 및 시간 등을 설정할 수 있습니다.

1. 화면의 오른쪽 위에 프로필 아이콘을 선택하세요.
2. 팝업 메뉴가 나타나면 **계정 관리** 버튼을 선택하세요.

3. 화면을 아래로 스크롤해 [알림 수신 설정](#) 섹션으로 이동하세요.
4. 수신 레벨, 수신 수단, 요일 및 시간을 설정한 다음 [저장](#) 버튼을 선택하세요.

## 경고 알림 언어 설정


프로젝트에서 발생하는 경고 알림 메시지의 언어를 변경할 수 있습니다.

1. 홈 화면에서 경고 알림 메시지의 언어를 변경할 프로젝트를 선택하세요.
2. 화면 왼쪽 메뉴에서 [관리](#) > [프로젝트 관리](#)를 선택하세요.
3. [알림 언어 관리](#) 섹션에서 원하는 언어를 선택하세요.
4. 화면 오른쪽 아래에 [저장](#) 버튼을 선택하세요.



# 이벤트 기록

홈 화면 > 프로젝트 선택 > 경고 알림 > 이벤트 기록

경고 알림이 발생한 이력을 확인할 수 있습니다. 최근 1년 이내의 이력까지 조회할 수 있습니다. 각 항목을 설정한 다음  버튼을 선택하세요.

이벤트 기록					
시간 선택		필터		애플리케이션	
<input type="text" value="2024/01/22 00:00 ~ 2024/01/23 00:00"/> <input type="button" value="1일"/>		<input type="text" value="제목"/> <input type="button" value="전체 선택"/> <input type="button" value="Q"/> <input type="button" value="CSV"/>			
번호	제목	이벤트 발생 시간	이벤트 해소 시간	애플리케이션	메시지
1	HITMAP_FLOOD_PATTERN	2024/01/22 09:08:00	2024/01/22 09:09:06		Flood pattern was detected in the project hitmap.
2	HITMAP_FLOOD_PATTERN	2024/01/22 08:57:00	2024/01/22 08:58:04		Flood pattern was detected in the project hitmap.
3	HITMAP_FLOOD_PATTERN	2024/01/22 08:46:00	2024/01/22 08:47:03		Flood pattern was detected in the project hitmap.
4	HITMAP_FLOOD_PATTERN	2024/01/22 08:35:00	2024/01/22 08:36:03		Flood pattern was detected in the project hitmap.
5	HITMAP_FLOOD_PATTERN	2024/01/22 08:24:00	2024/01/22 08:25:00		Flood pattern was detected in the project hitmap.
6	HITMAP_FLOOD_PATTERN	2024/01/22 08:13:00	2024/01/22 08:13:59		Flood pattern was detected in the project hitmap.
7	HITMAP_FLOOD_PATTERN	2024/01/22 08:02:00	2024/01/22 08:02:57		Flood pattern was detected in the project hitmap.
8	HITMAP_FLOOD_PATTERN	2024/01/22 07:51:00	2024/01/22 07:51:56		Flood pattern was detected in the project hitmap.
9	HITMAP_FLOOD_PATTERN	2024/01/22 07:40:00	2024/01/22 07:40:55		Flood pattern was detected in the project hitmap.
10	HITMAP_FLOOD_PATTERN	2024/01/22 07:29:00	2024/01/22 07:29:54		Flood pattern was detected in the project hitmap.
11	HITMAP_FLOOD_PATTERN	2024/01/22 07:18:00	2024/01/22 07:18:53		Flood pattern was detected in the project hitmap.
12	HITMAP_FLOOD_PATTERN	2024/01/22 07:07:00	2024/01/22 07:07:52		Flood pattern was detected in the project hitmap.
13	HITMAP_FLOOD_PATTERN	2024/01/22 06:56:00	2024/01/22 06:56:51		Flood pattern was detected in the project hitmap.
14	HITMAP_FLOOD_PATTERN	2024/01/22 06:45:00	2024/01/22 06:45:50		Flood pattern was detected in the project hitmap.
15	HITMAP_FLOOD_PATTERN	2024/01/22 06:34:00	2024/01/22 06:34:49		Flood pattern was detected in the project hitmap.
16	HITMAP_HORIZONTAL_PATTERN	2024/01/22 06:28:00	2024/01/22 06:28:48		Horizontal pattern was detected in the project hitmap.
17	HITMAP_FLOOD_PATTERN	2024/01/22 06:18:00	2024/01/22 06:18:46		Flood pattern was detected in the project hitmap.

ⓘ 선택한 프로젝트에 따라 화면 이미지는 다를 수 있습니다.

## • 시간 선택

- 오른쪽에 위치한 녹색 버튼을 선택해 조회 시간을 선택할 수 있습니다.
- < 또는 > 버튼을 선택해 선택한 조회 시간 만큼 간격을 이동할 수 있습니다.
- 세부 시간을 선택하려면 날짜 또는 시간 영역을 선택하세요. 세부 시간을 설정한 다음 **적용** 버튼을 선택하세요.

- **필터**: 제목 또는 메시지 내용을 기준으로 이벤트 기록을 필터링할 수 있습니다.

- **애플리케이션**: 프로젝트에 포함된 에이전트를 선택할 수 있습니다.
- **CSV**: 조회한 이벤트 기록 결과를 csv 파일로 저장할 수 있습니다. **CSV** 버튼을 클릭하면 **최대 CSV 라인 수**를 입력한 다음 **다운로드** 버튼을 선택하세요.
- **컬럼 선택**: 조회한 이벤트 기록 결과의 열 항목을 추가할 수 있습니다.
- **이벤트 설정**: **경고 알림 > 이벤트 설정** 메뉴로 이동합니다.
- **제목 / 메시지**: **이벤트 설정** 메뉴에서 추가한 이벤트의 **이벤트명**과 **메시지** 항목의 내용입니다.

- ① ○ 이벤트 추가에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.
- 알림 메시지 사용자 정의에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

- **이벤트 발생 시각**: 이벤트가 발생한 시각입니다.
  - 이벤트가 해소되지 않고 진행 중일 경우 **진행 중** 태그가 표시됩니다.
  - 정비 중인 경우 이벤트가 발생하면 **정비 중 발생** 태그가 표시됩니다.

- ① **정비 계획**에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

- **이벤트 해소 시각**: 설정한 이벤트가 해결된 시각입니다. 만약 해당 컬럼이 보이지 않는다면 **컬럼 선택** 버튼을 클릭한 다음 **이벤트 해소 시각**을 선택하세요.
- **애플리케이션**: **이벤트 설정** 메뉴에서 이벤트 추가 시 **이벤트 대상 필터링** 항목을 설정하면 표시됩니다.

# 이벤트 수신 포맷

## 공용 알림 템플릿

공통 템플릿은 대부분의 경우 적용 가능합니다. 다양한 환경에서 동일한 포맷으로 알림을 받을 수 있습니다.

- Product Type : 애플리케이션, 데이터베이스, 쿠버네티스
- Event Type : 애플리케이션 알림, 데이터베이스 알림, 메트릭스 알림
- Event Channel : sms, mobile, 3rd party plugin , plugin

## 이벤트 제목

### Event title format

```
[Level][Platform][ProjectName][ApplicationName][EventTitle]
```

### Event title example

```
[Info][JAVA][애플리케이션 프로젝트][TC-0-1-8081][CRITICAL_HIGH_MEMORY]
```

- ⓘ • ApplicationName을 설정하지 않은 경우 이벤트 제목에서 생략합니다.
- Platform은 다음 중 한 가지로 표시합니다.
  - JAVA
  - NODEJS
  - PYTHON
  - PHP
  - DOTNET
  - GO
  - POSTGRESQL
  - ORACLE
  - MYSQL
  - MSSQL
  - BSM\_JAVA
  - CLOUDWATCH
  - TIBERO

- ! ○ KUBERNETES
- KUBE\_NS
- URLCHECK
- URLCHECK\_ADMIN
- CUBRID
- ALTIBASE
- CLUSTER
- REDIS
- MONGODB
- VR
- RUM

## 이벤트 메시지

이벤트 메시지에 포함할 수 있는 정보입니다. `optional` 이 `false` 인 경우 항상 메시지에 포함합니다. `optional` 이 `true` 인 경우 해당 데이터를 확인할 수 있으면 표시합니다.

### Event message example

```
Project Name : 애플리케이션 프로젝트
Project Code : 3
Application Name : TC-0-1-8081
Event Message : RECOVERED: Memory is too high. less than 10%
Event ON Time : 2022-04-12 18:53:24 +0900
Event OFF Time : 2022-04-12 18:53:24 +0900
Alert Type : APPLICATION_MEMORY
Metric Name : memory
Metric Value : 20
Metric Threshold : 10
Stateful : true
```

다음은 이벤트 메시지 구성 요소입니다.

En	Ko	지원되는 알림 타입	설명
Project Name	프로젝트 이름	전체	-
Project Code	프로젝트 Code	전체	-

En	Ko	지원되는 알림 타입	설명
Application Name	에이전트 이름	전체(optional)	oname
Event Message	이벤트 메시지	전체	-
Alert Type	이벤트 종류	전체	아래의 AlertType 종류 표 참고
Event ON Time	이벤트 발생 시간	전체	2022-04-13 10:40:49 +0900에서 +0900는 GMT를 의미합니다.
Event OFF Time	이벤트 해제 시간	전체(optional)	2022-04-13 10:40:49 +0900에서 +0900는 GMT를 의미합니다.
Metric Name	메트릭스 이름	전체(optional)	이벤트 조건 판단에서 사용하는 메트릭스의 이름
Metric Value	메트릭스 값	전체(optional)	메트릭스 값이 메트릭스 임계치를 넘으면 이벤트 발생 조건이 만족한 경우입니다.
Metric Threshold	메트릭스 임계치	전체(optional)	메트릭스 값이 메트릭스 임계치를 넘으면 이벤트 발생 조건이 만족한 경우입니다.
Stateful	해결된 이벤트 알림	전체(optional)	해결된 이벤트 알림 기능 사용 중이면 true, 아니면 false
Event Rule	이벤트 발생 조건	메트릭스 알림	-
Event Target Filter	이벤트 대상 선택	메트릭스 알림	특정 대상에서 수집된 메트릭스에 대해서만 이벤트 조건을 확인합니다.
Repeat Count	이벤트 반복 횟수	메트릭스 알림	이벤트 조건이 이벤트 반복 시간동안 이벤트 반복 횟수만큼 만족해야 이벤트가 발생합니다.
Repeat Duration	이벤트 반복 시간	메트릭스 알림	이벤트 조건이 이벤트 반복 시간동안 이벤트 반복 횟수만큼 만족해야 이벤트가 발생합니다.
Receiver	수신자	메트릭스 알림	-
Query	MXQL 쿼리	복합 메트릭스 알림	-
Rule	이벤트 발생 조건	복합 메트릭스 알림	-

En	Ko	지원되는 알림 타입	설명
Query Period	쿼리 기간	복합 메트릭스 알림	-
Query Interval	쿼리 간격	복합 메트릭스 알림	-
Silent Time	무음 시간	복합 메트릭스 알림	-
Query	URL	Exception 알림	Exception을 발생시킨 요청의 URL
TXID	트랜잭션 ID	Exception 알림	-
Class	에러 클래스 이름	Exception 알림	-
Log Message	로그 메시지	서버 - 파일 로그 알림	-
Log File	로그 파일 경로	서버 - 파일 로그 알림	-
IP	IP	서버 알림 전체	-
CPU	CPU	서버 알림 전체	이벤트 발생 당시의 Snapshot
CPU_load1	CPU_load1	서버 알림 전체	이벤트 발생 당시의 Snapshot
CPU_loadPerCore	CPU_loadPerCore	서버 알림 전체	이벤트 발생 당시의 Snapshot
Memory	Memory	서버 알림 전체	이벤트 발생 당시의 Snapshot
Swap	Swap	서버 알림 전체	이벤트 발생 당시의 Snapshot
Disk Name   Used Percent   Free Size   IO Percent	디스크 퍼포먼스	서버 알림 전체	이벤트 발생 당시의 Snapshot
Name   Bps   Pps	트래픽 퍼포먼스	서버 알림 전체	이벤트 발생 당시의 Snapshot

En	Ko	지원되는 알림 타입	설명
Message   Time   Name	처리내역 메시지	서버 알림 전체	-

- ! • 해당 이벤트에서 제공할 수 있는 최대한 많은 정보를 보여줍니다.
- AlertType은 다음 중 한 가지로 표시합니다.

AlertType	설명
APPLICATION_CPU	애플리케이션 CPU 알림
APPLICATION_MEMORY	애플리케이션 MEMORY 알림
APPLICATION_DISK	애플리케이션 DISK 알림
APPLICATION_ACTIVE_TRANSACTION	애플리케이션 액티브 트랜잭션 알림
APPLICATION_ERROR_TRANSACTION	애플리케이션 에러 트랜잭션 알림
APPLICATION_SLOW_TRANSACTION	애플리케이션 트랜잭션 응답시간 알림
METRICS	메트릭스 알림
COMPOSITE_METRICS	복합 메트릭스 알림
ANOMALY	이상치 탐지 알림
LOG_REALTIME	로그 실시간 알림
COMPOSITE_LOG	복합 로그 알림
SERVER_REBOOT	서버 - 재시작 알림
SERVER_NO_DATA	서버 - 미수신 알림
SERVER_PORT	서버 - 포트 알림
SERVER_NETWORK_IOPS	서버 - 네트워크 IOPS 알림
SERVER_NETWORK_BPS	서버 - 네트워크 BPS 알림



AlertType	설명
SERVER_DISK_IO	서버 - 디스크 I/O 알림
SERVER_DISK_QUOTA	서버 - 디스크 사용량 알림
SERVER_DISK_INODE	서버 - inode 알림
SERVER_CPU	서버 - CPU 알림
SERVER_MEMORY	서버 - 메모리 알림
SERVER_CPU_STEAL	서버 - steal 알림
SERVER_MEMORY_SWAP	서버 - 스왑 알림
SERVER_LOG_FILE	서버 - 로그 파일 알림
SERVER_WINDOW_EVENT	서버 - 윈도우 이벤트 알림
SERVER_OFF	서버 - 알림 OFF 알림
SERVER_ACKNOWLEDGE	서버 - 처리내역 알림
SERVER_PROCESS_COUNT	서버 - 프로세스 수 알림
SERVER_PROCESS_CPU	서버 - 프로세스 CPU 알림
SERVER_PROCESS_MEMORY	서버 - 프로세스 메모리 알림
SERVER_PROCESS_OFF	서버 - 프로세스 알림 OFF 알림
AGENT_ACTIVE	에이전트 활성화 알림
AGENT_INACTIVE	에이전트 비활성화 알림
AGENT_REACTIVATED	에이전트 재활성화 알림
URL	URL 알림
TOO_MANY_EVENT	너무 많은 이벤트 발생 알림





AlertType	설명
CLOUD_WATCH	Cloud Watch 알림
EXCEPTION	Exception 알림

## 애플리케이션 경고 알림

애플리케이션 알림은 Event Title, Event Message 모두 제공합니다. [이벤트 상태가 해결되면 추가 알림](#) 기능을 사용하는 경우 이벤트 발생 조건이 해제되면 Evnet Off Message가 전송됩니다.

Event Type	Event Level	Event Title	Event Message	Event Off Message
애플리케이션 CPU	Warning	HIGH_CPU	CPU is high. $\${value}\%$ ( $\geq \${threshold}\%$ )	RECOVERED: CPU is high. less than $\${threshold}\%$
애플리케이션 CPU	Critical	CRITICAL_HIGH_CPU	CPU is too high. $\${value}\%$ ( $\geq \${threshold}\%$ )	RECOVERED: CPU is too high. less than $\${threshold}\%$
애플리케이션 메모리	Warning	HIGH_MEMORY	Memory is high. $\${value}\%$ ( $\geq \${threshold}\%$ )	RECOVERED: Memory is high. less than $\${threshold}\%$
애플리케이션 메모리	Critical	CRITICAL_HIGH_MEMORY	Memory is too high $\${value}\%$ ( $\geq \${threshold}\%$ )	RECOVERED: Memory is too high. less than $\${threshold}\%$
애플리케이션 디스크	Warning	HIGH_DISK	Disk is high $\${value}\%$ ( $\geq \${threshold}\%$ )	RECOVERED: Disk id high. less than $\${threshold}\%$
애플리케이션 디스크	Critical	CRITICAL_HIGH_DISK	Disk is too high $\${value}\%$ ( $\geq \${threshold}\%$ )	RECOVERED: Disk is too highf. less than $\${threshold}\%$
정상 트랜잭션	Warning	HIGH_ACTIVE_TRANSACTION	Active Transaction Count is over $\${value}$ ( $\geq \${threshold}$ )	RECOVERED: Active Transaction Count is less than $\${threshold}$
에러 트랜잭션	Warning	HIGH_ERROR_TRANSACTION	Error Transaction Count is over $\${threshold}$	RECOVERED: Error Transaction Count is less than

Event Type	Event Level	Event Title	Event Message	Event Off Message
			(\${value})	\${threshold}
느린 트랜잭션	Warning	TOO_MANY_SLOW_TX	Too many delayed transactions (\${value}, above \${time} ms)	RECOVERED: Too many delayed transactions. less than \${threshold}

# 인스턴스 성능 관리


홈 화면 > 프로젝트 선택 > [인스턴스 성능 관리](#)

애플리케이션의 환경을 확인하고 성능과 관련한 설정을 확인할 수 있습니다.

## 에이전트 목록


화면 왼쪽에 애플리케이션 목록에서는 프로젝트에 할당된 에이전트 목록을 확인할 수 있습니다. 개별 에이전트 항목을 선택하면 오른쪽 화면에 설정된 환경 변수 및 에이전트 설정, 성능과 관련한 정보를 조회할 수 있습니다.

No.	애플리케이션
1	demo-8100
2	demo-8101
3	demo-8102
4	demo-8103
5	demo-8104
6	demo-8105

에이전트 목록을 갱신하거나 비활성화된 에이전트를 재기동한 다음 목록에 자동으로 표시되지 않는다면  버튼을 선택하세요.

## 에이전트 및 애플리케이션 상세 정보

화면 오른쪽에서는 에이전트 및 애플리케이션과 관련한 상세 정보를 확인할 수 있습니다.

- 오른쪽 위에 텍스트 입력란을 통해 원하는 항목을 필터링할 수 있습니다.
- 모니터링 대상 서버에 위치한 에이전트 및 애플리케이션의 변경 사항이 자동 반영되지 않는다면  [새로고침](#) 버튼을 선택하세요.

## 에이전트 관련 항목

### 실행 환경 변수

홈 화면 > 프로젝트 선택 > [인스턴스 성능 관리](#) > [실행 환경 변수](#)

실행 환경 변수
🔍 필터링(필) 입력해주세요
🔄 새로고침

No.	애플리케이션	이름	상세
1	demo-8100	java.version	17.0.9
2	demo-8101	java.vendor	Private Build
3	demo-8102	java.home	/usr/lib/jvm/java-17-openjdk-arm64
4	demo-8103	os.name	Linux
5	demo-8104	os.arch	aarch64
6	demo-8105	file.encoding	utf-8
		user.timezone	Asia/Seoul
		user.language	en
		user.home	/home/whatap
		user.dir	/data/prod
		catalina.base	.
		-Xms	64m
		-Xmx	64m
		whatap.boot.start	premain
		java.name	990@virtual-java-agent
		java.start	20231219 05:27:37.137
		uptime	GMT-20000101 00:00:00.000
		whatap.home	/data/prod
		whatap.ip	10.21.1.26
		whatap.port	8100
		whatap.oid	-1128904592
		whatap.oidname	demo-8100
		whatap.name	demo-8100
		whatap.version	2.2.13 20230808

에이전트 실행과 관련한 환경 변수를 조회할 수 있습니다. 에이전트 버전 및 설치 경로, 이름, IP 주소 등을 확인할 수 있습니다.

### 에이전트 로그

홈 화면 > 프로젝트 선택 > [인스턴스 성능 관리](#) > [에이전트 로그](#)

에이전트 로그

No.	애플리케이션	이름	길이
1	demo-8100	whatap-20240318.log	292,017
2	demo-8101	whatap-20240319.log	333,990
3	demo-8102	whatap-20240320.log	366,563
4	demo-8103	whatap-20240321.log	369,999
5	demo-8104	whatap-20240322.log	333,170
6	demo-8105	whatap-20240323.log	297,550
		whatap-20240324.log	252,030
		whatap-20240325.log	33,521

284017 이동 END

```
erPack (wait=1) Q1Size=1 LastRecvTime=9404 LocalPort=52918 pid=974
20240318 23:07:45 [demo-8102] CounterPack (wait=1) Q1Size=3 LastRecvTime=9394
LocalPort=52918 pid=974
20240318 23:09:10 [demo-8100] CounterPack (wait=1) Q1Size=1 LastRecvTime=9431
LocalPort=52934 pid=990
20240318 23:10:15 [demo-8102] CounterPack (wait=1) Q1Size=3 LastRecvTime=9403
LocalPort=52918 pid=974
20240318 23:12:50 [demo-8104] CounterPack (wait=1) Q1Size=1 LastRecvTime=9403
LocalPort=52966 pid=945
20240318 23:18:00 [demo-8104] CounterPack (wait=1) Q1Size=3 LastRecvTime=9403
LocalPort=52966 pid=945
20240318 23:18:35 [demo-8102] CounterPack (wait=1) Q1Size=3 LastRecvTime=9403
LocalPort=52918 pid=974
20240318 23:19:15 [demo-8103] CounterPack (wait=1) Q1Size=1 LastRecvTime=9398
LocalPort=52972 pid=982
20240318 23:20:20 [demo-8103] CounterPack (wait=1) Q1Size=3 LastRecvTime=9399
LocalPort=52972 pid=982
20240318 23:20:40 [demo-8102] CounterPack (wait=1) Q1Size=3 LastRecvTime=9202
LocalPort=52918 pid=974
20240318 23:20:40 [demo-8105] CounterPack (wait=1) Q1Size=3 LastRecvTime=9400
LocalPort=52948 pid=999
```

모니터링 대상 서버에 저장된 에이전트 로그를 조회할 수 있습니다. 로그 파일의 이름은 `whatap-YYYYMMDD.log` 형식입니다. 각 로그를 선택해 로그에 캡처되는 오류 및 이벤트에 대한 정보를 액세스할 수 있습니다.

❗ 로그와 관련한 에이전트 설정은 [다음 문서](#)를 참조하세요.

## 에이전트 설정

홈 화면 > 프로젝트 선택 > [인스턴스 성능 관리](#) > [에이전트 설정](#) Old

모니터링 대상 서버에 위치한 `whatap.conf` 파일을 직접 수정하지 않고 에이전트 설정 옵션을 추가하거나 수정, 삭제할 수 있습니다.

❗ 에이전트 설정 기능은 사용성과 기능을 개선한 [관리 > 에이전트 설정](#) 메뉴를 이용할 것을 권장합니다. 화면 오른쪽 위에 [신규 에이전트 설정](#) 버튼을 선택하세요. 자세한 내용은 [다음 문서](#)를 참조하세요.

## 애플리케이션 관련 항목

### 환경변수

홈 화면 > 프로젝트 선택 > [인스턴스 성능 관리](#) > [환경변수](#)

환경변수

필터링(필) 입력해주세요

새로고침

No.	애플리케이션	이름	상세
1	demo-8100	catalina.base	.
2	demo-8101	catalina.home	.
3	demo-8102	catalina.useNaming	true
4	demo-8103	file.encoding	utf-8
5	demo-8104	file.separator	/
6	demo-8105	java.class.path	/data/prod/whatap-virtual-2.2.2.jar
		java.class.version	61.0
		java.home	/usr/lib/jvm/java-17-openjdk-arm64
		java.io.tmpdir	/tmp
		java.library.path	/usr/java/packages/lib:/usr/lib/aarch64-linux-gnu/jni:/lib/aarch64-linux-gnu:/usr/lib/aarch64-linux-gnu:/usr/lib/jni:/lib:/usr/lib
		java.runtime.name	OpenJDK Runtime Environment
		java.runtime.version	17.0.9+9-Ubuntu-120.04
		java.specification.name	Java Platform API Specification
		java.specification.vendor	Oracle Corporation
		java.specification.version	17
		java.vendor	Private Build
		java.vendor.url	Unknown
		java.vendor.url.bug	Unknown
		java.version	17.0.9

애플리케이션 실행과 관련한 환경 변수 정보를 조회할 수 있습니다.

## 힙 히스토그램

홈 화면 > 프로젝트 선택 > [인스턴스 성능 관리](#) > [힙 히스토그램](#)

힙 히스토그램

필터(을) 입력해주세요

No.	애플리케이션	번호	건수	크기	타입
1	java-demo1	1	130,622	3,134,928	java.lang.String
2	java-demo2	2	73,050	2,922,000	whatap.util.DateHelper\$Day
3	java-demo3	3	4,752	2,859,880	[]
4	java-demo4	4	327	1,267,696	[Lwhatap.lang.step.Step;
5	java-demo5	5	27,986	895,552	java.util.concurrent.ConcurrentHashMap\$Node
6	java-demo6	6	14,653	820,568	whatap.lang.step.MethodStepX
7		7	2,402	656,432	[Lwhatap.util.DateHelper\$Day;
8		8	1,052	642,296	[C
9		9	141	579,792	[Ljava.nio.ByteBuffer;
10		10	4,372	532,336	java.lang.Class
11		11	1,152	491,688	[Lwhatap.util.StringKeyLinkedMap\$StringKeyLinkedEntry;
12		12	5,339	404,792	[Ljava.lang.Object;
13		13	4,090	370,136	[Ljava.util.HashMap\$Node;
14		14	10,895	348,640	java.util.HashMap\$Node
15		15	2,165	324,752	[Ljava.util.concurrent.ConcurrentHashMap\$Node;
16		16	9,964	318,848	whatap.util.IntKeyLinkedMap\$IntKeyLinkedEntry
17		17	20	281,600	[Lwhatap.util.IntIntMap\$IntIntEntry;
18		18	10,817	259,608	whatap.lang.var.I2
19		19	5,335	256,080	java.util.HashMap
20		20	3,736	239,104	java.util.concurrent.ConcurrentHashMap

JVM(자바 가상 머신)의 메모리에 올라가 있는 Heap 점유 객체 현황(힙 메모리상의 객체별 사이즈)을 조회할 수 있습니다.

ⓘ Java 6 ~ 8 버전에서는 JVM 옵션 없이 기본 지원하지만, 일부 Java 버전에 따라 다음과 같이 JVM 옵션을 적용해야 합니다.

- Java 9 ~ Java 15 버전

```
-Djdk.attach.allowAttachSelf=true
```

example

```
java -javaagent:{WHATAP_HOME}/whatap.agent-X.Y.Z.jar -Djdk.attach.allowAttachSelf=true -jar {application.jar}
```

- Java 16 버전 이상

```
-Djdk.attach.allowAttachSelf=true
--add-opens=jdk.attach/sun.tools.attach=ALL-UNNAMED
```



example

```
java -javaagent:{WHATAP_HOME}/whatap.agent-X.Y.Z.jar -Djdk.attach.allowAttachSelf=true --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=jdk.attach/sun.tools.attach=ALL-UNNAMED -jar {application.jar}
```

- Java 5 버전 이하, IBM Java는 지원하지 않습니다.

## 로딩된 클래스

홈 화면 > 프로젝트 선택 > 인스턴스 성능 관리 > 로딩된 클래스

No.	애플리케이션	No	조회	재정의	클래스	슈퍼 클래스	인터페이스	리소스
1	demo-8100	1			org.slf4j.Marker			jar:file:/data/prod/lib4/slf4j-nop.jar/org/slf4j/M...
2	demo-8101	2			jdk.proxy2.\$Proxy2	java.lang.reflect.Proxy	oshi.jna.platform.linux.LinuxLibc	
3	demo-8102	3			org.slf4j.WhatapLogger	java.lang.Object	org.slf4j.Logger	jar:file:/data/prod/lib4/slf4j-nop.jar/org/slf4j/W...
4	demo-8103	4			org.slf4j.Logger			jar:file:/data/prod/lib4/slf4j-nop.jar/org/slf4j/Lo...
5	demo-8104	5			org.slf4j.ILoggerFactory			jar:file:/data/prod/lib4/slf4j-nop.jar/org/slf4j/ILo...
6	demo-8105	6			org.slf4j.LoggerFactory	java.lang.Object		jar:file:/data/prod/lib4/slf4j-nop.jar/org/slf4j/Lo...
		7			whatap.xtra.oshi.VmCpu	java.lang.Object		bytes:/whatap/xtra/oshi/VmCpu.class
		8			whatap.xtra.oshi.OshiMain	java.lang.Object	whatap.agent.proxy.IOshi	bytes:/whatap/xtra/oshi/OshiMain.class
		9			whatap.xtra.tools.JVM	java.lang.Object		bytes:/whatap/xtra/tools/JVM.class
		10			whatap.xtra.tools.ToolsMain	java.lang.Object	whatap.agent.proxy.IToolsMain	bytes:/whatap/xtra/tools/ToolsMain.class
		11			whatap.xtra.http.HttpClient43	java.lang.Object	whatap.agent.proxy.IHttpClient	bytes:/whatap/xtra/http/HttpClient43.class
		12			whatap.xtra.http.ResponseWrap	java.lang.Object	whatap.agent.api.trace.Response	bytes:/whatap/xtra/http/ResponseWrap.class
		13			whatap.xtra.http.RequestWrap	java.lang.Object	whatap.agent.api.trace.Request	bytes:/whatap/xtra/http/RequestWrap.class
		14			whatap.xtra.http.HttpTrace	java.lang.Object	whatap.agent.proxy.IHttpTrace	bytes:/whatap/xtra/http/HttpTrace.class
		15			whatap.xtra.sql.JdbcUri\$3	whatap.util.IntKeyLinkedMap		bytes:/whatap/xtra/sql/JdbcUri\$3.class
		16			whatap.xtra.sql.JdbcUri\$2	java.lang.Object	java.lang.Runnable	bytes:/whatap/xtra/sql/JdbcUri\$2.class
		17			whatap.xtra.sql.JdbcUri\$1	java.lang.Object	java.lang.Runnable	bytes:/whatap/xtra/sql/JdbcUri\$1.class
		18			whatap.xtra.sql.JdbcUri	java.lang.Object		bytes:/whatap/xtra/sql/JdbcUri.class
		19			whatap.xtra.sql.SqlTrace	java.lang.Object	whatap.agent.trace.sql.ISqlTrace	bytes:/whatap/xtra/sql/SqlTrace.class
		20			com.sun.tools.attach.AttachOperationFailedEx...	java.io.IOException		jdk.attach/com/sun/tools/attach/AttachOper...

Java 애플리케이션에 로딩된 클래스 정보를 확인할 수 있습니다.

## 라이브러리 버전

홈 화면 > 프로젝트 선택 > 인스턴스 성능 관리 > 라이브러리 버전

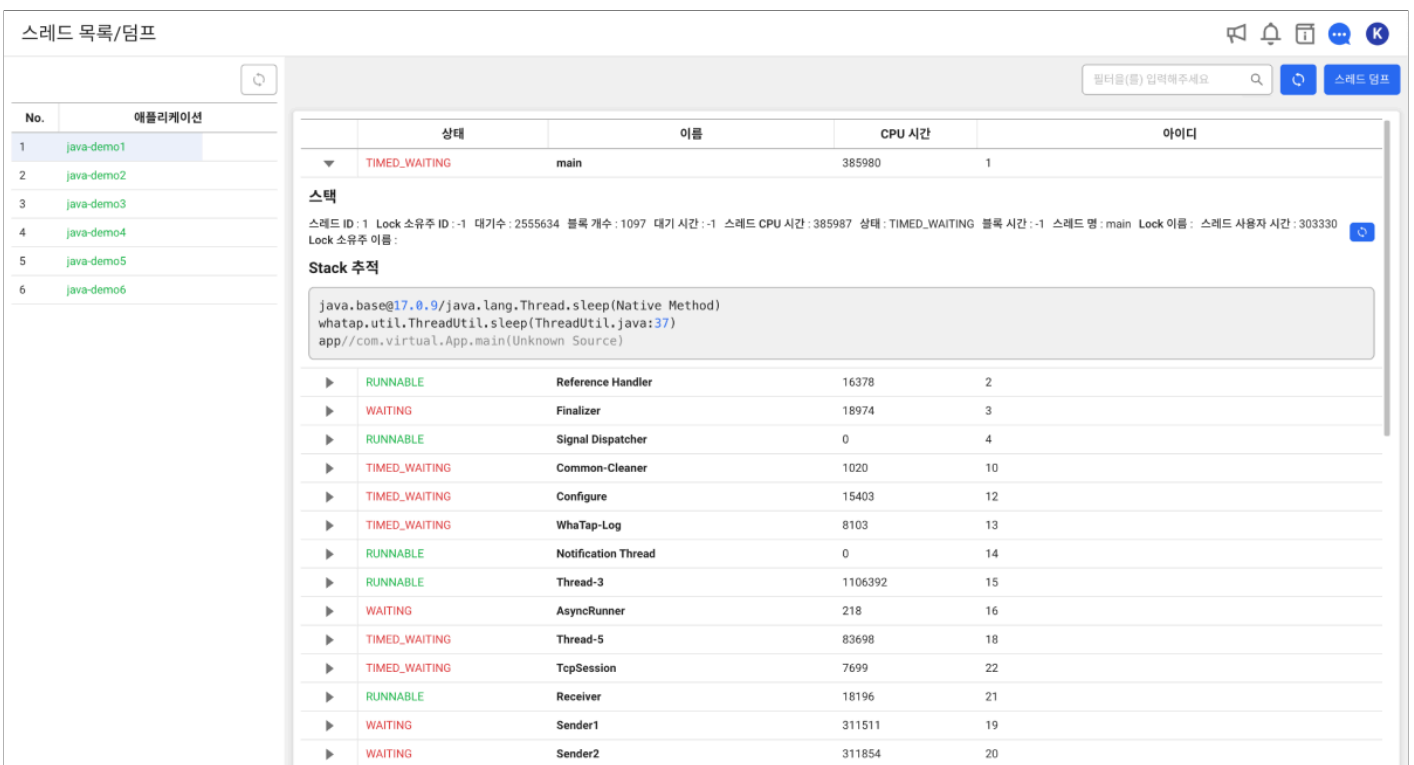




Java 애플리케이션을 구성하는 라이브러리의 버전을 확인할 수 있습니다.

## 스레드 목록/덤프

홈 화면 > 프로젝트 선택 > 인스턴스 성능 관리 > 스레드 목록/덤프



Java 프로세스의 모든 스레드 상태에 대한 스냅샷 정보를 확인할 수 있습니다. Java 애플리케이션의 성능 문제를 진단하는데 이용할 수 있습니다. 스레드 목록에서 각 스레드의 ▶ 버튼을 선택하면 스레드 덤프 정보를 확인할 수 있습니다. 자세한 내용은 [다음 문서](#)를 참조하세요.

## 소켓 오픈 개수

홈 화면 > 프로젝트 선택 > [인스턴스 성능 관리](#) > [소켓 오픈 개수](#)

소켓 오픈 개수

No.	애플리케이션	키	호스트	포트	건수	서비스	프로파일 조회 시간	스택 ↓
1	demo-8100	9151314447111823268	127.0.0.1	8100	67,736K	/order/remove/employee/kwangju	2024년 1월 9일 오후 3:13	<a href="#">스택 확보</a>
2	demo-8101	227344498856827336	3.39.176.158	6600	43		2024년 3월 20일 오후 6:41	<a href="#">스택 확보</a>
3	demo-8102							
4	demo-8103							
5	demo-8104							
6	demo-8105							

Java 애플리케이션이 TCP 기능을 수행하기 위해 오픈한 소켓(Socket) 정보를 확인할 수 있습니다.

## 메소드 성능 상태

홈 화면 > 프로젝트 선택 > [인스턴스 성능 관리](#) > [메소드 성능 상태](#)

Java 애플리케이션에서 수행 중인 메소드(Method)의 상세 정보를 확인할 수 있습니다.

## 데이터소스 상태

홈 화면 > 프로젝트 선택 > [인스턴스 성능 관리](#) > [데이터소스 상태](#)

데이터소스 상태

No.	애플리케이션	context	이름	키	값
1	demo-8100	BasicDataSourcePool	org.apache.commons.dbcp.BasicDataSource(8423...	getLoginTimeout	0
2	demo-8101	BasicDataSourcePool	org.apache.commons.dbcp.BasicDataSource(8423...	getUrl	jdbc:db:virtual
3	demo-8102	BasicDataSourcePool	org.apache.commons.dbcp.BasicDataSource(8423...	getNumActive	80
4	demo-8103	BasicDataSourcePool	org.apache.commons.dbcp.BasicDataSource(8423...	getNumIdle	20
5	demo-8104				
6	demo-8105				

데이터소스(DataSource)의 상태를 확인할 수 있습니다.

## 시스템 GC

홈 화면 > 프로젝트 선택 > [인스턴스 성능 관리](#) > [시스템 GC](#)

시스템 GC

No.	애플리케이션	전체 사용량		사용량	
		GC 실행 전	GC 실행 후	GC 실행 전	GC 실행 후
1	demo-8100	64 MB	64 MB	34.71 MB	20.05 MB
2	demo-8101				
3	demo-8102				
4	demo-8103				
5	demo-8104				
6	demo-8105				

JVM(자바 가상 머신)의 Heap 영역에서 동적으로 할당했던 메모리 중 필요 없게 된 메모리 객체(garbage)를 모아 제거할 수 있습니다. **실행** 버튼을 선택하면 GC 프로세스를 실행하게 되며, 실행 전 후의 메모리 용량을 확인할 수 있습니다.

## 힙 덤프

홈 화면 > 프로젝트 선택 > [인스턴스 성능 관리](#) > [힙 덤프](#)

힙 덤프

No.	애플리케이션	이름		크기
		이름	크기	
1	demo-8100	heapdump-demo-8100-20240308-170240.hprof		32,119K
2	demo-8101	heapdump-demo-8102-20231127-154340.hprof		33,691K
3	demo-8102	heapdump-demo-8100-20231025-113647.hprof		32,878K
4	demo-8103	heapdump-demo-8104-20240312-155643.hprof		32,402K
5	demo-8104	heapdump-demo-8100-20230829-151946.hprof		33,684K
6	demo-8105	heapdump-demo-8102-20231127-154013.hprof		30,886K
		heapdump-demo-8100-20240305-094447.hprof		33,361K
		heapdump-demo-8104-20231025-113610.hprof		33,091K
		heapdump-demo-8100-20230829-151907.hprof		33,928K
		heapdump-demo-8100-20240312-152304.hprof		33,699K
		heapdump-demo-8102-20231127-153938.hprof		33,645K
		heapdump-demo-8105-20230829-150539.hprof		32,783K
		heapdump-demo-8100-20231218-081457.hprof		32,238K
		heapdump-demo-8100-20240314-152313.hprof		33,001K
		heapdump-demo-8100-20231019-140249.hprof		32,441K
		heapdump-demo-8102-20231027-153503.hprof		31,384K
		heapdump-demo-8104-20240312-155539.hprof		32,031K

Java 애플리케이션 실행 중 메모리 누수 등의 문제가 발생하면 관련된 문제를 정리하여 덤프 파일을 생성할 수 있습니다. **힙 덤프 확보** 버튼을 선택하세요.

ⓘ 이 기능은 Java 에이전트 1.5.2 버전 이상에서 지원합니다. 에이전트 설정에서 `heapdump_enabled` 옵션을 `true` 로

❗ 설정하세요. **힙 덤프 확보** 버튼을 선택해 힙 덤프 파일을 만드는 기능을 활성화합니다. 매번 힙 덤프를 자동 생성하지 않으며 해당 옵션이 성능에 영향을 주진 않습니다. 그러나 덤프 확보 작업은 성능에 영향을 미칠 수 있습니다.

힙 덤프 확보 작업은 부하가 큰 작업에 속합니다. 부하와 별개로 힙 덤프가 모두 추출될 때까지 애플리케이션의 코드는 실행이 중지(stop the world)되기 때문에 사용자가 느끼는 성능 차이는 더 심할 수 있습니다.

## 쓰로틀링 설정

홈 화면 > 프로젝트 선택 > 인스턴스 성능 관리 > 쓰로틀링 설정

카테고리	특정 카테고리	키	설명	기본값	단위
쓰로틀링(throttling)	Java	throttle_enabled	최대 동시 처리 수를 제한하기 위한 쓰로틀 기능을 활성화합니다. 쓰로틀 정책의 우선 순위는 다음과 같습니다. 1.block - URL, IP 기준 차단 설정이 우선 적용됩니다. 2.passing - URL, IP 기준 허용 정책은 reject 설정에 우선합니다. 3.reject - block 및 passing 정책 대상을 제외하고 reject 정책이 적용됩니다.	false	bool
		throttle_limit	에이전트 단위로 최대 부하 제한(메트릭 트랜잭션 수)을 지정합니다.	10000	count
		throttle_rejected_message	reject 옵션에 따른 쓰로틀링 제한 시 사용자에게 표시될 메시지를 정의합니다.	{too many request}	string
		throttle_rejected_forward	쓰로틀링 제한 시 사용자가 표시할 URL을 지정합니다.		string
		reject_event_enabled	쓰로틀링 제한 시 이벤트 알림 발행 여부를 설정합니다.	false	bool
		reject_event_interval	쓰로틀링 제한으로 인한 이벤트 알림 발행 시, 중복 이벤트 억제 시간을 지정합니다.	300000	millisecond
		throttle_blocking_url	passing 및 reject 옵션과 무관하게 무조건적으로 처리를 거부할 URL을 지정합니다.		string
		throttle_blocking_ip	passing 및 reject 옵션과 무관하게 무조건적으로 처리를 거부할 remote address(IP)를 지정합니다.		string
		throttle_target_urls	white list로 쓰로틀 기능을 적용할 URL을 지정합니다. 복수 지정 시 굵기 구분자를 사용합니다.		string
		throttle_passing_url	쓰로틀링 적용 대상에서 제외할 URL을 지정합니다. 복수 지정 시 굵기 구분자를 사용합니다.		string
		throttle_passing_url_prefix	쓰로틀링 적용 대상에서 제외할 URL의 prefix를 지정합니다. 복수 지정 시 굵기 구분자를 사용합니다.		string
		throttle_blocked_message	요청이 차단된 사용자에게 표시할 메시지를 지정합니다.	{request blocked}	string
		throttle_blocked_forward	요청이 차단된 사용자에게 노출할 페이지 URL을 지정합니다.		string

Java 애플리케이션의 부하량 제어와 관련한 에이전트 설정을 확인하고 옵션을 설정할 수 있습니다. 부하량 제어와 관련한 에이전트 설정에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

## 스레드 덤프 분석하기

[스레드 목록/덤프](#) 탭에서 확인할 수 있는 정보에 대해 안내합니다.

### 스레드 상태

스레드의 상태는 `java.lang.Thread` 클래스 내부에 `State` 라는 이름의 Enumerated Types(열거형)으로 선언된 항목들입니다.

- **NEW**: 스레드가 생성되었지만 아직 실행되지 않은 상태입니다.

- **RUNNABLE** : 현재 CPU를 점유하고 작업을 수행 중인 상태입니다. 운영체제의 자원 분배로 **WAITING** 상태가 될 수도 있습니다.
- **WAITING** : `wait()`, `join()`, `park()` 메소드 등을 이용해 대기하고 있는 상태입니다.
- **TIMED\_WAITING** : `sleep()`, `wait()`, `join()`, `park()` 메소드 등을 이용해 대기하고 있는 상태, **WAITING** 상태와 다른 점은 주어진 시간 동안 대기하고 있는 상태입니다. 외부적인 변화 뿐만 아니라 시간에 의해서도 대기 상태가 해제될 수 있습니다.
- **BLOCKED** : 사용하려는 개체의 락(Lock)이 풀릴 때까지 대기하고 있는 상태입니다.
- **TERMINATED** : 실행을 마친 상태입니다.

## 스레드 덤프 정보

스레드 목록에서 각 스레드의 ▶ 버튼을 선택하면 스레드 덤프 정보를 확인할 수 있습니다.

▼ **TIMED\_WAITING**
**Thread-153**
18,778K
3,838 172

**스택**  
 스레드 ID : 172 Lock 소유주 ID : -1 대기수 : 1128749013 블록 개수 : 1128519768 대기 시간 : -1 스레드 CPU 시간 : 18780269 상태 : TIMED\_WAITING 블록 시간 : -1 스레드 명 : Thread-153 🔍  
 Lock 이름 : whatap.util.RequestQueue@5851c896 스레드 사용자 시간 : 12865620 Lock 소유주 이름 :

**Stack 추적**

```

        java.base@17.0.8/java.lang.Object.wait(Native Method)
        whatap.util.RequestQueue.get(RequestQueue.java:58)
        whatap.logsink.zip.ZipSendProxyThread.run(ZipSendProxyThread.java:54)
```

## 스택

항목	속성	설명
스레드 ID	<code>threadId</code>	스레드에 할당된 고유 ID
Lock 소유주 ID	<code>lockOwnerId</code>	스레드가 차단된 개체의 모니터(Monitor) 잠금을 유지하는 스레드 ID
대기수	<code>waitedCount</code>	스레드가 WAITING 또는 TIMED_WAITING 상태가 된 총 횟수
블록 개수	<code>blockedCount</code>	스레드가 BLOCKED 상태가 된 총 횟수
대기 시간	<code>waitedTime</code>	스레드가 WAITING 상태를 지속한 경과 시간(밀리초), 스레드 경합 모니터링이 비활성화된 경우 -1 반환

항목	속성	설명
스레드 CPU 시간	threadCpuTime	스레드의 CPU 시간 합계(나노초)를 밀리초로 계산하여 표시, CPU 시간 측정을 비활성화한 경우 -1 반환
상태	threadstate	스레드 상태
블록 시간	blockedTime	스레드가 BLOCKED 상태가 된 후 경과 시간(밀리초), 스레드 경합 모니터링이 비활성화된 경우 -1 반환
스레드 명	threadName	스레드 고유 이름
Lock 이름	lockName	스레드의 입력이 차단되거나 <code>Object.wait</code> 메소드를 통해 통지를 기다리는 모니터 잠금을 표시한 문자열
Lock 소유주 이름	lockOwnerName	스레드가 차단되는 객체의 모니터 잠금을 수용하는 스레드 이름
스레드 사용자 시간	threadUserTime	스레드가 사용자 모드에서 실행한 CPU 시간(나노초)을 밀리초로 계산하여 표시

① 스레드 정보에 대한 자세한 내용은 [다음 링크](#)를 참조하세요.

## Stack 추적

예외가 발생했을 때 스택(Stack) 추적을 표시합니다. 스택 프레임 목록으로 코드가 호출한 메소드 정보가 포함되어 있습니다.

# Log 모니터링

로그는 애플리케이션 및 시스템에서 발생하는 이벤트와 메시지 등을 기록한 파일입니다. 이상 징후를 파악해 시스템 악화를 막거나 발생한 장애의 원인을 이해하고자 한다면 로그를 확인하는 것이 중요합니다.

현대 IT 서비스 구축 환경은 MSA 또는 Kubernetes 환경으로 변화하고 있는 추세이며 이로 인해 관리 대상이 증가하고 있습니다. 일반적인 로그 모니터링은 서버에 접속해 `tail` 명령어나 편집기를 통해 확인합니다. 하지만 이러한 환경에서는 개별 서버에 일일이 접속하는 등의 단순한 방법을 사용하기 어렵습니다. 경우에 따라 수백 또는 수천 대의 서버에서 발생하는 로그를 어떻게 확인할 수 있을까요?

와탭 로그 모니터링 서비스를 통해 수많은 로그를 보다 쉽게 관리할 수 있습니다.

❗ 리눅스 `tail` 명령어는 시간에 따라 내용이 추가되는 로그 등을 확인하기 위한 용도로 많이 사용됩니다.

## 주요 특징점

### • 중앙 통합 관리

와탭은 대량의 로그를 중앙에서 통합 관리할 수 있습니다. 개별 서버에 접근하지 않고 중앙에서 로그의 내용을 확인할 수 있어 편리합니다.

### • 모든 로그 수집

로그를 선별해서 수집하는 경우 중요한 데이터가 누락될 수 있습니다. 와탭은 모든 로그를 수집합니다. 이렇게 수집되는 로그들은 [라이브 테일](#) 메뉴를 통해 실시간으로 확인할 수 있습니다.

### • 가시성 확보

와탭이 제공하는 다양한 차트에서 로그를 확인할 수 있습니다. 이를 통해 가시성을 확보해 에러 및 이슈 정보에 대한 접근성을 높이고 장애 상황을 조기에 감지할 수 있게 합니다.

### • 유연한 용량 관리

개별 서버에 로그 적재 시 로그로 인해 파일 시스템 용량이 과도하게 점유되는 문제가 발생할 수 있습니다. 하지만 와탭을 통해 로그를 중앙에 모은다면 개별 서버에서 발생하는 로그 파일을 유지할 필요 없이 중앙에 적재된 로그 데이터의 유지 기간만 관리하면 됩니다.

### • 다양한 분석 관점

장애 상황을 파악하고 예측하기 위해서는 다양한 관점으로 로그를 분석할 수 있어야 합니다. 와탭은 특정 태그가 포함된 로그의 건수 추이 또는 특정 태그가 포함된 로그만 필터링해 확인할 수 있습니다. 자주 사용하는 패턴이라면 차트로 저장해 언제든지 조회할 수

있도록 설정할 수 있습니다.

- **패턴 알림**

장애의 패턴을 파악했다면 이를 알림으로 설정해 문제를 예방하거나 최대한 빠르게 인지할 수 있습니다. 와탭은 개별 로그를 기준으로 특정 키워드가 포함되면 알림을 받는 [실시간 로그 알림](#)과 특정 태그가 포함된 로그의 건수 추이를 기준으로 알림을 받는 [복합 로그 알림](#)을 제공합니다.

## 로그 분석

### 라이브 테일

[라이브 테일](#) 메뉴를 통해 실시간으로 수집된 로그는 `tail` 명령어를 사용한 것과 마찬가지로 화면을 통해 흘러가는 로그를 조회할 수 있습니다. [라이브 테일](#)에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

### 로그 트렌드

[로그 트렌드](#) 메뉴를 통해 수집되는 전체 로그 또는 특정 태그가 포함된 로그의 건수 추이를 확인할 수 있습니다. 로그 발생 건수가 장애 발생 및 해소 시점과 밀접한 연관을 가진 경우, 로그 발생 건수 추이를 통해 장애의 원인 분석과 대응이 빨라질 수 있습니다. [로그 트렌드](#)에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

### 로그 검색

[로그 검색](#) 메뉴를 통해 수집되는 전체 로그 또는 특정 태그가 포함된 로그를 조회할 수 있습니다. 특정 시간대 또는 특정 서버에서 발생한 로그를 태그를 기준으로 조회하고 확인할 수 있습니다. 선택한 로그의 앞뒤에 발생한 로그를 확인하는 인접 로그 기능은 특정 Error 또는 Exception이 발생한 전후 상황 파악 시 활용할 수 있습니다. [로그 검색](#)에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.



# 로그 모니터링 적용하기

와탭 로그 모니터링 서비스 이용을 위한 기본 적용 방법을 안내합니다. 와탭 로그 모니터링은 추가적인 에이전트를 구성하거나 로그에 맞추어 parser를 적용할 필요가 없습니다. 간단한 설정으로 빠르게 시작할 수 있습니다.

## 로그 모니터링의 작동 원리



일반적인 로그 통합 서비스는 수집기, 처리기, 저장소 그리고 UI 모듈로 이루어져 있습니다. 단계별 설정과 구성 작업을 필요로 하기에 각각의 모듈을 구축하는 과정이 번거로우며 추가 비용이 발생합니다.

와탭 로그 모니터링은 적용이 간단합니다. 기존의 모니터링 에이전트가 수집기 역할을 하기에 에이전트 옵션을 켜는 것만으로 로그 모니터링을 시작할 수 있습니다.

[Java](#)
[PHP](#)
[Python](#)
[Go](#)
[Server](#)
[Kubernetes](#)

- 출력된 파일에서 로그를 읽지 않고 Java 애플리케이션의 로그 라이브러리로 전달되는 로그를 직접 수집합니다.
- 로그를 직접 수집하기 때문에 파일 I/O를 유발하지 않아 시스템에 미치는 성능 영향이 매우 낮습니다.
- 트랜잭션 트레이스와 로그의 연결 추적성을 확보하여 트레이스에서 로그를 확인할 수 있습니다.

### ❗ Java 로그 라이브러리

대표적인 Java 로그 라이브러리는 Apache Log4j, Logback 입니다.

### ❗ Java Agent 2.1.1 버전부터 사용할 수 있습니다.

- 기존의 에이전트에 로그 수집 기능을 추가했습니다. 모니터링 에이전트가 로그 파일에 추가로 출력된 로그를 읽어 수집하는 방식을 활용합니다.

### ❗ PHP Agent 2.3.2 버전부터 사용할 수 있습니다.

- 기존의 에이전트에 로그 수집 기능을 추가했습니다. 모니터링 에이전트가 로그 파일에 추가로 출력된 로그를 읽어 수집하는 방식을 활용합니다.
- 로그에 트랜잭션 ID를 출력하면, 트랜잭션 트레이스와 로그의 연결 추적성을 확보하여 트레이스에서 로그를 확인할 수 있습니다.

### ❗ Python Agent 1.2.2 버전부터 사용할 수 있습니다.

- 기존의 에이전트에 로그 수집 기능을 추가했습니다. 모니터링 에이전트가 로그 파일에 추가로 출력된 로그를 읽어 수집하는 방식을 활용합니다.
- 기존의 에이전트에 로그 수집 기능을 추가했습니다. 모니터링 에이전트가 로그 파일에 추가로 출력된 로그를 읽어 수집하는 방식을 활용합니다.

### ❗ Server Agent 2.1.2 버전부터 사용할 수 있습니다.

- 쿠버네티스 컨테이너에 로그를 수집할 수 있습니다.

- 쿠버네티스 컨테이너 내부 애플리케이션의 로그를 수집할 수 있습니다.

 Kubernetes Agent 1.1.35 버전부터 사용할 수 있습니다.

## 로그 모니터링 적용하기

사용하는 애플리케이션에 따른 적용 방법을 다음과 같이 제공합니다. 로그 모니터링을 적용하기 전에 [지원 버전](#)을 먼저 확인하세요.

1. 지원하는 에이전트 버전을 확인하고 **업데이트**하세요.
2. 로그 모니터링 **옵션**을 설정하세요.
3. 로그 모니터링을 **활성화**하세요.

### Java

Java 애플리케이션에서 로그를 수집하는 방법을 안내합니다.

### PHP

PHP 애플리케이션에서 로그를 수집하는 방법을 안내합니다.

### Python

Python 애플리케이션에서 로그를 수집하는 방법을 안내합니다.

### Go

Go 애플리케이션에서 로그를 수집하는 방법을 안내합니다.

## **Server**

Server 애플리케이션에서 로그를 수집하는 방법을 안내합니다.

## **Kubernetes**

쿠버네티스 컨테이너와 컨테이너 내부 애플리케이션의 로그를 수집하는 방법을 안내합니다.

# Java

자바 애플리케이션에서 로그를 수집하려면 다음 3단계를 모두 완료해야 합니다.

## 에이전트 업데이트

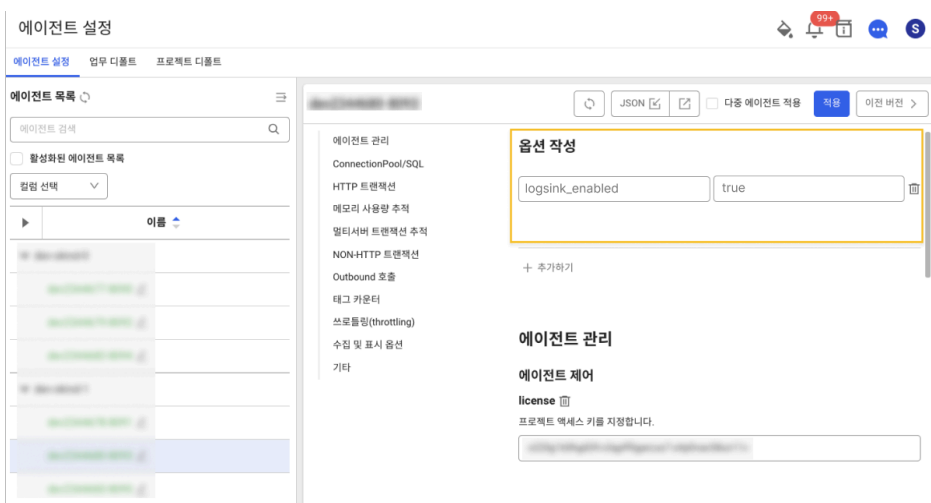
자바 에이전트 2.1.1 버전부터 가능합니다. 업데이트 방법은 [다음 문서](#)를 참조하세요.

## 에이전트 설정 확인

홈 화면 > 프로젝트 선택 > [로그](#) > [로그 설정](#)

와탭 모니터링 서비스 초기 화면에서 프로젝트를 선택한 다음 프로젝트 메뉴 하위에 [로그](#) > [로그 설정](#) 메뉴를 선택하세요. [로그 모니터링 시작하기](#) 섹션의 [에이전트 설정 확인](#) 탭의 안내를 참조해 진행하세요.

1. 프로젝트 메뉴 하위에 [관리](#) > [에이전트 설정](#) 메뉴를 선택하세요.
2. [옵션 작성](#) 탭에서 [직접 입력](#)을 선택하세요. 입력창에 다음과 같이 `logsink_enabled=true` 옵션을 추가하세요.



3. 로그 모니터링을 적용하기 위해 애플리케이션을 다시 시작하세요.

## 주요 옵션

- `hooklog_enabled` **Boolean**

기본값 `false`

Log 라이브러리를 hooking 하여 로그 모니터링을 활성화합니다.

! 애플리케이션 실행 전에 [whatap.conf](#)에 본 옵션이 활성화되어 있어야 이후 로그 모니터링의 On/Off를 `logsink_enabled` 설정을 통해 동적으로 제어할 수 있습니다. 애플리케이션 실행 전에 `logsink_enabled` 옵션이 `true`로 설정된 경우 본 옵션을 별도로 설정하지 않아도 로그 모니터링이 가능합니다.

! 앞으로 로그 모니터링을 활용할 가능성이 있다면 사전에 본 옵션을 꼭 설정할 것을 권장합니다.

### • hooklog\_custom\_methods

사용자 정의 로그를 등록합니다. 임의의 로그 프레임워크 내용을 전달합니다. 사이트에서 개별로 만든 로그 모듈의 로그를 추적할 때 사용하세요.

Java

```
package io.home.test;

public class MyLog {
    public void customLog(String log) { ... }
}
```

whatap.conf

```
hooklog_custom_methods=io.home.test.MyLog.customLog
```

### • logsink\_enabled Boolean

기본값 `false`

Log 모니터링 기능을 On/Off 합니다.

! 애플리케이션 실행 전에 [whatap.conf](#)에 `hooklog_enabled` 옵션이 설정되어 있으면 본 옵션을 통해 로그 모니터링의 On/Off를 동적으로 제어할 수 있습니다.

### • logsink\_trace\_enabled Boolean

기본값 `false`

Log에 트랜잭션 ID를 삽입하여, 트랜잭션 트레이스의 로그 탭을 노출할지 여부를 지정합니다.

## 로그 모니터링 활성화

홈 화면 > 프로젝트 선택 > [로그](#) > [로그 설정](#)

[로그 모니터링 시작하기](#) 섹션의 [로그 모니터링 활성화](#) 탭에서 토글 버튼으로 와탭 로그 모니터링을 활성화 또는 비활성화 할 수 있습니다.

**에이전트 설정 및 로그 모니터링 활성화**  ⓘ
요금제 보기

---

라이브 테일, 로그 트렌드 기능으로 애플리케이션의 흠어진 로그를 한 눈에 확인하실 수 있습니다.

---

- ▶ 1. 에이전트 설정 확인
- ▼ 2. 로그 모니터링 활성화

시작일 : 2022년 6월 13일 | 2022년 6월 28일부터 이용 요금이 청구됩니다.

- 토글 버튼을 켜면 로그 모니터링이 활성화됩니다. 활성화한 날부터 15일 동안 무료로 체험하실 수 있습니다.
- 토글 버튼을 끄면 로그 모니터링이 비활성화됩니다. 로그를 더 이상 저장하지 않습니다.

### ⓘ 권한

에이전트 설치 후 프로젝트에 대한 **수정 권한**이 있는 경우에만 로그 모니터링을 활성화할 수 있습니다. 권한에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

# PHP

PHP 애플리케이션에서 로그를 수집하려면 다음을 확인하세요.

## 에이전트 업데이트

PHP 에이전트 2.3.2 버전부터 가능합니다. 업데이트 방법은 [다음 문서](#)를 참조하세요.

## 에이전트 설정 확인

홈 화면 > 프로젝트 선택 > [로그](#) > [로그 설정](#)

와탭 모니터링 서비스 초기 화면에서 프로젝트를 선택한 다음 프로젝트 메뉴 하위에 [로그](#) > [로그 설정](#) 메뉴를 선택하세요. [로그 모니터링 시작하기](#) 섹션의 [에이전트 설정 확인](#) 탭의 안내를 참조해 진행하세요.

## 설정 파일 경로 확인

로그 모니터링을 원하는 파일의 [설정 파일 경로](#)를 확인하세요. CLI 환경에서 확인하거나 웹에서 확인할 수 있습니다.

### CLI 환경에서 확인하기

```
$ php -i | grep 'Scan'
Scan this dir for additional .ini files => /etc/php8.0.d
```

ⓘ 해당 경로가 "(None)"인 경우 설정 파일의 경로는 [/usr/whatap/php](#) 입니다.

### 웹에서 확인하기

CLI 환경과 Apache 환경의 PHP 설정이 다르면 웹에서 `phpinfo()` 함수의 결과 내용을 확인하세요.



PHP Version 8.0.12



System	Linux cent6default.com 2.6.32-754.35.1.el6.x86_64 #1 SMP Sat Nov 7 12:42:14 UTC 2020 x86_64
Build Date	Oct 26 2021 16:35:28
Build System	Linux cent6default.com 2.6.32-754.35.1.el6.x86_64 #1 SMP Sat Nov 7 12:42:14 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
Configure Command	'./configure' '--prefix=/usr/php8.0.12' '--bindir=/usr/bin' '--with-config-file-path=/etc/php8.0.12' '--program-prefix=' '--program-suffix=8.0.12' '--with-config-file-scan-dir=/etc/php8.0.d' '--with-libdir=lib64' '--with-apxs2' '--enable-fpm' '--with-curl' '--with-iconv' '--with-pdo-mysql' '--with-mysqli' '--with-pdo-oci' '--with-oci8' '--without-pgsql' '--without-pdo-pgsql' '--with-openssl' '--with-mhash' '--with-xsl' '--with-libxml' '--enable-sockets' '--enable-syssem' '--enable-sysshm' '--enable-soap' '--enable-gd' '--without-sqlite3' '--without-pdo-sqlite' 'PKG_CONFIG_PATH=/usr/local/lib/pkgconfig'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php8.0.12
Loaded Configuration File	(none)
Scan this dir for additional .ini files	<span style="border: 1px solid red; padding: 2px;">/etc/php8.0.d</span>

## 명령어 입력


설정 파일 경로를 포함한 명령어를 입력하면 로그 모니터링이 시작됩니다.

```
export LOGFILES=/some/path/file1,/some/other/file2
echo "whatap.logsink_enabled=true" | sudo tee -a [설정 파일 경로]/whatap.ini
echo "whatap.logsink.files=$LOGFILES" | sudo tee -a [설정 파일 경로]/whatap.ini
```

## 로그 모니터링 활성화

홈 화면 > 프로젝트 선택 > 로그 > 로그 설정

로그 모니터링 시작하기 섹션의 로그 모니터링 활성화 탭에서 토글 버튼으로 와탭 로그 모니터링을 활성화 또는 비활성화 할 수 있습니다.

에이전트 설정 및 로그 모니터링 활성화 

요금제 보기



라이브 테일, 로그 트렌드 기능으로 애플리케이션의 흠어진 로그를 한 눈에 확인하실 수 있습니다.


## ▶ 1. 에이전트 설정 확인

## ▼ 2. 로그 모니터링 활성화



시작일 : 2022년 6월 13일 | 2022년 6월 28일부터 이용 요금이 청구됩니다.

-  토글 버튼을 켜면 로그 모니터링이 활성화됩니다. 활성화한 날부터 15일 동안 무료로 체험하실 수 있습니다.
-  토글 버튼을 끄면 로그 모니터링이 비활성화됩니다. 로그를 더 이상 저장하지 않습니다.

 권한

에이전트 설치 후 프로젝트에 대한 **수정 권한**이 있는 경우에만 로그 모니터링을 활성화할 수 있습니다. 권한에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

# Python

Python 애플리케이션에서 로그를 수집하려면 다음을 확인하세요.

## 에이전트 업데이트

Python 에이전트 1.2.2 버전부터 가능합니다. 업데이트 방법은 [다음 문서](#)를 참조하세요.

## 에이전트 설정 확인

홈 화면 > 프로젝트 선택 > [로그](#) > [로그 설정](#)

와탭 모니터링 서비스 초기 화면에서 프로젝트를 선택한 다음 프로젝트 메뉴 하위에 [로그](#) > [로그 설정](#) 메뉴를 선택하세요. [로그 모니터링 시작하기](#) 섹션의 [에이전트 설정 확인](#) 탭의 안내를 참조해 진행하세요.

## 로그 수집 활성화

설정 파일 경로(WHATAP\_HOME)를 포함한 명령어를 입력하면 로그 수집이 바로 시작됩니다.

```
export LOGFILES={로그파일전체경로},...
echo "logsink_enabled=true" | sudo tee -a {설정파일경로}/whatap.conf
echo "logsink.files=$LOGFILES" | sudo tee -a {설정파일경로}/whatap.conf
```

## 로그와 웹 트랜잭션 연동

### 1.3.6 이후 버전

1.3.6 버전부터 다음의 방법으로 트랜잭션과 로그 연동을 설정할 수 있습니다. 현재 와탭은 Python의 logging, loguru 라이브러리를 지원하고 있습니다. 사용하는 Python Log 라이브러리에 따라 [whatap.conf](#)를 구성하세요.

- logging 모듈

```
logging
```

```
echo "trace_logging_enabled=true" | sudo tee -a {설정파일경로}/whatap.conf
```

- loguru 모듈

```
loguru
```

```
echo "trace_loguru_enabled=true" | sudo tee -a {설정파일경로}/whatap.conf
```

### 1.3.6 미만 버전

트랜잭션 별로 발생한 로그를 별도로 조회 가능하도록 트랜잭션 아이디 `{txid}` 를 로그에 출력합니다. 와탭 모니터링에서는 Python LogRecord에 `{txid}` 를 자동 주입하여 포매터 설정 시 로그 파일에 `{txid}` 를 출력할 수 있도록 합니다.

```
settings.py
...
LOGGING = {
...
    'formatters': {
        ...
    },
    'handlers': {
        ...
    },
    'loggers': {
        ...
        '{로거이름}': {
            'handlers': [...],
            ...
        },
    },
}

try:
import whatap.trace.mod.logging as whatap_logging
if whatap_logging.logging_injection_processed:
    LOGGING['formatters']['whatap.formatter']={
        '(): 'django.utils.log.ServerFormatter',
        'format': '[{server_time}] -- {{ "@txid" : "{txid}" }} -- {message}',
```

```

        'style': '{',
    }
    LOGGING['handlers']['whatap']={
        'level': 'DEBUG',
        'class': 'logging.handlers.RotatingFileHandler',
        'filename': os.path.join(BASE_DIR, 'logs','whatap_log.log'),
        'formatter': 'whatap.formatter',
    }
    LOGGING['loggers'][로그이름]['handlers'].append('whatap')
except:
    pass
...

```

## 로그 모니터링 활성화

홈 화면 > 프로젝트 선택 > 로그 > 로그 설정

로그 모니터링 시작하기 섹션의 로그 모니터링 활성화 탭에서 토글 버튼으로 와탭 로그 모니터링을 활성화 또는 비활성화 할 수 있습니다.

에이전트 설정 및 로그 모니터링 활성화 ⓘ
요금제 보기

---

라이브 테일, 로그 트렌드 기능으로 애플리케이션의 출어진 로그를 한 눈에 확인하실 수 있습니다.

---

- ▶ 1. 에이전트 설정 확인
- ▼ 2. 로그 모니터링 활성화
 

시작일 : 2022년 6월 13일 | 2022년 6월 28일부터 이용 요금이 청구됩니다.

- 토글 버튼을 켜면 로그 모니터링이 활성화됩니다. 활성화한 날부터 15일 동안 무료로 체험하실 수 있습니다.
- 토글 버튼을 끄면 로그 모니터링이 비활성화됩니다. 로그를 더 이상 저장하지 않습니다.

### ⓘ 권한

에이전트 설치 후 프로젝트에 대한 수정 권한이 있는 경우에만 로그 모니터링을 활성화할 수 있습니다. 권한에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

# Go

Go 애플리케이션에서 로그를 수집하려면 다음을 확인하세요.

## 에이전트 설정 확인

홈 화면 > 프로젝트 선택 > 로그 > 로그 설정

와탭 모니터링 서비스 초기 화면에서 프로젝트를 선택한 다음 프로젝트 메뉴 하위에 [로그](#) > [로그 설정](#) 메뉴를 선택하세요. [로그 모니터링 시작하기](#) 섹션의 [에이전트 설정 확인](#) 탭의 안내를 참조해 진행하세요.

## 로그 수집 활성화

다음 명령어를 입력하면 로그 수집이 바로 시작됩니다.

```
export LOGFILES=/some/path/file1,/some/other/file2
echo "logsink_enabled=true" | sudo tee -a /usr/whatap/agent/whatap.conf
echo "logsink.files=$LOGFILES" | sudo tee -a /usr/whatap/agent/whatap.conf
```

## 로그 모니터링 활성화

홈 화면 > 프로젝트 선택 > 로그 > 로그 설정

[로그 모니터링 시작하기](#) 섹션의 [로그 모니터링 활성화](#) 탭에서 토글 버튼으로 와탭 로그 모니터링을 활성화 또는 비활성화 할 수 있습니다.

에이전트 설정 및 로그 모니터링 활성화 
요금제 보기

---

라이브 테일, 로그 트랜드 기능으로 애플리케이션의 흠어진 로그를 한 눈에 확인하실 수 있습니다.

---

- ▶ 1. 에이전트 설정 확인
- ▼ 2. 로그 모니터링 활성화
 

시작일 : 2022년 6월 13일 | 2022년 6월 28일부터 이용 요금이 청구됩니다.

- 토글 버튼을 켜면 로그 모니터링이 활성화됩니다. 활성화한 날부터 15일 동안 무료로 체험하실 수 있습니다.
- 토글 버튼을 끄면 로그 모니터링이 비활성화됩니다. 로그를 더 이상 저장하지 않습니다.

#### ⓘ 권한

에이전트 설치 후 프로젝트에 대한 **수정 권한**이 있는 경우에만 로그 모니터링을 활성화할 수 있습니다. 권한에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

# Server

서버 애플리케이션에서 로그를 수집하려면 다음을 확인하세요.

## 에이전트 업데이트

서버 에이전트 2.1.2 버전부터 가능합니다. 업데이트 방법은 [다음 문서](#)를 참조하세요.

## 에이전트 설정 확인

홈 화면 > 프로젝트 선택 > [로그](#) > [로그 설정](#)

와탭 모니터링 서비스 초기 화면에서 프로젝트를 선택한 다음 프로젝트 메뉴 하위에 [로그](#) > [로그 설정](#) 메뉴를 선택하세요. [에이전트 설정 확인](#) 탭 상단의 OS 선택 탭에서 서버 OS([리눅스\(shell\)](#), [윈도우\(Powershell\)](#))를 선택하세요. [기본 설치](#) 또는 [카테고리와 함께 설치](#)를 참조해 진행하세요.

ⓘ 다음 설정은 에이전트 재시작이 필요합니다.

## Linux Shell

- [기본 설치](#)

Linux Shell

```
export LOGFILES=/some/path/file1,/some/other/file2
echo "logsink.files=$LOGFILES" | sudo tee -a /usr/whatap/infra/conf/whatap.conf
```

- [카테고리와 함께 설치](#)

Linux Shell

```
1 cd /usr/whatap/infra
2 sudo mkdir extension
3
```



```

4 cat >extension/logsink.conf<<EOL
5 [[inputs.logsink]]
6   category = "serverlog"
7   ## 로그 발생량 통계 별도 데이터로 전송 여부
8   stats_enabled = true
9   ## 로그 발생량 통계 카테고리
10  stats_category = "logsink_stats"
11  ## 로그 파일 경로(path)에 별표(*)가 포함되어 제외할 로그 비대상 파일명 설정
12  excludeNames = [ ".gz", ".zip" ]
13  [[inputs.logsink.file]]
14  ## 로그 파일 지정 시, 날짜 패턴(strftime.org) 지정 가능
15  path = "/some/path/%Y-%m-%d/.log"
16  disabled = false
17  encoding = "euc-kr"
18
19  [[inputs.logsink.file]]
20  path = "/some/other/log"
21  disabled = false
22  encoding = "utf-8"
23
24  [[inputs.logsink.file]]
25  ## 줄 단위 로그에서 해당 키워드 검색 시, 이전 로그에 병합
26  nowrap_keywords = ["Caused by:", "Test"]
27 EOL
28
29 sudo service whatap-infra restart

```

## Windows Powershell

- 기본 설치

Windows Powershell

```

$LOGFILES="c:\whatap\logs\%Y-%m-%d\*.log,c:\whatap\logs\*.log"
Add-Content "c:\Program Files\WhatapInfra\whatap.conf" -Value "logsink.files=$LOGFILES"

```

- 카테고리 와 함께 설치

Windows Powershell

```

1 # 관리자 권한 필요
2 New-Item -type "Directory" -Path "C:\Program Files\WhatapInfra\extension"
3
4 $contentToAdd = @"
5 [[inputs.logsink]]
6 category = "serverlog"
7 ## 로그 발생량 통계 별도 데이터로 전송 여부
8 stats_enabled = true
9 ## 로그 발생량 통계 카테고리
10 stats_category = "logsink_stats"
11 ## 로그 파일 경로(path)에 별표(*)가 포함되어 제외할 로그 비대상 파일명 설정
12 excludeNames = [ ".gz", ".zip" ]
13 [[inputs.logsink.file]]
14 ## 로그 파일 지정 시, 날짜 패턴(strftime.org) 지정 가능
15 path = "c:\whatap\logs%Y-%m-%d_.log"
16 disabled = false
17 encoding = "euc-kr"
18
19 [[inputs.logsink.file]]
20 ## 줄 단위 로그에서 해당 키워드 검색 시, 이전 로그에 병합
21 nowrap_keywords = ["Caused by:", "Test"]
22
23 "@
24
25 New-Item -path "C:\Program Files\WhatapInfra\extension" -name "logsink.conf" -type "file" -value $contentToAdd -Force
26
27 Restart-Service "Whatap Infra"

```

## 옵션 설정

- `stats_enabled` : 수집 현황 데이터의 수집 여부를 설정합니다. 기본값은 `false` 입니다. 값을 `true` 로 설정해야 합니다. `true` 로 설정하면 다음 `stats_category` 에서 설정한 카테고리(`logsink_stats`)로 통계 데이터가 발생합니다.
- `stats_category` : 수집 현황 데이터를 저장할 매트릭스 카테고리를 설정합니다. 값을 `logsink_stats` 로 설정해야 합니다. 통계 데이터 필드는 다음과 같습니다.
  - `file`
  - `checkInterval`
  - `encoding`
  - `filepos`

- `checkedLocalTime`
- `lastupdatedLocalTime`
- `fileSize`
- `error`
- `firstCheck`
- `transferBytes`
- `excludeNames`: 로그 파일 경로(path)에 별표(\*)를 포함한 경우 로그 비대상 파일을 제외하도록 파일명을 설정할 수 있습니다. 쉼표(,)를 구분자로 이용해 복수 설정할 수 있습니다.

#### Example

```
excludeNames = [ ".gz",".zip" ]
```

- `nowrap_keywords`: 줄 단위 로그 검색 시 해당 옵션값으로 지정한 키워드가 검색될 경우 이전 로그에 병합합니다.

## 윈도우 이벤트 로그 옵션 설정

윈도우 이벤트 로그 수집 시 다음과 같이 옵션을 설정할 수 있습니다.

```
# 관리자 권한 필요
New-Item -type "Directory" -Path "C:\Program Files\WhatapInfra\extension"

$contentToAdd = @"
[[inputs.win_eventlog]]
  category = "win_event_log"
  stats_category = "win_event_log_stats"
  stats_enabled = true
  enabled = true
[[inputs.win_eventlog.file]]
  #true | false
  enabled = true
  # Application, Security, Setup, System, Forwarded
  file = "Application"
  #1: Information, 2: Warning 3: Critical 4: Audit Success 5 Audit Fail
  #event_type =
  #event id
  #event_id =
  #event source name
```

```
#source_name = ""

"@
New-Item -path "C:\Program Files\WhatapInfra\extension" -name "win_eventlog.conf" -type "file" -value $contentToAdd -Force
Restart-Service "Whatap Infra"
```

- ⓘ • 지원 버전 2.5.2
- 운영체제: Windows

- 카테고리 지정( `category` ) 필수
  - | 예, `win_event_log`
- 통계 카테고리 지정( `stats_category` ) 필수
  - | 예, `win_event_log_stats`
- 통계 카테고리 On/Off( `stats_enabled` ) 필수
  - | 예, `true` 혹은 `false`
- 수집 기능 On/Off( `enabled` ) 필수
  - | 예, `true` 혹은 `false`
- 파일별 수집 기능 On/Off( `enabled` ) 필수
  - | 예, `true` 혹은 `false`
- 파일( `file` ) 필수
  - | 예, `Application` , `Security` , `Setup` , `System` , `Forwarded`
- 이벤트 타입( `event_type` ) 비필수
  - | 예, `1` , `2` , `3` , `4` , `5`

### ⓘ 이벤트 타입

#### 1. Information



2. Warning
3. Critical
4. Audit Success
5. Audit Fail

- 이벤트 아이디( `event_id` ) 비필수
- 이벤트 소스 이름( `source_name` ) 비필수

## 로그 모니터링 활성화

홈 화면 > 프로젝트 선택 > 로그 > 로그 설정

로그 모니터링 시작하기 섹션의 로그 모니터링 활성화 탭에서 토글 버튼으로 와탭 로그 모니터링을 활성화 또는 비활성화 할 수 있습니다.

에이전트 설정 및 로그 모니터링 활성화 ⓘ
요금제 보기

---

라이브 테일, 로그 트렌드 기능으로 애플리케이션의 출어진 로그를 한 눈에 확인하실 수 있습니다.

---

- ▶ 1. 에이전트 설정 확인
- ▼ 2. 로그 모니터링 활성화
 

시작일 : 2022년 6월 13일 | 2022년 6월 28일부터 이용 요금이 청구됩니다.

- 토글 버튼을 켜면 로그 모니터링이 활성화됩니다. 활성화한 날부터 15일 동안 무료로 체험하실 수 있습니다.
- 토글 버튼을 끄면 로그 모니터링이 비활성화됩니다. 로그를 더 이상 저장하지 않습니다.

### ⓘ 권한

에이전트 설치 후 프로젝트에 대한 수정 권한이 있는 경우에만 로그 모니터링을 활성화할 수 있습니다. 권한에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

# Kubernetes

쿠버네티스 컨테이너와 컨테이너 내부 애플리케이션의 로그를 수집하려면 다음을 확인하세요.

## 에이전트 업데이트

쿠버네티스 에이전트 1.1.35 버전부터 가능합니다. 업데이트 방법은 [다음 문서](#)를 참조하세요.

## 에이전트 설정 확인

홈 화면 > 프로젝트 선택 > [로그](#) > [로그 설정](#)

와탭 모니터링 서비스 초기 화면에서 프로젝트를 선택한 다음 프로젝트 메뉴 하위에 [로그](#) > [로그 설정](#) 메뉴를 선택하세요. [에이전트 설정 및 로그 모니터링 활성화](#) 섹션의 [에이전트 설정 확인](#) 탭의 안내를 참조해 진행하세요.

## 컨테이너 로그 수집 활성화

CP K8S-JINS-PORTAL-DIST ▾ 로그 설정

[로그모니터링 시작하기](#) [로그 1차 파서 설정](#) [로그 2차 파서 설정](#) [빠른 인덱스 설정](#) [로그 장기 보관 통계](#) [로그 1시간 통계 워젯 데이터 설정](#)

### 에이전트 설정 및 로그 모니터링 활성화 ⓘ

라이브 테일, 로그 트렌드 기능으로 애플리케이션의 흩어진 로그를 한 눈에 확인하실 수 있습니다.

▼ 1. 에이전트 설정 확인

컨테이너 로그를 수집하려면 다음을 확인해주세요.

1. 노드 에이전트 버전 확인 및 업그레이드 (v1.1.35 이후 버전부터 가능)
2. 아래의 버튼을 클릭하여 쿠버네티스 노드 에이전트에 로그 설정(logsink\_enabled=true)을 전체 적용합니다.

[로그 설정 적용하기](#)

쿠버네티스 컨테이너에 로그를 수집하려면 [에이전트 설정 확인](#) 탭에서 [로그 설정 적용하기](#) 버튼을 선택하세요.

## 컨테이너 내부 애플리케이션 로그 수집 활성화

Java 2.1.1, Python 1.2.2 버전부터 가능합니다. 쿠버네티스 컨테이너 상에서 실행되는 애플리케이션의 로그를 수집할 수 있도록 다음을 참조하세요.

The screenshot shows the '에이전트 설정' (Agent Settings) page. On the left, there's a sidebar with '에이전트 설정 및 로그 모니터링 활성화' (Activate Agent Settings and Log Monitoring) and '1. 에이전트 설정 확인' (1. Agent Settings Confirmation). The main content area has '에이전트 설정' (Agent Settings) and '2. 로그 모니터링 활성화' (2. Log Monitoring Activation). The right panel shows '에이전트 설정' (Agent Settings) with a '노드' (Node) dropdown, 'JSON', '복사' (Copy), and '적용' (Apply) buttons. Below is '옵션 작성' (Option Creation) with a dropdown for '옵션 선택' (Option Selection), a search bar, and '직접 입력' (Direct Input) section with two input fields for '키를 입력해주세요.' (Please enter the key) and '값을 입력해주세요.' (Please enter the value). Numbered callouts 1-4 point to the '에이전트 설정' button, the '옵션 선택' dropdown, the '직접 입력' section, and the '적용' button.

1. 에이전트 설정 확인 탭 하단의 1. 에이전트 설정 버튼을 선택하세요.
2. 애플리케이션 에이전트 설정 메뉴로 이동 후 2. 옵션 작성 창에서 직접 입력을 선택하세요.
3. 다음의 에이전트 설정 명령어의 키 `logsink_enabled` 와 값 `true` 를 3. 입력창에 입력하세요.

```
logsink_enabled=true
```

4. **적용** 버튼을 선택하세요. 쿠버네티스 컨테이너 내 애플리케이션의 로그를 수집할 수 있습니다.

- Java 애플리케이션 로그 수집에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.
- Python 애플리케이션 로그 수집에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.
- Go 애플리케이션 로그 수집에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

## 로그 모니터링 활성화

홈 화면 > 프로젝트 선택 > [로그](#) > [로그 설정](#)

[로그 모니터링 시작하기](#) 섹션의 [로그 모니터링 활성화](#) 탭에서 토글 버튼으로 와탭 로그 모니터링을 활성화 또는 비활성화 할 수 있습니다.

**에이전트 설정 및 로그 모니터링 활성화** 요금제 보기

---

라이브 테일, 로그 트렌드 기능으로 애플리케이션의 흩어진 로그를 한 눈에 확인하실 수 있습니다.

---

▶ 1. 에이전트 설정 확인

▼ 2. 로그 모니터링 활성화

시작일 : 2022년 6월 13일 | 2022년 6월 28일부터 이용 요금이 청구됩니다.

- 토글 버튼을 켜면 로그 모니터링이 활성화됩니다. 활성화한 날부터 15일 동안 무료로 체험하실 수 있습니다.
- 토글 버튼을 끄면 로그 모니터링이 비활성화됩니다. 로그를 더 이상 저장하지 않습니다.

### ⓘ 권한

에이전트 설치 후 프로젝트에 대한 **수정 권한**이 있는 경우에만 로그 모니터링을 활성화할 수 있습니다. 권한에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.



## 쿠버네티스 로그 모니터링 카테고리 안내

쿠버네티스 관련 다양한 로그를 확인할 수 있습니다. 와탭 쿠버네티스에서 제공하는 모니터링 카테고리는 다음과 같습니다. 설정에 따라 중복된 로그 내용이 저장될 수 있으니 반드시 중복 여부를 확인하세요.

카테고리	설명
#K8sEvent	<ul style="list-style-type: none"> <li>쿠버네티스에서 발생하는 이벤트가 저장된 로그</li> <li>사용자 설정과 무관하게 기본 생성</li> </ul>
#WhatapEvent	<ul style="list-style-type: none"> <li>와탭 이벤트 설정에 의해 발생한 이벤트가 저장된 로그</li> <li>사용자 설정과 무관하게 기본 생성</li> </ul>
containerStdout	<ul style="list-style-type: none"> <li>컨테이너 Standard Out 로그</li> <li>사용자 설정 시 생성</li> <li>노드 에이전트에 <code>logsink_enabled=true</code> 설정 추가 시</li> </ul>
AppLog	<ul style="list-style-type: none"> <li>컨테이너 내 애플리케이션 로그</li> <li>사용자 설정 시 생성</li> <li>애플리케이션 에이전트에 <code>logsink_enabled=true</code> 설정 추가 시</li> </ul>

# 로그 설정

홈 화면 > 프로젝트 선택 > 로그 > 로그 설정

로그 설정 메뉴에서 로그 모니터링 관련 설정을 할 수 있습니다. 상단의 탭을 통해 에이전트 설정 확인, 로그 모니터링의 활성화 여부 결정, 로그 데이터의 유지 기간 및 조회 비밀번호 설정, 로그 파서 등록, 빠른 인덱스 설정 등의 메뉴를 사용할 수 있습니다.

- ① • 로그 모니터링 활성화 기능을 사용하려면 프로젝트 수정 권한이 필요합니다.
- 로그 편집 권한을 통해 로그 모니터링 활성화 기능 외 로그 설정 메뉴를 수정할 수 있습니다.

## 로그 모니터링 시작하기

로그 설정
🔍 🔔 📄 ? 📄

로그모니터링 시작하기
로그 1차 파서 설정
로그 2차 파서 설정
빠른 인덱스 설정
로그 장기 보관 통계

**1** 에이전트 설정 및 로그 모니터링 활성화 📄
요금제 보기

라이브 테일, 로그 트렌드 기능으로 애플리케이션의 흩어진 로그를 한 눈에 확인하실 수 있습니다.

- ▶ 1. 에이전트 설정 확인
- ▶ 2. 로그 모니터링 활성화

**2** 로그 모니터링 데이터 설정 📄
초기화 저장

로그 사용량 230,777,963라인

로그 조회 비밀번호  로그 조회 비밀번호를 사용하려면 켜주세요.

데이터 유지 기간 30일 ▼

카테고리별 데이터 유지 기간

카테고리	데이터 보관일	금일 로그수 ①	어제 로그수 ①	일주일 로그수 ①	한달 로그수 ①	예상 로그수 ①
AppLog	5일 ▼	0	51,595,242	226,722,148	226,722,148	0
AppStdOut	10일 ▼	0	456,702	2,523,814	4,029,866	0
AppStdErr	6일 ▼	0	4,787	25,953	25,953	0

상단에서 로그 모니터링 시작하기 탭을 선택하세요. 📄 가이드 보기 아이콘과 요금제 보기 버튼을 선택하면 관련 안내 화면으로 이동합니다.

## 에이전트 설정 및 로그 모니터링 활성화

1 영역에서 에이전트 설정을 확인하고 [로그 모니터링 활성화](#) 토글 버튼으로 로그 모니터링 활성화 및 비활성화 여부를 설정하세요.

### 에이전트 설정 확인

로그 모니터링을 시작하기 위해 에이전트 버전과 설정 정보를 확인하세요. [에이전트 설정 확인](#) 메뉴를 선택하여 안내대로 과정을 진행하세요.

#### • 애플리케이션 모니터링

[적용하기](#) 메뉴 하위에 애플리케이션별 적용 안내를 확인하세요. 다음 설명서 [Java](#), [PHP](#), [Python](#), [Go](#) 등을 참조하세요.

#### • 서버 모니터링

[적용하기](#) 메뉴 하위에 개별 적용 안내를 확인하세요. [다음 문서](#)를 참조하세요. 안내대로 [whatap.conf](#) 에 로그 감시 대상 파일 설정을 추가하세요.

#### • 쿠버네티스 모니터링

[적용하기](#) 메뉴 하위에 개별 적용 안내를 확인하세요. [다음 문서](#)를 참조하세요.

### 로그 모니터링 활성화

[로그 모니터링 활성화](#) 메뉴를 선택하여 로그 모니터링 활성화 및 비활성화 여부를 설정하세요.

에이전트 설정 및 로그 모니터링 활성화 
요금제 보기

---

라이브 테일, 로그 트렌드 기능으로 애플리케이션의 흩어진 로그를 한 눈에 확인하실 수 있습니다.

---

- ▶ 1. 에이전트 설정 확인
- ▼ 2. 로그 모니터링 활성화
 

시작일 : 2022년 6월 13일 | 2022년 6월 28일부터 이용 요금이 청구됩니다.

- 토글 버튼을 켜면 로그 모니터링이 활성화됩니다. 활성화한 날부터 15일 동안 무료로 체험하실 수 있습니다.
- 토글 버튼을 끄면 로그 모니터링이 비활성화됩니다. 로그를 더 이상 저장하지 않습니다.

## 로그 모니터링 데이터 설정

2 영역에서 **로그 사용량**을 확인할 수 있습니다. 또한 **데이터 유지 기간** 및 **로그 조회 비밀번호** 설정을 변경할 수 있습니다.

### 데이터 유지 기간

공통으로 적용할 기본(default) 데이터 유지 기간입니다. 미지정 시 기본값은 1일입니다. 카테고리별 데이터 유지 기간을 별도로 설정하지 않으면 이 데이터 유지 기간이 기본적으로 적용됩니다. 카테고리별 데이터 유지 기간을 설정하고 **초기화** 버튼을 선택하면 기본 데이터 유지 기간으로 초기화됩니다.

### 카테고리별 데이터 유지 기간

카테고리별 로그 데이터 유지 기간을 지정할 수 있습니다. **로그 수**는 해당 기간 동안 쌓인 로그 라인을 의미합니다. 예를 들어 **금일 로그 수**는 하루 동안 쌓인 로그 라인 개수, **예상 로그 수**는 데이터 보관일에 금일 로그 수를 곱한 로그 라인 개수를 의미합니다.

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

- **트라이얼 프로젝트**

데이터 유지 기간으로 1일, 2일, 3일을 선택할 수 있습니다.

- **유료 프로젝트**

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

- **저장량 기준 과금**

데이터 유지 기간에 따라 비용이 달라집니다.

예시, 일 평균 200만 로그 라인이 쌓이고 데이터 유지 기간을 3일로 지정한 경우라면 평균 600만 로그 라인이 수집 서버에 유지되고 과금 대상이 됩니다.

### 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 경우 **로그 설정** 메뉴에서 새 비밀번호로 수정할 수 있습니다.

## 로그 1차 파서 설정

로그 설정 메뉴 상단에서 [로그 1차 파서 설정](#) 탭을 선택해 로그 파서를 등록 및 수정할 수 있습니다. 로그 1차 파서는 [GROK](#)과 [JSON](#) 파서를 제공합니다. 수집된 로그를 대상으로 패턴의 조건과 일치하는 키 정보 즉 검색 키와 검색 값을 추출합니다. 파싱된 로그 키는 로그의 유형을 분류하기 위한 용도 및 특정 로그를 검색하기 위한 인덱싱 용도로 활용합니다. 유형별 로그 발생 수를 집계하거나 특정 로그를 빠르게 찾아내기 위해 등록하는 필수 파서입니다.

- **GROK**: 기본은 정규 표현식 기반 파싱에 해당합니다. 예약 키워드 기반의 파싱을 제공합니다.
- **JSON**: 로그 중 JSON으로 출력된 부분에 대해서 일괄 파싱을 제공합니다.

### ❗ 파싱 로직 미등록 시 검색 가능한 key

category , oid , oname , okind , okineName , @txid , @login , httphost

### ❗ 파서 등록이 불가능한 예약어

다음 예약어의 경우 파서를 등록하더라도 인덱스가 생성되지 않습니다.

timestamp , message , pcode , category , content , logContent

❗ 로그 파서에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

## 설정 항목

설정 값	설명	기타
카테고리	패턴을 적용할 카테고리입니다.	required
로그 검출 조건	필터로 적용할 검색 키, 검색 값을 입력합니다. 로그 검출 조건에 맞는 로그 데이터에만 패턴을 적용합니다. 로그 검출 조건을 입력하지 않으면 모든 로그를 대상으로 패턴을 적용합니다.	optional
패턴	로그를 파싱(parsing)할 패턴입니다. 작성한 패턴에 맞추어서 파싱을 하고 인덱스를 생성합니다. GROK, 정규 표현식 문법을 지원합니다.	required

## 파서 목록

로그 설정

로그모니터링 시작하기 **로그 1차 파서 설정** 로그 2차 파서 설정 빠른 인덱스 설정 로그 장치 보관 통계

로그 1차 파서 설정 수집된 로그에 대한 파서를 등록할 수 있습니다. 적용 순서대로 파서가 적용되며, 최초로 일치하는 파서만 적용됩니다. + 추가하기 저장

적용 순서	파서	카테고리	필터	패턴	활성화
0	GROK	AppLog	모든 로그	\[%{TIMESTAMP_ISO8601:timestamp}\][\%{LOGLEVEL:loglevel}\s*\][\%{GREEDYDATA:classMethod}\][\%{GREEDYDATA:message}	<input checked="" type="checkbox"/>
1	JSON	AppLog	category:Applog	Prefix 없음 Postfix 없음 Ignore 없음	<input type="checkbox"/>

로그 설정 메뉴 상단에서 로그 1차 파서 설정 탭을 선택하면 등록된 파서를 조회하고 추가 및 편집이 가능한 파서 목록 화면을 확인할 수 있습니다.

- 상단 오른쪽 + 추가하기 버튼을 선택하면 파서 추가 창이 나타납니다.
- 파서 목록 적용 순서 컬럼의 ≡ 아이콘을 드래그해 파서 설정 순서를 변경할 수 있습니다.
- 파서 목록  활성화 토글을 통해 파서 활성화 여부를 지정할 수 있습니다.
- 파서 목록 수정 및 삭제 아이콘을 통해 등록된 파서를 수정 및 삭제할 수 있습니다.

## 파서 등록 순서

로그 설정 메뉴 상단에서 로그 1차 파서 설정 탭을 선택해 로그 파서를 등록 및 수정할 수 있습니다. 다음은 파서 등록 시 공통 순서를 안내합니다.

1. + 추가하기 버튼을 선택하면 파서 추가 창이 나타납니다.
2. 파서 선택 창에서 파서를 선택하세요. 각 파서 및 패턴 등록에 관한 자세한 내용은 다음 문서를 참조하세요.
  - [GROK 파서 및 패턴 등록](#)
  - [JSON 파서 및 패턴 등록](#)
3. 카테고리 선택 창에서 카테고리를 선택하거나 직접 입력하세요.
4. 로그 검출 조건으로 활용할 검색 키와 검색 값을 선택하거나 직접 입력하세요.  
필터 조건에 맞는 로그 데이터에만 패턴을 적용합니다. 로그 검출 조건을 지정하지 않으면 모든 로그를 대상으로 패턴을 적용합니다.
5. 패턴을 입력하세요.
6. 등록하려는 패턴이 정상적인지 시뮬레이션 버튼을 선택해 시뮬레이션 및 패턴의 퍼포먼스를 측정하세요.  
시뮬레이션과 퍼포먼스 측정에 관한 자세한 내용은 [다음 문서](#)를 참조하세요.
7. 시뮬레이션 결과가 정상적이라면 추가 버튼을 선택해 파서를 등록하세요.

❗ 로그 파서 등록 시 동일한 **카테고리**에 파서를 중복 등록할 수 없습니다.

## GROK 파서 패턴 등록

파서*	GROK <span>▼</span>	
카테고리*	카테고리를(를) 선택해주세요	
로그 검출 조건	검색 키	검색 값
	로그 검출 조건을 입력하지 않거나, 검색키/값중 한개만 입력하는 경우 모든 로그를 대상으로 패턴을 적용합니다.	
패턴*	<p>e.g. %{SYNTAX:SEMANTIC}</p>	

기본 문법은 `%{SYNTAX:SEMANTIC}` 입니다. GROK 파서에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

- **SYNTAX**

GROK 정의 패턴입니다.

- **SEMANTIC**

파싱된 데이터에 할당할 키입니다.

❗ SEMANTIC에는 예약어 등이 사용되지 않도록 조합어 사용을 권장합니다.



## JSON 포맷 파서 패턴 등록

파서*	JSON <span>▼</span>	
카테고리*	AppLog <span>▼</span>	<input type="checkbox"/> 직접 입력
로그 검색 조건	oname <span>▼</span>	demo-8100 <span>▼</span> <input type="checkbox"/> 직접 입력
	로그 검색 조건을 입력하지 않으면, 모든 로그를 대상으로 패턴을 적용합니다.	
패턴	Prefix	--
	Postfix	--
	Ignore	ignore

로그 전체 혹은 일부가 JSON 형태로 출력되는 경우 JSON 포맷 파서를 통해 JSON으로 출력된 부분을 파싱할 수 있습니다. 로그 중 JSON 형태로 출력된 부분을 검색하기 위하여 [Prefix](#), [Postfix](#) 옵션을 조합해 로그의 어느 부분을 JSON으로 인식해 파싱할지 지정합니다. JSON 파서에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

옵션	설명
Prefix	JSON 문자열의 시작 부분 앞의 문자열을 지정합니다. 미지정 시 로그 출력문의 맨 앞부터 JSON 문자열로 식별합니다.
Postfix	JSON 문자열의 종료 부분 뒤의 문자열을 지정합니다. 미지정 시 로그 출력문의 맨 뒤 까지를 JSON 문자열로 식별합니다.
Ignore	JSON 출력부 중 키 추출을 제외할 필드를 지정합니다.

### • 등록 예시

## Log

```
[2022-10-25 10:15:34:145]...(개행)
Request : {"key1":"value1","key2":"value2",...}(개행)
Response : {"key3":"value3","key4":"value4",...}
```

예시처럼 유입되는 로그가 Request JSON, Response JSON을 모두 파싱하고자 하는 경우 다음의 2가지 패턴을 등록합니다.

- Request 파싱용 패턴

| "Request : " 와 "Response" 사이의 문자열 `{"key1":"value1","key2":"value2",...}` 대상

- Response 파싱용 패턴

| "Response : " 부터 로그의 마지막 까지의 문자열 `{"key3":"value3","key4":"value4",...}` 대상

- JSON 커스텀 패턴 등록

로그 중 일부가 JSON 형태로 출력되는 경우 JSON으로 출력된 부분을 전용 커스텀 파서를 통해 파싱할 수 있습니다. 패턴을 다음과 같이 입력하세요.

```
io.whatap.logsink.parser.JsonFormatParser{}
```

로그 중 JSON 형태로 출력된 부분을 검출하기 위해 [Prefix](#), [Postfix](#) 옵션을 조합해 로그의 어느 부분을 JSON으로 인식하여 파싱할지 지정하세요.

`JsonFormatParser{}` 의 `{}` 에 옵션을 지정합니다.

- 등록 예시

## Log

```
[2022-10-25 10:15:34:145]...(개행)
Request : {"key1":"value1","key2":"value2",...}(개행)
Response : {"key3":"value3","key4":"value4",...}
```

예시처럼 유입되는 로그가 Request JSON, Response JSON을 모두 파싱하고자 하는 경우 다음의 2가지 패턴을 등록합니다.

- Request 파싱용 패턴

| "Request : " 와 "Response" 사이의 문자열 `{"key1":"value1","key2":"value2",...}` 대상

```
io.whatap.logsink.parser.JsonFormatParser {prefix:"Request : ",postfix:"Response"}
```

- Reponse 파싱용 패턴

"Response : " 부터 로그의 마지막 까지의 문자열 {"key3":"value3","key4":"value4",...} 대상

```
io.whatap.logsink.parser.JsonFormatParser {prefix: "Response : "}
```

## 파서 시뮬레이션

1. 파서 추가 화면에서 패턴을 입력한 후 **시뮬레이션** 버튼을 선택하세요. **파서 시뮬레이션** 창이 나타납니다.
2. **파서 시뮬레이션** 창에서 **로그**를 입력하세요.
3. 입력한 **로그**와 **패턴**을 확인하세요.

✕
파서 시뮬레이션

! 입력한 패턴으로 로그가 성공적으로 파싱되는지 시뮬레이션합니다.

\* 로그

2022-08-28T15:00:00Z This is a sample log.

\* 패턴

2022-08-28T15:00:00Z This is a sample log.

시뮬레이션

4. **시뮬레이션** 버튼을 선택해 등록하려는 패턴으로 파싱에 성공하는지 확인하세요.

- 시뮬레이션 성공 화면

패턴 적용
시뮬레이션

**시뮬레이션 결과**

키	timestamp
값	2023-08-28T15:30:45Z
결과	Ok

---

키	loglevel
값	INFO
결과	Ok

---

키	classMethod
값	MyApp::SomeClass::someMethod
결과	Ok

---

키	message
값	This is an example log message.
결과	Ok

퍼포먼스 측정 ▼

- 시뮬레이션 실패 화면

**시뮬레이션 결과**

**결과** Fail

**상세 원인** 파싱에 실패하였습니다.

5. **패턴 적용** 버튼 클릭 시 선택한 파서에 입력한 패턴이 적용됩니다.

## 퍼포먼스 측정

시뮬레이션 성공 후 [퍼포먼스 측정](#) 버튼을 선택해 파서에 대한 퍼포먼스를 측정할 수 있습니다. 시뮬레이션 수행 대상 문자열에 대하여 파서의 반복 파싱 소요 시간을 측정 후 다음과 같이 측정 결과를 확인하세요.

시뮬레이션 횟수	결과	최소 시간(ns)	최대 시간(ns)	평균 시간(ns)
1	SUCCESS	29,522	29,522	29,522
10	SUCCESS	5,752	29,628	11,145
100	SUCCESS	4,931	97,550	7,305
1,000	SUCCESS	2,215	178,706	4,431
10,000	SUCCESS	1,575	195,436	2,037
100,000	SUCCESS	451	604,623	1,286
1,000,000	SUCCESS	328	59,273,046	625

## 파싱 성공

파싱 로직을 등록해 키(key)가 생성되면 로그 조회 시 해당 키로 파싱된 값이 추가됩니다. 다음 [라이브 테일](#) 메뉴 예시와 같이 파싱된 키와 값이 추가됩니다.

Timestamp	로그
2022-08-17 14:28:00.612	<a href="#">oname</a> dev949400-8093 <a href="#">onodeName</a> node-1 <a href="#">oid</a> 413390913 <a href="#">category</a> AppStdOut <a href="#">okindName</a> dev-okind-1 load 52

파싱된 키는 [라이브 테일](#), [로그 검색](#), [로그 트렌드](#)에서 확인할 수 있습니다.

## 로그 2차 파서 설정

[로그 설정](#) 메뉴 상단에서 [로그 2차 파서 설정](#) 탭을 선택해 로그 파서를 등록 및 수정할 수 있습니다. [4xx, 5xx 상태 코드 파서](#)와 [상태 코드 성공률 파서](#)를 제공합니다. 로그 2차 파서는 GROK 또는 JSON과 같은 1차 파서가 파싱된 경우 사용할 수 있는 파서입니다. 1차 파서로 추출한 값을 가공해 통계 데이터를 생성합니다. 웹 혹은 API 응답 로그에 대해 Http Status Code를 기반으로 2차 통계를 추출합니다.

- 4xx, 5xx 상태 코드 파서: 비정상 응답에 대한 건수 정보를 집계합니다.
- 상태 코드 성공률 파서: 전체 건수 대비 비정상 응답 비율을 추출합니다.

❗ 로그 2차 파서는 1차 파싱된 결과에 대하여 특수 목적의 2차 파싱 기능을 제공합니다. 2차 파서를 사용하기 위해서는 **1차 파서가 등록되어** 있어야 합니다.

## 파서 목록

로그 2차 파서 설정 Grok 또는 JSON 파서가 이미 파싱된 경우 사용할 수 있는 파서입니다. + 추가하기 저장

적용 순서	파서	카테고리	필터	패턴	활성화
0	4xx, 5xx 상태 코드 파서	AppLog	없음		<input checked="" type="checkbox"/>
1	상태 코드 성공률 파서	AppLog	없음		<input checked="" type="checkbox"/>

로그 설정 메뉴 상단에서 **로그 2차 파서 설정** 탭을 선택하면 등록된 파서를 조회하고 추가 및 편집이 가능한 **파서 목록** 화면을 확인할 수 있습니다.

- 상단 오른쪽 **+ 추가하기** 버튼을 선택하면 **파서 추가** 창이 나타납니다.
- 파서 목록 **적용 순서** 컬럼의 **||** 아이콘을 드래그해 파서 설정 순서를 변경할 수 있습니다.
- 파서 목록 **활성화** 토글을 통해 파서 활성화 여부를 지정할 수 있습니다.
- 파서 목록 **수정** 및 **삭제** 아이콘을 통해 등록된 파서를 수정 및 삭제할 수 있습니다.

## 파서 등록 순서

로그 설정 메뉴 상단에서 **로그 2차 파서 설정** 탭을 선택해 로그 파서를 등록 및 수정할 수 있습니다. 다음은 파서 등록 시 공통 순서를 안내합니다.

X 파서 추가

<b>파서*</b>	파서(을) 선택해주세요 <span style="float: right;">v</span>
<b>카테고리*</b>	<p>4xx, 5xx 상태 코드 파서</p> <p>status가 파싱된 경우, 추가적으로 4xx, 5xx 상태 코드를 파싱하여 "4xx, 5xx 건수 데이터"를 생성함</p> <p><a href="#">4xx, 5xx 건수 데이터 생성</a></p>
<b>로그 검출 조건</b>	<p>상태 코드 성공률 파서</p> <p>status가 파싱된 경우, 추가적으로 2xx,3xx 상태 코드를 파싱하여 "요청 성공률 데이터"를 생성함</p> <p><a href="#">요청 성공률 데이터 생성</a></p>

1. **+ 추가하기** 버튼을 선택하면 **파서 추가** 창이 나타납니다.
2. **파서** 선택 창에서 파서를 선택하세요. 각 파서 설정 항목 및 제외할 상태 코드 등록에 관한 자세한 내용은 다음 문서를 참조하세요.
  - [4xx, 5xx 상태 코드 파서 설정 항목 및 제외할 상태 코드 등록](#)
  - [상태 코드 성공률 파서 설정 항목 및 제외할 상태 코드 등록](#)
3. **카테고리** 선택 창에서 카테고리를 선택하거나 직접 입력하세요.
4. **로그 검출 조건**을 선택하거나 직접 입력하세요.
5. **제외할 상태 코드**를 입력하세요.
6. **추가** 버튼을 선택해 파서를 등록하세요.

## 4xx, 5xx 상태 코드 파서 제외할 상태 코드 등록

X 파서 추가

**파서\*** 4xx, 5xx 상태 코드 파서 ▼

**카테고리\*** AppLog ▼  직접 입력

**로그 검출 조건** oname ▼ demo-8100 ▼  직접 입력

로그 검출 조건을 입력하지 않으면, 모든 로그를 대상으로 패턴을 적용합니다.

**제외할 상태 코드** 400 × 404 ×

입력한 상태 코드는 4xx, 5xx 상태 코드로 로그를 파싱할 때 제외됩니다.

추가

**4xx, 5xx 상태 코드 파서**는 status가 이미 파싱된 경우 사용할 수 있는 파서입니다. 파싱된 status를 이용하여 추가적으로 4xx, 5xx 상태 코드를 파싱합니다. 파싱한 데이터로 4xx, 5xx 건수 데이터를 생성할 수 있습니다. 제외할 상태 코드로 4xx, 5xx 상태 코드를 입력 또는 선택할 수 있습니다. 입력된 상태 코드는 로그에서 4xx, 5xx 상태 코드를 파싱할 때 제외됩니다.

### 설정 항목

설정 값	설명	기타
카테고리	4xx, 5xx건수 데이터를 생성할 카테고리입니다.	required
로그 검출 조건	필터로 적용할 검색 키, 검색 값을 입력합니다. 로그 검출 조건에 맞는 로그 데이터에 대해서만 4xx, 5xx건수 데이터를 생성합니다. 로그 검출 조건을 입력하지 않으면 모든 로그를 대상으로 데이터를 생성합니다.	optional
제외할	통계 데이터 생성 시 제외할 상태 코드입니다. 입력하지 않으면 4xx~5xx에 해당하는 전체 오류 상태	optional



설정 값	설명	기타
상태 코드	코드를 대상으로 4xx, 5xx건수 데이터를 생성합니다.	

## status 파서 등록 예시

**로그 1차 파서 설정** 수집된 로그에 대한 파서를 등록할 수 있습니다. 적용 순서대로 파서가 적용되며, 최초로 일치하는 파서만 적용됩니다. + 추가하기 저장

적용 순서	파서	카테고리	로그 검출 조건	패턴	활성화	
0	REGEX	AppLog	모든 로그	%(NUMBER:status)	<input type="checkbox"/>	<span style="font-size: 0.8em;">✎</span> <span style="font-size: 0.8em;">✖</span>
1	REGEX	AppLog	모든 로그	%(NUMBER:status)	<input type="checkbox"/>	<span style="font-size: 0.8em;">✎</span> <span style="font-size: 0.8em;">✖</span>
2	GROK	AppLog	모든 로그	%(NUMBER:status)	<input checked="" type="checkbox"/>	<span style="font-size: 0.8em;">✎</span> <span style="font-size: 0.8em;">✖</span>

유입되는 로그가 `{"msg":"message","status":404}` 이고 예시처럼 GROK 파서로 status를 파싱한다면, `status: 404` 와 같이 파싱됩니다. status가 정상적으로 파싱되는 것을 확인했다면 4XX,5XX 상태 코드 파서에서 제외할 상태 코드를 등록하세요.

## 데이터 조회

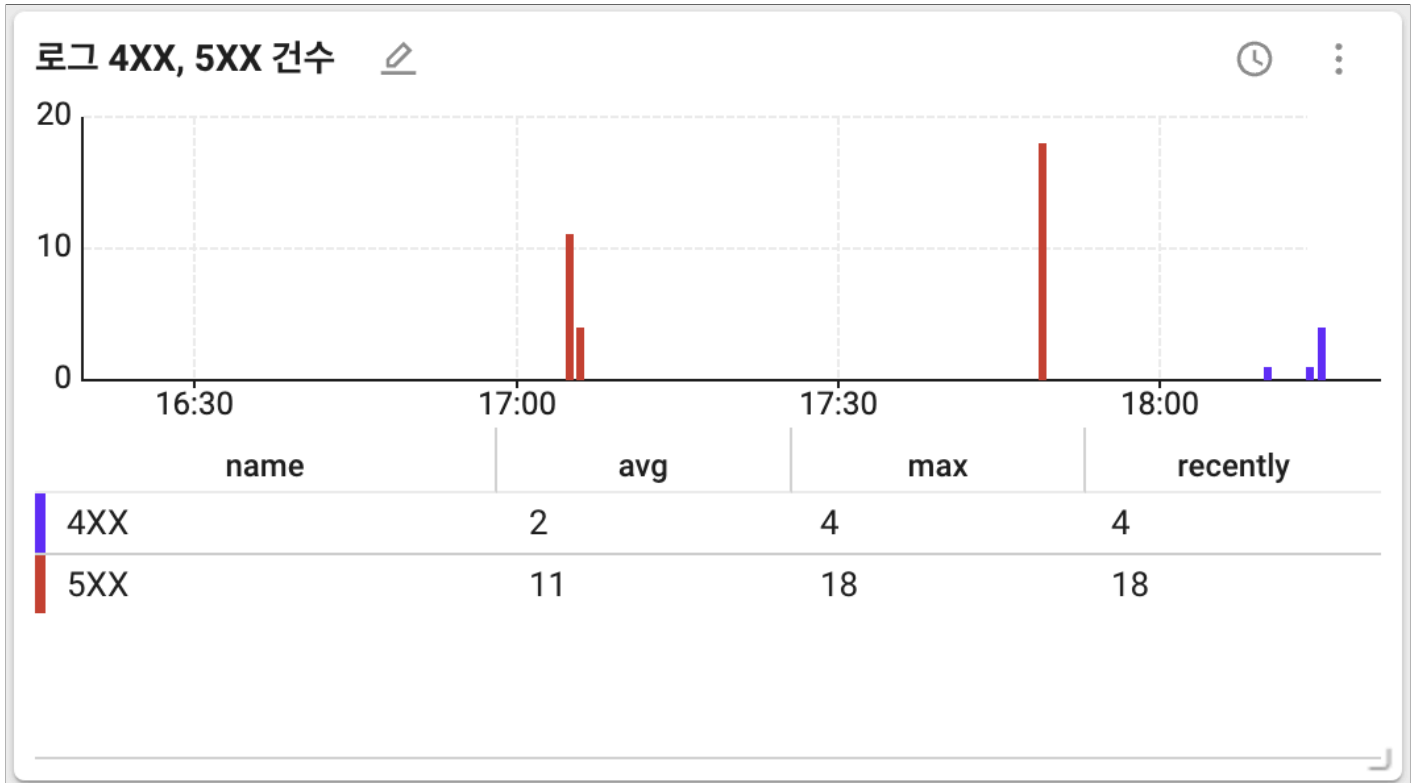
파서를 모두 등록하면 [Flex 보드](#)로 이동해 [로그 4XX, 5XX 건수](#) 위젯을 생성하세요.

**위젯 템플릿** ⇌ 모든 매트릭스

🔍

📈 로그 4XX, 5XX 건수

위젯을 생성하면 다음과 같이 데이터를 확인할 수 있습니다.



- **avg**: 조회 기간 데이터 평균값입니다.
- **max**: 조회 기간 데이터 중 최댓값입니다.
- **recently**: 조회 기간 데이터 중 마지막 값입니다.

## 상태 코드 성공률 파서 제외할 상태 코드 등록

X 파서 추가

**파서\*** 상태 코드 성공률 파서 ▼

**카테고리\*** AppLog ▼  직접 입력

**로그 검출 조건** oname ▼ demo-8100 ▼  직접 입력  
로그 검출 조건을 입력하지 않으면, 모든 로그를 대상으로 패턴을 적용합니다.

**제외할 상태 코드** 202 x  
입력한 상태 코드는 2xx, 3xx 상태 코드로 로그를 파싱할 때 제외됩니다.

추가

**상태 코드 성공률 파서**는 status가 이미 파싱된 경우 사용할 수 있는 파서입니다. status 파싱에 관한 내용은 [다음 문서](#)를 참조하세요. 파싱된 status를 이용해 추가로 2xx, 3xx 상태 코드를 파싱합니다. 파싱한 데이터로 HTTP 요청 성공률 데이터를 생성할 수 있습니다. 제외할 상태 코드에는 2xx, 3xx 상태 코드를 입력 또는 선택할 수 있습니다. 입력된 상태 코드는 로그에서 2xx, 3xx 상태 코드를 파싱할 때 제외됩니다.

### 설정 항목

설정 값	설명	기타
카테고리	요청 성공률 데이터를 생성할 카테고리입니다.	required
로그 검출 조건	필터로 적용할 검색 키, 검색 값을 입력합니다. 로그 검출 조건에 맞는 로그 데이터에 대해서만 요청 성공률 데이터를 생성합니다. 로그 검출 조건을 입력하지 않으면 모든 로그를 대상으로 데이터를 생성합니다.	optional

설정 값	설명	기타
제외할 상태 코드	요청 성공률 데이터 생성 시 제외할 상태 코드입니다. 입력하지 않으면 2xx~3xx에 해당하는 전체 성공 상태 코드를 대상으로 요청 성공률 데이터를 생성합니다.	optional

## 데이터 조회

파서를 모두 등록하면 [Flex 보드](#)로 이동해 로그 요청 성공률 위젯을 생성하세요.

위젯 템플릿
⇌ 모든 매트릭스

🔍

📄 로그 요청 성공률

위젯을 생성하면 다음과 같이 데이터를 확인할 수 있습니다.

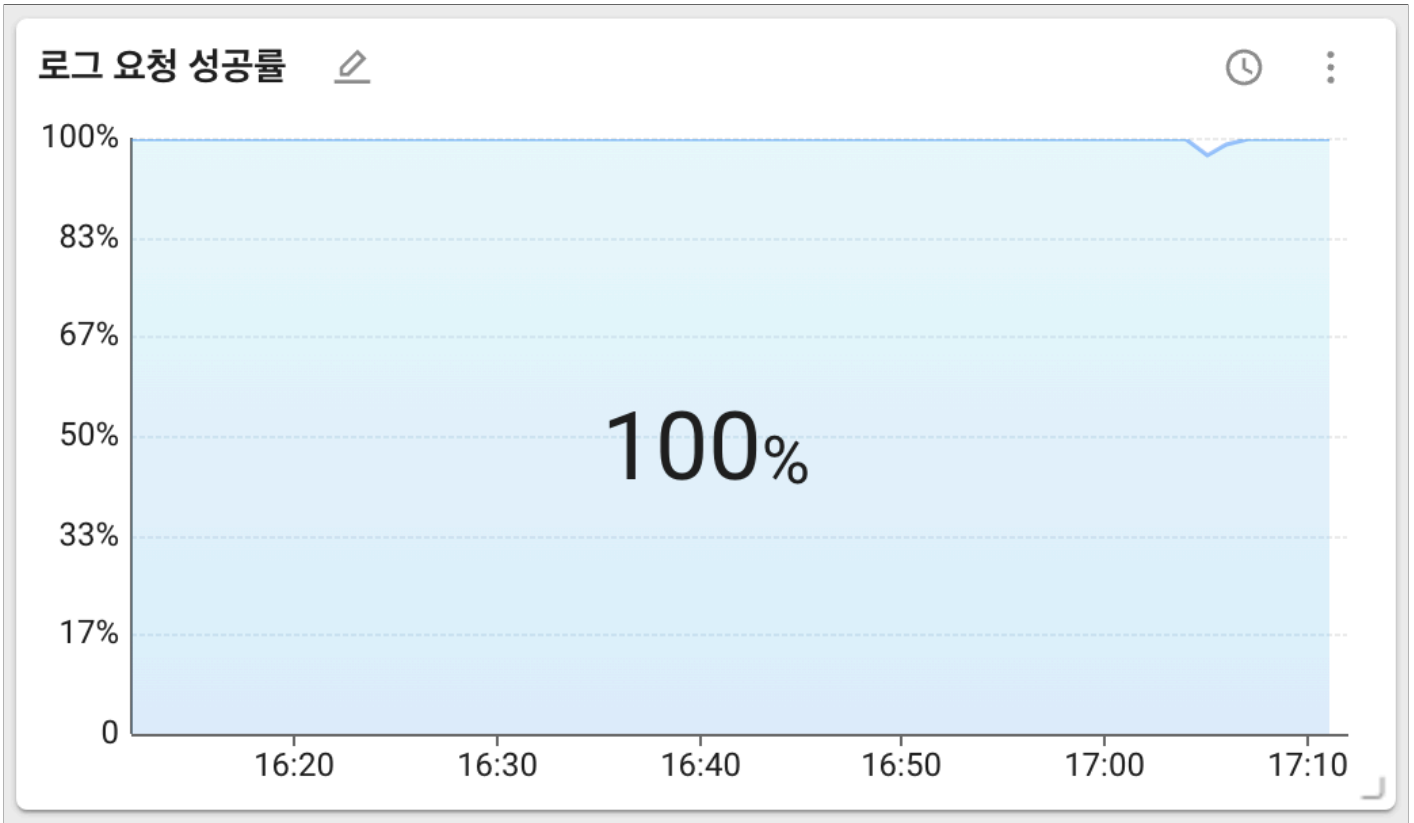


차트 위 데이터는 조회 기간에 대한 통계를 나타냅니다. 통계 방법을 최근값, 최댓값, 평균값 등으로 선택할 수 있습니다. 최근값이 기본으로 선택되어 있습니다.

## 빠른 인덱스 설정



[로그 설정](#) 메뉴 상단에서 [빠른 인덱스 설정](#) 탭을 선택하세요. 대량의 로그를 수집할 경우 로그 검색 성능이 현저하게 저하될 수 있습니다. 자주 사용하는 검색 조건을 **인덱스(index)**로 생성하면 로그 검색 성능을 개선해 빠른 탐색이 가능합니다. 설정 항목은 다음과 같습니다.


설정 값	필수 여부	설명
카테고리	필수	빠른 인덱스를 설정할 카테고리
검색 키	필수	빠른 인덱스를 설정할 검색 키

설정 값	필수 여부	설명
대소문자 구분 안 함	옵션	대소문자를 구분 여부
규칙	필수	* 한 개 이상 포함 필수
활성	필수	활성 또는 비활성 여부(기본값 true)

## 로그 설정 가져오기/내보내기

공통된 파서 설정 및 빠른 인덱스 설정 내용을 JSON 파일 형식으로 저장하고, 다른 프로젝트에서 JSON 파일을 가져와 적용할 수 있습니다. 프로젝트마다 여러 번 반복해서 설정을 작성하는 번거로움을 줄일 수 있습니다.

1. 하나의 프로젝트에 파서 설정 및 빠른 인덱스 설정을 추가하세요.
2. 각 설정 탭에서 화면 오른쪽 위에 **JSON**  버튼을 선택하세요.
3. **JSON 내보내기** 창이 나타나면 화면 오른쪽 위에 **내보내기** 버튼을 선택하세요.
4. JSON 설정 파일이 사용자 PC에 저장됩니다.
5. 다른 프로젝트로 이동한 다음 **로그** > **로그 설정** 메뉴로 이동하세요.
6. 앞서 JSON 설정 파일을 내보낸 설정 탭을 선택한 다음  버튼을 선택하세요.
7. 파일 선택 창이 나타나면 사용자의 PC에 저장한 JSON 설정 파일을 선택하세요.
8. **JSON 가져오기** 창이 나타나면 설정 내용을 확인한 다음 **목록에 추가하기** 또는 **덮어쓰기** 버튼을 선택하세요.
9. 화면 오른쪽 위에 **저장** 버튼을 선택하세요.

 JSON 설정 파일을 가져온 다음 **저장** 버튼을 선택하지 않으면 가져온 설정 내용을 저장할 수 없습니다.

## 로그 장기 보관 통계 설정

**로그 설정** 메뉴 상단에서 **로그 장기 보관 통계** 탭을 선택하세요. 로그 데이터는 용량이 매우 커서 장기간 보관하기가 어렵습니다. 따라서 로그 통계 데이터 설정 기능을 사용하여 **특정 조건을 만족하는 로그 데이터가 5분마다 몇 건씩 수집되었는지에 대한 정보를 저장할 수** 있습니다. 장기간 시 실제 로그 데이터는 삭제되어도 해당 조건을 만족하는 로그가 얼마나 수집되었는지 추이를 확인할 수 있습니다.

## 로그 장기 보관 통계 추가

×
로그 장기 보관 통계 추가

**통계 키\***

**데이터 보관일(디스크 사용량)**  ▼

**카테고리\***

**로그 검출 조건\***  ▼

X 검색 값을(를) 선택해주세요

**제외**  대소문자 구분

▼

X 검색 값을(를) 선택해주세요

**제외**  대소문자 구분

[+ 추가하기](#)

추가

로그 장기 보관 통계 탭에서 + 추가하기 버튼을 선택하면 로그 장기 보관 통계 추가 창이 나타납니다. + 추가하기 버튼을 통해 규칙을 추가하거나 생성한 규칙을 - 아이콘을 통해 삭제할 수 있습니다.

## 설정 항목

필드	설명
카테고리	규칙을 적용할 카테고리입니다.
통계 키(key)	규칙을 만족하는 로그 발생 시 저장할 키 값으로 동일한 키를 중복으로 설정할 수 없습니다.
로그 검출	로그 통계 데이터를 생성할 조건입니다. 이 조건을 만족하는 로그가 얼마나 수집되었는지를 기반으로 통계

필드	설명
조건	데이터를 생성합니다.
제외	제외를 체크하면 입력한 조건에 해당하지 않는 값으로 통계 데이터를 생성합니다.
대소문자 구분	입력한 로그 검색 조건의 값에 대해서 대소문자 구분 여부를 지정합니다.
활성	활성 또는 비활성 여부(기본값 true)

## 예시

다음과 같이 설정을 추가한 경우 수집된 로그 중 status가 200, 300 인 로그를 대상으로 TotalCount라는 키값으로 통계 데이터를 생성합니다.

카테고리	통계 키	로그 검색 조건	데이터 보관일(디스크 사용량)	활성화	
AppLog	TotalCount	status 200, 300	15일 (약 3 MB)	<input type="checkbox"/>	

## 데이터 조회

1. Flex 보드의 위젯 템플릿에서 로그 장기 보관 통계를 검색해 위젯을 생성하세요.

위젯 템플릿
⇌ 모든 메트릭스

🔍

로그 장기 보관 통계



2. 데이터를 조회할 **카테고리**와 **키**를 지정한 뒤 **적용** 버튼을 선택합니다.

### 로그 장기 보관 통계 ✎

i 카테고리, 키를 선택해 주세요.

해당 위젯이 나타낼 데이터 수집을 위해서는 **로그 장기 보관 통계 설정**이 필요합니다.  
해당 설정을 하지 않은 경우,  
[가이드](#)를 참고하여 **로그 장기 보관 통계 설정**을 해주세요.  
[로그 설정](#)

**카테고리 변경**

AppLog ▼

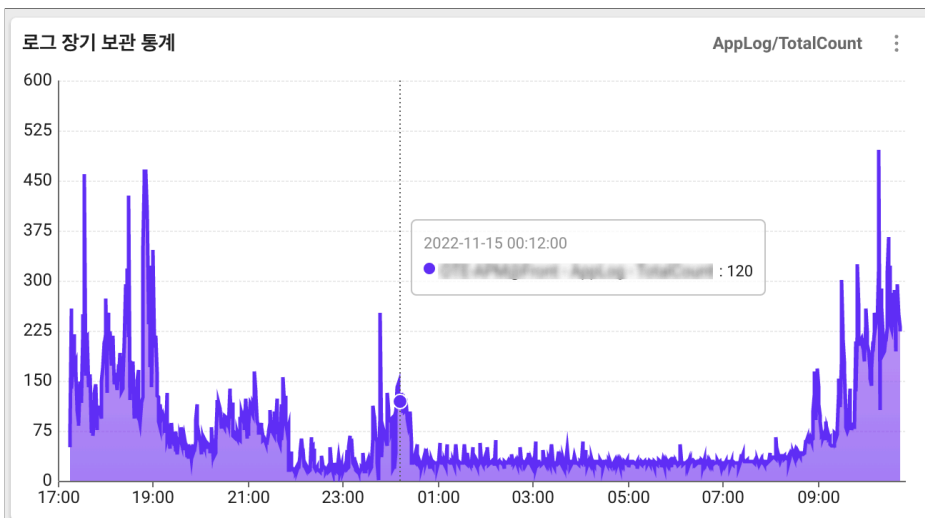
---

**통계 키**

TotalCount ▼

취소
적용

3. 추가한 설정값으로 **로그 장기 보관 통계** 데이터를 다음처럼 확인할 수 있습니다.



# 로그 파싱하기

로그 파서를 사용하면 불규칙한 형태의 로그를 쿼리가 가능한 구조화된 형태로 변경할 수 있습니다. 와탭 로그 모니터링은 다음과 같이 두 가지 유형의 파서를 제공합니다.

- **GROK 파서**: 임의의 형태로 수집되는 로그를 정규 표현식과 GROK 문법을 활용해 파싱합니다.
- **JSON 파서**: JSON 형태로 수집되는 로그를 파싱합니다.

## ① 공통 주의사항

- 같은 카테고리에 여러 개의 파서가 등록되어 있는 경우 첫 번째로 매칭되는 파서만 적용됩니다.
- 와탭은 와탭 서비스의 안정성에 영향을 줄 수 있는 파서를 비활성화할 수 있는 권한을 가집니다.

## GROK 파서

로그가 불규칙한 형태로 수집되는 경우 GROK 파서를 사용해 로그를 파싱할 수 있습니다. GROK 문법은 named regular expressions를 제공해 정규 표현식을 보다 쉽고 편리하게 사용할 수 있습니다.

GROK 파서 패턴 등록에 관해 다음 동영상 가이드를 참조하세요.

## GROK 시작하기

GROK은 두 가지 형태의 문법을 제공합니다.

1. `%{SYNTAX:SEMANTIC}`: GROK 라이브러리에서 제공하는 문법입니다. **named regular expressions**를 활용해 태그를 추출할 수 있습니다. 활용 예시는 [다음](#)을 참조하세요.
  - **SYNTAX**: GROK이 제공하는 named regular expressions를 지정합니다.
  - **SEMANTIC**: 매칭되는 값에 부여할 이름을 지정합니다.

### ① named regular expressions

GROK에서 제공하는 문법입니다. 복잡한 정규 표현식에 이름을 부여해 사용할 수 있도록 GROK에서 제공하는 기능입니다.

name	regular expression
WORD	<code>\b\w+\b</code>
SPACE	<code>\s*</code>
NOTSPACE	<code>\S+</code>
UUID	<code>[A-Fa-f0-9]{8}-(?:[A-Fa-f0-9]{4}-){3}[A-Fa-f0-9]{12}</code>

와탭에서 제공하는 모든 named regular expressions 확인을 원한다면 다음 [링크](#)를 참조하세요.

2. `?<SEMANTIC>REGX`: 정규 표현식의 **named capturing group** 문법입니다. 정규 표현식을 활용해서 사용자의 의도에 맞게 태그를 추출할 수 있습니다. 활용 예시는 [다음](#)을 참조하세요.
  - **SEMANTIC**: 매칭되는 값에 부여할 이름을 지정합니다.
  - **REGX**: 매칭에 사용할 정규 표현식을 입력합니다.

### ① named capturing group

정규 표현식에서 제공하는 문법입니다.

- capturing group: 여러 개의 토큰을 하나로 묶어 하나의 매칭 단위로 사용하는 기능을 의미합니다.
- named capturing group: capturing group에 이름을 부여한 것입니다.
- 문자열 매칭 예시를 살펴보겠습니다. [dev@whatap.io](#)

◦ 예시 1 `(\w+)@(\w+\.\w+)`

◦ 예시 2 이메일 전체 매칭 및 username과 domain 추가 매칭 시 `(?<username>\w+)@(?<domain>\w+\.\w+)`

**%(SYNTAX:SEMANTIC) 활용 예시**

다음은 %(SYNTAX:SEMANTIC) 문법을 활용하는 예시입니다.

**Sample log**

```
[2023-08-08 02:02:30,101 GMT][INFO ][i.w.y.l.c.LogSinkDexScheduleThread.realProcess(159)] 8 VirtualLog 20230808 02:01:00.000 {area=4, city=5} 56ms
```

샘플 로그를 보고 각 단어가 의미하는 내용을 유추할 수 있습니다. 각 부분을 semantic한 단어로 치환 시 다음과 같이 표현할 수 있습니다.

**semantic replace**

```
[date][logLevel][caller] projectCode logCategory dexBuildStartTime {area=areaEnum, city=cityEnum} dexBuildElapsed
```

semantic한 단어 모두 정규 표현식으로 대체할 수 있습니다. GROK 파서를 사용하면 사전 정의된 named regular expressions를 활용할 수 있습니다. 여기서 사용된 `TIMESTAMP_ISO8601`, `LOGLEVEL`, `DATA` 는 GROK에서 제공하는 named regular expressions입니다. 이 값들은 각각 다음의 정규 표현식으로 대체되어 매칭됩니다.

- name: `TIMESTAMP_ISO8601`
  - regular expression: `%(YEAR)-%(MONTHNUM)-%(MONTHDAY)[T ]%(HOUR):?(MINUTE):?(?:?(SECOND))?(?:%(ISO8601_TIMEZONE)?`
- name: `LOGLEVEL`
  - regular expression: `LOGLEVEL`  
`[(Aa)lert|ALERT|[Tt]race|TRACE|[Dd]ebug|DEBUG|[Nn]otice|NOTICE|[Ii]nfo|INFO|[Ww]arn?:?ing?|WARN?:?(?:ING)?|[Ee]rr?:?(?:or)?|ERR?:?(?:OR)?|[Cc]rit?:?(?:ical)?|CRIT?:?(?:ICAL)?|[Ff]atal|FATAL|[Ss]evere|SEVERE|EMERG(?::ENCY)?|[Ee]merg(?::ency)?`
- name: `DATA`
  - regular expression: `.*`

**GROK parsing pattern**

```
\[(TIMESTAMP_ISO8601:date)\sGMT\]\[(LOGLEVEL:level)\s\]\[(DATA:caller)\]
```

위와 같은 문법으로 파싱을 하면 다음과 같이 태그를 추출할 수 있습니다. 이렇게 GROK의 %(SYNTAX:SEMANTIC) 문법은 복잡하고 긴 정규 표현식을 쉽고 간결하게 적용할 수 있도록 도와주는 역할을 합니다.

**Tag extraction**

```
- date : 2023-08-08 02:02:30,101
- caller : i.w.y.l.c.LogSinkDexScheduleThread.realProcess(159)
- level : LEVEL
```

**(?<SEMANTIC>REGX) 활용 예시**

named regular expressions로 매칭되지 않는 부분은 (?<SEMANTIC>REGX) 패턴을 사용해서 파싱할 수 있습니다. 위의 [샘플 로그](#)에서 %(SYNTAX:SEMANTIC) 문법만으로 파싱되지 않는 영역은 다음과 같습니다.

**Unparsed area**

```
8 VirtualLog 20230808 02:01:00.000 {area=4, city=5} 56ms
```

해당 로그의 각 부분을 semantic한 단어로 치환 시 다음과 같이 표현할 수 있습니다.

**semantic replace**

```
projectCode logCategory dexBuildStartTime {area=areaEnum, city=cityEnum} dexBuildElapsed
```

이렇게 불규칙한 형태의 문자열은 다음과 같은 (?<SEMANTIC>REGX) 문법을 사용해 파싱할 수 있습니다.

샘플 로그 파싱 키워드별 매칭되는 정규 표현식

파싱 키워드	(?<SEMANTIC>REGX)
8	(?<projectCode>\d)
VirtualLog	(?<logCategory>\w*)
20230808 02:01:00.000	(?<dexBuildStartTime>\d{8}\s\d{2}:\d{2}:\d{2}\.\d{3})
area=4	area=(?<areaEnum>\d)
city=5	city=(?<cityEnum>\d)
56ms	(?<dexBuildElapsed>\d{2})ms

> 기본 정규 표현식 문법

문법	의미	별칭
?	0 or 1	-
+	1 or more	-
*	0 or more	-
a{5}	exactly 5	-
\w	word character	[a-zA-Z_0-9]
\s	white space	-
.	any character except newline	-
[abc]	any of	-
[^abc]	not a,b, or c	-
[a-z]	character between a and z	-
[1-3[7-9]]	union (combining two or more character classes)	-
[1-6&&[3-9]]	intersection (교집합)	-
[0-9&&[^2468]]	subtraction (차집합)	-
a{2,}	2 or more	-
a{1,3}	between 1 and 3	-
a+?	match as few as possible	-
{2,3}?	match as few as possible	-
(abc)	capturing group (여러 개의 문자열을 single unit으로 처리함)	-
\d	digit	[0-9]

문법	의미	별칭
\D	non-digit	[^0-9]
\W	non-word character	-
\S	non-white space	-

이렇게 파싱된 키워드를 띄어쓰기(\s)와 특수 문자 escape(\x, \., \y)로 연결하면 다음과 같이 패턴을 적용할 수 있습니다.

```
GROK parsing pattern
(?<projectCode>d)s(?<logCategory>\w*)s(?<dexBuildStartTime>d{8})\s
d(2):d(2):d(2)\.d(3))\s(area=?<areaEnum>d)\,s(city=?<cityEnum>d)\)\s(?<dexBuildElapsed>d(2))ms
```

위와 같은 문법으로 파싱을 하면 다음과 같이 태그를 추출할 수 있습니다.

```
Tag extraction
- projectCode : 8
- logCategory : VirtualLog
- dexBuildStartTime : 20230808 02:01:00.000
- areaEnum : 4
- cityEnum : 5
- dexBuildElapsed : 56
```

## GROK 적용하기

로그 설정 > 로그 1차 파서 설정

1. GROK 패턴 파서를 적용하려면 **로그 설정** 메뉴의 **로그 1차 파서 설정** 탭으로 이동하세요.



2. **+ 추가하기**를 선택 후 **파서** 입력란에서 **GROK** 파서를 선택하세요.

파서\*

카테고리\*

로그 검색 조건

패턴\*

3. **카테고리** 및 **로그 검색 조건**, **패턴**을 입력하세요. **파서 추가** 창의 구성 요소는 다음과 같습니다.

- **카테고리**

로그 카테고리를 선택하세요. **카테고리**는 필수로 입력해야 합니다.

- **로그 검출 조건**

- 조건에 만족하는 로그만 파서가 적용됩니다.
- **검색 키**와 **검색 값**을 선택하거나 직접 입력하세요.
- **로그 검출 조건**은 모든 파서가 수행되기 전에 적용됩니다. 즉 파서의 결과로 추가되는 **태그**를 사용할 수 없습니다.

- **패턴**

GROK 패턴을 지정하세요. 필수로 입력해야 합니다.

#### 4. 추가 버튼을 선택해 파서를 등록하세요.

- ① • 로그 파서 목록에서 해당 파서의 **적용 순서**를 변경하거나 **활성화** 및 **수정**, **삭제**할 수 있습니다.
- 파서를 등록하기 전에 **시뮬레이션**을 통해 등록하려는 패턴이 정상적인지 확인할 수 있습니다.

#### ① GROK 파서 주의사항

- GROK 파서는 `%(SYNTAX:SEMANTIC)`, `%(SYNTAX:SEMANTIC)` 두 가지 패턴을 지원합니다.
- `%(SYNTAX:SEMANTIC)` 패턴 사용 시 `SEMANTIC` 을 반드시 입력해야 합니다.
- `%(SYNTAX:SEMANTIC)` 패턴 사용 시 `SEMANTIC` 은 하나의 파서 안에서 **unique** 해야 합니다.
- `(?<SEMANTIC>REGX)` 패턴 사용 시 `SEMANTIC` 은 문자(a-z, A-Z)와 숫자(0-9) 그리고 지정된 특수문자(`.`, `_`, `-`)만 쓸 수 있습니다.
- `SEMANTIC` 은 문자(a-z, A-Z)로 시작해야 합니다.
- `SEMANTIC` 은 문자(a-z, A-Z) 또는 숫자(0-9)로 끝나야 합니다.

## 시뮬레이션

파서 시뮬레이션 창에서 **로그**와 **패턴**을 입력해 파싱 결과를 미리 확인할 수 있습니다.

**로그** 예시: `[2023-08-08 02:02:30,101 GMT][INFO ][i.w.y.l.c.LogSinkDexScheduleThread.realProcess(159)] 8 VirtualLog 20230808 02:01:00.000 {area=4, city=5} 56ms`

**패턴** 예시:

```
\[%{TIMESTAMP_ISO8601:date}\sGMT\}\[%{LOGLEVEL:level}\s\}\[%{DATA:caller}\s(?:<projectCode>\d)\s(?:<logCategory>\w*)\s(?:<dexBuildStartTime>\d{8})\s\d{2}\:\d{2}\:\d{3}\s\}\s\{area=(?:<areaEnum>\d)\,city=(?:<cityEnum>\d)\}\s(?:<dexBuildElapsed>\d{2})\}ms
```

1. **파서 추가** 창에서 **시뮬레이션** 버튼을 선택하세요.
2. **파서 시뮬레이션** 창에서 **로그**와 **패턴**을 입력하세요.
3. **로그**와 **패턴** 입력 후 **시뮬레이션** 버튼을 선택하세요. 다음과 같이 **시뮬레이션 결과**를 확인할 수 있습니다.

✕
파서 시물레이션

① 입력한 패턴으로 로그가 성공적으로 파싱되는지 시물레이션합니다.

**로그**

```
[2023-08-08 02:02:30,101 GMT][INFO ][i.w.y.l.c.LogSinkDexScheduleThread.realProcess(159)] 8 VirtualLog 20230808 02:01:00.000 {area=4, city=5} 56ms
```

**패턴**

```
[\%(TIMESTAMP_ISO8601:date)\sGMT\s\%\%(LOGLEVEL:level)\s\%\%(DATA.caller)\s\%(?<projectCode>d)\s\%(?<logCategory>w*)\s\?(<dexBuildStartTime>d(8)\s\d(2)\s\d(2)\s\d(3))\s\%(area=?<areaEnum>d)\s\%(cityEnum=?<d>d)\s\?(<dexBuildElapsed>d(2))ms
```

패턴 적용
시물레이션

**시물레이션 결과**

키	date
값	2023-08-08 02:02:30,101
결과	Ok
키	caller
값	i.w.y.l.c.LogSinkDexScheduleThread.realProcess(159)
결과	Ok
키	cityEnum
값	5
결과	Ok
키	level
값	INFO
결과	Ok
키	projectCode
값	8
결과	Ok
키	areaEnum
값	4
결과	Ok
키	dexBuildStartTime
값	20230808 02:01:00.000
결과	Ok
키	dexBuildElapsed
값	56
결과	Ok
키	logCategory
값	VirtualLog
결과	Ok

## JSON 파서

로그가 JSON 포맷으로 수집될 경우 JSON 파서를 사용해 쉽고 편리하게 파싱할 수 있습니다.

### JSON 적용하기

[로그 설정](#) > [로그 1차 파서 설정](#)

1. JSON 패턴 파서를 적용하려면 [로그 설정](#) 메뉴의 [로그 1차 파서 설정](#) 탭으로 이동하세요.





## JSON 포맷 활용 예시

### Sample log

```
{"host": "10.21.3.24", "method": "POST", "status": "200", "url": "http://devote.whatap.io/yard/api/flush"}
```

위와 같은 샘플 로그가 수집된 경우 **파서 추가** 창에서 **JSON** 파서를 선택하세요. 복잡한 파싱 로직을 작성할 필요없이 로그 분석에 필요한 **태그**를 다음과 같이 추출할 수 있습니다.

### Tag extraction

```
- host : 10.21.3.24
- method : POST
- status : 200
- url : http://dev.whatap.io/yard/api/flush
```

## JSON 포맷 일부 구성 시 활용 예시

### Some JSON format sample log

```
2023-08-08 02:43:28,615 -- {"host": "10.21.3.24", "method": "POST", "status": "200", "url": "http://devote.whatap.io/yard/api/flush"} --
```

만약 예시와 같이 로그의 일부만 JSON 포맷으로 구성되어있다면 **Prefix**와 **Postfix**를 지정해 주세요. 와탭 로그 모니터링은 **Prefix**와 **Postfix** 사이의 영역을 JSON 포맷으로 인식 후 파싱합니다.

### Tag extraction

```
- host : 10.21.3.24
- method : POST
- status : 200
- url : http://dev.whatap.io/yard/api/flush
```

# 로그 타임스탬프 기준

본 문서는 로그 모니터링 과정 중 처리(Processing) 단계에서 로그 타임스탬프 기준 시간에 대해 안내합니다.

## 타임스탬프 기준 시간

타임스탬프	로그
2023-11-27 14:26:00.343 와탭 수집 서버 시간	agenttime 1701062760115 에이전트 수집 시간
	[2023-11-27 05:25:56,588 GMT] 로그 발생 시간

로그 타임스탬프 기준 시간이 와탭 에이전트가 로그를 수집한 시간에서 와탭 수집 서버의 로그 처리 시간으로 변경되었습니다.

일반적인 상황에서는 변경 전과 큰 차이없이 기존 방식과 동일하게 로그를 검색할 수 있습니다. 변경 후 다음의 경우에도 사용자의 추가적인 수정 없이 로그 모니터링을 일관적으로 사용할 수 있습니다.

- NTP 사용 시, 모니터링 대상의 서버 시간이 표준 시간보다 과거 또는 미래 시간으로 설정되어있는 경우
- NTP 미사용 시, 2개 이상인 모니터링 대상의 서버 시간이 서로 다른 경우

## 타임셀렉터 조회 범위 지정

타임스탬프 기준 시간 변경으로 타임셀렉터 역시 와탭 수집 서버 시간을 기준으로 동작합니다. 이에 따라 에이전트 수집 시간과 와탭 수집 서버 시간에 차이가 생겨 조회 범위에 포함되지 않는 로그가 발생할 수 있습니다. 이 경우 조회 범위를 넓게 지정 시 조회가 가능합니다.

## 로그 트렌드

로그 트렌드 추이 차트의 X축은 최소 1분 기준입니다. 로그 기준 시간이 변경된 후에도 차트를 통한 전체적인 추이 파악에는 영향을 주지 않습니다.

## 로그 검색

로그 메시지에 로그 생성 시간을 나타내는 인덱스 agenttime 이 추가되었습니다. agenttime 값을 통해 에이전트 수집 시간을 확인하세요.

- ① UTC를 따르는 **와탭 수집 서버 시간**은 사용자 브라우저 시간에 따라 변환해 표기합니다.  
예시, 한국의 경우 UTC+9 기준으로 시간을 표기합니다.

## 단계별 기준 시간

로그는 다음의 3 단계를 거쳐 수집됩니다. 각 단계마다 서로 다른 기준 시간이 사용될 수 있습니다.

### 1. 로그 발생 시간

모니터링 대상의 시간 또는 Logging 정책에 따라 편차가 발생할 수 있습니다.

### 2. 에이전트 수집 시간

사용 중인 상품 또는 에이전트가 로그를 수집하는 방법 및 생성되는 로그에 따라 편차가 발생할 수 있습니다.

- **Application**

설정에 따라 로그 라이브러리 또는 로그 파일에서 실시간에 가깝게 로그를 수집합니다.

- ① 로그 라이브러리는 Java 상품의 경우만 지원합니다.

- **Server**

로그 파일에서 실시간에 가깝게 로그를 수집합니다.

- **AWS Log**

AWS Resource 정책에 따라 준 실시간 또는 수 분마다 로그를 수집합니다.

### 3. 와탭 수집 서버 시간

모니터링 대상 또는 로그 생성 방법과 상관없이 수집 서버에 저장되는 시간을 사용합니다.

**와탭 에이전트 수집 시간**과 **와탭 수집 서버 시간**은 큰 차이가 있습니다. 모니터링 대상의 서버 시간을 확인하세요. 모니터링 대상의 서버 시간이 과거 또는 미래 시간으로 설정된 경우 에이전트 수집 시간에 영향을 끼칩니다.

# 주요 메뉴 알아보기

와탭의 로그 모니터링은 통합 시스템 구축을 바탕으로 사용자 편의성과 접근성을 높였습니다. 와탭은 자체 기술력을 기반으로 탄탄한 데이터 수집을 통해 사용자들이 주로 사용하는 라이브 테일, 로그 트렌드, 로그 검색, 이벤트 알림은 물론 Parser의 효율성을 지원합니다.

로그 메뉴는 조회 및 분석과 옵션 설정 등의 기능을 제공합니다. [이벤트 설정](#) 메뉴는 로그 관련 이벤트 알림을 설정할 수 있습니다.

- 홈 화면 > 프로젝트 선택 > [로그](#)

와탭 모니터링 서비스 초기 화면에서 프로젝트를 선택한 다음 프로젝트 메뉴 하위에 [로그](#) 메뉴를 선택하세요. 다음 기능을 활용하면 복잡한 로그에 보다 손쉽게 접근하여 다양한 조건으로 실시간 확인 및 분석이 가능합니다.

- [라이브 테일](#)
- [로그 트렌드](#)
- [로그 검색](#)

- 홈 화면 > 프로젝트 선택 > [경고 알림](#) > [이벤트 설정](#)

와탭 모니터링 서비스 초기 화면에서 프로젝트를 선택한 다음 프로젝트 메뉴 하위에 [경고 알림](#) 메뉴를 선택해 [이벤트 설정](#) 메뉴에 진입하세요. 이벤트 조건을 설정하고 이메일, SMS, 메신저, App Push 등 다양한 경로로 알림을 수신할 수 있습니다.

와탭 로그 모니터링 서비스의 주요 메뉴 안내를 다음과 같이 제공합니다.

## 라이브 테일

로그 모니터링 라이브 테일을 안내합니다.

## 로그 트렌드

로그 모니터링의 로그 트렌드 메뉴를 안내합니다.

## 로그 검색

로그 모니터링의 로그 검색 메뉴를 안내합니다.



### 알림 설정하기

로그 이벤트 알림 메뉴를 안내합니다.

### 경고 알림 수신 설정

프로젝트에 포함하는 멤버들의 경고 알림 수신과 관련한 다양한 기능을 설정할 수 있습니다.

### 이벤트 기록

이벤트 기록 메뉴를 통해 발생한 경고 알림 이력을 확인할 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 접근 없이 모니터링 화면상에서 로그 데이터 스트림을 쉽게 확인할 수 있습니다. 대량의 로그 중 필요한 로그를 선별하고 하이라이트 기능을 통해 원하는 로그를 빠르게 인지할 수 있습니다.

라이브 테일
🔍 🔔 🗄️ ⋮ 👤

카테고리  
AppLog

필터  
필터를(을) 입력 후 엔터를 눌러 추가하세요.

🔍

Content 필터

🔍 ⏸ ⏪ ⏩ ⚙️

▶	oname	타임스탬프	로그
		09:06:52.656	select productid
▶	demo-8103	2023-08-30 09:06:52.656	<a href="#">@txid</a> -636988446309675660 <a href="#">pcode</a> 5490 <a href="#">oname</a> demo-8103 <a href="#">onodeName</a> node-1 <a href="#">oid</a> 633280970 <a href="#">okindName</a> demo-okind-1 select distinct pp.lastname, pp.firstname
▶	demo-8103	2023-08-30 09:06:52.657	<a href="#">@txid</a> 4276189847602958291 <a href="#">pcode</a> 5490 <a href="#">oname</a> demo-8103 <a href="#">onodeName</a> node-1 <a href="#">oid</a> 633280970 <a href="#">okindName</a> demo-okind-1 select distinct pp.lastname, pp.firstname
▶	demo-8103	2023-08-30 09:06:52.660	<a href="#">@txid</a> -5273513173028476407 <a href="#">pcode</a> 5490 <a href="#">oname</a> demo-8103 <a href="#">onodeName</a> node-1 <a href="#">oid</a> 633280970 <a href="#">okindName</a> demo-okind-1 select
▶	demo-8103	2023-08-30 09:06:52.658	<a href="#">@txid</a> -4475135982985646809 <a href="#">pcode</a> 5490 <a href="#">oname</a> demo-8103 <a href="#">onodeName</a> node-1 <a href="#">oid</a> 633280970 <a href="#">okindName</a> demo-okind-1 select tutorial_id, tutorial_title,
▶	demo-8103	2023-08-30 09:06:52.662	<a href="#">@txid</a> -636988446309675660 <a href="#">pcode</a> 5490 <a href="#">oname</a> demo-8103 <a href="#">onodeName</a> node-1 <a href="#">oid</a> 633280970 <a href="#">okindName</a> demo-okind-1 select productid, avg(orderqty) as averagequantity, sum(linetotal) as total
▶	demo-8103	2023-08-30 09:06:52.663	<a href="#">@txid</a> 4276189847602958291 <a href="#">pcode</a> 5490 <a href="#">oname</a> demo-8103 <a href="#">onodeName</a> node-1 <a href="#">oid</a> 633280970 <a href="#">okindName</a> demo-okind-1 select ename, job, sal + 100 from emp
▶	demo-8103	2023-08-30 09:06:52.662	<a href="#">@txid</a> -7150002028998370687 <a href="#">pcode</a> 5490 <a href="#">oname</a> demo-8103 <a href="#">onodeName</a> node-1 <a href="#">oid</a> 633280970 <a href="#">okindName</a> demo-okind-1 select (100-25)/15*(20-3) from dual
▶	demo-8103	2023-08-30 09:06:52.664	<a href="#">@txid</a> -4475135982985646809 <a href="#">pcode</a> 5490 <a href="#">oname</a> demo-8103 <a href="#">onodeName</a> node-1 <a href="#">oid</a> 633280970 <a href="#">okindName</a> demo-okind-1 select 'total income is', ((orderqty * unitprice) * (1.0 - unitpricediscount)), ' for ',
▶	demo-8103	2023-08-30 09:06:52.666	<a href="#">@txid</a> -5273513173028476407 <a href="#">pcode</a> 5490 <a href="#">oname</a> demo-8103 <a href="#">onodeName</a> node-1 <a href="#">oid</a> 633280970 <a href="#">okindName</a> demo-okind-1 select ename, job, sal + 100 from emp
▶	demo-8103	2023-08-30 09:06:52.667	<a href="#">@txid</a> -636988446309675660 <a href="#">pcode</a> 5490 <a href="#">oname</a> demo-8103 <a href="#">onodeName</a> node-1 <a href="#">oid</a> 633280970 <a href="#">okindName</a> demo-okind-1 select * into dbo.newproducts
▶	demo-8103	2023-08-30 09:06:52.668	<a href="#">@txid</a> 4276189847602958291 <a href="#">pcode</a> 5490 <a href="#">oname</a> demo-8103 <a href="#">onodeName</a> node-1 <a href="#">oid</a> 633280970 <a href="#">okindName</a> demo-okind-1 select distinct ename, deptno, sal, job from emp
▶	demo-8103	2023-08-30 09:06:52.668	<a href="#">@txid</a> -7150002028998370687 <a href="#">pcode</a> 5490 <a href="#">oname</a> demo-8103 <a href="#">onodeName</a> node-1 <a href="#">oid</a> 633280970 <a href="#">okindName</a> demo-okind-1 select corpus, count_corpus_words
▶	demo-8103	2023-08-30 09:06:52.671	<a href="#">@txid</a> 6994271475421077751 <a href="#">pcode</a> 5490 <a href="#">oname</a> demo-8103 <a href="#">onodeName</a> node-1 <a href="#">oid</a> 633280970 <a href="#">okindName</a> demo-okind-1 select productid, avg(orderqty) as averagequantity, sum(linetotal) as total

라이브 테일 메뉴에서 복잡한 로그들도 손쉽게 접근 가능합니다. 필요에 따라 필터 혹은 하이라이트 등의 기능을 활용해 실시간으로 조회할 수 있습니다. 로그 데이터 조회 주기는 2초입니다. 주요 용어는 다음과 같습니다.

- **Category**: 로그의 수집 및 조회 단위입니다.

- **Content:** 로그 메시지입니다.
- **Search Key:** 로그 파서 설정을 통해 생성합니다.
- **Tag:** 수집된 로그를 검색할 수 있는 검색 키입니다.

## 에이전트 옵션

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

```

2023-12-18 14:31:02.563 [level] INFO [pcode] 2277 [agenttime] 1702877462042 [oid] 778873916 [category] AppLog [loggerName] io.home.test.logback02starter.base.web.LogbackControllerGreeting
INFO Log in our greeting method.
2023-12-18 14:31:02.563 [level] WARN [pcode] 2277 [agenttime] 1702877462043 [oid] 778873916 [category] AppLog [loggerName] io.home.test.logback02starter.base.web.LogbackControllerError [threadName] http-nio-19090-exec-2
io.home.test.logback02starter.base.errors.exception.ApiException: [2204] Process failure. Please try again.
2023-12-18 14:31:02.586 [level] error [pcode] 2277 [agenttime] 1702877462043 [oid] 778873916 [category] AppServer [event_status] error [event_status_literal_name] level
Servlet.service() for servlet [DispatcherServlet] in context with path [] threw exception [Request processing failed: io.home.test.logback02starter.base.error
2023-12-18 14:31:02.586 [level] error [pcode] 2277 [agenttime] 1702877462057 [oid] 778873916 [category] AppServer [event_status] error [event_status_literal_name] level
Servlet.service() for servlet [DispatcherServlet] in context with path [] threw exception [Request processing failed: io.home.test.logback02starter.base.error
2023-12-18 14:31:02.586 [level] error [pcode] 2277 [agenttime] 1702877462081 [oid] 778873916 [category] AppServer [event_status] error [event_status_literal_name] level
Servlet.service() for servlet [DispatcherServlet] in context with path [] threw exception [Request processing failed: io.home.test.logback02starter.base.error
2023-12-18 14:31:02.586 [level] error [pcode] 2277 [agenttime] 1702877462087 [oid] 778873916 [category] AppServer [event_status] error [event_status_literal_name] level
Servlet.service() for servlet [DispatcherServlet] in context with path [] threw exception [Request processing failed: io.home.test.logback02starter.base.error
    
```

### ! 에이전트 옵션 설정

- 에이전트 옵션은 다음과 같습니다.

```

# 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.

## 1 필터 영역

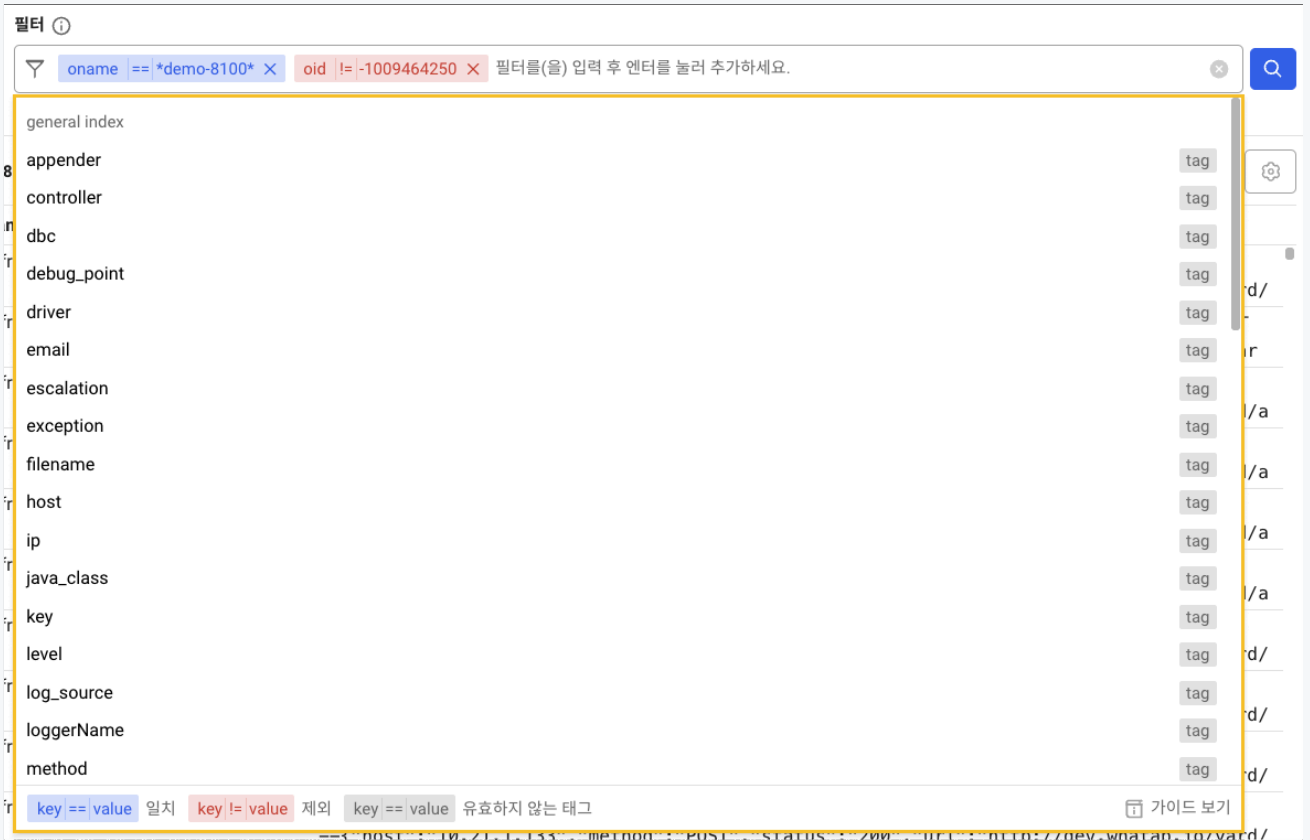
### 필터 적용

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 3 영역에서 조회할 수 있습니다.

① 가이드 UI

다음과 같이 입력 창 아래 가이드 UI를 제공합니다.



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.
- **검색 값** 입력 시 대소문자 구분 옵션을 활용해 검색할 수 있습니다.

① 필터 태그가 2줄 이상 길어지는 경우 ^ 접기 아이콘을 선택해 접어둘 수 있습니다.



## 필터 태그 추가

- 입력 창에 텍스트를 입력하고 키보드의 Enter, Tab키를 통해 추가할 수 있습니다.
- 입력 창 아래 가이드 UI에서 추천 값을 클릭하여 추가할 수 있습니다.
- 입력 창 아래 가이드 UI에서 키보드의 위아래 방향키로 추천 값을 선택할 수 있고 Enter, Tab키로 태그를 추가할 수 있습니다.

## 필터 태그 제거

- Backspace로 삭제할 수 있습니다.
- 태그의 X 아이콘 선택 시 태그를 삭제할 수 있습니다.
- 입력 창의 전체 삭제 X 아이콘 선택 시 전체 태그를 삭제할 수 있습니다.

## 필터 적용 예외 상황

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

- ① 라이브 테일 검색 키로 category를 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 설정을 통해 인덱스 키 값을 활용해 검색하는 것을 권장합니다.

필터 ⓘ

oid != -1009464250 × okind == -398596773 × content == \*select\* ×
필터를(을) 입력 후 엔터를 눌러 추가하세요.

oid != -1009464250 && okind == -398596773 && content == \*select\*

1. **카테고리**를 선택하세요.
2. **필터** 입력창에 content 기준 띄어쓰기 후 검색을 원하는 키워드를 입력하세요.
 

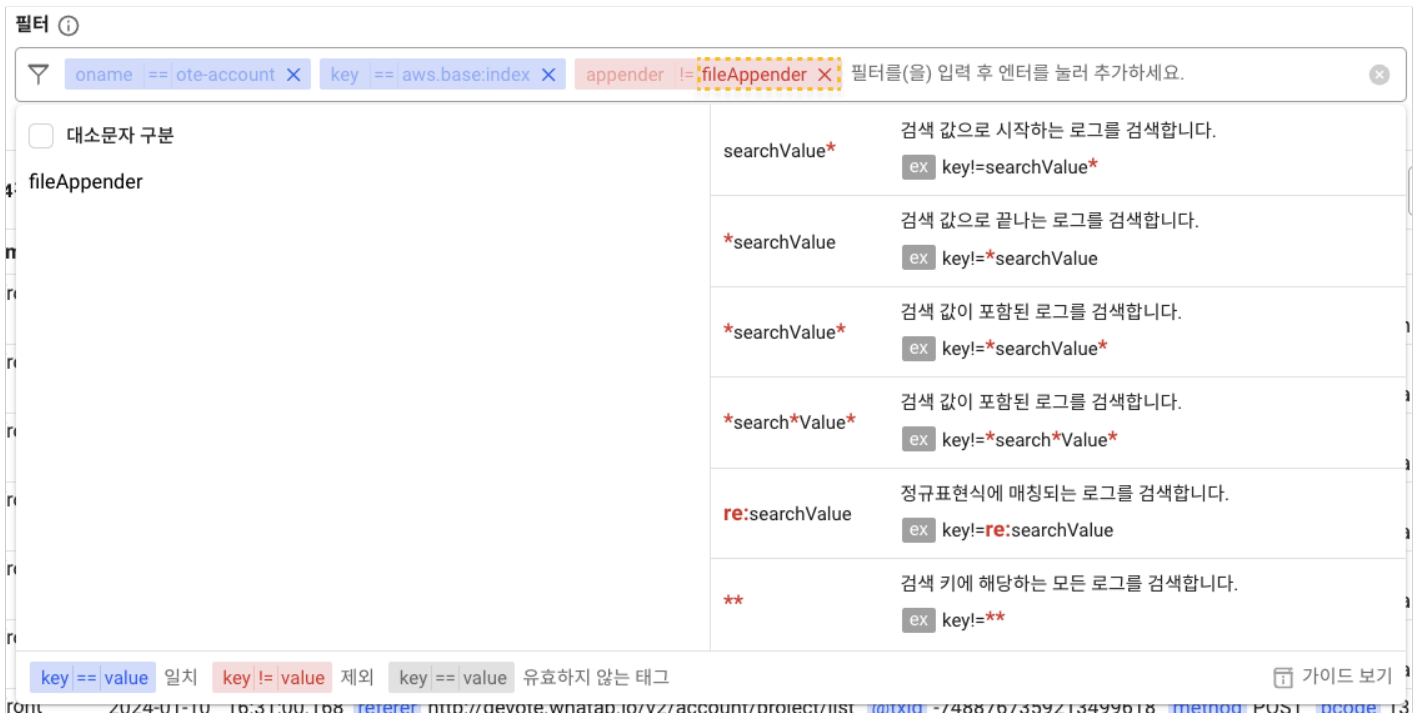
예시, content \*select\*

3. 🔍 검색 버튼을 클릭해 로그를 조회하세요.

- ① 라이브 테일의 경우 모든 로그 검색이 가능해 카테고리를 지정할 필요가 없습니다.
- 파서 설정에 대한 자세한 내용은 [다음 문서](#)를 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 UI를 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.

**date** 22/Aug/2022:03:16:47 +0000 **method** GET **ip** 116.32.201.189 **url** /MW/MyPage/mypageMain.tmall? 116.32.201.189 - - [22/Aug/2022:03:16:47 +0000] "GET /MW/MyPage/mypageMain.tmall?"

### 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.

▽ `exception == UnknownHostException` X

### 검색 키 종류

검색 키 종류	검색 키 포맷	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
<code>==searchValue</code>	검색 값과 일치하는 로그를 검색합니다.	<code>exception==RuntimeExceptionexception</code>
<code>!=searchValue</code>	검색 값을 제외한 로그를 검색합니다.	<code>exception!=RuntimeException</code>
<code>*searchValue</code>	검색 값으로 끝나는 로그를 검색합니다.	<code>word==*hello</code>
<code>searchValue*</code>	검색 값으로 시작하는 로그를 검색합니다.	<code>word==hello*</code>
<code>*searchValue*</code>	검색 값으로 중간에 포함된 로그를 검색합니다.	<code>word==*hello*</code>
<code>*search*Value*</code>	검색 값으로 포함된 로그를 검색합니다.	<code>word==*he*llo*</code>
<code>re:{regexpr}</code>	정규표현식에 매칭되는 로그를 검색합니다.	<code>caller==re:^i\.w\.a\.w\.s\.v\.r\.</code>
<code>**</code>	검색 키에 해당하는 모든 로그를 검색합니다.	

검색 키가 숫자 키워드(keyword.n)인 경우 문법

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

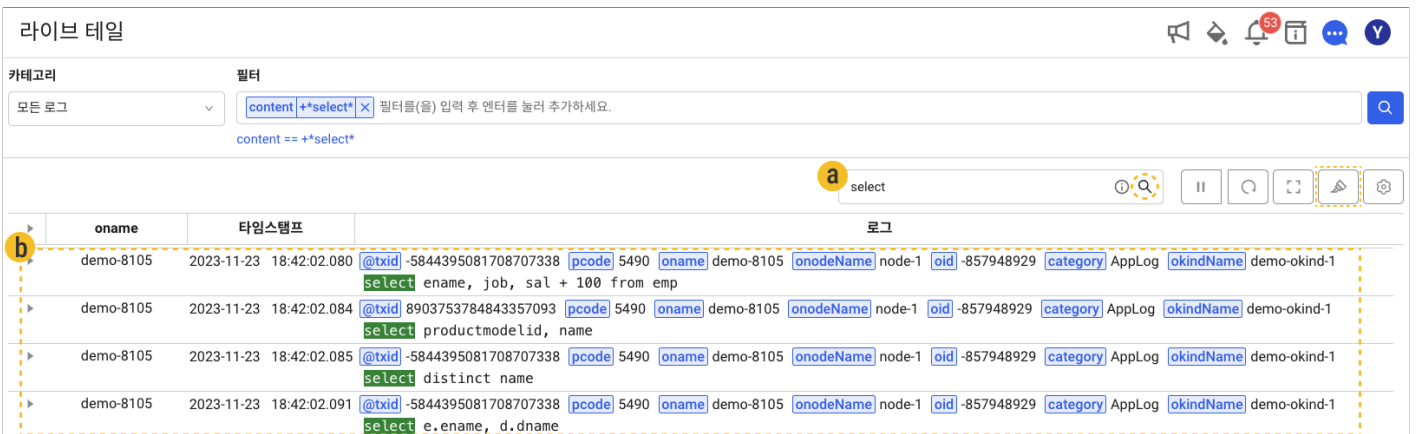
- 검색 값으로는 숫자만 올 수 있습니다.
- `.n` 키워드의 값에는 prefix를 붙이지 않습니다. `.n`이 아닌 키워드는 모두 prefix를 붙여야합니다.  
예, `+>searchValue`는 유효하지 않습니다.

문법 종류	설명	예시
<code>&gt;searchValue</code>	검색 값보다 큰 값이 포함된 로그를 조회합니다.	<code>response_time.n&gt;3000</code>
<code>&gt;=searchValue</code>	검색 값보다 크거나 같은 값이 포함된 로그를 조회합니다.	<code>response_time.n&gt;=3000</code>

문법 종류	설명	예시
==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

## 2 콘텐츠 하이라이트 영역

로그의 콘텐츠 중 원하는 키워드를 손쉽게 식별하기 위해 하이라이트 기능을 제공합니다.



- **a** 키워드 입력창에 하이라이트를 원하는 키워드를 입력 후 **Q** 검색 아이콘을 클릭하세요.

예시, `select`

- 예시 이미지와 같이 **b** 로그 목록에서 Content 내 키워드가 하이라이팅 됩니다.
- 단일 또는 복수 키워드로 필터를 걸 수 있습니다.
- **[ ]** 전체 화면 아이콘을 선택하면 **로그**와 **타임스탬프**를 전체 화면에서 확인할 수 있습니다.

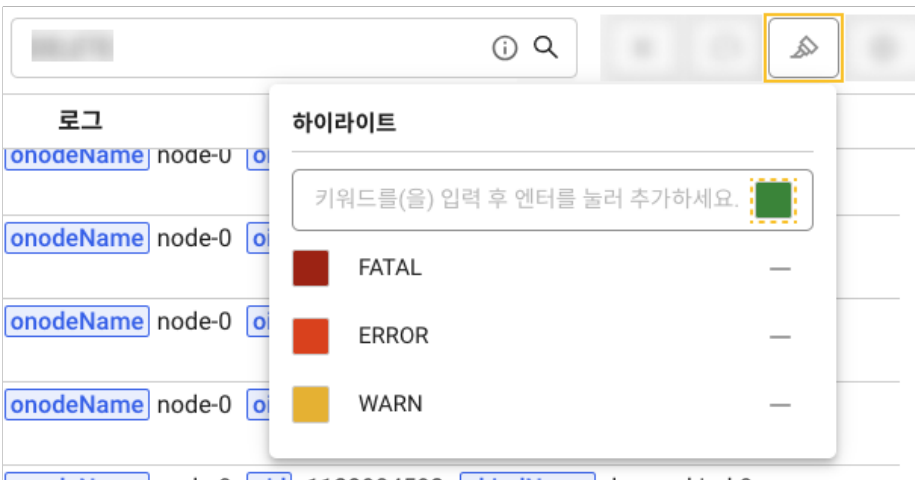
### 복수 키워드 조건

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 입력해야 합니다.	Whatap\ is good.

## 하이라이트 색상 설정

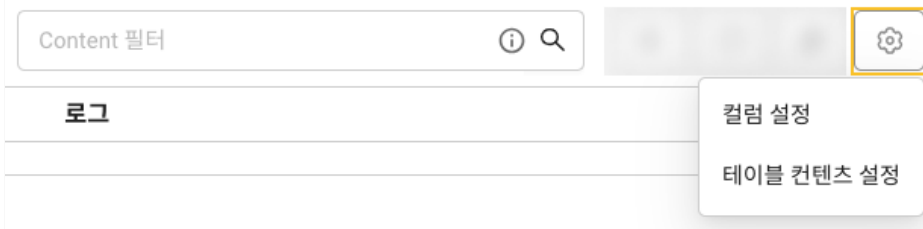
🔍 **하이라이트** 아이콘을 선택해 하이라이팅할 키워드 및 색상을 설정할 수 있습니다.



- 추가적으로 색상 설정을 원하는 키워드를 입력창에 입력하세요.
- 입력창 왼쪽 **색상** 클릭 시 선택할 수 있는 색상 메뉴가 나타납니다.
- 기본적으로 로그 레벨에 따른 하이라이팅(WARN, ERROR, FATAL)이 적용되어 있습니다.
- 설정한 내용은 **프로젝트 단위**로 저장됩니다.

## 테이블 설정

- ② 영역 오른쪽 **테이블 설정** 메뉴는 **라이브 테일**, **로그 검색**, **로그 트렌드**에서 사용할 수 있습니다.
- ⚙️ **테이블 설정** 버튼을 선택하면 **컬럼 추가**와 **테이블 콘텐츠 설정** 옵션 메뉴가 나타납니다.



1. 컬럼 설정

- **컬럼 추가:** 태그를 선택하여 테이블에 컬럼을 추가할 수 있습니다.
- **컬럼 순서 설정:** 컬럼을 추가하면 컬럼 순서 설정에 해당 컬럼이 추가됩니다. 원하는 컬럼을 드래그하여 컬럼의 순서를 변경하세요.

2. 테이블 설정



◦ **콘텐츠 표시 여부**

- 체크된 항목은 테이블에 표시되지 않습니다. 기본으로 **로그**, **태그** 모두 체크가 되어있으며 두 가지 항목 모두 표시합니다.
- 다음과 같이 **태그**를 해제할 경우 테이블에서 로그의 **태그**는 표시되지 않습니다.

```
@txid 2882146389875262493 pcode 5490 oname demo-8101 onodeName node-1 oid -877561626 okindName demo-okind-1
select distinct pp.lastname, pp.firstname
select distinct pp.lastname, pp.firstname
```

◦ **태그 관리**

- 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.

① 동일한 프로젝트 내 [라이브 테일](#), [로그 검색](#), [로그 트렌드](#) 메뉴는 테이블 설정을 공유합니다.



# 로그 트렌드

❗ 로그 조회 권한이 없을 경우 해당 메뉴에 진입할 수 없습니다.

홈 화면 > 프로젝트 선택 > 로그 > 로그 트렌드

로그 트렌드 메뉴에서 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 단위입니다.
- **Content:** 로그 메시지입니다.
- **Search Key:** 로그 파서 설정을 통해 생성합니다.
- **Tag:** 수집된 로그를 검색할 수 있는 검색 키입니다.

### 로그 트렌드


시간 < 2024/01/10 17:00 ~ 2024/01/10 18:00 60분 >

카테고리 AppLog

🔊 🔔 🗑️ 🗨️ 🔍

필터

- ▶ appender
- ▶ controller
- ▶ debug\_point
- ▶ email
- ▶ exception
- ▶ filename
- ▶ host
- ▶ ip
- ▶ java\_class
- ▶ key
- ▶ level
- ▶ log\_source
- ▶ loggerName
- ▶ method
- ▶ namespace
- ▶ oid
- ▶ okind
- ▶ okindName
- ▶ oname



2024-01-10 17:00:00 1min 전체 109,971건 조회

키워드(를) 입력해주세요 🔍

▶ oname	타임스탬프	로그
▶ ote-front	2024-01-10 17:00:00.433	referer http://devote.whatap.io/v2/project/database/878/db_dashboard pcode 13 oid 397157872 --{"host":"10.21.3.57","method":"POST","status":"200","url":"http://devote.whatap.io/y
▼ ote-front	2024-01-10 17:00:00.433	referer https://dev.whatap.io/v2/project/apm/13/transaction_map @txid -7828986607631991285 method GET pcode 13 level INFO oid 397157872 status_nxx status_2xx okind -398596773 threadName XNIO-1 task-4 url http://dev.whatap.io/yard/api responseTime.n 10 oname ote-front agentime 1704873600202 host 10.21.3.57 status.success.n 1 @mtid 2306616107552506289 category AppLog loggerName ACCESS okindName group-front email sa@whatap.io status 200 --{"host":"10.21.3.57","method":"GET","status":"200","url":"http://dev.whatap.io/yard/ap i","referer":"https://dev.whatap.io/v2/project/apm/13/transaction_map","email":"sa@whata p.io","responseTime.n":10}--
▶ ote-front	2024-01-10 17:00:00.433	referer https://dev.whatap.io/v2/project/apm/2216/dashboard pcode 13 oid 397157872 okind -398596773 --{"host":"10.21.1.133","method":"POST","status":"200","url":"http://dev.whatap.io/yard/
▶ ote-front	2024-01-10 17:00:00.433	referer https://dev.whatap.io/v2/project/apm/2216/dashboard @txid -7177366797654501422 method POST --{"host":"10.21.1.133","method":"POST","status":"200","url":"http://dev.whatap.io/yard/
▶ ote-front	2024-01-10 17:00:00.433	referer https://dev.whatap.io/v2/project/apm/2216/dashboard pcode 13 oid 397157872 okind -398596773 --{"host":"10.21.1.133","method":"POST","status":"200","url":"http://dev.whatap.io/yard/
▶ ote-front	2024-01-10 17:00:00.433	referer https://dev.whatap.io/v2/project/apm/2216/dashboard @txid -3688468993623540252 method POST --{"host":"10.21.1.133","method":"POST","status":"200","url":"http://dev.whatap.io/yard/
▶ ote-front	2024-01-10 17:00:00.434	referer https://dev.whatap.io/v2/project/apm/8/transaction_map pcode 13 oid 397157872 okind -398596773

# 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 **타임스탬프**를 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.

▶	2023-12-18 14:31:02.563	[level]	INFO	[pcode]	2277	[agenttime]	1702877462042	[oid]	778873916	[category]	AppLog	[loggerName]	io.home.test.logback02starter.base.web.LogbackControllerGreeting
			INFO										log in our greeting method.
▶	2023-12-18 14:31:02.563	[level]	WARN	[pcode]	2277	[agenttime]	1702877462043	[oid]	778873916	[category]	AppLog	[loggerName]	io.home.test.logback02starter.base.web.LogbackControllerError
			WARN										io.home.test.logback02starter.base.errors.exception.ApiException: [2204] Process failure. Please try again.
▶	2023-12-18 14:31:02.586	[level]	error	[pcode]	2277	[agenttime]	1702877462043	[oid]	778873916	[category]	AppServer	[_event_status_]	error
			error									[_event_status_literal_name_]	level
													Servlet.service() for servlet [dispatcherServlet] in context with path [] threw exception [Request processing failed: io.home.test.logback02starter.base.err
▶	2023-12-18 14:31:02.586	[level]	error	[pcode]	2277	[agenttime]	1702877462057	[oid]	778873916	[category]	AppServer	[_event_status_]	error
			error									[_event_status_literal_name_]	level
													Servlet.service() for servlet [dispatcherServlet] in context with path [] threw exception [Request processing failed: io.home.test.logback02starter.base.err
▶	2023-12-18 14:31:02.586	[level]	error	[pcode]	2277	[agenttime]	1702877462081	[oid]	778873916	[category]	AppServer	[_event_status_]	error
			error									[_event_status_literal_name_]	level
													Servlet.service() for servlet [dispatcherServlet] in context with path [] threw exception [Request processing failed: io.home.test.logback02starter.base.err
▶	2023-12-18 14:31:02.586	[level]	error	[pcode]	2277	[agenttime]	1702877462087	[oid]	778873916	[category]	AppServer	[_event_status_]	error
			error									[_event_status_literal_name_]	level
													Servlet.service() for servlet [dispatcherServlet] in context with path [] threw exception [Request processing failed: io.home.test.logback02starter.base.err

## ! 에이전트 옵션 설정

- 에이전트 옵션은 다음과 같습니다.

```
# 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**, **INFO**를 포함할 경우 로그 레벨 색상을 표시합니다.

## 로그 Content 확인하기

### ! Content란?

Content는 로그 메시지를 의미합니다.

- 로그 컬럼의 첫 번째 줄은 로그의 파싱(parsing)된 키와 값이고 두 번째 줄은 로그의 Content입니다.
- 3 로그 테이블의 행(로그)마다 ▶ **더보기** 버튼이 있습니다. ▶ **더보기** 버튼을 선택하면 5처럼 해당 로그의 전체 content를 확인할 수 있습니다.

## 차트로 로그 조회하기

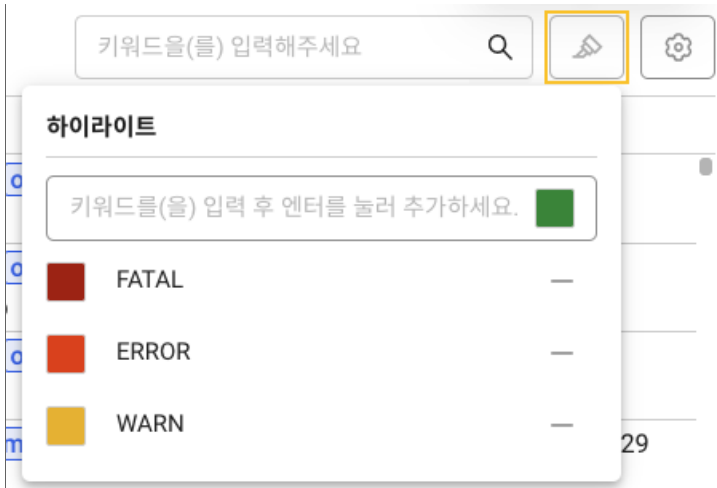


- 바 차트에서 a 원하는 시간을 클릭하여 해당 시간의 로그를 확인할 수 있습니다.
- 바 차트 아래 로그 테이블 상단 왼쪽의 b 시간 선택 옵션을 이용해 다음과 같이 선택한 시간대에서 더 세분화해 로그를 검색할 수 있습니다.

- 1min: interval (차트의 막대 사이 간격)
- 시간 선택 옵션: 선택된 시간 범위를 6개의 구간으로 나눈 시간대

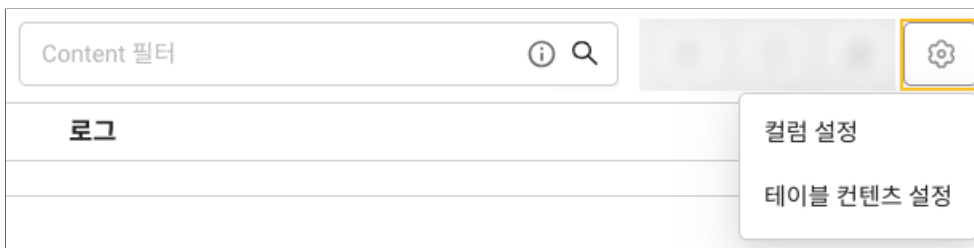
## 하이라이트

- 원하는 키워드를 손쉽게 식별할 수 있도록 하이라이트 기능을 제공합니다.
- **3** 영역 오른쪽 **입력창**에 하이라이트 할 키워드를 입력하세요. 다음과 같이 하이라이팅 된 키워드를 확인할 수 있습니다.



## 테이블 설정하기


- **3** 영역 오른쪽 **테이블 설정** 메뉴는 **라이브 테일**, **로그 검색**, **로그 트렌드**에서 사용할 수 있습니다.
- **테이블 설정** 버튼을 선택하면 **컬럼 추가**와 **테이블 콘텐츠 설정** 옵션 메뉴가 나타납니다.



### 1. 컬럼 설정

- **컬럼 추가**: 태그를 선택하여 테이블에 컬럼을 추가할 수 있습니다.
- **컬럼 순서 설정**: 컬럼을 추가하면 컬럼 순서 설정에 해당 컬럼이 추가됩니다. 원하는 컬럼을 드래그하여 컬럼의 순서를 변경하세요.

## 2. 테이블 설정



테이블 콘텐츠 설정

콘텐츠 표시 여부

로그  태그

태그 관리

추가할 태그를 입력 후 엔터를 치면 태그가 추가됩니다.

취소 확인

### ◦ 콘텐츠 표시 여부

- 체크된 항목은 테이블에 표시되지 않습니다. 기본으로 **로그**, **태그** 모두 체크가 되어있으며 두 가지 항목 모두 표시합니다.
- 다음과 같이 **태그**를 해제할 경우 테이블에서 로그의 **태그**는 표시되지 않습니다.

```
@txid 2882146389875262493 pcode 5490 oname demo-8101 onodeName node-1 oid -877561626 okindName demo-okind-1
select distinct pp.lastname, pp.firstname
```

↓

```
select distinct pp.lastname, pp.firstname
```

### ◦ 태그 관리

- 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.

① 동일한 프로젝트 내 **라이브 테일**, **로그 검색**, **로그 트렌드** 메뉴는 테이블 설정을 공유합니다.

# 로그 검색

① 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 단위입니다.
- **Content**: 로그 메시지입니다.
- **Search Key**: 로그 파서 설정을 통해 생성합니다.
- **Tag**: 수집된 로그를 검색할 수 있는 검색 키입니다.

**1** 로그 검색

시간: < 2023/07/03 13:16 ~ 2023/07/03 14:16 60분 > 필터 (클) 입력 후 엔터를 눌러 추가하세요.

**2** [AppLog, AppStdErr, AppStdOut, VirtualLog] (4,490,257, 353, 19,439, 20,767)

**3** 전체 4,532,168건 조회 Timestamp 과거 순 최근 순 키워드(클) 입력해주세요

**4**

oname	타임스탬프	로그
dev3679006-8091	2023-07-03 13:16:00.000	@@txid -2717737560424755676 pcode 8 oname dev3679006-8091 onodeName node-1 oid 777628269 category AppLog insert into emp values
dev3679007-8092	2023-07-03 13:16:00.001	@@txid -4707233158811724771 pcode 8 oname dev3679007-8092 onodeName node-0 oid 2081171570 category AppLog select ename, deptno, sal + comm from emp
dev3679006-8091	2023-07-03 13:16:00.002	@@txid -7559227993102384977 pcode 8 oname dev3679006-8091 onodeName node-1 oid 777628269 category AppLog okindName dev-okind-1 okind 867318026 onode 334634079 select distinct pp.lastname, pp.firstname from person.person pp join humanresources.employee e on e
dev3679007-8092	2023-07-03 13:16:00.002	@@txid 5627186231377631605 pcode 8 oname dev3679007-8092 onodeName node-0 oid 2081171570 category AppLog http://127.0.0.1:8092/remote/order/kill/employee/pusan status=200
dev3679011-8095	2023-07-03 13:16:00.002	@@txid -6614897519727834472 pcode 8 oname dev3679011-8095 onodeName node-1 oid -1840884360 category AppLog select distinct pp.lastname, pp.firstname
dev3679007-8092	2023-07-03 13:16:00.004	@@txid -959317925843459419 pcode 8 oname dev3679007-8092 onodeName node-0 oid 2081171570 category AppLog select ename, 1000 from emp
dev3679007-8092	2023-07-03 13:16:00.005	@@txid 5627186231377631605 pcode 8 oname dev3679007-8092 onodeName node-0 oid 2081171570 category AppLog okindName dev-okind-0 okind 1152715164 onode 1693789385 update table set x=1 where key=1 elapsed=2ms
dev3679007-8092	2023-07-03 13:16:00.006	@@txid -4707233158811724771 pcode 8 oname dev3679007-8092 onodeName node-0 oid 2081171570 category AppLog select ename, deptno, sal, job from emp
dev3679007-8092	2023-07-03 13:16:00.007	@@txid 5627186231377631605 pcode 8 oname dev3679007-8092 onodeName node-0 oid 2081171570 category AppLog delete from posts
dev3679006-8091	2023-07-03 13:16:00.008	@@txid -7559227993102384977 pcode 8 oname dev3679006-8091 onodeName node-1 oid 777628269 category AppLog select
dev3679011-8095	2023-07-03 13:16:00.008	@@txid -6614897519727834472 pcode 8 oname dev3679011-8095 onodeName node-1 oid -1840884360 category AppLog select e.ename, d.dname
dev3679006-8091	2023-07-03 13:16:00.013	@@txid -7559227993102384977 pcode 8 oname dev3679006-8091 onodeName node-1 oid 777628269 category AppLog select
dev3679011-8095	2023-07-03 13:16:00.013	@@txid -6614897519727834472 pcode 8 oname dev3679011-8095 onodeName node-1 oid -1840884360 category AppLog select productmodelid, name
dev3679007-8092	2023-07-03 13:16:00.016	@@txid 8725913945553755591 pcode 8 oname dev3679007-8092 onodeName node-0 oid 2081171570 category AppLog http://127.0.0.1:8092/remote/edu/save/dept/daejun status=200
dev3679007-8092	2023-07-03 13:16:00.017	@@txid 5479838888404626132 pcode 8 oname dev3679007-8092 onodeName node-0 oid 2081171570 category AppLog select distinct pp.lastname, pp.firstname
dev3679007-8092	2023-07-03 13:16:00.018	@@txid -6044091401645216416 pcode 8 oname dev3679007-8092 onodeName node-0 oid 2081171570 category AppLog http://127.0.0.1:8092/remote/account/create/unit/jeju status=200
dev3679006-8091	2023-07-03 13:16:00.019	@@txid -7559227993102384977 pcode 8 oname dev3679006-8091 onodeName node-1 oid 777628269 category AppLog select * from emp
dev3679011-8095	2023-07-03 13:16:00.019	@@txid -6614897519727834472 pcode 8 oname dev3679011-8095 onodeName node-1 oid -1840884360 category AppLog select productmodelid, name

## 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 **Timestamp** 과거 순과 최근 순 중 원하는 조회 방식을 선택하세요.
- 시간 범위 지정 후 적용 버튼을 선택 해 조회 시간을 설정하고 **Q** 검색 버튼을 선택해 데이터를 조회합니다.
- **3** 로그 테이블 상단 오른쪽 **[ ] 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.

2023-12-18 14:31:02.563	[level] INFO [pcode] 2277 [agenttime] 1702877462042 [oid] 778873916 [category] AppLog [loggerName] io.home.test.logback02starter.base.web.LogbackControllerGreeting
	INFO Log in our greeting method.
2023-12-18 14:31:02.563	[level] WARN [pcode] 2277 [agenttime] 1702877462043 [oid] 778873916 [category] AppLog [loggerName] io.home.test.logback02starter.base.web.LogbackControllerError [threadName] http-nio-19090-exec-2
	io.home.test.logback02starter.base.errors.exception.ApiException: [2204] Process failure. Please try again.
2023-12-18 14:31:02.586	[level] error [pcode] 2277 [agenttime] 1702877462043 [oid] 778873916 [category] AppServer [event_status] error [event_status_literal_name] level
	Servlet.service() for servlet [dispatcherServlet] in context with path [] threw exception [Request processing failed: io.home.test.logback02starter.base.err
2023-12-18 14:31:02.586	[level] error [pcode] 2277 [agenttime] 1702877462057 [oid] 778873916 [category] AppServer [event_status] error [event_status_literal_name] level
	Servlet.service() for servlet [dispatcherServlet] in context with path [] threw exception [Request processing failed: io.home.test.logback02starter.base.err
2023-12-18 14:31:02.586	[level] error [pcode] 2277 [agenttime] 1702877462081 [oid] 778873916 [category] AppServer [event_status] error [event_status_literal_name] level
	Servlet.service() for servlet [dispatcherServlet] in context with path [] threw exception [Request processing failed: io.home.test.logback02starter.base.err
2023-12-18 14:31:02.586	[level] error [pcode] 2277 [agenttime] 1702877462087 [oid] 778873916 [category] AppServer [event_status] error [event_status_literal_name] level
	Servlet.service() for servlet [dispatcherServlet] in context with path [] threw exception [Request processing failed: io.home.test.logback02starter.base.err

### ① 에이전트 옵션 설정

- 에이전트 옵션은 다음과 같습니다.

```
# 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 Content 확인하기

### ① Content란?

Content는 로그 메시지를 의미합니다.

- 로그 컬럼의 첫 번째 줄은 로그의 파싱(parsing)된 키와 값이고 두 번째 줄은 로그의 Content입니다.
- ③ 로그 테이블의 행(로그)마다 ▶ [더보기](#) 버튼이 있습니다. ▶ [더보기](#) 버튼을 선택하면 ④ 처럼 해당 로그의 전체 Content를 확인할 수 있습니다.
- 로그의 태그를 선택하면 복사, 검색, 제외 검색, 인접 로그 검색을 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 **3** 영역에서 조회할 수 있습니다.

### 가이드 UI

다음과 같이 입력 창 아래 **가이드 UI**를 제공합니다.

The screenshot shows the filter input area with the following content:

필터 ⓘ

oname == \*demo-8100\* × oid != -1009464250 × 필터를(을) 입력 후 엔터를 눌러 추가하세요.

general index

8 appender	tag
controller	tag
dbc	tag
debug_point	tag
driver	tag
email	tag
escalation	tag
exception	tag
filename	tag
host	tag
ip	tag
java_class	tag
key	tag
level	tag
log_source	tag
loggerName	tag
method	tag

key == value 일치 key != value 제외 key == value 유효하지 않는 태그

가이드 보기

## 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 **파란색**으로, 제외 검색(`!=`)일 때 **붉은색**으로 하이라이팅합니다.
- **검색 값** 입력 시 대소문자 구분 옵션을 활용해 검색할 수 있습니다.

① 필터 태그가 2줄 이상 길어지는 경우 **접기** 아이콘을 선택해 접어들 수 있습니다.

## 필터 태그 추가

- 입력 창에 텍스트를 입력하고 키보드의 Enter, Tab키를 통해 추가할 수 있습니다.
- 입력 창 아래 **가이드 UI**에서 추천 값을 클릭하여 추가할 수 있습니다.
- 입력 창 아래 **가이드 UI**에서 키보드의 위아래 방향키로 추천 값을 선택할 수 있고 Enter, Tab키로 태그를 추가할 수 있습니다.

## 필터 태그 제거

- Backspace로 삭제할 수 있습니다.
- 태그의 **X** 아이콘 선택 시 태그를 삭제할 수 있습니다.
- 입력 창의 전체 삭제 **X** 아이콘 선택 시 전체 태그를 삭제할 수 있습니다.

## 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

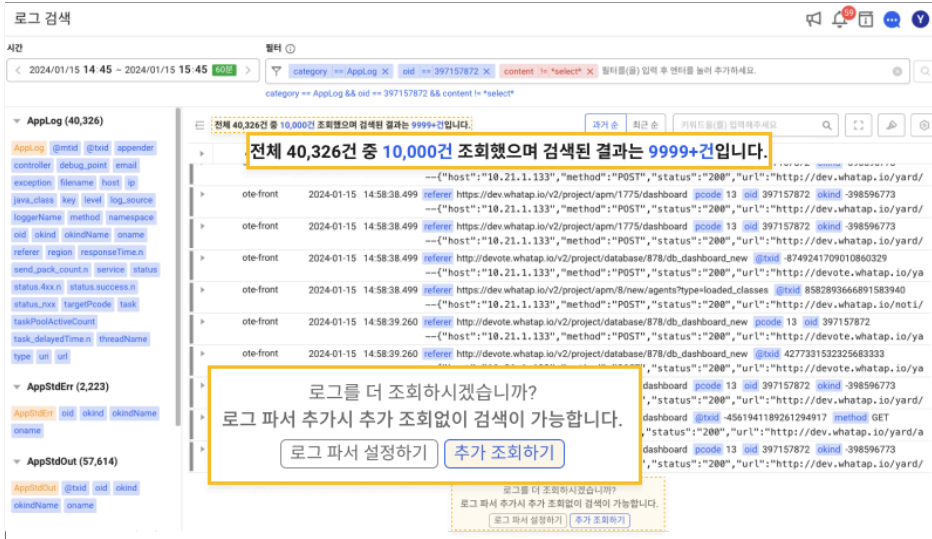
로그에서 파싱되지 않은 즉 인덱스가 생성되지 않은 키워드를 포함한 로그를 조회할 수 있습니다. 이 경우 지정 범위 내 모든 로그를 Full Scan합니다. 그렇기 때문에 인덱스가 생성된 키와 비교해 검색 속도가 다소 떨어질 수 있습니다. 정형화된 로그 데이터의 경우 **로그 파서 설정**을 통해 인덱스 키 값을 활용해 검색하는 것을 권장합니다.

필터 ⓘ

필터를(을) 입력 후 엔터를 눌러 추가하세요.

oid != -1009464250 && okind == -398596773 && content == \*select\*

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.



- ① 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 문서](#)를 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 UI를 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 설정의 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.

exception == UnknownHostException X

## 검색 키 종류

검색 키 종류	검색 키 포맷	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:{regexpr}	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
<code>&gt;searchValue</code>	검색 값보다 큰 값이 포함된 로그를 조회합니다.	<code>response_time.n&gt;3000</code>
<code>&gt;=searchValue</code>	검색 값보다 크거나 같은 값이 포함된 로그를 조회합니다.	<code>response_time.n&gt;=3000</code>
<code>==searchValue</code>	검색 값보다 같은 값이 포함된 로그를 조회합니다.	<code>response_time.n==3000</code>
<code>!=searchValue</code>	검색 값보다 다른 값이 포함된 로그를 조회합니다.	<code>response_time.n!=3000</code>
<code>&lt;searchValue</code>	검색 값보다 작은 값이 포함된 로그를 조회합니다.	<code>response_time.n&lt;3000</code>
<code>&lt;=searchValue</code>	검색 값보다 작거나 같은 값이 포함된 로그를 조회합니다.	<code>response_time.n&lt;=3000</code>

## 로그 태그 옵션

로그 태그 선택 시 다음과 같이 드롭다운 메뉴가 나타납니다. [검색](#), [제외 검색](#), [인접 로그](#) 옵션을 확인할 수 있습니다.

▶ 2023-09-07 15:06:00.002	@txid -3761935652943885420	pcode 5490	oname demo-8103	onodeName node-1	oid 633280970	category AppLog
▶ 2023-09-07 15:06:00.002	@txid -8799764694958745204	pcode 5490	select distinct ename, deptno, sal,	onodeName node-1	oid 633280970	category AppLog
▶ 2023-09-07 15:06:00.003	@txid 5498362167526616791	pcode 5490	select	onodeName node-0	oid 1387800924	category AppLog
▶ 2023-09-07	@txid 3613588890639779125	pcode 5490	select	onodeName node-0	oid 1387800924	category AppLog

- 복사
- 검색
- 제외 검색
- 인접 로그

• 검색

검색 옵션을 선택하면 필터에 해당 태그가 포함('==') 조건으로 입력됩니다.

• 제외 검색

제외 검색 옵션을 선택하면 필터에 해당 태그가 제외('!=') 조건으로 입력됩니다.

• 인접 로그

인접 로그 옵션을 선택하면 인접 로그 상세 창이 나타납니다. 선택한 로그의 서버를 대상으로 선택한 로그와 인접한 시간대의 로그를 조회합니다. 시간 선택 버튼을 클릭해 인접한 시간대의 로그를 조회할 수 있습니다. 기준 로그는 파란색 바탕으로 표시됩니다.

인접 로그
✕

**시간 선택**

밀리초   초    분

**적용된 서버**

demo-8101

	타임스탬프	로그
▶	2023-09-07 15:24:59.995	<a href="#">@txid</a> -8789311500480722351 <a href="#">pcode</a> 5490 <a href="#">oname</a> demo-8101 <a href="#">onodeName</a> node-1 <a href="#">oid</a> -877561626 <a href="#">category</a> AppLog select distinct ename, deptno, sal, job from emp
▶	2023-09-07 15:24:59.998	<a href="#">@txid</a> 3991620568819529441 <a href="#">pcode</a> 5490 <a href="#">oname</a> demo-8101 <a href="#">onodeName</a> node-1 <a href="#">oid</a> -877561626 <a href="#">category</a> AppLog select distinct pp.lastname, pp.firstname
▶	2023-09-07 15:25:00.001	<a href="#">@txid</a> -8789311500480722351 <a href="#">pcode</a> 5490 <a href="#">oname</a> demo-8101 <a href="#">onodeName</a> node-1 <a href="#">oid</a> -877561626 <a href="#">category</a> AppLog select quantity, max(price)
▶	2023-09-07 15:25:00.007	<a href="#">@txid</a> 3991620568819529441 <a href="#">pcode</a> 5490 <a href="#">oname</a> demo-8101 <a href="#">onodeName</a> node-1 <a href="#">oid</a> -877561626 <a href="#">category</a> AppLog select ename, sal=sal+1000 from emp
▶	2023-09-07 15:25:00.007	<a href="#">@txid</a> -8789311500480722351 <a href="#">pcode</a> 5490 <a href="#">oname</a> demo-8101 <a href="#">onodeName</a> node-1 <a href="#">oid</a> -877561626 <a href="#">category</a> AppLog select
▶	2023-09-07 15:25:00.014	<a href="#">@txid</a> -8789311500480722351 <a href="#">pcode</a> 5490 <a href="#">oname</a> demo-8101 <a href="#">onodeName</a> node-1 <a href="#">oid</a> -877561626 <a href="#">category</a> AppLog select
▶	2023-09-07 15:25:00.022	<a href="#">@txid</a> 3991620568819529441 <a href="#">pcode</a> 5490 <a href="#">oname</a> demo-8101 <a href="#">onodeName</a> node-1 <a href="#">oid</a> -877561626 <a href="#">category</a> AppLog select ename, deptno, sal + comm from emp
▶	2023-09-07 15:25:00.022	<a href="#">@txid</a> -8789311500480722351 <a href="#">pcode</a> 5490 <a href="#">oname</a> demo-8101 <a href="#">onodeName</a> node-1 <a href="#">oid</a> -877561626 <a href="#">category</a> AppLog select
▶	2023-09-07 15:25:00.022	<a href="#">@txid</a> 3823483774615642623 <a href="#">pcode</a> 5490 <a href="#">oname</a> demo-8101 <a href="#">onodeName</a> node-1 <a href="#">oid</a> -877561626 <a href="#">category</a> AppLog select productid
▶	2023-09-07 15:25:00.029	<a href="#">@txid</a> 3823483774615642623 <a href="#">pcode</a> 5490 <a href="#">oname</a> demo-8101 <a href="#">onodeName</a> node-1 <a href="#">oid</a> -877561626 <a href="#">category</a> AppLog select

닫기

## 콘텐츠 하이라이트

로그의 콘텐츠 중 원하는 키워드를 손쉽게 식별하기 위해 하이라이트 기능을 제공합니다.



The screenshot shows the log search interface. At the top, there's a search bar with filters: `category == AppLog` and `content == *select*`. Below the search bar, there are three sections: **AppLog (107,548)**, **AppStdErr (2,016)**, and **AppStdOut (57,562)**. The main area displays a table of search results. One result is highlighted with a yellow dashed box and labeled 'b'. This result shows a log entry for a POST request to `/yard/api/flush` with a status of 200. The log content is a JSON object with various fields like `responseTime`, `controller`, `uri`, `method`, `pcode`, `level`, `oid`, `status_nxx`, `status_2xx`, `okind`, `uri`, `threadName`, `responseTime`, `oname`, `agenttime`, `status_success_n`, `category`, `loggerName`, `okindName`, and `status`. A search icon labeled 'a' is visible in the top right of the search bar area.

- **a** 키워드 입력창에 하이라이트를 원하는 키워드를 입력 후 **Q 검색** 아이콘을 클릭하세요.  
예시, `select`
- 예시 이미지와 같이 **b** 로그 목록에서 Content 내 키워드가 하이라이팅 됩니다.
- 단일 또는 복수 키워드로 필터를 걸 수 있습니다.
- **[ ] 전체 화면** 아이콘을 선택하면 **로그**와 **타임스탬프**를 전체 화면에서 확인할 수 있습니다.

## 복수 키워드 조건

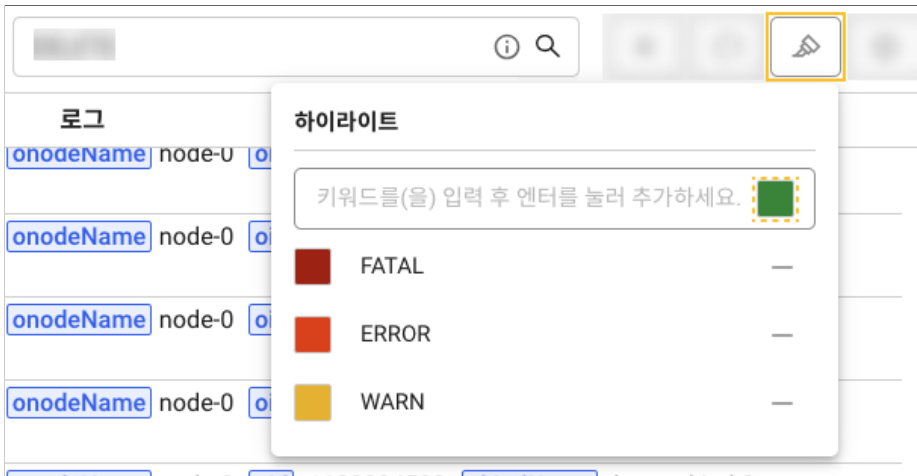
복수 키워드로 하이라이팅을 할 경우 다음과 같이 작성합니다.

입력 문자열	설명	결과
a b c	띄어쓰기로 각 키워드를 구분합니다.	a, b, c

입력 문자열	설명	결과
"Whatap is good."	띄어쓰기를 키워드에 포함하고 싶은 경우 ' ' 또는 ""로 감쌉니다.	Whatap is good.
"Whatap\\ is good."	""로 감싸진 키워드에서 \를 포함할 경우, \\로 입력해야 합니다.	Whatap\ is good.

## 하이라이트 색상 설정

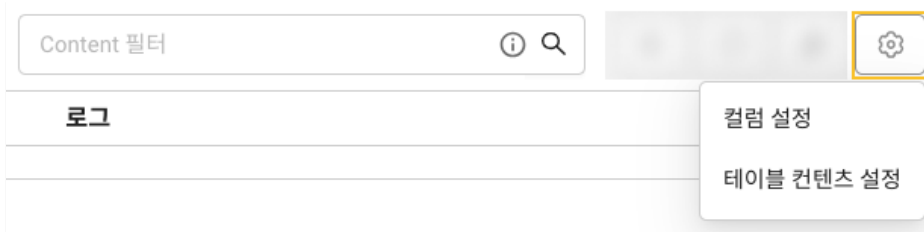
👉 **하이라이트** 아이콘을 선택해 하이라이팅할 키워드 및 색상을 설정할 수 있습니다.



- 추가적으로 색상 설정을 원하는 키워드를 입력창에 입력하세요.
- 입력창 왼쪽 **색상** 클릭 시 선택할 수 있는 색상 메뉴가 나타납니다.
- 기본적으로 로그 레벨에 따른 하이라이팅(WARN, ERROR, FATAL)이 적용되어 있습니다.
- 설정한 내용은 **프로젝트** 단위로 저장됩니다.

## 테이블 설정하기

- ③ 영역 오른쪽 **테이블 설정** 메뉴는 **라이브 테일**, **로그 검색**, **로그 트렌드**에서 사용할 수 있습니다.
- ⚙ **테이블 설정** 버튼을 선택하면 **컬럼 추가**와 **테이블 콘텐츠 설정** 옵션 메뉴가 나타납니다.



1. 컬럼 설정

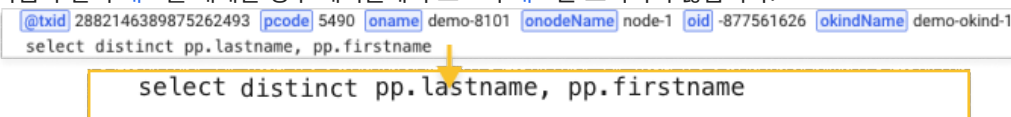
- **컬럼 추가:** 태그를 선택하여 테이블에 컬럼을 추가할 수 있습니다.
- **컬럼 순서 설정:** 컬럼을 추가하면 컬럼 순서 설정에 해당 컬럼이 추가됩니다. 원하는 컬럼을 드래그하여 컬럼의 순서를 변경하세요.

2. 테이블 설정



◦ 콘텐츠 표시 여부

- 체크된 항목은 테이블에 표시되지 않습니다. 기본으로 **로그**, **태그** 모두 체크가 되어있으며 두 가지 항목 모두 표시합니다.
- 다음과 같이 **태그**를 해제할 경우 테이블에서 로그의 **태그**는 표시되지 않습니다.



◦ 태그 관리

- 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.

① 동일한 프로젝트 내 [라이브 테일](#), [로그 검색](#), [로그 트렌드](#) 메뉴는 테이블 설정을 공유합니다.

# 알림 설정하기

홈 화면 > 프로젝트 선택 > 경고 알림 > 이벤트 설정 > 로그 탭

수집한 로그 데이터를 조건에 맞춰 필터링해 경고 알림을 설정할 수 있습니다. + 이벤트 추가 버튼을 선택해 로그 이벤트 경고 알림을 설정하세요. 모든 설정을 완료한 다음 저장 버튼을 선택하세요.

### 이벤트 설정

실시간 로그 이벤트
JSON
+ 이벤트 추가

이벤트 이름	카테고리	검색 키	검색 값	이벤트 발생 일시 중지	이벤트 대상 필터링	이벤트 수신 태그	설정
실시간 로그 이벤트	AppLog	onodeName	node-0	20분	onode == '334634079' && okindName != 'demo-okind-1'	test-tag	🔍 <input type="checkbox"/>
이벤트 테스트	AppLog	oid	-1128904592	사용 안 함	사용 안 함	전체 멤버 수신	🔍 <input checked="" type="checkbox"/>
test	AppLog	oname	demo-8100	사용 안 함	oname && 'demo-8100'	전체 멤버 수신	🔍 <input checked="" type="checkbox"/>

복합 로그 이벤트
JSON
+ 이벤트 추가

이벤트 이름	템플릿	카테고리	규칙	이벤트 상태가 해결되면 추가 알림	이벤트 발생 일시 중지	인터벌	이벤트 수신 태그	설정
이벤트 이름_테스트	2xx 상태코드 건수 count	AppLog	include_minus_exclude_count > 10	Off	1분	5분	전체 멤버 수신	🔍 <input checked="" type="checkbox"/>
retreter	사용 안 함	#WhatapEvent	rows > 10	Off	1분	5분	전체 멤버 수신	🔍 <input checked="" type="checkbox"/>
test	사용 안 함	*	rows > 10	Off	1분	5분	전체 멤버 수신	🔍 <input checked="" type="checkbox"/>

추가할 수 있는 로그 이벤트 다음과 같습니다.

- **실시간 로그 이벤트** : 실시간으로 수집한 로그에서 검색 값이 등장하면 경고 알림을 보냅니다.
- **복합 로그 이벤트** : 최근에 수집한 로그 중 일정 조건을 만족하는 로그가 일정 개수 이상 수집한 경우에 경고 알림을 보냅니다.

- ⓘ • 이벤트를 추가하거나 설정하려면 **알림 설정** 권한이 있어야 합니다. 사용자별 권한에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.
  - 경고 알림과 관련해 모니터링 플랫폼별 지원되는 이벤트 종류를 확인하려면 [다음 문서](#)를 참조하세요.

## 이벤트 추가 공통 옵션

다음은 이벤트 추가 시 공통으로 설정할 수 있는 옵션입니다.

- **이벤트 이름** : 추가하려는 이벤트 이름을 입력하세요.
- **이벤트 활성화** : 토글 버튼을 클릭해 경고 알림 활성화 여부를 선택할 수 있습니다.
- **레벨** : **위험**, **경고**, 정보 중 하나의 레벨을 선택하세요.
- **메시지** : 이벤트 발생 시 출력하는 알림 메시지를 입력합니다. `#{태그 또는 필드키}` 입력으로 메시지에 변수를 적용할 수 있습니다. 변수에 입력할 키는 선택한 매트릭스 데이터 **카테고리**에 포함된 값이어야 합니다.
- **카테고리** : 로그 구분 명칭(로그 폴더명)을 목록에서 선택하거나 직접 입력할 수 있습니다.
- **이벤트 발생 일시 중지** : 과도한 경고 알림 발생을 방지할 수 있는 옵션입니다. 첫 번째 경고 알림 이후 선택한 시간 동안 경고 알림을 보내지 않습니다. 또한 **이벤트 기록** 메뉴에 기록되지 않습니다.
- **이벤트 수신 태그** : 이벤트 수신 태그를 선택하면 해당 태그를 가진 프로젝트 멤버와 3rd-party 플러그인에 알림을 전송할 수 있습니다. 이벤트 수신 태그를 선택하지 않으면 프로젝트 전체 멤버에게 경고 알림을 보냅니다.

태그를 추가하지 않으면 전체 멤버에게 경고 알림을 보냅니다. **+ 태그 추가**를 클릭한 다음 **태그 목록**에서 경고 알림 수신 대상을 선택하세요. **+ 새 태그 생성**을 선택해 태그를 추가할 수도 있습니다.

ⓘ 경고 알림 > 이벤트 수신 설정 메뉴에서 프로젝트 멤버와 3rd-party 플러그인에 태그를 설정할 수 있습니다. **이벤트 수신 설정** 메뉴에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

## 실시간 로그 이벤트 추가

## 이벤트 추가

이벤트 이름 \*

이벤트 활성화



레벨 \*

위험 경고 정보

메시지 \*

카테고리 ⓘ \*

검색 키 \*

검색 값 \*

입력된 단어가 일치하는 경우 알림이 발생합니다.

이벤트 대상 필터링 ⓘ

선택 입력  직접 입력

입력값이 없을 경우, 실시간으로 수정

선택 입력  직접 입력

검색 키 ▾

연산자 ▾

검색 값 ▾

+ 추가

ex. oid == '12345678' && level == 'ERROR'

이벤트 발생 일시 중지 \*

사용 안 함 ▾

알림 수신 후 선택한 시간 동안 이벤트가 발생하지 않습니다.

단, "이벤트 상태가 해결되면 추가 알림" 기능을 활성화한 경우에는 RECOVERED 알림 수신 후 선택한 시간 동안 이벤트가 발생하지 않습니다.

이벤트 수신 태그 ⓘ

전체 멤버 수신 + 태그 추가

[프로젝트 이벤트 수신설정 메뉴 바로가기](#)

저장



- **검색 키** : 로그 데이터 내에서 특정 값에 접근하기 위한 식별자를 의미합니다. 목록에서 선택하거나 직접 입력할 수 있습니다.

예시, HTTP 응답 상태 코드를 나타내는 값에 접근하고자 할 경우 **검색 키** `status`

- **검색 값** : **검색 키**에 해당하는 실제 데이터를 의미합니다. 로그에서 입력한 단어를 포함할 경우 경고 알림을 보냅니다. 목록에서 선택하거나 직접 입력할 수 있습니다.

예시, **검색 키** `status` **검색 값** `200`을 설정한 경우 HTTP 응답 상태 코드 200을 포함하는 로그 데이터 수집 시 경고 알림 발생

- **이벤트 대상 필터링** : **선택 입력** 옵션을 통해 **검색 키**와 **연산자**, **검색 값**을 선택해 대상을 필터링하거나 **직접 입력** 옵션을 선택할 수 있습니다. 입력값이 없을 경우 실시간으로 수집하는 데이터 전체에 대한 알림 발생 여부를 판단합니다.

예시, `AppLog` 카테고리의 로그 중 `level` 이 `ERROR`인 로그를 찾습니다. 일치하는 로그 중에서 `oid`가 `12345678`인 경우 경고 알림을 보냅니다.

## 복합 로그 이벤트 추가

✕ 이벤트 추가

**\*이벤트 이름**

**이벤트 활성화**

**레벨** 위험 경고 정보 이벤트 상태가 해결되면 추가 알림 ⓘ

---

**템플릿** 사용 안 함

**\*메시지**

**\*카테고리 ⓘ**

**사용 안 함**

2xx 상태코드 건수 count

3xx 상태코드 건수 count

4xx 상태코드 건수 count

5xx 상태코드 건수 count

정상 상태코드(2xx,3xx) 건수 count

에러 상태코드(4xx,5xx) 건수 count

에러 수신 건수 count

**데이터 조회 범위** 최근에  ▼  
선택 시간동안 수집된 로그를 조회합니다.

**로그 검색 조건 ⓘ**  선택 입력  직접 입력

▼  ▼  제외 —

+ 추가  
이벤트 발행 조건을 입력하기 위해서 카테고리를 먼저 선택해 주세요.

**\*이벤트 발행 조건** 조건에 맞는 로그 > ▼

---

**인터벌 ⓘ**  ▼

**무음 ⓘ**  ▼

**이벤트 수신 태그 ⓘ** 전체 멤버 수신 + 태그 추가

[프로젝트 이벤트 수신설정 메뉴 바로가기](#)

저장

- **템플릿** : 복합 로그 템플릿을 제공합니다.
- **로그 검색 조건**
  - **검색 키**에서 이벤트 발생 조건 대상을 선택할 수 있습니다. 선택한 **검색 키**에 해당하는 검색 값을 선택할 수 있습니다.
  - **검색 키**에서 동일한 항목을 추가할 경우 'OR' 조건으로, 다른 항목을 추가할 경우 'AND' 조건으로 동작합니다.
  - **제외** 체크 박스를 선택해 선택한 검색 값을 이벤트 발행 조건에서 제외할 수 있습니다.
  - **+ 추가**를 선택해 여러개의 이벤트 발행 조건을 추가 또는 제외 설정할 수 있습니다.
- **데이터 조회 범위** : 선택한 시간동안 수집한 로그를 조회합니다. 데이터 조회 시간을 5분으로 선택하면 최근 5분 동안 수집한 데이터를 조회해서 이벤트 발생 조건을 확인합니다.
- **이벤트 발행 조건** : 이벤트가 입력한 횟수와 선택한 연산자의 조건과 같이 발생할 때 경고 알림을 보냅니다.

예시, AppLog 카테고리의 로그 중 조건 입력에 해당하는 로그를 필터링 합니다. 조건 입력에서 제외를 체크한 경우 해당 조건으로 찾은 로그를 제외하겠다는 의미입니다. 따라서 level 이 ERROR 인 로그는 제외합니다. 최근 10분 동안 수집한 로그 중 이벤트가 5 보다 작을 경우 경고 알림을 보냅니다.

## 로그 이벤트 설정 수정하기

1. 경고 알림 > 이벤트 설정 메뉴로 이동하세요.
2. 로그 탭을 선택하세요.
3. 로그 이벤트 목록 중 수정하려는 이벤트 항목에서 오른쪽에 ✎ 버튼을 선택하세요.
4. 이벤트 설정 창이 나타나면 옵션을 수정한 다음 저장 버튼을 선택하세요.

선택한 로그 이벤트를 삭제하려면 이벤트 설정 창에서 오른쪽 위에 삭제 버튼을 선택하세요.




## 로그 이벤트 끄기

1. 경고 알림 > 이벤트 설정 메뉴로 이동하세요.
2. 로그 탭을 선택하세요.
3. 로그 이벤트 목록 중 경고 알림을 끄려는 이벤트 항목의 가장 오른쪽에 토글 버튼을 선택하세요.

다시 토글 버튼을 선택하면 해당 경고 알림이 활성화됩니다.

## 로그 이벤트 내보내기/불러오기

로그 이벤트의 설정 내용을 json 파일로 저장한 다음 불러와 재사용할 수 있습니다.

1. 경고 알림 > 이벤트 설정 메뉴로 이동하세요.
2. 로그 탭을 선택하세요.
3. 로그 이벤트 목록 위에 JSON  버튼을 선택하세요. JSON 내보내기 창이 나타납니다.
4. 내보내기 할 항목을 수정 또는 편집하세요.
5. 오른쪽 위에  내보내기 버튼을 선택하세요. 브라우저에서 json 파일을 다운로드합니다.
6. 로그 이벤트 목록 위에  버튼을 선택하세요.
7. 파일 선택 창이 나타나면 앞서 다운로드 받은 json 파일을 선택하세요.
8. JSON 가져오기 창이 나타나면 내용을 수정한 다음 + 목록에 추가하기 버튼을 선택하세요.

❗ 이벤트에 id가 존재합니다. id를 제거한 뒤 다시 시도하세요.

- 메시지가 나타나면 JSON 가져오기 창에서 id 항목을 삭제한 다음 + 목록에 추가하기 버튼을 선택하세요.
- 기존의 이벤트 항목에 덮어쓰기를 하려면 id 항목을 삭제한 다음 덮어쓰기 버튼을 선택하세요.

# 경고 알림 수신 설정

홈 화면 > 프로젝트 선택 > 경고 알림 > 이벤트 수신 설정

프로젝트 멤버들의 경고 알림 수신과 관련한 다양한 기능을 설정할 수 있습니다.

이벤트 수신 설정



▼ 사용자별 이벤트 수신 설정 (64)

이벤트 알림의 일괄 수신설정 및 접근 설정을 위한 모바일 기기 관리는 [계정 정보 메뉴](#)에서 가능합니다. [계정 정보 >](#)

이름	이메일 알림	SMS 알림	WhatsApp 알림	모바일 알림	반복 알림 (에스컬레이션)	이벤트 수신 태그
JH	<input checked="" type="checkbox"/> 수신 레벨: 위임	<input type="checkbox"/> 수신 레벨: 위임	등록된 번호가 없습니다.	<input type="checkbox"/> 모바일 기기 6대 수신 레벨: 전체	0 저장	<input type="checkbox"/> 수신 태그 미설정 알림 받기 testtag 테스트 태그 +
JT	<input checked="" type="checkbox"/> 수신 레벨: 전체	<input type="checkbox"/> 수신 레벨: 전체	등록된 번호가 없습니다.	등록된 기기가 없습니다.	0 저장	<input checked="" type="checkbox"/> 수신 태그 미설정 알림 받기 + 태그 추가
JU	<input checked="" type="checkbox"/> 수신 레벨: 전체	등록된 번호가 없습니다.	등록된 번호가 없습니다.	등록된 기기가 없습니다.	0 저장	<input checked="" type="checkbox"/> 수신 태그 미설정 알림 받기 + 태그 추가
KJ	<input type="checkbox"/> 수신 레벨: 전체	<input type="checkbox"/> 수신 레벨: 전체	등록된 번호가 없습니다.	등록된 기기가 없습니다.	0 저장	<input checked="" type="checkbox"/> 수신 태그 미설정 알림 받기 + 태그 추가
KY	<input checked="" type="checkbox"/> 수신 레벨: 전체	<input type="checkbox"/> 수신 레벨: 전체	등록된 번호가 없습니다.	등록된 기기가 없습니다.	0 저장	<input checked="" type="checkbox"/> 수신 태그 미설정 알림 받기 + 태그 추가
	<input checked="" type="checkbox"/> 수신 레벨: 전체	<input type="checkbox"/> 수신 레벨: 전체	등록된 번호가 없습니다.	등록된 기기가 없습니다.	0 저장	<input checked="" type="checkbox"/> 수신 태그 미설정 알림 받기

3rd 파티 플러그인

Slack, Telegram, Teams, Jandi, Webhook 등을 이용하여 알림 메시지를 받으실 수 있습니다.

플러그인 이름	인증 키	인증 값	반복 알림 (에스컬레이션)	이벤트 수신 태그
SLACK	[인증 키]	[인증 값]	0 저장	<input checked="" type="checkbox"/> 수신 태그 미설정 알림 받기 + 태그 추가
SLACK	[인증 키]	[인증 값]	0 저장	<input checked="" type="checkbox"/> 수신 태그 미설정 알림 받기 + 태그 추가
SLACK	[인증 키]	[인증 값]	0 저장	<input checked="" type="checkbox"/> 수신 태그 미설정 알림 받기 + 태그 추가
SLACK	[인증 키]	[인증 값]	0 저장	<input checked="" type="checkbox"/> 수신 태그 미설정 알림 받기 + 태그 추가
TELEGRAM	[인증 키]	[인증 값]	0 저장	<input checked="" type="checkbox"/> 수신 태그 미설정 알림 받기 PREV_TEST +
WEBHOOK_JSON	[인증 키]	[인증 값]	0 저장	<input checked="" type="checkbox"/> 수신 태그 미설정 알림 받기 + 태그 추가

➕ 추가하기

대량 알림 발생 방지

알림이 대량으로 발생하면 지정된 시간 동안 알림이 일시적으로 중지됩니다. 대량 알림 차단 기능을 해제하려면 (이메일 주소 옆의) '중단 해제' 버튼을 눌러주세요.

활성화  
 탐지 시간: 5분  
 탐지 횟수: 10  
 정지 시간: 3시간

저장

## 수신 수단 선택하기

이메일 알림 이외에 SMS, 모바일 알림을 선택할 수 있습니다. 원하는 알림 수신 수단의 체크 박스를 체크하면 경고 알림을 받을 수 있습니다. 알림 수신 수단의 체크 박스를 해제하면 경고 알림을 보내지 않습니다.

- ① • 이메일 알림은 회원 가입 시 입력한 이메일 주소로 알림을 보냅니다.
- 프로젝트 최고 관리자를 제외한 모든 사용자는 자신의 수신 설정만 변경할 수 있습니다.

## SMS 알림 수신 추가하기

SMS 알림 수신이 필요한 경우 [계정 관리](#)에서 사용자 전화번호를 설정하세요.

**사용자 전화번호**

전화번호는 경고 알림 문자에 사용됩니다. 전화번호를 변경하려면 SMS 인증이 필요합니다.

전화번호

일반 휴대전화는 **한국 휴대전화 번호만 등록** 가능합니다.

1. 화면 오른쪽 위에 프로필 아이콘을 선택하세요.
2. 팝업 메뉴가 나타나면 [계정 관리](#) 버튼을 선택하세요.
3. [사용자 전화번호](#) 섹션에서 [일반 휴대전화](#) 버튼을 선택하세요.
4. [전화번호](#)에 인증번호를 수신할 전화번호를 입력하세요.
5. [인증번호 전송](#) 버튼을 선택하세요.
6. 사용자의 휴대전화로 전송된 인증 번호를 입력하세요.
7. [인증하기](#) 버튼을 선택하세요.

- ① • 등록된 전화번호를 변경하려면 [번호 변경하기](#) 버튼을 선택한 다음 위의 과정을 반복하세요.
- SMS를 알림으로 수신할 수 있는 전화번호는 **한국 휴대전화 번호만** 등록할 수 있습니다.

## WhatsApp 알림 수신 추가하기

WhatsApp을 통해 알림을 수신할 수 있습니다.

**사용자 전화번호**

전화번호는 경고 알림 문자에 사용됩니다. 전화번호를 변경하려면 SMS 인증이 필요합니다.

일반 휴대전화    **WhatsApp**

전화번호    KR (+82)    전화번호 입력    인증번호 전송

1. 화면 오른쪽 위에 프로필 아이콘을 선택하세요.
2. 팝업 메뉴가 나타나면 **계정 관리** 버튼을 선택하세요.
3. **사용자 전화번호** 섹션에서 **WhatsApp** 버튼을 선택하세요.
4. **전화번호**에 인증번호를 수신할 전화번호를 입력하세요.
5. **인증번호 전송** 버튼을 선택하세요.
6. WhatsApp 애플리케이션으로 전송된 인증번호 6자리를 입력하세요.
7. **인증하기** 버튼을 선택하세요.

ⓘ 등록된 전화번호를 변경하려면 **번호 변경하기** 버튼을 선택한 다음 위의 과정을 반복하세요.

## 수신 레벨 선택하기

경고 알림 레벨에 따라 알림 수신 여부를 선택할 수 있습니다. **사용자별 이벤트 수신 설정** 섹션의 사용자 목록에서 **수신 레벨**을 **전체** 또는 **위험**을 선택하세요.

- **전체**: 모든 경고 알림을 수신할 수 있습니다.
- **위험**: 위험 레벨의 경고 알림만 수신할 수 있습니다.

## 요일 및 시간별 알람 설정하기

요일별, 시간별 알림 수신 여부를 선택할 수 있습니다. **사용자별 이벤트 수신 설정** 섹션의 사용자 목록에서 가장 왼쪽에 ▾ 버튼을 선택하세요. 경고 알림 수신을 원하는 요일을 선택하거나 시간을 입력하세요. 알림 수신 수단별로 설정할 수 있습니다.



▼ 사용자별 이벤트 수신 설정 (64)

이벤트 알림의 일괄 수신설정 및 접근 설정을 위한 모바일 기기 관리는 [계정 정보 메뉴에서](#) 가능합니다. [계정 정보 >](#)

이름	이메일 알림	SMS 알림	WhatsApp 알림	모바일 알림	반복 알림 (에스컬레이션)	이벤트 수신 태그
HS	<input checked="" type="checkbox"/> 수신 레벨: 전체	<input type="checkbox"/> 0***** 등록된 번호가 없습니다.	<input type="checkbox"/>	<input type="checkbox"/> 모바일 기기 2대 수신 레벨: 전체	0 저장	<input checked="" type="checkbox"/> 수신 태그 미설정 알림 받기 + 태그 추가
알림 수신 언어	요일 <input checked="" type="checkbox"/> 월 <input checked="" type="checkbox"/> 화 <input checked="" type="checkbox"/> 수 <input checked="" type="checkbox"/> 목 <input checked="" type="checkbox"/> 금 <input checked="" type="checkbox"/> 토 <input checked="" type="checkbox"/> 일 시간 00:00 ~ 00:00	요일 <input checked="" type="checkbox"/> 월 <input checked="" type="checkbox"/> 화 <input checked="" type="checkbox"/> 수 <input checked="" type="checkbox"/> 목 <input checked="" type="checkbox"/> 금 <input checked="" type="checkbox"/> 토 <input checked="" type="checkbox"/> 일 시간 00:00 ~ 00:00	요일 <input checked="" type="checkbox"/> 월 <input checked="" type="checkbox"/> 화 <input checked="" type="checkbox"/> 수 <input checked="" type="checkbox"/> 목 <input checked="" type="checkbox"/> 금 <input checked="" type="checkbox"/> 토 <input checked="" type="checkbox"/> 일 시간 00:00 ~ 00:00	모바일 기기 테스트	테스트	

## 경고 알림 반복 설정하기

경고 알림 발생 시간으로부터 알림 발생 상황이 해소되지 않을 경우 최초 알림 발생 시각으로부터의 알림 반복 간격을 설정할 수 있습니다. 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.

**반복 알림 (에스컬레이션) ⓘ**

0,1H,1D 저장

- ⓘ • 이 기능은 **Critical** (또는 **위험**) 등급의 모든 이벤트를 대상으로 합니다. 이벤트 추가 시 설정한 **레벨** 항목을 참조하세요.
- **M**: 분, **H**: 시간, **D**: 일, 단위를 생략하면 분 단위로 시간을 설정합니다.
- **저장** 버튼을 선택하지 않으면 설정을 완료할 수 없습니다.
- 숫자 또는 숫자+단위(**M**, **H**, **D**)로 입력하세요. 입력이 올바르지 않으면 메시지가 표시됩니다.

# 팀별, 사용자별 알림 설정하기

프로젝트의 멤버 중 특정 멤버 또는 팀을 대상으로 알림 수신 여부를 설정합니다. [메트릭스](#) 및 [이상치 탐지](#), [로그](#) 이벤트 설정의 [이벤트 수신 태그](#) 항목에서 태그를 추가하세요. 이벤트별로 경고 알림을 수신하는 멤버 또는 팀을 선택할 수 있습니다.

반대로 이벤트 수신 태그를 설정하지 않으면 전체 멤버에게 경고 알림을 보낼 수 있습니다.

> 이벤트 수신 태그 사용 예시

이벤트 수신 태그에 대한 사용 예시를 통해 팀별 또는 사용자별로 경고 알림을 전송하는 설정 방법에 대해 알아봅니다. 다음과 같이 팀 별로 서로 다른 경고 알림을 전송하도록 설정합니다.

알림 종류	서버팀 수신 여부	데브옵스팀 수신 여부
메트릭스 경고 알림	O	X
이상치 탐지 경고 알림	O	O
로그 실시간 경고 알림	X	O

## 1. 이벤트 수신 설정하기 (경고 알림 > 이벤트 수신 설정)

▼ 사용자별 이벤트 수신 설정 (64)

이벤트 알림의 일괄 수신설정 및 접근 설정을 위한 모바일 기기 관리는 [계정 정보 메뉴에서](#) 가능합니다. [계정 정보 >](#)

이름	이메일 알림	SMS 알림	WhatsApp 알림	모바일 알림	반복 알림 (에스컬레이션)	이벤트 수신 태그
DE	<input checked="" type="checkbox"/> 수신 레벨: 전체	등록된 번호가 없습니다.	등록된 번호가 없습니다.	등록된 기기가 없습니다.	0 저장	<input checked="" type="checkbox"/> 수신 태그 미설정 알림 받기 Server +
DE	<input checked="" type="checkbox"/> 수신 레벨: 전체	등록된 번호가 없습니다.	등록된 번호가 없습니다.	등록된 기기가 없습니다.	0 저장	<input checked="" type="checkbox"/> 수신 태그 미설정 알림 받기 DevOps +

프로젝트에 속한 멤버들 중 서버팀 소속은 [서버팀](#), 데브옵스팀 소속은 [데브옵스팀](#) 으로 이벤트 수신 태그를 설정하세요.

## 2. 메트릭스 경고 알림 설정하기 (경고 알림 > 이벤트 설정 > 메트릭스 > 이벤트 추가 > 이벤트 수신 설정)

## 이벤트 수신 태그

Server +

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

[프로젝트 이벤트 수신설정 메뉴](#)

이벤트 설정 시 태그를 선택하지 않은 경우 프로젝트 이벤트 수신 설정 메뉴의 나머지 수신 조건(활성화 여부 등)에 따라 알림이 발생합니다.

[태그 추가](#) 또는 + 버튼을 클릭하세요. 태그 목록에서 원하는 태그를 선택하거나 새 태그를 생성하세요. 메트릭스 이벤트에 대한 알림을 `서버팀`으로 설정한 경우입니다.

### 3. 로그 실시간 경고 알림 설정 (경고 알림 > 이벤트 설정 > 로그 > 이벤트 추가 > 이벤트 수신 태그)

## 이벤트 수신 태그 ⓘ

DevOps +

[프로젝트 이벤트 수신설정 메뉴 바로가기](#)

[태그 추가](#) 또는 + 버튼을 클릭하세요. 태그 목록에서 원하는 태그를 선택하거나 새 태그를 생성하세요. 로그 실시간 이벤트에 대한 알림을 `데브옵스팀`으로 설정한 경우입니다.

### 4. 이상치 탐지 이벤트는 전체 멤버에게 경고 알림을 전송하므로 이벤트 수신 태그를 설정하지 않습니다.

## 이벤트 수신 태그 추가하기

1. 사용자별 이벤트 수신 설정 섹션의 사용자 목록에서 [태그 추가](#) 또는 + 버튼을 선택하세요.
2. 이벤트 수신 태그 팝업 창이 나타나면 태그 입력란에 태그 이름을 입력한 다음 엔터를 입력하거나 [새 태그 생성](#)을 선택하세요.
3. 태그 색상을 선택하세요.
4. [태그 생성](#) 버튼을 선택하세요.

[태그 목록](#)에서 생성한 태그를 확인할 수 있습니다. 생성한 태그를 적용하려면 해당 태그를 선택하세요.

## 이벤트 수신 태그 해제하기

1. 사용자별 이벤트 수신 설정 섹션의 사용자 목록에서 + 버튼을 선택하세요.
2. 이벤트 수신 태그 팝업 창이 나타나면 적용된 태그 옆에 × 버튼을 선택하세요.
3. 이벤트 수신 태그 팝업 창을 닫으세요.

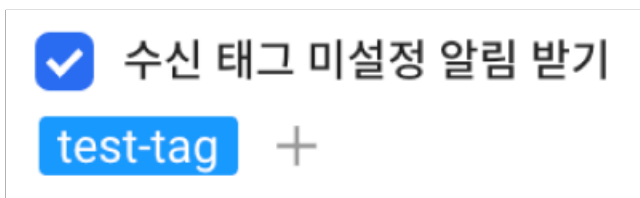
이벤트 수신 태그를 해제합니다.

## 이벤트 수신 태그 수정 및 삭제하기

1. 사용자별 이벤트 수신 설정 섹션의 사용자 목록에서 **태그 추가** 또는 **+** 버튼을 선택하세요.
2. 이벤트 수신 태그 팝업 창이 나타나면 **태그 목록**에서 수정 또는 삭제할 항목의 **✎** 버튼을 선택하세요.
3. 태그 이름을 수정하거나 색상을 변경한 다음 **적용** 버튼을 선택하세요.  
태그를 삭제하려면 **🗑️ 태그 삭제** 버튼을 선택하세요.

❗ 이벤트에 적용 중인 이벤트 수신 태그 항목은 삭제할 수 없습니다.

## 수신 태그 미설정 알림



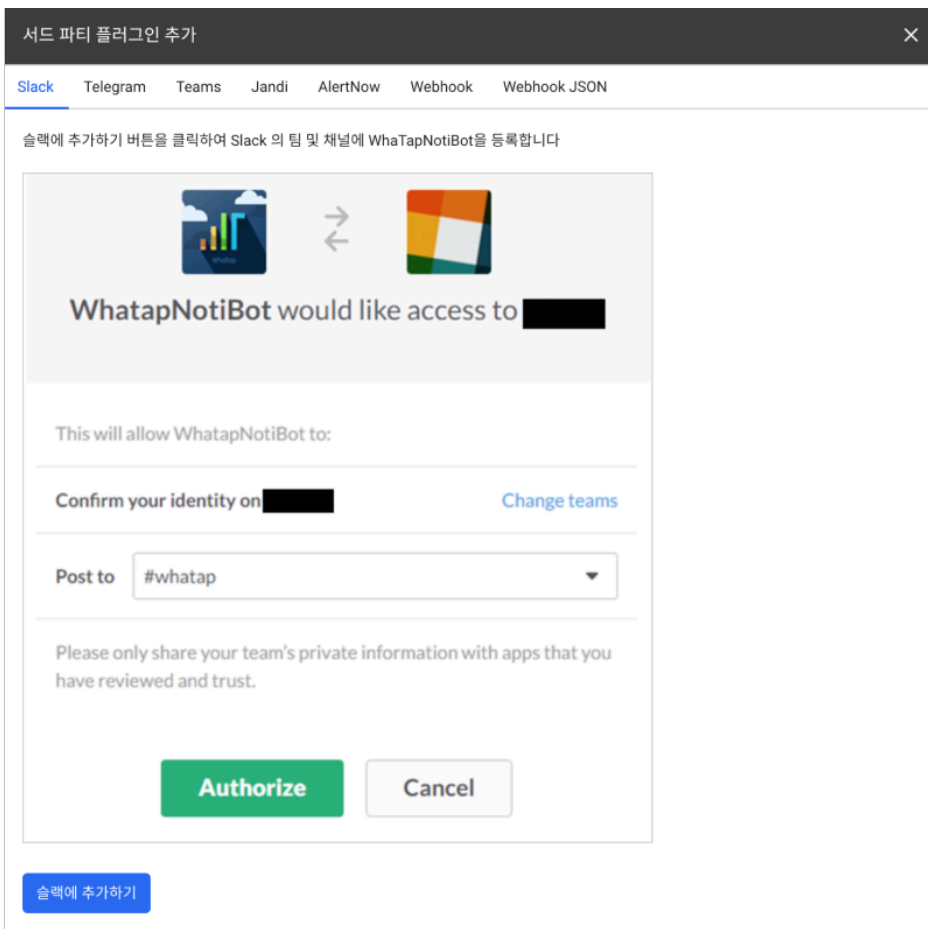
이벤트 수신 태그가 설정되지 않은 경고 알림을 받으려면 **수신 태그 미설정 알림 받기** 옵션을 선택하세요. 이벤트 수신 태그가 설정된 경고 알림만 받고 싶다면 선택을 해제하세요.

❗ 모든 경고 알림을 받지 않으려면 해당 옵션을 해제하고 선택한 이벤트 수신 태그가 없어야 합니다.

## 3rd 파티 플러그인 알림 추가하기

Slack, Telegram, Teams, Jandi, Webhook 등의 외부 애플리케이션을 통해 경고 알림을 받을 수 있습니다.

1. **경고 알림 > 이벤트 수신 설정** 메뉴에서 **3rd 파티 플러그인** 섹션의 **추가하기** 버튼을 선택하세요.
2. 원하는 서비스를 선택하세요.



3. 선택한 서비스의 화면 안내에 따라 설정을 진행하세요.
4. 모든 과정을 완료했다면 추가 버튼을 선택하세요.

ⓘ 와탭랩스의 지원 범위에 포함하지 않는 사내 메신저는 표준 Webhook, webhook json을 통해 연동할 수 있습니다.

## 대량 알림 발생 방지

알림이 대량으로 발생하면 설정한 시간 동안 경고 알림을 일시적으로 중단합니다. [경고 알림](#) > [이벤트 수신 설정](#) 메뉴에서 [대량 알림 발생 방지](#) 섹션으로 이동하세요.

## 대량 알림 발생 방지

알림이 대량으로 발생하면 지정한 시간 동안 알림이 일시적으로 중지됩니다.  
대량 알림 차단 기능을 해제하려면 (이메일 주소 옆의) '중단 해제' 버튼을 눌러주세요.

**활성화**

**탐지 시간**

**탐지 횟수**

**정지 시간**

저장

- **활성화** 토글 버튼을 선택해 기능을 켜거나 끌 수 있습니다.
- **탐지 시간** 동안 **탐지 횟수** 이상의 이벤트가 발생하면 **정지 시간** 동안 경고 알림을 중지합니다.

예를 들어, 5분 동안 20회의 이벤트가 발생하면 5분 동안 경고 알림을 중지합니다. 설정한 **정지 시간** 시간이 지나면 대량 알림 발생 방지 기능은 해제됩니다.

❗ 문자 메시지 알림이 하루 200건 이상 발생하면 일시 중지하며 다음 메시지를 표시합니다. 문자 알림 중단 기능을 해제하려면 **문자알림 중단 해제** 버튼을 선택하세요.

다량의 문자메시지가 전송되었습니다. (200건 / 일)

- ① 프로젝트에서 발생한 많은 양의 문자 메시지로 인해 문자 메시지 알림이 중지됩니다.  
- 시작 시간 : 2024-02-02 18:40:01

문자알림 중단 해제

## 경고 알림 사용자 설정하기

**계정 관리** 메뉴에서 사용자 개인의 알림 수신 레벨, 수신 수단, 요일 및 시간 등을 설정할 수 있습니다.

1. 화면의 오른쪽 위에 프로필 아이콘을 선택하세요.
2. 팝업 메뉴가 나타나면 **계정 관리** 버튼을 선택하세요.

3. 화면을 아래로 스크롤해 [알림 수신 설정](#) 섹션으로 이동하세요.
4. 수신 레벨, 수신 수단, 요일 및 시간을 설정한 다음 [저장](#) 버튼을 선택하세요.


## 경고 알림 언어 설정



프로젝트에서 발생하는 경고 알림 메시지의 언어를 변경할 수 있습니다.

1. 홈 화면에서 경고 알림 메시지의 언어를 변경할 프로젝트를 선택하세요.
2. 화면 왼쪽 메뉴에서 [관리](#) > [프로젝트 관리](#)를 선택하세요.
3. [알림 언어 관리](#) 섹션에서 원하는 언어를 선택하세요.
4. 화면 오른쪽 아래에 [저장](#) 버튼을 선택하세요.


# 이벤트 기록

홈 화면 > 프로젝트 선택 > 경고 알림 > 이벤트 기록

경고 알림이 발생한 이력을 확인할 수 있습니다. 최근 1년 이내의 이력까지 조회할 수 있습니다. 각 항목을 설정한 다음  버튼을 선택하세요.

이벤트 기록					
시간 선택	필터	애플리케이션			
< 2024/01/22 00:00 ~ 2024/01/23 00:00 <span style="color: green;">1일</span> >	제목	전체 선택			
번호	제목	이벤트 발생 시간	이벤트 해소 시간	애플리케이션	메시지
1	HITMAP_FLOOD_PATTERN	2024/01/22 09:08:00	2024/01/22 09:09:06		Flood pattern was detected in the project hitmap.
2	HITMAP_FLOOD_PATTERN	2024/01/22 08:57:00	2024/01/22 08:58:04		Flood pattern was detected in the project hitmap.
3	HITMAP_FLOOD_PATTERN	2024/01/22 08:46:00	2024/01/22 08:47:03		Flood pattern was detected in the project hitmap.
4	HITMAP_FLOOD_PATTERN	2024/01/22 08:35:00	2024/01/22 08:36:03		Flood pattern was detected in the project hitmap.
5	HITMAP_FLOOD_PATTERN	2024/01/22 08:24:00	2024/01/22 08:25:00		Flood pattern was detected in the project hitmap.
6	HITMAP_FLOOD_PATTERN	2024/01/22 08:13:00	2024/01/22 08:13:59		Flood pattern was detected in the project hitmap.
7	HITMAP_FLOOD_PATTERN	2024/01/22 08:02:00	2024/01/22 08:02:57		Flood pattern was detected in the project hitmap.
8	HITMAP_FLOOD_PATTERN	2024/01/22 07:51:00	2024/01/22 07:51:56		Flood pattern was detected in the project hitmap.
9	HITMAP_FLOOD_PATTERN	2024/01/22 07:40:00	2024/01/22 07:40:55		Flood pattern was detected in the project hitmap.
10	HITMAP_FLOOD_PATTERN	2024/01/22 07:29:00	2024/01/22 07:29:54		Flood pattern was detected in the project hitmap.
11	HITMAP_FLOOD_PATTERN	2024/01/22 07:18:00	2024/01/22 07:18:53		Flood pattern was detected in the project hitmap.
12	HITMAP_FLOOD_PATTERN	2024/01/22 07:07:00	2024/01/22 07:07:52		Flood pattern was detected in the project hitmap.
13	HITMAP_FLOOD_PATTERN	2024/01/22 06:56:00	2024/01/22 06:56:51		Flood pattern was detected in the project hitmap.
14	HITMAP_FLOOD_PATTERN	2024/01/22 06:45:00	2024/01/22 06:45:50		Flood pattern was detected in the project hitmap.
15	HITMAP_FLOOD_PATTERN	2024/01/22 06:34:00	2024/01/22 06:34:49		Flood pattern was detected in the project hitmap.
16	HITMAP_HORIZONTAL_PATTERN	2024/01/22 06:28:00	2024/01/22 06:28:48		Horizontal pattern was detected in the project hitmap.
17	HITMAP_FLOOD_PATTERN	2024/01/22 06:18:00	2024/01/22 06:18:46		Flood pattern was detected in the project hitmap.

이전 1 다음

 선택한 프로젝트에 따라 화면 이미지는 다를 수 있습니다.

## • 시간 선택

- 오른쪽에 위치한 녹색 버튼을 선택해 조회 시간을 선택할 수 있습니다.
- < 또는 > 버튼을 선택해 선택한 조회 시간 만큼 간격을 이동할 수 있습니다.
- 세부 시간을 선택하려면 날짜 또는 시간 영역을 선택하세요. 세부 시간을 설정한 다음 **적용** 버튼을 선택하세요.

- **필터:** 제목 또는 메시지 내용을 기준으로 이벤트 기록을 필터링할 수 있습니다.



- **애플리케이션**: 프로젝트에 포함된 에이전트를 선택할 수 있습니다.
- **CSV**: 조회한 이벤트 기록 결과를 csv 파일로 저장할 수 있습니다. **CSV** 버튼을 클릭하면 **최대 CSV 라인 수**를 입력한 다음 **다운로드** 버튼을 선택하세요.
- **컬럼 선택**: 조회한 이벤트 기록 결과의 열 항목을 추가할 수 있습니다.
- **이벤트 설정**: **경고 알림 > 이벤트 설정** 메뉴로 이동합니다.
- **제목 / 메시지**: **이벤트 설정** 메뉴에서 추가한 이벤트의 **이벤트명**과 **메시지** 항목의 내용입니다.
- **이벤트 발생 시각**: 이벤트가 발생한 시각입니다.
  - 이벤트가 해소되지 않고 진행 중일 경우 **진행 중** 태그가 표시됩니다.
  - 정비 중인 경우 이벤트가 발생하면 **정비 중 발생** 태그가 표시됩니다.

ⓘ **정비 계획**에 대한 자세한 내용은 [다음 문서](#)를 참조하세요.

- **이벤트 해소 시각**: 설정한 이벤트가 해결된 시각입니다. 만약 해당 컬럼이 보이지 않는다면 **컬럼 선택** 버튼을 클릭한 다음 **이벤트 해소 시각**을 선택하세요.
- **애플리케이션**: **이벤트 설정** 메뉴에서 이벤트 추가 시 **이벤트 대상 필터링** 항목을 설정하면 표시됩니다.