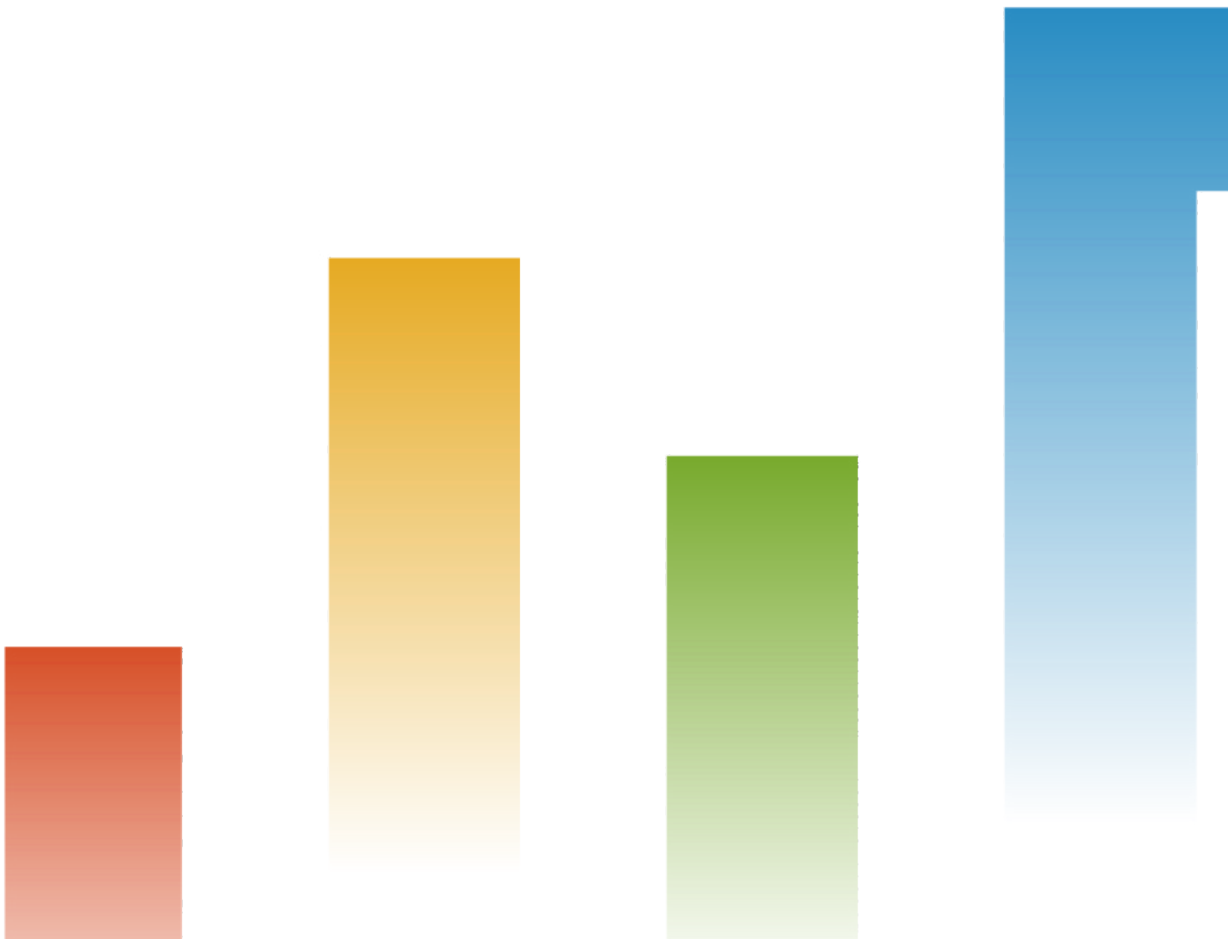
WhaTap

Java Monitoring

release date. 2026. 09. 11.



Java Documentation Contents

- JAVA Monitoring 7
  - Getting Started 9
  - Supported Environment 11
  - Agent Installation 25
    - Tomcat 31
    - JBoss 35
    - JEUS 38
    - WebLogic 42
    - WebSphere 46
    - Resin 50
    - Liberty 54
    - Jetty 57
    - Spring Boot 60
    - Play2 65
    - IBM Bluemix 68
    - Elastic Beanstalk 72
    - Docker 77
    - Batch Program 82
    - Older Java Versions 101
  - Open Source Tracking
    - Java Helper 104
    - Weaving 108
  - Install Check 117
  - Troubleshooting 119
  - Configuration 126

- Agent Name 138
- Function Control 142
- Resource Usage 146
- Network Configuration 148
- Performance 150
- Log Configuration 151
- Transaction 156
- DB/SQL 174
- HTTPC/API Call 180
- Script Plugin 182
- Number of Users 195
- Load Control 197
- Notification 200
- Apdex 204
- Statistics 205
- Error Stack 208
- Additional Options 211
- Async Tracking 213

◦ Management

- Update Agent 216
- Uninstall Agent 217
- MSA Group 218
- AES 256 Encryption 219
- Scale-In Auto Delete 220

◦ Dashboard 221

- Application Dashboard 222
- Transaction Map 250

- Active Transaction 260
- Hitmap/Trace 265
- Resource Dashboard 296
- Business Dashboard 302
- Flex Board 308
  - Flex Board Guide 311
  - Manage Widgets 321

◦ Analysis 326

- Daily Status 328
- Performance Trend 330
- Cube Analysis 334
- Cube 345
- Hitmap Analysis 349
- Multi Transaction 377
- TX Profile 383
- MSA Analysis 393

◦ Metrics 401

- Application Metrics 404
- Performance Counters 424
- Metric Chart v2 432
- Metric Chart 455
- Metric Search 458
- Anomaly Detection 460

◦ Transactions 463

- Transaction View 465
- Transaction Profile 475

- Active Transactions 503
- Multi Trace 506
- Endpoint Settings 510
- Collect Stacks 514

◦ Application Topology 516

◦ Transaction Analysis 523

◦ Report 536

- APM Report 539
- Integrated Report 547

◦ Alert 556

- Event Configuration 0
  - Metric Event 574
  - Composite Metric 596
  - Hitmap Pattern 603
  - Anomaly Event 609
  - Real-time logs 613
  - Composite log 626
  - Transaction Event 640
- Application (Classic) 652
- Event Notification Settings 656
  - Notification Format 678
- Event History and Monitoring 688
  - Real-time Notifications 696
  - System Events 699

◦ Instance Performance 702

◦ Linked Projects 720

◦ Log Monitoring 737
◦ Labs 0

Java Application Monitoring

It can be applied to all applications running in the JVM environment, such as Web Application Server (WAS) and batch applications. Free from the hassle of obtaining thread dump for application failure analysis, you can accurately identify the cause of the problem without reproducing the failure situation through the obtained stack.

WhaTap's Application Monitoring can monitor application in real time without reproducing failures.

Key features

- **Real-time transaction monitoring**

  WhaTap displays the status of active transactions in real time. Any application issue manifests itself as an increase of active transactions. If you can immediately check SQLs, outbound calls, and method stacks being executed the very moment when transactions increase, you can recognize or analyze the problem very fast.

- **Transaction performance Deep Dive**

  WhaTap can collect and analyze the performance history of all transactions. Transaction performance can be traced down to the method level as well as SQLs and HTTP calls.

  In particular, the **active stack** function enables you to analyze the method level without a cumbersome task of complicated method profile setting. With WhaTap's proprietary technology, called **Active Stack**, it provides improvement points to locate the source of problems from hidden areas that were not found in other products.

  > ⓘ Active Stack is only supported in Java, Python and .NET.

- **MSA call pattern analysis**

  WhaTap visualizes the ratio of calls based on the transaction URL for the relationships between complex applications that have the MSA structure. This enables you to view the call relations and change of call ratios.

- **Analysis of AI-based response distribution patterns**

  WhaTap automatically recognizes specific patterns through machine learning and sends alert notifications to operators. This can inform you of the issues that the operator did not recognize. It is also available in on-

premises environments, and the scope of application is defined under a separate agreement.
Application number: 10-2020-0037381

- **Analysis of the multi-project transaction linkage**

  WhaTap provides the transaction-based tracing in the MSA environment. All transactions can be traced to determine which applications were delayed.

- **Stack statistics analysis**

  Statistical analysis can find tuning points that are difficult to find in real-time monitoring or profile data. You can also select improvement targets through graphs and tables, or check response delays in specific time periods.

- **Post analysis** (**Cube**)

  You can check various metrics at a time based on the time axis. Because it is possible to specify the time zone where the response time, error count, or throughput is high, you can identify problematic factors in the same time zone or accurately view the service status.

- **Load control in service congestion**

  If there are excessive requests that the system cannot handle, part of them are rejected or incorrect ones are blocked for service stability.

- **Asynchronous transaction trace**

  WhaTap's unique BCI (Byte Code Instrumentation) technology effectively traces the performance of asynchronous transactions. It collects transaction profiles for applications that are using WebFlux, Reactor, Hystrix, etc.

Getting Started With Java Monitoring

WhaTap monitoring starts with the following steps:

Step 1. Sign up
Step 2. Select a product
Step 3. Create a project
Step 4. Check the access key

After confirming the supported environment, proceed to the agent installation step.

Sign up

To use the WhaTap monitoring service, you must first sign up. For more information about signing up, see the Account Management > Sign Up documentation.

Select a product

1. Log in to the WhaTap Monitoring Service.
2. Select **All Projects** > **Project**.
3. On the **Select product** screen, select the Go product to monitor and then click the Create button.

Create a project

After selecting the product to monitor, create a project before installing the agent.

1. On the **Create project** screen, enter or select the following items to create a project:
    - **Project name** (required): Enter the project name.
    - **Data server region** (required): Select the region where the data server is located. A region is a collection of data centers that provide cloud services. Your data is stored in the selected region.

- **Time zone** (required): Set the time zone that serves as the reference for generating alerts and reports.
- **Notification language setting** (required): Set the language for alert notification messages generated in the project (supports Korean and English).
- **Project groups**: You can group and manage multiple projects. Select a group if there is one to join.
- **Project description**: Enter an additional description for the project.

2. After completing all settings, click the **Creating a project** button.

> ⓘ When adding a project with an organization selected, you must set the **Groups of organization** as required. For more information about groups, see the following documentation.

Check the access key

The **access key** is a unique ID for activating the WhaTap service.

In the installation guide section, select the **Getting the access key** button. After automatically issuing the **access key**, proceed to the next step.

> ⓘ After creating a project, you are automatically redirected to the **Agent installation** page. If you are not redirected to the **Agent installation** page, select **Management** > **Agent installation** from the left menu.

Java Supported Environment

Java agent support environment

Before installing the agent, make sure that the JVM application server meets the following specifications.

It can be applied not only to Web Application Server (WAS) but also to all applications running on JVM, such as daemons and batch applications. For the versions earlier than Java 1.6, contact support@whatap.io or see the following.

Supported Versions

Category / JDK Version	Java 6 ~ 21	Java 22	Java 23
Support	✅	✅	✅
Support Level	Limited Support	Fully support	

Operating System

OS Family	Supported Version	Architecture
Unix	HP-UX 11.x	32bit, 64bit
	Oracle Solaris 2.8, 2.9, 10, 11	32bit, 64bit
Linux	Red Hat	Itanium 64bit
	x86 Intel Linux	32bit
Windows	Windows Server	2000 or later

OS Family	Supported Version	Architecture
	Windows	XP or later
Others	Itanium	64bit

ⓘ Supports all operating systems that can install JDK 6.

Application Server

Vendor	Product	Supported Version
Apache	Jakarta Tomcat	6.x ~ 10.x
Red Hat	JBoss Application Server	6.x, 7.x
TmaxSoft	JEUS	6.x, 7.x
Oracle	WebLogic Server	9.x ~ 12.x
	GlassFish	3.x ~ 5.x
	SUN Application Server	8.x, 9.x
IBM	WebSphere, WebSphere Liberty	-
	BlueMix	-
Caucho	Resin	3.x, 4.x
Eclipse	Jetty	-
Fujitsu	Interstage	5.x ~ 7.x
Hitachi	Cosminexus	7.x ~ 9.x

Vendor	Product	Supported Version
AWS	Elastic Beanstalk	-
Docker	Container Environment	-

ⓘ Supports all application servers that comply with Java EE Spec. See this link for details.

Framework / Library

Category	Framework	Supported Version
Web Framework	Spring Boot	1.5.x ~ 3.x
Async Framework	Netty	-
	Akka HTTP	-
	Play Framework	-
Microservices	Quarkus	-
	Quarkus Reactive	1.x ~ 2.x
Integration Framework	Mule Framework	3.9.5, 4.5

ⓘ Supports all libraries that operate on Application Servers complying with Java EE Spec.

Database Driver

Category	Database	Driver	Supported Version
RDBMS	Generic	JDBC	any compliant driver
	MySQL	JDBC	5.x ~ 8.x
	AWS MySQL	JDBC	1.x
	MariaDB	JDBC	1.x ~ 3.x
	PostgreSQL	JDBC	-
	Oracle	ojdbc	5, 6, 7, 8, 14
	Microsoft SQL Server	JDBC	-
	DB2	JDBC	-
	Derby	JDBC	-
	H2	JDBC	-
	jTDS	JDBC	-
	Tibero	-	-
NoSQL	Redis	Jedis	-
	Redis	Lettuce	5.x ~ 6.x

Compatibility and data sources for each application server

Tomcat

Compatibility

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

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

DataSource

DB	JDBC Driver file name	JDBC Ver.	JDK Ver.	Compatibility
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

JBoss

Compatibility

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

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 (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 file name	JDBC Ver.	JDK Ver.	Compatibility
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

Compatibility

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

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

DataSource

DB	JDBC Driver file name	JDBC Ver.	JDK Ver.	Compatibility
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

Compatibility

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

DataSource

DB	JDBC Driver file name	JDBC Ver.	JDK Ver.	Compatibility
Mysql	mysql-connector-java-5.1.39-bin.jar	5.1.39	JDK 1.7.0_80	O

DB	JDBC Driver file name	JDBC Ver.	JDK Ver.	Compatibility
Oracle	ojdbc6-11.2.0.2.0.jar	11.2.0.2.0	JDK 1.7.0_80	O

JETTY

Compatibility

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 file name	JDBC Ver.	JDK Ver.	Compatibility
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

Common supported environment

Service deployment models

You can use WhaTap Monitoring as a SaaS or on-premises (self-hosted) service. Whichever you choose, you use the same agents and the same monitoring features.

Deployment	Support	Description
SaaS	✅	Connects to collection servers operated by WhaTap. See the firewall section for regional addresses
On-premises (self-hosted)	✅	Collection servers you build and operate yourself (air-gapped environments supported). Contact your sales representative for the scope

> ⓘ You can also deploy WhaTap on-premises in air-gapped environments without internet connectivity.

Feature differences by deployment model

Whichever you choose, the monitoring features are the same. The mobile app is the only exception and is available on the service.whatap.io SaaS service only.

> ⓘ To use certain features on-premises, a separate agreement is required.

Supporting the browser

WhaTap Monitoring is available in web browsers and mobile Apps.

Browser	Recommended	Supported version
Google Chrome	O	111 or later

Browser	Recommended	Supported version
Mozilla FireFox	X	Latest version
Edge	X	Latest version
Safari	X	Latest version

Support guidance

- For browser compatibility and performance reasons, the latest version of Chrome is recommended.
- The user interface (UI) is implemented using HTML5 standard technologies, so Internet Explorer is not supported.

Limitations

WhaTap's web interface does not support mobile browsers. To access WhaTap on a mobile device, install the WhaTap app for Android or iOS. The WhaTap mobile app is optimized for mobile environments. For more information about the WhaTap mobile app, see this document.

Firewall

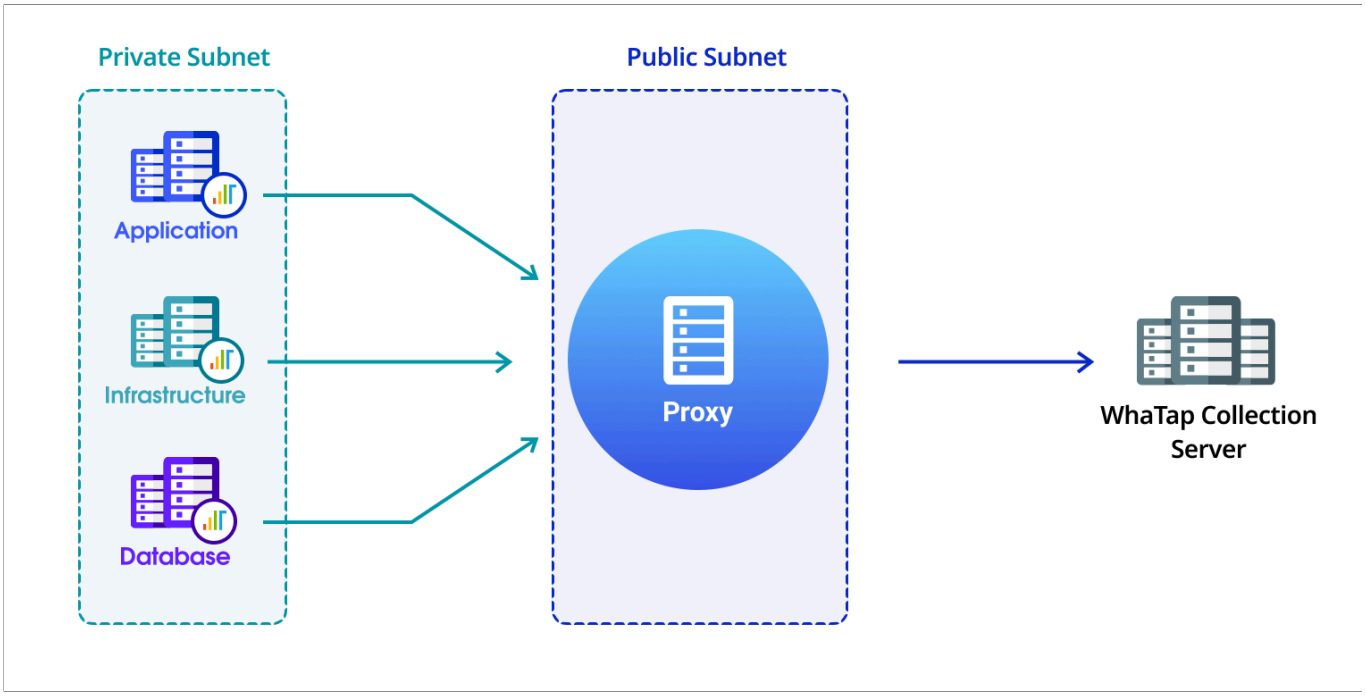
The WhaTap agent must be able to access the collection server through **TCP port 6600**. Allow access to the collection server address that is closest to the monitored target.

Source: WhaTap agent

Cloud	Destination	Destination IP	Port
AWS	WhaTap collection server (Seoul)	13.124.11.223 / 13.209.172.35	TCP 6600
	WhaTap collection server (Tokyo)	52.68.36.166 / 52.193.60.176	TCP 6600
	WhaTap collection server (Singapore)	18.138.0.93 / 18.139.67.236	TCP 6600
	WhaTap collection server (Jakarta)	108.136.91.69 / 108.137.158.44	TCP 6600

Cloud	Destination	Destination IP	Port
	WhaTap collection server (Thailand)	43.209.109.219 / 43.209.48.86	TCP 6600
	WhaTap collection server (Mumbai)	13.127.125.69 / 13.235.15.118	TCP 6600
	WhaTap collection server (California)	52.8.223.130 / 52.8.239.99	TCP 6600
	WhaTap collection server (Virginia)	107.23.220.101 / 54.236.221.105	TCP 6600
	WhaTap collection server (Frankfurt)	3.125.142.162 / 3.127.76.140	TCP 6600
Azure	WhaTap collection server (Seoul)	52.231.66.38 / 20.194.5.115	TCP 6600
	WhaTap collection server (Tokyo)	52.246.169.54 / 20.210.27.232	TCP 6600
Kakao	WhaTap collection server (Seoul)	61.109.237.237 / 61.109.238.166	TCP 6600

If the agent cannot directly access the collection server, use the provided proxy module to access it.



Mobile app

The WhaTap mobile app supports both Android and iOS environments. You can install the WhaTap app by following the link or scanning the QR code. For detailed information about the mobile app, see the following.

iOS	Android
iOS 12 or later	Android 5.0 or later

Install Java Agent

Download Agent

If the **access key** has been granted, in the **Download agent** section, the **Download** button is enabled.

1. Select **Download** to download the WhaTap agent file.

   > You can directly download it from the server by using the following command:
   >
   > ```
   > wget https://api.whatap.io/agent/whatap.agent.java.tar.gz
   > ```

2. Connect the server to be monitored.
3. Unzip the agent file at the desired location of the target server to monitor. The unzipped path becomes `$WHATAP_HOME` .

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

Agent installation file configuration

The agent file consists of a tracer that extracts data for application monitoring and passes to the WhaTap collection server, and the elements that help the tracer send the data.

whatap

Configuring the whatap.conf file

Entry of basic information

Go to the unzipped `$WHATAP_HOME` path. Enter the access key and data collection server's IP address in the $WHATAP_HOME/whatap.conf file. Go to the WhaTap monitoring service screen and then copy and paste the **access key** and **collection server IP address** from the **Download agent** item of the **Install guide** section.

```
license={access key}
whatap.server.host={collection server IP}
```

> For more information about whatap.conf, see the following.

Setting the security key

Set a security key to query SQL variables and perform HTTP queries, or use the Thread stop function.

- Java Agent 2.2.2 or later

  After creating the security.conf file in the `$WHATAP_HOME` path, enter a 6-character password consisting of alphabets and numbers as follows:

  security.conf

  ```
  paramkey=ABCDEF # SQL variable and HTTP query lookup
  threadkill=ABCDEF # Thread stop function
  ```

- Java agent 2.2.2 or earlier

  After creating the paramkey.txt file in the `$WHATAP_HOME` path, enter a 6-character password consisting of alphabets and numbers as follows:

  paramkey.txt

  ```
  ABCDEF # SQL variable and HTTP query lookup, thread stop function
  ```

When updating the Java agent version to 2.2.2, the key values in the existing paramkey.txt file are automatically applied to the security.conf file. For example, if you used `FEDCBA` in paramkey.txt and updated to the version 2.2.2, it is applied to the security.conf file as follows:

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

If the paramkey.txt file does not exist, the key value of the security.conf file is automatically created as the key value of **WHATAP**.

- For agent settings related to recording SQL parameters, see the following.
- For agent settings related to recording the HTTP parameters, see the following.

Adding the JVM options

The method how to set depending on the application server environment is provided as follows. Before installing and applying the agent, check Support environment first.

Tomcat

Configure JVM options to apply the Java agent in a Tomcat environment.

JBoss

Configure JVM options to apply the Java agent in a JBoss environment.

JEUS

Configure JVM options to apply the Java agent in a JEUS environment.

WebLogic

Configure JVM options to apply the Java agent in a WebLogic environment.

WebSphere

Configure JVM options to apply the Java agent in a WebSphere environment.

Resin

Configure JVM options to apply the Java agent in a Resin environment.

Liberty

Configure JVM options to apply the Java agent in a Liberty environment.

Jetty

Configure JVM options to apply the Java agent in a Jetty environment.

Spring Boot

Configure JVM options to apply the Java agent in a Spring Boot environment.

Play2

Configure JVM options to apply the Java agent in a Play2 framework.

IBM Bluemix

Configure JVM options to apply the Java agent in an IBM Bluemix environment.

Elastic Beanstalk

Configure Java agent settings for AWS Elastic Beanstalk environment.

Docker

Configure JVM options to apply the Java agent in a Docker environment.

Batch Program

Describes how to install and configure the WhaTap agent for batch applications.

Older Java Versions

JVM options required when applying the agent on Java 5 to 8 or earlier versions.

Add JVM Options For Tomcat

Adding the boot environment

This guide explains how to add required Tomcat options to effectively use the Java monitoring agent in the WebLogic application server environment. Set it according to the operating system.

Linux

At the top of the catalina.sh file, add the `JAVA_OPTS` environment variable.

{Tomcat_PATH}/bin/catalina.sh

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

> ⓘ The `ls ${WHATAP_HOME}/whatap.agent-*.jar | sort -V | tail -1` entry is used to apply the latest WhaTap agent for future update.

Windows

- In case of running the setup.bat file

  At the top of the catalina.bat file, add the `JAVA_OPT` environment variable.

  {Tomcat_PATH}/bin/catalina.bat

  ```
  rem ########## WHATAP START #############
  set WHATAP_HOME=C:\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_OPTS%"=="x" goto setWhatap

set JAVA_OPTS_TMP=%JAVA_OPTS:"=%
if not "x%JAVA_OPTS_TMP:whatap=%"=="x%JAVA_OPTS_TMP%" goto endWhatap

:setWhatap
set JAVA_OPTS=%JAVA_OPTS% %WHATAP_OPTS%

:endWhatap
rem ########## WHATAP END #############
```

- In case tomcat has been installed with the Windows installer
    1. Select **Start** > **Apache Tomcat** `X.Y.Z.` > **Configure Tomcat**.
    2. Select the **Java** tab.
    3. Add the `-javaagent` option to the **Java Options** field.

```
-javaagent:{WhaTap installation path}/whatap.agent-X.Y.Z.jar
```

Apache Tomcat

Java 17 or later

In case of Java 17 or later, add the following options for reflection.

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

Viewing the heap histogram

WhaTap's Java Monitoring provides the function to view the status of heap occupancy objects in the JVM memory

(sizes for each object in heap memory). In **APP** > **Instance performance analysis**, select the **Heap histogram** tab.

Java versions 6 to 8 support basic functions without the JVM options. However, for some Java versions, the JVM options must be applied as follows:

- Java 9 to Java 15

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

example

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

- Java 16 or later

```
-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}
```

> ⓘ It does not support Java 5 or earlier and IBM Java.

Next steps

Open source tracing

Depending on the framework or open source library used in applications, you may need to add agent options. For example, when using spring-boot-3.x, feign-client-11, or okhttp3-4.4 with a framework or open source, set the options as follows:

whatap.conf

```
weaving=spring-boot-3.0,feign-11,okhttp3-4.4
```

For information on how to configure the framework or open source library that suits your application environment, see the following.

Checking the installation

If you have created a project, installed an agent, and applied all JVM options, see the checklist in the following.

Installation troubleshooting

It provides various problems that may occur when installing the agent and specific instructions for resolving them. It provides solutions against various situations, including firewall configuration, SpringBoot and Tomcat JMX configuration, and server configuration using the OSGI framework. For more information, see the following.

Agent setting

It provides various features for monitoring by applying some options to the agent configuration file (whatap.conf). It includes basic configuration, configuration for server connection and data transfer, how to manage the configuration files for multiple application servers, and transaction tracing, and more. For more information, see the following.

Starting the monitoring

After all settings are made, restart the application server. The agent starts collecting data. First, check whether the monitoring data has been collected in **Application Dashboard**. For more information about **Application Dashboard**, see the following.

Add JVM Options For JBoss

Adding the boot environment

This guide explains how to add required JVM options to effectively use the Java monitoring agent in the JBoss application server environment.

For JVM options, add options for `-javaagent` and `-Djboss.modules.system.pkgs`.

JBOSS 7.0 or later & EAP 6.0 or later

Installation location: (JBoss installation path)/bin/standalone.conf(domain.conf)

```
#!/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 #############
```

> For JBoss 5.0 or earlier, set in the (JBoss installation path)/bin/run.conf path. If any error occurs, add the `-Djboss.platform.mbeanserver=true` option.

Java 17 or later

In case of Java 17 or later, add the following options for reflection.

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

Viewing the heap histogram

WhaTap's Java Monitoring provides the function to view the status of heap occupancy objects in the JVM memory (sizes for each object in heap memory). In **APP** > **Instance performance analysis**, select the **Heap histogram** tab.

Java versions 6 to 8 support basic functions without the JVM options. However, for some Java versions, the JVM options must be applied as follows:

- Java 9 to Java 15

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

example

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

- Java 16 or later

```
-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}
```

> ⓘ It does not support Java 5 or earlier and IBM Java.

Next steps

Open source tracing

Depending on the framework or open source library used in applications, you may need to add agent options. For example, when using spring-boot-3.x, feign-client-11, or okhttp3-4.4 with a framework or open source, set the options as follows:

whatap.conf

```
weaving=spring-boot-3.0,feign-11,okhttp3-4.4
```

For information on how to configure the framework or open source library that suits your application environment, see the following.

Checking the installation

If you have created a project, installed an agent, and applied all JVM options, see the checklist in the following.

Installation troubleshooting

It provides various problems that may occur when installing the agent and specific instructions for resolving them. It provides solutions against various situations, including firewall configuration, SpringBoot and Tomcat JMX configuration, and server configuration using the OSGI framework. For more information, see the following.

Agent setting

It provides various features for monitoring by applying some options to the agent configuration file (whatap.conf). It includes basic configuration, configuration for server connection and data transfer, how to manage the configuration files for multiple application servers, and transaction tracing, and more. For more information, see the following.

Starting the monitoring

After all settings are made, restart the application server. The agent starts collecting data. First, check whether the monitoring data has been collected in **Application Dashboard**. For more information about **Application Dashboard**, see the following.

Add JVM Options For JEUS

Adding JVM options

This guide explains how to add required JVM options to effectively use the Java monitoring agent in the JEUS application server environment.

1. Refer to the path of the configuration file for startup options depending on the JEUS version, and then add the `-javaagent` option.

- JEUS 7 JEUS 6

```xml title="$JEUS_HOME/domains/jeus_domain/config.xml" {6-9} showLineNumbers
<domain>
  <servers>
    <server>
      <name>server1</name>
      <jvm-config>
        <jvm-option>
          -Xmx1024m -XX:MaxPermSize=128m
          -javaagent:/whatap/whatap.agent-X.Y.Z.jar
        </jvm-option>
      </jvm-config>
    </server>
  </servers>
  ...
</domain>
```

```xml title="$JEUS_HOME/config/$hostname/JEUSMain.xml" {6-9} showLineNumbers
<node>
  <name>node01</name>
  <engine-container1>
    <name>container1</name>
    ...
    <command-option>-Xmx1024m -XX:MaxPermSize=128m
```

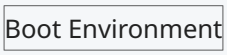
```
        -javaagent:/whatap/whatap.agent-#.#.#.jar
    </command-option>
    ...
  </engine-container1>
</node>
```

1. Restart the application server.

```
jdown && jboot
```

2. Through the application server log file and the agent log file, check whether the agent is operating normally or any error occurs. For the location of log files, see the following:
   - Agent: `$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
3. Check whether the agent has recognized the application server type and service container name.

Log in WhaTap Monitoring Service and then select a project. Select **Application** > **Instance performance analysis** and then select the **Runtime boot environment** tab. Check the items for **whatap.name** and **whatap.type**. The last element of the **whatap.name** entry must be the container name. The application server's type must be specified in the **whatap.type** field.



In case of Java 17 or later, add the following options for reflection.

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

Viewing the heap histogram

WhaTap's Java Monitoring provides the function to view the status of heap occupancy objects in the JVM memory (sizes for each object in heap memory). In **APP** > **Instance performance analysis**, select the **Heap histogram** tab.

Java versions 6 to 8 support basic functions without the JVM options. However, for some Java versions, the JVM options must be applied as follows:

- Java 9 to Java 15

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

example

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

- Java 16 or later

```
-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}
```

> ⓘ It does not support Java 5 or earlier and IBM Java.

Next steps

Open source tracing

Depending on the framework or open source library used in applications, you may need to add agent options. For example, when using spring-boot-3.x, feign-client-11, or okhttp3-4.4 with a framework or open source, set the options as follows:

whatap.conf

```
weaving=spring-boot-3.0,feign-11,okhttp3-4.4
```

For information on how to configure the framework or open source library that suits your application environment, see the following.

Checking the installation

If you have created a project, installed an agent, and applied all JVM options, see the checklist in the following.

Installation troubleshooting

It provides various problems that may occur when installing the agent and specific instructions for resolving them. It provides solutions against various situations, including firewall configuration, SpringBoot and Tomcat JMX configuration, and server configuration using the OSGI framework. For more information, see the following.

Agent setting

It provides various features for monitoring by applying some options to the agent configuration file (whatap.conf). It includes basic configuration, configuration for server connection and data transfer, how to manage the configuration files for multiple application servers, and transaction tracing, and more. For more information, see the following.

Starting the monitoring

After all settings are made, restart the application server. The agent starts collecting data. First, check whether the monitoring data has been collected in **Application Dashboard**. For more information about **Application Dashboard**, see the following.

Add JVM Options For WebLogic

Adding JVM options

This guide explains how to add required JVM options to effectively use the Java monitoring agent in the WebLogic application server environment.

Check the OS and then complete the configuration.

- 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 #############
```

> ⓘ In case of Java 17 or later, add the following options for reflection.

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

Viewing the heap histogram

WhaTap's Java Monitoring provides the function to view the status of heap occupancy objects in the JVM memory (sizes for each object in heap memory). In **APP** > **Instance performance analysis**, select the **Heap histogram** tab.

Java versions 6 to 8 support basic functions without the JVM options. However, for some Java versions, the JVM options must be applied as follows:

- Java 9 to Java 15

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

example

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

- Java 16 or later

```
-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}
```

It does not support Java 5 or earlier and IBM Java.

Next steps

Open source tracing

Depending on the framework or open source library used in applications, you may need to add agent options. For example, when using spring-boot-3.x, feign-client-11, or okhttp3-4.4 with a framework or open source, set the options as follows:

whatap.conf

```
weaving=spring-boot-3.0,feign-11,okhttp3-4.4
```

For information on how to configure the framework or open source library that suits your application environment, see the following.

Checking the installation

If you have created a project, installed an agent, and applied all JVM options, see the checklist in the following.

Installation troubleshooting

It provides various problems that may occur when installing the agent and specific instructions for resolving them. It provides solutions against various situations, including firewall configuration, SpringBoot and Tomcat JMX configuration, and server configuration using the OSGI framework. For more information, see the following.

Agent setting

It provides various features for monitoring by applying some options to the agent configuration file (whatap.conf). It includes basic configuration, configuration for server connection and data transfer, how to manage the configuration files for multiple application servers, and transaction tracing, and more. For more information, see the following.

Starting the monitoring

After all settings are made, restart the application server. The agent starts collecting data. First, check whether the monitoring data has been collected in **Application Dashboard**. For more information about **Application Dashboard**, see the following.

Add JVM Options For WebSphere

Adding JVM options

This guide explains how to add required JVM options to effectively use the Java monitoring agent in the WebSphere application server environment. To add JVM options, access the Admin console.

1. Log in the Admin Console in your web browser.
2. Select **Environment** > **Virtual Hosts** > **default host** > **Host Aliases** and then check the port number of the server to install the agent to.

   WebSphere port

3. Go to **Servers** > **Server Types** > **Websphere application servers** and then select a server to install the agent to.

   WebSpherer application servers

4. Select the **Configuration** tab on the selected server screen and then select **Java and Process Management** > **Process definition** in the **Server Infrastructure** section.

   WebSphere Configuration

5. Among the menus on the right, select **Additional Properties** > **Java Virtual Machine**.
6. In the **Configuration** tab, add `-javaagent` and `-Dwhatap.port` to the **Generic JVM arguments** field.

   ```
   -javaagent:/whatap/whatap.agent-X.Y.Z.jar
   -Dwhatap.port={port number} ## Port number of the item 2 ##
   ```

   Add JVM arguments

> In case of Java 17 or later, add the following options for reflection.
>
> ```
> 
> ```

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

Viewing the heap histogram

WhaTap's Java Monitoring provides the function to view the status of heap occupancy objects in the JVM memory (sizes for each object in heap memory). In **APP** > **Instance performance analysis**, select the **Heap histogram** tab.

Java versions 6 to 8 support basic functions without the JVM options. However, for some Java versions, the JVM options must be applied as follows:

- Java 9 to Java 15

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

example

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

- Java 16 or later

```
-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}
```

It does not support Java 5 or earlier and IBM Java.

In case the client IP address is not collected

If the client IP addresses are not collected from the dashboard or other menus, see the following.

To collect valid client IP addresses from the Java agent, it is required to pass the original client IP address from the IBM HTTP server to the backend WebSphere application server. To pass the original client IP address, add `X-Forwarded-For` to the HTTP header on the IBM HTTP server.

This is handled in more detail in the IBM Support document: Passing the real client IP from IBM HTTP Server 9.0 to WebSphere Application Server.

> ⓘ You may need an IBM account to view the IBM support documentation.

Next steps

Open source tracing

Depending on the framework or open source library used in applications, you may need to add agent options. For example, when using spring-boot-3.x, feign-client-11, or okhttp3-4.4 with a framework or open source, set the options as follows:

whatap.conf

```
weaving=spring-boot-3.0,feign-11,okhttp3-4.4
```

For information on how to configure the framework or open source library that suits your application environment, see the following.

Checking the installation

If you have created a project, installed an agent, and applied all JVM options, see the checklist in the following.

Installation troubleshooting

It provides various problems that may occur when installing the agent and specific instructions for resolving them. It provides solutions against various situations, including firewall configuration, SpringBoot and Tomcat JMX configuration, and server configuration using the OSGI framework. For more information, see the following.

Agent setting

It provides various features for monitoring by applying some options to the agent configuration file (whatap.conf). It includes basic configuration, configuration for server connection and data transfer, how to manage the configuration files for multiple application servers, and transaction tracing, and more. For more information, see the following.

Starting the monitoring

After all settings are made, restart the application server. The agent starts collecting data. First, check whether the monitoring data has been collected in **Application Dashboard**. For more information about **Application Dashboard**, see the following.

Add JVM Options For Resin

Adding JVM options

Check the method how to add JVM options in the application server environment using Resin. Depending on the Java version, the JVM options to apply may differ. Apply the JVM options for your environment. Apply the options and then restart the application server.

- Option 1: Add the `javaagent` parameter by busing the `jvm_args` property in the resin.properties file.
- Option 2: Set the `-javaagent` parameter by adding the `<jvm-args>` section to the `conf/resin.conf` or `conf/resin.xml` file.

> ⓘ
> - Through the application server log file and the agent log file, check whether the agent is operating normally or any error occurs. For the location of log files, check the following:
>   - Agent: `$WHATAP_HOME` /logs/whatap- `{SERVER_NAME}` - `{DATE}` .log
>   - RESIN 4.x: $RESIN_HOME/log/jvm-app-#.log
> - If the agent does not work normally or an error occurs, check the following.

Java 6 to Java 8

resin.properties

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

resin.conf | resin.xml

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

Java 9 to Java 15

resin.properties

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

resin.conf | resin.xml

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

Java 16 or later

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

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

Viewing the heap histogram

WhaTap's Java Monitoring provides the function to view the status of heap occupancy objects in the JVM memory (sizes for each object in heap memory). In **APP** > **Instance performance analysis**, select the **Heap histogram** tab.

Java versions 6 to 8 support basic functions without the JVM options. However, for some Java versions, the JVM options must be applied as follows:

- Java 9 to Java 15

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

  example

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

- Java 16 or later

```
-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}
```

> ⓘ It does not support Java 5 or earlier and IBM Java.

Next steps

Open source tracing

Depending on the framework or open source library used in applications, you may need to add agent options. For example, when using spring-boot-3.x, feign-client-11, or okhttp3-4.4 with a framework or open source, set the options as follows:

whatap.conf

```
weaving=spring-boot-3.0,feign-11,okhttp3-4.4
```

For information on how to configure the framework or open source library that suits your application environment, see the following.

Checking the installation

If you have created a project, installed an agent, and applied all JVM options, see the checklist in the following.

Installation troubleshooting

It provides various problems that may occur when installing the agent and specific instructions for resolving them. It provides solutions against various situations, including firewall configuration, SpringBoot and Tomcat JMX configuration, and server configuration using the OSGI framework. For more information, see the following.

Agent setting

It provides various features for monitoring by applying some options to the agent configuration file (whatap.conf). It includes basic configuration, configuration for server connection and data transfer, how to manage the configuration files for multiple application servers, and transaction tracing, and more. For more information, see the following.

Starting the monitoring

After all settings are made, restart the application server. The agent starts collecting data. First, check whether the monitoring data has been collected in **Application Dashboard**. For more information about **Application Dashboard**, see the following.

Add JVM Options For Liberty

Adding JVM options

The following explains how to add required JVM options to effectively use the Java monitoring agent in the Liberty application server environment.

In the jvm.options file, add the agent option. If no file, create a new one.

```
-javaagent:/app/whatap/whatap.agent-2.1.1.jar
-Dorg.osgi.framework.bootdelegation=whatap.* ## Additional options for OSGI framework configuration ##
```

> In case of Java 17 or later, add the following options for reflection.
>
> ```
> --add-opens=java.base/java.lang=ALL-UNNAMED
> ```

Viewing the heap histogram

WhaTap's Java Monitoring provides the function to view the status of heap occupancy objects in the JVM memory (sizes for each object in heap memory). In **APP** > **Instance performance analysis**, select the **Heap histogram** tab.

Java versions 6 to 8 support basic functions without the JVM options. However, for some Java versions, the JVM options must be applied as follows:

- Java 9 to Java 15

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

  example

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

- Java 16 or later

```
-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}
```

> ⓘ It does not support Java 5 or earlier and IBM Java.

Next steps

Open source tracing

Depending on the framework or open source library used in applications, you may need to add agent options. For example, when using spring-boot-3.x, feign-client-11, or okhttp3-4.4 with a framework or open source, set the options as follows:

whatap.conf

```
weaving=spring-boot-3.0,feign-11,okhttp3-4.4
```

For information on how to configure the framework or open source library that suits your application environment, see the following.

Checking the installation

If you have created a project, installed an agent, and applied all JVM options, see the checklist in the following.

Installation troubleshooting

It provides various problems that may occur when installing the agent and specific instructions for resolving them. It provides solutions against various situations, including firewall configuration, SpringBoot and Tomcat JMX configuration, and server configuration using the OSGI framework. For more information, see the following.

Agent setting

It provides various features for monitoring by applying some options to the agent configuration file (whatap.conf). It includes basic configuration, configuration for server connection and data transfer, how to manage the configuration files for multiple application servers, and transaction tracing, and more. For more information, see the following.

Starting the monitoring

After all settings are made, restart the application server. The agent starts collecting data. First, check whether the monitoring data has been collected in **Application Dashboard**. For more information about **Application Dashboard**, see the following.

Add JVM Options For Jetty

Adding JVM options

The following explains how to add required JVM options to effectively use the Java monitoring agent in the Jetty application server environment.

- Adding JVM options in the jetty.sh file

  $JETTY_HOME/bin/jetty.sh

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

- Starting Jetty without the shell script

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

ⓘ In case of Java 17 or later, add the following options for reflection.

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

Viewing the heap histogram

WhaTap's Java Monitoring provides the function to view the status of heap occupancy objects in the JVM memory (sizes for each object in heap memory). In **APP** > **Instance performance analysis**, select the **Heap histogram** tab.

Java versions 6 to 8 support basic functions without the JVM options. However, for some Java versions, the JVM options must be applied as follows:

- Java 9 to Java 15

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

example

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

- Java 16 or later

```
-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}
```

> ⓘ It does not support Java 5 or earlier and IBM Java.

Next steps

Open source tracing

Depending on the framework or open source library used in applications, you may need to add agent options. For example, when using spring-boot-3.x, feign-client-11, or okhttp3-4.4 with a framework or open source, set the options as follows:

whatap.conf

```
weaving=spring-boot-3.0,feign-11,okhttp3-4.4
```

For information on how to configure the framework or open source library that suits your application environment, see the following.

Checking the installation

If you have created a project, installed an agent, and applied all JVM options, see the checklist in the following.

Installation troubleshooting

It provides various problems that may occur when installing the agent and specific instructions for resolving them. It provides solutions against various situations, including firewall configuration, SpringBoot and Tomcat JMX configuration, and server configuration using the OSGI framework. For more information, see the following.

Agent setting

It provides various features for monitoring by applying some options to the agent configuration file (whatap.conf). It includes basic configuration, configuration for server connection and data transfer, how to manage the configuration files for multiple application servers, and transaction tracing, and more. For more information, see the following.

Starting the monitoring

After all settings are made, restart the application server. The agent starts collecting data. First, check whether the monitoring data has been collected in **Application Dashboard**. For more information about **Application Dashboard**, see the following.

Add JVM Options For Spring Boot

Check the method how to add JVM options in the application server environment using Spring Boot.

Adding JVM options

- In case of Java 17 or later, the application can be started by adding JVM options related to the reflection.

```
# Only support Java 17 or later
java -javaagent:{WHATAP_HOME}/whatap.agent-X.Y.Z.jar -Dwhatap.oname={Agent_Name} --add-opens=java.base/
java.lang=ALL-UNNAMED -jar {application.jar}
```

- For the JVM options in the startup script, add `-javaagent`.

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

Add the JVM options and then proceed with the appropriate agent configuration according to the versions of Spring Boot and agents.

Spring Boot 3

Spring Boot version 3.0.0 or later

- Java agent v2.2.9 or later (whatap.agent-2.2.9.jar)

  whatap.conf

```
weaving=spring-boot-3.0
```

- Java agent v2.2.5 or later (whatap.agent-2.2.5.jar)

  whatap.conf

```
weaving=spring-boot-3.0,tomcat10
```

- Java agent v2.2.4 or earlier (whatap.agent-2.2.4.jar)
  - `jakarta.servlet.http.HttpServlet` tracing (synchronization method)

    whatap.conf

    ```
    weaving=tomcat10
    ```

  - WebClient tracing (Asynchronization method)

    whatap.conf

    ```
    weaving=webflux-6.0
    ```

Spring Boot 2

Spring Boot version 2.7.0 or later

- Java agent v2.2.9 or later (whatap.agent-2.2.9.jar)

  whatap.conf

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

- Java agent v2.2.5 or later (whatap.agent-2.2.5.jar)

  whatap.conf

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

- Java agent v2.2.4 or earlier (whatap.agent-2.2.4.jar)

  WebClient tracing (Asynchronization method)

whatap.conf

```
weaving=webflux-5.3
```

Spring Boot version 2.5.0 or later

- Java agent v2.2.9 or later (whatap.agent-2.2.9.jar)

  whatap.conf

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

- Java agent v2.2.5 or later (whatap.agent-2.2.5.jar)

  whatap.conf

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

- Java agent v2.2.4 or earlier (whatap.agent-2.2.4.jar)

  WebClient tracing (Asynchronization method)

  whatap.conf

  ```
  weaving=webflux-5.3
  ```

Viewing the heap histogram

WhaTap's Java Monitoring provides the function to view the status of heap occupancy objects in the JVM memory (sizes for each object in heap memory). In **APP** > **Instance performance analysis**, select the **Heap histogram** tab.

Java versions 6 to 8 support basic functions without the JVM options. However, for some Java versions, the JVM options must be applied as follows:

- Java 9 to Java 15

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

example

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

- Java 16 or later

```
-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}
```

> ⓘ It does not support Java 5 or earlier and IBM Java.

Next steps

Open source tracing

Depending on the framework or open source library used in applications, you may need to add agent options. For example, when using spring-boot-3.x, feign-client-11, or okhttp3-4.4 with a framework or open source, set the options as follows:

whatap.conf

```
weaving=spring-boot-3.0,feign-11,okhttp3-4.4
```

For information on how to configure the framework or open source library that suits your application environment, see the following.

Checking the installation

If you have created a project, installed an agent, and applied all JVM options, see the checklist in the following.

Installation troubleshooting

It provides various problems that may occur when installing the agent and specific instructions for resolving them. It provides solutions against various situations, including firewall configuration, SpringBoot and Tomcat JMX configuration, and server configuration using the OSGI framework. For more information, see the following.

Agent setting

It provides various features for monitoring by applying some options to the agent configuration file (whatap.conf). It includes basic configuration, configuration for server connection and data transfer, how to manage the configuration files for multiple application servers, and transaction tracing, and more. For more information, see the following.

Starting the monitoring

After all settings are made, restart the application server. The agent starts collecting data. First, check whether the monitoring data has been collected in **Application Dashboard**. For more information about **Application Dashboard**, see the following.

Add JVM Options For Play2

Adding JVM options

This explains how to add required JVM options to effectively use the Java monitoring agent in the Play2 application server environment.

Set the agent options in the $PLAY_HOME/framework/build file. If you have a separate startup or stop script, apply it to that file.

```
#! /usr/bin/env sh

########## WHATAP START #############
WHATAP_HOME=/apps/whatap
WHATAP_JAR=`ls ${WHATAP_HOME}/whatap.agent-*.jar | sort -V | tail -1`
JAVA_OPTS="${JAVA_OPTS} -Dwhatap.play2=2.2.6 -javaagent:${WHATAP_JAR}"
########## WHATAP END #############

if [ -z "${PLAY_VERSION}" ]; then
  PLAY_VERSION="2.2.6"
...
```

ⓘ For Play, add the `-Dwhatap.play2=2.2.6` option other than the default settings.

ⓘ In case of Java 17 or later, add the following options for reflection.

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

Viewing the heap histogram

WhaTap's Java Monitoring provides the function to view the status of heap occupancy objects in the JVM memory (sizes for each object in heap memory). In **APP** > **Instance performance analysis**, select the **Heap histogram** tab.

Java versions 6 to 8 support basic functions without the JVM options. However, for some Java versions, the JVM options must be applied as follows:

- Java 9 to Java 15

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

example

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

- Java 16 or later

```
-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}
```

> ⓘ It does not support Java 5 or earlier and IBM Java.

Next steps

Open source tracing

Depending on the framework or open source library used in applications, you may need to add agent options. For example, when using spring-boot-3.x, feign-client-11, or okhttp3-4.4 with a framework or open source, set the options as follows:

whatap.conf

```
weaving=spring-boot-3.0,feign-11,okhttp3-4.4
```

For information on how to configure the framework or open source library that suits your application environment, see the following.

Checking the installation

If you have created a project, installed an agent, and applied all JVM options, see the checklist in the following.

Installation troubleshooting

It provides various problems that may occur when installing the agent and specific instructions for resolving them. It provides solutions against various situations, including firewall configuration, SpringBoot and Tomcat JMX configuration, and server configuration using the OSGI framework. For more information, see the following.

Agent setting

It provides various features for monitoring by applying some options to the agent configuration file (whatap.conf). It includes basic configuration, configuration for server connection and data transfer, how to manage the configuration files for multiple application servers, and transaction tracing, and more. For more information, see the following.

Starting the monitoring

After all settings are made, restart the application server. The agent starts collecting data. First, check whether the monitoring data has been collected in **Application Dashboard**. For more information about **Application Dashboard**, see the following.

Add JVM Options For IBM Bluemix

Applying the agent

The following explains how to apply the WhaTap agent when deploying with **IBM BlueMix**.

IBM BlueMix provides the **WebSphere Liberty** environment with the container. **Liberty** is a lightweight environment different from **WebSphere Application Server**, and it is similar to how **Spring Boot** works.

> ⓘ In the **Platform as a Service (PaaS)** environment, including the WhaTap agent settings, deployment is performed to the target application.

The following is an example of the configuration environment. These files are created in the local development environment when configuring the environment according to Guide.

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

1. In `${APP_HOME}`, create the src/main/resources/whatap-agent/ directory and then copy jar files and conf files.

```
$ 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. Add options in the `${APP_HOME}` /manifest.yml file. Because it is a yml file, write it with according to spacing and indentation standards.

```
---
applications:
- name: sample-java-helloworld
random-route: true
memory: 256M
path: target/JavaHelloWorldApp.war
# Add from here.
env:
  JAVA_OPTS: "-javaagent:/{APPLICATION_DIR}/WEB-INF/classes/whatap-agent/whatap.agent-X.Y.Z.jar
-Dorg.osgi.framework.bootdelegation=whatap.* "
```

ⓘ In case of Java 17 or later, add the following options for reflection.

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

ⓘ The whatap.conf configuration is applied the same as in non-PaaS environments. Because it may be difficult to identify the agent name after application, apply the agent name that suits the situation. For more information about the agent naming, see the following.

Viewing the heap histogram

WhaTap's Java Monitoring provides the function to view the status of heap occupancy objects in the JVM memory (sizes for each object in heap memory). In **APP** > **Instance performance analysis**, select the **Heap histogram**

tab.

Java versions 6 to 8 support basic functions without the JVM options. However, for some Java versions, the JVM options must be applied as follows:

- Java 9 to Java 15

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

example

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

- Java 16 or later

```
-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}
```

> ⓘ It does not support Java 5 or earlier and IBM Java.

Next steps

Open source tracing

Depending on the framework or open source library used in applications, you may need to add agent options. For example, when using spring-boot-3.x, feign-client-11, or okhttp3-4.4 with a framework or open source, set the options as follows:

whatap.conf

```
weaving=spring-boot-3.0,feign-11,okhttp3-4.4
```

For information on how to configure the framework or open source library that suits your application environment, see the following.

Checking the installation

If you have created a project, installed an agent, and applied all JVM options, see the checklist in the following.

Installation troubleshooting

It provides various problems that may occur when installing the agent and specific instructions for resolving them. It provides solutions against various situations, including firewall configuration, SpringBoot and Tomcat JMX configuration, and server configuration using the OSGI framework. For more information, see the following.

Agent setting

It provides various features for monitoring by applying some options to the agent configuration file (whatap.conf). It includes basic configuration, configuration for server connection and data transfer, how to manage the configuration files for multiple application servers, and transaction tracing, and more. For more information, see the following.

Starting the monitoring

After all settings are made, restart the application server. The agent starts collecting data. First, check whether the monitoring data has been collected in **Application Dashboard**. For more information about **Application Dashboard**, see the following.

Add JVM Options For Elastic Beanstalk

Applying the agent

The following explains how to apply the WhaTap agent when deploying with **Elastic Beanstalk**.

> ⓘ In the **Platform as a Service (PaaS)** environment, including the WhaTap agent settings, deployment is performed to the target application.

1. To release the Spring Boot result as jar, the content of .ebextensions cannot be applied. Compress the following files and deploy the zip file.
   - .elasticbeanstalk/config.yml: Automatically created under the directory where the eb command is executed
   - .ebextensions/`{config_name}`.config: eb configuration file
   - Procfile: File to configure the JVM command line options
   - `{application}`.jar: applicaion.jar file to execute
2. Compress the WhaTap agent file to be applied to the service and then upload it to S3 (or a public downloadable path).
   - paramkey.txt
   - whatap.agent-2.0_25.jar
   - whatap.conf

   ```
   # Compressing a directory
   zip -r whatap-agent.zip agent
   ```
3. Enter the WhaTap agent path in the .ebextensions/`{config_name}`.config file.

   When its zip file to download exists, execute the eb command to download the file and then unzip it.

```
sources:
  target directory:
    S3 path
```

Example

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

4. Enter the JVM command with the WhaTap agent option added to create a Procfile.

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. Compress the following files: Procfile, .ebextensions, and `{application}` .jar.

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

6. Add the zip file path to deploy in the .elasticbeanstalk/config.yml file.

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

7. Execure the eb deployment command.

```
eb deploy
```

In case of Java 17 or later, add the following options for reflection.

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

For more information, see AWS Guide.

Viewing the heap histogram

WhaTap's Java Monitoring provides the function to view the status of heap occupancy objects in the JVM memory (sizes for each object in heap memory). In **APP** > **Instance performance analysis**, select the **Heap histogram** tab.

Java versions 6 to 8 support basic functions without the JVM options. However, for some Java versions, the JVM options must be applied as follows:

- Java 9 to Java 15

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

example

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

- Java 16 or later

```
-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}
```

> ⓘ It does not support Java 5 or earlier and IBM Java.

Next steps

Open source tracing

Depending on the framework or open source library used in applications, you may need to add agent options. For example, when using spring-boot-3.x, feign-client-11, or okhttp3-4.4 with a framework or open source, set the options as follows:

whatap.conf

```
weaving=spring-boot-3.0,feign-11,okhttp3-4.4
```

For information on how to configure the framework or open source library that suits your application environment, see the following.

Checking the installation

If you have created a project, installed an agent, and applied all JVM options, see the checklist in the following.

Installation troubleshooting

It provides various problems that may occur when installing the agent and specific instructions for resolving them. It provides solutions against various situations, including firewall configuration, SpringBoot and Tomcat JMX configuration, and server configuration using the OSGI framework. For more information, see the following.

Agent setting

It provides various features for monitoring by applying some options to the agent configuration file (whatap.conf). It includes basic configuration, configuration for server connection and data transfer, how to manage the configuration files for multiple application servers, and transaction tracing, and more. For more information, see the following.

Starting the monitoring

After all settings are made, restart the application server. The agent starts collecting data. First, check whether the monitoring data has been collected in **Application Dashboard**. For more information about **Application Dashboard**, see the following.

Add JVM Options For Docker

Docker is a container-based virtualization platform. When you run Java applications in a Docker container, you can adjust the application's performance and resource usage by adding JVM options. This document explains how to modify the Dockerfile to add JVM options.

> ⓘ The Docker must have been installed and the Docker images must be built for Java applications.

Applying the agent

Based on an existing Docker image, build an image with the WhaTap agent settings added.

> ⓘ This document covers only the installation of the Java agent. To apply with WhaTap Kubernetes monitoring, see the following.

Creating the Docker build directory

Create a Docker build directory.

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

Creating the agent configuration file

Create the whatap.conf file.

```
1 cat >{Docker build Dir}/whatap.conf <<EOL
2 #Enter the access key.
3 license=XXXXXXXXXXXXX-XXXXXXXXXXXXX-XXXXXXXXXXXXX
4 #Enter the collection server IP.
5 whatap.server.host=xx.xx.xx.xx/yy.yy.yy.yy
6 EOL
```

Creating the Dockerfile

To build an image, you can copy the jar files from the WhaTap image to the `-javaagent` option.

```
cat >/home/silver/whatap/docker/Dockerfile <<EOL
FROM whatap/kube_mon as build
## Create an actual image (Add the WhaTap image)
## $Image_Name (image name)
FROM $Image_Name
RUN mkdir -p /whatap
COPY --from=build /data/agent/micro/whatap.agent-*.jar /whatap
COPY ./whatap.conf /whatap/
#...
EOL
```

Applying `JAVA_OPTS`

Add the following to `JAVA_OPT`:

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

Building the Docker

Build the Docker.

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

ⓘ In case of Java 17 or later, add the following options for reflection.

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

You can modify the Java agent file name by using the `Rename` function. Once the Java agent name has been modified, enter a new name in `JAVA_OPTS`.

Example of how to modify the Java agent name

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

Viewing the heap histogram

WhaTap's Java Monitoring provides the function to view the status of heap occupancy objects in the JVM memory (sizes for each object in heap memory). In **APP** > **Instance performance analysis**, select the **Heap histogram** tab.

Java versions 6 to 8 support basic functions without the JVM options. However, for some Java versions, the JVM options must be applied as follows:

- Java 9 to Java 15

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

example

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

- Java 16 or later

```
-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}
```

> ⓘ It does not support Java 5 or earlier and IBM Java.

Next steps

Open source tracing

Depending on the framework or open source library used in applications, you may need to add agent options. For example, when using spring-boot-3.x, feign-client-11, or okhttp3-4.4 with a framework or open source, set the options as follows:

whatap.conf

```
weaving=spring-boot-3.0,feign-11,okhttp3-4.4
```

For information on how to configure the framework or open source library that suits your application environment, see the following.

Checking the installation

If you have created a project, installed an agent, and applied all JVM options, see the checklist in the following.

Installation troubleshooting

It provides various problems that may occur when installing the agent and specific instructions for resolving them. It provides solutions against various situations, including firewall configuration, SpringBoot and Tomcat JMX configuration, and server configuration using the OSGI framework. For more information, see the following.

Agent setting

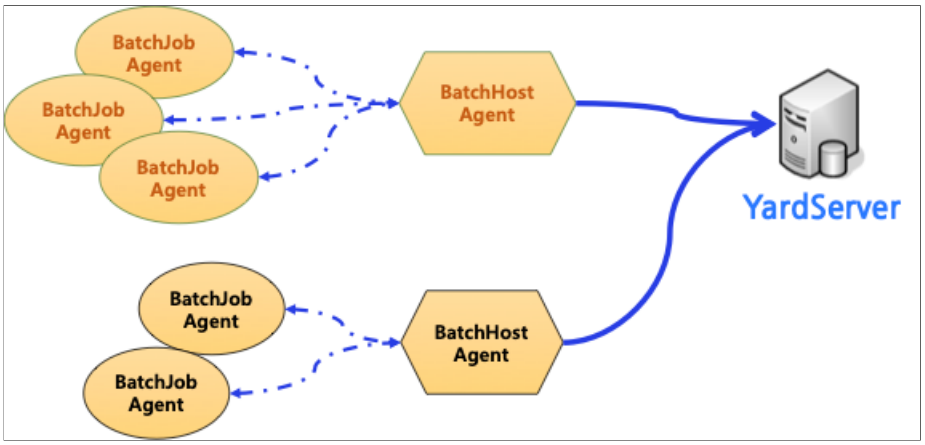
It provides various features for monitoring by applying some options to the agent configuration file (whatap.conf). It includes basic configuration, configuration for server connection and data transfer, how to manage the configuration files for multiple application servers, and transaction tracing, and more. For more information, see the following.

Starting the monitoring

After all settings are made, restart the application server. The agent starts collecting data. First, check whether the monitoring data has been collected in **Application Dashboard**. For more information about **Application Dashboard**, see the following.

Batch Program Monitoring

Describes how to install and configure the WhaTap agent for batch applications.



Batch monitoring architecture

Unlike a WAS, a batch application starts and stops repeatedly. Batch monitoring therefore uses two separate processes.

Process	Role	Instances
Batch Host	Sends data received from the Batch Agent to the collection server	One per host
Batch Agent	Collects batch job data and sends it to the Batch Host	One per batch job

The two processes communicate over UDP. This design lets you run multiple batch jobs in parallel. Install the Batch Host and the Batch Agent on the same server that runs the batch jobs.

Network requirements

Path	Protocol/Port
Batch Host → collection server	TCP 6600 (outbound)
Batch Agent → Batch Host	UDP 6611



Batch program monitoring is an add-on to Java monitoring, but it runs as an independent system separate from the Java monitoring agent. Java agent options do not apply to batch monitoring. Use only the options described in this document. Options that use the `profile_` prefix in the Java agent use the `trace_` prefix in batch programs.

Downloading the agent

Run the following command to download the agent for batch application monitoring.

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

The extracted files are structured as follows.

```
{BATCH_HOME}/
├── whatap.agent.batch.host-X.Y.Z.jar   Batch Host
├── whatap.agent-X.Y.Z.jar              Java agent
├── whatap.conf                     Batch Host configuration file
└── agent/
    ├── whatap.agent.batch-X.Y.Z.jar    Batch Agent
```

```
└── whatap.conf          Batch Agent configuration file
```

If the `whatap.conf` file does not exist, create it and add the `license` and `whatap.server.host` values found in the **Settings** > **Agent installation** menu.

Configuration file location

> This is where deployments fail most often. Be sure to check it.
>
> WhaTap determines the location of configuration files and logs based on `whatap.home`. How `whatap.home` is resolved depends on how you start the process. If the configuration file is not found, the startup banner shows `PCODE(0)` and no data is collected.

How whatap.home is resolved

The resolution order is as follows.

1. If you specify `-Dwhatap.home`, that value always takes precedence.
2. Without it, a process started with `-javaagent` (the Batch Agent) uses the directory containing the agent jar as `whatap.home`.
3. A process started by specifying a main class with `-cp` (the Batch Host) does not get this automatic resolution, so the working directory at startup becomes `whatap.home`.

Process	Startup method	whatap.home	Configuration file read
Batch Host	`-cp` + main class	Working directory	`whatap.conf` in the working directory
Batch Agent	`-javaagent`	Directory containing the agent jar	`whatap.conf` in the same directory as the jar

> Always move to the installation directory before starting the Batch Host. When a job scheduler or job management product starts the process, the working directory may differ from what you expect. Change the working directory explicitly inside the startup script.

Batch Host configuration

You can find the `license` and `whatap.server.host` values in the project's **Settings** > **Agent installation** menu.

{BATCH_HOME}/whatap.conf

```
license=
whatap.server.host=
whatap.server.port=6600
```

Option	Default	Description
`license`	-	Project access key
`whatap.server.host`	-	Collection server address. Separate redundant servers with a slash (/). Example) `10.0.0.1/10.0.0.2`
`whatap.server.port`	`6600`	Collection server port

These three items are enough to run. Add the options described below only when you need them.

UDP listener configuration

The address and port on which the Batch Host receives data from the Batch Agent.

{BATCH_HOME}/whatap.conf

```
net_udp_listen_ip=0.0.0.0
net_udp_listen_port=6611
```

Option	Default	Description
net_udp_listen_port	6611	Port that receives UDP data. Change it if the default port is already in use
net_udp_listen_ip	0.0.0.0	IP address to bind for receiving UDP data. Can be changed, but the default is recommended

If you change the UDP port, you must change it on both the Batch Host and the batch job.

Target	Configuration file	Option
Receiver (Batch Host)	{BATCH_HOME}/whatap.conf	net_udp_listen_port
Sender (batch job)	whatap.conf of the batch job	net_udp_port

If you change only one side, transaction data is lost without any error. The batch job keeps sending to the old port while the Batch Host listens only on the new one. Restart both the Batch Host and the batch jobs after the change.

Trace step collection

An individual transaction trace cannot collect steps indefinitely. Trace steps are stored in a length-limited buffer per transaction, and the default configuration collects up to **1,000 steps per batch job**.

{BATCH_HOME}/whatap.conf

```
split_trace_enabled=false
```

Option	Default	Description
split_trace_enabled	false	Splits the trace before sending

Steps beyond the limit are **dropped without any indication in the trace.** If the step numbers in the trace analysis screen stop near 1,000, this is the cause.

> Contact technical support if you need the full processing history of a batch job with a large number of steps.

Profiling expiration time

The Batch Host periodically checks running jobs and force-terminates profiling for any job that has not sent a step for a certain period.

{BATCH_HOME}/whatap.conf

```
profile_expired_time=30000
```

Option	Default	Description
profile_expired_time	30000	Force-terminates profiling when this time (ms) passes after the last step was received

This cleans up automatically so that transactions do not remain in a running state when a batch job terminates abnormally. A `lost-profile` step is recorded on force-terminated transactions.

If a period with no instrumented processing, such as waiting for an external system response, exceeds this time, profiling is force-terminated even while the job is running normally. For batch jobs with such periods, set a value with enough margin over the actual idle period.

Batch Agent configuration

{BATCH_HOME}/agent/whatap.conf

```
net_udp_ip=127.0.0.1
net_udp_port=6611

job_name=sample-batch-job
time_limit=300000

trace_httpc_limit=1000000
```

```
trace_sql_limit=1000000

active_stack_enabled=true
active_stack_start_wait_time=1000
active_stack_time1=5000
active_stack_time2=10000
```

Option	Default	Description
net_udp_ip	127.0.0.1	IP address of the Batch Host to send data to. No change needed when running on the same server
net_udp_port	6611	UDP port of the Batch Host to send data to. Must match the Batch Host's net_udp_listen_port
job_name	-	Name identifying the batch job. Shown as the transaction name. Set a unique value per job
time_limit	300000	Threshold (ms) for showing a delay warning. Does not stop the job
trace_httpc_limit	1000000	Maximum HTTP calls collected per batch job
trace_sql_limit	1000000	Maximum SQL statements collected per batch job
active_stack_enabled	true	Whether to collect active stacks
active_stack_start_wait_time	1000	Wait time (ms) before collecting the first active stack after the job starts
active_stack_time1	5000	Active stack collection interval (ms) for the first five collections
active_stack_time2	10000	Active stack collection interval (ms) from the sixth collection onward

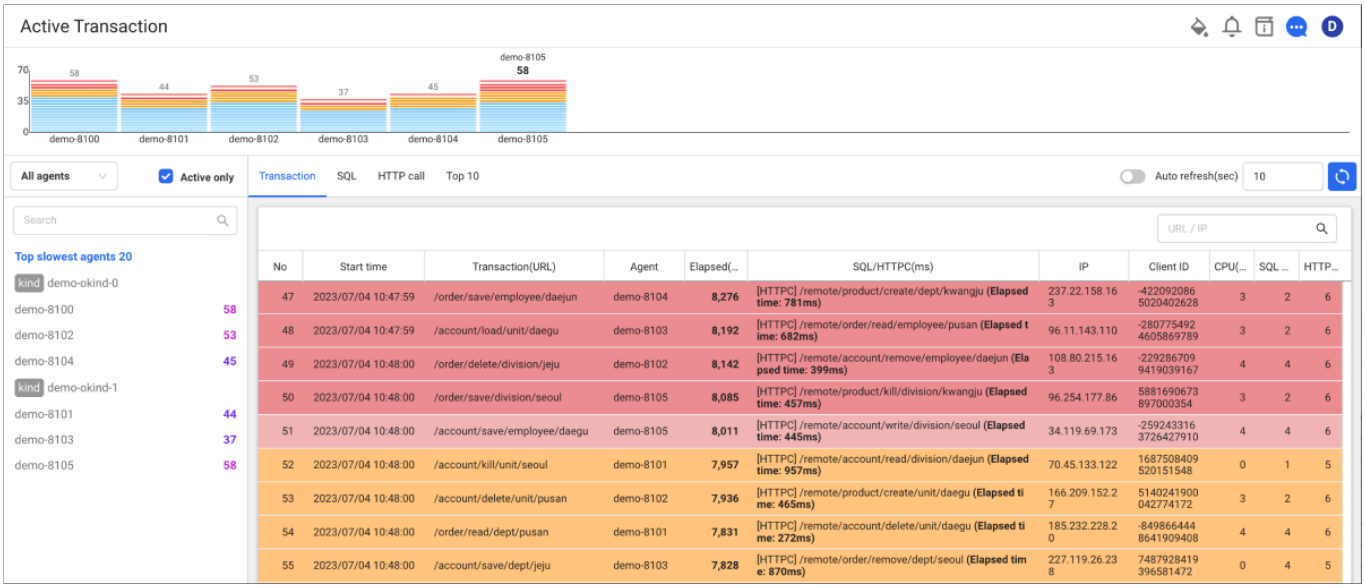
net_udp_ip and net_udp_port work with their defaults in the typical setup where the Batch Agent runs on the same server as the Batch Host. Add the other options only when you need them.

Setting time_limit

Batch jobs usually run for a long time, and the expected duration differs per job. Set `time_limit` for each job.

When you set `time_limit`, the warning thresholds on the active transaction screen are calculated automatically. No separate configuration is needed.

Elapsed time	Display
Up to 70% of `time_limit`	Default color
Over 70% of `time_limit`	Yellow
Over `time_limit`	Red



The color is determined by the elapsed time of the entire job. Even when the trace is split, the decision uses the elapsed time of the whole job rather than each segment.

Active stack collection timing

The Batch Agent collects stack traces of the running batch job at fixed intervals. You can review collected active stacks as **ActiveStack** steps in the trace analysis screen.

`active_stack_time1` and `active_stack_time2` are collection intervals, not thresholds. With the default values, the collection timing is as follows.

Collection	Timing	Applied option
1st	1 second after the batch job starts	`active_stack_start_wait_time`
2nd to 5th	5 seconds after the previous collection	`active_stack_time1`
6th onward	10 seconds after the previous collection	`active_stack_time2`

With the default values, a batch job running for 90 seconds collects about 12 active stacks.

The switchover point of five collections is fixed and cannot be changed. You can change only the three time settings, `active_stack_start_wait_time`, `active_stack_time1`, and `active_stack_time2`.

The minimum value for `active_stack_time1` and `active_stack_time2` is 100 milliseconds. Values below 100 are applied as 100.

To stop collecting active stacks, set `active_stack_enabled=false`.

Collection limits

The maximum number of HTTP calls and SQL statements collected per batch job. Each defaults to one million.

Option	Default
`trace_httpc_limit`	`1000000`
`trace_sql_limit`	`1000000`

When the configured count is exceeded, collection of that data type stops at that point and the excess is not reflected in statistics. The batch job itself keeps running and the transaction completes normally.

> No indication is left in the trace when collection stops. If the actual processing count may exceed the limit, set a sufficiently large value in advance.

> If the SQL count or HTTP call count shown in the transaction list exactly matches the configured limit, the limit may have been reached.
>
> Neither option has a product-side maximum. However, a large value increases the memory usage of the batch process, so verify it in a test environment before applying it to production.

Separating configuration per job

Set different values of `job_name` and `time_limit` for each job. Configuration files are read based on `whatap.home`, so specifying a per-job directory with `-Dwhatap.home` is recommended.

Keep the agent jar in one place and prepare only the configuration files per job.

```
{BATCH_HOME}/agent/whatap.agent.batch-X.Y.Z.jar   Shared by all jobs

{JOB_HOME}/
├── job-a/
│   ├── job-a.jar
│   └── whatap.conf      job_name=job-a, time_limit=600000
└── job-b/
    ├── job-b.jar
    └── whatap.conf      job_name=job-b, time_limit=120000
```

A `logs` directory is created in the directory the configuration file was read from. If the job directory has no `logs` directory, the configuration in that directory was not used. Use this as an indicator when troubleshooting.

Running

Startup order

Start only one Batch Host for all batch jobs.

1. Start the Batch Host once before running any batch job.
2. Run each batch job with the `-javaagent` option. You can run multiple jobs in parallel.

3. Stop the Batch Host after all batch jobs have finished.

> Configuration files are read only once at startup. After a configuration change, the Batch Agent applies it from the next job, and the Batch Host requires a restart.
>
> Force-terminating the Batch Host process can lose data sent by running jobs. Stop it after all jobs have finished.

Starting the Batch Host

- Linux Windows

Batch Host

```
cd {BATCH_HOME}
java -cp {BATCH_HOME}/whatap.agent.batch.host-X.Y.Z.jar:{BATCH_HOME}/whatap.agent-X.Y.Z.jar \
    -Dwhatap.name=batch whatap.agent.batch.App
```

For Java 17 or later, add the reflection option.

Batch Host, Java 17 or later

```
cd {BATCH_HOME}
java --add-opens=java.base/java.lang=ALL-UNNAMED \
    -cp {BATCH_HOME}/whatap.agent.batch.host-X.Y.Z.jar:{BATCH_HOME}/whatap.agent-X.Y.Z.jar \
    -Dwhatap.name=batch whatap.agent.batch.App
```

Batch Host

```
cd /d {BATCH_HOME}
java -cp "{BATCH_HOME}\whatap.agent.batch.host-X.Y.Z.jar;{BATCH_HOME}\whatap.agent-X.Y.Z.jar" -Dwhatap.name=batch
whatap.agent.batch.App
```

For Java 17 or later, add the reflection option.

Batch Host, Java 17 or later

```
cd /d {BATCH_HOME}
```

```
java --add-opens=java.base/java.lang=ALL-UNNAMED -cp "{BATCH_HOME}\whatap.agent.batch.host-
X.Y.Z.jar;{BATCH_HOME}\whatap.agent-X.Y.Z.jar" -Dwhatap.name=batch whatap.agent.batch.App
```

On Windows the classpath separator is a semicolon (;). The command prompt treats a semicolon as an argument separator, so wrap the entire classpath in double quotation marks.

Running a batch job

- Linux Windows

Batch Job

```
java -javaagent:{BATCH_HOME}/agent/whatap.agent.batch-X.Y.Z.jar \
    -Dwhatap.home={JOB_HOME}/job-a \
    -jar {JOB_HOME}/job-a/job-a.jar
```

For Java 17 or later, add `--add-opens=java.base/java.lang=ALL-UNNAMED` .

Batch Job

```
java -javaagent:{BATCH_HOME}\agent\whatap.agent.batch-X.Y.Z.jar ^
    -Dwhatap.home={JOB_HOME}\job-a ^
    -jar {JOB_HOME}\job-a\job-a.jar
```

For Java 17 or later, add `--add-opens=java.base/java.lang=ALL-UNNAMED` .

Verifying startup

Check whether the configuration was applied using the banner shown at startup. The Batch Host and Batch Agent banners differ.

Batch Host

```
WhaTap vX.Y.Z YYYYMMDD PCODE(project code)
Java <JDK version> asmlib<version> asm<version>
whatap.agent-X.Y.Z.jar
```

```
./whatap.conf
```

Display	Description
PCODE(number)	Normal. The number is the project code
PCODE(0)	The license or the configuration file was not applied. See Troubleshooting

The third line shows the Java agent jar file name, not the Batch Host jar. This is expected behavior.

Batch Agent

```
WhaTap vX.Y.Z YYYYMMDD
Java <JDK version> asmlib<version> asm<version>
whatap.agent.batch-X.Y.Z.jar
{JOB_HOME}/job-a/whatap.conf
```

The Batch Agent banner does not show `PCODE`. **The configuration file path printed on the fourth line** is how you confirm that the per-job configuration was applied. If it is not the path of the intended job directory, `whatap.home` was resolved differently than expected.

The Java and asm values in the banner depend on the JDK that ran the process. Starting the two processes with different JDKs shows different values.

Reviewing in the product

Batch jobs appear as transactions in the Hitmap and the transaction list. The transaction name is the value set in `job_name`.

Virtual transactions

When `split_trace_enabled=true`, jobs with many steps have their traces split and sent in fixed-size segments. Each split segment appears as a virtual transaction with a sequence number appended to its name. **The default is `false`, so virtual transactions do not appear with the default configuration.**

The following example uses `job_name=sample-job` with about 5,000 steps.

Transaction name	Content	Duration shown
`sample-job-1` to `sample-job-4`	Split segments (virtual transactions)	Processing time of each segment
`sample-job`	Actual transaction	Duration of the whole job

Check the duration of the whole job on the transaction without a sequence number. Step numbers are continuous across the entire job.

- Virtual transactions are reflected in the Hitmap and TPS, but statistics count them as a single job. The transaction count is not inflated.
- Virtual transactions show a CPU time of 0 and a client IP address of `0.0.0.0`.
- When you download step execution history as a file from the trace analysis screen, a split trace produces files split the same way. Download each virtual transaction if you need the step history of the whole job.

Metrics collected by the Batch Agent

The dashboard collects the Heap Memory of the Batch Host, and collects batch application performance indicators as metrics (TagCount: `batch_job_counter`).

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`

Troubleshooting

The Batch Host startup banner shows PCODE(0)

1. Check whether `whatap.conf` exists in the path printed on the fourth line of the banner.
2. If it does not, `whatap.home` was resolved differently than expected. Check the configuration file location and specify it explicitly with `-Dwhatap.home` if needed.
3. If the configuration file exists, check that the `license` value is correct. An invalid value records `WhaTap AccessKey Error` in the log.

java.net.BindException error

UDP 6611 is already in use. If you force-terminated the Batch Host, a Java process may remain and keep holding the port.

- **Linux** **Windows**

```
ss -anu | grep 6611
ps -ef | grep whatap.agent.batch.App
```

```
netstat -ano | findstr ":6611"
```

Terminate the remaining process and restart the Batch Host.

Per-job configuration is not applied

The directory was not recognized as `whatap.home`. Check the path printed on the fourth line of the startup banner.

The most common cause is keeping the agent jar in a shared location while placing configuration files only in job directories. The directory of the jar specified with `-javaagent` becomes `whatap.home`, so the `whatap.conf` in the job directory is not read. Specify the job directory with `-Dwhatap.home`.

Monitoring data does not appear

Check the following in order.

1. Check that the Batch Host is running.
2. Check that `PCODE` in the startup banner is not 0.
3. Check that TCP 6600 from the Batch Host to the collection server is allowed.
4. Check that UDP 6611 from the Batch Agent to the Batch Host is reachable.
5. If you changed the UDP port, check that the Batch Host's `net_udp_listen_port` matches the batch job's `net_udp_port`.
6. Check whether errors are recorded in `{whatap.home}/logs/whatap.log`.

Steps are cut off midway

A collection limit was reached. No indication is left in the trace when a limit is reached, so check the following.

Check	Related setting
Whether step numbers stop near 1,000	Trace step collection
Whether the SQL count matches `trace_sql_limit`	Collection limits
Whether the HTTP call count matches `trace_httpc_limit`	Collection limits
Whether a `lost-profile` step appears at the end of the trace	Profiling expiration time

A transaction records lost-profile

Profiling was force-terminated because no step arrived for `profile_expired_time` (30 seconds by default).

- If the job terminated abnormally, for example with `kill -9`, this is the expected automatic cleanup.
- If the job was running normally, a period with no instrumented processing, such as waiting for an external system response, may have exceeded 30 seconds. See Profiling expiration time and adjust the value.

Automatically calculated settings

The following values are calculated automatically from `time_limit`. You do not need to write them in the configuration file.

Option	Calculation
`warning_time`	70% of `time_limit`
`yellow_time`	70% of `time_limit`
`red_time`	100% of `time_limit`

Deployment checklist

#	Item	How to check
1	Agent extracted	Jars exist under `{BATCH_HOME}` and in the `agent` directory.
2	Host configured	`license` and `whatap.server.host` are set.
3	Network requirements	TCP 6600 outbound is allowed.
4	Batch Host started	`PCODE` in the Batch Host banner is not 0.
5	Agent configured	`job_name` and `time_limit` are set per job.
6	Job started	The job directory is specified with `-Dwhatap.home`.
7	Configuration applied	A `logs` directory was created in the job directory.
8	Data verified	Transactions with `job_name` appear in the Hitmap.

#	Item	How to check
9	Duration verified	The whole job duration was checked on the transaction without a sequence number.
10	Parallel execution	Multiple jobs run at once appear separated by `job_name`.
11	Shutdown	The Batch Host was stopped after all jobs finished.

Next steps

Open source tracing

Depending on the framework or open source library used in applications, you may need to add agent options. For example, when using spring-boot-3.x, feign-client-11, or okhttp3-4.4 with a framework or open source, set the options as follows:

whatap.conf

```
weaving=spring-boot-3.0,feign-11,okhttp3-4.4
```

For information on how to configure the framework or open source library that suits your application environment, see the following.

Checking the installation

If you have created a project, installed an agent, and applied all JVM options, see the checklist in the following.

Installation troubleshooting

It provides various problems that may occur when installing the agent and specific instructions for resolving them. It provides solutions against various situations, including firewall configuration, SpringBoot and Tomcat JMX configuration, and server configuration using the OSGI framework. For more information, see the following.

Agent setting

It provides various features for monitoring by applying some options to the agent configuration file (whatap.conf). It includes basic configuration, configuration for server connection and data transfer, how to manage the configuration files for multiple application servers, and transaction tracing, and more. For more information, see the following.

Starting the monitoring

After all settings are made, restart the application server. The agent starts collecting data. First, check whether the monitoring data has been collected in **Application Dashboard**. For more information about **Application Dashboard**, see the following.

JVM Options For Older Java Versions

You can apply the WhaTap agent on the versions earlier than Java 1.6. However, some functions are limited.

Java 1.5

Java 1.5 supports the `-javaagent` option. However, there are restrictions on information inquiry for external threads, so some functions are limited.

The following functions do not work.

- Real-time trace on active transactions
- Stack analysis (Top stack, Unique stack, Active stack)
- Detailed stack of the thread list

Java 1.4

Java 1.4 does not have the `-javaagent` option. Install the WhaTap agent differently. Of course, Java 1.4 must be used for the whatap.agent.jar file.

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

Creating boot.jar

Create boot.jar by using the setup.sh file. If [JAVA_HOME] is not entered, it automatically recognizes the current path as the location of JDK.

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

setup.sh execution result: whatap.java14.boot-`X.Y.Z`.jar

Installing in bootclasspath

Using the `bootclasspath` option, add the built jar and tracer jar as prepend.

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

Constraints

In JDK 1.4, the following functions do not work.

- Real-time trace on active transactions
- Stack analysis (Top stack, Unique stack, Active stack)
- Component version
- Thread list, heap history
- Loaded class
- Open socket
- Agent dump

Next steps

Open source tracing

Depending on the framework or open source library used in applications, you may need to add agent options. For example, when using spring-boot-3.x, feign-client-11, or okhttp3-4.4 with a framework or open source, set the options as follows:

whatap.conf

```
weaving=spring-boot-3.0,feign-11,okhttp3-4.4
```

For information on how to configure the framework or open source library that suits your application environment, see the following.

Checking the installation

If you have created a project, installed an agent, and applied all JVM options, see the checklist in the following.

Installation troubleshooting

It provides various problems that may occur when installing the agent and specific instructions for resolving them. It provides solutions against various situations, including firewall configuration, SpringBoot and Tomcat JMX configuration, and server configuration using the OSGI framework. For more information, see the following.

Agent setting

It provides various features for monitoring by applying some options to the agent configuration file (whatap.conf). It includes basic configuration, configuration for server connection and data transfer, how to manage the configuration files for multiple application servers, and transaction tracing, and more. For more information, see the following.

Starting the monitoring

After all settings are made, restart the application server. The agent starts collecting data. First, check whether the monitoring data has been collected in **Application Dashboard**. For more information about **Application Dashboard**, see the following.

Java Helper

Java Helper is a CLI tool that automatically analyzes your application's libraries and generates optimal `weaving` settings. To track frameworks and open source libraries used in Java applications, the WhaTap agent requires `weaving` configuration. Java Helper is included in the agent package as `whatap.javahelper.jar` and requires **JDK 8 or later**.

To manually specify `weaving` options, see Manual Configuration.

Supported Formats

Supports JAR, Spring Boot Fat JAR, Maven Shade JAR, and WAR formats. For EAR, it analyzes up to the root-level JAR modules.

Java Helper Commands

```
java -jar whatap.javahelper.jar conf generate <input> [options]
```

Input

Input	Description	Example
`<JAR path>`	Analyze by specifying a JAR file directly	`/app/myapp.jar`
`--pid <PID>`	Analyze a running Java process by PID	`--pid 12345`

You must specify either a JAR path or `--pid`.

Options

Option	Description	Default
`-o <file>`	Save matching results to a whatap.conf file	Screen output only if not specified
`-v`	Output detailed analysis logs (verbose)	off

Usage Scenarios

New Agent Installation

Specify a JAR file path to generate whatap.conf directly.

```
java -jar whatap.javahelper.jar conf generate /app/myapp.jar -o whatap.conf
```

Generated whatap.conf example:

whatap.conf

```
weaving=spring-boot-3.2,tomcat10,mongodb-4.8

# Recommended modules (uncomment if needed)
#weaving+=,reactor-kafka-1.3
#weaving+=,resilience4j
```

Add your license key and server information to the generated file to complete the installation.

Configuration Check During Operation

Analyze a running process by PID to check matching results.

```
java -jar whatap.javahelper.jar conf generate --pid 12345
```

Output example:

```
========================================
 WhaTap Weaving Module Matching Results
========================================

 Detected libraries: 25
 Matched modules:    8
 Version mismatch:   1
 Auto-included:      3

 LIBRARY              VERSION   WEAVING MODULE       STATUS
 spring-boot          3.2.5     spring-boot-3.2      MATCHED
 mongodb-driver-sync  4.11.1    mongodb-4.8          MATCHED
 lettuce-core         6.3.2     lettuce-6.1          AUTO
 ...
```

Compare with the current whatap.conf to check for missing modules. Add the `-o` option to save as a configuration file.

After Library Upgrade

When libraries change, such as a Spring Boot version upgrade, re-analyze the upgraded JAR to get new `weaving` settings.

```
java -jar whatap.javahelper.jar conf generate /app/myapp-upgraded.jar
```

Matching Status

Status	Description	Action
`MATCHED`	artifactId, groupId, and version all match; weaving module applicable	Auto-applied
`AUTO`	Auto-included module (applied by default without weaving setting)	Auto-applied

Status	Description	Action
VERSION_MISMATCH	Library detected but version is outside weaving module support range	Check for agent update
NO_MATCH	No corresponding module in weaving registry	N/A

Reference

Dependency Analysis Method

The process of identifying library versions from JAR files operates with a 3-level priority. Information extracted at a higher level is not overwritten by lower levels.

1. **Maven pom.properties**: Extracts `groupId`, `artifactId`, and `version` from `META-INF/maven/{groupId}/{artifactId}/pom.properties` inside the JAR. The most accurate method.
2. **MANIFEST.MF**: If pom.properties is unavailable, reads standard attributes such as `Implementation-Title` and `Implementation-Version` from `META-INF/MANIFEST.MF`.
3. **Filename parsing**: If both methods above fail, infers from the JAR filename. (e.g., `jackson-databind-2.14.0.jar` → `version="2.14.0"`)

Weaving Plugin Configuration

Specify `weaving` options directly in the whatap.conf file to track frameworks and open source libraries. To automatically analyze libraries and generate settings, see Auto Configuration (Java Helper).

For example, when using `spring-boot-3.x`, `feign-client-11`, and `okhttp3-4.4`, set the options as follows:

whatap.conf

```
weaving=spring-boot-3.0,feign-11,okhttp3-4.4
```

Trackable Open Source List

For configuration methods for frameworks or open sources being traced through the Java agent, refer to the following:

Apache Camel

CXF

camel-cxf Version	Configuration value	Minimum agent version
3.15 or later	weaving=camel-cxf-3.15	v2.2.27

Netty

camel-netty4 Version	Configuration value	Minimum agent version
2.25 or later	weaving=camel-netty4-2.25	v2.2.42

SEDA

camel-seda Version	Configuration value	Minimum agent version
2.22 or later	weaving=camel-seda-2.22	v2.2.20
3.2 or later	weaving=camel-seda-3.2	v2.2.20

DynamoDB

DynamoDB Version	Configuration value	Minimum agent version
1.11	weaving=dynamodb-1.11	v2.2.39
2.25	weaving=dynamodb-2.25	v2.2.39

Feign Client

Feign Version	Configuration value	Minimum agent version
11 or later	weaving=feign-11	v2.2.6

Hystrix

Hystrix Version	Configuration value	Minimum agent version
1.5 or later	weaving=hystrix-1.5	v2.0_21

Kafka

Kafka Version	Configuration value	Minimum agent version
kafka-clients 2.4.0 or later	weaving=kafka-clients-2.4.0	v2.2.15
reactor-kafka 1.3 or later	weaving=reactor-kafka-1.3	v2.2.5

Logging

Log4j2

Log4j2 Version	Configuration value	Minimum agent version
2.17 or later	weaving=log4j-2.17	v2.2.28

ⓘ Agent option to exclude specific appenders

Logback

Logback Version	Configuration value	Minimum agent version
1.2.8 or later	weaving=logback-1.2.8	v2.2.28

ⓘ Agent option to exclude specific appenders

MongoDB

MongoDB Version	Configuration value	Minimum agent version
3.8.2 or later	weaving=mongodb-3.8	v2.2.11
4.0.3 or later	weaving=mongodb-4.0	v2.2.11
4.4 or later	weaving=mongodb-4.4	v2.2.11
4.8 or later	weaving=mongodb-4.8	v2.2.11
4.11 or later	weaving=mongodb-4.11	v2.2.45
5.0 or later	weaving=mongodb-5.0	v2.2.45

Mule Framework

Mule Version	Configuration value	Minimum agent version
3.9.5 or later	weaving=mule-3.9.5	v2.2.23
4.5 or later	weaving=mule-4.5	v2.2.23

OkHttp

OkHttp Version	Configuration value	Minimum agent version
2.7 or later	weaving=okhttp-2.7	v2.0_15
3.x	weaving=okhttp3	v2.0_15
3.x (4.4 or later)	weaving=okhttp3-4.4	v2.2.9

Quarkus Reactive

Quarkus Reactive Version	Configuration value	Minimum agent version
1.13 or later	weaving=quarkus-reactive-1.13	v2.2.19
2.10 or later	weaving=quarkus-reactive-2.10	v2.2.19

RabbitMQ

reactor-rabbitmq Version	Configuration value	Minimum agent version
1.2 or later	weaving=reactor-rabbitmq-1.2	v2.0_06

Redis

Jedis

Jedis Version	Configuration value	Minimum agent version
2.9.0 or later	weaving=jedis-2.9	v2.2.43
2.9.3 or later	weaving=jedis-2.9	v2.0_33
3.2 or later	weaving=jedis-3.2	v2.0_09

Lettuce

Lettuce Version	Configuration value	Minimum agent version
5.1 or later	weaving=lettuce-5.1	v2.2.7

Lettuce Version	Configuration value	Minimum agent version
6.2 or later	weaving=lettuce-6.2	v2.2.16

Retrofit

Retrofit Version	Configuration value	Minimum agent version
2.5 or later	weaving=retrofit-2.5	v2.2.39

Cannot be used simultaneously with okhttp configuration

Ribbon

Ribbon Version	Configuration value	Minimum agent version
All versions	weaving=ribbon	v2.2.10

Spring Boot

Spring Boot Version	Configuration value	Minimum agent version
2.1 or later	weaving=spring-boot-2.1	v2.2.23
2.5 or later	weaving=spring-boot-2.5	v2.2.9
2.7 or later	weaving=spring-boot-2.7	v2.2.9
3.0 or later	weaving=spring-boot-3.0	v2.2.9
3.2 or later	weaving=spring-boot-3.2	v2.2.38

Spring Boot Version	Configuration value	Minimum agent version
4.0 or later	weaving=spring-boot-4.0	v2.2.69

View included technology stack

- **spring-boot-2.1**: r2dbc-mysql-0.8.2, spring-cloud-gateway-2.1, spring-kafka-2.7, spring-webflux-5.1, tomcat9
- **spring-boot-2.5**: r2dbc-mysql-0.8.2, spring-cloud-gateway-3.0, spring-kafka-2.7, spring-webflux-5.3, mongodb-4.0.3, reactor-kafka-1.3, rxjava2, tomcat9
- **spring-boot-2.7**: jasync-r2dbc-mysql-2.1.23, r2dbc-mysql-0.9.3, spring-cloud-gateway-3.1, spring-kafka-3.0, spring-webflux-5.3, mongodb-4.4, reactor-kafka-1.3, rxjava2, tomcat9
- **spring-boot-3.0**: jasync-r2dbc-mysql-2.1.23, r2dbc-mysql-1.0.2, spring-cloud-gateway-4.0, spring-kafka-3.0, spring-webflux-6.0, mongodb-4.8, redis(lettuce-6.2), reactor-kafka-1.3, rxjava3, tomcat10
- **spring-boot-3.2**: jasync-r2dbc-mysql-2.1.23, r2dbc-mysql-1.1.3, spring-cloud-gateway-4.1, spring-kafka-3.1, spring-webflux-6.1, redis(lettuce-6.2), reactor-kafka-1.3, rxjava3, tomcat10
- **spring-boot-4.0**: RestTemplate, RestClient, WebClient, R2DBC

Tomcat

Tomcat Version	Configuration value	Minimum agent version
9.x	weaving=tomcat9	v2.2.5
10.x	weaving=tomcat10	v2.2.5

Undertow

Undertow Version	Configuration value	Minimum agent version
2.2 or later	weaving=undertow-2.2	v2.2.39

Undertow Version	Configuration value	Minimum agent version
2.3 or later	weaving=undertow-2.3	v2.2.14

Vert.x

Vert.x Version	Configuration value	Minimum agent version
3.5.3	weaving=vertx-3.5	v2.2.42
3.9	weaving=vertx-3.9	v2.2.44
4.5	weaving=vertx-4.5	v2.2.39

Preventing False Positives in Java Agent CVE

Library classes included in the Java agent are only loaded when actually used by the application. Even if libraries are detected during CVE vulnerability scans, there are no actual vulnerabilities if they are not being used.

To remove unnecessary tracing classes, execute the following command:

```
java -cp whatap.agent-X.Y.Z.jar whatap.agent.setup.RemoveWeaving -remove [weaving jar filename]
```

You must perform the deletion operation again with each Java agent update.

The command to remove specific library classes is supported in **Java agent version 2.2.37** or later.

Usage:

```
java -cp whatap.agent-X.Y.Z.jar whatap.agent.setup.RemoveWeaving [arguments] [weaving jar filename]
```

Example

```
java -cp whatap.agent-2.2.37.jar whatap.agent.setup.RemoveWeaving -remove spring-boot-2.5.jar spring-boot-2.7.jar
```

Arguments:

- `-r` or `-remove` : Deletes weaving jar files from the Java agent and creates a new Java agent.
- `-l` or `-list` : Prints the list of weaving jar files.
- `-d` or `-debug` : Enables debug logging.
- `-e` or `-error` : Displays full stack trace information for errors.
- `-h` or `-help` : Prints help and exits.

Java Installation Check

The items must be checked after installing the agent. If any problem occurs during installation, see the following.

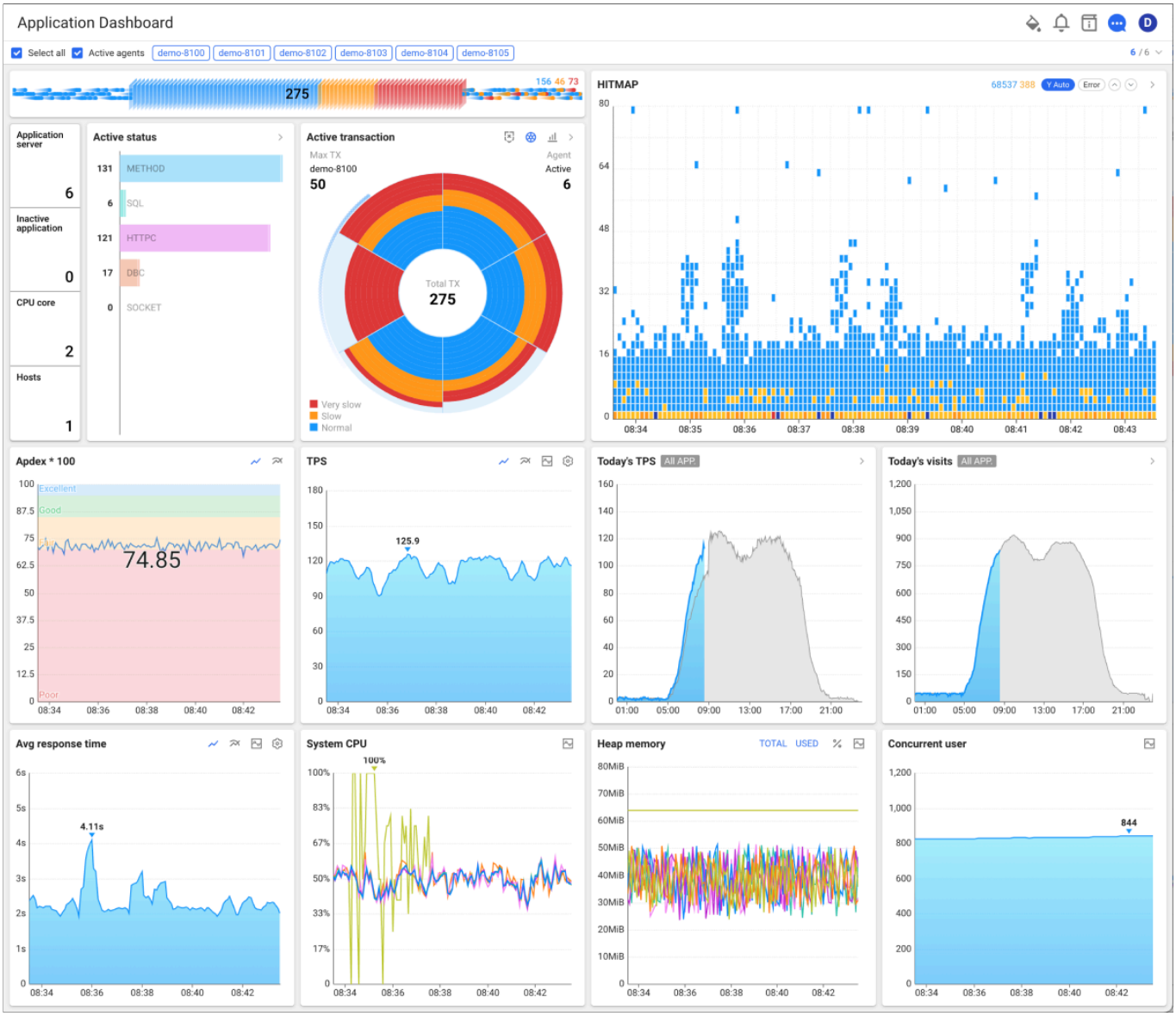
Restarting WAS

Restart the application server. If the following WhaTap character log appears in a running log, it indicates that the `JAVA_OPTS` setting has been completed.

```
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
_       ____     _____
| | /| / / /  ___ /_  __/__ ____
| |/ |/ / / _ \/ _ `// / / _ `/ _ \
|__/|__/_//_/\_,_//_/  \_,_/ .__/
                          /_/
Just Tap, Always Monitoring
WhaTap Agent version 0.3.9 20161115
```

Checking the agent connection

Go to Whatap Monitoring Service. After logging in, select a created project from the project list. If the data of the server connected by the **Application Dashboard** menu appears through the widget, all settings are completed.



ⓘ The default agent name is `{type}-{ip2}-{ip3}-{port}` . To change the agent name or add properties, see the following.

Java Agent Troubleshooting

Firewall configuration checking

If the TCP outbound policy for the WhaTap data collection server has not been set, monitoring data cannot be transferred. **Deselect** the **firewall blocking**.

Use the `telnet` command to check "Connected to `IP`."

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

> ⓘ For the connection server details, select a desired project in WhaTap Monitoring Service, and then check in Management > Agent Installation.

SpringBoot 2.2 or later, Tomcat JMX off

If the embedded Tomcat is used in SpringBoot 2.2 or later, the JMX functions do not work with the default settings. In this case, activate the JMX functions. Please refer to the related document.

application.properties

```
spring.jmx.enabled=true
server.tomcat.mbeanregistry.enabled=true # In case of tomcat embedded #
```

application.yml

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

```
tomcat:
  mbeanregistry:
    enabled: true #tomcat embedded
```

In case of the application server using the OSGI framework

Register any agent package prefix (whatap) in the JVM options for an OSGI framework-based application server.

The application server is available in Jboss EAP 6.0 or later, Jboss AS 7.0 or later, Wildfly 8.0 or later, IBM WebSphere AS 7.0 or later, and such.

JBoss AS, Wildfly, JBoss EAP 6.0 or later

Register a prefix in the following file:

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

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

JBOSS EAP 7.0

WebSphere

As an JVM option, add the following.

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

- If "*" is specified as the default value, no additional setting is required.
- For the setting location, see WebSphere.

Add the security.policy role as follows:

$WEBSPHERE_HOME/properties/server.policy || $WEBSPHERE_PROFILE_HOME/properties/server.policy

```
grant codeBase "file:$WHATAP_HOME/-"
{
```

```
  permission java.security.AllPermission;
};
```

Errors of Log Manager

If any Log Manager error occurs in JBoss AS 7.0 or later and JBoss EAP 6.0 or later, add the JVM options.

- Set the package name of Log Manager in `-Djava.util.logging.manager`.
- Set the JBoss Log Manager JAR file in `-Xbootclassloader`.

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

```
# Specify the exact Java VM executable to use.
#
if [ "x$JBOSS_MODULES_SYSTEM_PKGS" = "x" ]; then
   JBOSS_MODULES_SYSTEM_PKGS="org.jboss.byteman,org.jboss.logmanager,whatap"
fi

...
# Specify options to pass to the Java VM.
#
if [ "x$JAVA_OPTS" = "x" ]; then
   JAVA_OPTS="-Xms1303m -Xmx1303m -Djava.net.preferIPv4Stack=true"
   JAVA_OPTS="$JAVA_OPTS -Djboss.modules.system.pkgs=$JBOSS_MODULES_SYSTEM_PKGS -Djava.awt.headless=true"
   JAVA_OPTS="$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"
else
   echo "JAVA_OPTS already set in environment; overriding default settings with values: $JAVA_OPTS"
fi
```

In case the MBeanServerBuilder error occurs

If any MBeanServerBuilder-related error is output in JBoss 5.0 or earlier, add the JVM options.

(JBoss_Path)/bin/run.conf

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

In case of occurrence of a permission error

If any Java security policy-related error occurs, add the role setting in $JAVA_HOME/jre/lib/security/java.policy.

To apply all roles collectively, add the setting in java.policy as follows:

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

```
grant {
  permission java.security.AllPermission;
};
```

In case of occurrence of the java.io.FilePermission error

```
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)
```

Add the setting in java.policy.

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

```
grant {
  ...
  permission java.io.FilePermission {package path seen in the error message}, "read"
```

```
};
```

In case of occurrence of the java.util.PropertyPermission error

Add the setting in java.policy.

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

```
grant {
  ...
  permission java.util.PropertyPermission {package path see in the error message}, "read"
};
```

In case the sigar library cannot be loaded

Check whether the execution role for $WHATAP_HOME/lib1/*.so exists. If no permission, give the execution role.

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

If under $WHATAP_HOME/lib1 of AIX 7, libsigar-ppc64-aix-7.so does not exist and the error occurs, copy the file.

```
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
```

In case CPU metrics are displayed as -1

When running IBM Java in the environment with the latest version of the Java agent, the CPU metrics may

sometimes be displayed as **-1**. This issue occurs because the **System Load Average** value is not supported on specific platforms.

The Java agent basically uses the Java Management Extensions (JMX) to collect metrics. It uses the `getSystemLoadAverage()` method of the `OperatingSystemMXBean` interface. This method returns the average load of the system over the last minute depending on the operating system. However, some platforms do not support this value and may return **-1**.

> The `getSystemLoadAverage()` method can be checked in the OpenJDK source code. For example, the source code for the OpenJDK 7u version can be seen in the GitHub repository.

Recommendation: Configure metric collection via built-in libraries.

To resolve the CPU metrics displayed as **-1**, enable the built-in library (OSHI or SIGAR) via the Java agent settings.

- **Java 8 or later**: Use the **OSHI** library.

  You can use the **OSHI** library to reliably collect operating system level metrics. See the following how to configure the agent.

  whatap.conf

  ```
  # default false
  oshi_enabled=true
  ```

- **Below Java 8**: Use the **SIGAR** library.

  For environments below Java 8, you can collect metrics via the **SIGAR** library. For more information on how to configure the agent, see the following.

  whatap.conf

  ```
  # default false
  sigar_enabled=true
  ```

> The **OSHI** (`oshi_enabled`) and **SIGAR** (`sigar_enabled`) options are disabled by default. They must be explicitly set if needed.

- Be careful not to enable both settings at the same time. Select a library that best suits your environment and then configure it.

Java Agent Configuration

The Whatap agent writes the necessary settings for each agent in whatap.conf. The agent locates and loads the configuration file through the environment variables.

If any separate setting, the agent is set to find whatap.conf under the (whatap.agent-X.Y.Z.jar) directory. It checks whether whatap.conf has been changed every 5 seconds and then reloads the file.

For user convenience, Whatap Monitoring Service also provides the agent configuration function.

Default settings

whatap.conf

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

- `whatap.home` : Enter the path of whatap.agent.jar as the default.
- `whatap.config` : It is recommended to enter only the file name. You can also specify the relative path.

In order for the agent to collect and transfer data to the server, the access key and server address must have been set.

whatap.conf

```
license=x220g160hgd29-x3qpff0garcus7-z4p0kao58un11a
whatap.server.host=10.10.0.1
```

When the WhaTap agent is run for the first time, you can check which jar file has been used as the agent and from which path whatap.conf has been loaded, as the following.



Server connection and data transmission

The WhaTap agent is connected to the server address that has been set in whatap.conf.

whatap.conf

```
whatap.server.host=10.10.1.1
whatap.server.port=6600 # default value
```

- When the TCP session is connected, it receives the communication key by using the `license`. When a wrong `license` is set, the server terminates the session.
- If the TCP connection is terminated repeatedly, check for any firewall problem or check whether the `license` value is correct.
- When a TCP session is connected, data is transferred through the secure communication channel based on the private key received from the server.

Management of the configuration file for multiple applications

When running multiple applications on one server, to separately control the configuration files for each application, change the name of whatap.conf and then add the JVM options to your startup script.

Service A

```
-Dwhatap.config=whatap_Aservice.conf
```

Service B

```
-Dwhatap.config=whatap_Bservice.conf
```

> The whatap.conf configuration file must be located in the `${WHATAP_HOME}` path.

Configuring the agent on the service screen

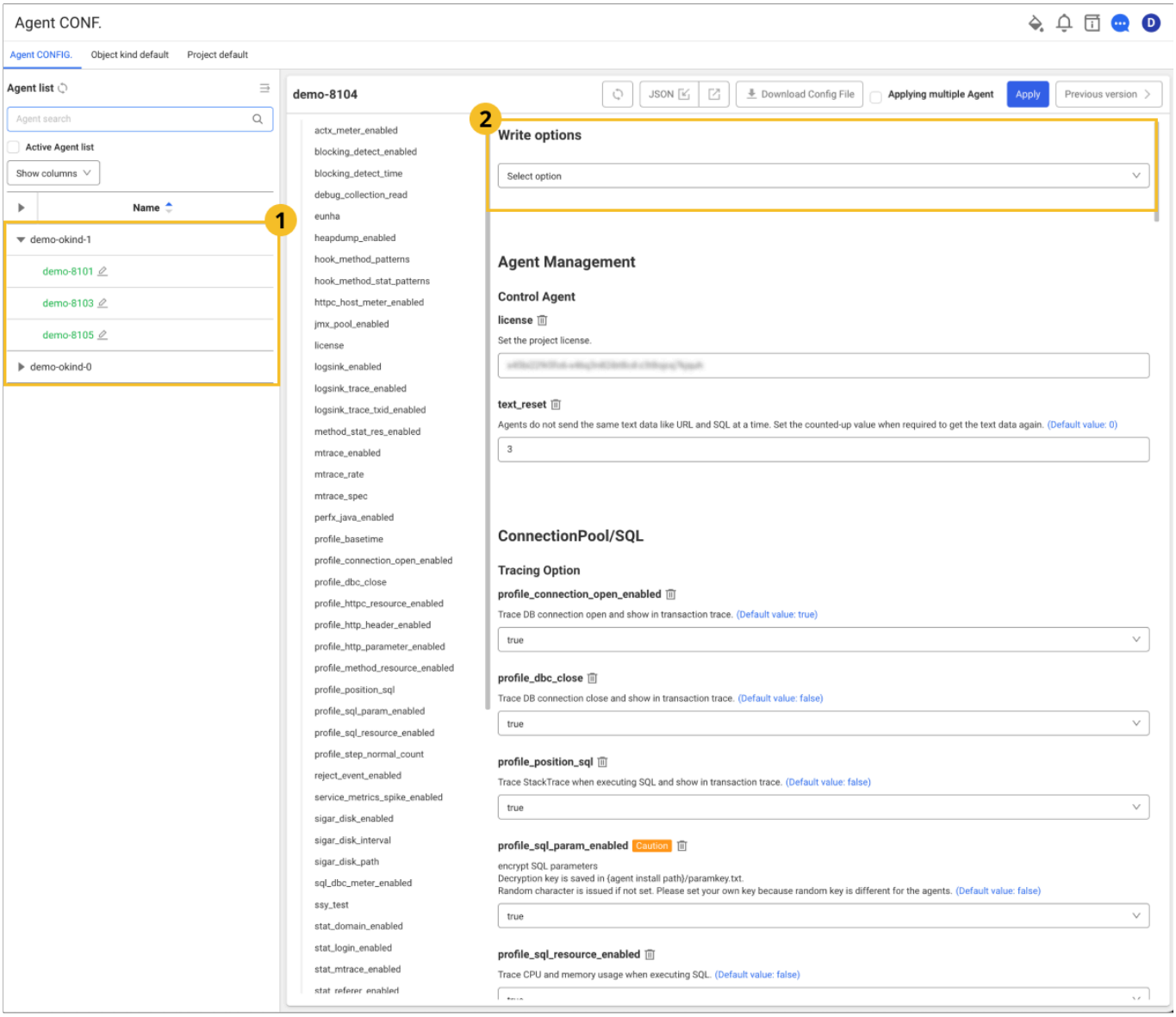
Home > Select Project > **Management** > **Agent CONF.**

You can add, modify, or delete agent configuration options in the Whatap Monitoring Service without directly modifying the whatap.conf file that is in the monitoring target server. You can also click **Download configuration file** to download the whatap.conf file.

- This function is only available to the members with **Edit** role. The members without **Edit** role can only view the settings.
- You can set the options in the following format.
  - Select `true` or `false` for a boolean value.
  - For numeric values, you can enter only number.
  - When entering or modifying a text (string) value, check the option description in detail.

- You cannot select any option that is not editable. (e.g. `license`)
- You may restart each agent depending on the option that has been added, modified, or deleted.
- The option keys that can be applied may differ depending on the application type and agent version.

Adding options



1. In **Agent List**, select the 1 agent to add options to.

2. In 2 **Write options**, select options to add.
   - You can find options to add in **Search**. Enter text to filter matching options.
   - If you select **Direct Input**, you can enter the option keys and values.
3. Check the description and default value of the selected option key, and then enter the setting.

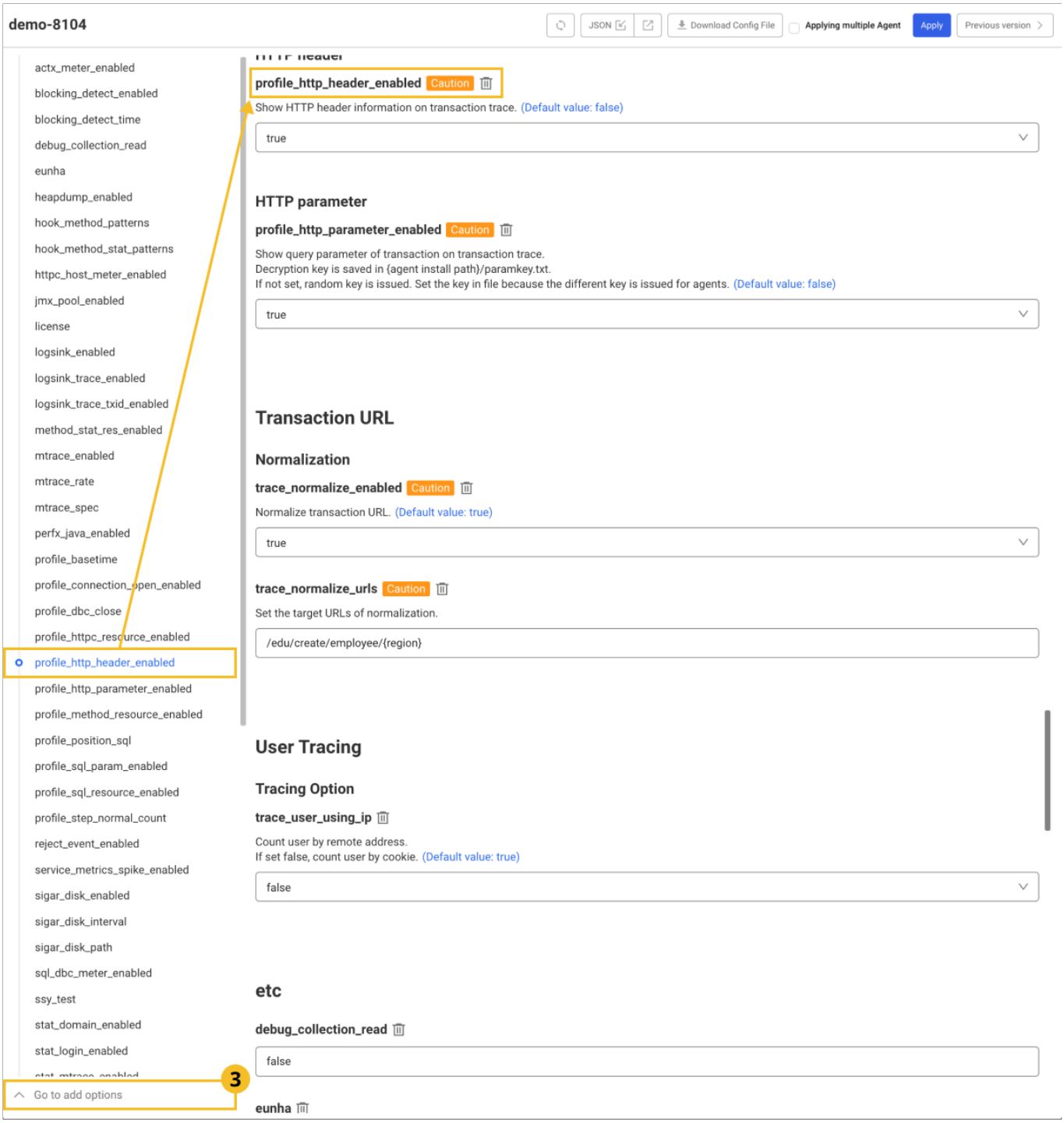


   - To cancel the selected option, select 🗑.
   - To add new options, select **Add** and then repeat Step 2.
4. After adding the desired options, select **Apply** on the upper right of the screen.

Apply the selected options and settings to agents.

- To initialize the content in modification, select ↻ on the upper right of the screen.
- If you select **Apply** without entering options, the options are deleted.
- The added options cannot be selected in the option list.
- The option keys that can be applied may differ depending on the application type and agent version.
- The example images may differ depending on the application type and agent version.

Modifying or deleting options



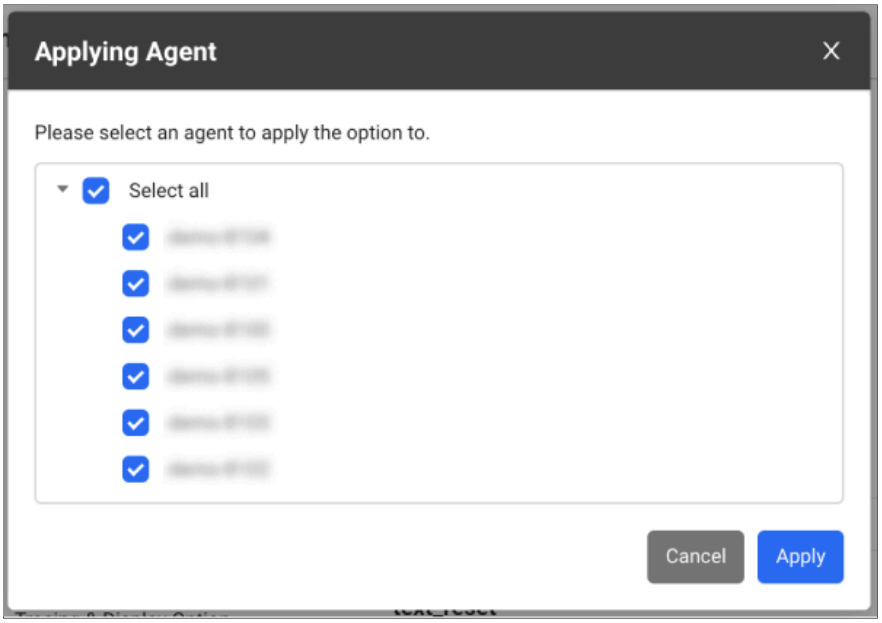
1. Scroll up or down the screen, or select a desired option to edit or delete from the option list on the left.
2. Select or edit the desired value for the option to change. To delete an option, select 🗑.
3. Select **Apply** to apply your changes.

> • If you want to move to the top of the screen, select 3 **Go to add options**.
> • If you select **Apply** without entering options, the options are deleted.

Applying to multiple agents simultaneously

Changed options can be applied simultaneously to multiple agents that belong to a project.

1. If you check **Applying multiple Agent** on the upper right of the screen, checkboxes are created for each option.
2. Select the checkboxes of the options you want to apply simultaneously. You can select multiple items.
3. On the upper right of the screen, select **Apply**.
4. If the **Applying Agent** window appears, select an agent to apply the changed options to. To select all, select **Select all**.



5. Select **Apply**.

Setting the default values of the agent

To avoid the hassle of repeatedly applying the settings when adding a new agent to a project, you can create and apply default settings for each project and task.



- **Object kind default**: In the whatap.conf file, the options can be commonly applied to the agents classified as `whatap.okind`, and the applied options can be checked.
- **Project default**: The options can be commonly applied to all agents belonging to the project, and the applied options can be checked.

> ⓘ
> - If you select **Object kind default** or **Project default** tab and then apply the options, it is useful to commonly manage agents.
> - The application order of options is as follows: **Object kind default**, **Project default**, and **Agent CONFIG.**.
>   - If the value of option b is 1000 in **Project default** and it is 2000 in **Object kind default**, the option value set in **Object kind default**, 2000 takes precedence.
>   - If the value of option b is null in **Project default** and it is 2000 in **Object kind default**, the option value set in **Object kind default** takes precedence.

Sharing the agent settings

Agent settings can be saved as a json file, and the file can be loaded and applied to other agents.

1. In the **Agent list**, select an agent to export the settings to a json file.
2. On the upper right of the screen, select ↗.
3. Download the json file.

4. In **Agent list**, select other agents.
5. On the upper right of the screen, select .

Check the agent settings fetched from the json file.

Guide to the agent configuration options

The following links provide descriptions of the options that can be set.

Agent Name

Set a Java agent name to identify monitoring targets.

Function Control

Enable or disable Java agent functions via configuration options.

Resource Usage

Check system resources used by the Java agent.

Network Configuration

Configure network communication between the Java agent and collection server.

Performance

Optimize Java agent overhead by configuring performance-related options.

Log Configuration

Configure log file location and log level for the Java agent.

Transaction

Configure Java transaction tracking and profiling options.

DB/SQL

Configure database connection and SQL tracking options in the Java agent.

HTTPC/API Call

Configure external HTTP call and API integration tracking options in the Java agent.

Script Plugin

Extend Java agent monitoring with user-defined scripts.

Number of Users

Configure concurrent user counting methods for the Java agent.

Load Control

Configure throttling and load control options in the Java agent.

Notification

Configure notification options for events occurring in the Java agent.

Apdex

Configure Apdex score options in the Java agent.

Statistics

Configure statistics data collection for the Java agent.

Error Stack

Configure stack trace collection when Java transaction errors occur.

Additional Options

Reference for additional Java agent configuration options.

Async Tracking

Configure transaction linking and tracking for Java async calls.

Agent Name Identification

WhaTap basically uses the data collected from application servers to identify each server while collecting monitoring data. Basic information uses the application server as a unique identifier by a combination of application server's type, IP address, and such. If necessary, use the name set by the user or change the pattern to a unique value. The agent name must be unique.

The reason why the data extracted from the application server is used, is like the following. When a communication loss caused by the stop of the application server, network disconnection, or agent problem between the collection server and the agent, is restored, the continuity of the data from the reconnected agent is maintained.

- The agent names must be unique on a project basis.
- If the agent ID and name have been changed, previous data is not connected.

- In WhaTap Monitoring, the object and agent can be used interchangeably as the monitoring target.

How to determine the default name

If there is no setting for `whatap.oname` as an JVM option, agent names are automatically assigned in the following priority.

1. `whatap.type` is identified.
2. When `whatap.type` is a servlet container, the port used by the application is identified.
3. If the port cannot be identified, the process ID (PID) is retrieved.
4. By adding the third and fourth IPv4 addresses to the identified information, the final name is determined.

For example, if **Tomcat** is started with the port **8080** on a server whose IP address is 10.11.**12.13**, the agent name is automatically assigned with `TC-12-13-8080`.

Variable of the agent name

Available name variables are as follows:

Variable	Description
String	The entered string is used.
{type}	The container type is automatically identified and assigned. Targets to be identified automatically: JAVA, TOMCAT, JBOSS, RESIN, SPRINGBOOT, JETTY, JEUS, WEBLOGIC, WEBSPHERE, UNDERTOW, PLAY
{ip0}	The first byte of the IPv4 address is used. (e.g. **10** in **10**.11.12.13)
{ip1}	The second byte of the IPv4 address is used. (e.g. **11** in 10.**11**.12.13)
{ip2}	The third byte of the IPv4 address is used. (e.g. **12** in 10.11.**12**.13)
{ip3}	The fourth byte of the IPv4 address is used. (e.g. **13** in 10.11.12.**13**)
{pid}	The process ID (PID) of Java is used.
{port}	The port that the application listens to is used.

Setting the agent name

Instead of the name that has been automatically assigned, you can enter the name by adding the JVM option into the startup script.

Option	Setting location	Description
-Dwhatap.name	JVM Options	Set the agent name as a pattern. default: {type}-{ip2}-{ip3}-{port}

Option	Setting location	Description
-Dwhatap.oname	JVM Options	Set the agent name to a fixed value. If the -Dwhatap.oname=java-sales-01 option is not set, the pattern set in the -Dwhatap.name option is applied.

In the environment where the IP address is 10.11.12.13 and the port number is 8000,

- **In case the -Dwhatap.oname option** is not applied: **whatap.oname** defaults to JAVA-12-13-8000 according to the pattern set in -Dwhatap.name .
- **In case the -Dwhatap.oname option is applied**: The -Dwhatap.name option is ignored and the agent name is applied as the value set in the -Dwhatap.oname option.

- In the scale out environment of the cloud, VM can be copied and used. Thus, it is important that object names are automatically assigned.
- If two VPC environments have been created and the network has been set to use the same private IP, the two VPCs must not be duplicated. For example, it can be separated as another project or the -Dwhatap.name option can be set by combining the name variables in different ways.

The following options can be applied as an agent name to whatap.conf.

- **whatap.name** String

  Default {type}-{ip2}-{ip3}-{port}

  Set the agent name with patterns.

- **auto_oname_enabled** Boolean

  Default false

  The function to automatically obtain the agent name (whatap.oname) from the server is activated. Upon application, the options for -Dwhatap.name and -Dwhatap.oname are ignored. It obtains whatap.oname through communication with the collection server and then starts the agent's operation.

- **auto_oname_prefix** String

Default `agent`

When an agent name is automatically granted from the server, the prefix of the agent name (business name as usual) is used. The prefix + serial number (from 1) is assigned.

> The `auto_oname_enabled` option that automatically assigns agent names, must be enabled (`true`) to use the option.

- **auto_oname_reset** Int

  Default `0`

  Modify it to obtain a new agent name from the server. If the agent name is automatically granted, the system environment variable, `whatap.oname` is set. Once set, the system environment variables are not changed until the Java instance is restarted. For a reset, modify the value of `auto_oname_reset` . If you change it to a value different from the current setting, it is to be applied.

- **whatap.okind** String

  You can configure monitoring units by grouping multiple agents into "type-based" groups. This option allows you to monitor and manage multiple agents in groups, analyze performance metrics, and set the alerts.

  > You can check it as a group option classified as **By kind** in the **Dashboard** and **Analysis**, **Statistics** menus.

- **whatap.onode** String

  You can configure monitoring units by grouping multiple agents into "node-based" groups. This option allows you to monitor and manage multiple agents in groups, analyze performance metrics, and set the alerts.

  > You can check it as a group option classified for nodes in the **Dashboard** and **Analysis**, **Statistics** menus.

Agent Function Control

The following explains how to fine-tune the agent functionality for monitoring Java application servers. It includes the functions such as enabling/disabling agents, tracing transactions and performance counters, and collecting OS data. You can see various setting options that control each function.

- **shutdown** Boolean

  Default `false`

  If it is `true`, all actions of the agent are stopped and the connection with the server is finished.

- **enabled** Boolean

  Default `true`

  All functions are enabled. Even when the value is `false`, the minimum data to maintain communication with the server is sent.

- **transaction_enabled** Boolean

  Default `true`

  It enables the transaction tracing. It corresponds to transaction data recorded in the hitmap. If the `enabled` option is `false`, this is disabled.

- **counter_enabled** Boolean

  Default `true`

  The performance counter tracing is enabled. It includes the number of active transactions and users, JVM usage, process CPU usage, and DB pool usage. If the `enabled` option is `false`, this is disabled.

- **stat_enabled** Boolean

  Default `true`

  The statistical data tracing is enabled. The statistical data includes transactions, SQLs, HTTP calls, user agents, and client IPs that are collected every 5 minutes. If the `enabled` option is `false`, this is disabled.

- **sigar_enabled** Boolean

  Default `false`

  The OS data collection through the sigar library is enabled. The OS resource data includes CPU, memory, and

disk that are collected every 5 minutes through the sigar library. If the `enabled` option is `false`, this is disabled.

> ◦ In earlier than whatap.agent.2.1.0, the default value is `true`. In whatap.agent.2.1.0 or later, the default value is `false`.
>
> ◦ In Java 1.6 or earlier, you can enable the sigar library and collect performance metrics by adding the `sigar_enabled=true` option.

- **active_stack_enabled** Boolean

  Default `true`

  Active stack tracing is enabled. The tracing data includes Top Stack, Unique Stack, and Active Stack in the Stack menus. If the value of `enabled` or `counter_enabled` is `false`, this is disabled.

- **license** String

  To install the agent, set the access key granted from the server. The access key includes the agent's project and the encryption key for secured communication.

- **cypher_level** aes_bit [128, 256]

  Default `128`

  Set the cypher level for the AES security algorithm. It can be expanded up to 256 bits. For more information about the AES 256 encryption configuration, see the following.

- **encrypt_level** encrypt_level [1, 2, 3]

  Default `2`

  The WhaTap agent encrypts data selectively depending on the attributes when transferring data to the server. It has performance benefits along with high security. To apply encryption policies in bulk regardless of the data type, see the following options.

  ◦ `1`: Encrypted transmission disabled

  ◦ `2`: Encrypted transmission for sensitive attributes such as SQL parameters and plain texts.

  ◦ `3`: Encrypted transmission for all items

- **stat_ip_enabled** Boolean

  Default `true`

Among the side menus of the project, the **Statistics** > **Client IP** menu enables whether to use the IP statistics.

> ⓘ See the following.

- **stat_ipurl_enabled** Boolean

  Default `false`

  Among the side menus of the project, the **Statistics** > **IP Transaction** menu sets whether or not to collect the statistics data.

  > ⓘ See the following.

- **realtime_user_enabled** Boolean

  Default `true`

  Among the side menus of the project, the **Dashboard** >**Concurrent user** menu can set whether to collect the **Application Dashboard** out of the metrics that can be checked.

- **hook_direct_patch_classes** ClassFile_FullPath

  It is used to directly replace a specific class at the loading time. After compiling the class, it creates a separate file and sets the full path to that file.

- **active_stack_second** Seconds

  Default `10`

  Set the interval for tracing the active stacks.

  > ⓘ It is recommended not to change the given default value.

- **boot_redefine_size** Int

  Default `100`

  If the agent has been installed in the Attach or Watcher method, BCI is newly performed to trace the already loaded classes. At this time, it sets the number of classes to redefine at the same time.

- **trace_component_enabled** Boolean

Default `true`

Among the side menus of the project, the **Instance Performance Management** > **Library Version** menu is enabled.

- **realtime_user_thinktime_max / wclient_thinktime_max** MiliSeconds

  Default `300000`

  Set the maximum call interval credited by the same user when measuring the number of real-time browser users.

  > In case of Java Agent 2.2.0 or later, use the `wclient_thinktime_max` option.

- **time_sync_interval_ms** MiliSeconds

  Default `300000`

  Set the synchronization time interval between the agent and the server. For no synchronization, set it to `0`.

- **detect_deadlock_enabled** Boolean

  Default `false`

  It generates an event when any Java thread is deadlocked. The occurrence interval is 5 seconds. Events for the same deadlock occur once per hour.

- **text_reset** Int

  Default `0`

  The WhaTap agent does not transmit the text data once sent until the next day again because it is hashed. If a value is set different from the previous setting, the data is sent again.

  > Transaction URLs and SQL strings correspond to text data.

Agent Resource Usage

The following explains how the Java agent collects CPU, memory, and disk usage. Java's JMX library is basically used. If necessary, you can see how to collect performance metrics through Linux's proc directory or oshi and sigar libraries.

- **linux_proc_stat_enabled** Boolean

  Default `false`

  For CPU, memory, and disk usages, set whether or not to collect from the Linux's proc directory.

- **oshi_enabled** Boolean Java Agent v2.1.0 or later

  Default `false`

  When measuring the CPU, memory, and disk usages, set whether or not to use the OSHI libraries.

  > It is recommended to use it in the environment with Java 8 or later.

- **oshi_netstat_enabled** Boolean Java Agent v2.2.31 or later

  Default `false`

  Set whether or not to collect netstat metrics. The Maximum Transmission Unit (MTU) metric is added.

- **sigar_enabled** Boolean

  Default `false`

  The OS data collection through the sigar library is enabled. The OS resource data includes CPU, memory, and disk that are collected every 5 minutes through the sigar library. If the `enabled` option is `false`, this is disabled.

  > - In earlier than whatap.agent.2.1.0, the default value is `true`. In whatap.agent.2.1.0 or later, the default value is `false`.
  > - In Java 8 or earlier or Java 1.6 or earlier, you can enable the sigar library and collect performance metrics by adding the `sigar_enabled=true` option.
  > - Depending on the application environment, you may see CPU metrics displayed as **-1**. You can

check the recommendations provided by WhaTap and then resolve the issues. See the following.

Agent Network Configuration

The following explains how to configure the network to transmit the data collected by agents from the application servers to the server. The major configuration options are the collection server host, port, timeout value, network transfer size, and data transfer queue size, which are important to ensure the agent's efficient data management and reliable server communication.

- **whatap_server_host** ip_address

  Default `127.0.0.1,127.0.0.1`

  Specify the server to transfer the data collected by the agent. If there are multiple IP addresses for collection server redundancy, use comma (,) as the delimiter. The proxy of the collection server must be in listening state for the set IP address.

- **whatap_server_port** tcp_port

  Default `6600`

  Set the collection server port. Because only one port can be set, the collection servers set in the `whatap_server_host` option must use the same port.

- **tcp_so_timeout** MiliSecond

  Default `60000`

  Set the socket time out of the TCP session that communicates with the collection server.

- **tcp_connection_timeout** MiliSecond

  Default `5000`

  Set the connection time out of the TCP session that communicates with the collection server.

- **net_send_max_bytes** Byte

  Default `5242880`

  Maximum data size in bytes the agent can collect and transfer over the network at a time.

- **net_send_queue1_size** Int

  Default `256`

  Set the size of the queue for data transfer excluding the trasaction traces and active stacks. Set it to 10, when

the set size is 10 or less.

- **net_send_queue2_size** Int

  Default `512`

  Set the size of the queue to be used for transfer of transaction traces and active stack data. Set it to 10, when the set size is 10 or less.

Agent Performance Configuration

The following guides you to the performance options for Java agents. You can check the maximum number of transactions that can be processed simultaneously and the initial size of the buffer that stores transaction data. This plays an important role in optimizing the system load and resource usage.

Below whatap.agent.2.1.0

- **whatap.tx_max_count** Int

  Default `5000`

  Maximum number of transactions to store. If there are numerous concurrent transactions, increase this size.

- **whatap.tx_default_capacity** Int

  Default `101`

  Initial size of the buffer in which transactions are stored.

Agent Log Configuration

The following explains how to manage log data generated in the Java application environment through the agent. It includes the configuration of the path and name of the log file, retention period setting, log monitoring, and how to enable custom log tracing. It also provides various options for log management to help manage system logs efficiently.

Setting the path and name

The path and name of the agent log can be set in whatap.conf. The default value of `log_root` is the `${WHATAP_HOME}/logs` path.

whatap.conf

```
log_root=whatap/logs
log_name=whatap
```

Log management options

By writing the log-related options in whatap.conf, you can manage the log content.

whatap.conf

```
watchlog_enabled=true
log_rotation_enabled=true
log_oname_enabled=true
log_keep_days=7
```

- **watchlog_enabled** Boolean

  Default `false`

  If the value is `true`, the log monitoring is enabled.

- **watchlog_check_interval** Int

Default `2000`

It is the interval for monitoring the logs. At every interval, it checks whether any record has been added to the log file.

- **watchlog_read_count** Int

  Default `8`

  It is the maximum count to read logs. It reads as many as `watchlog_buffer_size` at a time.

- **watchlog_buffer_size** Int

  Default `131072(128k)`

  Size for reading logs at a time.

- **watchlog.LOGKEY**

  This is the setting to monitor a log file. Enter the `file` name and set multiple `words` with the delimiter, comma (,). It watches the logs at `check_interval` and alerts you when any keywords are found. Once an alert has occurred, the occurrence of alert stops for the `silent` period.

  - **watchlog.LOGKEY.enabled** Boolean Default `true`
  - **watchlog.LOGKEY.file** String
  - **watchlog.LOGKEY.words** String
  - **watchlog.LOGKEY.silent** Int Default `10000`
  - **watchlog.LOGKEY.check_interval** Int Default `1000`

- **log_root** String

  Default `${WHATAP_HOME}` /logs

  Set the agent's log path.

- **log_datasource_lookup_enabled** Boolean

  Default `true`

  The function to record the log is activated in case of DataSource upon InitialContext Lookup.

- **log_rotation_enabled** Boolean

  Default `true`

  The function to store agent log files by date is activated. The log file name is stored in the format of whatap-

'yyyymmdd'.log. If the value is `false`, the log file name is stored in the format of whatap.log.

- **log_keep_days** Int

  Default `7`

  It sets the log file retention period. It works only when the `log_rotation_enabled` value is `true`.

- **log_oname_enabled** Boolean

  Default `false`

  If the value is `true`, add agent names for each log.

```
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.
```

LogSink

In whatap.agent-2.1.0 or later, the LogSink function is provided for integrated monitoring of the application server.

> The LogSink option is available in whatap.agent-2.1.0 or later.

Enabling the log monitoring function

- **logsink_enabled** Boolean

  Default `false`

  It enables the log monitoring function.

- **logsink_trace_enabled** Boolean

  Default `false`

Set whether or not to expose the log tab of the transaction trace by inserting the transaction ID into the logs.

Collecting framework logs

To monitor logs collected from the framework, add the setting and plugin to intercept logs from the module. Set the default category name for logs collected in the framework to `AppLog`. The default value is the same as in the `hooklog_enabled` option.

- **hooklog_enabled** Boolean

  Default `logsink_enabled`

  Set whether or not to enable the log tracing.

  > ◦ In Java Agent 2.2.4 or later, the changed options can be applied without restarting the application.
  >
  > ◦ If the `hooklog_enabled` value has been changed, restarting is required for Java agent versions less than 2.2.4. The default value of `hooklog_enabled` is `logsink_enabled`. Accordingly, if `logsink_enabled=true` is set and the Java application is restarted, `hooklog_enabled` is set to `true`.

- **hooklog_custom_methods** String

  Register a custom log. An arbitrary log framework content is delivered. Use this to trace logs of the log module created on the website.

  Java

```java
package io.home.test;

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

  whatap.conf

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

Setting to exclude a specific log appender

It provides agent options to exclude specific log appenders. The following agent options are provided to exclude specific appenders from monitoring.

- **ignore_logback_appender_set** String Java Agent v2.2.28 or later

  It is the agent option to exclude specific appenders when tracing logback-1.2.8. For example, to exclude `CONSOLE` and `ACCESS_APPENDER`, see the following example.

  whatap.conf

```
weaving=logback-1.2.8
# example: ignore 'CONSOLE', 'ACCESS_APPENDER' appender
ignore_logback_appender_set=CONSOLE,ACCESS_APPENDER
```

  - If you apply this agent option, you must additionally apply the `weaving=logback-1.2.8` option.
  - `ACCESS_APPENDER` is an example value. Specify an appender name that matches your environment.

- **ignore_log4j_appender_set** String Java Agent v2.2.28 or later

  It is the agent option to exclude specific appenders when tracing log4j-2.17. For example, to exclude `CONSOLE` and `ACCESS_APPENDER`, see the following example.

  whatap.conf

```
weaving=log4j-2.17
# example: ignore 'CONSOLE', 'ACCESS_APPENDER' appender
ignore_log4j_appender_set=CONSOLE,ACCESS_APPENDER
```

  - When applying this agent option, you must additionally apply the `weaving=log4j-2.17` option.
  - `ACCESS_APPENDER` is an example value. Specify an appender name that matches your environment.

Agent Transaction Configuration

A transaction is a server side Logical Unit of Work (LUW) that processes a request from a user's browser. In a web application, the entire process of receiving a single HTTP request and returning a response constitutes a transaction.

Application performance can be evaluated through transaction processing performance. Transaction performance includes information such as response time from transaction start to completion, resource usage, and caller attributes.

Transaction performance can be analyzed based on response time distributions and statistical data.

Transaction Tracing

The following agent options allow you to trace the execution paths of each transaction within applications in detail. You can adjust various aspects, from the minimum response time to the number of execution steps for transactions.

- **profile_basetime** MiliSecond

  Default `500`

  Set the minimum response time to collect trace details. The trace data is not collected, if a transaction is terminated within the set value. The first URL called in every 5 minutes and the erroneous profile data are collected.

  > ⓘ To collect errors regardless of the `profile_basetime` option, set the `profile_concern_error_enabled` option to `true`. If set to the default value of `false`, errors are sampled.

- **profile_step_normal_count** Int

  Default `1000`

  Set the maximum step count for tracing transactions.

- **profile_step_heavy_count** Int

  Default `1020`

In case of heavy steps, even if they exceed the default step count in the trace, the set value is recorded.

- **profile_step_max_count** Int

  Default `1024`

  Set the maximum number of trace steps. If the number of collected trace steps exceeds this value, all steps collected later are ignored. If `profile_step_heavy_count` is set up to 1000, active stacks are collected as many as `profile_step_max_count`.

- **profile_step_heavy_time** MiliSecond

  Default `100`

  Set the criterion of heavy steps. If the execution time is greater than the set value and `profile_step_normal_count` is exceeded, recording is made as many as `profile_step_heavy_count`.

- **active_stack_count** Int

  Default `100`

  Set the maximum number of active stacks to collect in transactions.

- **profile_method_resource_enabled** Boolean

  Default `false`

  When collecting method steps from the transaction trace, the CPU and memory usages in the step is traced.

- **profile_position_method** String

  It records the stack traces that run the set methods.

- **profile_position_depth** Int

  Default `50`

  Set the maximum number of lines when recording stack traces for position tracing.

- **trace_error_callstack_depth** Int

  Default `50`

  Set the maximum number of stack trace lines collected when errors occur.

- **trace_active_callstack_depth** Int

  Default `50`

  Set the maximum number of stack trace lines collected by active stacks.

- **trace_active_transaction_yellow_time** MiliSecond

  Default 3000

  Set the criteria to display in yellow on the active transaction arc equalizer.

- **trace_active_transaction_red_time** MiliSecond

  Default 8000

  Set the criteria to display in red on the active transaction arc equalizer.

- **hook_method_patterns** String

  It sets the method to measure the response time. Class full names and methods are separated by the period (.) as the delimiter, and asterisk (*) can be used as the wildcard. In case of multiple items, comma (,) is used as the delimiter.

  whatap.conf

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

  - The methods registered through this option can be checked through the Method step in **Trace analysis** or **Transaction information** window.
  - It is recommended to apply it locally as much as possible. If applied extensively, it may cause performance degradation in applications.

- **hook_method_supers** String

  Set the super class when measuring the response time of the method inherited from a specific class. It sets the full name of the class. Multiple items are separated with comma (,).

  whatap.conf

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

- **hook_method_interfaces** String

  When measuring the response time of the method that implements a specific interface, it sets the interface. It sets the full name of the interface. Multiple items are separated with comma (,).

- **hook_method_ignore_classes** String

When setting a method trace, it sets the classes to exclude from the trace.

- **hook_method_access_public_enabled** Boolean

  Default `true`

  When setting a method trace, it sets whether or not to treat only the public methods as separate targets.

- **hook_method_access_private_enabled** Boolean

  Default `false`

  When setting a method trace, it sets whether or not to treat only the private methods as separate targets.

- **hook_method_access_protected_enabled** Boolean

  Default `true`

  When setting a method trace, it sets whether or not to treat only the protected methods as separate targets.

- **hook_method_access_none_enabled** Boolean

  Default `true`

  When setting a method trace, it sets whether or not to treat only the no access indicated methods as separate targets.

- **stacklog_socket_port** TCP_PortNumber

  Default `0`

  If the destination TCP port has been set, the stack trace is recorded as an agent log at the time of Socket.connect(). It can be used when tracing DB connections, HTTPC connections, and such that are not checked by default settings.

  > Stack traces are recorded whenever the set destination is connected. It may cause performance degradation. It must be used temporarily only in the agents selected for debugging purposes.
  >
  > Example of DB connection tracing: `stacklog_socket_port=1521`

- **trace_concern_error_enabled** Boolean

  Default `true`

  It sets whether or not to collect detailed traces with errors, regardless of the `trace_basetime` or `profile_basetime` option.

Exceptional handling

The following options are used to manage and monitor exceptions that occur in applications. You can check the stability of applications by registering business exceptions and collecting related statistics. You can also set whether to ignore specific exceptions or treat them as errors if the HTTP status code is an error. Exception-handled items appear as normal transactions in the **Hitmap** or **Trace analysis** window.

> For more information about the Java agent's exception handling criteria, see the following.

- **biz_exceptions** String

  It registers the business exception.

  whatap.conf

  ```
  # example
  biz_exceptions=com.home.test.exception.BizException
  ```

  > Error class collection/Hitmap display level: Includes the INFO (blue)/error statistics.

  > In the Java agent v2.2.20 or higher, the registered business exceptions are collected as error statistics. In the versions below v2.2.20, the registered business exceptions are excluded from error statistics. However, they appear in the trace details.

- **biz_exceptions_status** String Java Agent v2.2.20 or later

  Set the business exception and status to restrict event notifications when any exception occurs. For **Business Exception:Status** set by this option, the event level is changed to INFO. Business exceptions and status are separated by colon (:). If multiple statuses are used for a business exception, they are separated by ampersand (&). You can set multiple values using comma(,) as the delimiter.

  whatap.conf

  ```
  # example
  ```

```
biz_exceptions_status=java.util.concurrent.TimeoutException:400&404&408,java.lang.Exception:200
```

> Error class collection/Hitmap display level: Includes the INFO (blue)/error statistics.

- **ignore_exceptions** String

  The errors of the registered exceptions are ignored.

  whatap.conf

```
# example
ignore_exceptions=com.home.test.exception.BizException
```

> Ignore error class/Hitmap display level: Excludes the INFO (blue)/error statistics.

- **transaction_status_error_enable** Boolean

  Default `true`

  If an HTTP status code that is not a normal response such as HTTP 401 or 403, is returned, set whether or not to handle it as an error.

- **status_ignore** String

  You can set the HTTP status codes to ignore. For multiple values, use comma (,) as the delimiter.

  whatap.conf

```
# example
status_ignore=408,500
```

> Ignore status error/Hitmap display level: Excludes the INFO (blue)/error statistics.

- **status_ignore_set** String

  `whatap.error.STATUS_ERROR` is ignored. For multiple values, use comma (,) as the delimiter.

whatap.conf

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

Ignore status error set/Hitmap display level: Excludes the INFO (blue)/error statistics.

- **httpc_status_error_enable** Boolean

  Default `true`

  Set whether or not to collect when the HTTP status code is an error. It includes client error responses (`400` or higher) and server error responses (`500` or higher).

  For more information about the HTTP status codes, see the following.

- **httpc_status_ignore** String

  If the HTTP status code is an error (`HTTPC_ERROR`), it can be ignored. For multiple values, use comma (,) as the delimiter.

  whatap.conf

  ```
  # example
  httpc_status_ignore=404,500
  ```

  Ignore HTTP status error code/Exclude error statistics

- **httpc_status_url_ignore_set** String

  `whatap.error.HTTPC_ERROR` is ignored. For multiple values, use comma (,) as the delimiter.

  Set it as `httpc_status_url_ignore_set =/a/b/c:400,/a/ab/c:404` . In Statistics/Error Analysis, enter the value for `HTTPC_URL` .

  whatap.conf

```
# example
httpc_status_url_ignore_set=/a/b/c:404,/d/e/f:500
```

> Ignore HTTP status error code/Exclude error statistics

- **trace_sql_exception_enabled** Boolean

  Default `true`

  It sets whether to trace SqlException when occurred in the JDBC driver. By setting the option value to `false`, you can perform `biz_exceptions` as custom exception handling.

- **trace_exception_handler_overwrite_enabled** Boolean

  Default `true`

  If an exception that passes through the spring-boot ExceptionHandler occurs, it can replace the previous exception. In the **Trace analysis** tab of the **Record summary** window, you can mark the exception as an exception registered in the ExceptionHandler of the spring-boot.

HTTP transaction tracing

The following agent options trace and record various data related to HTTP requests and responses. This allows you to closely monitor and analyze the HTTP transactions for applications. You can add parameter values to the transaction names or exclude specific URLs or HTTP methods. You can also record transactions in detail, including HTTP headers and parameters.

- **trace_normalize_enabled** Boolean

  Default `true`

  It enables the function to parse and normalize transaction URLs.

> If the value is changed to `false`, it disables the path parameter parsing. In this case, it is recommended to temporarily use the setting only for debugging purposes because the statistical data is less meaningful.

- **trace_auto_normalize_enabled** Boolean

  Default `true`

  It enables automatic parsing of pattern values by extracting them from annotations when normalizing transaction URLs.

- **trace_normalize_urls** String

  Set transaction URL patterns to normalize. It deletes the path parameter after parsing the call URL patterns.

  > For example, if declared as `/a/{v}/b` , `a/123/b` is replaced by `a/{v}/b` . For multiple values, use comma (,) as the delimiter. A complement is required after arranging substitution patterns.

- **web_static_content_extensions** String

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

  Set the extension to determine the static content. Transactions with the set extensions are excluded from the transaction tracing and counting.

- **trace_transaction_name_header_key** String

  The value extracted from the set HTTP header key is added at the end of the transaction name.

- **profile_http_header_enabled** Boolean

  Default `false`

  To record HTTP headers in the trace details, set it to `true` .

- **profile_http_parameter_enabled** Boolean

  Default `false`

  To record HTTP parameters in the trace details, set this to `true` . Parameters can be viewed by entering a separate security key.

  > - **Earlier than Java Agent 2.2.2**: The security key is written in the WAS server `${WHATAP_AGENT_HOME}` /paramkey.txt file with a 6-letter value. If the paramkey.txt file does not exist, it is automatically created with a random value.
  > - **Java Agent 2.2.2 or later**: For the security key, check the `paramkey` key value in the WAS server

> ${WHATAP_AGENT_HOME}/security.conf file. If the security.conf file does not exist, the paramkey key value is automatically created as **WHATAP**.
>
> - For more information about the security key configuration file, see the following.

- **profile_http_header_url_prefix** String

  It is used to define the prefix of URL to record HTTP headers in the trace history.

- **profile_http_parameter_url_prefix** String

  It is used to define the prefix of URL to record HTTP parameters in the trace details.

- **trace_transaction_name_key** String

  If you set the HTTP request parameter value in the option, the parameter value is extracted and added to the end of the transaction name.

  For example, when calling the /api/test URL with paramKey as an HTTP parameter, if you set it as in the following example, the transaction name in the trace history appears as /api/test$paramKey={value}. However, multiple parameters cannot be registered.

  whatap.conf

```
# example
trace_transaction_name_key=paramKey
```

- **recursive_max** Int

  Default 1000000

  This option is used to detect whether any transaction has been recursively called. Set the criteria for issuing event notifications by counting the number of recursive calls that are caused by a single transaction.

> It targets the HTTP URL recursive calls. It counts the recursive calls delivered through jsp:forward.

- **hook_httpservlet_classes** String

  It additionally sets END POINT for HTTP transactions. For the first 2 parameters of the method, only HttpServletRequest and HttpServletResponse can be set.

- **hook_jsp_patterns** String

  Default `org.apache.jasper.servlet.JspServlet.serviceJspFile`

  Set the method to load the JSP files. It displays the JSP information returned as a result of the transaction call in a trace. Default values are automatically added to the settings added through this option.

- **trace_ignore_url_set** String

  It sets URLs to exclude from transaction tracing. For multiple values, use comma (,) as the delimiter.

  > The URLs registered through this option are excluded from the **Hitmap**, **Trace analysis**, or **Transaction information** window for collection.

- **trace_ignore_active_url_set** String

  It sets URLs to exclude from the **Active transaction** list. For multiple values, use comma (,) as the delimiter.

  > The URLs registered through this option are only excluded from the **Active transaction** window for collection, and do not affect other transaction data such as Hitmap, Trace analysis, or Transaction info.

- **trace_ignore_url_prefix** String

  It sets URL prefixes to exclude from transaction tracing. For multiple values, use comma (,) as the delimiter.

  > The URLs registered through this option are excluded from the **Hitmap**, **Trace analysis**, or **Transaction information** window for collection.

- **ignore_http_method** String

  Default `PATCH,OPTIONS,HEAD,TRACE`

  Transactions requested by the specified HTTP method are not collected. To set multiple items, use comma (,) as the delimiter.

  whatap.conf

```
# e.g. tracing the transactions whose http_method is OPTIONS or HEAD
ignore_http_method=PATCH,TRACE
```

```
# e.g. ignoring the transactions whose http_method is OPTIONS or HEAD
ignore_http_method=OPTIONS,HEAD
```

> ⓘ The HTTP methods registered through this option are excluded from the **Hitmap**, **Trace analysis**, or **Transaction information** window for collection.

- **trace_tx_name_with_method_enabled** Boolean

  Default `false`

  Statistical data can be collected using URL+method where the URL has been changed. You can check with URL+method in **Statistics** or **Transaction Search**.

  e.g. /api/user+GET, /api/user+POST

Non-HTTP transaction tracing

The following options are used to trace and monitor non-HTTP transactions: These options allow you to closely trace background tasks or transactions for other protocols.

- **trace_auto_transaction_enabled** Boolean

  Default `false`

  Collection is not allowed when the trace target method is not performed within the transaction start point (`Javax.http.httpservlet` , `hook_service_*`). In this case, set to make the trace target method become the transaction start point.

  > ⓘ This is used when finding the end point of the background transaction mainly in the development environment.

- **trace_auto_transaction_backstack_enabled** Boolean

  Default `true`

  If the value of the `trace_auto_transaction_enabled` option is `true` , the stack trace is recorded upon start of

transaction. As the above, the start point of transaction can be found.

- **trace_background_socket_enabled** Boolean

  Default `true`

  Tracing is recorded even when the socket is opened by the background, not by the transaction.

- **async_stack_enabled** Boolean

  Default `false`

  Set whether or not to use the active stack function for the background threads.

- **async_thread_match** String

  Set the background thread name for dumping the active stacks. To set multiple items, use comma (,) as the delimiter. When setting the name, you can use a comparison pattern using an asterisk (*), such as "Thread-*."

- **async_thread_parking_class** String

  Default `sun.misc.Unsafe`

  If the top method of the stack is a class or method registered in `async_thread_parking`, the system considers that the thread is parked and does not generate any dump.

- **hook_service_patterns** String

  It sets the starting point pattern for tracing NON-Http transactions. Wildcard characters cannot be used. To set multiple values, use comma (,) as the delimiter.

- **hook_serivce_ignore_methods** String

  Among the settings in `hook_service_patterns`, unnecessary methods as the start point can be added.

- **hook_service_supers** String

  If the common denominator of the starting points for NON-HTTP transaction tracing inherits the method of a specific class, set the method of the common denominator. Enter the full path to the method. You can use wildcards ("*") and register multiple methods by using comma (,) as the delimiter.

  whatap.conf

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

- **hook_service_interfaces** String

Set this if the common denominator of the start points for non-HTTP transaction tracing has implemented a specific interface.

- **hook_service_access_public_enabled** Boolean

  Default `true`

  When setting the transactions for the non-HTTP daemon process, it sets the access permission criteria to on or off only for public methods.

- **hook_service_access_private_enabled** Boolean

  Default `true`

  When setting the transactions of the non-HTTP daemon process, it sets the access permission criterion to on or off only for private methods.

- **hook_service_access_protected_enabled** Boolean

  Default `true`

  When setting the transactions of the non-HTTP daemon process, it sets the access permission criterion to on or off only for protected method.

- **service_name_mode** String

  Default `full`

  You can use the following options with the transaction name.

  - `full` : Full class name is used.
  - `class` : The service name is used with the service name.
  - `method` : The service name is used with the method name.
  - `string` : The service name is used with the first parameter in the string.
  - `arg` : Among parameters, the service name is used with the parameter of the index set in the `service_name_index` option.

Multiple transaction trace

The following options are used to trace and monitor cross-application calls. The options can help you understand the interactions between multiple applications. Various options provided allow you to fine-tune the behaviors for

tracing multiple transactions.

- **mtrace_enabled** Boolean

  Default `true`

  It sets whether or not to use the multi-transaction tracing (MTID). By tracing the MTID, you can check the calls between all the registered applications.

- **mtrace_rate** Percentage

  Default `10`

  With this option, you can set the rate of MTIDs issued when the first transaction occurs.

- **mtrace_caller_key** String

  Default `x-wtap-mst`

  Set the caller key name to be used for MTID tracing.

- **mtrace_callee_key** String

  Default `x-wtap-tx`

  Set the callee key name to be used for MTID tracing.

- **mtrace_send_url_length** Int

  Default `80`

  The HTTP caller passes its URL to the callee. At this time, the URL length is restricted. This sets the length value.

- **mtrace_callee_id_send_enabled** Boolean

  Default `false`

  When tracing the MTIDs, the callee ID issued upon the HTTPC call is included in the trace.

- **mtrace_callee_id_recv_enabled** Boolean

  Default `false`

  Upon MTID tracing, the received callee ID is represented in the trace.

- **mtrace_alltx_enabled** Boolean

  Default `false`

MTIDs of all transactions can be traced to link with the logging system like log4j.

- **mtrace_basetime** Miliseconds

  Default `100`

  If the value of `mtrace_alltx_enabled` is `true`, it may leave too many logs. In this case, it needs to reduce the amount of trace logging.

- **stat_mtrace_enabled** Boolean

  Default `false`

  It collects correlation statistics between callers and callees. When applied to a caller, the caller details are sent. When applied to a callee, the caller-callee call statistics are sent to the collection server in URL units.

Trace data sampling

The following options are used to control sampling of trace data and set whether to transmit data. The options allow you to manage the performance and data transfer volume for applications.

Agent support scope: Java Agent v2.2.4 or later

- **trace_sampling_enabled** Boolean

  Default `false`

  You can set whether or not to collect trace data samples.

- **trace_sampling_tps** Int

  Default `10000`

  You can set the trace data sampling count. The data exceeding the set count is not transmitted and is initialized every 5 seconds.

- **trace_send_enabled** Boolean

  Default `true`

  You can set whether or not to transmit the trace data.

- **debug_trace_samling** Boolean

  Default `false`

You can turn on or off the trace data sampling debug option.

Modifying the trace URL name

- **hook_tx_name_patterns / hook_tx_name_mode** String Java Agent v2.2.4 or later

  You can modify the name of the trace URL when calling a specific URL or method. For example, if the `test100000()` method is called while calling a URL registered in `io.home.test.TestController.test1`, you can see it as `/api/100000+test1+test100000` on the service screen.

  The `hook_tx_name_mode` option allows you to select one of **class**, **method**, **string**, and **return**.

  whatap.conf

```
# The full path of the call method and wildcard (*) can be used.
hook_tx_name_patterns=ab.cd.ef.GH.ij

# Select one of class, method, string, and return.
hook_tx_name_mode=method
```

Ignoring a specific exception

- **ignore_exception_tx_pattern** String Java Agent v2.2.4 or later

  You can configure a service to ignore specific exceptions. When registering multiple items, use a comma (,) as the delimiter.

  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
```

Tracing HttpURLConnection

The following options are used to trace HTTP calls by using the `HttpURLConnection` class.

- **HttpURLConnection** Boolean

  Default `true`

  The option is used to trace methods in the `HttpURLConnection` class.

- **hook_HttpURLConnection_startup_enabled** Boolean Java Agent v2.2.4 or later

  Default `false`

  You can set to trace methods in the `HttpURLConnection` class even when called for the first time.

- **HttpURLConnection_weblogic** Boolean Java Agent v2.2.4 or later

  Default `true`

  In case of calling HTTP from Weblogic with `HttpURLConnection`, the option traces methods in the `weblogic.net.http.HttpURLConnection` class.

- **hook_HttpURLConnection_weblogic_startup_enabled** Boolean Java Agent v2.2.4 or later

  Default `false`

  You can set to trace methods in the `weblogic.net.http.HttpURLConnection` class even when called for the first time.

Collecting memory quota

- **trace_malloc_enabled** Boolean

  Default `false`

  It collects average allocated memory per transaction

  > When it is enabled by setting the option to `true`, you can check **Memory allocation** in the **Record summary** tab of the **Trace analysis** window. For more information, see the following.

DB/SQL Configuration

The following provides various options for Java agents to handle the database and collect SQL performance data. They can trace various DB connection pools such as DBCP, Hikari, and Tomcat and record SQL execution details. Additionally, it provides information collection options for performance optimization, such as recording SQL parameters and tracing DB connection leaks.

- **dbcp_pool_enabled** Boolean

  Default `true`

  It is used to trace DB connection data of DBCP without using JMX.

- **hikari_pool_enabled** Boolean

  Default `false`

  It is used to trace DB connection data of the hikari pool without using JMX.

- **tomcat_ds_enabled** Boolean

  Default `false`

  It enables the function to trace Tomcat DB connection pool data without using JMX.

- **tomcat_pool_enabled** Boolean

  Default `true`

  It enables the function to trace Tomcat DB connection pool data without using JMX.

- **weblogic_ds_enabled** Boolean

  Default `false`

  It enables the function to trace the Weblogic DB connection pool data by using JMX.

- **weblogic_pool_enabled** Boolean

  Default `true`

  It enables the function to trace the Weblogic DB connection pool data without using JMX.

- **jeus_pool_enabled** Boolean

  Default `true`

It enables the function to trace the JEUS DB connection pool data without using JMX.

- **profile_connection_open_enabled** Boolean

  Default `true`

  It records DB connection open data in the trace details.

- **trace_dbc_close_enabled** Boolean Java Agent v2.2.3 or later

  Default `false`

  It works only when `profile_connection_open_enabled` is `true`. DB connection close-up data is recorded in the transaction trace details.

  > ⓘ **For versions earlier than Java Agent 2.2.3**, use the `profile_dbc_close` option name.

- **profile_sql_param_enabled** Boolean

  Default `false`

  It is used to record SQL parameters in the transaction trace details. The parameters can be viewed by entering a separate security key.

  > ⓘ
  > - **Earlier than Java Agent 2.2.2**: The security key is written in the WAS server `${WHATAP_AGENT_HOME}` /paramkey.txt file with a 6-letter value. If the paramkey.txt file does not exist, it is automatically created with a random value.
  > - **Java Agent 2.2.2 or later**: For the security key, check the `paramkey` key value in the WAS server `${WHATAP_AGENT_HOME}` /security.conf file. If the security.conf file does not exist, the `paramkey` key value is automatically created as **WHATAP**.
  > - For more information about the security key configuration file, see the following.

- **profile_sql_resource_enabled** Boolean

  Default `false`

  It traces the CPU and memory usages during collection of SQLs in the transaction trace.

- **profile_update_count** Boolean

  Default `false`

It collects the number of updates when SQL update statements have been executed through the `executeUpdate()` method.

- **custom_pool_classes** String

  When using a separate connection pool that has not been defined previously, set the class name.

- **ds_update_interval** MiliSeconds

  Default `5000`

  Set the interval for counting DB connection data.

- **profile_position_sql** Boolean

  Default `false`

  It records the stack traces when SQL statements are executed.

- **profile_sql_param_length** Int

  Default `40`

  Set the length of the SQL parameter.

  > The maximum value is 128. Even if you enter a number than this value, parameters are saved up to 128 bits.

- **trace_dbc_leak_enabled** Boolean

  Default `false`

  It enables to trace DB connection leaks.

  > Because leaks are traced using the connection wrapper, be sure to apply it after a test as it can affect the operation services.

- **trace_dbc_leak_fullstack_enabled** Boolean

  Default `false`

  When any DB connection leak is detected, it collects stack traces at this time.

> Do not apply this at the peak time, if possible. If the option is applied, the CPU usage may increase slightly. It is recommended to apply it temporarily for troubleshooting purposes only.

- **trace_sql_normalize_enabled** Boolean

  Default `true`

  It enables the function to normalize any SQL statements by extracting the literal part from each SQL statement.

- **profile_error_jdbc_fetch_max** Int

  Default `10000`

  When the SQL Fetch Count (the number of `ResultSet.next()` calls) exceeds the set value, it is treated as the **TOO_MANY_RECORDS** error. If it is set to `0`, it is not treated as an error.

- **profile_error_sql_time_max** Int

  Default `30000`

  When the SQL execution time exceeds the set value, it is treated as the **SLOW_SQL** error. If it is set to `0`, it is not treated as an error.

- **hook_connection_open_patterns** String

  It registers the method to call upon DB connection open. It is common to register getConnection of any connection pool that has not been set.

  Example

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

- **hook_jdbc_con_classes** String

  Set an unregistered JDBC connection class.

  Example

  ```
  hook_jdbc_con_classes=mypool.ConPool
  ```

- **hook_jdbc_pstmt_classes** String

Set an unregistered JDBC PreparedStatement class. Note that it must be a structure where the SQL string is delivered to the creator parameter.

Example

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

- **hook_jdbc_cstmt_classes** String

  Set an unregistered JDBC CallableStatement class.

  Example

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

- **hook_jdbc_stmt_classes** String

  Set an unregistered JDBC statement class.

  Example

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

- **hook_jdbc_rs_classes** String

  Set an unregistered JDBC result set class.

  Example

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

- **hook_jdbc_wrapping_driver_patterns** String

  The obfuscated JDBC driver like the DB2 driver is difficult to directly perform BCI with the `hook_jdbc_xxx` option. In this case, SQLs can be traced with the wrapper method. At this time, set `Driver.connect` for tracing.

- **debug_dbc_stack_enabled** Boolean

  Default `false`

  StackTrace is saved in the transaction trace at the time of DB connection. It is used to obtain connection pool

information that is used by applications.

- **ignore_sql_hash_set** String Java Agent v2.2.4 or later

  You can set to ignore the SQL statements that match the SQL hash value in the option.

HTTPC/API Call Configuration

The following explains how to set options for the Java agent to collect and analyze the data related to HTTP outbound calls and API calls. It provides various options for detailed monitoring, such as **SLOW_HTTPC** error handling, CPU and memory usage tracing, stack trace records upon calls, and URL normalization. This allows developers to efficiently manage external dependencies for applications and identify performance issues.

- **profile_error_httpc_time_max** Int

  Default `10000`

  If the HTTPC execution time exceeds the specified value, it is handled as the **SLOW_HTTPC** error. If set to `0`, it is not treated as an error.

- **profile_httpc_resource_enabled** Boolean

  Default `false`

  When collecting HTTP call steps from the transaction trace, the CPU and memory usages in the step is traced.

- **profile_position_httpc** Boolean

  Default `false`

  It records the stack traces at the execution time of HTTPC.

- **trace_httpc_normalize_enabled** Boolean

  Default `true`

  It enables the function to parse and normalize HTTPC URLs for transactions.

- **trace_httpc_normalize_urls** String

  Set the HTTPC URL patterns to normalize. It deletes the path parameter after parsing the call URL patterns.

  > ⓘ For example, if declared as `/a/{v}/b`, `a/123/b` is replaced by `a/{v}/b`. For multiple values, use comma (,) as the delimiter. A complement is required after arranging substitution patterns.

- **hook_httpc_patterns** String

  Set the `full package class name`.`method` for HTTP outbound calls. Register and use the full path of the method that performs HTTP calls. After registration, you need to redefine the class or restart the agent.

whatap.conf

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

- To register multiple classes, use comma (,) as the delimiter.
- Only asterisk (*) can be used as the wildcard, regular expressions cannot be used.

Script Plugin

The plugin option provided by the Java agent provides a way for you to inject desired code into the trace data or add additional information before and after execution of the method. It includes the options for setting where to run specific code at method entry/exit time and how to configure a class to monitor custom pools. More specific usages are provided through actual plug-in application cases, and it guides you to various configuration examples and API usage that can increase the flexibility and precision for monitoring.

Plug-in options

The Java agent plug-in is executed by inserting in the method start and end. The locations (class, method) to apply the plug-in can be set as agent options.

- `hook_trace_helper_start_patterns`

  At the beginning of the method, set the point (class, method) where the trace plug-in is to be inserted. Write the plug-in code in the `$WHATAP_HOME` /plugin/TraceHelperStart.x file.

- `hook_trace_helper_end_patterns`

  At the end of the method, set the point (class, method) where the trace plug-in is to be inserted. Write the plug-in code in the `$WHATAP_HOME` /plugin/TraceHelperEnd.x file.

- `hook_trace_helper_patterns`

  At the beginning and end of the method, set the point (class, method) where the trace plug-in is to be inserted. Write the plug-in codes in the `$WHATAP_HOME` /plugin/TraceHelperStart.x and `$WHATAP_HOME` /plugin/TraceHelperEnd.x files.

- `custom_pool_classes`

  To monitor the custom pool, set the class with the pool usage. Write the plug-in code in `$WHATAP_HOME` /plugin/CustomPool.x.

Plug-in setting example

In case of setting multiple classes, use the comma (,) as the delimiter. Part or all of the string of the package

name can be replaced with "*."

- **In case of setting the package and method name**

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

- **In case of replacing the package and method name with** `*`

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

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

- **In case of targeting the whole**

```
*.*
```

- **In case of setting the custom pool**

  The custom pool is set in the form of identifier@package name.

  For example, before the class name, set the identifier, `whatap_plugin_guide`. Identifiers and classes are separated by @.

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

Plug-in code writing example

In most cases, the explicit type casting is required because objects are returned. The following examples are regarding the plug-in code writing.

Writing the start time of the set method in the transaction trace

1. Write the method to apply in the Java agent's `hook_trace_helper_patterns` option in whatap.conf.

   $WHATAP_HOME/whatap.conf

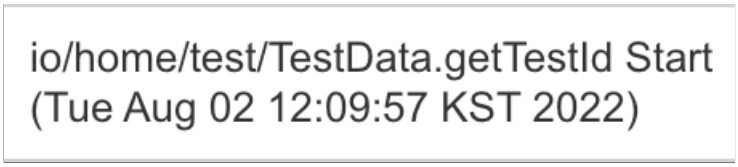
```
hook_trace_helper_patterns=org.apache.catalina.connector.RequestFacade.*
```

2. Write the plug-in code in TraceHelperStart.x.

   $WHATAP_HOME/plugin/TraceHelperStart.x

```
// Substitute the class name and method name in the string prefix.
String prefix = $point.class1 + "." + $point.method;
// Set the WhaTap trace property name to `prefix` + "st" as the current time.
$ctx.setAttribute(prefix + "st", new Long(System.currentTimeMillis()));
// Add the current time in the trace information.
$ctx.profile(prefix + " Start", new java.util.Date().toString());
```

Display the method start time in the transaction trace as follows.



Recording the set method's end time and execution time in the transaction trace

1. Write the method to apply in the Java agent's `hook_trace_helper_patterns` option in whatap.conf.

   $WHATAP_HOME/whatap.conf

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

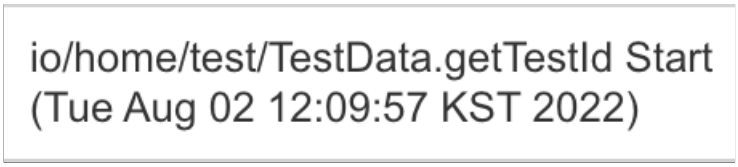
2. Write the plug-in code in TraceHelperEnd.x.

   $WHATAP_HOME/plugin/TraceHelperEnd.x

```
1  String prefix = $point.class1 + "." + $point.method;
2  // Import the trace attributes added in the TraceHelperStart.x file.
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  // Add the gap (execution time) between the start time and the current time to the trace information.
8  $ctx.profile(prefix + " End", sb.toString());
```

It displays the method start time and execution time in the transaction trace as follows.



Plug-in API

The plug-in APIs are divided into common items: `$ctx` , `$point` , `$req` , and `$res` . Upon the transaction start/end, it can be applied in HTTPC section, specific method execution section, etc.

> It is prohibited to store personally identifiable information by using plug-in APIs. When collecting personal information, this must be clearly stated in the privacy policy.

Provision of parameters - common plug-in files

Parameter	Method/variable	Description
$ctx	Property management	
	void setAttribute(String key, Object value)	It sets an attribute.
	Object getAttribute(String key)	It fetches the attribute value.
	Trace Data	

Parameter	Method/variable	Description
	void profile(String desc)	Recording the trace message
	void profile(String name, String msg)	Recording the trace step name and message
	void profile(String name, String msg, int elapsed)	It records the transaction trace step name, message, and execution time.
	void profile(String name, String msg, int elapsed, int value)	Recording the transaction trace step name, message, execution time, and sequence.
	HTTP Request	
	String service()	It returns the service name.
	void service(String name)	It sets the service name.
	int serviceHash()	It returns the hash value of the service name.
	String remoteIp()	It returns the service request IP.
	void remoteIp(String ip)	It sets the service request IP.
	boolean isError()	It returns the value whether it is the error or not.
	void login(String id)	It sets the log-in ID.
	String login()	It returns the log-in ID.
	String httpMethod()	It returns the HTTP method.
	String httpQuery()	It returns the HTTP query.
	String httpContentType()	It returns the HTTP content type.
	String userAgent()	It returns the user agent.

Parameter	Method/variable	Description
	int status()	It returns the HTTP status.
$point	String class1	Class name
	String method	Method name
	Object this1	Hooking class/method
	Object[] args	Arguments
	Object return1	Return

Parameters not provided - common plug-in file

Parameter	Method/variable	Description
No	void log(Object c)	Logging through the logger
	void println(Object c)	Printing through System.out.printin()
	Object field(Object o, String field)	It returns the field value.
	Object method(Object o, String method)	Invoke method
	Object method(Object o, String method, String param)	Invoke method
	String toString(Object o)	It returns toSting() as object.
	String toString(Object o, String def)	It returns toSting() as object and returns def in case of null.
	int syshash(Object o)	It returns the hash value.

Parameter	Method/variable	Description
	int syshash(HookArgs hook, int x)	It returns the hash value of the xth argument.
	int syshash(HookArgs hook)	It returns the hash value of the argument.
	int cint(Object o)	It returns an integer value.
	float cfloat(Object o)	It returns a floating value.
	String cString(Object o)	It returns a string.
	long clong(Object o)	It returns a value of long type.
	double cdouble(Object o)	It returns a double value.
	String desc(Object o)	It returns the class signature.
	String toJson(Object o)	It returns as json.
	void shell(final String cmd, final String env)	It executes a shell command.

Daemon and batch

This API is applied to the methods set as starting points when monitoring the application servers such as daemons and batches. Write code in the following file.

- Application of the service start area: `$WHATAP_HOME` /plugin/AppServiceStart.x
- Application of the service end area: `$WHATAP_HOME` /plugin/AppServiceEnd.x

Plug-in file	Parameter	Method/variable	Description
AppServiceStart.x	$ctx	See the common attributes.	
	$point	See the common attributes.	
AppServiceEnd.x	$ctx	See the common attributes.	

HTTP service

This API is applied to the common web applications that use the `javax.servlet.http.HTTPServlet` class. Write code in the following file.

- Application of the service start area: `$WHATAP_HOME` /plugin/HttpServiceStart.x
- Application of the service end area: `$WHATAP_HOME` /plugin/HttpServiceEnd.x

Plug-in file	Parameter	Method/variable	Description
HttpServiceStart.x/ HttpServiceEnd.x	$ctx	See the common attributes.	
	$req	String getCookie(String key)	It returns the cookie value.
		String getRequestURI()	It returns RequestURI.
		String getRemoteAddr()	It returns RemoteAddr.
		String getMethod()	It returns the HTTP method.
		String getQueryString()	It returns HTTP QueryString.
		String getParameter(String key)	It returns the parameter value.
		Object getAttribute(String key)	It returns the attribute value.
		String getHeader(String key)	It returns the header value.
		Enumeration getParameterNames()	It returns getParameterNames.
		Enumeration getHeaderNames()	It returns getHeaderNames.
		WrSession getSession()	It returns the session wrapper.
		Set getSessionNames()	It returns getSessionNames.
		Object getSessionAttribute(String key)	It returns getSessionAttribute.

Plug-in file	Parameter	Method/variable	Description
		boolean isOk()	It returns the plug-in status.
		Throwable error()	It returns an error.
	$res	String getContentType()	It returns the content type.
		String getCharacterEncoding()	It returns the character encoding.
		boolean isOk()	It returns the plug-in status.
		Throwable error()	It returns an error.

HTTP Outbound

This API can be applied for HTTP outbound call by using the library such as `HttpClient`. Write code in the following file.

- Application of the service start area: `$WHATAP_HOME`/plugin/HttpCallStart.x
- Application of the service end area: `$WHATAP_HOME`/plugin/HttpCallEnd.x

Plug-in file	Parameter	Method/variable	Description
HttpCallStart.x	$ctx	See the common attributes.	
	$req	String url()	It returns a URL.
		String host()	It returns the host name.
		int port()	It returns the port.
		boolean isOk()	It returns the plug-in status.
		Throwable error()	It returns an error.
HttpCallEnd.x	$ctx	See the common attributes.	

Plug-in file	Parameter	Method/variable	Description
	$res	String getContentType()	It returns the content type.
		String getCharacterEncoding()	It returns the character encoding.
		boolean isOk()	It returns the plug-in status.
		Throwable error()	It returns an error.

Specific method

This API is applied to the method that has been set through the Java agent option (`hook_trace_helper_*`).

Plug-in file	Parameter	Method/variable	Description
TraceHelperStart.x	$ctx	See the common attributes.	
	$point	See the common attributes.	
TraceHelperEnd.x	$ctx	See the common attributes.	
	$point	See the common attributes.	

Custom pool

This API is used to monitor the custom pool. If the method with the pool usage is set, you can observe it and check statistical information through the WhaTap monitoring service.

Plug-in file	Parameter	Method/variable	Description
CustomPool.x	$id	ID set in custom_pool_classes	
	$pool	Class set in custom_pool_classes	

Plug-in file	Parameter	Method/variable	Description
	$result	active(Object o)	It sets the active pool count.
		int idle(Object o)	It sets the idle pool count.

Plug-in application example

The following is the actual plug-in application example.

Search request monitoring of the elastic search engine.

Because the Elasticsearch (ES) engine is not a servlet engine, unstructured monitoring is required. If other products perform metric-based monitoring, you can use WhaTap's plug-in to trace ES requests, process times, and search keywords.

whatap.conf

```
# Set org.elasticsearch.search.SearchService.executeQueryPhase as the transaction end point.
hook_service_patterns=org.elasticsearch.search.SearchService.executeQueryPhase

# Each time the method registered in hook_trace_helper_start_patterns is performed, the code in TraceHelperStart.x is executed.
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         // The query method call's result for the first argument is to be saved in String query.
11         // org.elasticsearch.search.query.QueryPhase.execute(SearchContext searchContext)
```

```
        // It indicates that the result of the invoked SearchContext.query() method is saved in String query.
        String query = " " + method($point.args[0], "query");
        // It outputs the string query as the trace information.
        $ctx.profile(query);
    }
} catch(Exception e) {
    $ctx.profile(e.toString());
}
```

Excluding the collection of transactions flowed into a specific IP

Health Check, a plug-in can be used to exclude transactions caused by internal users.

${WHATAP_HOME}/plugin/HttpServiceStart.x

```
// Set the ignoreIP value as the IP to exclude for collection.
String ignoreIP = "123.234.123.234";
String remoteIP = $req.getRemoteAddr();

if ($ctx.ok() && remoteIP != null) {
    if (remoteIP.equals(ignoreIP)) {
        $ctx.ignore();
    }
}
```

Atomikos pool monitoring

Apply it by referencing API document.

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);
```

Presenting the cookie value in the transaction trace

You can see the cookie value in the transaction trace.

${WHATAP_HOME}/plugin/HttpServiceStart.x

```
1  // Set the cookie to check with the $req.getCookie() parrameter.
2  String cookie = $req.getCookie("AWSALB");
3
4  if ($ctx.ok()) {
5      if (cookie != null) {
6          $ctx.profile(cookie);
7      }
8  }
```

Number Of Users Configuration

The following explains how to count the number of users connected to the Java web application server through agent settings. It includes the setting whether to enable real-time user counting, setting of the cookie limit to count users, the method how to count users based on the IP address or specific HTTP header values, and setting of HTTP headers to trace client IPs.

- **trace_user_enabled** Boolean

  Default `true`

  Set whether or not to aggregate real-time users. If the user tracing option setting is duplicated, the action priority is as follows.

  1. **trace_user_using_ip / wclient_using_ip**
  2. **trace_user_using_jsession**
  3. **user_header_ticket / wclient_header_ticket**

  > In case of Java Agent 2.2.0 or later, use the `wclient_using_ip` and `wclient_header_ticket` options.

- **trace_user_cookie_limit / wclient_cookie_limit** Int

  Default `2048`

  If cookies are issued for counting users, a cookie overflow may occur if there are too many cookies. To prevent this from happening, set the limit.

  > In case of Java Agent 2.2.0 or later, use the `wclient_cookie_limit` option.

- **trace_user_using_ip / wclient_using_ip** Boolean

  Default `true`

  It aggregates real-time users based on the IP address. If you want to distinguish users based on the cookie rather than IP addresses, change the value to `false`.

> In case of Java Agent 2.2.0 or later, use the `wclient_using_ip` option.

- **user_header_ticket / wclient_header_ticket** String

  To aggregate the numbers of users with a specific value in the HTTP header, set the key value. If headers are used to distinguish users in case of mobile clients, the following options are available.

  whatap.conf

```
user_header_ticket=login
```

> In case of Java Agent 2.2.0 or later, use the `wclient_header_ticket` option.

- **trace_http_client_ip_header_key** String

  This function sets the client IP (remote IP) with a specific HTTP header value. In the proxy environment, the `X-Forwarded-For` header can be set with the client IP.

  If the load balancer like L4 is located before WEB/WAS, in some cases, the IP address of L4 becomes the remote address instead of the IP address of the client. In this situation, if the actual client IP is recorded as a specific key value in the HTTP header, it can be replaced with the corresponding key value.

- **trace_user_using_jsession** Boolean

  Default `false`

  It aggregates real-time users based on SESSIONID.

Load Control Configuration

This guide describes the agent options that configure the traffic throttling options for Java application servers. The following includes the features to limit the maximum number of concurrent requests for applications and control service accesses based on specific users or URLs. It provides various options such as sending custom messages, setting URL redirections, and enabling event notifications, to ensure stability of applications under overload situations.

> ⓘ The throttling option of the Java agent applies to only Servlet-based applications. The servers that support this option are servlets such as Tomcat, Weblogic, and Undertow. However, the latest version of Spring does not support this option.

- **throttle_enabled** Boolean

  Default `false`

  It activates the throttling function that limits the maximum concurrent processes for an application. All options that start with "throttle_" work in `throttle_enabled=true`.

  The priorities of throttling control policies are as follows:

  1. Block: Services are blocked based on the URL and user IP address. It is applied first.
  2. Passing: URLs applied to Passing take precedence over Reject policies.
  3. Reject: Reject policies are applied after Block and Passing policies.

- **throttle_limit** Int

  Default `10000`

  If the number of concurrent requests (transactions) for each agent exceeds the set value, additional requests are rejected.

- **throttle_rejected_message** String

  Default `too many request!!`

  Set the message to send to users when throttling is restricted.

- **throttle_rejected_forward** String

Set the guide page URL that is delivered to users when user requests are rejected due to over the limit. If multiple `throttle_rejected_message` settings have been set, `throttle_rejected_forward` is applied first.

> If the guide pages are served in the same container, it can be an obstacle due to recursive calls because the pages are also considered as the number of concurrent processes. Thus, the guide page must be created as a static HTML page or must be outside.

- **reject_event_enabled** Boolean

  Default `false`

  Set whether or not to issue an event notification when any user request is rejected.

- **reject_event_interval** MiliSecond

  Default `30000`

  Alerts for duplicate events do not occur for the set time after an event notification for Reject has occurred.

- **throttle_blocking_url** String

  Set the URL to block (or reject) even when `throttle_limit` is not exceeded. It can urgently block URLs that cause system failures.

- **throttle_blocking_ip** ip_address

  Set to block users based on the IP address. It can be used to block DDoS attacks and users based on the IP address. Use comma (,) as the delimiter in case of multiple items.

- **throttle_target_urls** String

  Apply the throttling function to the registered URLs. Use comma (,) as the delimiter in case of multiple items.

- **throttle_passing_url** String

  Even when `throttle_limit` is exceeded, set the URL for processing. Use comma (,) as the delimiter in case of multiple items.

- **throttle_passing_url_prefix** String

  Even when `throttle_limit` is exceeded, set the URLs for processing. Use comma (,) as the delimiter in case of multiple items.

- **throttle_blocked_message** String

Default `request blocked!!`

Set the message to be delivered to the users whose requests have been blocked.

- **throttle_blocked_forward** String

  Set the URL to forward to the users whose requests have been blocked. If multiple `throttle_rejected_message` settings have been set, `throttle_rejected_forward` is applied first.

Agent Notification Configuration

The following explains how to configure notifications for various events that occur in the application server through Java agents. It includes the options to configure event notifications for transaction recursive calls, denial of service, HTTPC connection errors, excess of heap and disk usage, CPU usage threshold violation, DB connection duplicate allocations, and exceptions. For each event, you can make detailed adjustments such as release interval, release status, and threshold settings.

- **recursive_event_interval** MiliSeconds

  Default `300000`

  Set the interval between event notifications for recursive calls of transactions.

- **reject_event_enabled** Boolean

  Default `false`

  Set whether or not to issue an event notification when any service is rejected (call load restriction/decline).

- **reject_event_interval** MiliSeconds

  Default `300000`

  Set the event notification interval when any service is rejected (call load restriction/decline).

- **httpc_event_enabled** Boolean

  Default `false`

  Set whether or not to issue event notification when an HTTPC connection error occurs.

- **httpc_event_interval** MiliSeconds

  Default `300000`

  Set the event notification interval when an HTTPC connection error occurs.

- **heap_event_enabled** Boolean

  Default `false`

  Set whether or not to issue an event notification when the heap usage threshold is reached.

- **heap_event_percent** Percentage

  Default `90`

Set the threshold for issuing heap usage event notifications.

- **heap_event_duration** MiliSeconds

  Default `30000`

  Set the duration for issuing heap usage event notifications.

- **heap_event_interval** MiliSeconds

  Default `300000`

  Set the interval for issuing heap usage event notifications.

- **heap_event_action** String

  Default `NONE`

  Set the dynamic loading code to run when a heap usage event occurs.

  > ⓘ ID to deliver to the Java code written in the `$WHATAP_HOME` /plugin/ActionScript.x file (delivered to `$id`)

- **disk_event_enabled** Boolean

  Default `false`

  Set whether or not to issue an event notification when the disk usage threshold is reached.

- **disk_event_percent** Percentage

  Default `90`

  Set the threshold for issuing disk usage event notifications.

- **disk_event_interval** MiliSeconds

  Default `300000`

  Set the interval for issuing disk usage event notifications.

- **disk_event_action** String

  Set the dynamic loading code to run when a disk usage event occurs.

> ⓘ ID to deliver to the Java code written in the `$WHATAP_HOME` /plugin/ActionScript.x file (delivered to `$id`)

- **cpu_event_enabled** Boolean

  Default `false`

  Set whether or not to issue an event notification when the CPU usage threshold is reached.

- **cpu_event_percent** Percentage

  Default `90`

  Set the threshold for issuing CPU usage envent notificaiton.

- **cpu_event_duration** MiliSeconds

  Default `30000`

  Set the duration for issuing CPU usage event notifications.

- **cpu_event_interval** MiliSeconds

  Default `300000`

  Set the interval for issuing CPU usage event notifications.

- **cpu_event_action** String

  Set the ID to passed to the dynamic loading code to run when a CPU usage event occurs.

  > ⓘ ID to deliver to the Java code written in the `$WHATAP_HOME` /plugin/ActionScript.x file (delivered to `$id`)

- **dbc_dup_event_enabled** Boolean

  Default `false`

  Set whether or not to issue an event notification when the DB connection is duplicated.

- **dbc_dup_event_fullstack_enabled** Boolean

  Default `false`

  Set whether or not to secure a stack when the allocated DB connection is duplicated.

ID to deliver to the Java code written in the `$WHATAP_HOME` /plugin/ActionScript.x file (delivered to `$id`)

Apdex Configuration

The following describes the Java agent options that set Apdex scores to measure the user satisfaction. Apdex metrics help you objectively evaluate the application's performance and improve user experience.

whatap.apdex_time millisecond

Default `1200`

The default value (S) can be modified in Agent CONFIG.

Application Performance Index (Apdex) is the indicator of the application performance. You can measure the customer satisfaction about your web applications. It can be used as an indicator of user satisfaction and the value is between 0 and 1.

(Number of satisfied cases + (Number of tolerating cases * 0.5))/Total number of requests

- S (Satisfied): No problem in business ≤ 1.2 seconds (Default value)
- T (Tolerating): Business is possible with any delay ≤ 4.8 seconds (S * 4)
- F (Frustrated): Business impossible > 4.8 seconds (exceeds T and error)

> ⓘ It is available in the agent 2.0 or later.

Statistics Configuration

The following guides you to the agent options for various statistics collections to monitor Java applications. It includes performance counter extensions, transaction collection by domain, multi-server transaction dependency analysis, statistics collection by login type and referral, and more. You can also check the maximum record limit settings for collecting SQLs, HTTP calls, error statistics, and user agent information.

- **addin_monitor_num** Int

  Default `0`

  Set the number of classes to declare for expansion of the performance counter. Declare `addin_monitor_x` as many as the `addin_monitor_num` setting.

- **addin_monitor_0** Int

  Default `0`

  whatap.conf

  ```
  addin_monitor_num=2
  addin_monitor_0=a.a.Class1
  addin_monitor_1=a.a.Class2
  ```

- **stat_domain_enabled** Boolean

  Default `false`

  It enables the function to collect transaction statistics for each domain that connects to clients.

- **stat_domain_max_count** Int

  Default `7000`

  Maximum number of records of the transaction statistics for each domain to collect by a single JVM for 5 minutes.

- **stat_mtrace_enabled** Boolean

Default `false`

It collects the version-dependent statistics between callers and callees among multiple server transactions.

- **mtrace_spec** String

  Default: `v1`

  Set the application version of the current instance. You can arbitrarily set a string. This data is used for call statistics.

- **stat_mtrace_max_count** Int

  Default `7000`

  It collects the version-dependent statistics between callers and callees among multiple server transactions.

- **stat_login_enabled** Boolean

  Default `false`

  It collects transaction statistics for each log in.

- **stat_login_max_count** Int

  Default `7000`

  Maximum number of records for log-in based transaction statistics to collect by a single JVM for 5 minutes.

- **stat_referer_enabled** Boolean

  Default `false`

  It collects transaction statistics by referer.

- **stat_referer_max_count** Int

  Default `7000`

  Maximum number of records for the referer based transaction statistics to collect by a single JVM for 5 minutes.

- **stat_tx_max_count** Int

  Default `5000`

  It limits the number of transaction statistics. It limits the maximum number of records for statistics collected for 5 minutes that are delivered to the server.

- **stat_sql_max_count** Int

Default `5000`

The number of SQL statistics are limited. It limits the maximum number of records for statistics collected for 5 minutes that are delivered to the server.

- **stat_httpc_max_count** Int

  Default `5000`

  It limits the number of HTTP call statistics. It limits the maximum number of records for statistics collected for 5 minutes that are delivered to the server.

- **stat_error_max_count** Int

  Default `1000`

  The number of error statistics are limited. It limits the maximum number of records for statistics collected for 5 minutes that are delivered to the server.

- **stat_useragent_max_count** Int

  Default `500`

  The number of user agent statistics are limited. It limits the maximum number of records for statistics collected for 5 minutes that are delivered to the server.

Transaction Error Stack Configuration

The following guides you to the agent configuration options that detect errors during transactions in Java applications and display stack traces. It collects errors from various areas such as transactions, methods, SQLs, HTTP calls, and SOCKET calls, and includes methods how to set the error stack length, title length, and such. It helps you improve error handling through agent settings and support more efficient error diagnosis for applications.

It is not related to the `status_code` of the calling service. When multiple errors occur, the error stack displays only the first error stack.

> The agent configuration in this document is supported by **Java Agent 2.2.15** or later.

Agent CONFIG.

Total configuration

whatap.conf

```
# Whether or not to display error stacks
trace_exception_stack=false

# Default value for the error stack size
trace_exception_depth=50

# Title size of the error stack message
trace_exception_title_len=128
```

Method

whatap.conf

```
# Whether or not to display the method error stack
```

```
# You can turn the feature on or off with true/false values, following the trace_exception_stack value.
trace_method_error_stack=trace_exception_stack

# Method error stack size
# You can follow the trace_exception_depth value and set it separately.
trace_method_error_depth=trace_exception_depth
```

SQL

whatap.conf

```
# Whether or not to display the SQL error stack
# You can turn the feature on or off with true/false values, following the trace_exception_stack value.
trace_sql_error_stack=trace_exception_stack

# SQL error stack size
# You can follow the trace_exception_depth value and set it separately.
trace_sql_error_depth=trace_exception_depth
```

HTTP Call

whatap.conf

```
# Whether or not to display the HTTP call error stack
# You can turn the feature on or off with true/false values, following the trace_exception_stack value.
trace_httpc_error_stack=trace_exception_stack

# HTTP call error stack size
# You can follow the trace_exception_depth value and set it separately.
trace_httpc_error_depth=trace_exception_depth
```

SOCKET

whatap.conf

```
# Whether or not to display the socket error stack
# You can turn the feature on or off with true/false values, following the trace_exception_stack value.
```

```
trace_socket_error_stack=trace_exception_stack

# Socket error stack size
# You can follow the trace_exception_depth value and set it separately.
trace_socket_error_depth=trace_exception_depth
```

Transactions

whatap.conf

```
# Whether or not to display the transaction error stack
# You can turn the feature on or off with true/false values, following the trace_exception_stack value.
trace_endtx_error_stack=trace_exception_stack

# Transaction error stack size
# You can follow the trace_exception_depth value and set it separately.
trace_endtx_error_depth=trace_exception_depth
```

Additional Agent Options

The following guides you to additional agent configuration options that can help you tailor the level of monitoring for Java applications. It includes collection of SQL parameters, collection of HTTP parameters and header data, extraction of user IPs, selection of user count tracing method, SAP function tracking, and the like. Through these options, more granular data collection and analysis can be tailored, especially in security-critical environments.

The following are the options excluded from the agent default settings. If necessary, you can add the following options for application.

- **profile_sql_param_enabled** Boolean

  Default `true`

  It collects the SQL parameters.

- **profile_http_parameter_enabled** Boolean

  Default `true`

  It collects the HTTP parameters.

- **profile_http_header_enabled** Boolean

  Default `false`

  It collects HTTP headers.

- **trace_http_client_ip_header_key** String

  Default `X-Forwarded-For`

  It extracts the user IPs from the HTTP header keys.

- **trace_user_method / wclient_cookie_limit** Boolean

  Default `cookie`

  It traces user counts based on the cookie.

  > ⓘ In case of Java Agent 2.2.0 or later, use the `wclient_trace_type` option.

- **hook_httpc_sap3_enabled** Boolean

Default `true`

If SAP JCO is used, it traces the SAP functions.

Asynchronous Tracking

The following explains how to trace activities of asynchronous applications through the Java agent settings. It includes the methods for collecting `CompletableFuture` methods and hooking those methods into user applications. It provides the options to add to the whatap.conf file along with configuration examples, to help enhance performance monitoring for asynchronous calls.

CompletableFuture

It guides you how to collect CompletableFuture methods through the Java agent settings.

Adding the CompletableFuture method

Add the CompletableFuture method to hook into your Java application project as an agent.

- Method name: Fixed to `trace`.
- ReturnType: `Supplier`, `Consumer`, `Runnable`, `Future`, `Function`

```
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;
    }
    public static <T, R> Function<T, R> trace(Function<T, R> f) {
    return f;
}
}
```

Adding the agent settings

Add the following options to the whatap.conf file to configure the agent.

whatap.conf

```
hook_completablefuture_patterns=io.home.test.util.W.*
```

- If you have a context, add the `trace` method to connect the transaction.
- If you have no context, start the service with the `hook_service_patterns` option.

Usage examples

- **Original**

```
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  }
```

- Application of `io.home.test.util.W.trace()`

```
1  // Application of io.home.test.util.W.trace()
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   }
```

Update Java Agent

Downloading the latest version of agent

1. Log in the WhaTap monitoring service (https://service.whatap.io).
2. Select a project to update the agent.
3. On the left of the screen, select **Management** > **Agent update**.
4. Click ⬇ **Latest version** or ⬇ **Download to all agents with the latest version**.
5. Download the agent and then restart the application server. Update is automatically applied.

Automatic configuration of the update agent

To update the Java agent, the server must be restarted. If whatap.agent.jar is modified before restart, a graceful shutdown may fail while the application server is restarting.

To update the Java agent, proceed in the following order:

1. App shutdown
2. Modification of the agent jar
3. App start

To easily process the above steps, the agent jar file name is to include the version number. It is important to set to automatically apply the latest versions during the update.

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

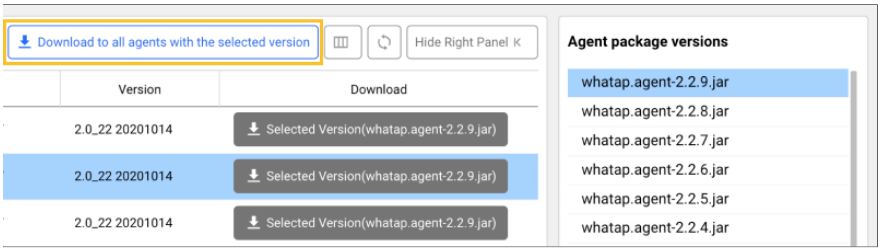
Uninstall Java Agent

Agent deletion

On the application server, delete the JVM's `-javaagent` option and then restart the server. Afterwards, delete the `$WHATAP_HOME` environment variable.

Agent rollback

1. Select a project for rollback in WhaTap Monitoring Service.
2. On the left of the screen, select **Management** > **Agent Update**.
3. Select **Other versions**.
4. Select a version for rollback and then click **Selected Version** or **Download to all agents with the selected version**.



5. During agent installation, in the modified script, change the `WHATAP_JAR` environment variable to the version for rollback.

```
########## 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 Group

In an Auto Scale Out environment, it may be difficult to recognize business unit information with only the IP, port, and PID of the automatically registered agent. In this case, the information can be easily recognized by giving the `okind` attribute.

The following describes the typical Docker environment.

- Each agent is a very small component of the Micro Service Architecture (MSA).
- Agents are automatically scaled in or out.
- You cannot predict which PID your application will start on which IP server.

As the JVM option to identify the business along with the agent name in the Docker environment, the `okind` attribute can be added by adding `whatap.okind`.

JAVA_OPTS

```
-Dwhatap.okind=mobile_ui
```

In whatap.conf, more settings can be added.

whatap.conf

```
whatap.okind=common_api
```

> To use various functions in the MSA environment, add the `okind` attribute.

AES 256 Encryption

The WhaTap application agent transfers the collected data after encryption. For this purpose, the XOR (Exclusive-OR) operation and Advanced Encryption Standard (AES) encryption are used. Encryption and decryption are performed by dividing plaintext into 128-bit units. To enhance security, the encryption can be extended up to 256 bits.

The Java Cryptography Extension (JCE) supports 128 bits by default. To apply the AES 256, JCE must be updated.

If you apply AES 256 in the default environment, the following error occurs.

```
Unsupported keysize or algorithm parameters.
##Or
Illegal key size or default parameters.
```

1. Download the file for your version from the following links.

   > ⓘ **Download versions for each Java version**
   >
   > - JAVA 8
   > - JAVA 7
   > - JAVA 6
   > - JAVA 5
   > - JAVA 1.42

2. Overwrite the file in the `$JAVA_HOME` /jre/lib/security path.
3. If JCE has been applied, add the following settings in `${WHATAP_HOME}` /whatap.conf.

   whatap.conf

   ```
   cypher_level=256
   ```

4. Once the application server has been restarted, the ACE 256 encryption is applied.

Scale-In Auto Delete

If an agent name is automatically assigned in the Auto Scale environment, administrator's intervention is not required when a scale out occurs. The system scales up automatically. If a scale in occurs, the system must recognize it as a graceful shutdown.

For automatic deletion of the agent, the WhaTap server must receive messages from the Java agent. The agent sends a **SILENT_SHUTDOWN** event to the Whatap server when the whatap `{java-process-pid}` .shutdown_ file of the Java process appears in `${WHATAP_HOME}` . Transfers are completed within 5 seconds.

When the **SILENT_SHUTDOWN** event has been transferred to the WhaTap server and then the agent is shut down within 1 minute, the WhaTap server deletes the agent from the list. Any inactive event does not occur.

> In case of Tomcat, you can configure in the shutdown.sh file.
>
> ```
> touch $WHATAP_HOME/whatap_{java-process-pid}.shutdown
> ```

Java Dashboard

After installing the WhaTap agent, log in WhaTap Monitoring Service. The initial screen is **Project list**. In **Project list**, you can see the application server in which the agent has been installed. An application server in which the agent has been installed is a project that is a monitoring unit. If a project is selected from the project list, you can enter **Application dashboard**.

What is the dashboard?

Through the dashboard, you can see the overall status of the project at a glance. You can check the total resource size for monitoring and filter the resources to be monitored on the real-time dashboard. It updates the active and terminated transactions in real time and provides visual charts. The administrator can intuitively see the exact status of services and systems. In addition, you can immediately recognize failures and potential problem factors.

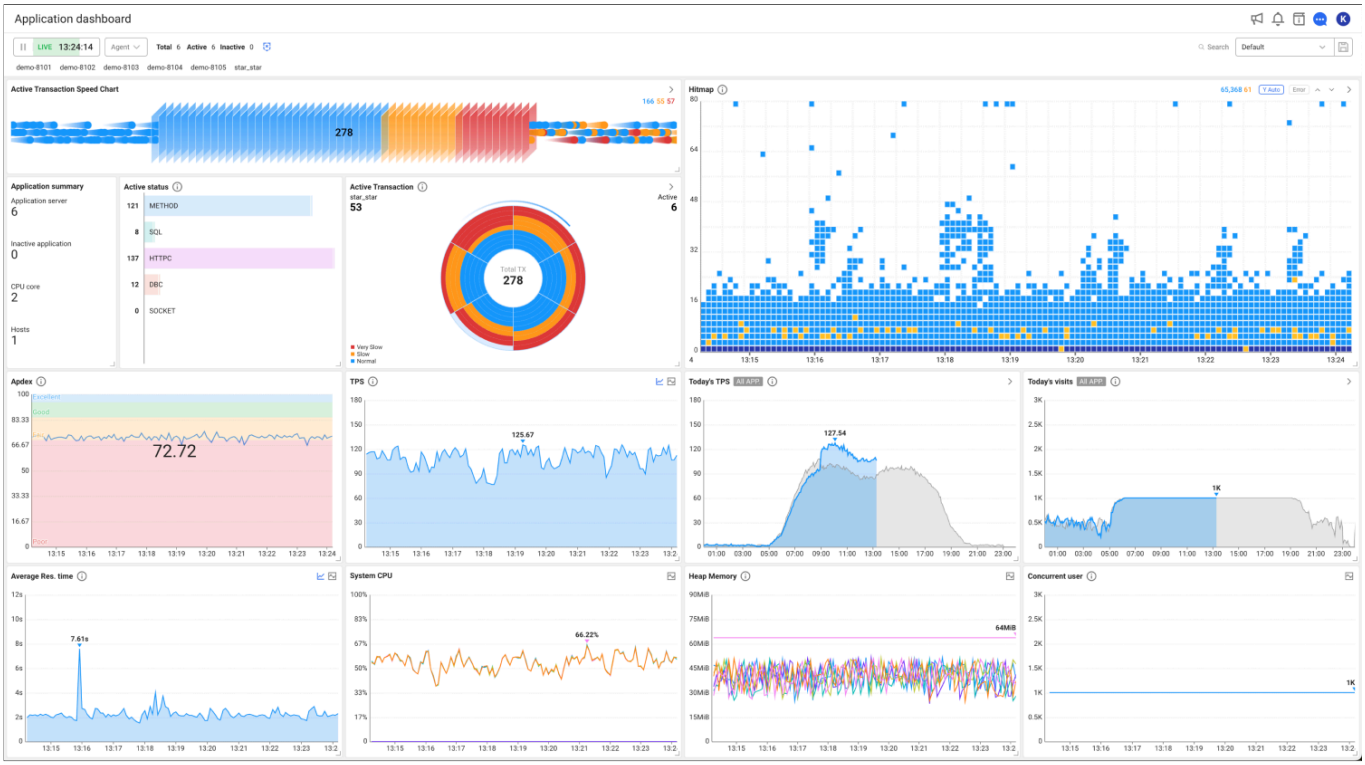
From the dashboard, you can see the following information:

- You can adjust the response time range or filter error data through the response time distribution chart.
- The dashboard analyzes the active transactions and updates the terminated transactions in real time.
- You can see the resource status such as TPS, response time, CPU, memory, and real-time usage.
- It provides the real-time monitoring service by automatically updating the latest data.

Learn more about Application Dashboard in the following video clip:

Application Dashboard

Home > Select Project > **Dashboard** > **Application dashboard**



Application dashboard allows you to collect key metrics of applications through agents, configure charts to monitor the metrics in real time, and view the past data. You can quickly identify and analyze application problems by using the widgets on the dashboard.

In case of general charts excluding CPU and memory, the stable data appears in blue and the elements identified as problems appear in red, making it easy to recognize the current status.

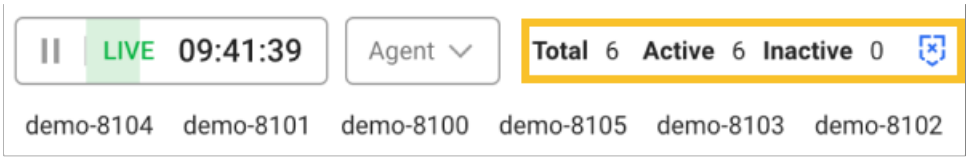
Query Time Setting

The dashboard basically provides the real-time monitoring feature but the past data can be viewed using the time selector at the top of the screen. In the time selector, select ⏸ and then set the following time. It updates

the data in widgets placed on the dashboard based on the set time.

Agent Check

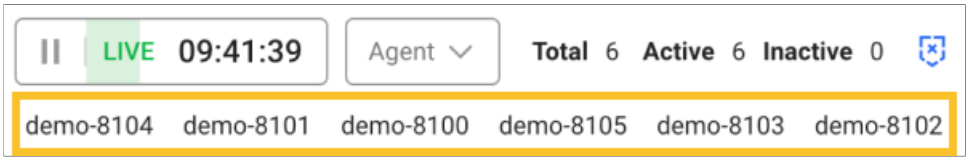
Checking the agent connection status



On the upper left of the screen, the right area of the time selector provides information that allows you to check the status of agents connected to the project. This allows you to immediately check whether the target application server is running.

- Total: Number of all agents connected to the project
- Active: Number of active agents
- Inactive: Number of inactive agents
- : Can display or hide inactive agents.

Agent-based monitoring



By default, the dashboard displays the metrics collected from all agents in charts and you can also view the data by agent. Select one or more agents under the time selector. The data of the widgets on the dashboard is updated with the metrics of the selected agent(s).

> To select all agents again while one or more agents is selected, deselect them or click Total.

> If there are numerous agents connected to the project, it is efficient to set the agent names to be short. For more information about the agent name setting, see the following.

Monitoring each classified agents

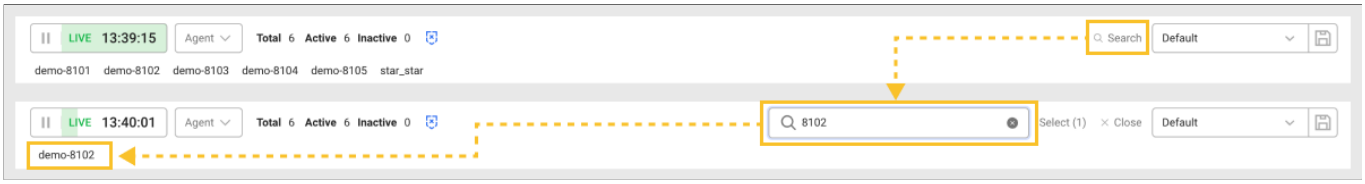
You can easily monitor multiple agents by categorizing them in Agent Settings.

- **Agent**: Individual Agent
- **By kind**: Group classified by the agent type (e.g. web server, database). Set it to `whatap.okind`.
- **By server**: Group categorized by the server (node). Set it to `whatap.onode`.
- **Agent by kind**: List of agents classified as `whatap.okind`.
- **Agent by server**: List of agents classified as `whatap.onode`.

> For more information about the agent configuration, see the following.

Searching the agent

If the agents connected to the project are numerous and difficult to find, use the agent search function. If you click **Search** on the upper right of the screen, a box appears where you can enter a string.



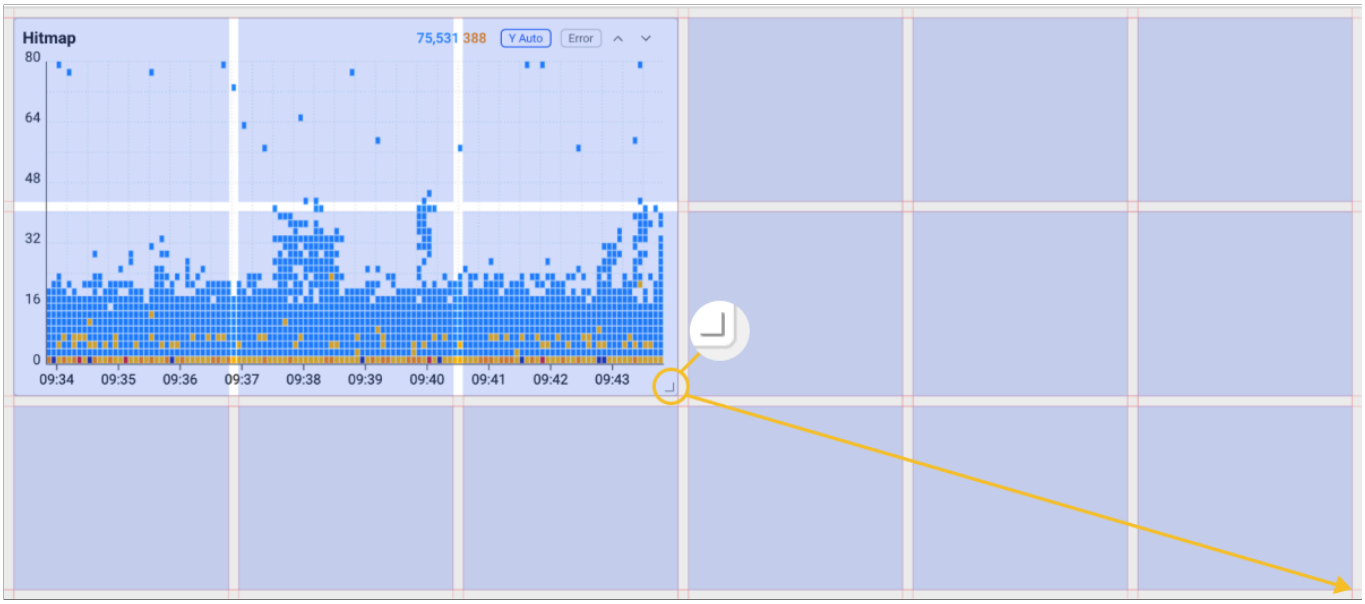
The agents that match the entered string appear in the list. To filter the data in the dashboard based on the searched agents, click **Select** on the right of the text entry box.

To reset the selected agent entries, click **Close** on the right of the input field.

Widget Edit

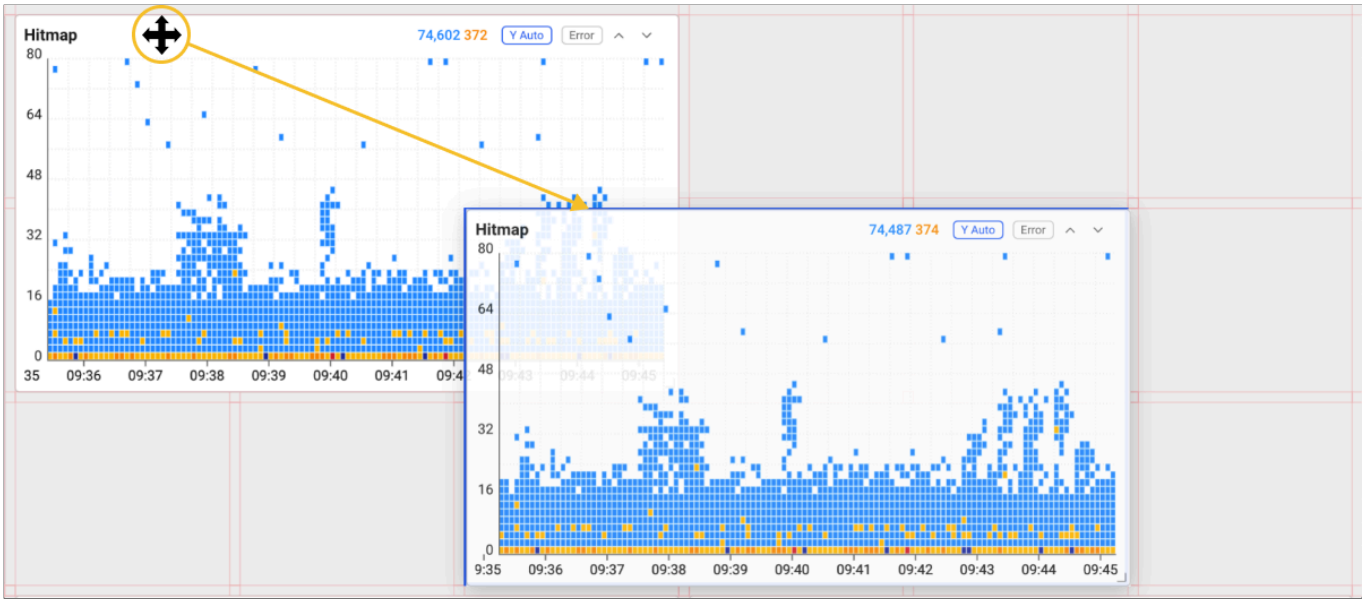
The widgets on the dashboard can be adjusted to the desired size and placed in the desired positions. You can also delete unnecessary widgets or add them again.

Resizing a widget



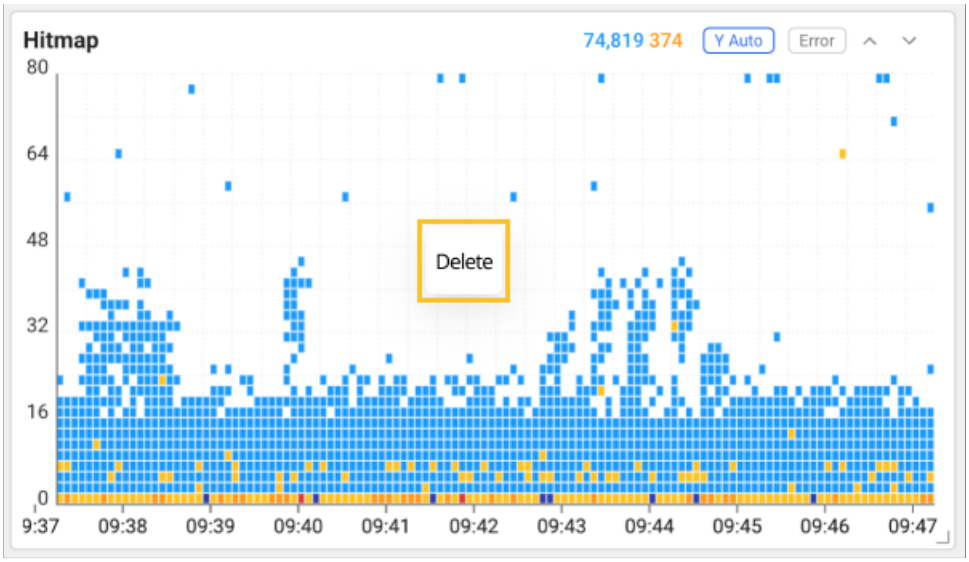
Click and hold the mouse on the ⌟ element at the lower right of the widget, and then drag it to the desired size. Grids with a uniform horizontal-to-vertical ratio appear and the size of the widget can be adjusted in each grid.

Moving a widget



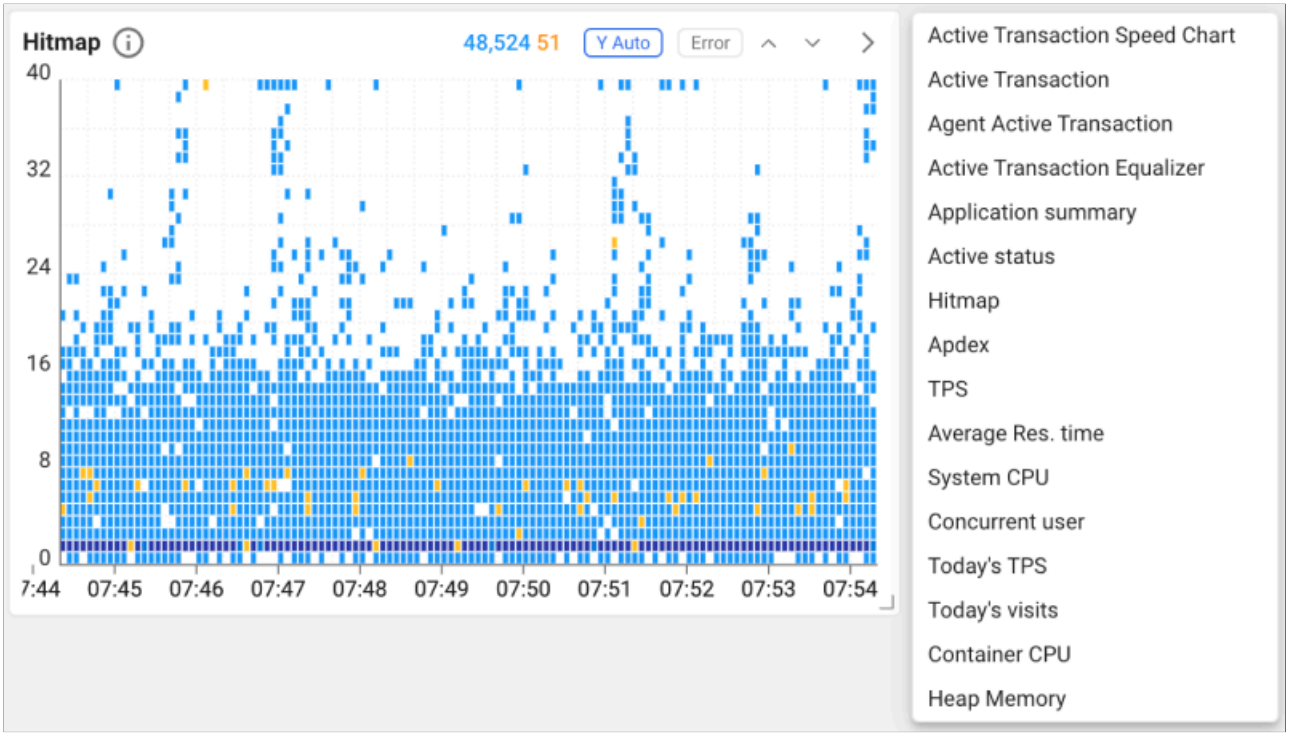
When you move the mouse cursor to the top of the widget, the cursor shape changes to ✢. At this time, you can move the widget by dragging with your left mouse button and move to the desired position.

Deleting a widget



Right-click on a widget to delete. If you select **Delete**, the widget is deleted from the dashboard.

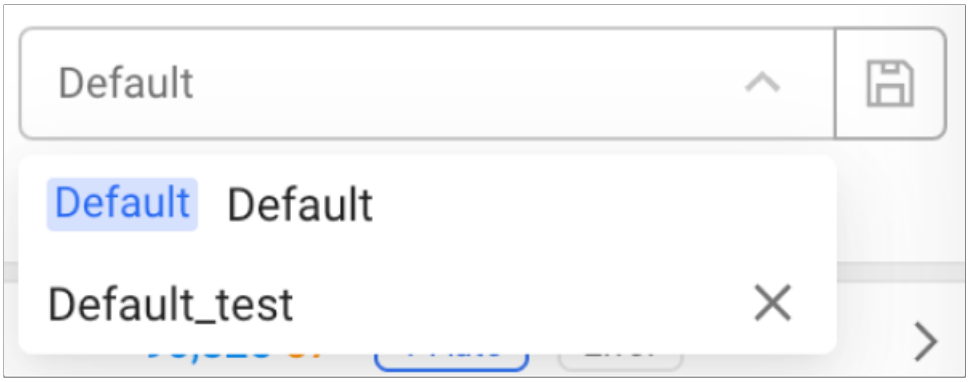
Adding a widget



Move the mouse cursor to an empty space on the dashboard and then right-click on it. Select a widget to add from the pop-up menu. Place the widget to the desired position and then resize it.

- For more information about the widgets that can be placed on the dashboard, see the following.
- The widgets that can be added are fixed, but we plan to increase supports through future updates.
- In addition to the metrics provided by default on the dashboard, you can add the desired metrics. For more information about custom widgets, see the following.

Preset Setting

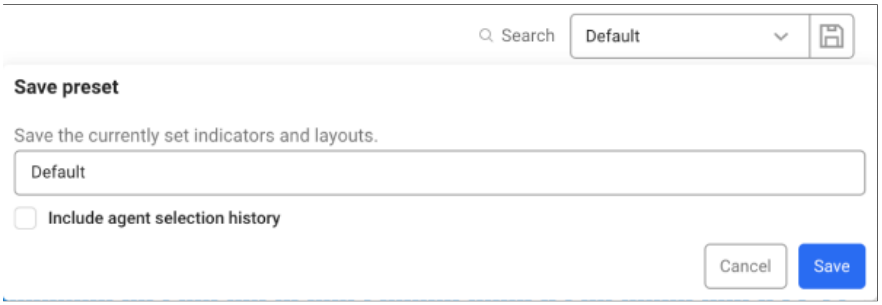


You can save and load the settings and layouts for widgets on the dashboard. You can create a new preset by adjusting its size and placing it on the desired position.

- Using **Default**, the specified preset cannot be changed.
- You can save or modify the presets with the **Edit project** role. For more information about the member roles, see the following.

Creating a new preset

1. Place the widgets in the desired format on the dashboard. You can also resize and place only the widgets to be frequently checked.
2. On the upper right of the screen, select .
3. Enter a new preset name.



To save the selected agent data, select **Include agent selection history**.

4. Select **Save**.

You can see the newly saved preset from the preset list.

- If any changes are made to the newly created preset, save the preset again. Select and then save the preset with the same name. Any changes are overwritten in the existing preset.
- If you go to another menu without saving changes on the dashboard, the changes are not saved.
- Presets are saved on a project basis and can be shared with other users.

Deleting presets

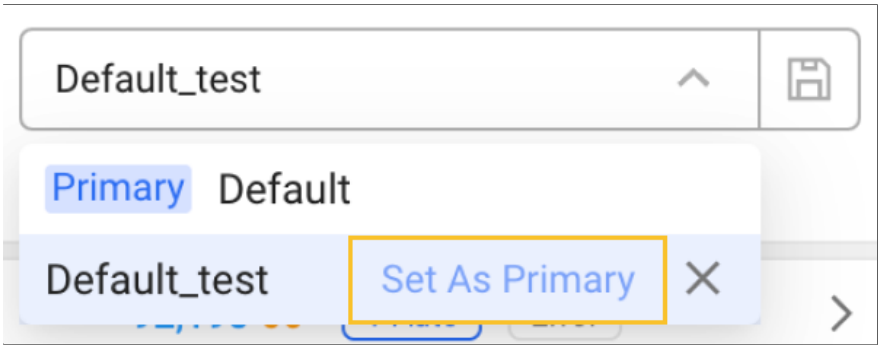
If you have unused presets, you can delete them from the preset list. Select ✕ on the right of the item to delete from the preset list.

The **Default** preset cannot be deleted.

Changing the default preset

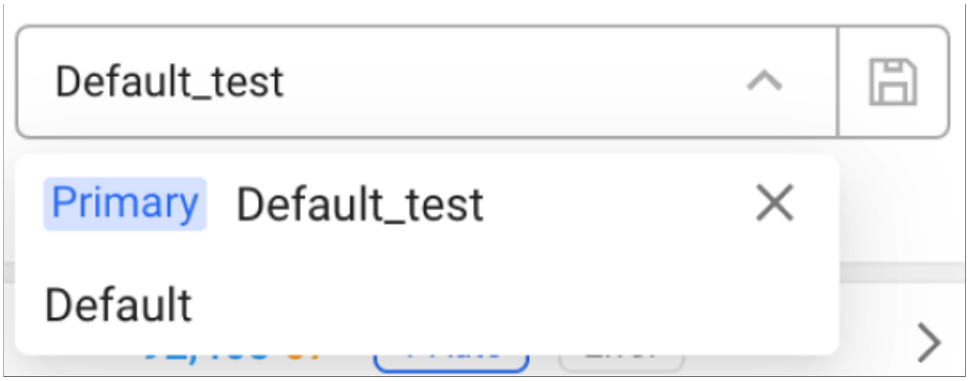
You can change the preset displayed when you first enter the **Application Dashboard** menu.

1. Check the preset selection box on the upper right of the dashboard.
2. Hover your mouse over a desired preset to set it as default.



3. If the **Set as primary** button appears, select it.

The selected preset is changed to the default preset, and the **Primary** tag appears in the preset list.



Widget Options

The icons displayed on the widget function as follows:

- ⓘ: You can see the features and information for major widgets.
- **Merge/view each**: The agent data for the widget is displayed individually or merged in a graph.
- **Detail**: A modal window appears where you can search the data of the widgets separately by agent.

> ⓘ The provided options may differ depending on the widget.

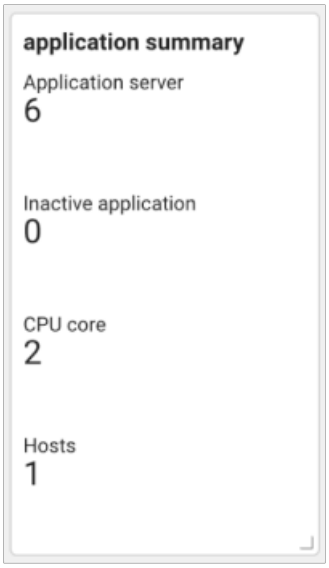
Learning about the widgets

> This document mainly describes the widgets placed on the **Application dashboard**. For more information about how to analyze transactions and users, services, and resources through **Application dashboard**, see the following documents:
>
> - Exploring the Application Dashboard
> - HITMAP transaction
> - DB connection delay and connection pool

Agent-related widgets

You can see the application information through the agent installed on the application.

Application summary



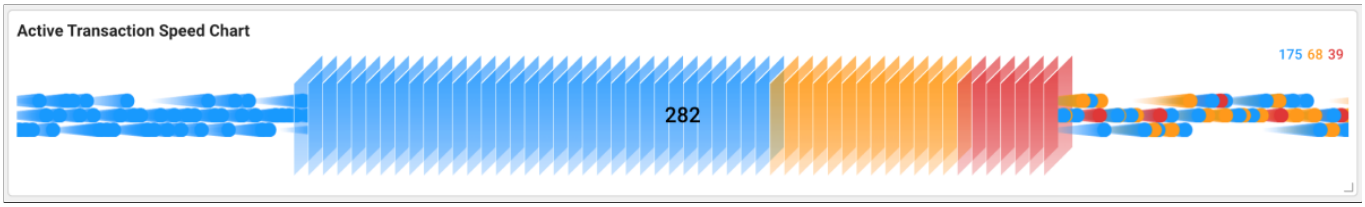
It displays the execution status and number of applications registered in the project, CPU core count, and host count.

Transaction-related widget

The failure status that can be checked through the transaction, can be recognized through the priority response time. If a transaction in progress does not end, this should be considered a failure. WhaTap divides sections for display by the time of the progress status. Blue (Normal) means transactions within the specified response time, Orange (Slow) means slow transactions with the response time of around 8 seconds, and Red (Very slow) means transactions with the response time that is twice as slow as normal. This allows you to intuitively recognize failures as fast as possible.

> If you add a database project through the **Correlated Project Management** menu, you can see the performance metrics of the project linked in the following widgets.
>
> - **Active Transaction Speed Chart**, **Active Transaction**: You can view the SQL performance of the database associated with the active transaction.
> - **Hitmap**: You can view performance metrics of the database associated with transactions in the past.
>
> For more detailed information about adding and analyzing a linked project, see [the following] (linked-projects).

Active Transaction Speed Chart



You can see the status of active transactions (middle area) and completed transactions (right area). The water drop icon passing from left to right means a transaction. If you click a chart area, a pop-up window appears where you can see a list of active transactions. If you select one of the transaction lists, you can view the transaction details, SQL query, and call stack information.

- Area on the left

  This indicates the number of requested transactions and displays data in proportion to the number of transactions.

- Area in the middle

  The status of active transactions is color displayed by the speed along with the count at the upper right. The number in the chart is the total number of active transactions. The colors in the chart indicate as follows:

  - Blue (Normal): 0 to 3 seconds
  - Orange (Slow): 3 to 8 seconds
  - Red (Very Slow): 8 seconds or more

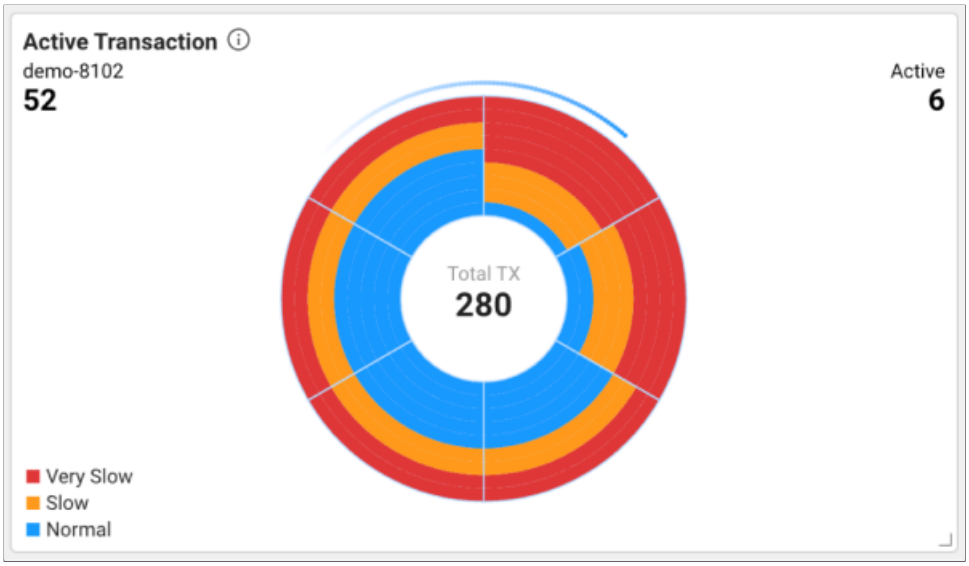
- Area on the right

  This indicates the Application Performance Index (Apdex). This metric is measured for customer satisfaction of a web application and has a value between 0 and 1. The droplet flying speed is proportional to the total number of Apdex transactions, and the droplet colors mean as follows:

  - Blue: Apdex satisfied count
  - Orange: Apdex tolerating count
  - Red: Apdex frustrated count = `Total number of APDEX transactions` - (`Apdex satisfied count` + `Apdex tolerating count`)

> For more information about the Application Performance Index (Apdex), see the following link.

Active Transaction



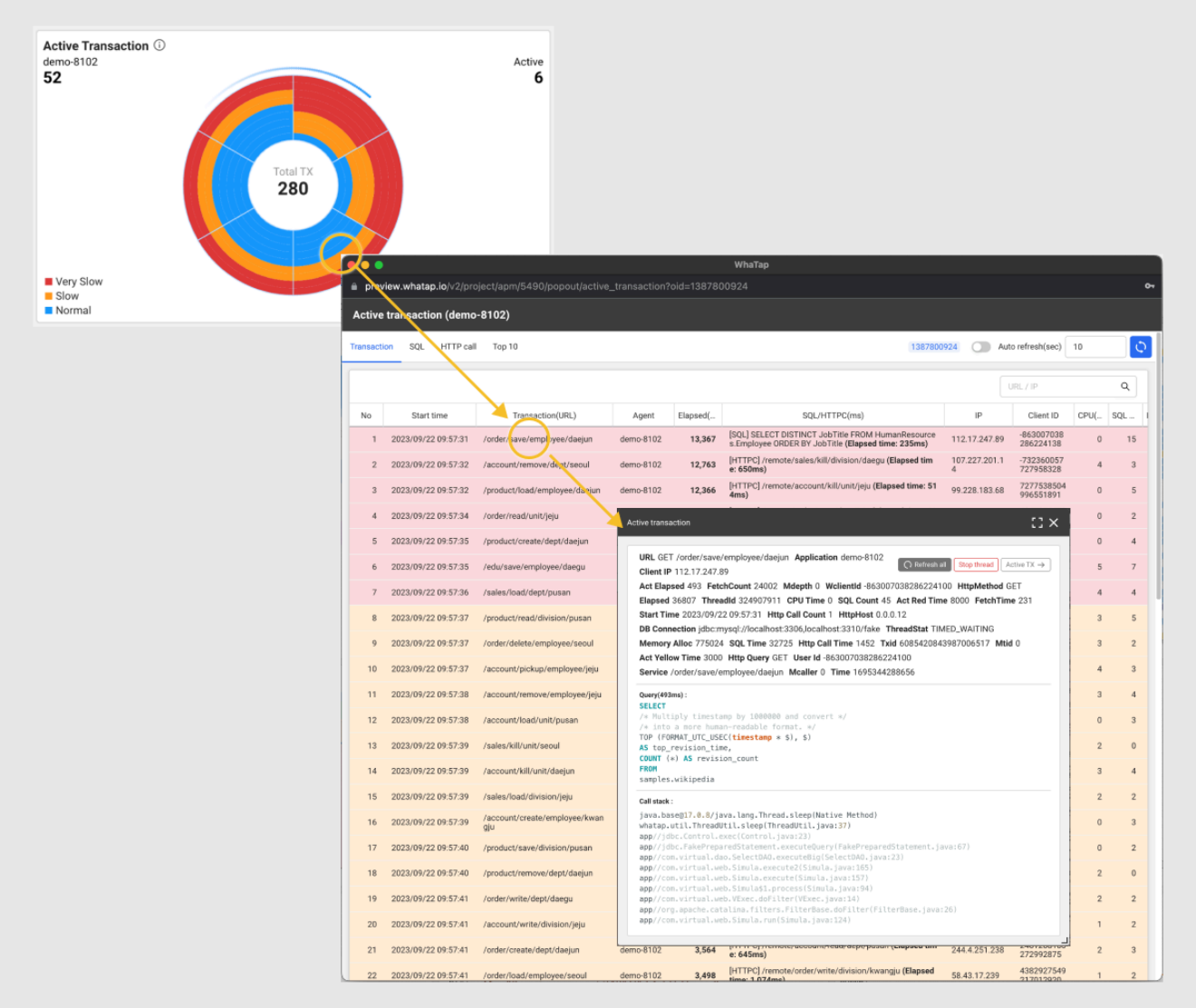
In the **Active Transaction** widget, the transactions in progress are displayed in the **arc equalizer** chart. You can see the status of delays by agent. The arc is divided by the number of agents. By representing the number of requests being processed by the server every 5 seconds, you can tell how long the request has been processed. Requests detected at an interval of 5 seconds are classified by color as follows to determine whether they are dangerous or not.

- Red (Very Slow): 8 seconds or more
- Orange (Slow): 3 to 8 seconds
- Blue (Normal): 0 to 3 seconds

The situation where lots of blue colors are represented is not a problem. You must watch the trend to see whether some of them turn to orange or red. In case of a failure, the number of active transactions increases and the percentage of red color increases. On the other hand, a very responsive system with a high number of process transactions (TPS) might have a low number of active transactions.

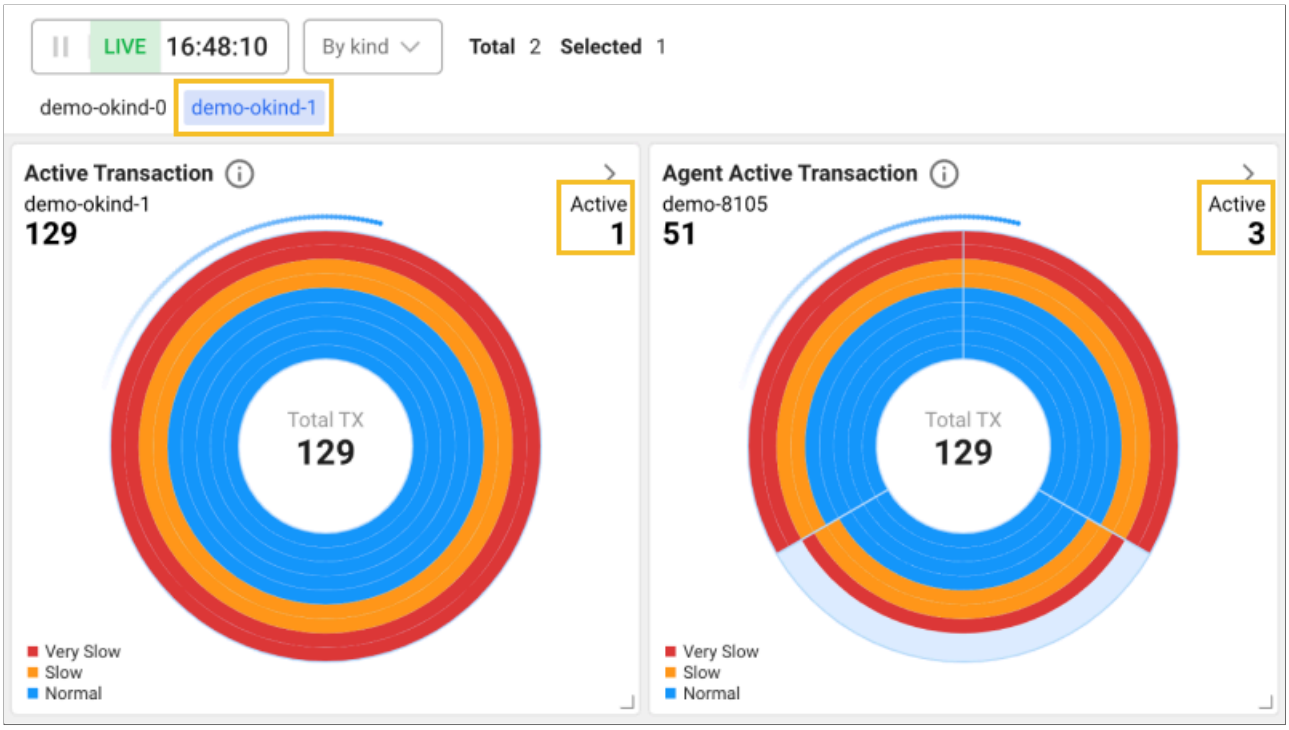
- On the upper left, the agent name with the most active transactions appears.
- The middle number is a sum of active transactions of all agents.
- Two bars around active transactions rotate around the chart at a three-step rate based on the throughput (TPS).
- When you select a divided arc area, a pop-up window appears listing the transactions in progress on the

agent. To check the transaction details, select an item in the transaction list.



- If the item selected in the transaction list is a transaction that has already ended, select **Show trace**. The **Transaction information** window appears. For more information about the **Transaction information** window, see the following.
- For more information about transactions, see the following.

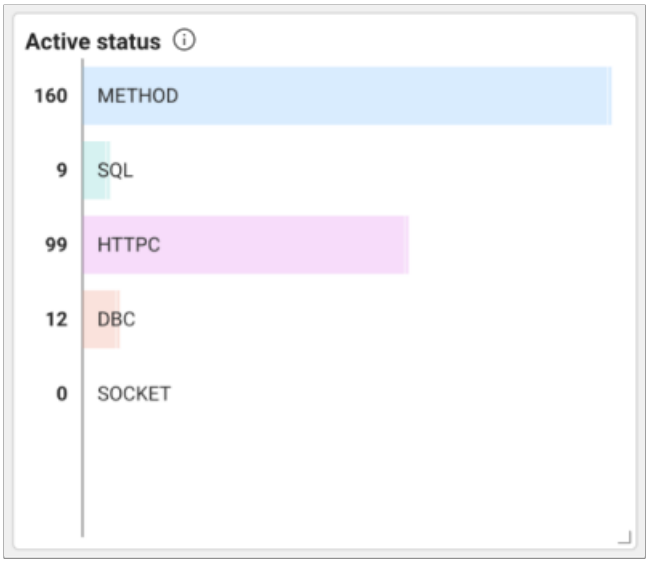
Agent Active Transaction



The **Active Transaction** widget displays on a chart the data merged based on the criteria selected in the agent selection options. On the other hand, the **Agent Active Transaction** widget allows you to individually check the agents included in the selected agent classification criteria.

Right-click the dashboard and then select the **Agent Active Transaction** widget from the widget list.

Active status



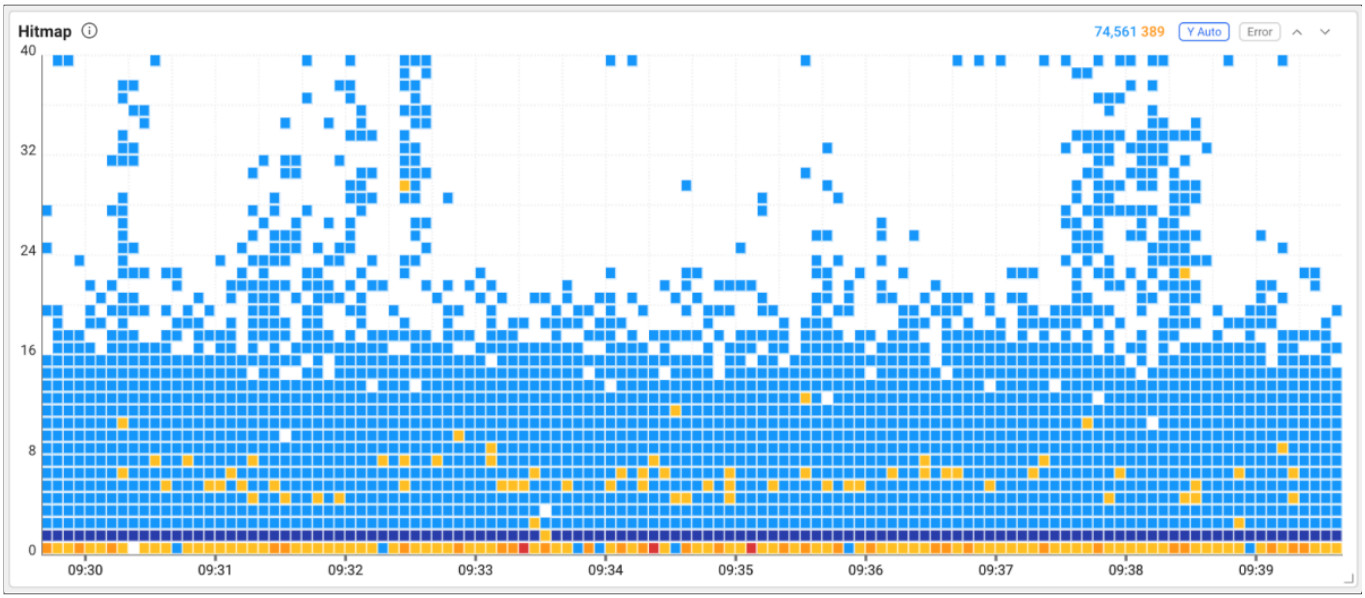
All agents in the project collect the active transaction count every 5 seconds, along with the progress status. The progress status is classified into **METHOD**, **SQL**, **HTTPC**, **DBC**, and **SOCKET**.

- **METHOD**: Status in which the method is executing. Normal situations are displayed in blue.
- **SQL**: SQL is running. If the **SQL** number increases, check for problems connecting to the database server.
- **HTTPC**: It is calling any external API. If the **HTTPC** number increases, it means that the server connected to outside is not responding normally.
- **DBC**: The transaction is attempting to get a new connection from the connection pool. If DB connection pools are insufficient, a delay occurs whenever a new connection request occurs, causing the performance failure. In this case, the **DBC** number increases.

  > ⓘ For more information about the DB connection delay, see the following.

- **SOCKET**: Indicates the TCP connection attempt to an external system. If the **SOCKET** number continues to increase, there is a high possibility of failures due to disconnection of the external system.

Hitmap

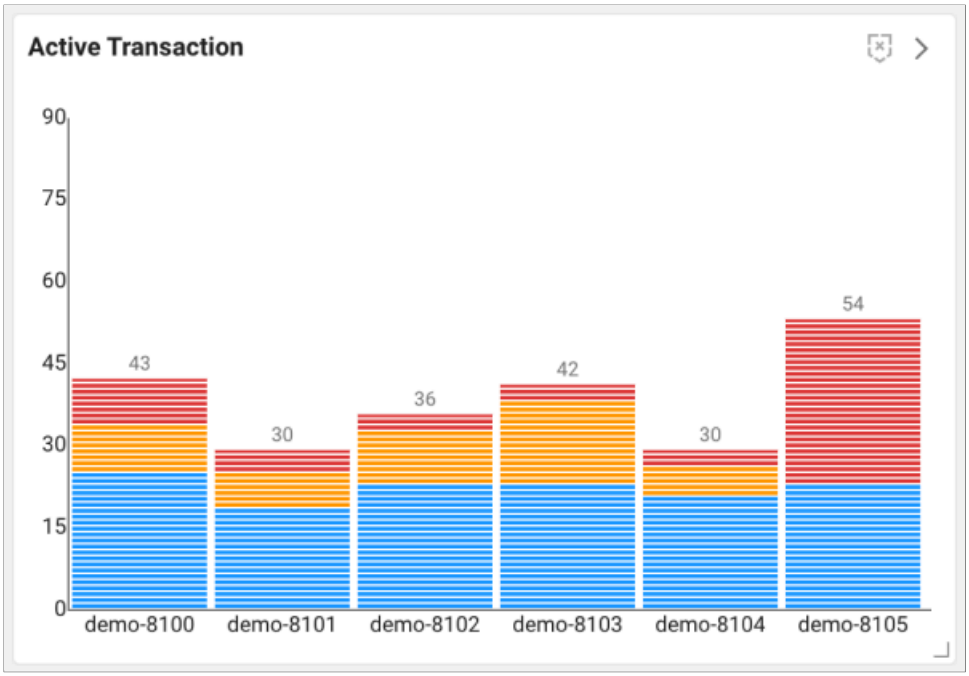


The **Hitmap** chart represents the distribution of response times of completed transactions. The horizontal axis is the transaction end time. The vertical axis is the execution time. Transactions with longer execution times are at the top of the distribution. If you drag the chart area in **Hitmap**, the **Trace analysis** window appears that you can see the transaction information details. For more information about **Trace analysis**, see the following.

- Normal transactions are displayed in light blue → blue → Indigo in order.
- Erroneous transactions are displayed in yellow → orange → red in order.
- The number on the upper right indicates the transaction count/error count in the chart.
- If **Y Auto** is selected, the Y-axis (execution time) is automatically adjusted.
- If **Error** is selected, only the error transactions appear.
- Select ∧ or ∨ to zoom in/out the chart from 5 seconds to 80 seconds.

> ⓘ
> - The detailed information for the transactions with no error whose execution time is less than 500 ms is collected once every 5 minutes for each URL.
> - It does not affect the statistics such as TPS and average response time.
> - You can set a policy through the agent configuration. For more information, see the following.
> - See the following document for the procedure to analyze transactions through the hitmap widget.

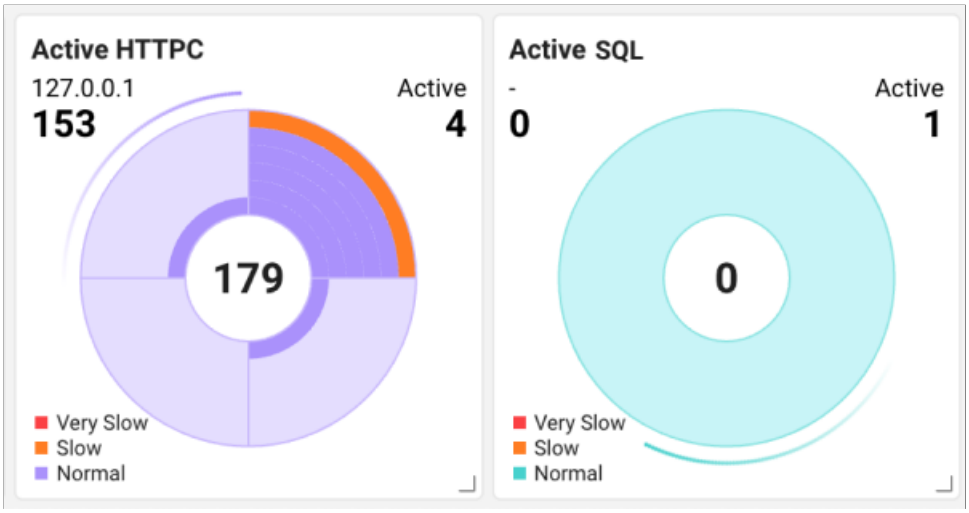
Active Transaction Equalizer



It displays a list of all agents and the total number of active transactions. Hover your mouse over each graph on the chart to display the transaction speed. The transaction speeds are classified into Normal, Slow, and Very Slow.

- To display or hide inactive agents, select .
- If you select >, the **Active Transaction** menu appears.

Active HTTPC/SQL



The **Active HTTPC/SQL** widget monitors the performance of external calls (HTTP calls and database queries) occurring in the application in real time.
Unlike the **Active Transaction** widget, which shows all transactions, this widget is specialized for outbound calls, allowing you to quickly identify external system bottlenecks. Placing the two widgets together helps distinguish whether transaction increases are caused by external system delays or internal logic issues, enabling more effective multilayered analysis.

Ongoing HTTPC/SQL operations are visualized as an **Arc Equalizer** chart, showing latency status by host or database URL. The arc is divided by the number of hosts or database URLs, and every 5 seconds it displays the number of requests being processed on the current server and their response times. Requests detected in 5-second intervals are classified into intuitive colors according to their severity.

- Red (Very Slow): 8 seconds or more
- Orange (Slow): 3 to 8 seconds
- Normal: 0 to 3 seconds
  - Purple (HTTPC Normal)
  - Mint (SQL Normal)

A large number of Normal values is not a concern. You need to monitor if some values turn orange or red. In failure scenarios, the number of ongoing transactions increases and the proportion of red rises. Conversely, in very fast systems, TPS may be high but the number of ongoing transactions may remain low.

This widget is especially useful in microservice environments to track HTTP communication delays between services or monitor DB performance individually in systems using multiple databases. For services with many external API calls, it allows monitoring of response times by API provider, making dependency management effective.

Arc Equalizer Chart

This is the core visualization element of the widget, dividing ongoing HTTPC/SQL operations by host or database URL. If issues are detected in the overall transactions, this chart helps pinpoint specific external bottlenecks.

- **Top Left Indicator**: Displays the host/database URL with the most active transactions for easy identification of key targets.
- **Center Number**: Shows the total sum of active HTTPC/SQL.
- **TPS-Based Animation**: Bars around the equalizer rotate at three speeds depending on throughput (TPS), visually expressing system load.
- **5-Second Refresh**: Provides near real-time monitoring for immediate response.
- **Arc Segmentation**: Divides arcs by host/database URL for quick identification of where delays are occurring.

Host/Database URL Details

Selecting a segmented arc area opens a popup showing the **Top 100 ongoing transactions** for that host or database URL.

- **Transaction List**: Displays the Top 100 ongoing transactions for the selected host or database URL.
- **Trend Chart**: Shows TPS, response time, error count, and error rate trends over the last 10 minutes.
- **Transaction Details**: Selecting a transaction from the list opens detailed information.
  - For already finished transactions, select the **View Transaction** button to open the **Transaction Info** window.

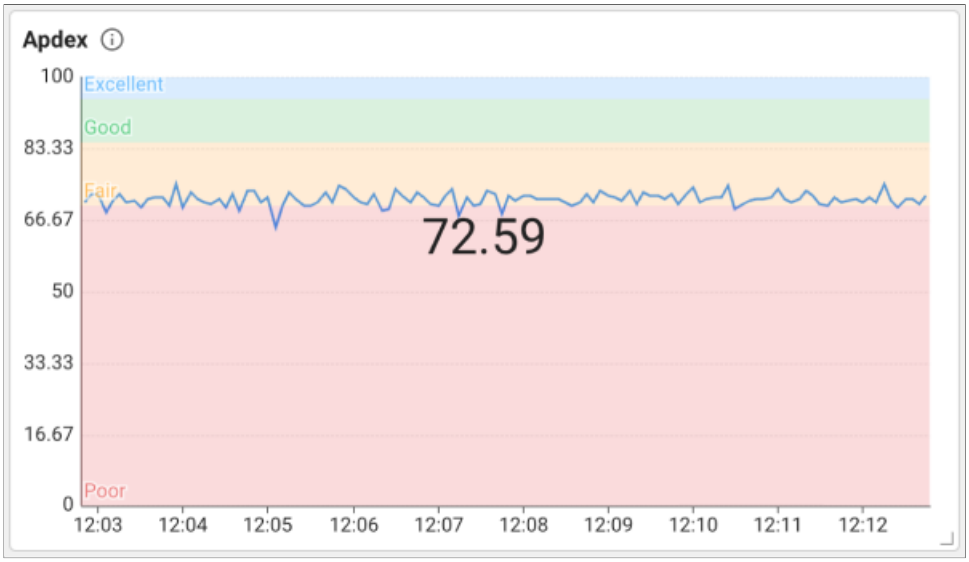
1. In the **Active HTTPC Top 100** popup, check the transaction list and trends.
2. Select a transaction that requires detailed analysis.

   a. In the Arc Equalizer, click the arc for the host or database URL you want to analyze.
3. In the Transaction Info window, review the details.

Service-related widget

The service-related widget can be used as the metric for application performance tuning.

Apdex



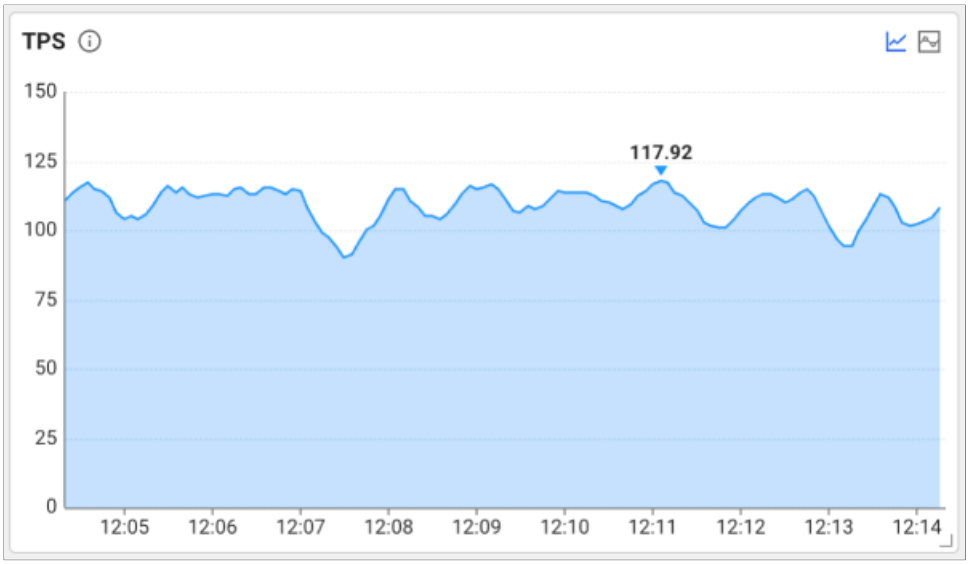
Application Performance Index (Apdex) is the indicator of the application performance. You can measure the customer satisfaction about your web applications. It can be used as an indicator of user satisfaction and the value is between 0 and 1.

(Number of satisfied cases + (Number of tolerating cases * 0.5))/Total number of requests

- S (Satisfied): No problem in business ≤ 1.2 seconds (Default value)
- T (Tolerating): Business is possible with any delay ≤ 4.8 seconds (S * 4)
- F (Frustrated): Business impossible > 4.8 seconds (exceeds T and error)

> The default value (S) can be modified in **Management** > **Agent CONFIG.**. The value of the setting key is `apdex_time` . It is available only in the agent 2.0 or later.

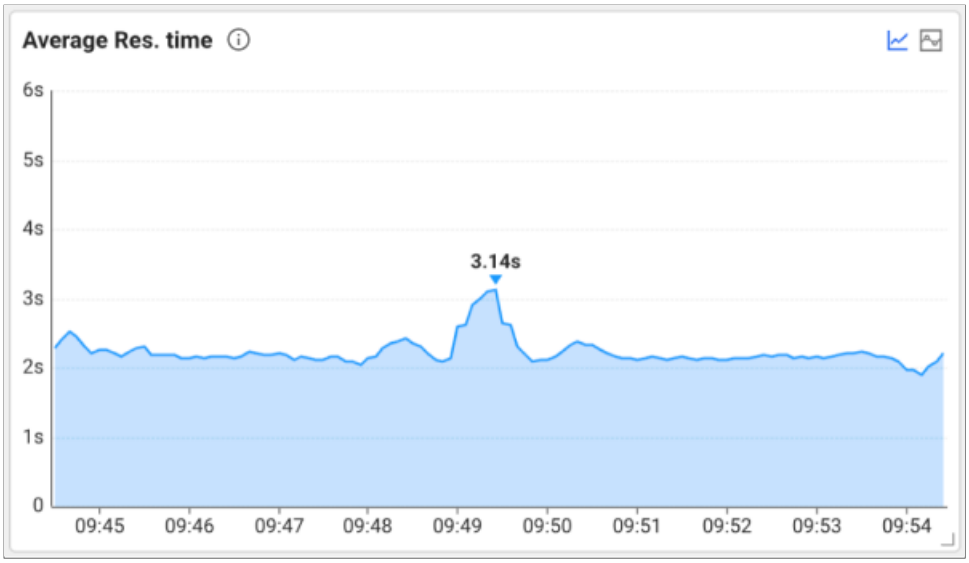
TPS



Transaction Per Second (**TPS**) indicates the number of transactions processed per second and is the reference for service performance metric. The number of transactions processed every 5 seconds is converted to the number of transactions per second and is expressed with a chart. It displays TPS for the last 10 minutes.

> For more information about analyses for TPS, response time, and CPU utilization, see the following.

Average Res. time

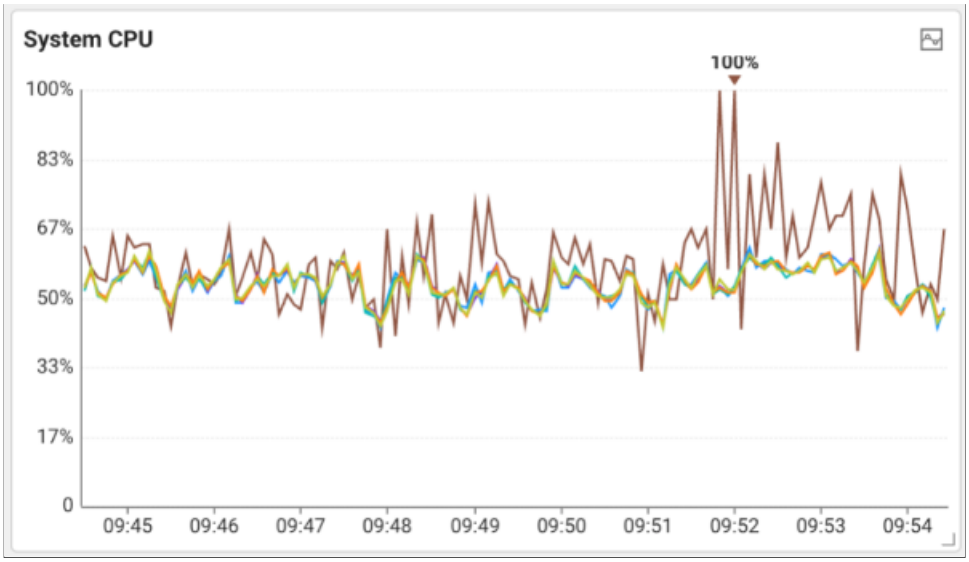


It monitors the times needed for the application server to return the request result. **Average Res. time** is meaningful as a tuning metric. The WhaTap's monitoring service calculates the average response time for transactions every 5 seconds.

> For more information about analyses for TPS, response time, and CPU utilization, see the following.

Resource-related widget

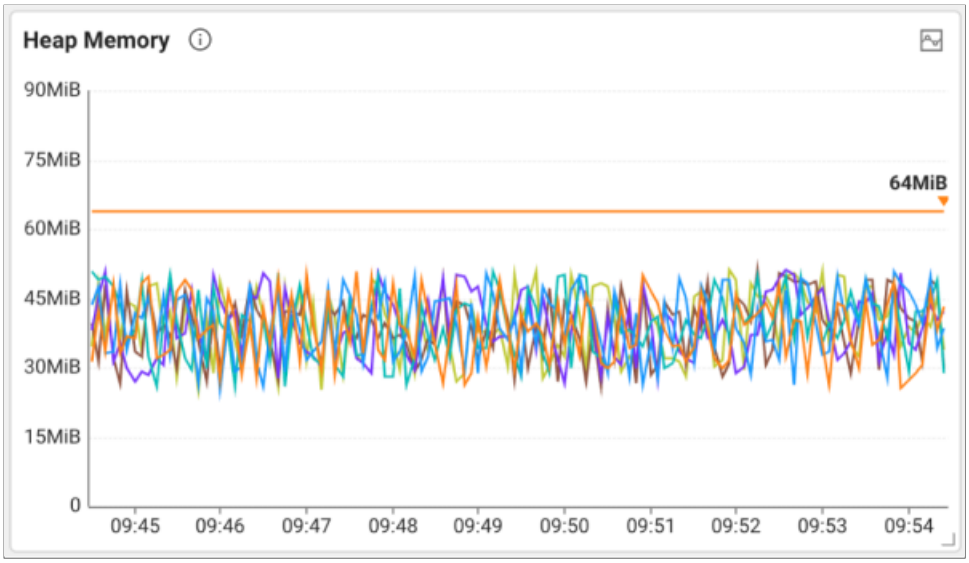
System CPU



It is the CPU usage of the application server. You can see the change trend in CPU usage in real time.

> For more information about analyses for TPS, response time, and CPU utilization, see the following.

Heap Memory



You can see which servers are at risk by displaying the maximum available memory and the current memory for each server. The changes in memory usage over time can be seen in real time.

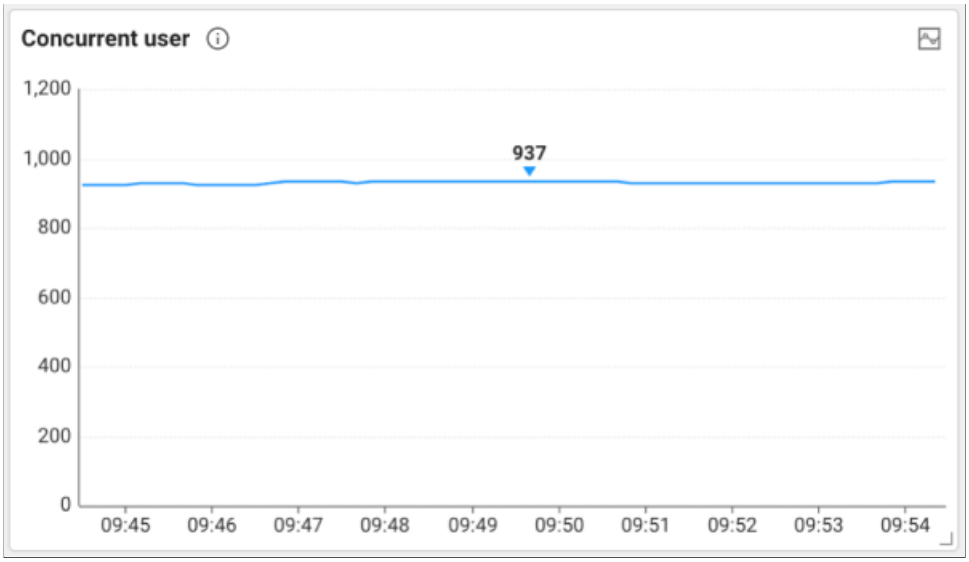
The memory line chart usually ripples continuously. It increases when using memory to process the requests from the application server. It decreases when memory is cleaned up through GC (Garbage Collection). **Heap Memory** metric is displayed with average values.

For more information about the heap memory chart analysis, see the following links.

- Analysis of Java heap memory chart: Observing the Ch.1 heap chart
- Analysis of JAVA heap memory chart: Analysis of Ch.2 memory leak and heap dump

User-related widget

Concurrent user



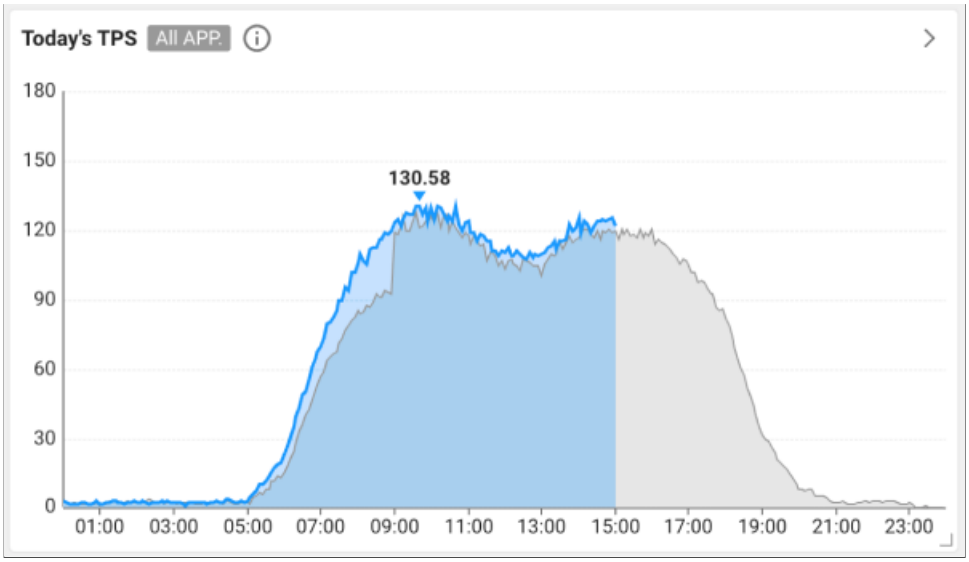
It displays the number of real-time browser users. Every 10 seconds, users who made transactions within the last 5 minutes are counted and displayed without duplicates. Users are counted based on their browser's IP.

- In the agent settings, the IP or cookies can be used to distinguish users. For more information, see the following.
- In general, the connected user means a connected user who has access to the current network. However, web systems are different because they use connectionless networks. It does not mean that you are connected, but rather the user who sent the recent request can be measured.
- Real-time (concurrent access) users indicate the users who are using the system in the same time zone. There are various ways to count or measure real-time (concurrent access) users.
- The number of users is calculated by ** HyperLogLog** rather than simply adding up. ** HyperLogLog** is a probabilistic data structure that estimates the number of elements in a set with very little memory.
- For more information about the configuration of agent options for aggregating the user count, see the following.

Comparison widget on a daily basis

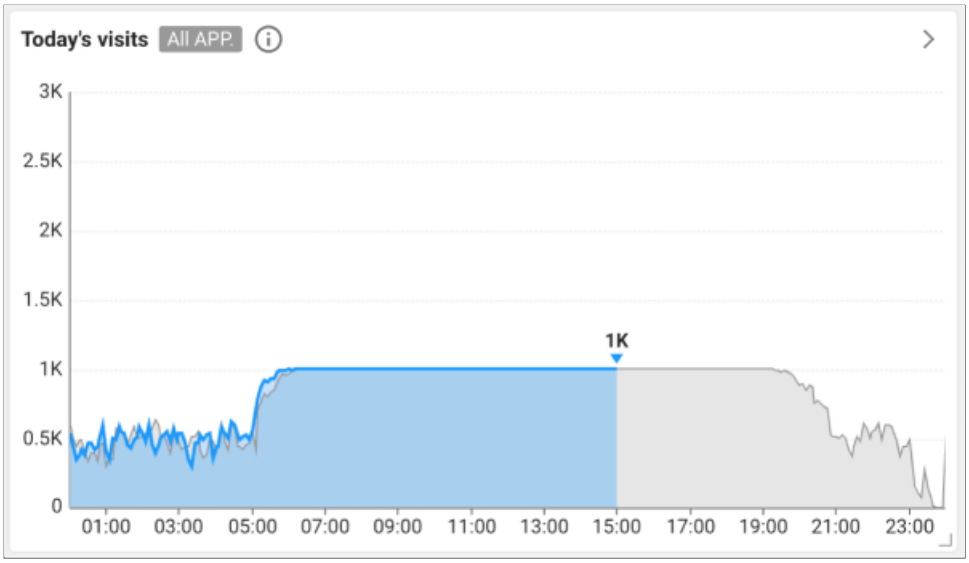
> The widgets marked with the `All APPS.` tag display the data for all agents.

Today's TPS



It displays the TPS for today by hour. You can check it by comparing with the TPS (gray) in the same time zone of yesterday. If you click ›, **Stat/Report** > **Daily APP. Status** appears.

Today's visits



This metric shows the number of **Concurrent user** for the day by time zone. You can check it by comparing with the user count (gray) in the same time zone of yesterday. If you click ›, **Stat/Report** > **Daily APP. Status** appears.

Real-time Alert

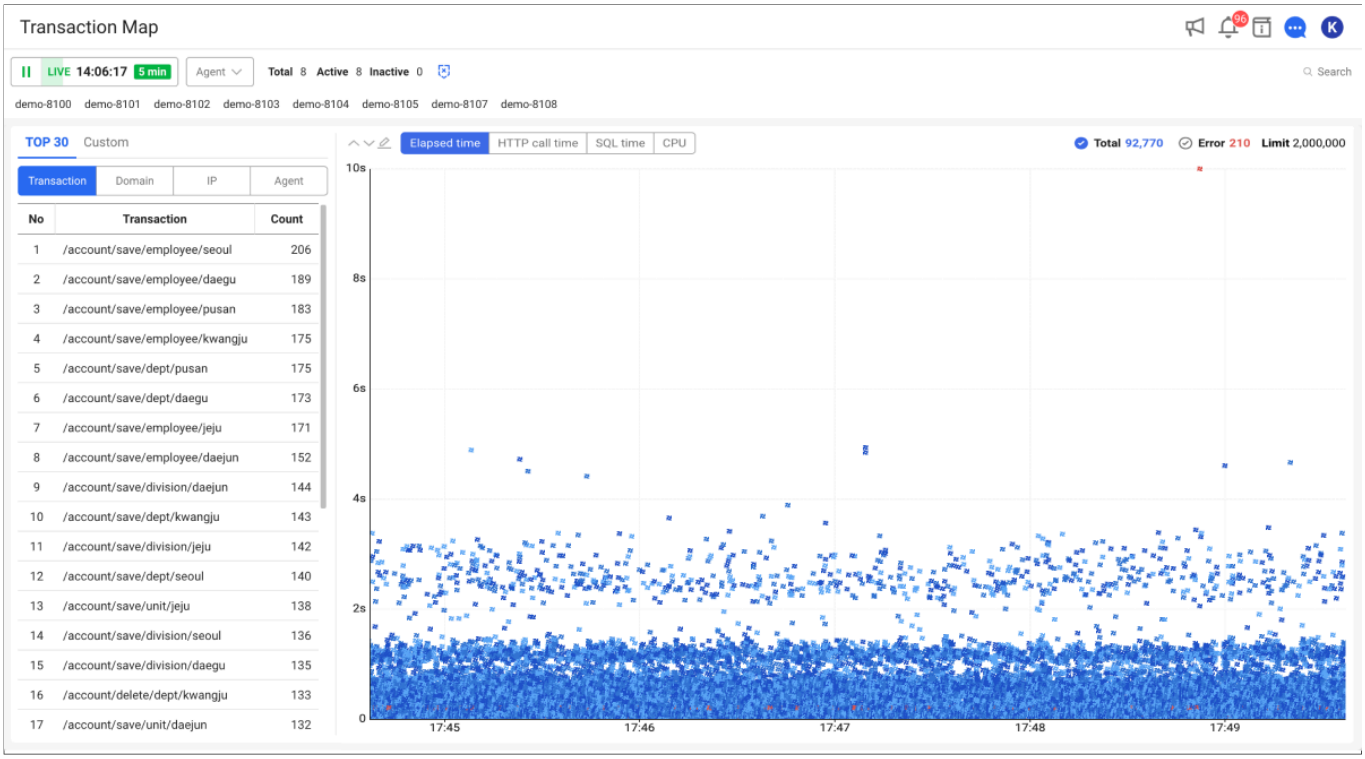
If you select **Alert Status** on the upper right of the screen, the recent events appear. You can turn on or off the browser notifications by clicking the toggle menu.

> For more information about basic elements in the fixed menu area at the top of the screen, see the following.

Transaction Map

Home > Select Project > **Dashboard** > **Transaction map**

The **Transaction map** chart displays the response times for each completed transaction in the distribution chart form. Same as the hitmap, you can find and analyze problems based on the distribution patterns. **Hitmap** displays the grouped transactions every 5 seconds, but **Transaction map** displays transactions individually.



How is it different from the hitmap analysis?

The hitmap uses the aggregated data to display the overall trend over a long-term period. On the other hand, the transaction map uses individual transactions to check the transaction distribution within a short-term period with no time constraints. It also differs in that it allows users to apply various filters to select only the desired transactions and observe their trends.

- To analyze the distribution of the long-term, high-volume transactions, use the hitmap analysis (**Analysis** > **Hitmap**).

- To do performance test, or analyze the transaction distribution within 1 to 5 minutes upon a failure, or check the transaction distribution without limiting the execution time, check the **Transaction map**.

- For more information about **Hitmap**, see the following.
- For more information about the analysis method of the **HITMAP transaction** chart, see the following.
- If the user's PC time is set to earlier than the server time, or the difference between two times is 5 minutes or more, the data may not be displayed in the chart.

Analyzing a trace

If you drag the chart area of Transaction map, the **Trace analysis** window appears where you can check the transaction details. You can see the transaction list and the steps under each transaction at once.

After dragging the chart area, you can see up to 1,000 items in the **Trace analysis** window.

For more information about **Trace analysis**, see the following.

Changing the lookup criteria

Select the chart area of Transaction map and then the button at the upper left to change the data lookup criteria.

- **Elapsed time**: The data is displayed based on the transaction execution time.
- **HTTP call time**: The data is displayed based on the response time against the HTTP call.
- **SQL time**: The data is displayed based on the SQL execution time.
- **CPU usage time**: The data is displayed based on the CPU used time.

Filtering the searched results

In the chart area of Transaction map, you can distinguish the results by selecting Total, Error, and Read phrases in the upper right corner.

Total: Displays total transactions

Error: Only the transactions in which errors occurred, are displayed.

Read: The number of transactions read from the server is displayed.

For example, if you select an error and then drag a transaction on the chart, the **Trace analysis** window appears to display a list of erroneous transactions. To view all transactions, click Total.

Adjusting the Y-axis

- You can easily change the maximum value of the Y-axis by selecting ^ or v on the upper left of the chart. You can also access the same function by pressing the up or down button on the keyboard.

  For more information on how to use the ^ or v button or keyboard buttons depending on the Y-axis section, see the following:

  - The segment from 10 to 50 seconds is changed to the 10-second interval.
  - The segment from 50 to 300 seconds is changed to the 50-second interval.
  - For the segment of 300 or more seconds, it is changed to the 100-second interval.
- You can change the maximum value of the Y-axis by directly entering it. Select ✎, enter a desired value, and then select **Apply**.

The maximum number of transactions (**Limit**) displayed on the chart in Transaction map is 2,000,000.

When you apply filters for a search, the Read value may be greater than the Total (number displayed on the screen).

Viewing the past data

Transaction map provides the real-time monitoring by default. By default, the data can be viewed in real time for 5 minutes. Click the green button in the time selector to select a desired lookup time.

To view the historical data, click ⏸ in the time selector. It changes to period viewing mode and you can view the transaction data for up to 1hour. The chart data of **Transaction map** based on the date and time set by the user is refreshed.

You can stop the query at any time while searching the past data. _ To change the filters, first stop the query, and then change the filters. _ If you press the Search button again without changing the filters, it continues searching from where it was stopped.

To switch to the Realtime mode after a historical inquiry, click ▶.

> • The range of the historical inquiries may differ depending on the transaction volume.

Applying filters

In case of viewing the past data, you can filter out the data by applying filters.

1. When you click the Pause (⏸) button, the filter input box appears.
2. If you click the filter input box, the list of filters that can be selected appears as follows.
   - General filters

Item	Description
Login ID	Viewing with the login ID of the user
WClientID	Searched by the unique identifier per client (Note: Searching is performed by exact match.)

Item	Description
Transaction ID	Unique ID per transaction (Note: Searching is performed by exact match.)
Referer	Searching is performed using the Referer (referring page address) delivered by a transaction request.
Multi Transaction ID	Unique ID that groups multiple transactions (Note: Searching is performed by exact match.)
Custom TID	Searching by the custom transaction ID
Error(Class, Step, Message)	Searching by error type, step, and message in transactions (partial match included)
City	Searching by the user's town for transactions
Country/Region	Searching by the user's country for transactions
HTTP Method	HTTP request method (GET, POST, etc) (Note: Searching is performed by exact match.)
UserAgent	Searching by browser or client information (user agent) for transaction requests

- STEP Filter
  Step indicates the profile data collected while performing transactions. When dragging the hitmap, the sequential data displayed in Table View or Tree View by clicking a transaction in the transaction list is called Step.

```
| Item      | Description                                                                                       |
| --------- |
------------------------------------------------------------------------------------------------------------------ |
| DBC       | Step that contains the method running while performing transactions
|
| HTTPC     | Step that contains the HTTP call
```

```
|
| MESSAGE   | Step that contains various texts while performing transactions
|
| METHOD    | Step that contains the method running while performing transactions
|
| SQL       | Step that contains executed queries
|
| SQL_BIND  | Data that does not exist actually but is included in the SQL step. Step with which bind variables and substitution variables can be viewed during the SQL call (Note: Searching is possible with the authority and decryption key to view sensitive information.) |
| ALL       | Step with which all the above steps can be searched (Note: Searching is possible with the authority and decryption key to view sensitive information.)                                  |
```

- Number input filter

Item	Description
Elapsed time	Total time taken for the transaction to be performed
HTTP Call Time	Response time for HTTP Requests
HTTP Call Count	Number of HTTP requests while performing transactions
SQL time	Total SQL execution time for transactions
SQL Count	Number of SQL queries executed for transactions
Status	Transaction status(e.g. Normal, Error, etc.)

3. Enter the desired values for filtering. For filtering, enter all values for the filter key, operators, and values. If there are any filters that have not been entered yet, searching is not possible.
4. Aftering entering all filters, click  .

It updates the filtered data in the Top 30 list and the **Transaction map** chart.

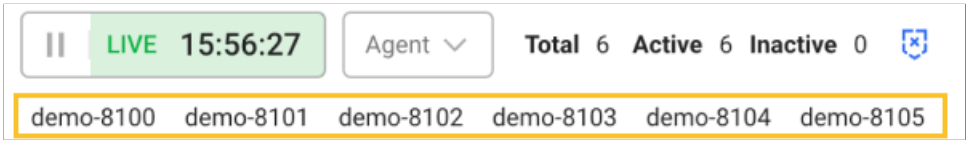
> • You can perform filtering by setting multiple conditions. Added conditions are applied based on

&& . For example, if the transaction is /account/save/employee/seoul and the elapsed time is more than 3 seconds, you can apply as follows:

Transaction = /account/save/employee/seoul Elapsed time (ms) >= 3

Agent Check

Checking the agent connection status



On the upper left of the screen, the right area of the time selector provides information that allows you to check the status of agents connected to the project. This allows you to immediately check whether the target application server is running.

- Total: Number of all agents connected to the project
- Active: Number of active agents
- Inactive: Number of inactive agents
- : Can display or hide inactive agents.

Agent-based monitoring

By default, the chart displays the metrics collected from all agents in charts and you can also view the data by agent. Select one or more agents under the time selector. The chart is updated with the transactions of the selected agent.

To select all agents again while one or more agents is selected, deselect them or click Total.

If there are numerous agents connected to the project, it is efficient to set the agent names to be short.

> For more information about the agent name setting, see the following.

Monitoring each classified agents

You can easily monitor multiple agents by categorizing them in Agent Settings.

- **Agent**: Individual Agent
- **By kind**: Group classified by the agent type (e.g. web server, database). Set it to `whatap.okind`.
- **By server**: Group categorized by the server (node). Set it to `whatap.onode`.
- **Agent by kind**: List of agents classified as `whatap.okind`.
- **Agent by server**: List of agents classified as `whatap.onode`.

> For more information about the agent configuration, see the following.

Searching the agent

If the agents connected to the project are numerous and difficult to find, use the agent search function. If you click **Search** on the upper right of the screen, a box appears where you can enter a string.

The agents that match the entered string appear in the list. To filter the data in the dashboard based on the searched agents, click **Select** on the right of the text entry box.

To reset the selected agent entries, click **Close** on the right of the input field.

Checking the upper list

In the Top 30 section, the list of top 30 items with the highest transactions performed based on **Transaction**, **Domain**, **IP**, and **Agent**.

- **Transaction**: Top 30 items, aggregated by the transaction URL
- **Domain**: Top 30 items aggregated based on the Internet address specified in the address of the IP connected by the client

- **IP**: Top 30 items, aggregated by the client IP address
- **Agent**: Top 30 items aggregated by the agent included in the project

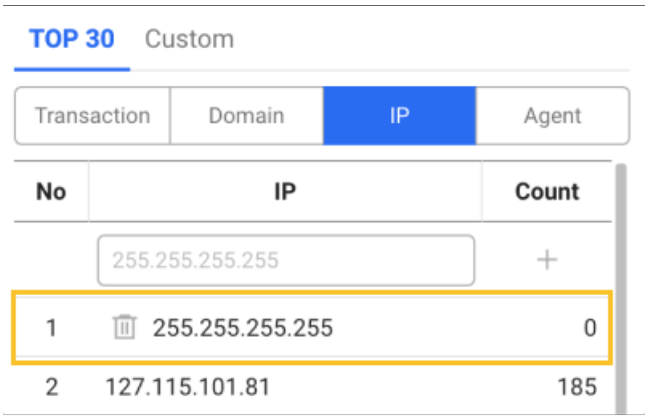
If you select each item from the list, the data is applied based on the item selected in the Transaction map chart. To select multiple items, hold down the `Ctrl` (Windows/Linux) or `CMD` (Mac) key and then click desired items.

> You can move an item in the list of the Top 30 section by pressing the up or down arrow button on the keyboard.

Add custom IP address

When viewing the number of transactions executed based on **IP** in the **TOP 30** section, the function is provided to check constantly by adding a specific IP address.

1. In the **TOP 30** section, select **IP**.
2. Enter a specific IP address in the field at the top of the list.
3. Click ＋.



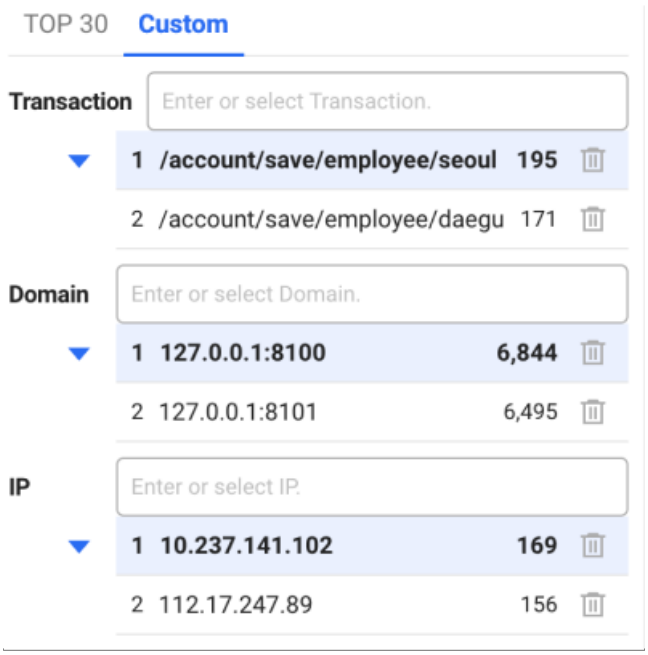
The specified IP address appears at the fixed top of the list.

> To delete any added IP address, click 🗑.

Custom monitoring

It provides the feature to check only the selected items in the response time distribution chart by adding specific transactions, domains, and IPs to the **Custom** tab. This allows users to efficiently monitor and analyze specific problematic items.

1. Click the **Custom** tab.
2. **Transaction** or **Domain**, **IP** can be entered directly, selected from the **Top 30** list, or searched.
3. Click the added item.



You can see the response time distribution for the selected items filtered in real time on the chart on the right.

> To delete any added custom items, click 🗑.

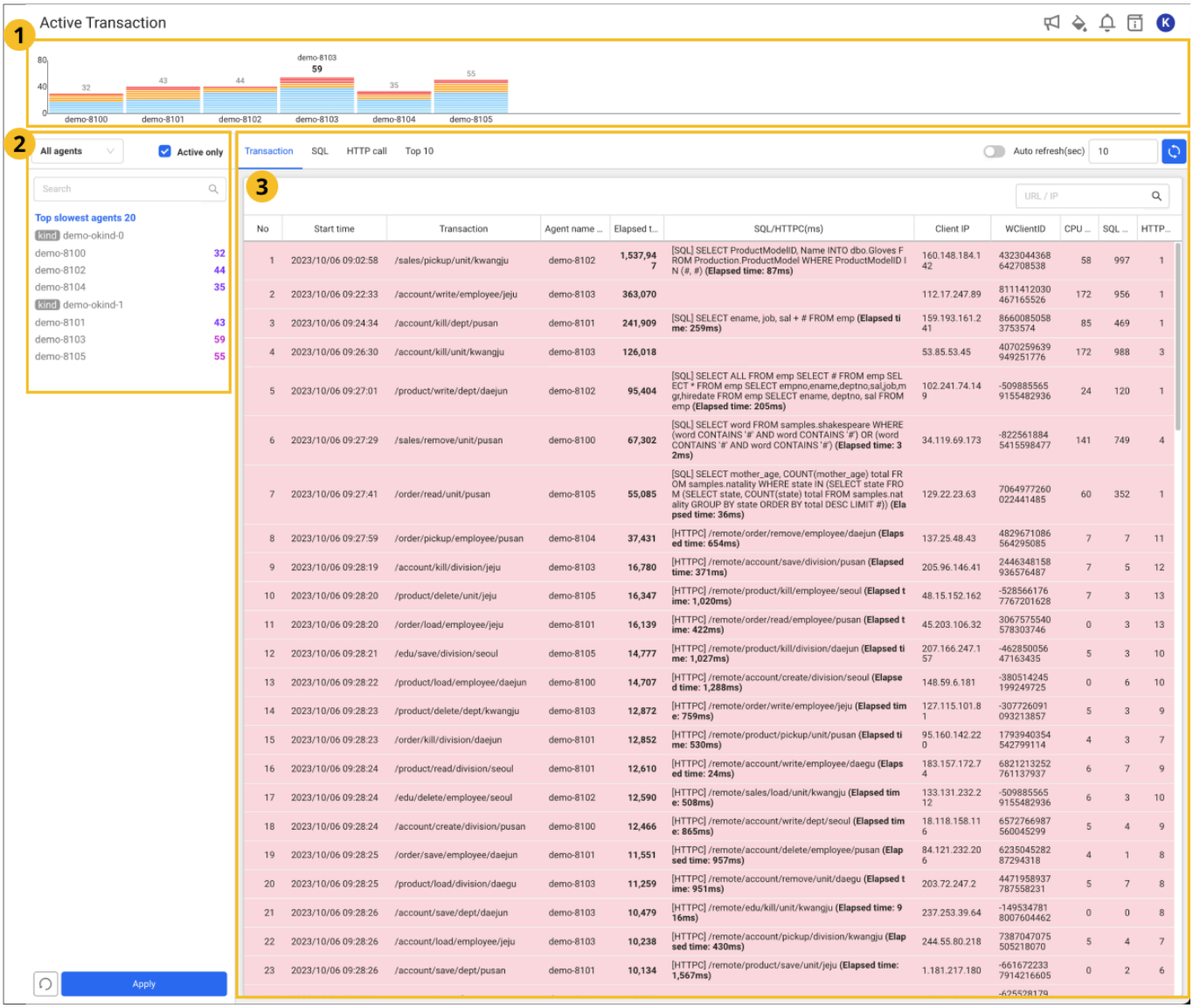
Active Transaction Dashboard

Home > Select Project > **Dashboard** > **Active Transaction**

You can quickly recognize abnormal transactions by using URLs, SQLs, and HTTP calls of the active transactions, and can analyze the delayed sections in detail through the transaction trace details.

> By adding a database project via the **Correlated Project Management** menu, you can view the SQL performance of the database linked to the current transaction. For more information about the **Correlated Project Management** menu and linked project analysis, see the following.

Screen layout



1 Active transactions for each agent

It displays a list of all application servers and the total number of active transactions. Hover your mouse over each graph on the chart to display the transaction speed. The speed of the transaction state is classified into Normal, Slow, and Very Slow.

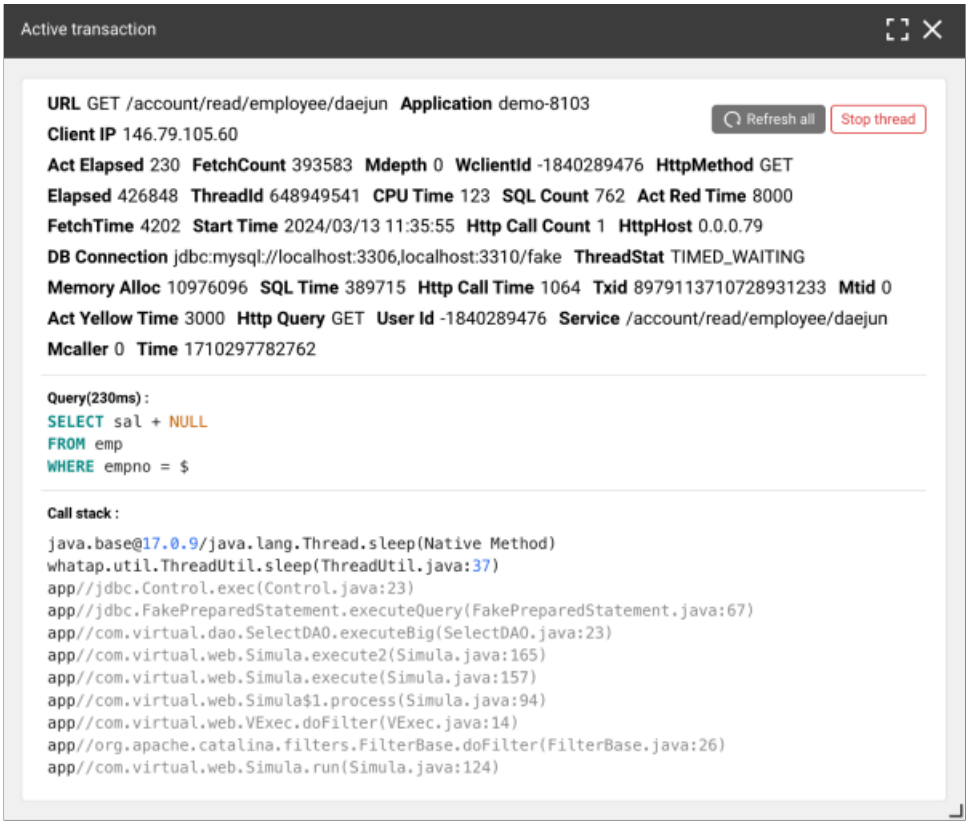
2 Agent List

It displays the application server list and the total number of active transactions. If you select the application server, In the 3 transaction list, the detailed information appears.

3 Transaction List

It displays the detailed information of active transactions including URLs, SQLs, and HTTP calls. If you select the Top 10 Items tab, you can see a table of items sorted by the elapsed time in the ascending order. Likewise, the speeds of the transaction state are classified into Normal, Slow, and Very Slow.

When you select an item in the transaction list, the **Active transaction** window appears where you can see the details on the selected transaction.



- You can see detailed information of the running transactions, external calls, queries, call stacks, and the like.
  - **Refresh all**: You can update information of the running transactions.
  - **Stop thread**: You can stop a transaction in progress. To stop threading, **Secret key** is required. For more information about **Secret key**, see the following.

- **Correlated DB session**: You can query the SQL performance of the database linked to the active transaction. It appears only when the selected transaction is the one that executed SQL as a linked project. For more information about addition and management of the linked projects, see the following.

> **Stop thread** forcibly stops threads running in the application. However, this feature may not work in all cases. For example, if a thread performs an IO operation, the thread cannot be stopped immediately.
>
> **Stop thread** attempts to terminate the thread even if it is performing an IO operation. However, the thread does not stop until it finishes its IO operation. The thread accepts the termination request, but it does not stop until its operation is completed and it goes to the next work.
>
> In this case, forcible termination is not recommended because it may cause problems with business logic and cause unexpected operations. If you need to stop the thread, stop and restart the entire application.

- If a transaction has already ended, the **Show trace** button appears. When you select the button, the **Transaction information** window appears. For more information about **Transaction information**, see the following.

Agent list filtering

In 2 **Agent List**, you can select or search the filtering targets and see them individually in the transaction list.

- To see the active agents, select the **Active only** checkbox.
- Click the **All agents** list box. You can view the agents by category.
- To see active transactions by agent, select an agent and then click **Apply**. To cancel the selection, select .

Automatic refresh

The active transactions can be automatically updated at the set intervals. Select **Auto refresh(sec)** to activate the option, and then enter the refresh interval in the field on the right.

To deactivate the auto refresh setting, select **Auto refresh(sec)** again.

- The time is updated in seconds.
- To update the transaction list when the auto refresh is deactivated, select .

Transaction list filtering

In 3 **Transaction List**, you can search for URL or IP for filtering. Enter a filtering target in the URL/IP field on the upper right and then press Enter or select . It filters with the transaction list that includes the input text.

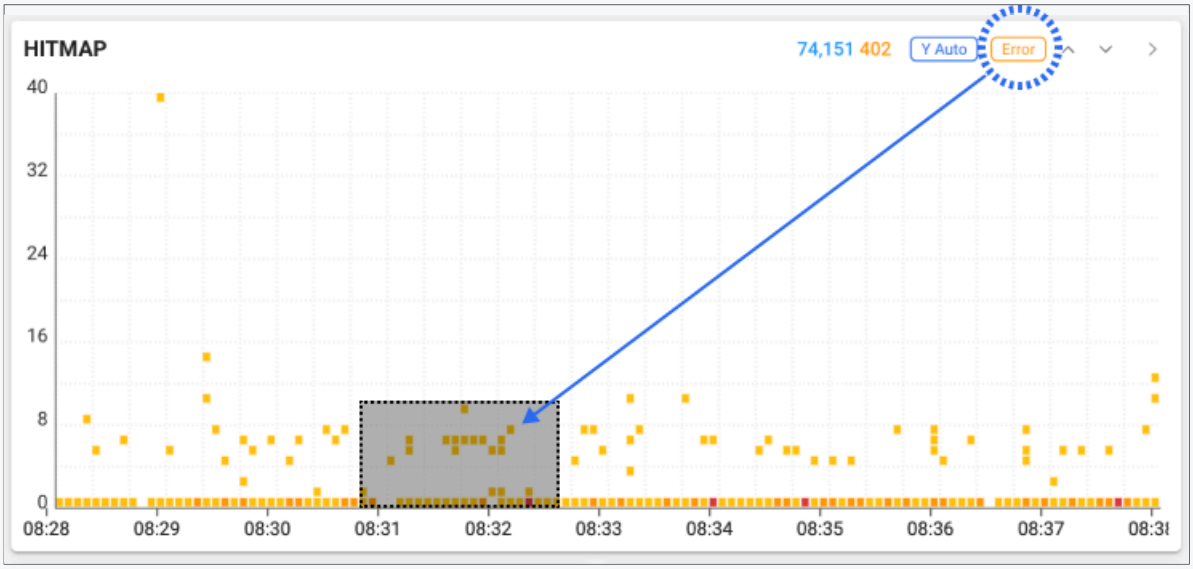
To initialize the filtered list, delete text in the URL/IP field on the upper right, and then press Enter or select .

For more information about active transaction settings, see the following.

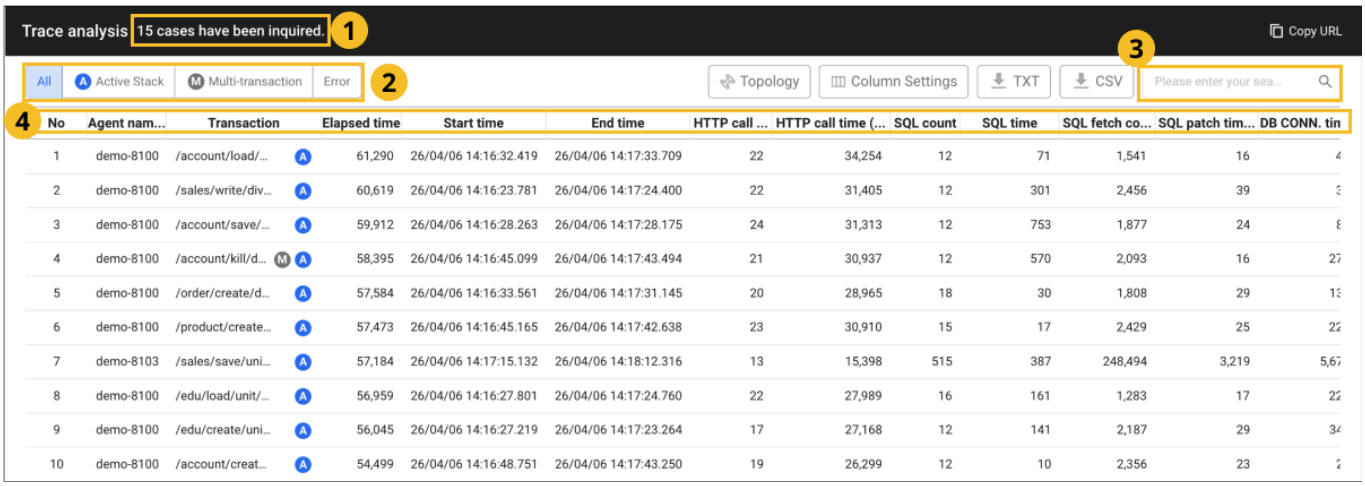
Hitmap/Trace Analysis

Through the **Hitmap** widget in **Application Dashboard**, you can open the **Trace analysis** window to check the transaction list and step information of each transaction at once. Drag the desired area on the hitmap widget. The **Trace analysis** window appears that contains the transaction list and sub-steps of the selected area.

To classify and see only the transactions with errors, select **Error** on the upper right of the **Hitmap** widget in **Application Dashboard** and then drag the chart area.

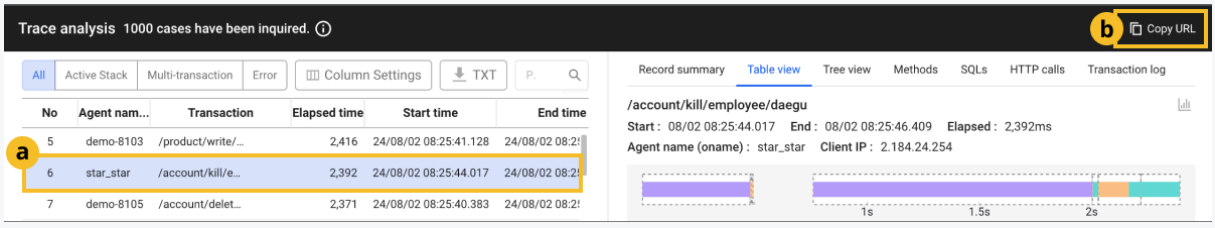


Basic screen guide



- 1 **Transaction lookup count**: You can see the number of transaction lists in the dragged area on the hitmap widget. You can query up to 1000 cases.
- 2 **Transaction classification**: By separating Ⓐ **Active Stack**, Ⓜ **Multi-transaction**, and **Error**, you can view the list.
- 3 **Transaction search**: You can search the transaction list by transaction name, agent name, and error message.
- 4 **Sorting the list**: You can sort the lists of transactions by the selected item by clicking a header column in the table.
- The transactions with errors appear in red in the list.

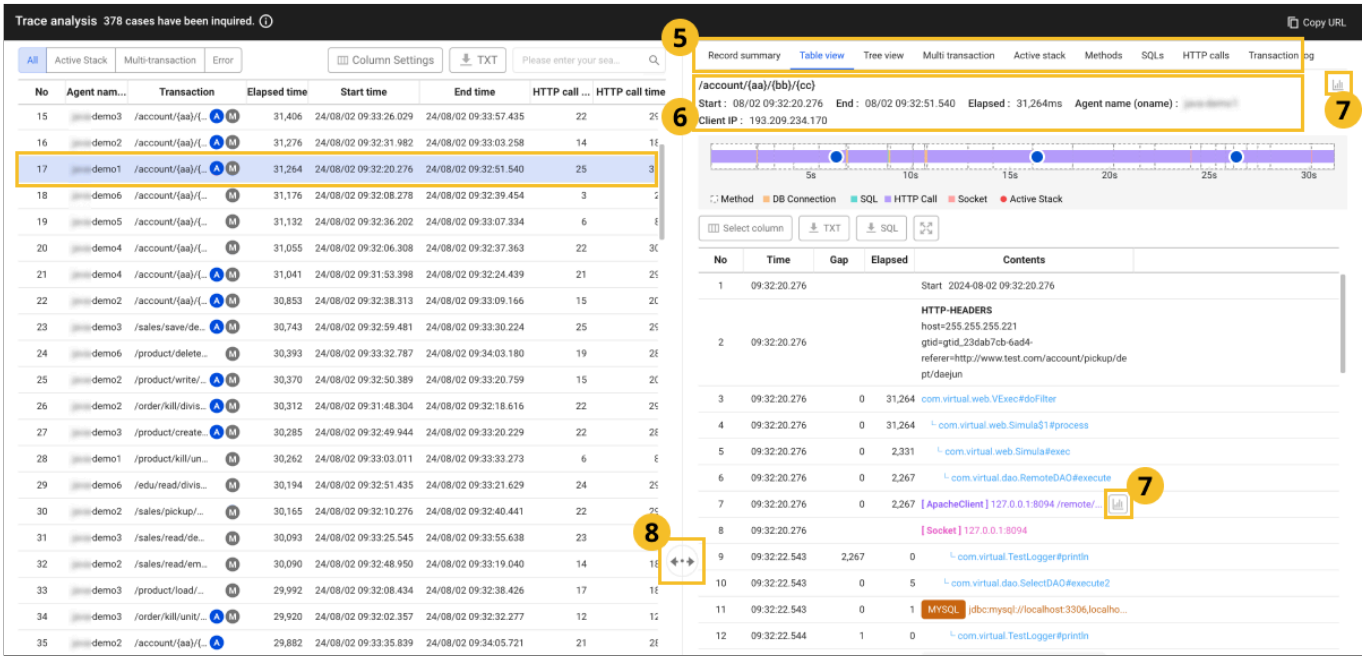
> - You can place, add, or hide desired columns to the front of the list. For more information, see the following.
> - If you drag another area in the **Hitmap** widget, a new trace analysis window appears. You can check it by comparing with the previous trace analysis window.
> - The transactions that are viewed can be shared with other users. With the transaction item selected (ⓐ), click ⓑ ![copy icon](/img/ ico-copy.svg) **Copy URL** on the trace analysis window. Pass the copied URL address to another user. Any user who has received the URL address can quickly see the same items by entering the URL address in the browser address bar.



Analyzing a trace

Select an item to view details from the transaction list. On the right of the screen, the detailed analysis of the selected transaction appears. In this state, if another transaction item is selected, the detailed analysis of the selected transaction item is changed.

To analyze the transaction performance, WhaTap collects and displays the properties such as name and client data, transaction performance, and detailed execution records for each section.



- In the 5 area, the step data can be seen in various formats. Select a desired type of tab.
- In the 6 area, you can see the basic information for transactions. The information matches the columns in

the list on the left.

- 7 : The statistical data window for the corresponding URL or each execution section appears. Click a desired time on the graph chart in the statistical data window to go to **Statistics** or **Transaction Search**. You can view the statistical data based on the selected time.
- By dragging 8 to the left or right, you can adjust the screen size. This is useful when you want to see more information.

> ⓘ
> - For more information about **Statistics**, see the following.
> - For more information about **Transaction Search**, see the following.
> - If you click ◀ or ▶, you can change the transaction list or analysis details to full screen.

The collected types of **Step** are as follows. The diagrams and texts are color coded.

- DB Connection step START-END

  It includes the performance for connection to RDB. Step information includes the name, response time, and error.

- SQL step START-END

  It includes the performance for JDBC SQL. Step information includes connection data, SQL statements, and errors.

- HTTP Call step START-END

  It includes the performance for the outbound HTTP service call. Step information includes URL, host, port, response time, and errors.

- Message step ADD

  While collecting traces, the message steps are used to collect the records of all irregular sections.

- Socket step ADD

  Step to represent the socket open.

- Method step START-END

  The method response time is traced.

- Active Stack step START-END

It includes the active stack information. It is created and collected by a separate thread and it is added to the transaction trace.

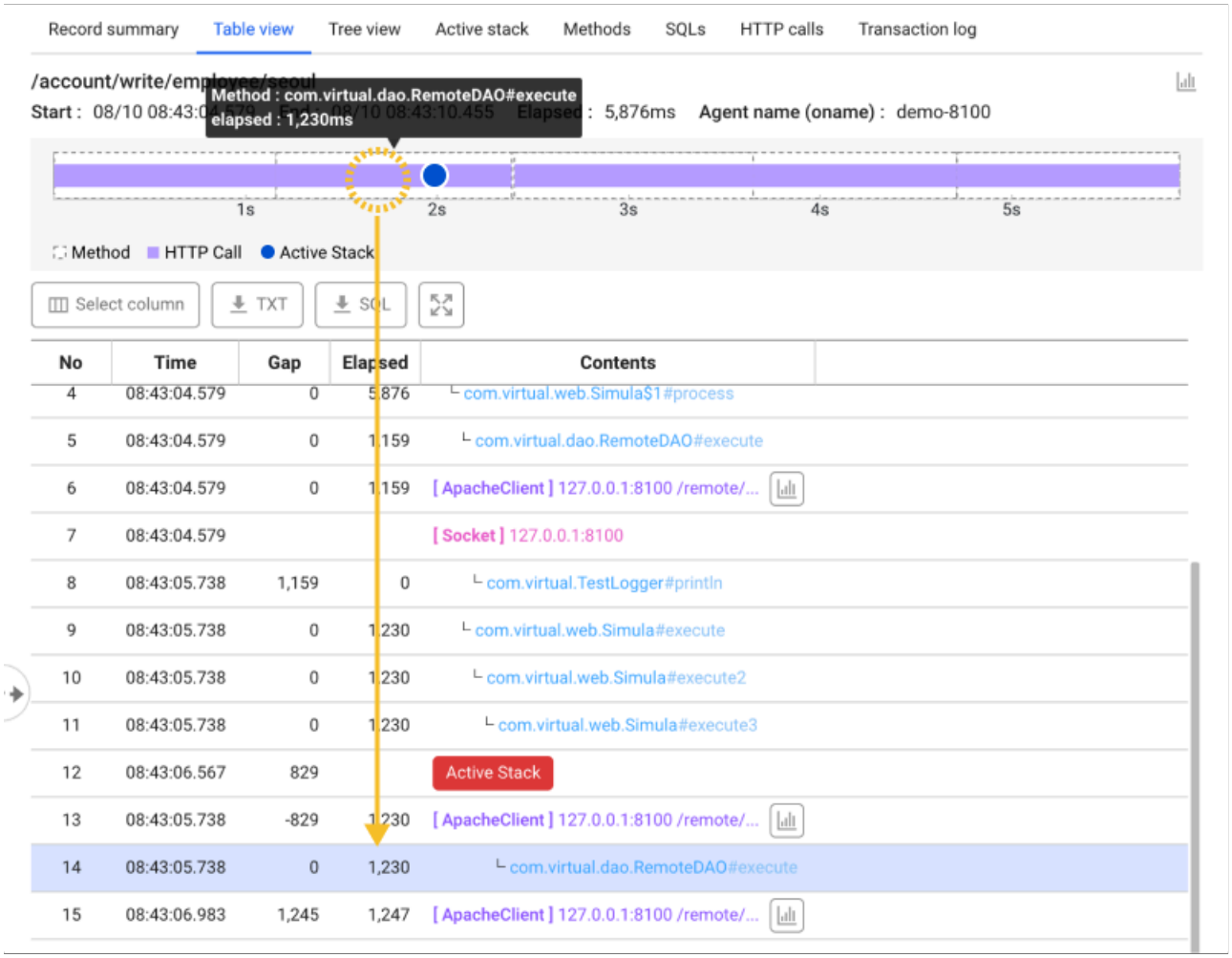
If you add a database project through the **Correlated Project Management** menu, you can see the performance metrics of the linked project. For more detailed information about adding and analyzing a linked project, see [the following] (linked-projects).

- In the WhaTap monitoring service, **Step** is used as the same meaning as **Span**.
- To see the updates in the **Trace analysis** function released on July 25, 2023, see the following.

Table view

In the **Table view** tab, you can see the transaction execution steps in chronological order.

- You can classify transactions by each execution section and see them in a diagram. Out of all elapsed times, you can quickly find the times spent for each section and the section that took the longest.
- If you select each section in the diagram, the corresponding table list appears where the step is located.



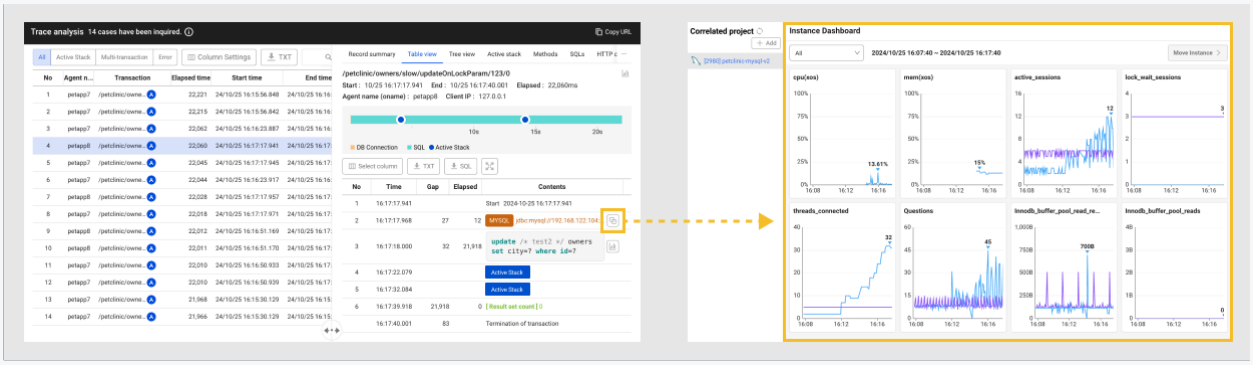
- The area where the blue circle (🔵) is located, is the moment when the active stack has been collected. If you select the blue circle, the table list appears where the **Active Stack** button is located. If you click **Active Stack**, you can check the step performed during the section.

  ⓘ
  - In the transaction list, you can see them from the items marked with the Ⓐ icon.
  - For more information about active stacks, see the following.

- **Select column**: You can add or hide the memory accumulation and CPU accumulation data as columns in the table list. The columns are as follows:
  - **No**: Step occurrence sequence

  - **Time**: Start time of each step
  - **Gap**: The waited time between the start time of the previous step and the current step may have a difference, while there may be a gap from the elapsed time if a delay happens due to external factors.
  - **Elapsed**: Total elapsed time from the start to the end of each method
  - **Contents**: Execution details of the step
- **TXT**: You can download basic transaction data and execution data for each section in a text file.
- **SQL**: The basic transaction information and SQL execution data can be downloaded as a text file.
  - **Download with substitute value**: It downloads the SQL statistics data after replacing the bind variable value with its original one. If a security key (`paramkey`) is set, the security key input window appears. You must enter the security key for downloading. For more information about the agent configuration related to the security key, see the following.
  - **Download without substitute value**: It downloads SQL statistics data without replacing the bind variable value.
- : You can see only the table list, with hiding the transaction summary and diagrams. If you select , the hidden information reappears. This feature is useful when you have a long list of tables.
- If you select the SQL step, the **SQL** window appears to display parameters. For more information, see the following.

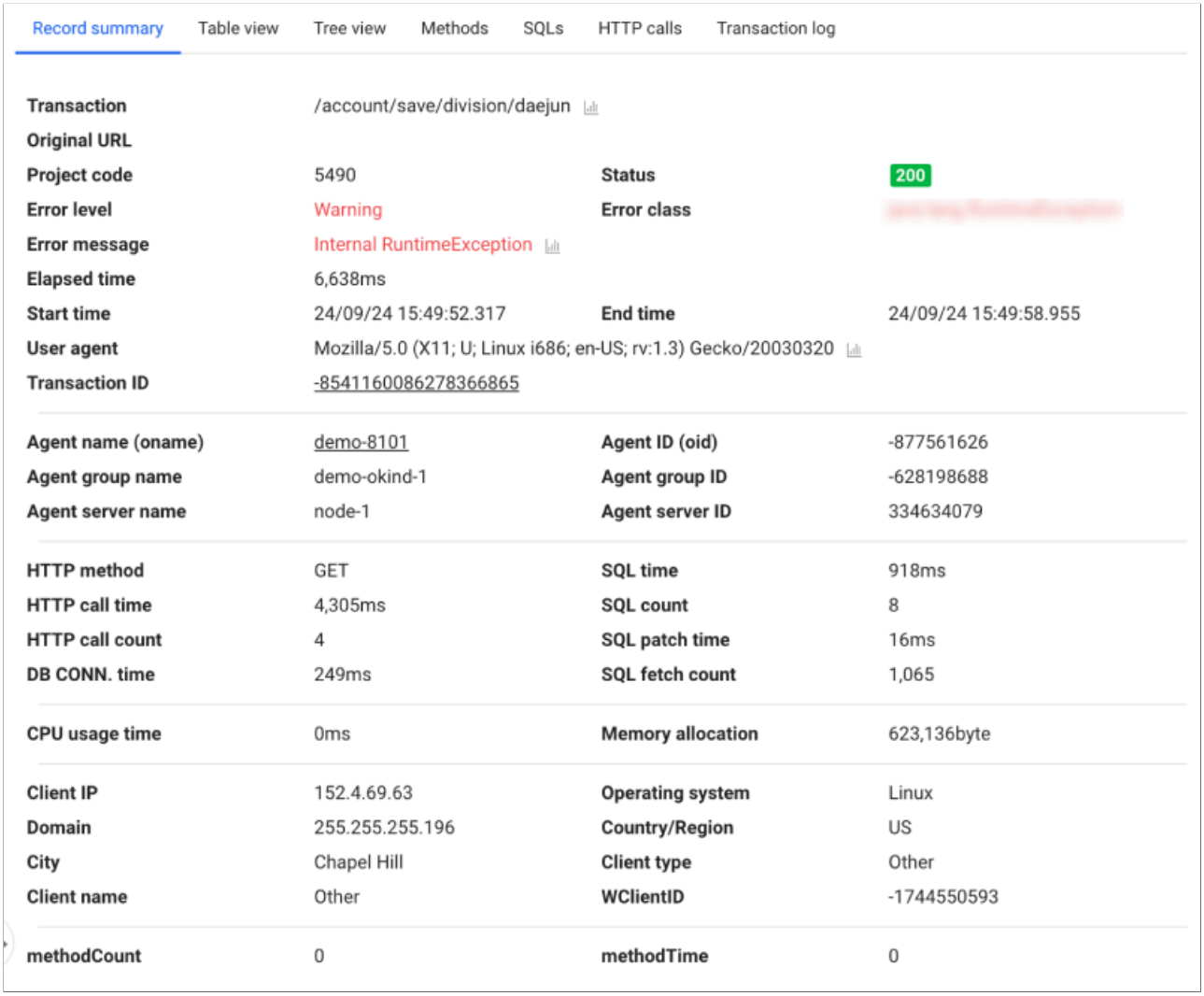
If you add a database project through the **Correlated Project Management** menu, you can see the performance metrics of the project linked in the **DB connection step**.



For more detailed information about adding and analyzing a linked project, see [the following] (linked-projects).

Record summary

To see the basic information for transactions, select the **Record summary** tab. You can see the basic information of the transaction, agent on which the transaction was performed, memory allocation usage, and client information.



Classification	Properties	Description
Transaction	Transaction	API or URL call executed by the application, and the path of the transaction. It converts to a specific format using regular expressions.

Classification	Properties	Description
	Original URL	It is the URL path that was actually called. A concrete request path that does not include variables. If there is an original URL, the original URL appears. If there is no original URL, it is not displayed.
	Project code	Identification code of the project registered in the WhaTap monitoring service.
	Status	HTTP response status code for the transaction.
	Error level	Level of the error that occurred in the transaction. It is displayed as Warning or Critical.
	Error class	Class of the error that occurred in the transaction.
	Error message	Message of the error that occurred in the transaction.
	Elapsed time	Transaction execution time.
	Start time	Transaction start time.
	End time	Transaction end time.
	User agent	Browser used to extract client-related data.
	Referer	Page address or incoming path used by the browser of the client.
	Transaction ID	It is a unique ID of the transaction.
	Multi transaction ID	It is a unique ID of the multi-transaction. It appears when executed in parallel with transactions of other applications.
Agent	Agent name (oname)	Agent name

Classification	Properties	Description
	Agent ID (oid)	It is a unique ID of the agent.
	Agent group name	Agent group name set with the `okind` option.
	Agent group ID	It is a unique ID of the agent group.
	Agent server name	Agent server name set with the `ondoe` option.
	Agent server ID	It is a unique ID of the agent server.
Step	HTTP method	HTTP method. (GET, POST, PUT, HEAD, etc.)
	HTTP call time	Outbound HTTP call time.
	HTTP call count	Number of outbound HTTP calls.
	DB CONN. time	Time taken to connect to the database.
	SQL time	SQL execution time.
	SQL count	SQL execution count.
	SQL patch time	Time taken to search for the record. Nesting may occur or other logic may be included.
	SQL fetch count	Number of the records retrieved from SQL.
Resource	CPU usage	CPU time used for processing transactions.

Classification	Properties	Description
	time	
	Memory allocation	Amount of memory used by the transaction.
Containers	Pod name	It is the name of the Kubernetes Pod that processed the transaction.
	Container Id	It is a unique identifier for a specific container running within a Pod.
Client	Client IP	Client IP address.
	OS	OS environment in which the browser runs.
	Domain	Internet address assigned to the IP address that the client connected to.
	Country/ Region	Client's country.
	City	City in the client's country.
	Client type	Browser type used by the client.
	Client name	Device name used by the client.
	WClientID	This is a unique ID of the client.
Method	methodCount	Number of methods called by transactions.
	methodTime	Time taken for method execution.

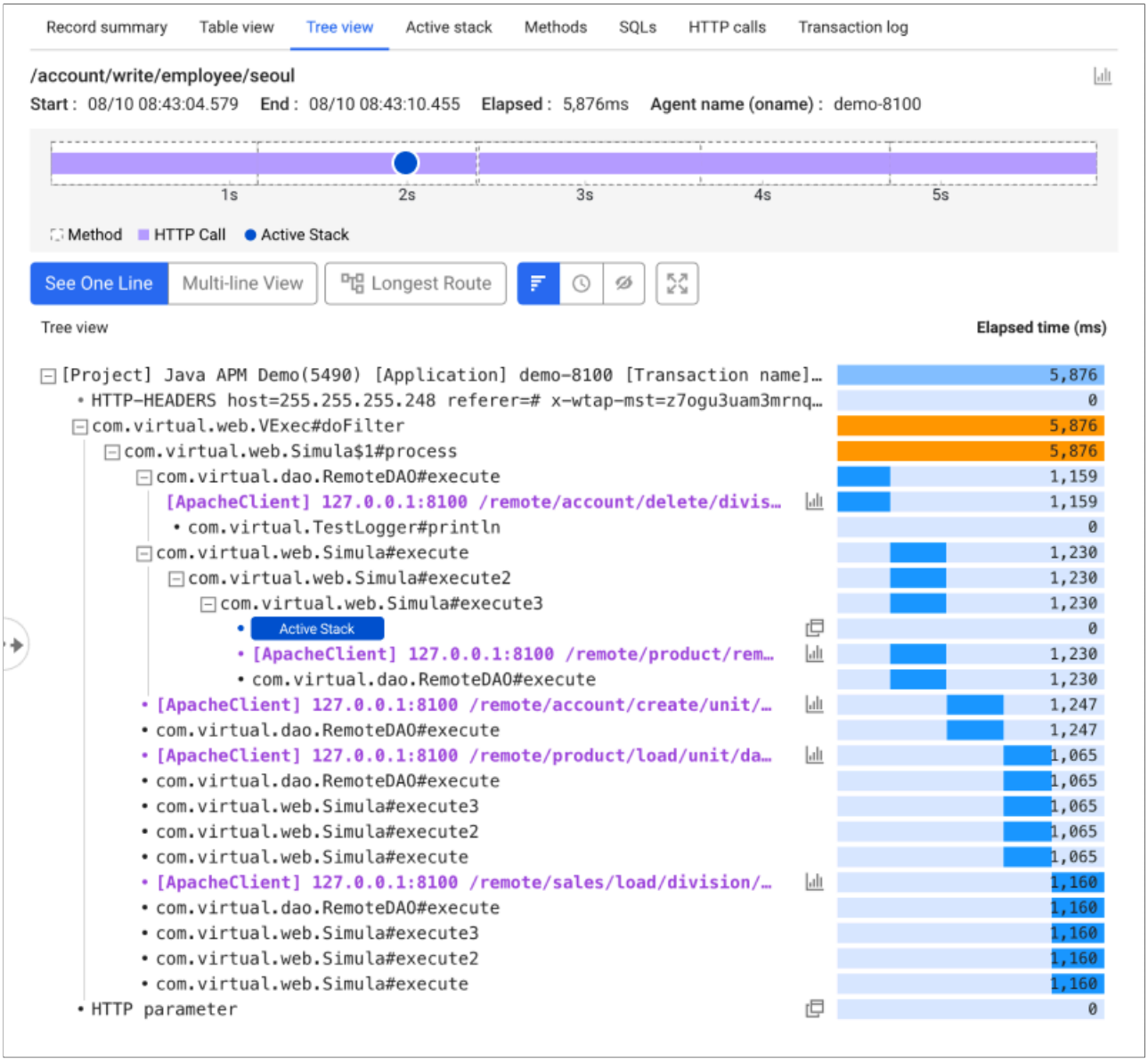
- The collected data may differ depending on the application type, settings, and step type. **Container** entries appear only when the application is running in the container environment.
- Error related items are displayed when an error occurs from the transaction.
- Some transaction properties provide statistical features. If you select the button, a window

appears where you can see the status of transactions occurred during the query time through a time series chart. If you select a bar graph in the chart, you can go to the **Analysis** > **Transaction Search** menu to view details.

- WhaTap basically stores the client-related information. For more information about the agent settings for user data collection, see the following.

Tree view

To see the transaction execution steps in a tree format, select the **Tree view** tab. You can see the details of each transaction and its trace details, trace start time, step time, and call relationship. If you select each section of the diagram, the tree view of the step appears.

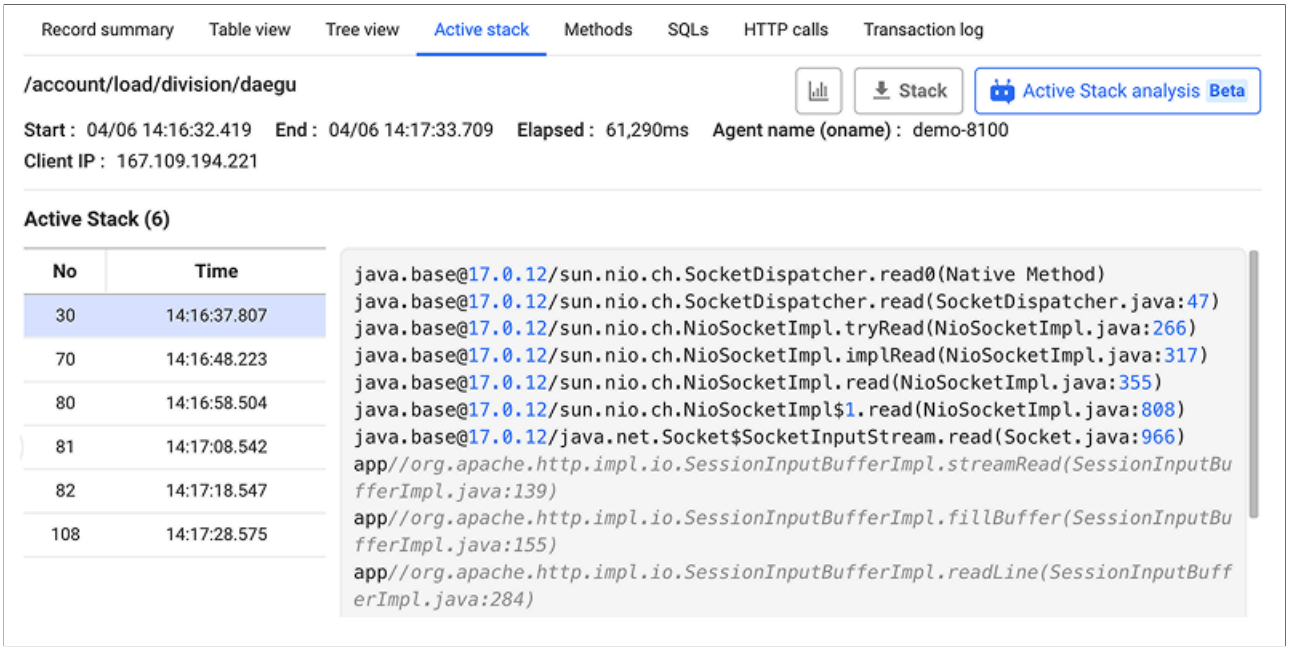


- **Single-line View**: You can arrange the tree format at intervals by displaying the texts displayed in the execution data for each section in a single line.
- **Multi-line View**: All the texts displayed in the execution data for each section, appear by wrapping words.
- **Longest Path**: The longest route can appear.
- **Show Time Bar**: The elapsed times are displayed in a bar chart.
- **Show Time**: The timestamps, gaps, and elapsed times for each segment are displayed in text.

  - 8 seconds or more: Excessive delay state is distinguished in red.
  - Between 3 and 8 seconds: Delay state is distinguished in orange.
  - Less than 3 seconds: Normal state is distinguished in blue.
- **Hide Time**: The time is hidden.
- : You can see only the tree view that hides transaction summary and diagrams. If you select , the hidden information reappears. This feature is useful when the tree view list is long.
- : A window appears where you can view SQL variables and HTTP queries. For more information, see the following.

> In case of start and step times, the time differences caused by the transaction call environment are corrected on the screen through traces and mappings in the parent transactions. Accordingly, there may be differences from the actual time data.

Active Stack



Select the **Active stack** tab to view the list of active stacks collected from the transaction.

- Select a collection point (No, Time) from the **Active Stack** list on the left to display the stack information for that point on the right.
- **Stack**: You can download the active stack details as a TXT file.

> - You can check the active stack from items marked with the A icon in the transaction list.
> - For more information about active stacks, see the following.

AI Active Stack Analysis

Beta

Click the **Active Stack Analysis** button to automatically analyze collected active stacks with AI. It identifies performance issues based on the stack dumps of in-progress transactions and suggests improvements.

The analysis results consist of the following 4 sections:

- **Summary**: Summarizes the number of analyzed stacks and the overall execution flow.
- **Common Patterns**: Displays repeatedly observed execution patterns, including pattern name, occurrence count, and description.
- **Issues Found**: Displays identified performance issues with severity levels (**High**, **Medium**, **Low**) along with the stack information used for the assessment.

Severity	Criteria
High	Deadlock, complete blocking, same blocking repeated 4+ times
Medium	I/O wait, 2-3 lock contentions, slow external calls
Low	Single-occurrence wait, minor optimization opportunities

- **Recommendations**: Suggests specific improvement actions by priority.

Each analysis item displays the collected stack number (`No.N`). Click the stack number to view the original stack.

> - The analysis results are generated by AI and may differ from actual conditions. Use them as

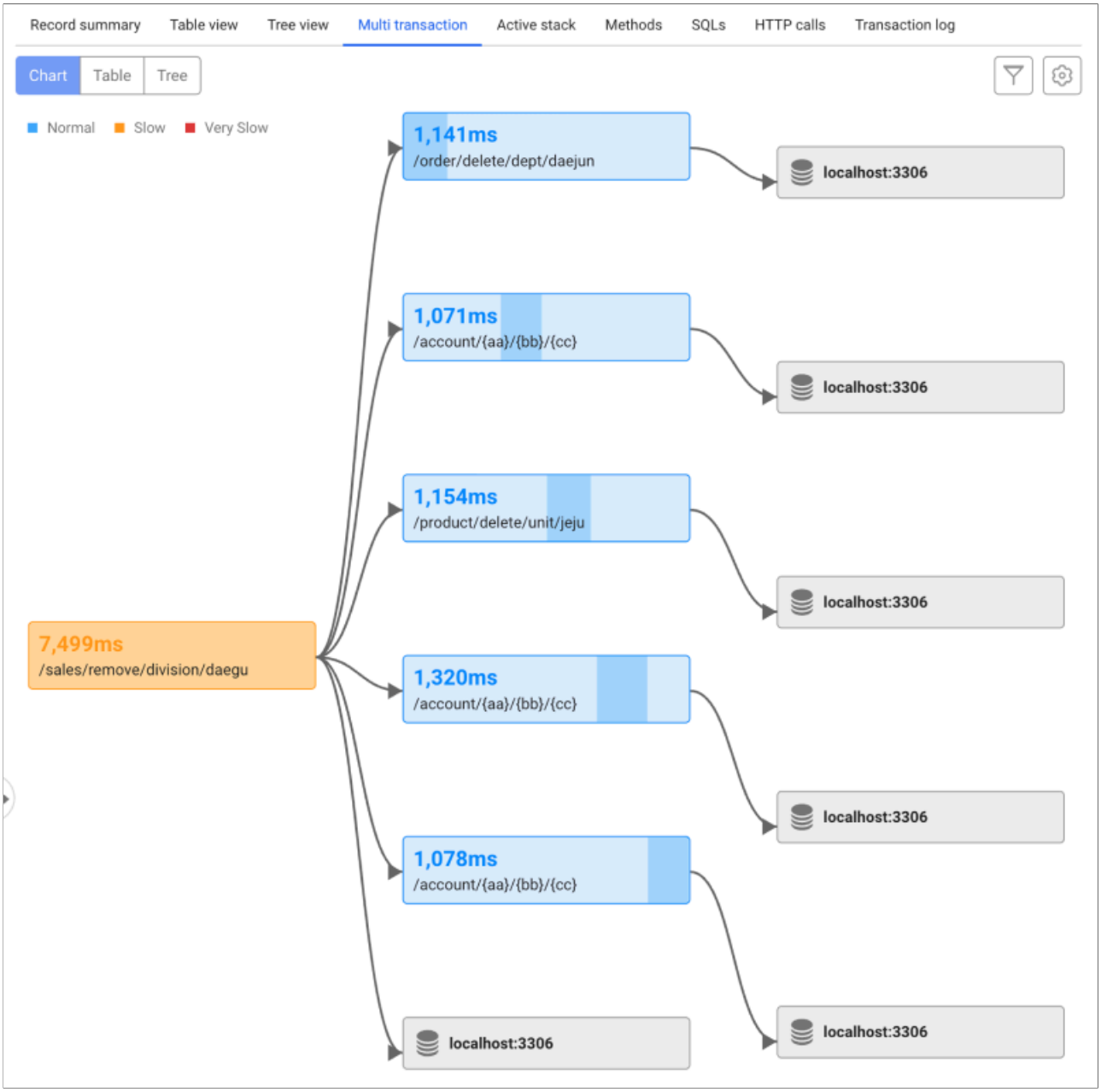
reference material.

- **Member** role or above is required.

Multi-transaction

Multi-transaction indicates the transaction involving other agents or projects. In the **Multi transaction** tab, you can see the call relationships between applications registered in the WhaTap monitoring service.

To trace multi-transactions, set the `mtrace_enabled` option to `true` in **Management** > **Agent CONF.**. For more information about the agent configuration, see the following.



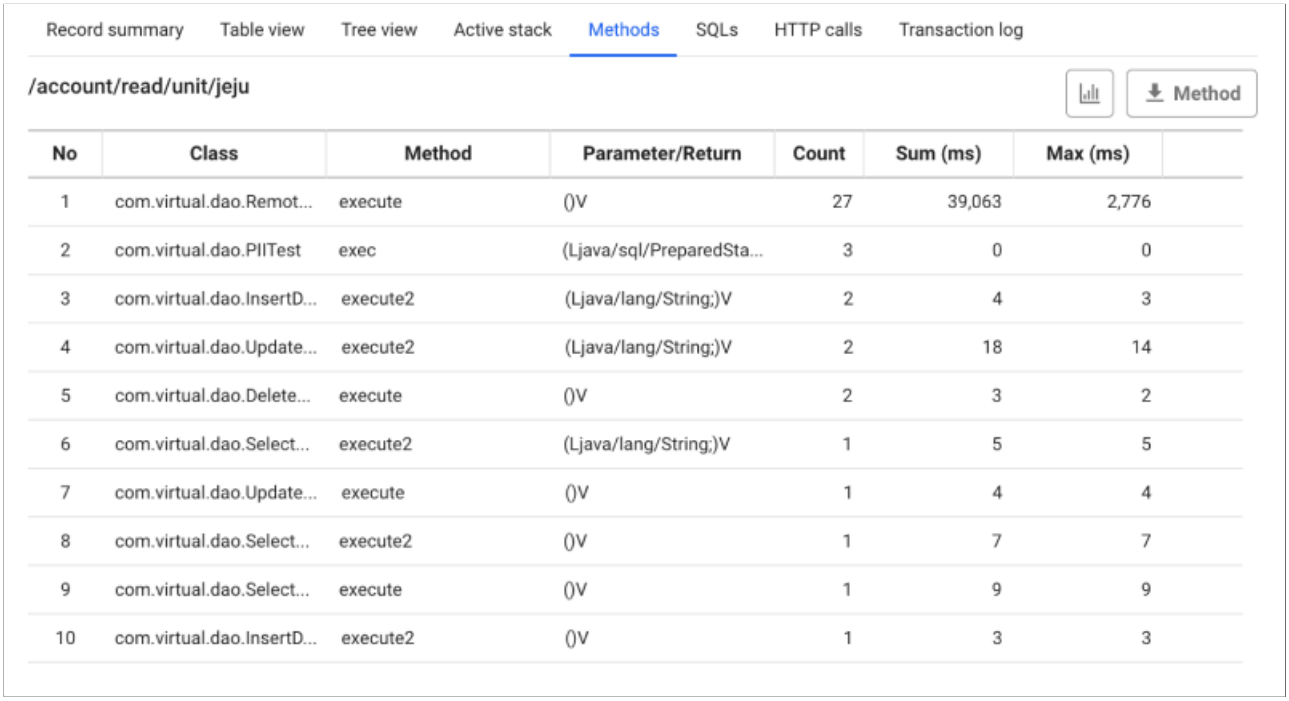
- **Chart**: Provides the call relationships for each transaction in a flowchart. Select a transaction node to see the trace analysis for the transaction node. In **Chart** mode, you can use the mouse to go to a desired position or zoom in or out by scrolling.
  - ⚙ **Chart view settings**: You can display or hide the elements on the chart.
- **Table**: In the form of table, you can check the data for each transaction included in the multi-transaction. You

can select the column selection icon to edit the table header columns. You can see the trace analysis data by selecting each transaction item.

- **Tree**: You can see the call relationships between transactions in a tree format. The related additional functions are the same as those of the **Tree view** tab.
- **Project selection**: You can select or deselect projects to be displayed on the chart.

> - You can see them in the transaction list marked with the M icon.
> - For more information about the multi-transaction, see the following.

Methods



No	Class	Method	Parameter/Return	Count	Sum (ms)	Max (ms)
1	com.virtual.dao.Remot...	execute	()V	27	39,063	2,776
2	com.virtual.dao.PIITest	exec	(Ljava/sql/PreparedSta...	3	0	0
3	com.virtual.dao.InsertD...	execute2	(Ljava/lang/String;)V	2	4	3
4	com.virtual.dao.Update...	execute2	(Ljava/lang/String;)V	2	18	14
5	com.virtual.dao.Delete...	execute	()V	2	3	2
6	com.virtual.dao.Select...	execute2	(Ljava/lang/String;)V	1	5	5
7	com.virtual.dao.Update...	execute	()V	1	4	4
8	com.virtual.dao.Select...	execute2	()V	1	7	7
9	com.virtual.dao.Select...	execute	()V	1	9	9
10	com.virtual.dao.InsertD...	execute2	()V	1	3	3

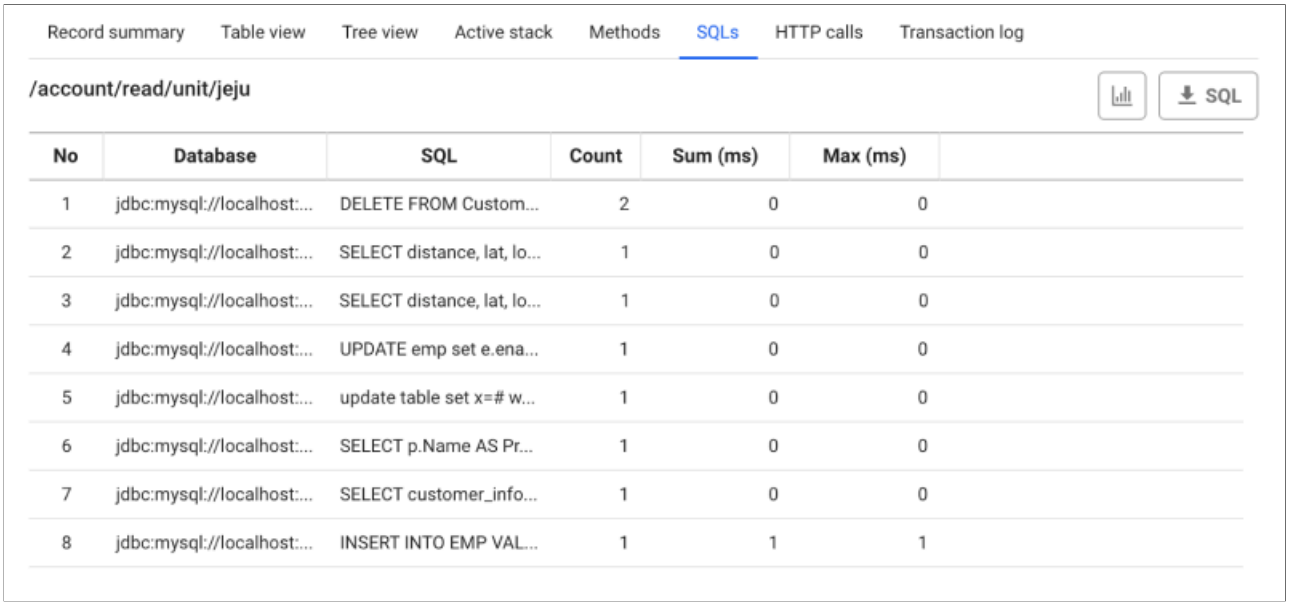
To see only the method data, select the **Methods** tab. It displays the method name and step time for which tracing has been set on the agent. When unnecessary logics are repeatedly executed, it can be used as analysis data to improve the method logic by finding the case where the execution takes longer than the low count.

- : You can see the **Transaction statistics** information. If a graph on the chart is selected, the **Analysis** > **Transaction Search** menu appears. You can see the list of transactions for the selected time zone.

- **Method**: You can download the method-related details as a TXT file.

> For more information about the agent configuration options on methods, see the following. Method tracing support may differ depending on the product in use.

SQLs

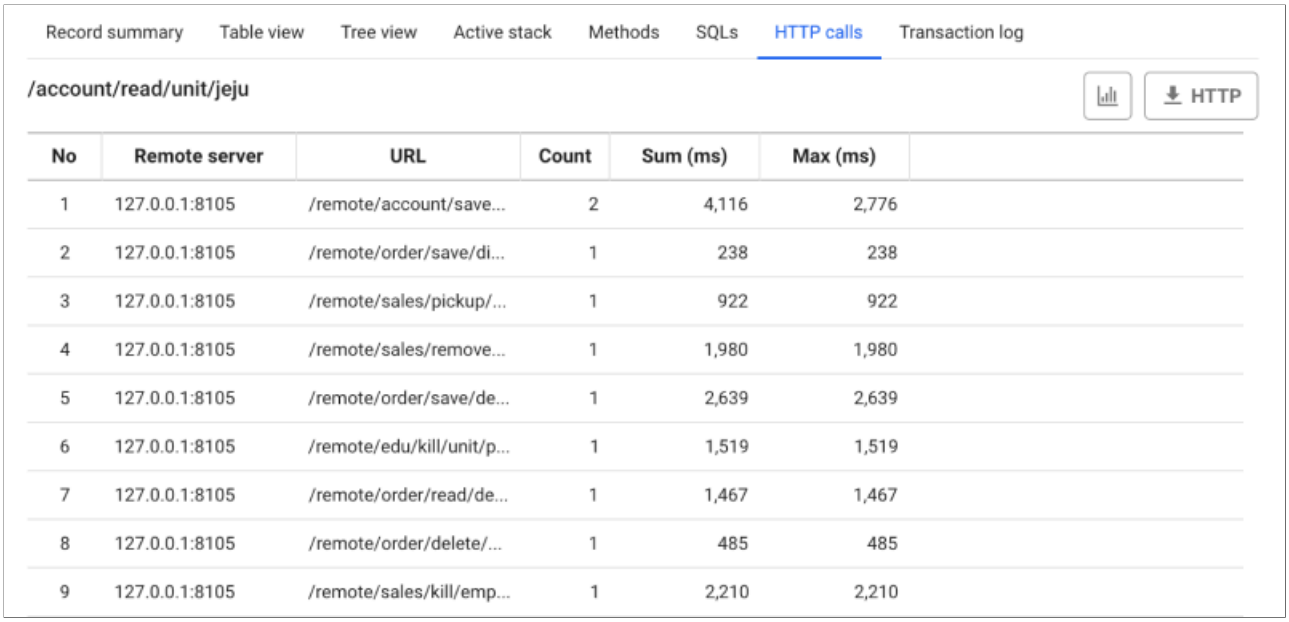


No	Database	SQL	Count	Sum (ms)	Max (ms)
1	jdbc:mysql://localhost:...	DELETE FROM Custom...	2	0	0
2	jdbc:mysql://localhost:...	SELECT distance, lat, lo...	1	0	0
3	jdbc:mysql://localhost:...	SELECT distance, lat, lo...	1	0	0
4	jdbc:mysql://localhost:...	UPDATE emp set e.ena...	1	0	0
5	jdbc:mysql://localhost:...	update table set x=# w...	1	0	0
6	jdbc:mysql://localhost:...	SELECT p.Name AS Pr...	1	0	0
7	jdbc:mysql://localhost:...	SELECT customer_info...	1	0	0
8	jdbc:mysql://localhost:...	INSERT INTO EMP VAL...	1	1	1

To check the SQL statement information, select the **SQLs** tab. In case of unnecessarily repetitive execution, it can be used as analytical data to improve query performance by finding the case where the execution time takes longer compared to the low count.

- : You can see the **Transaction statistics** information. If a graph on the chart is selected, the **Analysis** > **Transaction Search** menu appears. You can see the list of transactions for the selected time zone.
- **SQL**: You can download the SQL-related details as a TXT file.

HTTP calls



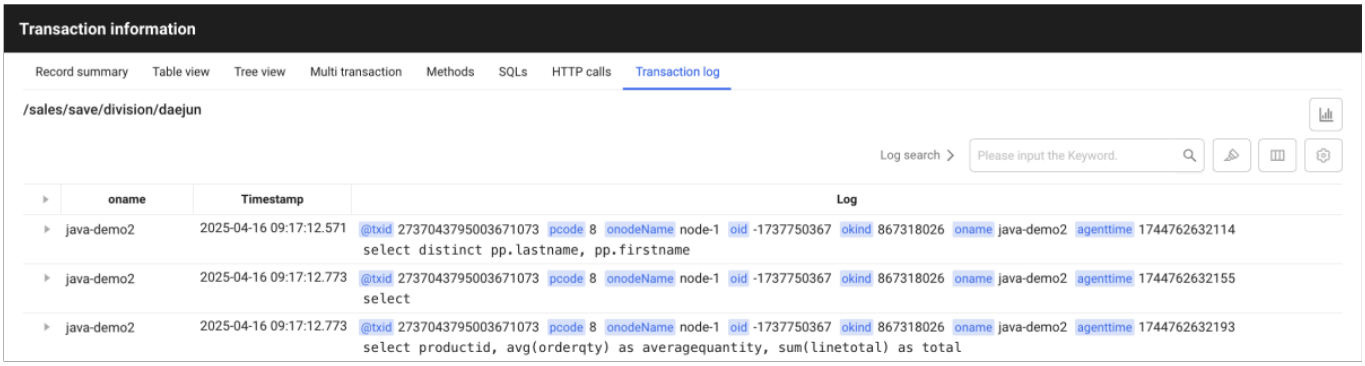
No	Remote server	URL	Count	Sum (ms)	Max (ms)
1	127.0.0.1:8105	/remote/account/save...	2	4,116	2,776
2	127.0.0.1:8105	/remote/order/save/di...	1	238	238
3	127.0.0.1:8105	/remote/sales/pickup/...	1	922	922
4	127.0.0.1:8105	/remote/sales/remove...	1	1,980	1,980
5	127.0.0.1:8105	/remote/order/save/de...	1	2,639	2,639
6	127.0.0.1:8105	/remote/edu/kill/unit/p...	1	1,519	1,519
7	127.0.0.1:8105	/remote/order/read/de...	1	1,467	1,467
8	127.0.0.1:8105	/remote/order/delete/...	1	485	485
9	127.0.0.1:8105	/remote/sales/kill/emp...	1	2,210	2,210

To see the count, time sum, average time, and such for HTTP calls, select the **HTTP calls** tab. If unnecessary external calls are repeatedly executed, it can be used as analytical information to determine whether the transaction delays are external or internal by finding the cases where the execution time takes longer compared to the low count.

- : You can see the **Transaction statistics** information. If a graph on the chart is selected, the **Analysis** > **Transaction Search** menu appears. You can see the list of transactions for the selected time zone.
- **HTTP**: You can download the HTTP call-related details as a TXT file.

Transaction log

To see the logs of the transaction, select the **Transaction log** tab.

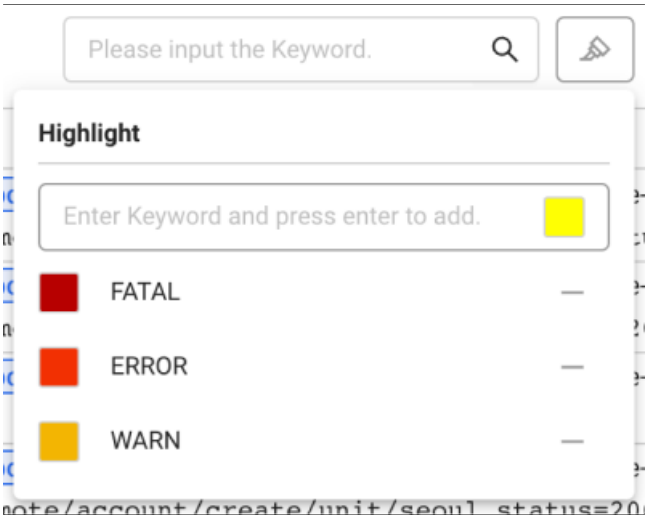


- : You can see the **Transaction statistics** information. If a graph on the chart is selected, the **Analysis** > **Transaction Search** menu appears. You can see the list of transactions for the selected time zone.
- **Log Search**: You can go to Log Search to easily find the desired log data.
- **Keyword search**: Enter the desired text to search in the keyword search field, and then press Enter or select the button. It highlights the keywords that match the entered text.
- **Keyword highlight**: If you set the keyword and color, the texts that match the keyword are automatically highlighted in the log list. For more information, see the following.
- **Column settings**: You can add columns or change the order of columns displayed in the log list table. For more information, see the following.
- **Log view detail setting**: You can configure the items to be displayed in the log list. For more information, see the following.

- To enable the **Transaction log** tab, apply the agent setting options related to the logs. For more information about the log settings, see the following.
- The **Transaction log** tab can be accessed by only the members with the **View log** role. For more information about the member roles, see the following.

Setting the keyword highlight

1. To add keywords to highlight, select .



2. Enter a desired keyword.
3. Select the color icon and then select a desired color.
4. Press Enter. The added keywords are added in the list.

Deleting the keyword highlight

To delete the items unused from the highlight list, select —.

Displaying or hiding the table columns

1. Select ▥ on the upper right of the table.
2. If the **Column settings** window appears, select or unselect the desired item in the list on the left.

   You can also hide columns by selecting the ✕ button in the list on the right.
3. After all settings are finished, select **Confirm**. The set content is applied to the table.

Reordering the table columns

1. Select ▥ on the upper right of the table.
2. When the **Column settings** window appears, change the order by dragging the desired items from the list on the right.

3. After all settings are finished, select **Confirm**. The set content is applied to the table.

Setting to display the table content

1. Select ⚙ on the upper right of the table.
2. When the **Log view detail setting** window appears, set the desired options.
   - Select **content** or **Tag** to determine whether or not to display the log content.
   - **Tag management**: Tags are displayed in the log in the order of addition. The order of tags can be changed by dragging. Disabled tags do not appear in the logs.
3. After all settings are finished, click ×. The set content is applied to the table.

> ⓘ For more information about **Log view detail setting**, see the following.

Additional options

Viewing HTTP parameters

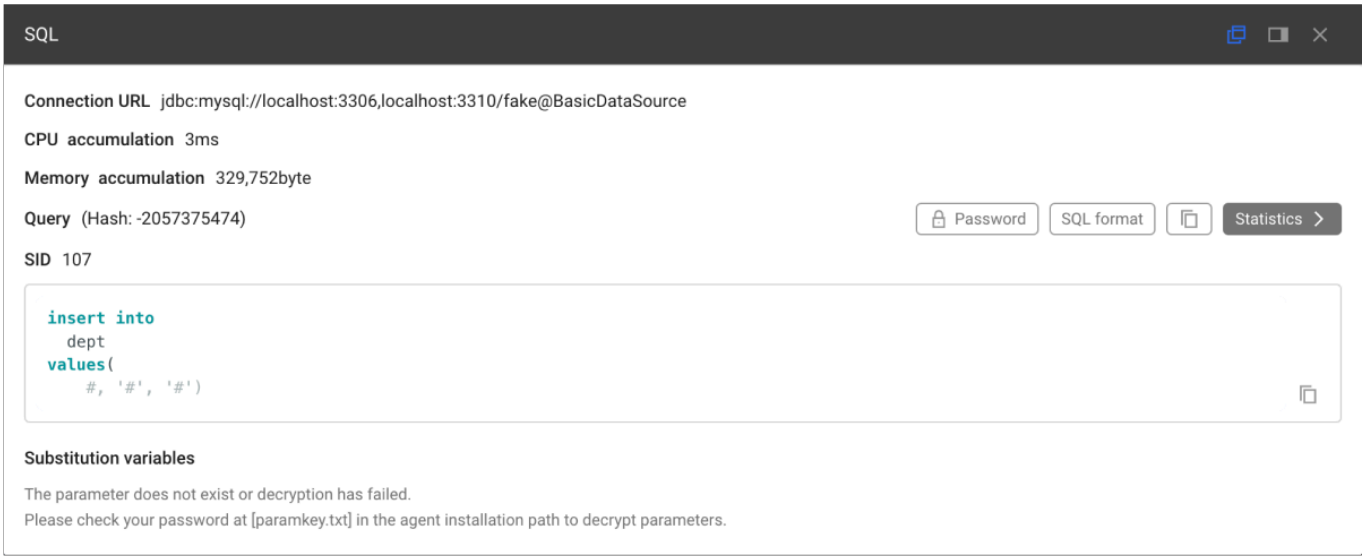
You can view the HTTP parameters of the transaction in the **Table view** tab.

1. Scroll down the page to the final step of the transaction.
2. Select the 🔒 **HTTP parameters** item.
3. If the **HTTP-PARAMETERS** window appears, select **Password**.
4. Enter the specified Param Key. You can see the hidden parameters.

> ⓘ
> - For the agent settings for HTTP parameters, see the following.
> - **Password**: You can see the decrypted parameter value. The password is a 6-digit string in the `WHATAP_HOME` /security.conf (paramkey.txt) file. You can change it to another string. It is required to view the SQL variables and HTTP queries and to stop the threads. For more information, see the following.

Viewing SQL parameters

Select the SQL step in the **Table view** tab or select in the **Tree view** tab. The **SQL** window appears to display parameters.



- **Password**: You can see the decrypted parameter value. The password is a 6-digit string(`paramkey`) in the `WHATAP_HOME` /security.conf file. You can change it to another string.

  > **Password** is required to view the SQL variables and HTTP queries, and stop the threads. For more information, see the following.

- **SQL format**: You can indent and format your SQL statements for better readability.
- : A SQL statement can be copied to the clipboard.
- **Statistics**: Go to **Statistics** > **SQL**.

> To search SQL variables and HTTP queries, add the following options to the agent settings.
>
> - For agent settings related to recording SQL parameters, see the following.
> - For agent settings related to recording the HTTP parameters, see the following.
>
> whatap.conf

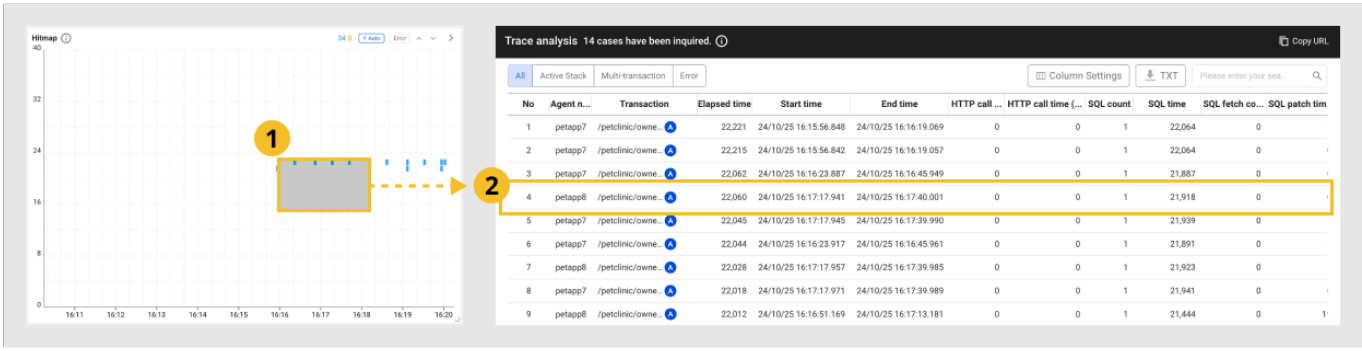
```
# SQL parameter lookup option: If the option is applied, SQL parameters are collected after being encrypted.
profile_sql_param_enabled=true

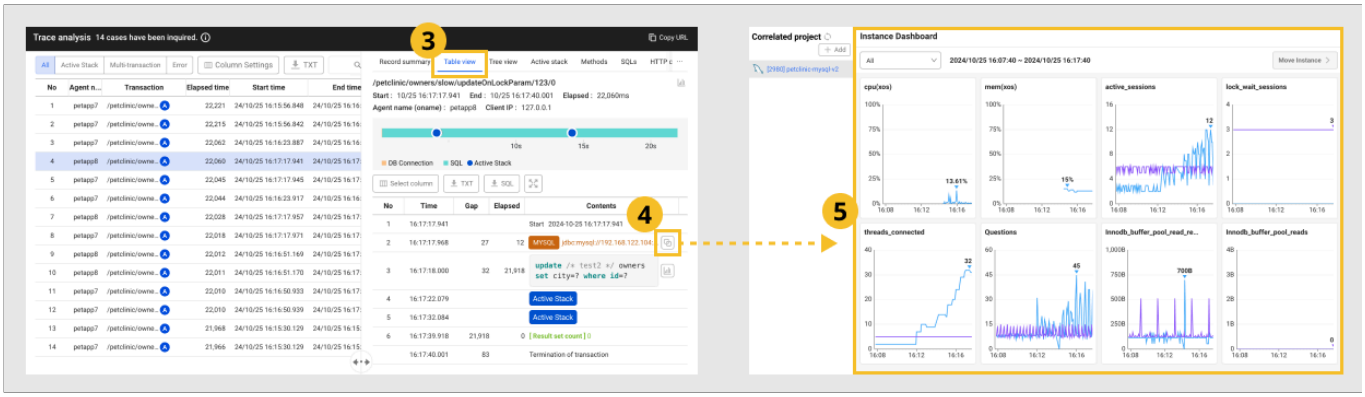
// HTTP parameter lookup option: If the option is applied, HTTP query parameters are collected after being encrypted.
profile_http_parameter_enabled=true
```

Viewing the linked project performance metrics

If you add a database project through the **Correlated Project Management** menu, you can see the performance metrics of the project linked in the **DB connection step** and **SQL step.

Checking in the DB connection step



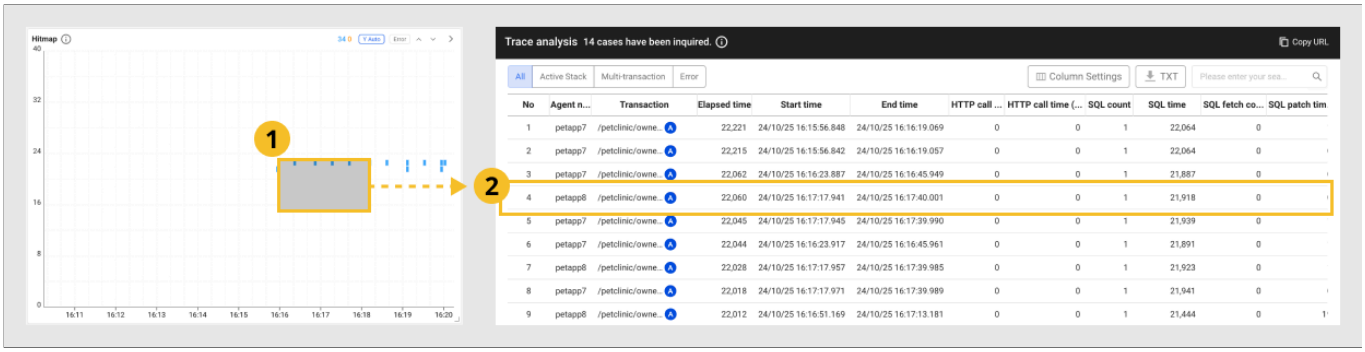


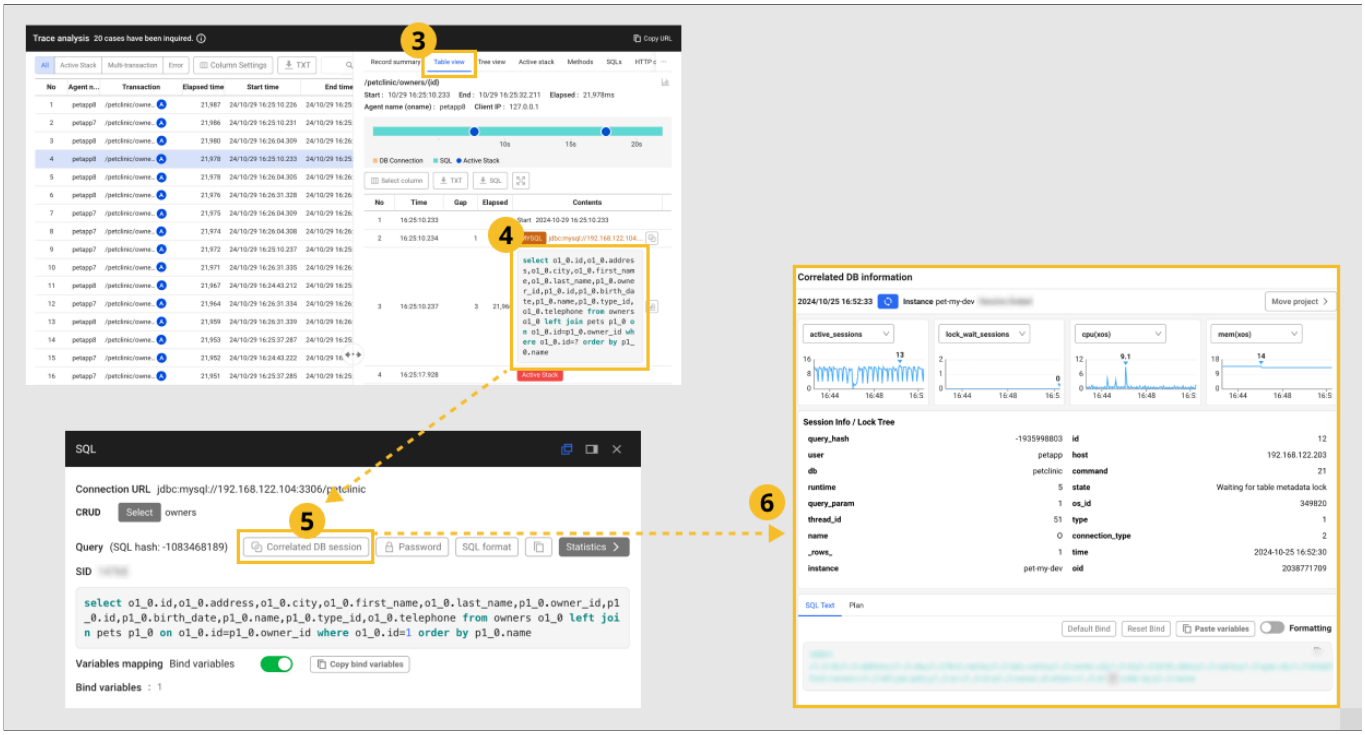
1. Drag the chart area of the **Hitmap** widget on the dashboard.

2. When the **Trace analysis** window appears, select a transaction to view.
3. When the transaction details appear on the right, select the **Table view** tab.
4. In the DB Connection step, select .
5. When the **Instance dashboard** window of the linked project appears, check the related metrics.

- The button appears on the **Table view** tab only when the DB of the database project registered for linked analysis executes the SQL statement of the selected transaction.
- For more detailed information about adding and analyzing a linked project, see [the following] (linked-projects).
- For more information about the **Instance dashboard** window, see the following.

Checking in the SQL step

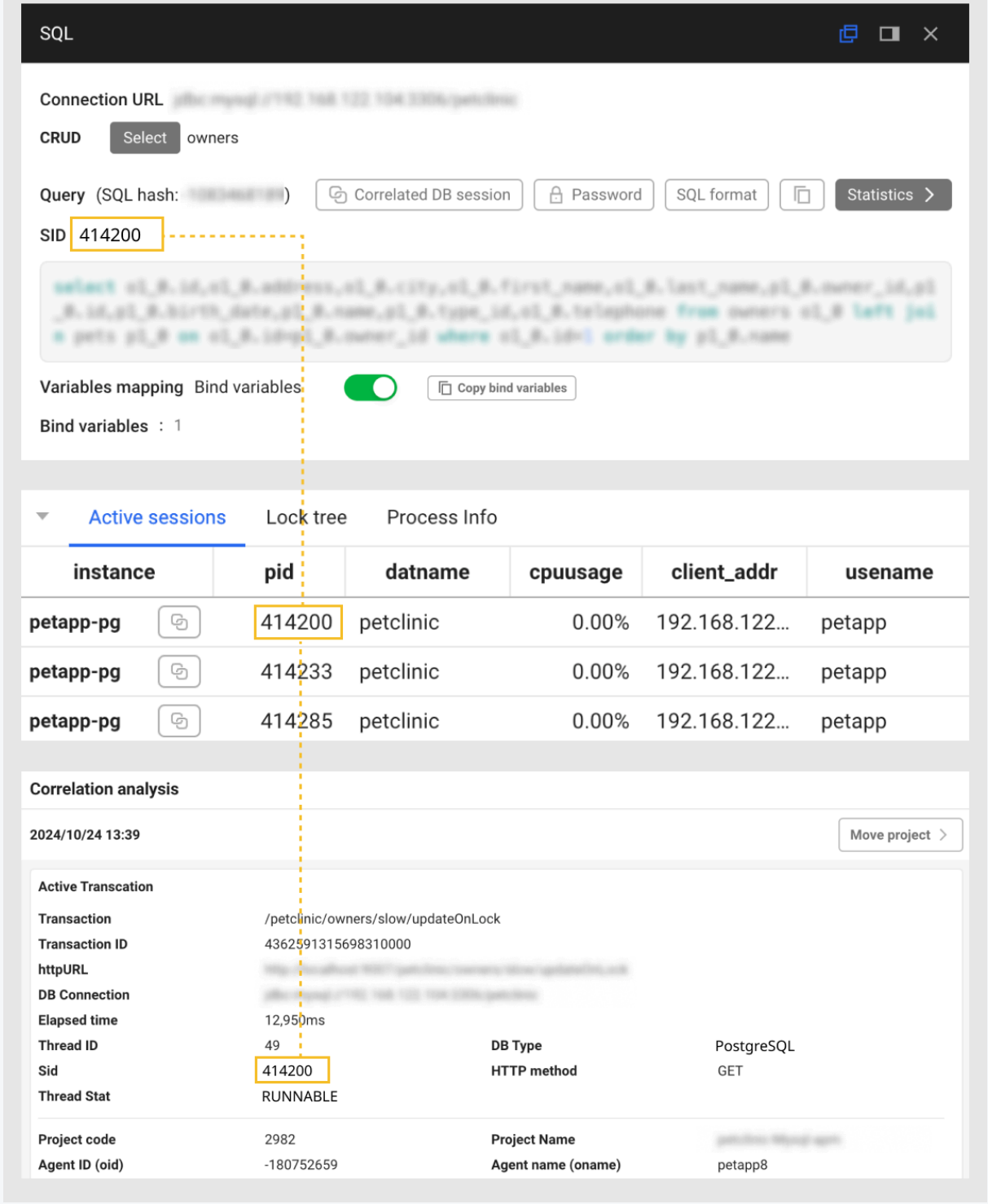




1. Drag the chart area of the **Hitmap** widget on the dashboard.
2. When the **Trace analysis** window appears, select a transaction to view.
3. When the transaction details appear on the right, select the **Table view** tab.
4. Select a SQL statement in the SQL step.
5. If the **SQL** window appears, select **Correlated DB session**.
6. When the **Correlated DB information** window appears, check the related metrics.

- The **Correlated DB session** button appears in the **SQL** window only when the DB of the database project registered for linked analysis executes the SQL statement of the selected transaction.
- The **Copy bind variables** feature allows you to copy variables and paste them into the **SQL Text** tab of the **Correlated DB information** window. This feature is available when you select **Password**, enter the password, and then display the value of the bind variable.
- For more information about the **Correlated DB information** window, see the following.
- The databases that support this feature are **PostgreSQL**, **Oracle**, **Oracle Pro**, and **MySQL** platforms.
- On the **PostgreSQL** platform, the **PID** entry in the **Active sessions** table is the same as the **SID**

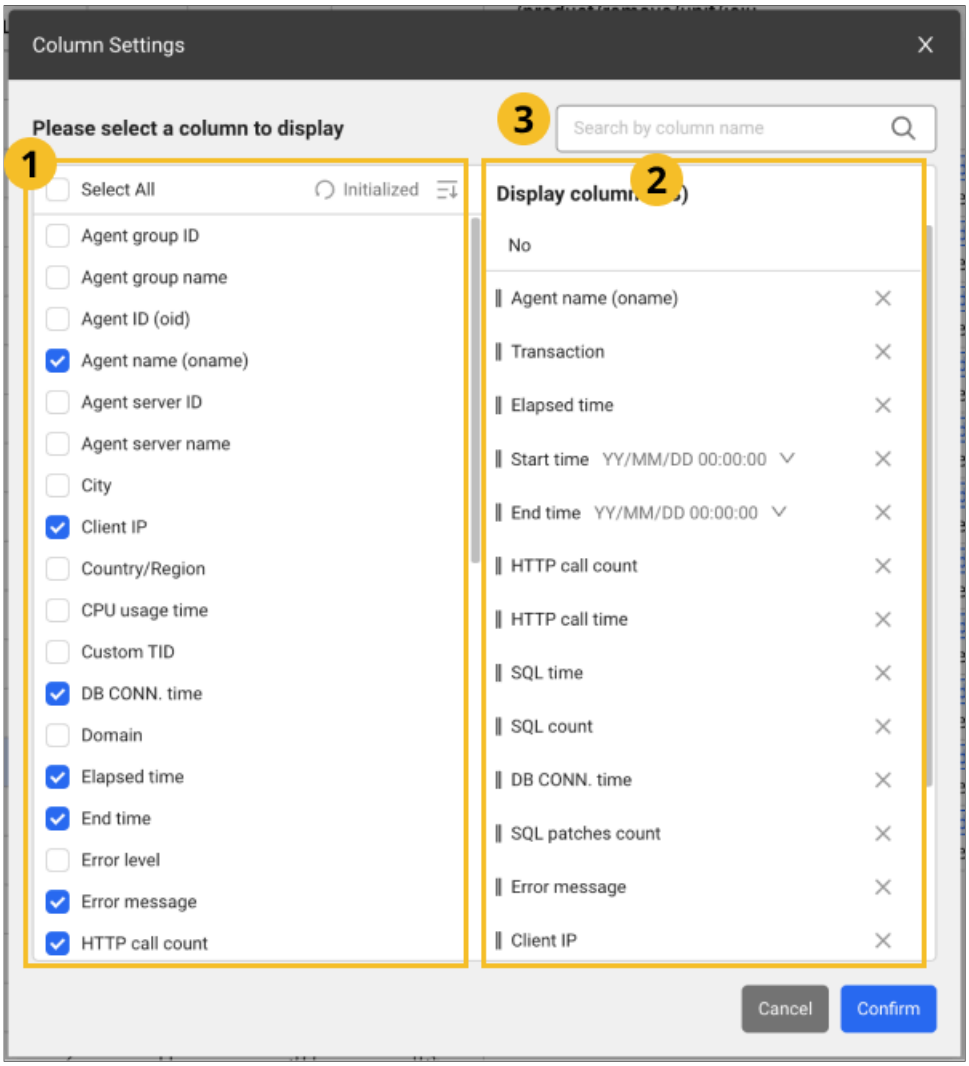
entry.


SQL
Connection URL
CRUD
Select
owners
Query (SQL hash:
Correlated DB session
Password
SQL format
Statistics
SID
414200
Variables mapping
Bind variables
Copy bind variables
Bind variables : 1
Active sessions
Lock tree
Process Info
instance
pid
datname
cpuusage
client_addr
usename
petapp-pg
414200
petclinic
0.00%
192.168.122...
petapp
petapp-pg
414233
petclinic
0.00%
192.168.122...
petapp
petapp-pg
414285
petclinic
0.00%
192.168.122...
petapp
Correlation analysis
2024/10/24 13:39
Move project
Active Transcation
Transaction
/petclinic/owners/slow/updateOnLock
Transaction ID
4362591315698310000
httpURL
DB Connection
Elapsed time
12,950ms
Thread ID
49
DB Type
PostgreSQL
Sid
414200
HTTP method
GET
Thread Stat
RUNNABLE
Project code
2982
Project Name
Agent ID (oid)
-180752659
Agent name (oname)
petapp8

> In **the versions before Java Agent 2.2.40**, the **ThreadStat** values appear in numeric format on the **Active transaction** window. To display in text format, install Java Agent 2.2.40 or later.

Configuring the table columns

In the transaction list, you can hide the table header columns or add any of them. You can also change the column order. Select **Column Settings**.



- After configuration, select **Confirm** to apply the settings in the table.
- In the 3 search bar, enter text to search the desired columns. Only the columns that meet the entered text are displayed.

Adding columns

From the 1 list, select the items to add as table header columns. To select all items, select **Select All**.

Deleting columns

From the 1 list, unselect the columns to delete. Alternatively, select × on the right of the item to delete from the 2 list.

Changing the column order

Drag an item to reposition from the 2 list, and then move it to the desired position.

Initializing the configuration

To cancel all changes and reset them, select **Initialized**.

Downloading the viewed data

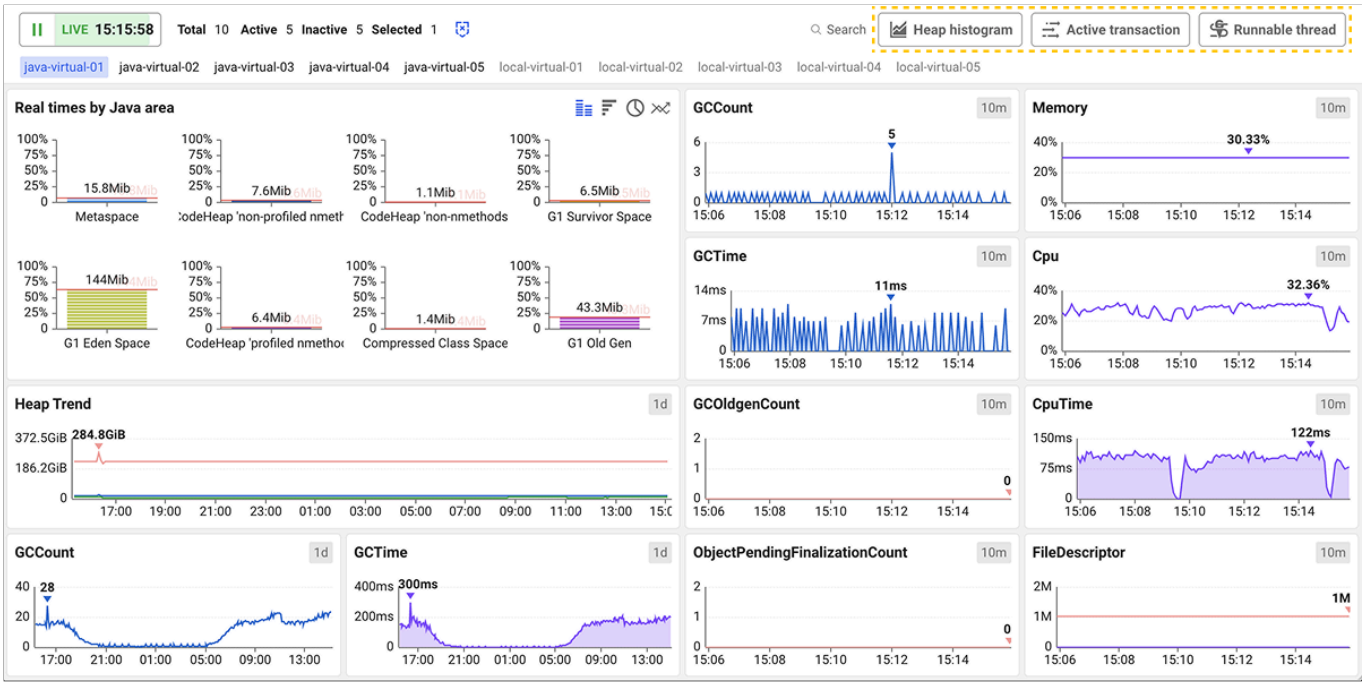
To download the data output in the table in .txt file format, select TXT.

Agent Resource Dashboard Config

Home > **Dashboard** > **Agent Resource Dashboard**

> **Requirements**
>
> The WhaTap Java agent 2.2.39 or later is required.

The **Agent Resource Dashboard** menu monitors key metrics such as memory, transactions, and threads for individual Java agents in real time, and displays data up to 3 hours ago. This allows you to see changes in memory usage, active transactions, and executing threads, and analyze the root causes of resource issues.



Basic screen

Agent Resource Dashboard allows you to check the memory pool status and related metric change trend of the running agent at a glance. Real-time Java memory area and key metrics you can check on the dashboard are as

follows.

Widget guide

Real-time Java memory widget

You can check the status of the JVM main memory pool through the **Real times by Java area** widget in real time. Each Java memory area chart displays the memory amount (in MB) and percentage (in %) in use.

- **Metaspace**

  It is the area where class metadata is stored.

- **CodeHeap non-profiled nmethod**, **CodeHeap profiled nmethod**, **CodeHeap non-nmethods**

  This area stores the machine code generated by the JIT (Just-In-Time) compiler. **CodeHeap non-profiled nmethod**, **CodeHeap profiled nmethod**, and **CodeHeap non-nmethods** store each compiled code according to the purpose.

- **G1 Survivor Space**

  This is the area where the objects survived in G1 GC are temporarily stored.

- **G1 Eden Space**

  This is the area where newly created objects are allocated first.

- **G1 Old Gen**

  This is the area where the objects survived through Young GC are moved. As memory increases, you can check if the leak is centered around Old Gen.

- **Compressed Class Space**

  This area compresses and stores class-related pointers and is managed separately from the Metaspace.

> - Individual chart: Displays each metric as a separate bar, allowing you to clearly compare maximum values for each metric.
> - Integrated chart: Relative sizes for each metric compared to the whole are displayed at a glance.
> - Donut chart: The proportions for each metric can be checked by expressing the whole as circular

> proportions.
>
> - 🔀 Multi-line chart: The trends for all metrics can be checked.

Widget for key metrics

- **Memory**

  It displays the total memory utilization (%). It is the total usage that synthesizes internal configurations such as Eden, Old Gen, and Metaspace.

- **CpuTime**

  The CPU time that the JVM spent executing application code is represented in milliseconds (ms).

- **GCCount**, **GCOldgenCount**

  GCCount collects only the number of GCs that occurred within a specific period, and **GCOldgenCount** collects only the number of GCs targeting the Old area separately.

- **GCTime**

  The time taken for garbage collection is represented in ms.

- **Cpu**

  The CPU utilization (%) of the JVM process is displayed.

- **TPS**

  The number of processed transactions per second is displayed.

- **FileDescriptor**

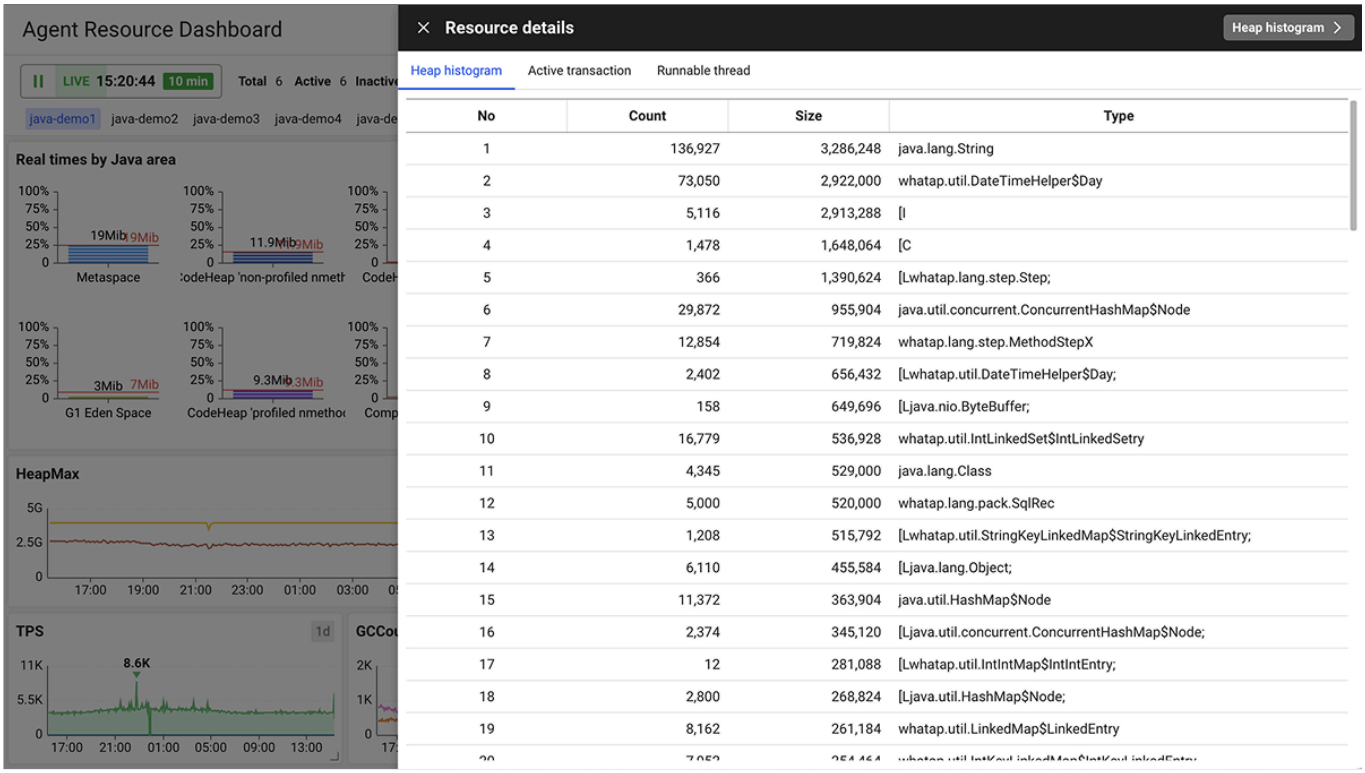
  It displays the number of file descriptors in use, such as open files or sockets.

- **ObjectPendingFinalizationCount**

  It displays the number of objects that remain in `finalize()` state that the GC has to process.

Real-time inquiry

When viewing in real time, you can check the **Resource details** panel through the **Heap histogram**, **Active transaction**, and **Runnable thread** buttons on the upper right.



Heap histogram

In the **Heap histogram** tab, you can analyze the heap memory resident Java class information and view the top 100 classes that occupy the most memory. If a class occupies excessive amount of memory, you may suspect a memory leak.

Active transaction

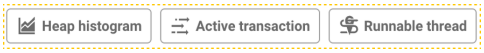
You can view a list of running transactions in the **Active transaction** tab. You can check the detailed information such as start time, transaction elapsed time, and SQL/HTTP calls. You can find bottlenecks by checking if specific transactions take a long time or use a lot of resources.

Runnable thread

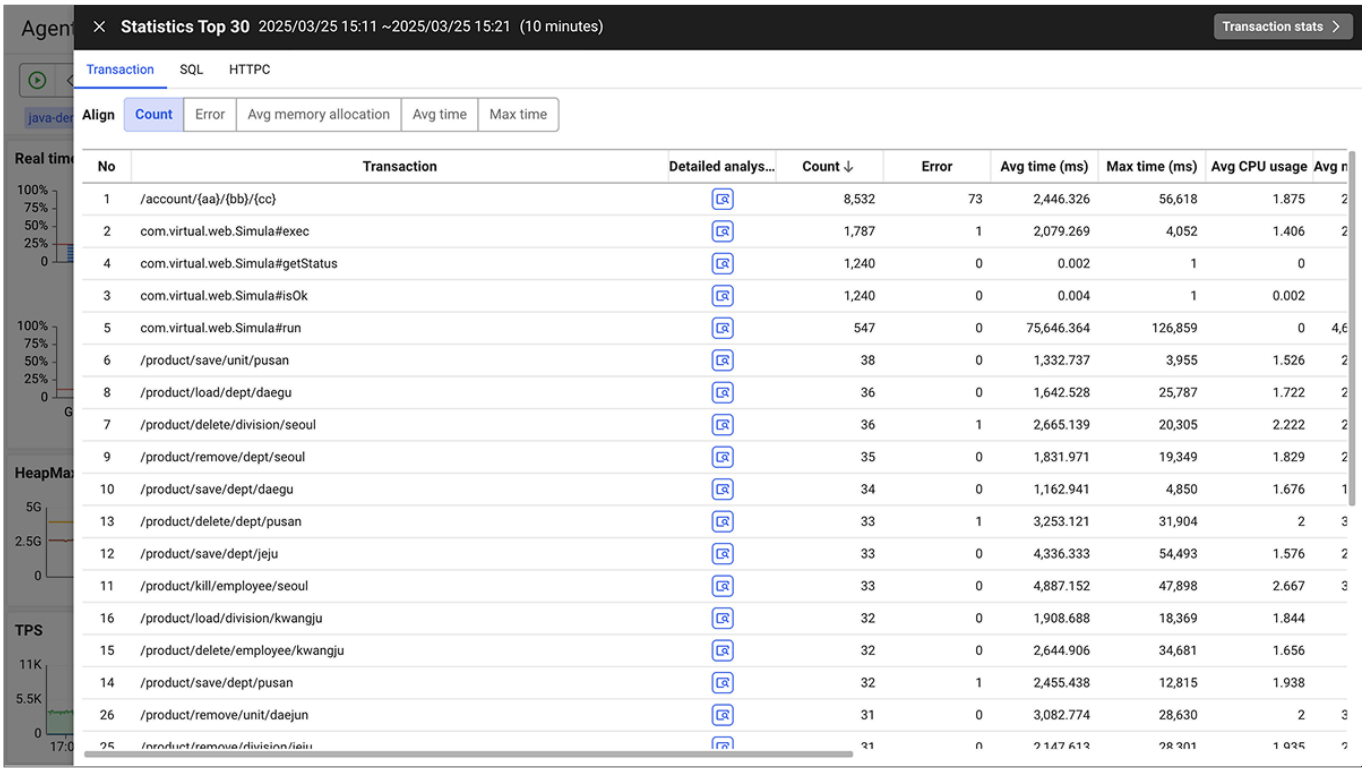
In the **Runnable thread** tab, you can check the list of threads in runnable state within the JVM in real time. It is useful to determine which thread is causing issues upon a high CPU usage or delayed response.

Historical inquiry

You can use the time selector to select a specific past point and check the resource status at that time point. You can query the resource status up to 3 hours ago and analyze transactions, SQLs, and HTTP calls that occurred at that time. This allows for effectively tracing of the root cause of a sudden spike in CPU or memory usage or a sharp drop in TPS.



When viewing a past time point, you can view the **Top 30 Statistics** panel where you can check the top 30 statistical data by clicking the **Transaction**, **SQL statistics**, and **HTTPC statistics** buttons on the upper right.



Transaction statistics

In the **Transaction** tab, you can sort the transactions performed at a specific time point by conditions such as response time and execution count, and view the top 30 lists. You can check the items with long response times or high execution counts first.

> **Avg memory allocation query settings**
>
> If the **Avg memory allocation** entry is not displayed in the transaction statistics of a past time point, check whether the following option is enabled. However, this option may have a performance impact in the applications with a high average TPS, so it is recommended to apply it only in the **environment with an average TPS of 300 or less**.
>
> - **trace_malloc_enabled** Boolean
>
> Default `false`
>
> It collects average allocated memory per transaction.

SQL statistics

In the **SQL** tab, you can view the top 30 lists of SQL queries executed at a specific time point, sorted by the execution count, total time, and such. When the DB load increases, you can find which SQL queries are costing the most among those executed at the same time.

HTTPC statistics

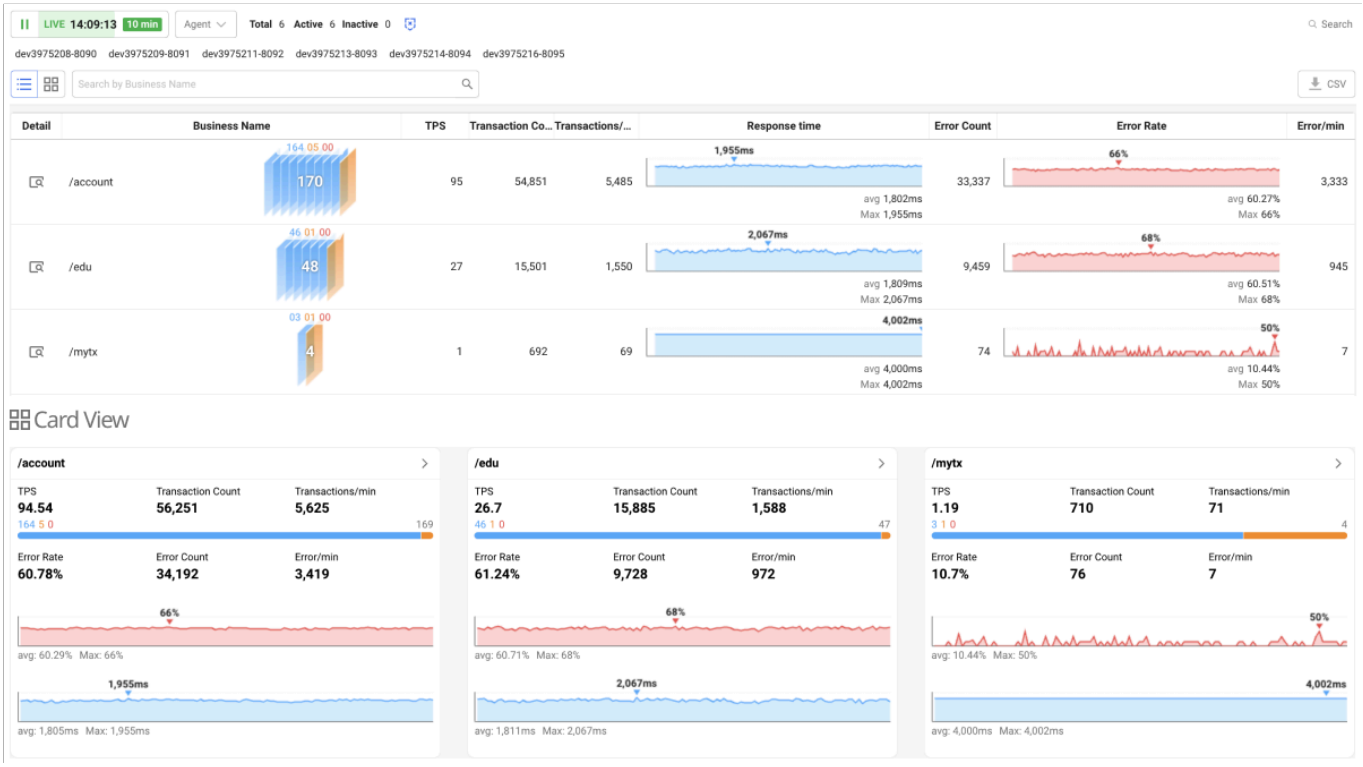
You can sort the external HTTP calls in the **HTTPC** tab to view the top 30 lists. You can check if there is any bottleneck when there are many external APIs or internal HTTP requests.

Business Dashboard

Home > Select Project > Dashboard > Business Dashboard

The **Business Dashboard** allows you to monitor application performance by business unit (App Context).
You can analyze both real-time and historical performance data for each business context, defined by URL patterns or paths.
By viewing key metrics like TPS, response time, and error rate per business context, you can quickly identify performance issues in specific operations or services.

Layout Overview



Time Range

Use the time selector at the top left to set the desired time range for data retrieval.
To the right of the time selector, you can view the status of agents connected to the project to immediately check whether the monitored servers are running.

- **Live Mode**: Auto-refreshes every 5 seconds, displaying data from the last 10 minutes.
- **Historical Mode**: Allows you to query data up to the past month.
- **Data aggregation unit**:
  - Under 3 hours: Raw data
  - 3 hours to 3 days: 5-minute statistics
  - Over 3 days: 1-hour statistics

Agent Selection

Below the time selector, you can see the list of agents under the selected project.
Click an agent to highlight it with a blue tag and filter the dashboard to show only its data.

> For more information on agent selection, see the following.

View Mode

You can toggle between two view modes based on your preference by clicking the icons on the top left:

- **Table View**
  Displays all business items in a table format, making it easy to compare and sort data.
  Suitable for quickly understanding the overall situation at a glance.
- **Card View**
  Presents each business item as a card.
  You can choose from 1 to 4 columns, making it flexible depending on your monitor size or working environment.
  Ideal for dashboard-style viewing or focusing on specific business performance.

Sort Options

You can select a metric to sort data by task in both the table view and the card view.

Search by Business Name

Use the search field to quickly locate a specific business by name.

Task Alias Settings

Click the button to open a drawer where you can assign aliases to task names displayed as URL prefixes.

- Enter the task name (URL prefix) to apply the alias to in the left column
- Enter the alias to apply to the task name in the right column
- Use the **Add Item** button to add an entry, and the **Delete** button to remove an entry
- Click the **Apply** button at the bottom to apply the changes

CSV Download

Click the CSV icon next to the search bar (top-right) to download performance data for all business items on the screen as a CSV file.

Business Performance Metrics

For each business unit, the following key performance indicators are available.
These metrics include both real-time data and aggregated stats over the selected time range.
Charts are available for **active transactions**, **response time**, and **error rate**.

- **Active Transactions**: Shows the number of currently processing transactions via a speedometer for intuitive real-time load insight.
- **TPS (Transactions Per Second)**: Measures system throughput.
- **Transaction Count**: Total transactions collected during the selected period.
- **Transactions/min**: Average number of transactions per minute, helpful for identifying traffic patterns.

- **Response Time**: Includes average and max response times; trends are shown in charts.
- **Error Count**: Total errors during the selected period.
- **Error Rate (%)**: Percentage of failed transactions.
- **Errors/min**: Average number of errors per minute.

Business Detail Analysis

Top Transaction Stats

Click the detail icon in either the table or card view to open a drawer on the right showing the top 30 transactions for the selected business context.

> How to open detail view:
>
> - Table View: Click in the left-most column.
> - Card View: Click the arrow (>) at the top right of the card.

Steps:

1. Click the detail icon in table or card view.
2. A drawer will open on the right displaying top 30 transactions for that business.
3. Use the sort dropdown (default: by count).
4. Available transaction data includes:
    - Transaction name
    - Detailed analysis ()
    - Count
    - Errors
    - Avg time (ms)
    - Max time (ms)
    - Avg CPU time

- Avg memory allocation
- SQL count
- Avg SQL time (ms)
- SQL fetch count
- Avg SQL fetch count
- HTTP call count
- Avg HTTP call time (ms)

5. Use the search bar at the top of the drawer to filter specific transactions.
6. Click the detailed analysis icon for each transaction to view more information.
7. If a transaction event based on the URL occurs within the selected time range, the configured level is displayed.
   If the level is ⚠ (Warning) or ⚠ (Critical), you can click the icon in the Health column to view the event history.

Event History

If a transaction event based on **context** is triggered within the selected time range, the defined level is displayed. Click the icon to view the event history.

- You can check the event name, occurrence time, and message.
- Use the search function to find specific events.

Agent Configuration

To use the Business Dashboard, agent configuration is required.

Supported Agent Versions

- **Java**: 2.2.40 or higher

Required Setting

```
# Enable business context monitoring
app_context_enabled=true
```

Optional Settings

```
# Parser type (default/prefix/match)
app_context_parser=default

# Reset parser's internal cache/settings
whatap.app_context_parser_reset=0

# Depth of path segment to extract
whatap.app_context_path_depth=1

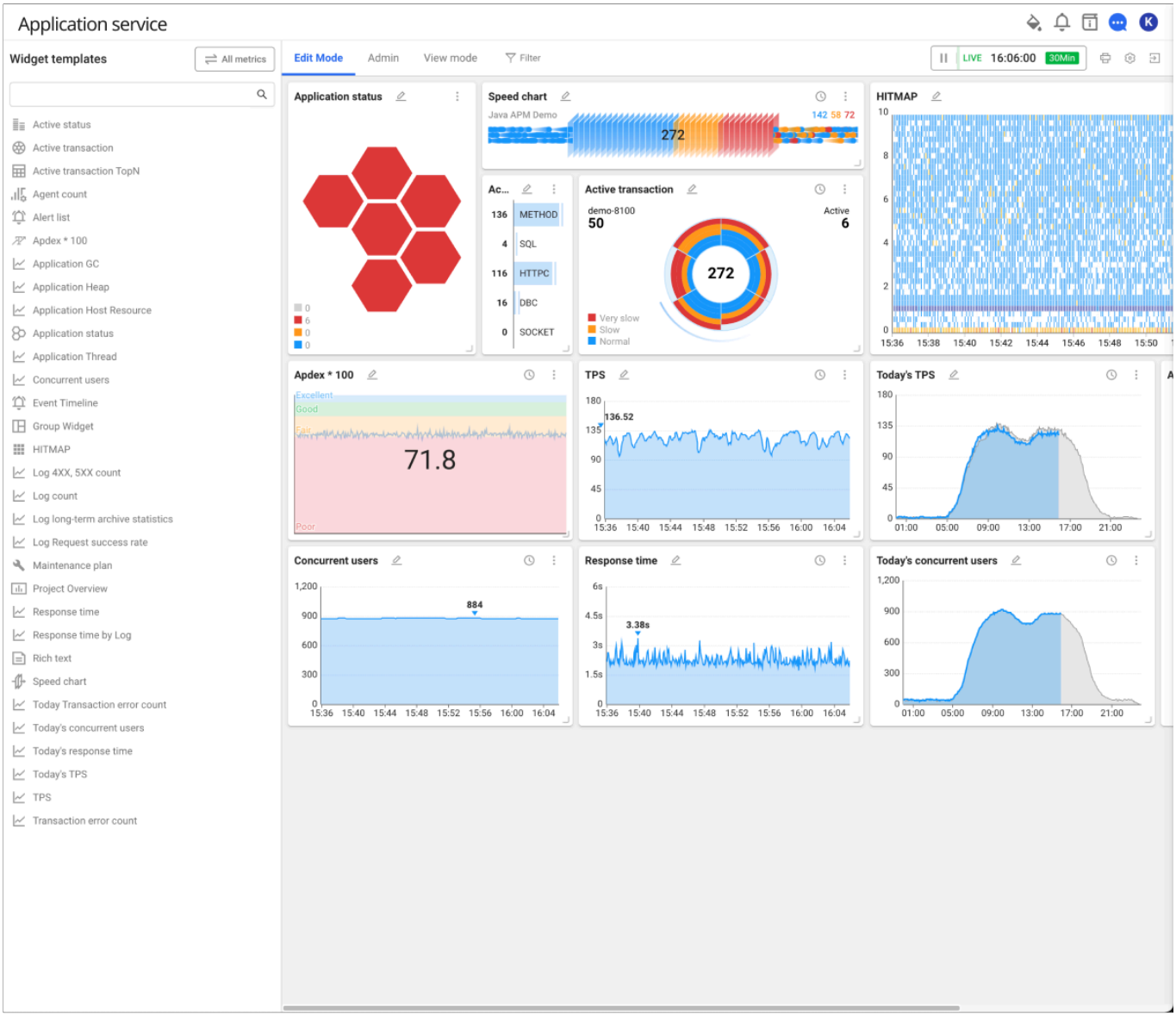
# Mapping of names to monitored paths (comma-separated)
whatap.app_context_path_set=name1@url1,name2@url2,url3,name4@url4
```

- If no data appears in the dashboard, verify the agent settings.
- If the agent version is too old, upgrade to the latest supported version.
- Business context can be defined automatically or manually based on URL patterns.

Flex Board

Flex Board is a custom integrated dashboard. You can create a real-time dashboard with the project data such as applications, servers, databases, and containers.

You can quickly complete the initial setup with preconfigured templates and build dashboards in your preferred format. By adding data widgets or modifying properties, you can display information in the form you need, and with filtering functions, you can narrow down monitoring targets. When you set a time range, you can check data from a specific point in time, and by using auxiliary charts, you can analyze data in various ways. Dashboards can be added to favorites for easy access, and personalized dashboards can be copied to other accounts for reuse.



The usage scope of the Flex board differs depending on the following entry:

Home > **Integrated Flex Board**

- When creating a widget, all projects that can be viewed are provided as options.
- The dashboard is saved in your account and can be shared with other users with the "To copy" feature.
- Individual account dashboards are not affected by the role, but the shared dashboards are read-only and cannot be edited.

Home > Select Project > **Dashboard** > **Flex Board**

- When creating a widget, the related project data is automatically entered.
- The created Flex board is automatically shared with other project members.
- The users with the project **Edit** role or **Project flexboard update** role, and **site admin** role can modify the board.

Users with edit permissions for the Flex Board can use the following features:

- Export and import dashboards as JSON files
- Check data requests and responses within the dashboard
- View and edit widget configuration options in JSON format

In the **Flex Board** menu within the project, only the users with the dashboard editing role can access **Edit Mode**, **Admin** Mode, and **Filter**.

- Accessible member permissions
  - Project **Edit** role
  - **Project flexboard update** role
  - **Site Admin** role
- For more information about the **View mode** and **Filter**, see the following.
- For more information about the member roles, see the following.

Flex Board User Guide

> You can build dashboards using various widget templates, including metrics widgets.
> For details on the features and usage of each template, see the Widget template document.

Apply a preset template

Use a preset template to speed up the initial setup, then modify it as needed.

1. Select the **Flex Board** menu.
2. From the **Template** list on the left, select the template you want.
3. When the **Flex board management** window appears, enter a name in **Rename**.
   - In **Layout**, select a widget and drag it to the position you want.
4. Click the **Save** button. A Flex board is created based on the selected template.
   - You can check the created board in the board list of **Flex Board**.

Create a Flex board

You can create a Flex board and place widgets to build your own dashboard.

1. In **Flex Board**, click the **Create Dashboard** button at the top right.
2. When the **Create Dashboard** window appears, enter the dashboard **Name** in the input box.
3. Select a widget placement method, then click the **Create Dashboard** button.
   - **Fixed layout flexboard**: free widget placement with pixel basis
   - **Responsive flexboard**: responsive grid layout based on browser size
     (configure the grid, then click the **Create Dashboard** button)
4. In the **Widget templates** selection screen, choose the widgets to place on the dashboard.

- To add metrics widgets, click the **All Metrics** button on the right of **Widget templates**.

> ⓘ
> - **Widget templates**: choose from a predefined widget list that summarizes key indicators commonly used in monitoring.
> - **All metrics**: create widgets based on all metrics data collected in your project. For details about metrics widgets, see the following document.

5. Move the placed widgets or adjust their size.
   - When a crosshair cursor appears at the top of a widget, click and drag the widget to move it.
   - When an arrow cursor appears at the bottom right of a widget, click and drag to resize it.
6. After you finish placing widgets, select **View mode** to review the layout. To edit the layout again, select **Edit Mode** and adjust widget placement.
7. When all steps are complete, click the (**Go to list**) icon at the top right.
   - You can check the created board in **Flex Board** under **Dashboard list**. Select the created board to view the new Flex board.

> ⓘ We plan to add dashboard templates or widgets based on customer feedback. If you need a specific dashboard template or widget, contact us at support@whatap.io.

Edit a Flex board

> ⓘ **Flex board edit permissions**
> - **Home** > **Integrated Flex Board**: this is a personal account dashboard, so permissions do not apply; however, dashboards shared as read only cannot be edited.
> - **Home > Select project** > **Dashboard** > **Flex Board**: users with project edit permission, project Flex board edit permission, or Site Admin permission can edit.
>
> For details on member permissions, see the following document.

1. In **Flex Board** under **Dashboard list**, click the edit icon. The **Flex board management** window appears on the right.
2. In the **Flex board management** window, configure Name, Project, Layout, widget.json, and option.json.
   - **Rename**: edit the dashboard name.
   - **Project**: select the project from which widgets will fetch data.

     > You can configure the **Project** option only in **Home** > **Integrated Flex Board**.
     > If you entered from **Integrated Flex Board**, you can select the data source project for each widget.
     >
     > 1. Click the + button at the top right of a widget, then search and select one or multiple projects.
     > 2. After selecting projects, click the **Apply** button.

   - **Layout**
     - Drag widgets to move them.
     - Click and drag the bottom right of a widget to resize it.
     - To delete a widget, click the icon at the top right of the widget.
   - **Widget.json**: import dashboard widget settings in JSON format. Click the icon to copy the JSON.
   - **Option.json**: import dashboard option settings in JSON format. Click the icon to copy the JSON.

Delete a Flex board

1. In **Flex Board** > **Dashboard list**, click the icon for the item to delete.
2. In the confirmation message, click the **Delete** button.

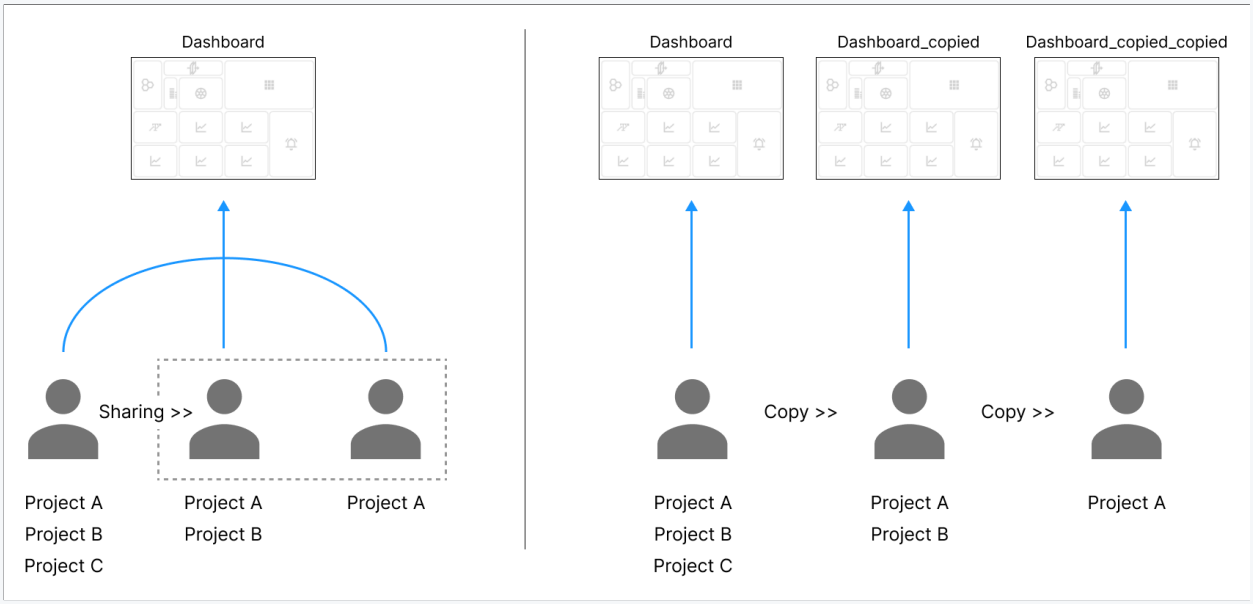
> Dashboard deletion behavior differs for owners and shared users.
>
> - If the dashboard owner deletes a dashboard, it is deleted for all user accounts with whom it was shared.

- If a user who received a shared dashboard deletes it, it is deleted only from that user's **Dashboard list**. The original dashboard remains.

Share a Flex board

The Flex board sharing feature is available only in **Home > Integrated Flex Board**.
It is not provided in a project's **Flex board** menu.

Difference between share and copy



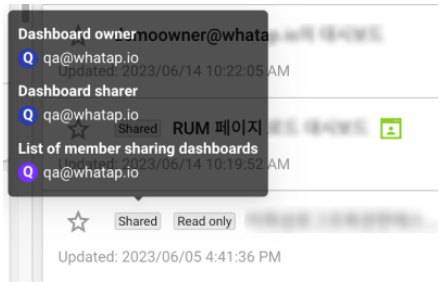
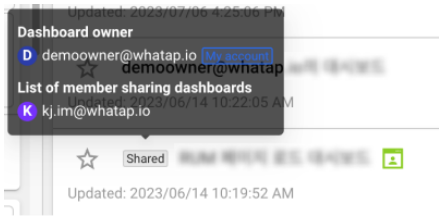
- Share: all members you share with view the same dashboard.
- Copy: if you copy a dashboard to another member or to yourself, you view a separate copied dashboard.
- Dashboards shared or copied as **Read only** cannot be edited. They can be edited only if shared as **Edit mode**.
- Members who receive a dashboard as **Read only** can also **re-share as read only** or **copy** that dashboard.

- You can share a dashboard only with members who belong to at least **one** of the projects included in the dashboard. Check whether the member to share or copy to belongs to the relevant project.



You can share a Flex board created in **Integrated Flex Board** with other members. When the owner edits a shared dashboard, the changes apply to all members it was shared with.

1. In **Integrated Flex Board** under **Dashboard list**, click the icon for the dashboard to share.
2. When the **Sharing a dashboard** window appears, select one or multiple members to share with, or enter accounts directly.
   - If you select **Read only**, members cannot edit the shared dashboard.
   - Share targets must belong to at least one of the projects included in the dashboard.
3. Click the **Share** button.
   - Shared items display the **Shared** tag. Hover over the **Shared** tag to view information about the members you shared with.
   - Items shared as read only display the **Read only** tag.

When you receive a shared dashboard	When you share a dashboard (dashboard c
	

Dashboard sharing conditions

- The dashboard sharing feature is available only in **Home** > **Integrated Flex Board**.

- Anyone can share their own dashboard, and you can re-share or copy a dashboard you received.
- Dashboards received as **Read only** cannot be edited; dashboards received as **Edit mode** can be edited.
- If the **owner** deletes a dashboard, it is deleted for all accounts that received it.
- If a **recipient** deletes a dashboard, it is removed only from that user's **Dashboard list** and the original remains.
- For projects in a received dashboard where you do not have **View** permission, you cannot view data.

Copy a Flex board

Users can copy a Flex board and send it to other members.

1. In **Integrated Flex Board** under **Dashboard list**, click the icon.
2. Choose a copy option.
   - **Copy to me**: copy to your own account.

     a. An item with the `_copied` suffix is added to **Dashboard list**.
   - **Copy to others**: copy to another member's account.

     a. Click ＋ **Account**.
     b. Select the email of the member to copy to and click the **Copy dashboard** button.
     c. An item with the `_copied` suffix is added to that member's **Dashboard list**.

Share as a JSON file

You can export Flex board settings to a JSON file to share with other members, or import settings from other members.

Export

1. Go to **Integrated Flex Board** or a project's **Flex board** menu.
2. In **Dashboard list**, click the button.
3. Choose an export option.
   - **Include Project Information**: include dashboard project information in the JSON file when sharing.
   - **Except Project Information**: share without including dashboard project information in the JSON file.
4. Download the JSON file and send it to the member you want to share with.

Import

1. Go to **Integrated Flex Board** or a project's **Flex board** menu.
2. Click the **Import** button at the top right.
3. Select the downloaded JSON file.

> Only members with **Edit** permission for the project can use this feature.

Flex board screen modes

Features vary by Flex board screen mode.

> **Accessible member permissions**
>
> Only users with **Edit** permission in a project's **Flex Board** menu can access **Edit Mode**, **Admin**, and **Filter**. For details on member permissions, see the following document.
>
> - **Project Edit** permission
> - **Project flexboard update** permission
> - **Site Admin** permission

Modes

Available features differ by **Flex Board** screen mode.

- **Edit Mode**: place widgets from the widget template list on the Flex board layout; edit, delete, or reposition widgets.
- **Admin**: manage widget properties in JSON format. Click the 🔧 button at the top right of a widget on the layout.
- **View mode**: after finishing layout, sizing, and detailed options, monitor each widget's data in real time like a dashboard.

Common features

- Data filtering
- Real time data time range settings
- Print mode
- Flex board management
- Data merge options

Data filtering

Use data filtering to narrow monitoring targets and build an efficient Flex board.

1. At the top of the screen, click the ▽ **Filter** tab.
2. In the **Filter** window on the right, select filter targets. To select all filter targets, choose **Select all**.
   - **Agent** option: select agent types or an application list, with **Search** available
   - **Metrics** option: select a project or agent type; add **+ And** conditions, or auto select by string with **Contain** or **Exclude**
3. After selecting filter targets, click the **Apply** button.
4. Save frequently used conditions with the **Save filter** button.

   a. Enter a filter name and select **Save**. **Saved filters** appears at the bottom of the **Filter** window.

Filtering conditions

Filtering conditions are divided into agent type and metrics type.
Agents provide only basic monitoring targets as selectable options, while metrics provide selectable tags for all metrics data collected in the project. The metrics type is useful when you collect data in a non agent form and want to apply filtering. For details on metrics, see the following document.

Type	Agent	Metrics
Selection options	WhaTap project monitoring targets (applications, servers, databases, containers, Kubernetes PODs and NODEs)	All metrics tags including agents
Scope	All widgets	All widgets that use metrics data
Filtering method	Checkboxes	**Equal**, **Contain**, **Exclude**

Set the lookup time

Use the time selector at the top right of **Flex Board** to set data lookup time for widgets.

- ||: stop real time mode and set a specific lookup time
- ▷: switch back to real time mode

Print

You can save the Flex board screen as a PDF and print it.

1. On the Flex board screen, click the 🖨 icon at the top right to switch to print mode.
2. Configure options.
   - **Viewing filter options**: toggle filter options on the screen
   - **Convert to {mode} orientation / Convert to {mode} orientation**: change orientation to horizontal or vertical
3. After finishing, select **PDFDownload**.

4. To exit print mode, click the ✕ **Exit print mode** at the top right.

Flex board management

You can change the Flex board layout; move, resize, or rename widgets; and use other management features. You can also open **Flex board management** by clicking the ✎ icon in **Dashboard** > **Flex Board** under **Dashboard list**.

1. On the Flex board screen, click the ⚙ icon at the top right to open the **Flex board management** window.
   - **Rename**: change the Flex board name
   - **Layout**: move, resize, and delete widgets
   - **Don't Allow Other Users to Modify.**: restrict modifications
     - Members with edit permission can edit, delete, and modify; if you enable the option, it changes to **Don't Allow Other Users to Modify.: On**
   - **Widget.json**: edit and manage widgets on the Flex board in JSON
   - **Option.json**: edit and manage Flex board settings in JSON (project info, Flex board filters, time selection, and other settings)

   > ⚠ The **Widget.json** and **Option.json** features are available only to dashboard administrators.

2. After completing all settings, click the **Save** button.

Add a dashboard to favorites

You can add a Flex board you created to favorites for quick access. In **Dashboard** > **Flex Board** under **Dashboard list**, click the ☆ button for the dashboard to add. It changes to ★.

Manage Flex Board Widgets

You can customize your dashboard by handling the widgets placed on the Flex board.

> You can build dashboards using various widget templates, including metrics widgets.
> For details on the features and usage of each template, see the Widget template document.

Adding a widget

You can add any widget to the created Flex board.

1. In **Dashboard list** of **Flex Board**, select a dashboard to add the widget.
2. On the upper left of the screen, select **Edit Mode**.
3. Select a desired widget from the **Widget templates** list on the left. To add a metrics widget, select **All metrics** and then click a desired widget.

- If you have entered **Integrated Flex Board** on the Home screen and added a widget, click **Project selection** and then select target projects to collect.
- If you have more widgets to add, repeat the same procedure.

4. After all steps are finished, select **View mode** to check the layout and then select (**Go to list**).

Changing the widget attributes

In **Dashboard list** of **Flex Board**, select a board to change the widget properties. It goes to **Edit Mode**.

Changing the widget name

1. On the right of the widget name to change, select .
2. Set the widget values for **Title**, **Font size**, and **Color**, and then select **Save**.

- If the **Apply to the entire widget** toggle button is enabled, the font size and color are applied to other widgets.
- To lock the widget title, select Configuration icon and then click **Save**.

Time Setting

1. On the upper right of the widget to set the time, select .
2. Set the time in the **Select a time** window.
   - To apply the set time to the dashboard, check **Use dashboard settings**. The time setting on the dashboard is located on the upper right.
   - To apply a desired time setting, uncheck **Use dashboard settings**. Use < or > to set the desired time.
3. After setting the time, select ×.

> Depending on the data type of the widget, the time setting may not be available.

Setting the output data options

By setting the output data options of the chart widget, you can select the desired data.

1. At the top right of the widget, click the ⋮ icon to configure the output data options.
   - You can filter data by selecting types for project, agent, application, etc.
   - You can change the type of the graph displayed on the chart by selecting individual or merged data.
   - Select ⋮ > ⚙. You can change the option details through the **Chart settings** window. The options that can be set are as follows.
     - **Chart type**: This option allows you to display the current aggregated figures on the chart.
     - **y axis Chart max value**: The chart maximum value on the Y axis can be selected with one of **Auto** and **Fix**.
     - **Set horizontal line**: Select **Add** to display the threshold or range as a horizontal line. You can

apply multiple items added. After adding the horizontal line, select **Apply** to apply it to the chart.

- **Show chart max value**: The maximum value of the graph is displayed on the chart.
- **Sub-chart**: You can add a sub-chart to the main charts of the widget. Select a type of sub-chart and then select its layout, size, and whether or not to merge data. To apply to other widgets, select **Apply to the entire widget**. Sub-charts are useful to view data in various ways.

2. After setting all options, on the left of the **Chart settings** window, click ✕ or click the outside of the **Chart settings** window.

> Depending on the widget's data type, available options may differ.

Changing the metrics widget data

You can change the data query conditions of the metrics widget.

1. In **Flex Board** > **Dashboard list**, select the dashboard where you want to change the widget data.
2. In **Edit Mode**, click the ⋮ icon at the top right of the widget you want to modify.
3. Click the <> icon to open the data settings edit window.
4. Change the settings referencing the following, and then select < **Apply to the widget**.
   - Metrics options that are created by default when creating a widget appear on the tab.
   - Metrics options are **Category**, **Field**, and **Tag**.
   - The **Category** currently restricts the editing function.
   - In **Field** and **Tag**, multiple items can be selected. The chart data is grouped using the selected tag options.
   - Select 🔍 to query the original data.
   - For more information about**Object Merge** and **Time merge**, see the Data merge options.

Data merge options

The metrics widget placed on the **All metrics** list of the Flex board provides the data merge options. Data merge provides the following methods: **Object Merge** and **Time merge**.

- **Object Merge**

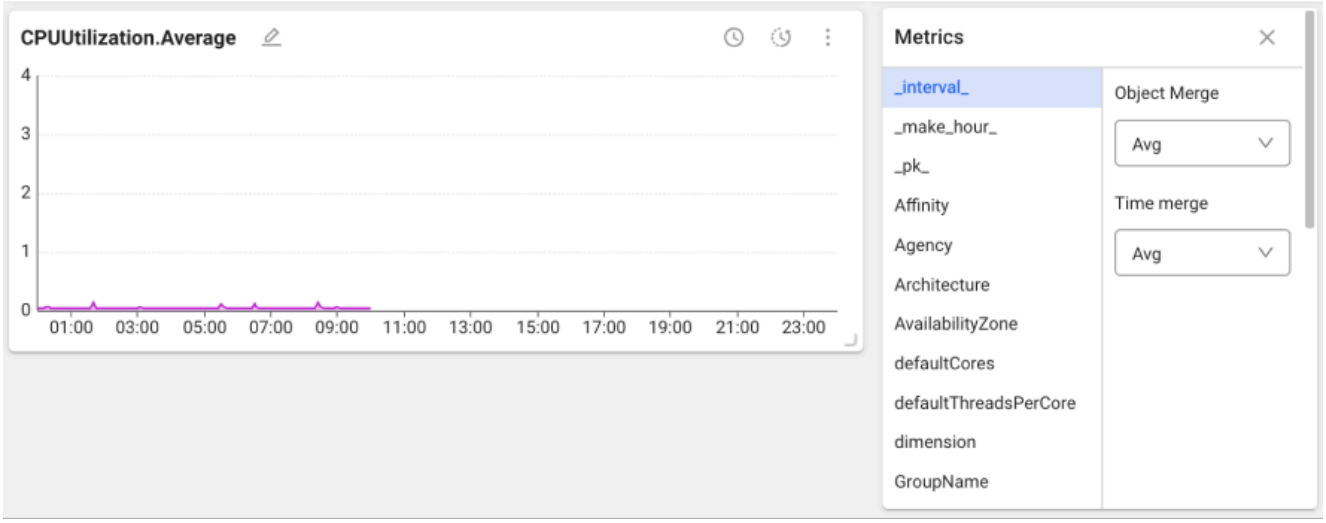
  It is used to group the chart data based on the tag (e.g. agent name) selected on the left. At this time, if the tags match among the data with different fields (e.g. CPU), the corresponding data is merged.

- **Time merge**

  It is mainly used if the lookup time is too long. Data with the same fields in the original data are merged at regular time intervals (5 minutes or 1 hour) depending on the time range. This method is called "Time Merge."

- **AWS ColudWatch Metrics**

  In case of AWS monitoring metrics, the statistical method is added after the field name in the form of suffix. Data merging defaults reference this suffix.



1. In **Edit Mode**, select **All metrics** and then change the **Widget templates** list to the **All metrics** list.
2. Place the metrics widget on the layout.
3. On the upper right of the metrics widget, select ⋮.
4. If a tag (e.g. agent name) option is selected, the data merge options appear. Depending on the widget size, the metrics option's location may differ.

Copying a widget

1. Click ⋮ on the upper right of the widget.
2. Click ⧉.
3. If the **Copy** window appears, click ＋ **Add widget**.
4. Change the widget name in the list.
5. Click **Apply**.
6. In **Edit Mode**, you can see the copied widget.

Delete Widget

1. Click ⋮ on the upper right of the widget.
2. Click ✕.
3. If the confirmation message appears, click **Delete**. To cancel the deletion, select **Cancel**.

Analysis Report

It provides a menu that allows you to analyze the application environment from various perspectives. You can identify the zones with high load or poor performance and provide various performance metrics to check the daily status. You can also trace and analyze transactions for performance measurement.

Through the analysis provided by WhaTap, you can analyze key performances, identify the problems, and respond to any failures that may occur.

Daily APP. Status

You can check the daily trends of the application's key performance metrics through the hourly chart. For more information, see the following.

Performance Trend

By charting the trends of key application performance metrics for a specific period, you can quickly identify the zones with high load or poor performance, and point the time zone when a problem occurred. For more information, see the following.

Metrics Chart

You can search the metrics through visualized charts. For more information, see the following.

Metrics Search

You can view the performance metrics for categorized applications by using the data organized based on the tags and fields. For more information, see the following.

Stack

It collects transaction stacks and provides the usage statistics for executing methods and the statistics on commonly used stacks. Through the stacks for running transactions, you can identify long-running methods or short-running but frequently running methods. For more information, see the following.

Cube

Performance statistics created every 5 minutes are called "Cubes." A statistical analysis tool for post analysis can set a zone with high response time, error count, and throughput. During the same time zone, the problem factors can be identified or the services can be monitored. The geographic traffic distribution and resource usage can be identified. For more information, see the following.

Hitmap

It provides a distribution chart expressing the response time to user requests over time. Depending on the type of distribution chart, it is helpful to identify the type of failure occurred. For more information, see the following.

Multiple Transaction Trace

It traces calls between transactions associated with different agents or projects. The transactions and trace data are provided so that you can see various call relationships at a glance within or between systems, and identify where problems occurred for improvement. For more information, see the following.

Transaction Search

You can view the statistics on the performance properties and execution status for each transaction for a specific period of time. You can view all transactions and quickly check their status (normal or error). Through the provided information, you can find which transactions are busy and resource-consuming. Various filters can be enabled depending on the transaction properties, making it useful in finding transactions with specific conditions. For more information, see the following.

Daily Application Status

Home > Select Project > **Analysis** > **Daily APP. Status**

You can check the daily trends of the application's key performance metrics through the hourly chart.



You can check the application status for the day. Set the search conditions (**Select date**, **Type**, **Application**) and then select . The conditions of the **Type** option are as follows:

Classification	Description
Agent	Application agent name included in the project

Classification	Description
Agent kind	Group unit categorized as the `whatap.okind` option in agent settings
Agent node	Group unit categorized as the `whatap.onode` option in agent settings

The provided charts are as follows: The numbers on the right of the chart display the accumulated value or average, and maximum value for the day.

- **Transaction per hour**
- **Users per hour**

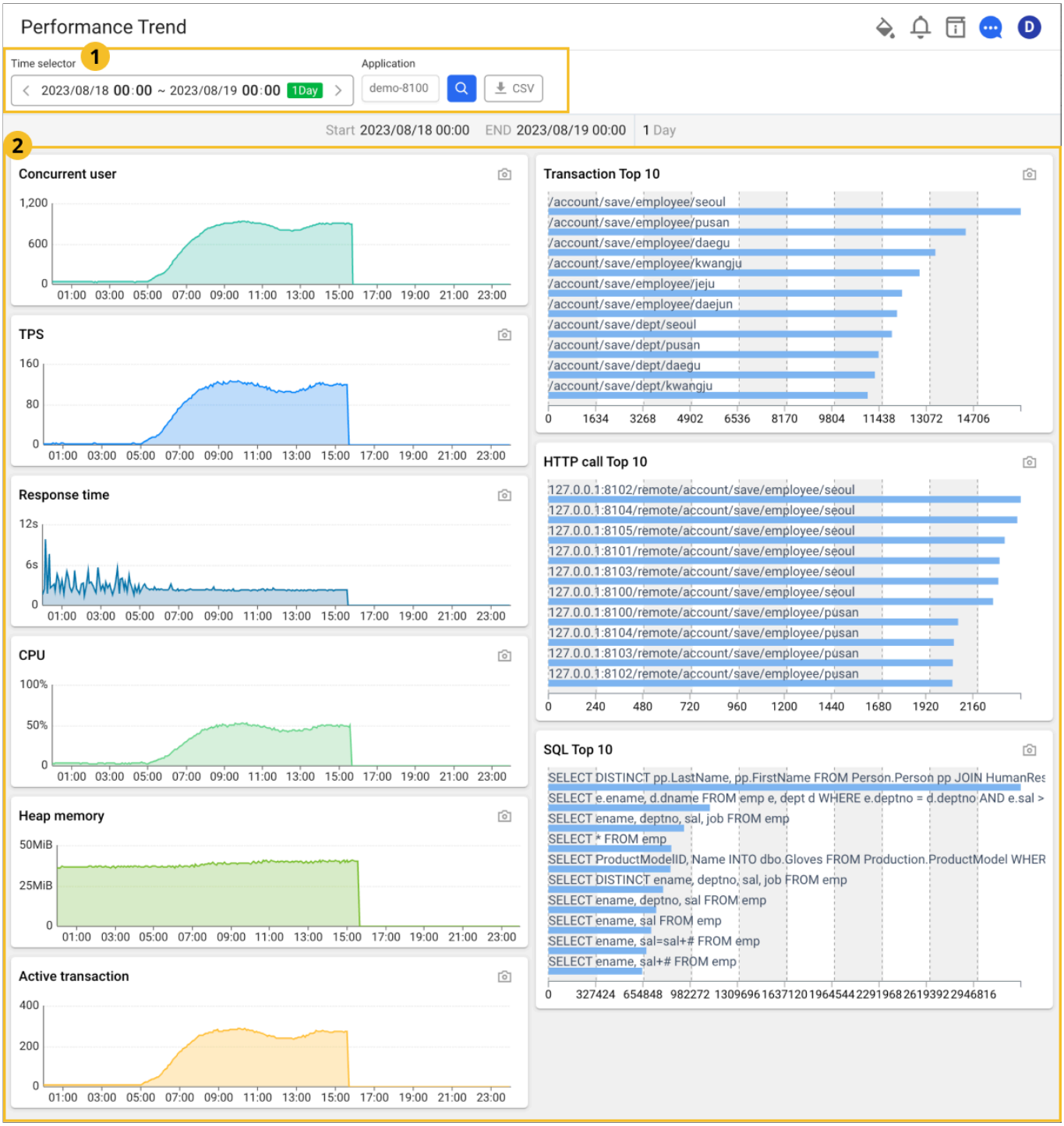
> **Today's visits** is the sum of users accumulated for the day without duplication. Instead of simply adding up, it is calculated using **HyperLogLog**. **HyperLogLog** is a probabilistic data structure that estimates the number of elements in a set with very little memory.

- **HTTP call count per hour**
- **SQL count per hour**
- **SQL fetch count per hour**
- **Response time**
- **Transaction per second**
- **Active transaction**
- **Errors in 5mins**
- **Error rate**

Performance Trend

Home > Select Project > **Analysis** > **Performance Trend**

On the initial screen of the WhaTap Monitoring service, select a project and then click **Analysis** > **Performance Trend**. You can see numerous metrics about the performance within the specific time range to view.


Performance Trend
Time selector
1
2023/08/18 00:00 ~ 2023/08/19 00:00
1Day
Application
demo-8100
CSV
Start 2023/08/18 00:00
END 2023/08/19 00:00
1 Day
2
Concurrent user
1,200
600
0
01:00 03:00 05:00 07:00 09:00 11:00 13:00 15:00 17:00 19:00 21:00 23:00
TPS
160
80
0
Response time
12s
6s
0
CPU
100%
50%
0
Heap memory
50MiB
25MiB
0
Active transaction
400
200
0
Transaction Top 10
/account/save/employee/seoul
/account/save/employee/pusan
/account/save/employee/daegu
/account/save/employee/kwangju
/account/save/employee/jeju
/account/save/employee/daejun
/account/save/dept/seoul
/account/save/dept/pusan
/account/save/dept/daegu
/account/save/dept/kwangju
0 1634 3268 4902 6536 8170 9804 11438 13072 14706
HTTP call Top 10
127.0.0.1:8102/remote/account/save/employee/seoul
127.0.0.1:8104/remote/account/save/employee/seoul
127.0.0.1:8105/remote/account/save/employee/seoul
127.0.0.1:8101/remote/account/save/employee/seoul
127.0.0.1:8103/remote/account/save/employee/seoul
127.0.0.1:8100/remote/account/save/employee/seoul
127.0.0.1:8100/remote/account/save/employee/pusan
127.0.0.1:8104/remote/account/save/employee/pusan
127.0.0.1:8103/remote/account/save/employee/pusan
127.0.0.1:8102/remote/account/save/employee/pusan
0 240 480 720 960 1200 1440 1680 1920 2160
SQL Top 10
SELECT DISTINCT pp.LastName, pp.FirstName FROM Person.Person pp JOIN HumanRes
SELECT e.ename, d.dname FROM emp e, dept d WHERE e.deptno = d.deptno AND e.sal >
SELECT ename, deptno, sal, job FROM emp
SELECT * FROM emp
SELECT ProductModelID, Name INTO dbo.Gloves FROM Production.ProductModel WHER
SELECT DISTINCT ename, deptno, sal, job FROM emp
SELECT ename, deptno, sal FROM emp
SELECT ename, sal FROM emp
SELECT ename, sal=sal+# FROM emp
SELECT ename, sal+# FROM emp
0 327424 654848 982272 1309696 1637120 1964544 2291968 2619392 2946816

1 Performance trend search conditions

- The time selector allows you to select a time range to search. The default search period is 1 day.
  - If the selected range is 3 hours or less, a 5-second trend is displayed using raw data.
  - If the selected range is within 3 days, a 5-minute trend is displayed using 5-minute statistics.
  - If the selected range exceeds 3 days, a 1-hour trend is displayed using 1-hour statistics.
- The application selector allows you to select all or specific applications to view.
- Select CSV CSV download icon, you can download the list data displayed on the screen as a CSV file.

2 Performance Trend Chart

- You can see the specified lookup time range at the top of the chart.
- In the chart area, you can see the **Concurrent user**, **TPS**, **Response time**, **CPU**, **Heap memory**, **Active transaction**, **Transaction**, **Top 10**, **HTTP call Top 10**, **SQL Top 10** and such.
- By selecting the **Snapshot** icon, you can snapshot charts except for widget options.

Concurrent user

Real-time users are counted based on unique browser IP addresses.

- When using raw data, the number of unique browser IPs that sent requests within 10 seconds is counted.
- When using 5-minute statistics, the number of unique browser IPs that sent requests in the past 5 minutes is counted.
- When using 1-hour statistics, the number of unique browser IPs that sent requests in the past hour is counted.

TPS

TPS (Transaction Per Second) means the number of transactions processed per second.

Response time

It indicates the average response time for requests. If the average is very high, there may be many slow

transactions or failures at any specified time.

CPU

It indicates the CPU usage of the application server. You can see the trend in usage change.

Heap memory

It indicates the **heap memory** usage of the monitored process. The changes in memory usage over time can be seen.

Active transaction

Active transaction indicates a transaction in progress. For more information about active transactions, see the following.

Transaction Top 10

We use the transaction call statistics to provide you with the top 10 list with the most frequent transaction calls.

HTTP call Top 10

We use the HTTPC statistics to provide the top 10 list with the highest numbers of HTTP call URLs.

SQL Top 10

It provides the top 10 list with the highest numbers of SQL calls using SQL statistics.

Cube Analysis

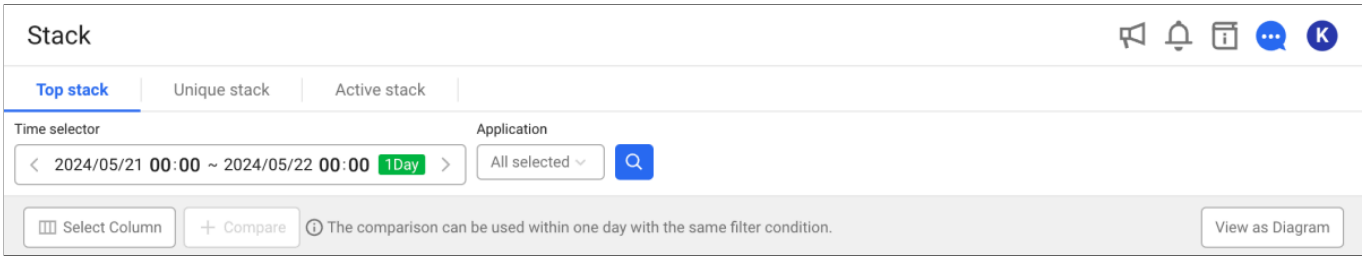
Home > Select Project > **Analysis** > **Stack**

You can check the call stack frequency and detailed information about methods and transactions in a specific time period. You can view **Top stack**, **Unique stack**, and **Active stack**. **Top stack** displays the usage of frequently called methods, while **Unique stack** analyzes the frequency of the same call stack. **Active stack** enables real-time performance analysis by regularly collecting stack information of active transactions.

> The stack analysis is supported in the **Java**, **Python**, and **.NET** product platforms.

Guide to major features

The following major features are available. Stack viewing allows you to quickly identify and resolve performance issues in your application. By checking the frequency of a specific method call, call path, response time, and the like, you can devise measures to improve the performance.



- **Selecting a stack type**

  You can select a desired stack from **Top stack**, **Unique stack**, and **Active stack** for a view.

  - **Top stack**: Information of the stack with the most frequent calls is provided. For more information, see the following.
  - **Unique stack**: Statistical information about the cases where the set of executed methods is the same, is provided. For more information, see the following.
  - **Active stack**: The stack information of the running transaction is provided. For more information, see the following.

- **Time selector / Start date**: Set the time to view. You can view the stacks for a specific time period or set a desired time period. Depending on the stack type, the lookup period may be restricted.

  > ◦ The maximum period to view **Top stack** and **Unique stack** is 3 days.

- **Application**: You can select an agent to view. You can select a specific agent. If nothing is selected, all agents in the project are viewed.
- **Select Column**: You can display or hide the desired columns.
- **Compare**: When searching within 1 day by the same filter conditions, you can compare the selected stacks over time. This allows you to see the changes in the call stack over time and see the performance trend. For more information, see the following.
- : When you expand a stack or transaction in the table list for each stack type, a snapshot icon appears on the utmost right. Selection of this icon allows you to save the stack or transaction information as an image file.

Basic usage guide

1. Select a desired stack type to view on the screen.
2. In **Time selector** or **Start date**, set the desired viewing time.
3. In **Application**, select an agent to view. If nothing is selected, all agents are viewed.
4. Select .

You can see the search results in the list of the stack table.

Top stack

Top stack provides information on the most frequently called methods. Through the call frequency of the method at the top of the stack, you can quickly find the method that has the most impact on the service. By checking the frequency of method calls, you can analyze the cause of the load on the CPU or memory. Through this, the measures to improve the performance can be derived.

WhaTap can use the thread stacks collected every 10 seconds (default value) to analyze the performance delay

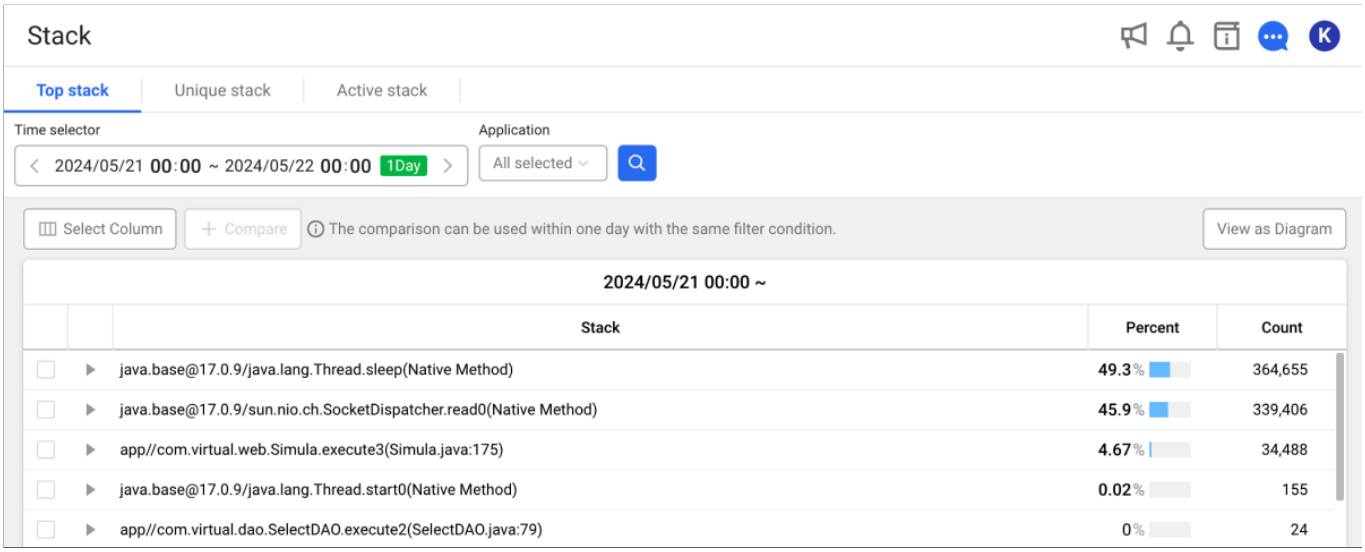
sections in the method level.

st1

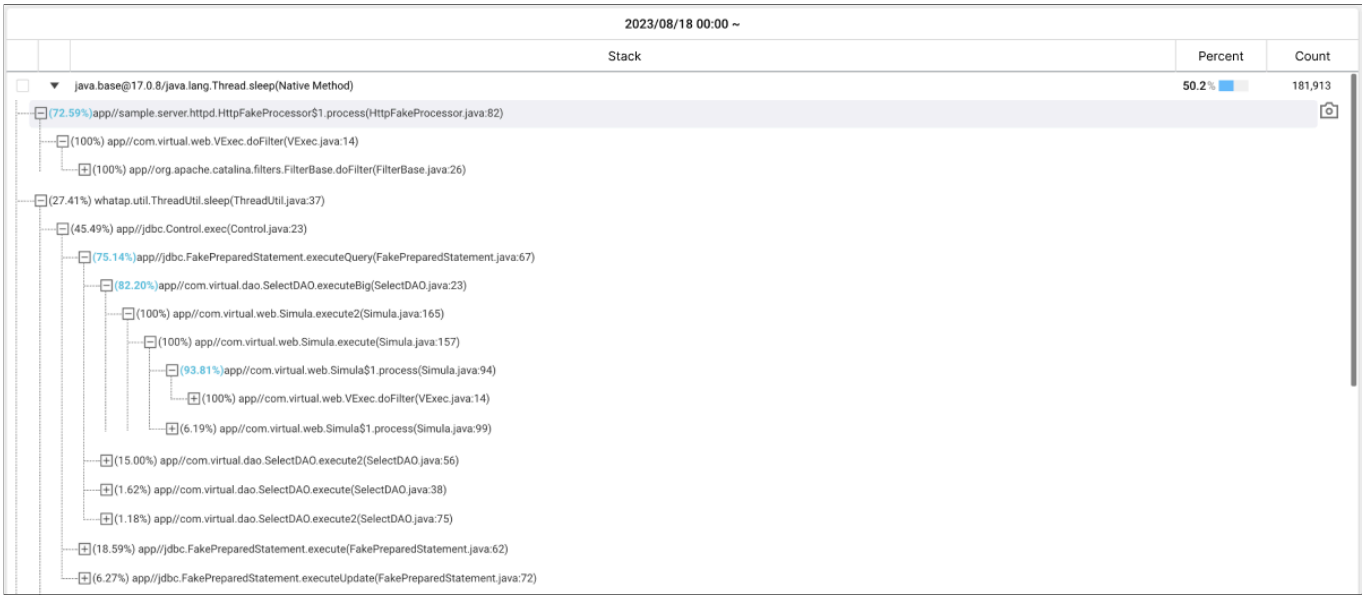
In the example stack, the **top line** is `socketRead0`.

```
java.net.SocketInputStream.socketRead0(Native Method)
```

Top line indicates that the thread to be dumped is executing that method. It may be caught momentarily, but it appears on the stack as a percentage of the sum of the module processing times. By calculating the frequency of this **top line** method, you can determine the method-level performance. WhaTap calls the **top line frequency statistics** as **Top stack**.



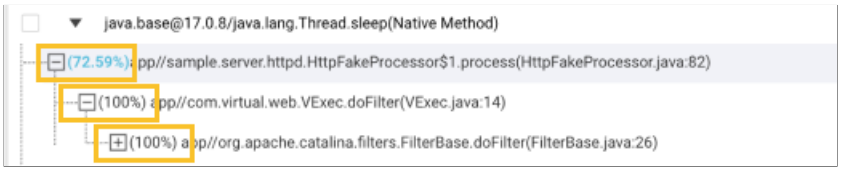
Through the **Top stack** analysis, you can analyze the frequencies of methods that were called. Based on each step on the stack trace, it provides the call rates between steps as percentage. The backlog frequency of the top step is calculated as percentage and the results are sorted in descending order.



If you select ▸ on the utmost left of each step, the call frequency reference of the upper step that calls the step is calculated as percentage.

The **Top stack** statistics are good to make decision with enough data. If the number of collected stacks is a prime number of less than 10, it is insufficient to have a statistical significance.

Top stack is useful to find tuning points that were difficult to recognize while tuning. The most frequent stacks can be determined to cause the most response delays in the application server. The rate on the utmost left is how much it affects the performance of the application server.



Even if it is a stable application server, a stack with a high call frequency is likely to cause performance degradation, so it is recommended to look at the class carefully.

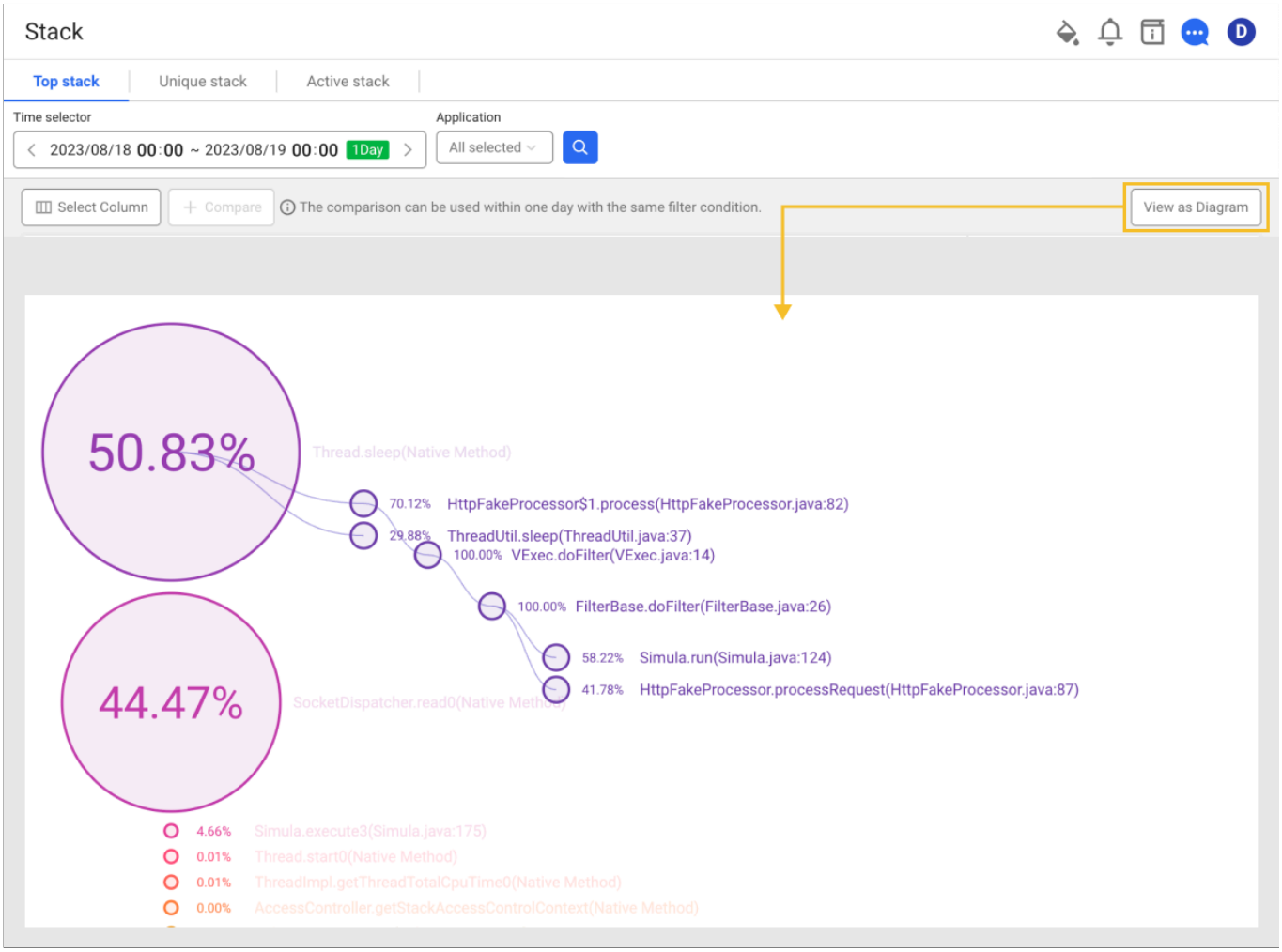
If you click **Top stack**, the frequency of calls on the top stack can be checked. Because the call relationship of the **Top stack** is one-to-one, the data accuracy may decrease as the depth of the **Top stack** goes down. Use the information on the sub-depths for reference purposes and proceed with tuning.

To improve the application performance, the module with a high backlog rate in the top-level step must be checked for bottleneck. For modules with high backlog rates, even a small performance improvement can lead

to significant enhancement throughout all applications.

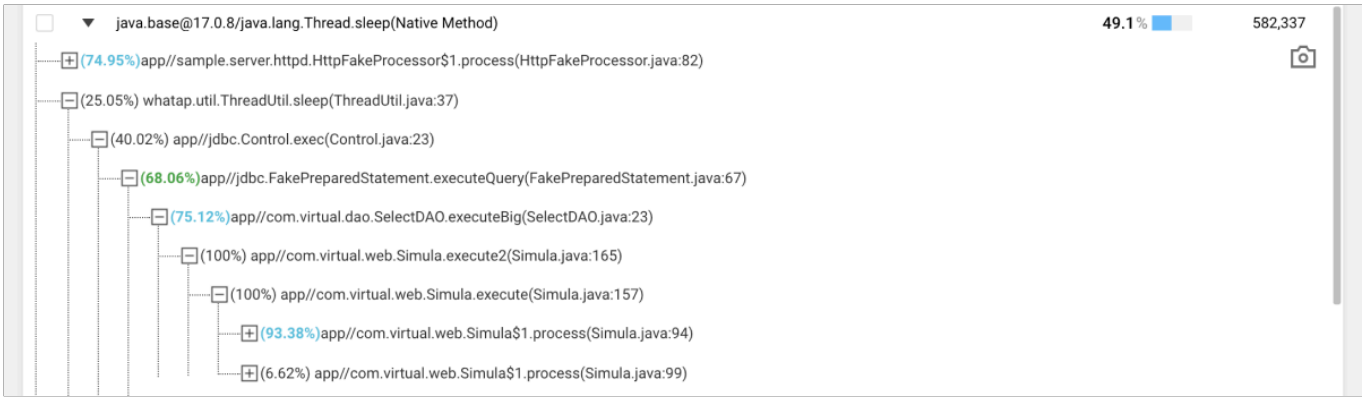
View as Diagram

It provides the diagram chart to help you easily understand the call steps of complex stacks. Select **View as Diagram** at the upper right of the stack table. This allows you to see at a glance the call frequencies and execution paths of each step.



- If you select each node, the stacks of lower depths appear.
- You can zoom in/out by scrolling the mouse or move it by dragging.
- To go back to the list of stack table, select **View as table**.

Precautions



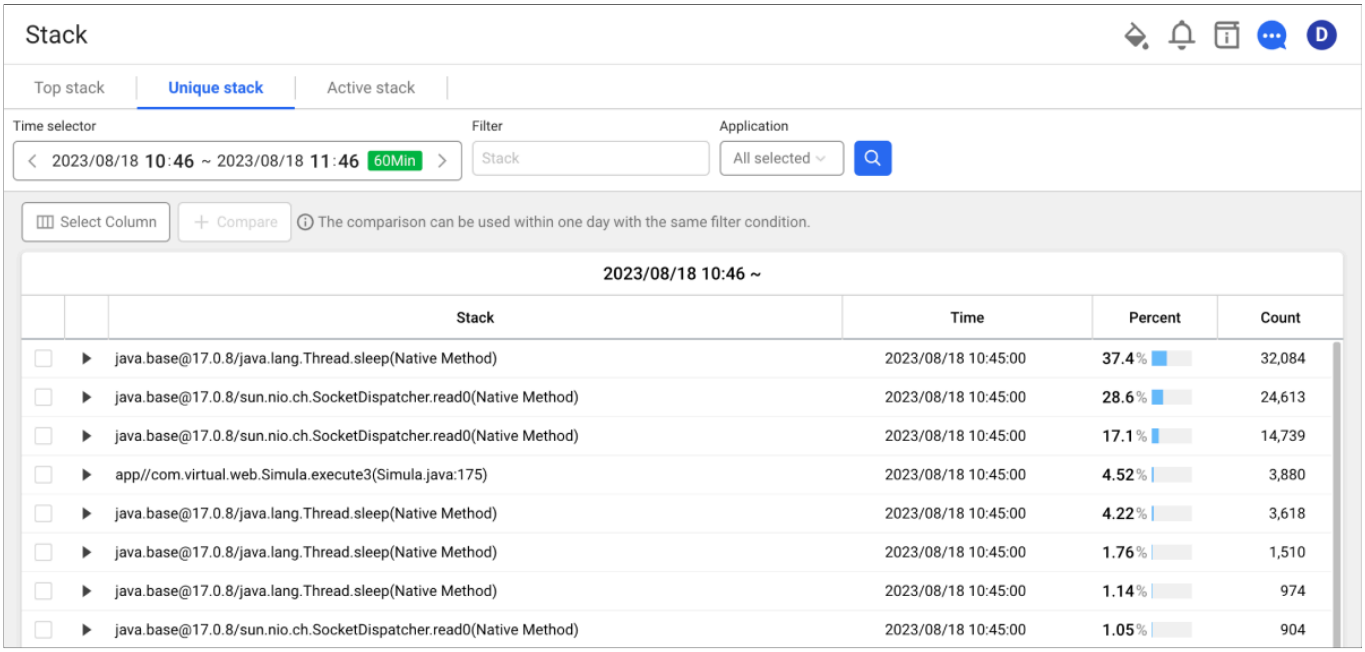

```
whatap.util.ThreadUtil.sleep
// The call rate of jdbc.Control.exec is 40.02%.
jdbc.Control.exec
// The call rate of jdbc.FakePreparedStatement.executeQuery is 68.06%.
```

The call rate of `whatap.util.ThreadUtil.sleep` ← `jdbc.Control.exec` ← `jdbc.FakePreparedStatement.executeQuery` does not indicate 40.02% * 68.06%. It is because the possibility of calling other modules exist in `jdbc.Control.exec` .

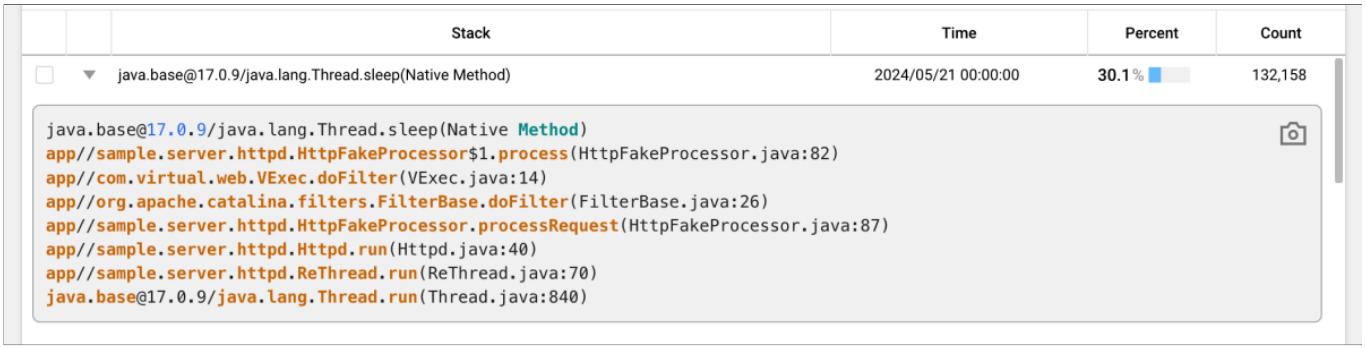
When determining the call rate through **Top stack**, do not multiply the call rates between steps to calculate the overall call relationship. Because the call rate of the **top stack** is the result of the call rates between steps of the data exposed on the stack trace, the total call rate derived with the call rates between steps can lead to a distorted result.

Unique stack

Unique stack provides statistical information about the cases where the set of methods executed based on the exact call information throughout all stack traces is the same.

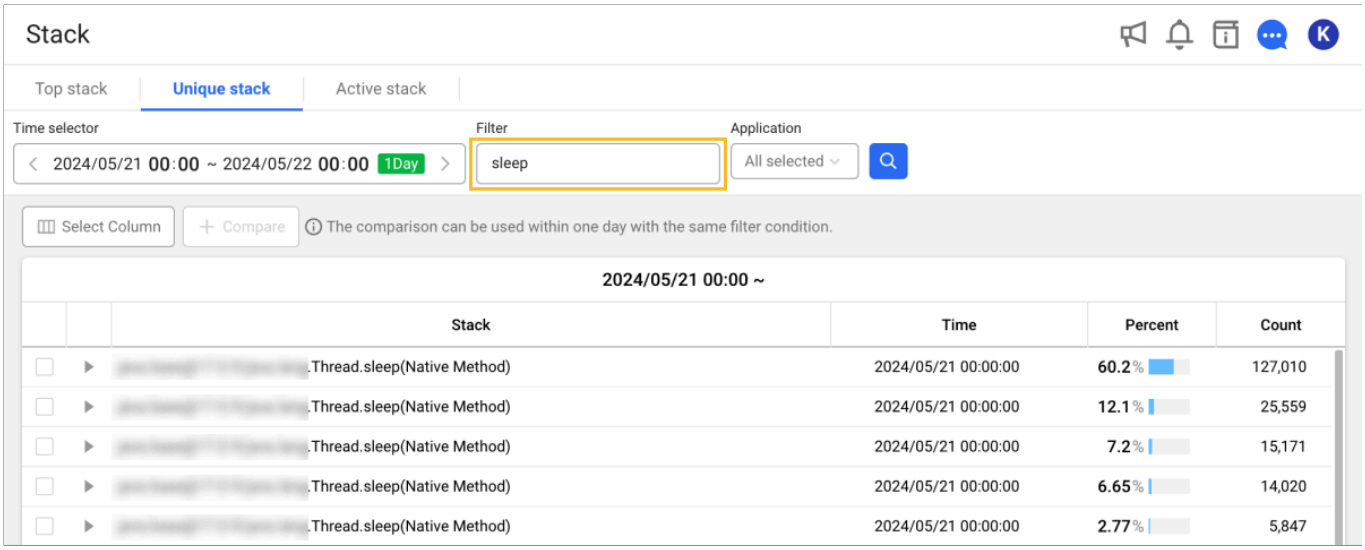


You can find information about the most frequently used stack. Even if the same stack is repeated multiple times, duplicates can be removed to identify the stack with the highest call frequency. For example, you can identify the stack traces with high backlog rates. You can also determine whether abnormal modules exist on the call path by reviewing detailed call steps.



Each stack displays the call path (**Stack**), occupied percentage (**Percent**), and total call count of the method. Displays (**Count**). If you select ▶ on the utmost left of each step, useful information for problem analysis is provided, such as method name, file name, and line number of the call stack.

Filtering

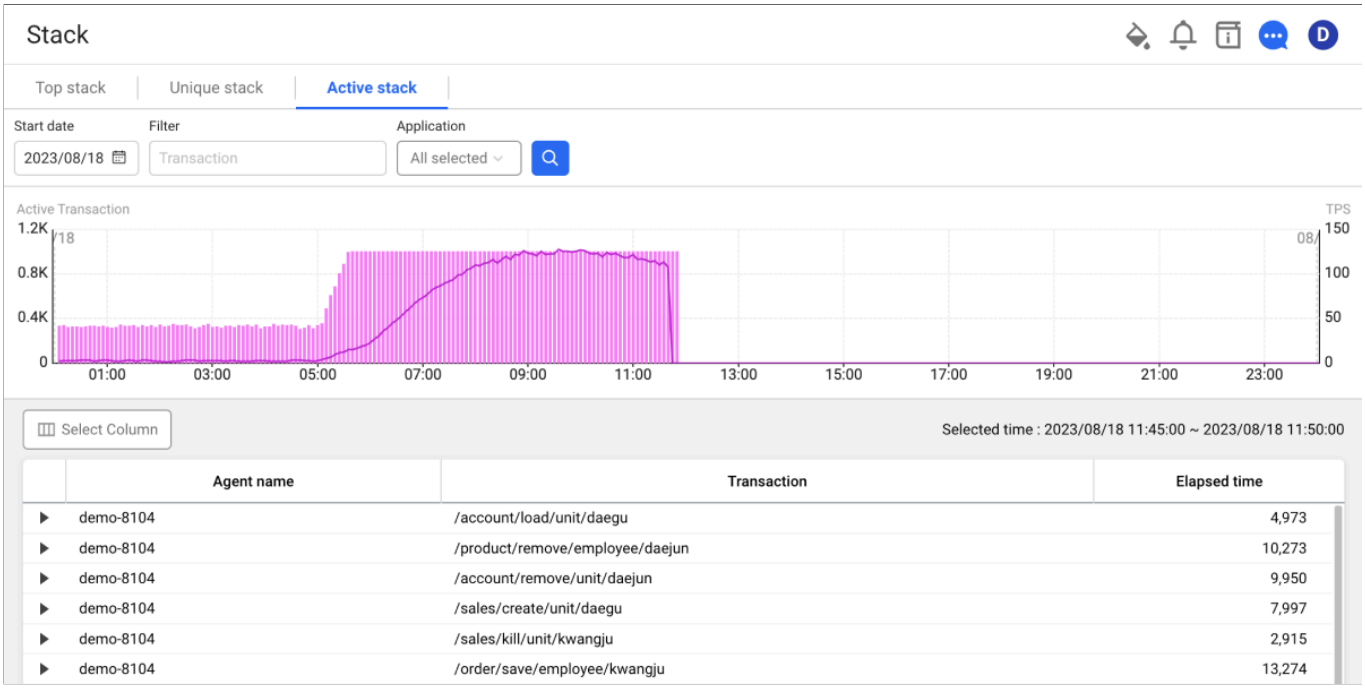


You can filter the search results by entering the method name in **Filter**. Enter the string and then select .

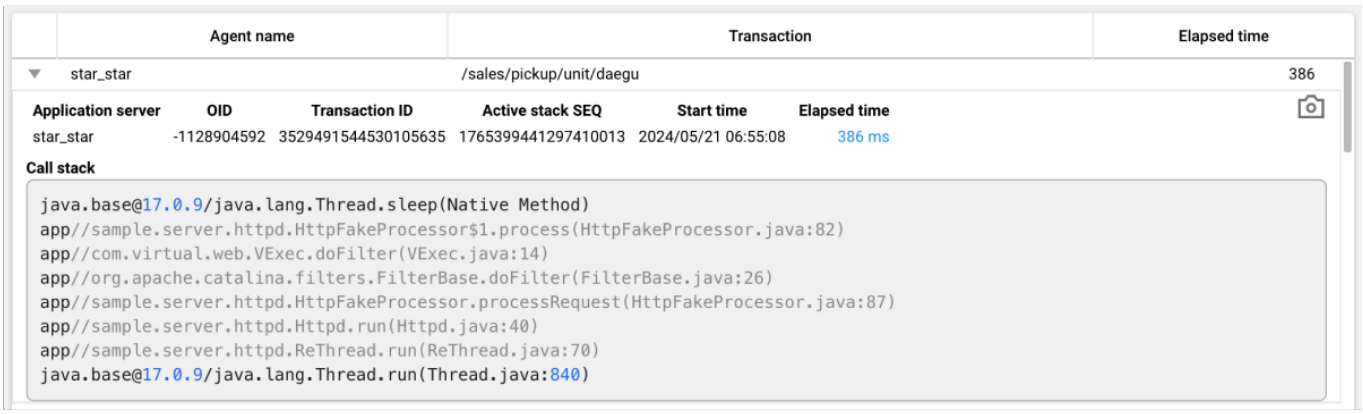
Active stack

Transactions in progress are called "active transactions." **The stacks regularly dumped by active transactions** are called **Active stack**.

In **Active stack**, you can see the collected **Active stack** in a time-series chart. The ratios for long-running methods and short-running but frequently executed methods allow you to understand the status of active transactions in real time.



The chart uses the 5-minute unit statistical data to display the number of **active transactions** on a bar graph and **TPS** on a line graph. If you click the bar graph for a specific time zone, you can see a list of **active transactions** for the time zone.



If you select ▸ on the utmost left of the transaction list, you can see the **Active stack** information of the transaction.

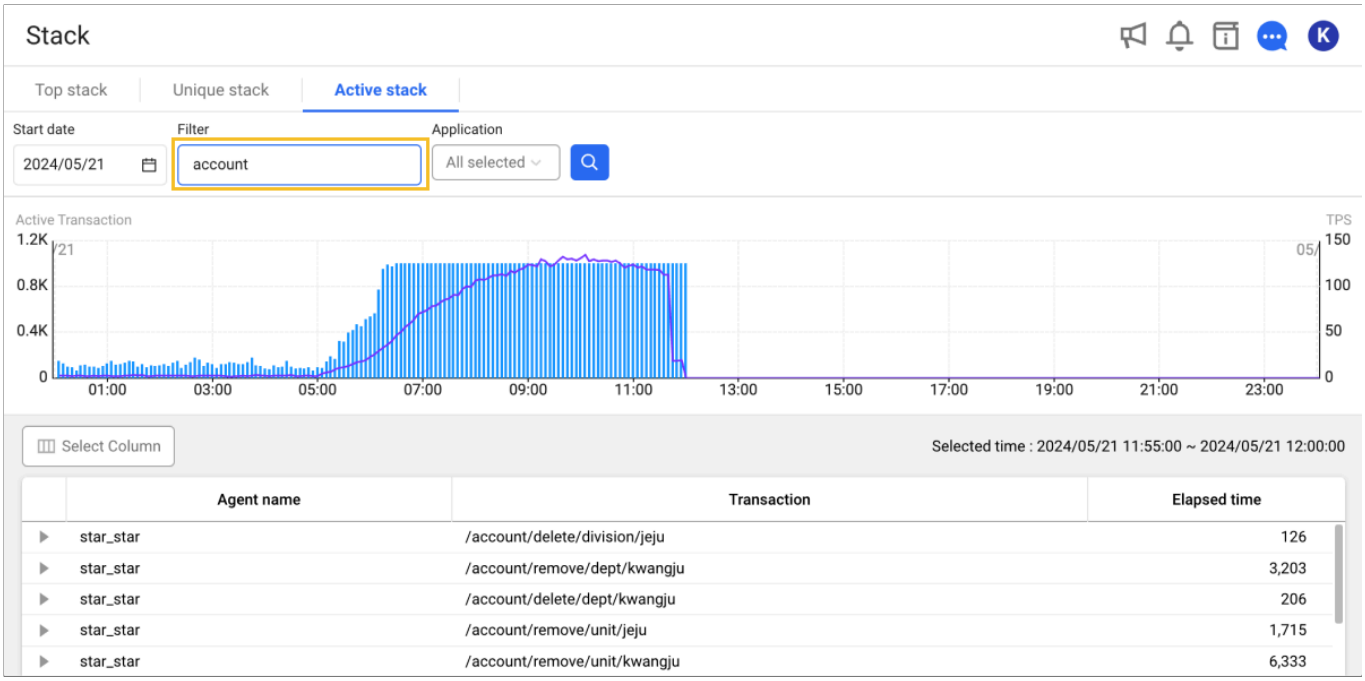
> ⓘ • For more information about collection of **active transactions**, see the following. For other agent options related to the collection, see the following.

- The WhaTap agent transmits **active stacks** dumped from **active transactions** to the server every 10 seconds (optional). For the agent option about the collection interval, see the following example.

  whatap.conf

```
active_stack_second=10
```

Filtering



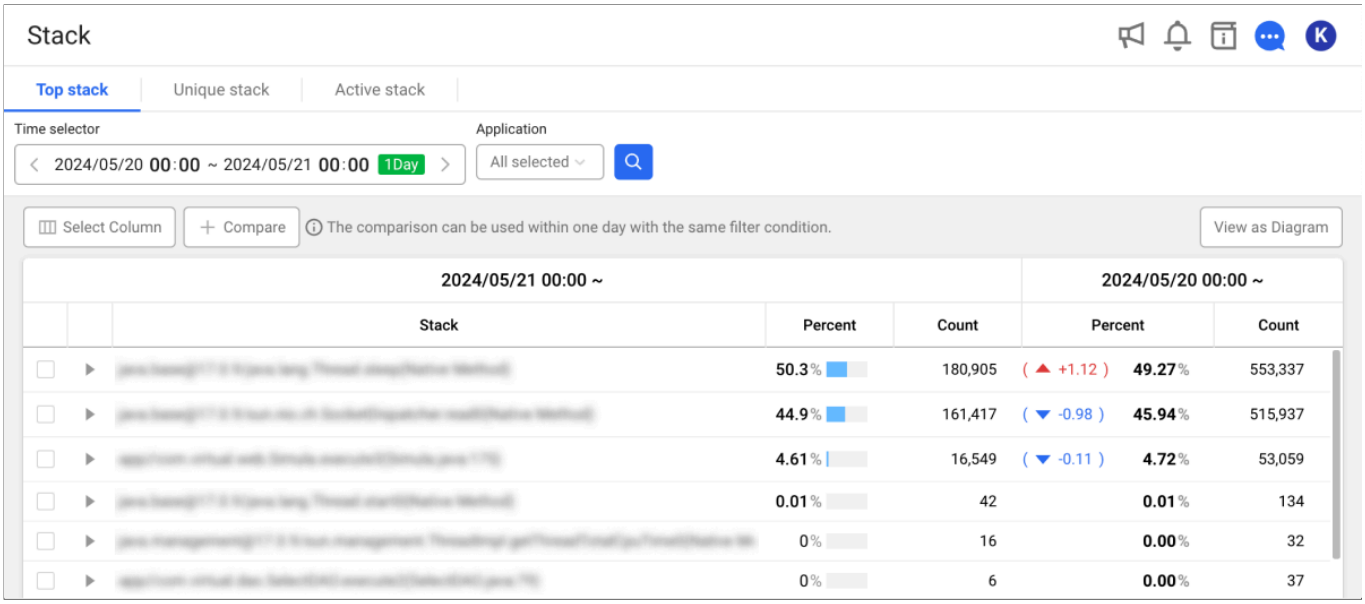
You can filter the search results by entering the transaction name in **Filter**. Enter the string and then select .

Comparing stacks

Top stack and **Unique stack** provide the history of rate changes and collection counts over time based on a specific period of time.

Set the time to view in **Time selector**, and then select **Compare**. You can compare stacks over time when viewing

the **within 1 day** period under the same filtering condition. This allows you to see the changes in the call stacks over time and see the performance trend. It can also be used as useful data for checking incident points and comparison before/after the performance improvement.



- **Percent**: You can see the rate changes of the selected stack during the inquiry period.
- **Count**: The number of collected stacks is proportional to the number of active transactions. For example, if the collection amount increased in a specific section, it can be seen that there was a service delay or a sudden increase in inflow.

> If **Compare** is not enabled, you may have exceeded the inquiry time range or there is no enough data for comparison. Set the inquiry time range in **Time selector** to enable the **Compare** button.

Cube

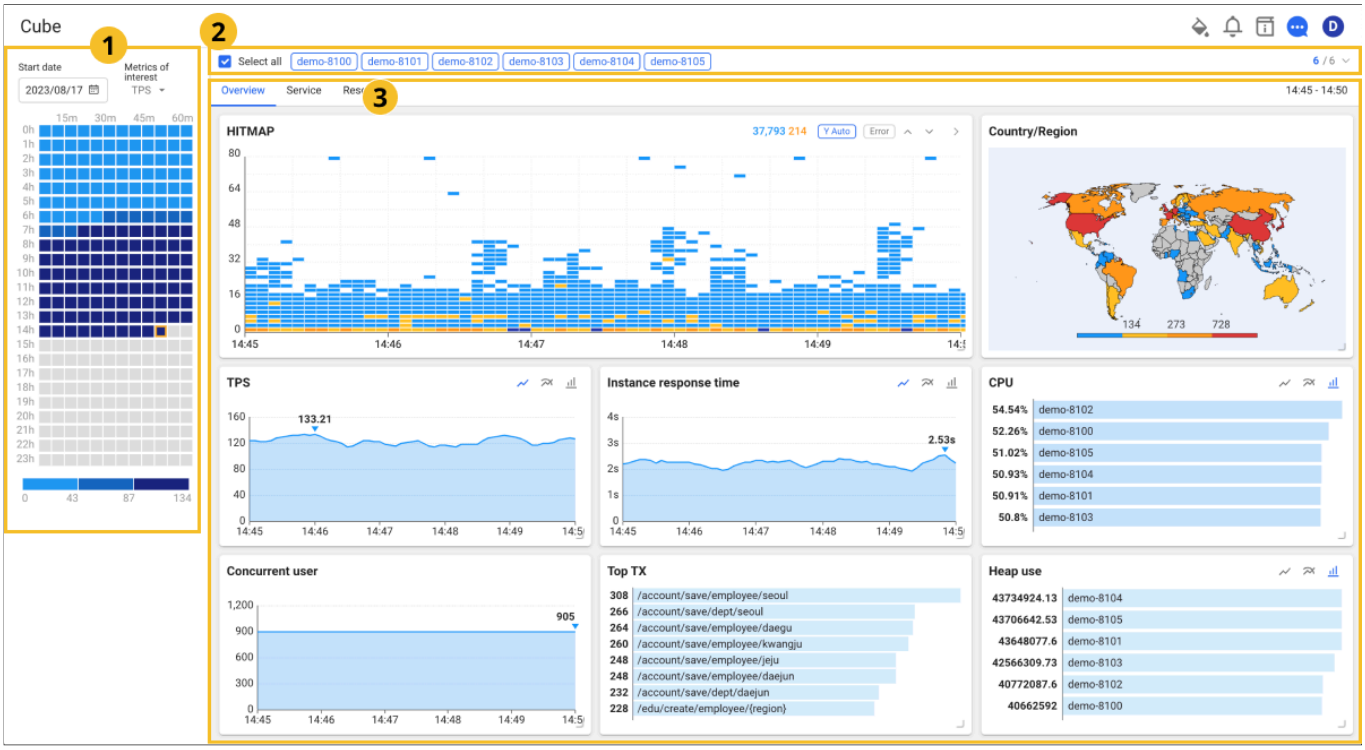
Home > Select Project > **Analysis** > **Cube**

It provides information for connections, hitmap, transactions, TOP transactions, TPS, response times, and resource usage for each country.

- See the application performance trends over time. You can quickly identify the changes in application performance by using statistical data at an interval of 5 minutes.
- You can see the geographic distribution of the traffic. Through the traffic data for applications, you can see the geographic distribution of traffic data and compare the performances across regions.
- You can see the usage of resources. You can see the usage of resources used by applications and analyze the causes of poor performance.

What is Cube?

We call the performance statistics generated every 5 minutes as **Cube**. Cube analysis uses each 5-minute performance data stored in cubes.



1 Cube selection panel

Select a cube based on the specific time in the cube selection panel in the left 1 area. Because the cubes are stored every 5 minutes, select a specific time.

- **Panel interest metric**
  - **TPS**
  - **Response time**
  - **TX error count**
  - **Active TX count**
  - **Concurrent user**
- **Cube color**

  The cubes are displayed in different colors depending on the number of the selected metrics.

  - Dark blue: high figure

- Light blue: low figure

2 Application selection

You can select data for a specific application through the application selection option in the top 2 area. If you select ⌄ on the right, you can see all the applications that can be selected as follows.



- **Select all**: All applications are viewed. By default, **Select all** and **Active application** are selected.
- **Active application**: Among the applications, the active applications are specified for inquiry.
- **View only selected**: Among the applications, only the specified applications are queried.
- **Search**: If there are many applications, you can search for them through the input field.
- Number/Number: Displays the number of applications specified/number of all applications.

3 Cube data

In the area of 3, the cube data can be checked through the charts for **Hitmap**, **Country/Region**, **TPS**, **Instance response time**, **CPU**, **Concurrent user**, **Top TX**, and **Heap use**.

- **Hitmap**

  It displays the distribution of transactions over the cube time. Detailed trace analysis is possible.

- **Country/Region**

  By mapping countries based on the client IP, it displays from which country the transactions are coming. The colors on the map display the relative amount of transactions.

  > ⓘ WhaTap basically stores the client-related information. For more information about the agent settings for user data collection, see the following.

- **TPS**

  Transaction Per Second (TPS) means the number of transactions processed per second.

- **Instance response time**

  Average response time for transactions

- **CPU**

  After sorting the project applications (agents) by the IP, it displays CPUs of the busy server with a bar chart.

- **Concurrent user**

  It displays the number of real-time browser users.

- **Top TX**

  It displays 5 transactions of high invocation. Transaction statistics provide more detailed data.

- **Heap use**

  Among the applications (instances), the instances are displayed in order of heap memory usage.

> ⓘ
> - If you select the icon on the upper right of the chart, merged line charts are provided.
> - If you select the icon on the upper right of the chart, individual line charts are provided.
> - If you select the icon on the upper right of the chart, the bar-type chart appears.

> ⓘ WhaTap uses the IP2Location LITE database for IP geolocation.

Hitmap Analysis

What is Transaction Tracing?

While the transaction performance means the summary metrics or attributes between the start and end of transaction, the transaction profile traces the steps during transactions. If any transaction is slow or erroneous, the execution records must be traced step by step to find the cause. This is called **transaction tracing**.

The types of steps collected for transaction performance tracing are as follows.

- **DB connection step** START-END

  It includes the performance for connection to RDB. Step information includes the name, response time, and error.

- **SQL Step** START-END

  It includes the performance for JDBC SQL. Step information includes connection data, SQL statements, and errors.

- **HTTP Call step** START-END

  It includes the performance for the outbound HTTP service call. Step information includes URL, host, port, response time, and errors.

- **Message step** ADD

  While collecting traces, the message steps are used to collect the records of all irregular sections.

- **SOCKET Step** ADD

  Step to represent the socket open.

- **METHOD Step** START-END

  The method response time is traced.

- **ACTIVE STACK Step** START-END

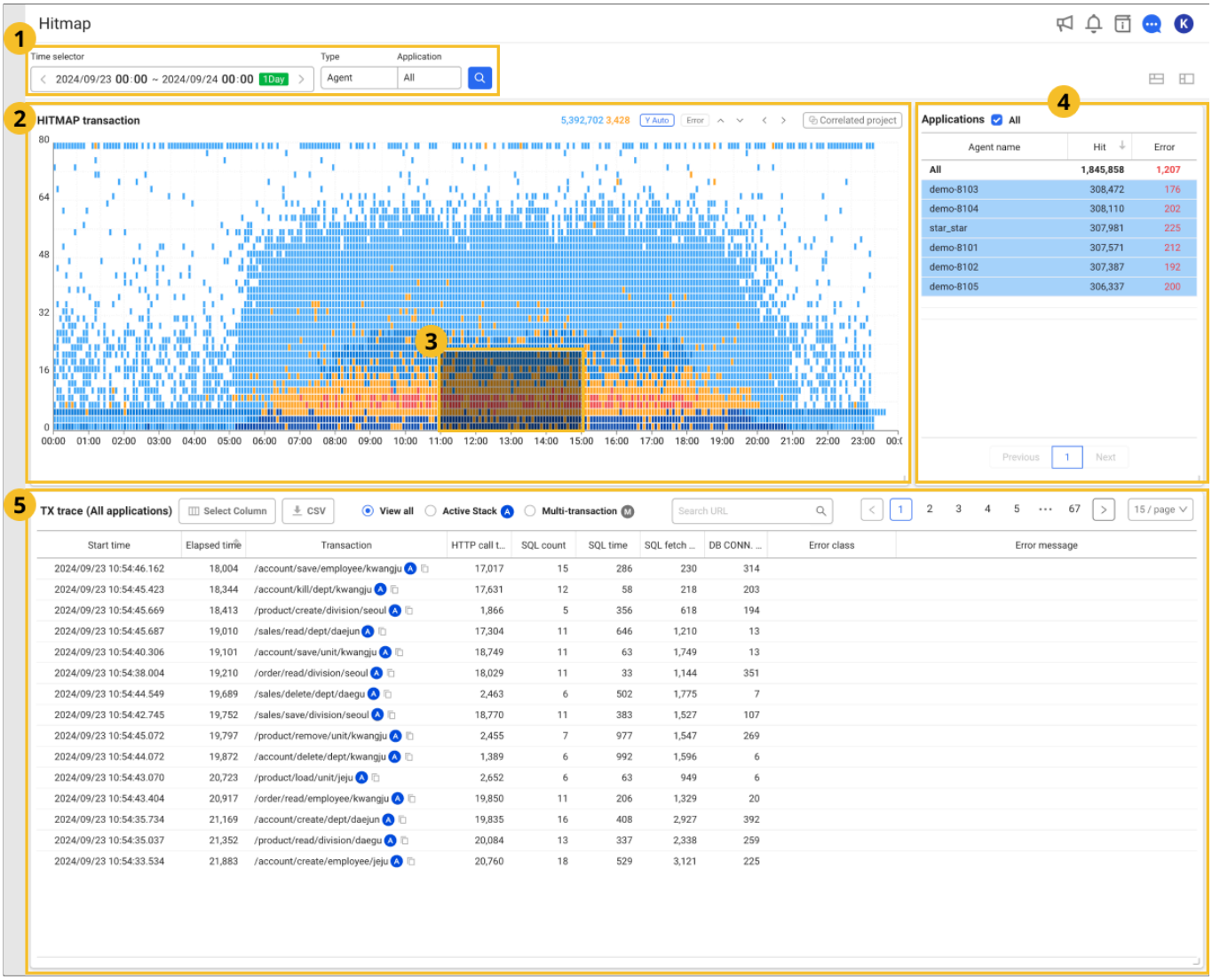
  It includes the active stack information. It is created and collected by a separate thread and it is added to the transaction trace.

If you add a database project through the **Correlated Project Management** menu, you can see the performance metrics of the linked project. For more detailed information about adding and analyzing a linked project, see [the following] (linked-projects).

In the WhaTap monitoring service, **Step** is used as the same meaning as **Span**.

Detailed analysis

Home > Select Project > **Analysis** > **Hitmap**

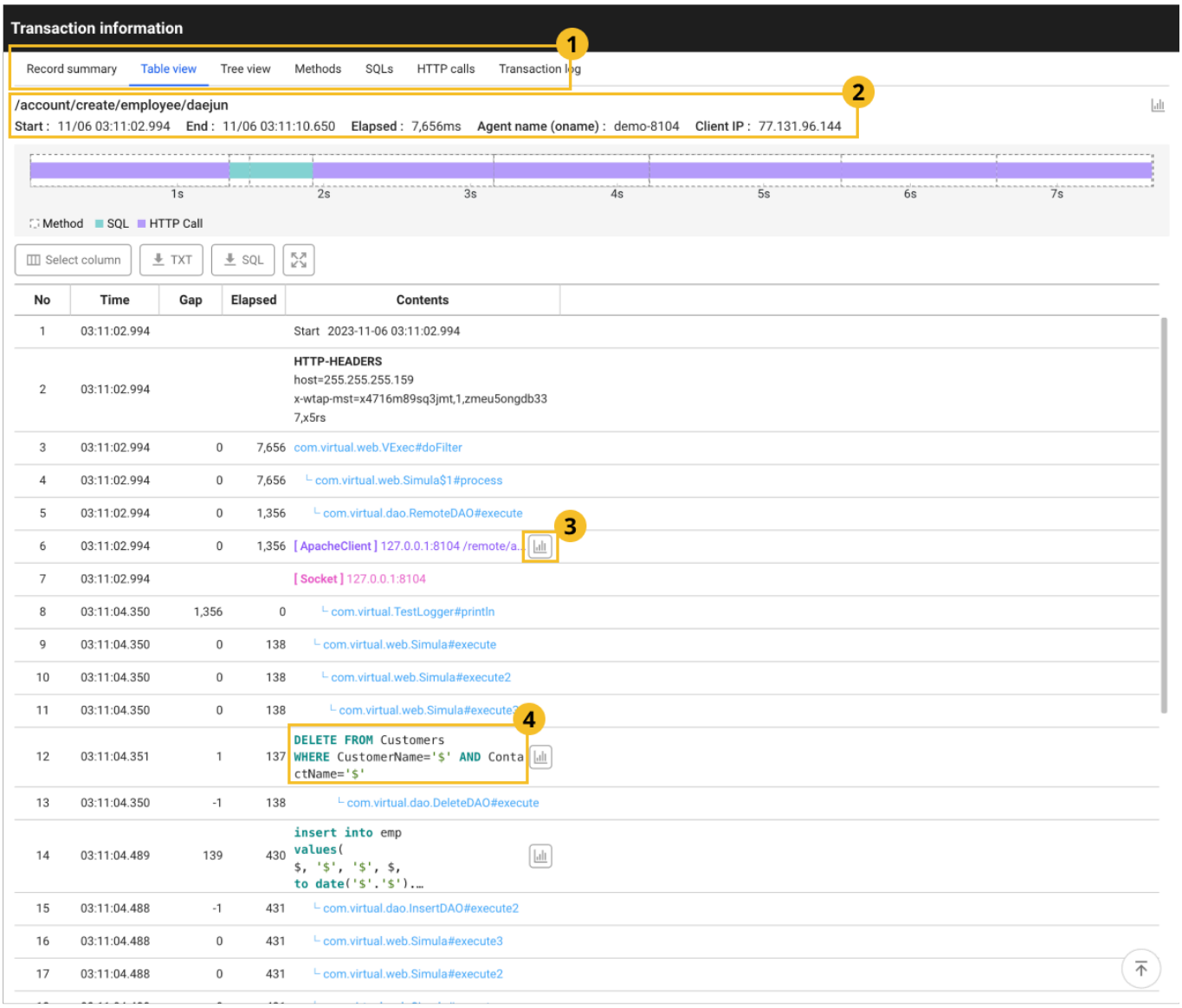


1. In 1 **Time selector**, select a time period to see the hitmap.
2. In 1 **Type** and **Application**, select the targets to view.
3. Select .
4. In 2 **HITMAP transaction**, drag a desired area as in 3.
5. If the list appears in 5 **TX trace**, select a transaction to analyze.

The **Transaction information** window appears where you can check properties such as client information for transaction performance analysis, transaction processing performance, and detailed execution history for each section.

- For more information about the **Hitmap** menu, see the following.
- For more information about the hitmap widgets on the application dashboard, see the following.

Guide to transaction information window



- In the 1 area, the step data can be seen in various formats. Select a desired type of tab.
- In the 2 area, you can see the basic information for transactions.

- 3 📊: The statistical data window for the corresponding URL or each execution section appears. Click a desired time on the graph chart in the statistical data window to go to **Statistics** or **Transaction Search**. You can view the statistical data based on the selected time.

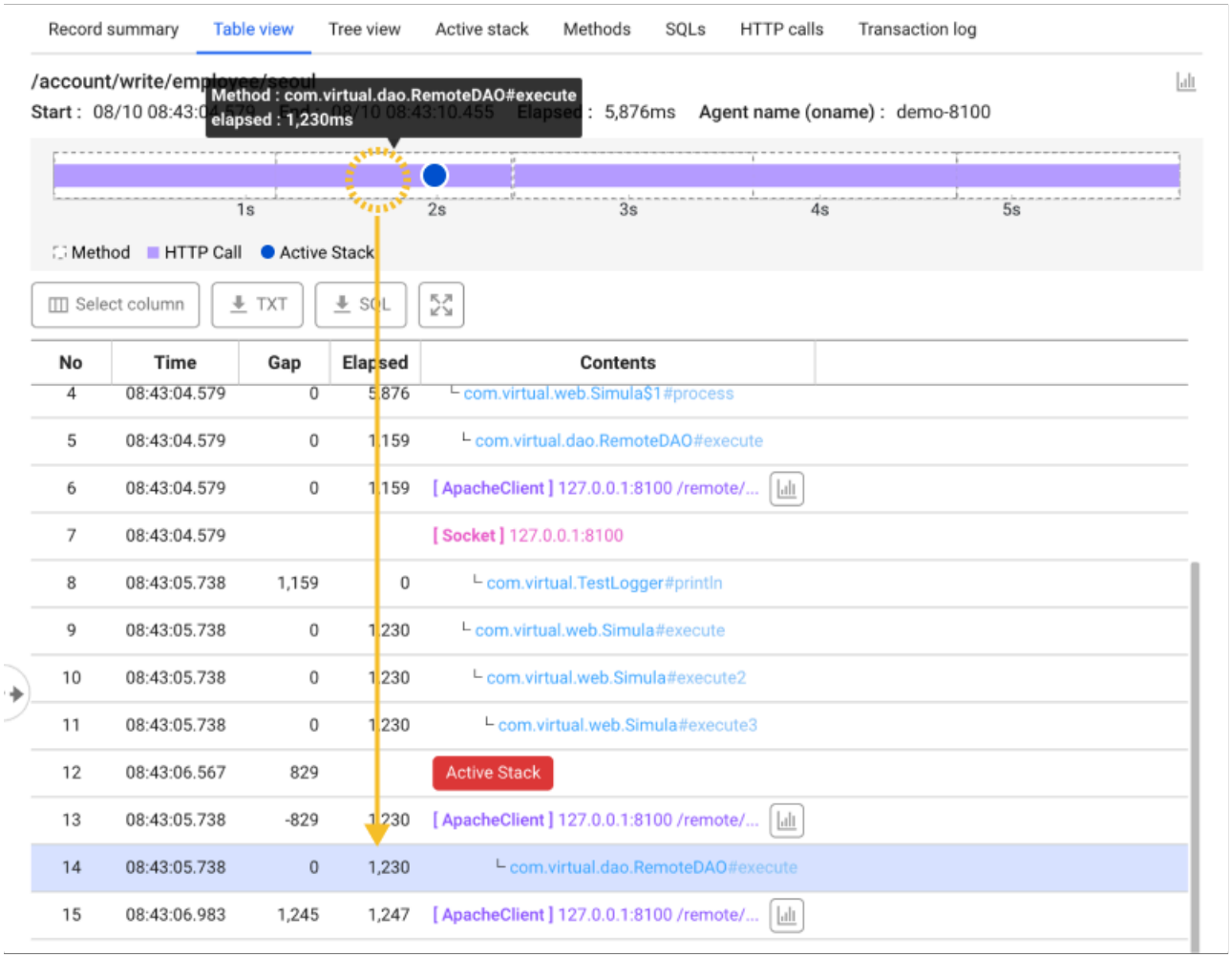
  > ⓘ
  > - For more information about **Statistics**, see the following.
  > - For more information about the **Transaction Search** menu, see the following.

- If you select 4 SQL step, the SQL variables and HTTP queries can be viewed. For more information, see the following.

Table view

In the **Table view** tab, you can see the transaction execution steps in chronological order.

- You can classify transactions by each execution section and see them in a diagram. Out of all elapsed times, you can quickly find the times spent for each section and the section that took the longest.
- If you select each section in the diagram, the corresponding table list appears where the step is located.



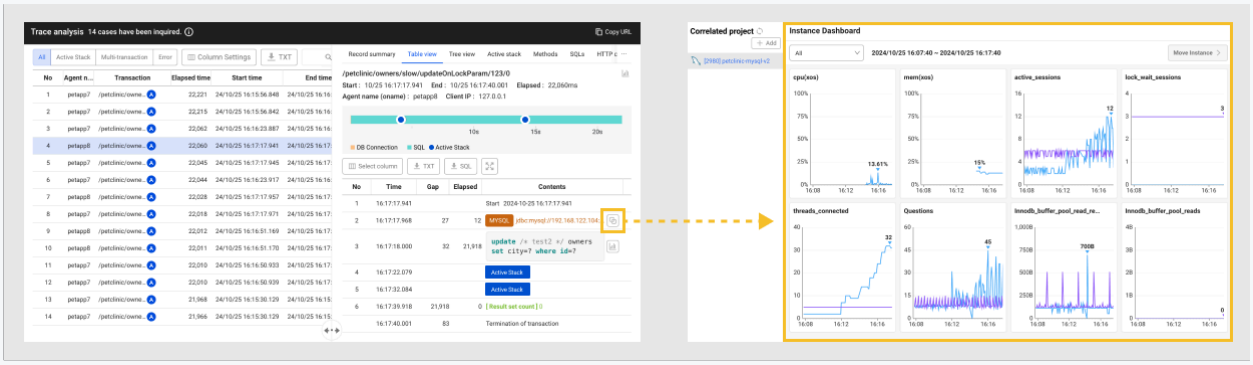
- The area where the blue circle (🔵) is located, is the moment when the active stack has been collected. If you select the blue circle, the table list appears where the **Active Stack** button is located. If you click **Active Stack**, you can check the step performed during the section.

  > ⓘ
  > - In the transaction list, you can see them from the items marked with the Ⓐ icon.
  > - For more information about active stacks, see the following.

- **Select column**: You can add or hide the memory accumulation and CPU accumulation data as columns in the table list. The columns are as follows:
  - **No**: Step occurrence sequence

  - **Time**: Start time of each step
  - **Gap**: The waited time between the start time of the previous step and the current step may have a difference, while there may be a gap from the elapsed time if a delay happens due to external factors.
  - **Elapsed**: Total elapsed time from the start to the end of each method
  - **Contents**: Execution details of the step
- **TXT**: You can download basic transaction data and execution data for each section in a text file.
- **SQL**: The basic transaction information and SQL execution data can be downloaded as a text file.
  - **Download with substitute value**: It downloads the SQL statistics data after replacing the bind variable value with its original one. If a security key (`paramkey`) is set, the security key input window appears. You must enter the security key for downloading. For more information about the agent configuration related to the security key, see the following.
  - **Download without substitute value**: It downloads SQL statistics data without replacing the bind variable value.
- : You can see only the table list, with hiding the transaction summary and diagrams. If you select , the hidden information reappears. This feature is useful when you have a long list of tables.
- If you select the SQL step, the **SQL** window appears to display parameters. For more information, see the following.

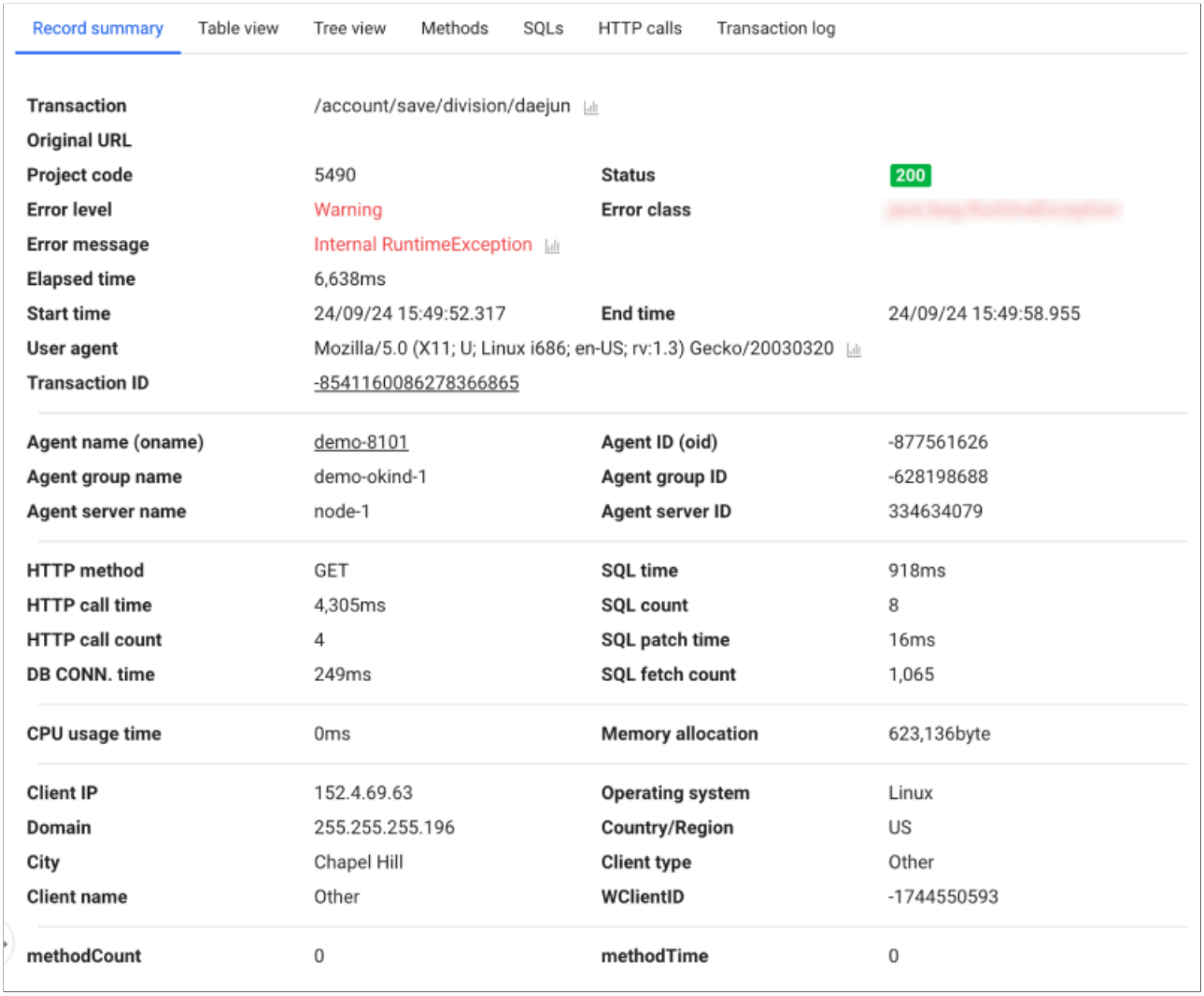
If you add a database project through the **Correlated Project Management** menu, you can see the performance metrics of the project linked in the **DB connection step**.



For more detailed information about adding and analyzing a linked project, see [the following] (linked-projects).

Record summary

To see the basic information for transactions, select the **Record summary** tab. You can see the basic information of the transaction, agent on which the transaction was performed, memory allocation usage, and client information.



Classification	Properties	Description
Transaction	Transaction	API or URL call executed by the application, and the path of the transaction. It converts to a specific format using regular expressions.

Classification	Properties	Description
	Original URL	It is the URL path that was actually called. A concrete request path that does not include variables. If there is an original URL, the original URL appears. If there is no original URL, it is not displayed.
	Project code	Identification code of the project registered in the WhaTap monitoring service.
	Status	HTTP response status code for the transaction.
	Error level	Level of the error that occurred in the transaction. It is displayed as Warning or Critical.
	Error class	Class of the error that occurred in the transaction.
	Error message	Message of the error that occurred in the transaction.
	Elapsed time	Transaction execution time.
	Start time	Transaction start time.
	End time	Transaction end time.
	User agent	Browser used to extract client-related data.
	Referer	Page address or incoming path used by the browser of the client.
	Transaction ID	It is a unique ID of the transaction.
	Multi transaction ID	It is a unique ID of the multi-transaction. It appears when executed in parallel with transactions of other applications.
Agent	Agent name (oname)	Agent name

Classification	Properties	Description
	Agent ID (oid)	It is a unique ID of the agent.
	Agent group name	Agent group name set with the `okind` option.
	Agent group ID	It is a unique ID of the agent group.
	Agent server name	Agent server name set with the `ondoe` option.
	Agent server ID	It is a unique ID of the agent server.
Step	HTTP method	HTTP method. (GET, POST, PUT, HEAD, etc.)
	HTTP call time	Outbound HTTP call time.
	HTTP call count	Number of outbound HTTP calls.
	DB CONN. time	Time taken to connect to the database.
	SQL time	SQL execution time.
	SQL count	SQL execution count.
	SQL patch time	Time taken to search for the record. Nesting may occur or other logic may be included.
	SQL fetch count	Number of the records retrieved from SQL.
Resource	CPU usage	CPU time used for processing transactions.

Classification	Properties	Description
	time	
	Memory allocation	Amount of memory used by the transaction.
Containers	Pod name	It is the name of the Kubernetes Pod that processed the transaction.
	Container Id	It is a unique identifier for a specific container running within a Pod.
Client	Client IP	Client IP address.
	OS	OS environment in which the browser runs.
	Domain	Internet address assigned to the IP address that the client connected to.
	Country/ Region	Client's country.
	City	City in the client's country.
	Client type	Browser type used by the client.
	Client name	Device name used by the client.
	WClientID	This is a unique ID of the client.
Method	methodCount	Number of methods called by transactions.
	methodTime	Time taken for method execution.

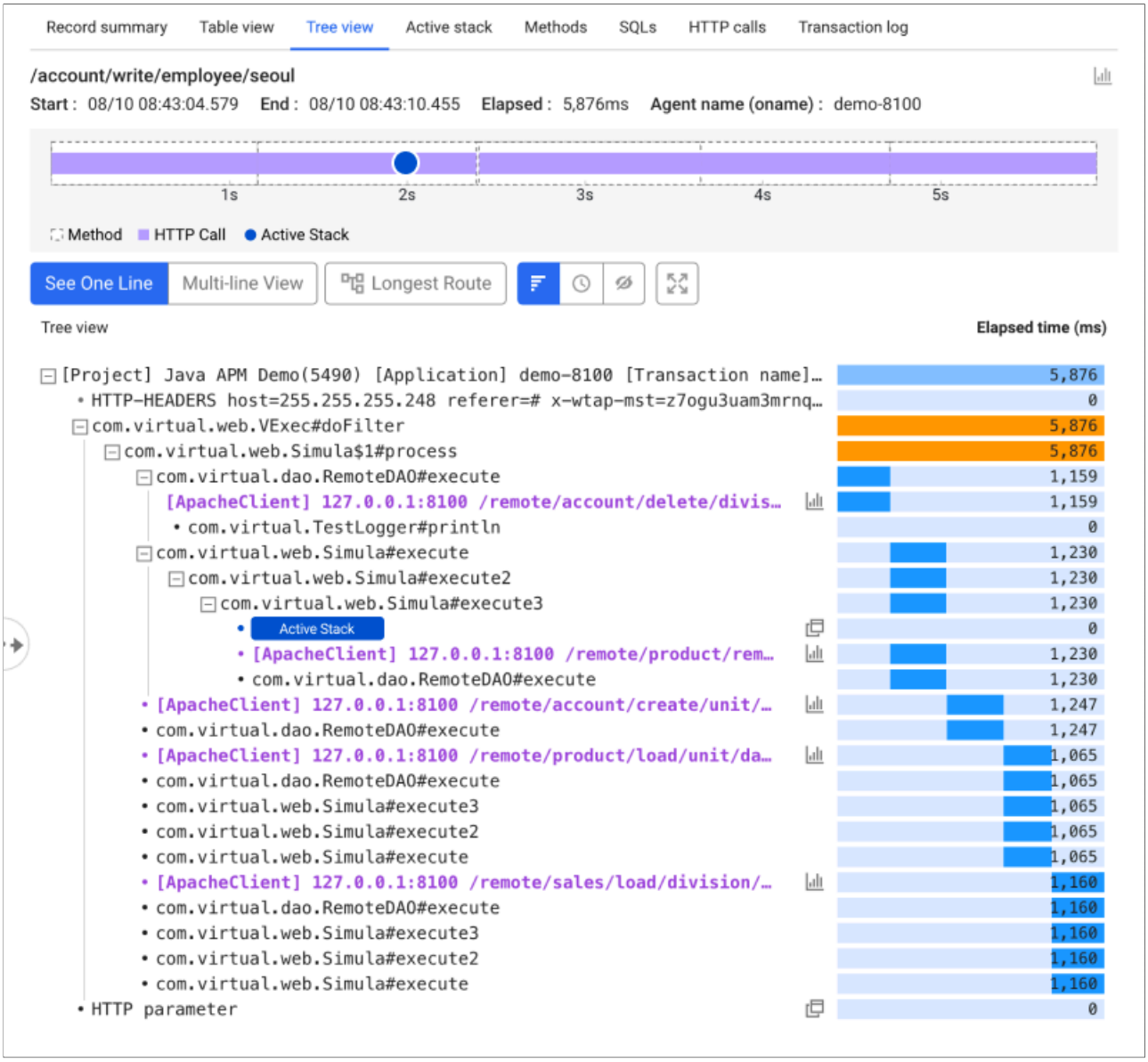
- The collected data may differ depending on the application type, settings, and step type. **Container** entries appear only when the application is running in the container environment.
- Error related items are displayed when an error occurs from the transaction.
- Some transaction properties provide statistical features. If you select the button, a window

appears where you can see the status of transactions occurred during the query time through a time series chart. If you select a bar graph in the chart, you can go to the **Analysis** > **Transaction Search** menu to view details.

- WhaTap basically stores the client-related information. For more information about the agent settings for user data collection, see the following.

Tree view

To see the transaction execution steps in a tree format, select the **Tree view** tab. You can see the details of each transaction and its trace details, trace start time, step time, and call relationship. If you select each section of the diagram, the tree view of the step appears.

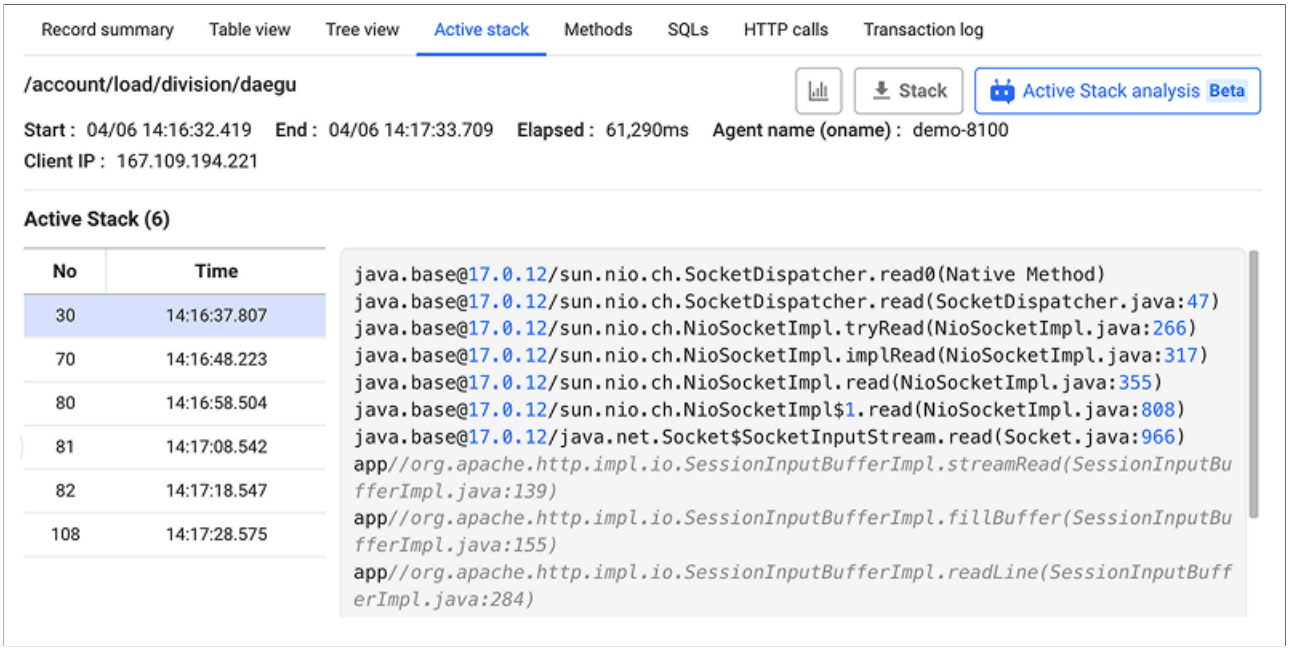


- **Single-line View**: You can arrange the tree format at intervals by displaying the texts displayed in the execution data for each section in a single line.
- **Multi-line View**: All the texts displayed in the execution data for each section, appear by wrapping words.
- **Longest Path**: The longest route can appear.
- **Show Time Bar**: The elapsed times are displayed in a bar chart.
- **Show Time**: The timestamps, gaps, and elapsed times for each segment are displayed in text.

  - 8 seconds or more: Excessive delay state is distinguished in red.
  - Between 3 and 8 seconds: Delay state is distinguished in orange.
  - Less than 3 seconds: Normal state is distinguished in blue.
- **Hide Time**: The time is hidden.
- : You can see only the tree view that hides transaction summary and diagrams. If you select , the hidden information reappears. This feature is useful when the tree view list is long.
- : A window appears where you can view SQL variables and HTTP queries. For more information, see the following.

> In case of start and step times, the time differences caused by the transaction call environment are corrected on the screen through traces and mappings in the parent transactions. Accordingly, there may be differences from the actual time data.

Active Stack



Select the **Active stack** tab to view the list of active stacks collected from the transaction.

- Select a collection point (No, Time) from the **Active Stack** list on the left to display the stack information for that point on the right.
- **Stack**: You can download the active stack details as a TXT file.

> - You can check the active stack from items marked with the A icon in the transaction list.
> - For more information about active stacks, see the following.

AI Active Stack Analysis

Beta

Click the **Active Stack Analysis** button to automatically analyze collected active stacks with AI. It identifies performance issues based on the stack dumps of in-progress transactions and suggests improvements.

The analysis results consist of the following 4 sections:

- **Summary**: Summarizes the number of analyzed stacks and the overall execution flow.
- **Common Patterns**: Displays repeatedly observed execution patterns, including pattern name, occurrence count, and description.
- **Issues Found**: Displays identified performance issues with severity levels (**High**, **Medium**, **Low**) along with the stack information used for the assessment.

Severity	Criteria
High	Deadlock, complete blocking, same blocking repeated 4+ times
Medium	I/O wait, 2-3 lock contentions, slow external calls
Low	Single-occurrence wait, minor optimization opportunities

- **Recommendations**: Suggests specific improvement actions by priority.

Each analysis item displays the collected stack number (`No.N`). Click the stack number to view the original stack.

> - The analysis results are generated by AI and may differ from actual conditions. Use them as

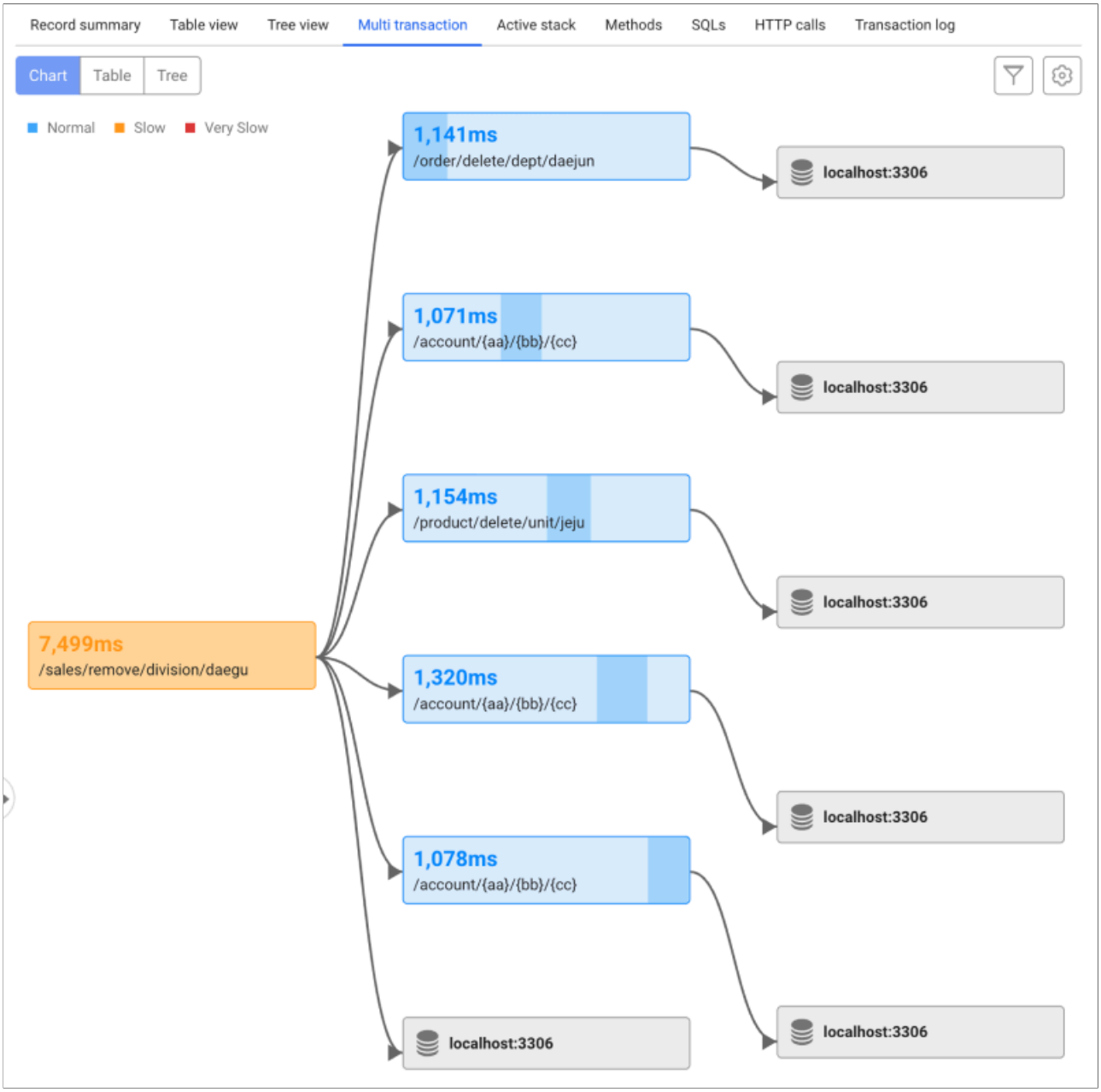
reference material.

- **Member** role or above is required.

Multi-transaction

Multi-transaction indicates the transaction involving other agents or projects. In the **Multi transaction** tab, you can see the call relationships between applications registered in the WhaTap monitoring service.

To trace multi-transactions, set the `mtrace_enabled` option to `true` in **Management** > **Agent CONF.**. For more information about the agent configuration, see the following.



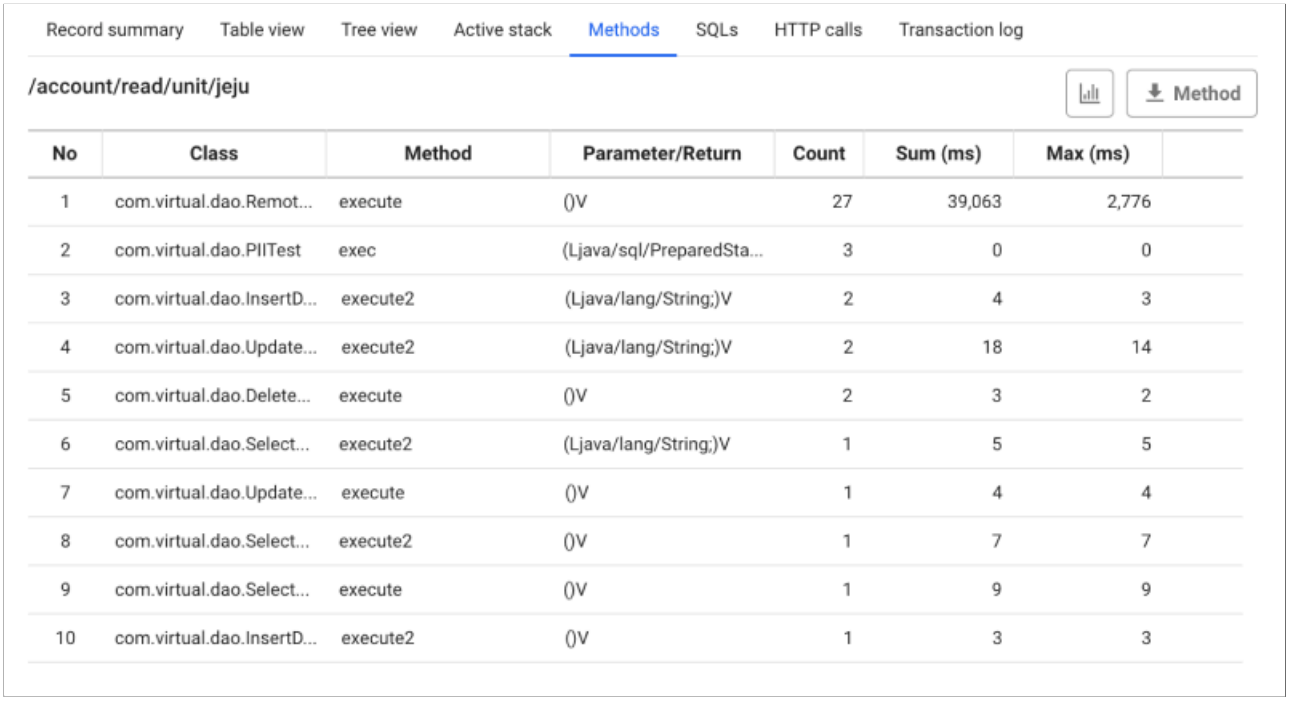
- **Chart**: Provides the call relationships for each transaction in a flowchart. Select a transaction node to see the trace analysis for the transaction node. In **Chart** mode, you can use the mouse to go to a desired position or zoom in or out by scrolling.
  - **Chart view settings**: You can display or hide the elements on the chart.
- **Table**: In the form of table, you can check the data for each transaction included in the multi-transaction. You

can select the column selection icon to edit the table header columns. You can see the trace analysis data by selecting each transaction item.

- **Tree**: You can see the call relationships between transactions in a tree format. The related additional functions are the same as those of the **Tree view** tab.
- **Project selection**: You can select or deselect projects to be displayed on the chart.

> - You can see them in the transaction list marked with the M icon.
> - For more information about the multi-transaction, see the following.

Methods



No	Class	Method	Parameter/Return	Count	Sum (ms)	Max (ms)
1	com.virtual.dao.Remot...	execute	()V	27	39,063	2,776
2	com.virtual.dao.PIITest	exec	(Ljava/sql/PreparedSta...	3	0	0
3	com.virtual.dao.InsertD...	execute2	(Ljava/lang/String;)V	2	4	3
4	com.virtual.dao.Update...	execute2	(Ljava/lang/String;)V	2	18	14
5	com.virtual.dao.Delete...	execute	()V	2	3	2
6	com.virtual.dao.Select...	execute2	(Ljava/lang/String;)V	1	5	5
7	com.virtual.dao.Update...	execute	()V	1	4	4
8	com.virtual.dao.Select...	execute2	()V	1	7	7
9	com.virtual.dao.Select...	execute	()V	1	9	9
10	com.virtual.dao.InsertD...	execute2	()V	1	3	3

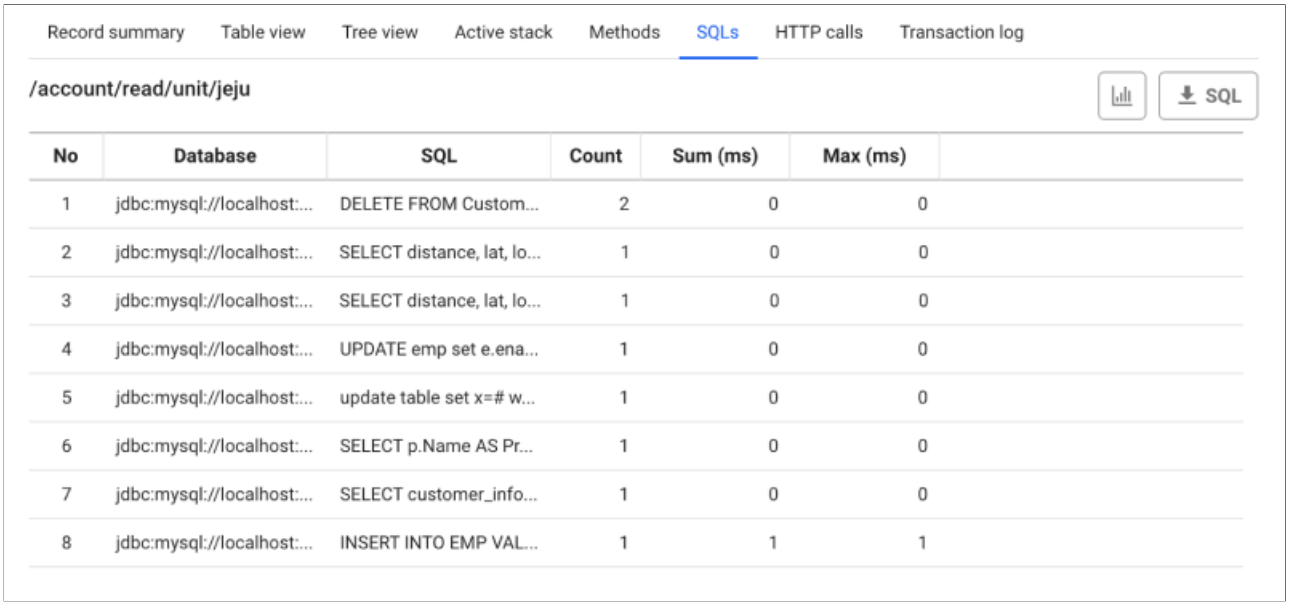
To see only the method data, select the **Methods** tab. It displays the method name and step time for which tracing has been set on the agent. When unnecessary logics are repeatedly executed, it can be used as analysis data to improve the method logic by finding the case where the execution takes longer than the low count.

- : You can see the **Transaction statistics** information. If a graph on the chart is selected, the **Analysis** > **Transaction Search** menu appears. You can see the list of transactions for the selected time zone.

- **Method**: You can download the method-related details as a TXT file.

> For more information about the agent configuration options on methods, see the following. Method tracing support may differ depending on the product in use.

SQLs

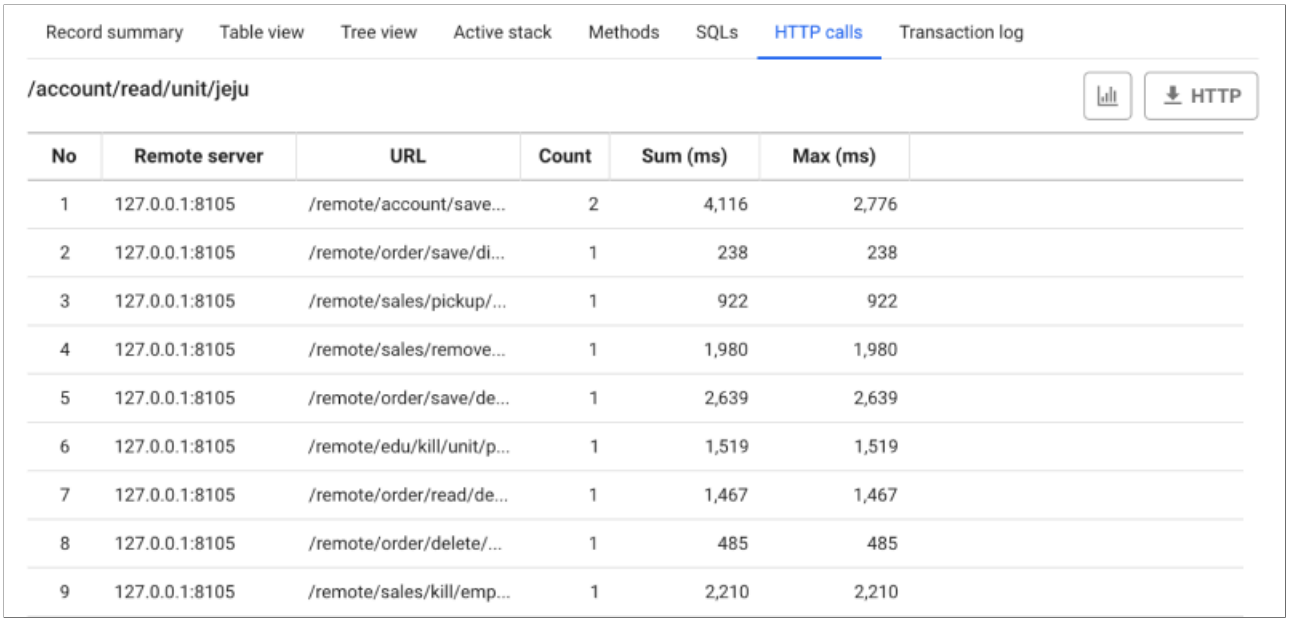


No	Database	SQL	Count	Sum (ms)	Max (ms)
1	jdbc:mysql://localhost:...	DELETE FROM Custom...	2	0	0
2	jdbc:mysql://localhost:...	SELECT distance, lat, lo...	1	0	0
3	jdbc:mysql://localhost:...	SELECT distance, lat, lo...	1	0	0
4	jdbc:mysql://localhost:...	UPDATE emp set e.ena...	1	0	0
5	jdbc:mysql://localhost:...	update table set x=# w...	1	0	0
6	jdbc:mysql://localhost:...	SELECT p.Name AS Pr...	1	0	0
7	jdbc:mysql://localhost:...	SELECT customer_info...	1	0	0
8	jdbc:mysql://localhost:...	INSERT INTO EMP VAL...	1	1	1

To check the SQL statement information, select the **SQLs** tab. In case of unnecessarily repetitive execution, it can be used as analytical data to improve query performance by finding the case where the execution time takes longer compared to the low count.

- : You can see the **Transaction statistics** information. If a graph on the chart is selected, the **Analysis** > **Transaction Search** menu appears. You can see the list of transactions for the selected time zone.
- **SQL**: You can download the SQL-related details as a TXT file.

HTTP calls



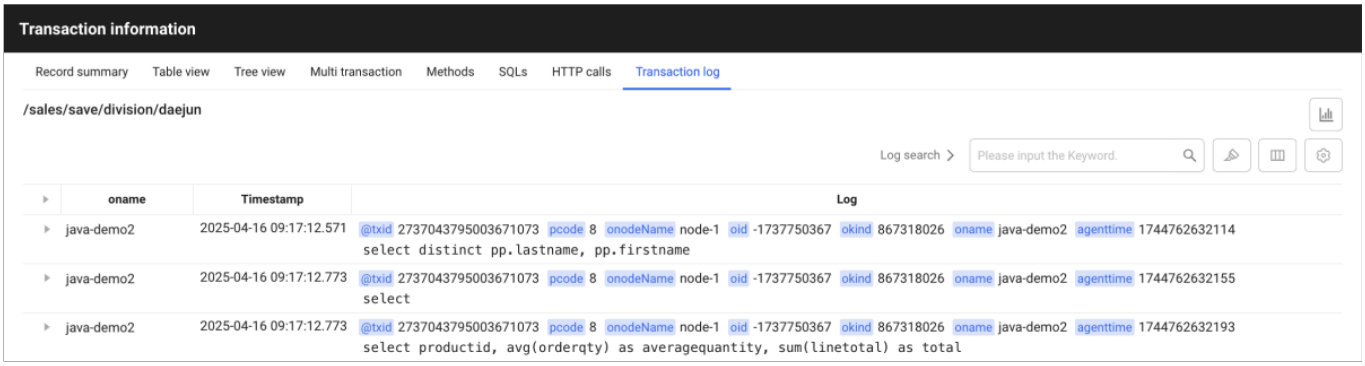
No	Remote server	URL	Count	Sum (ms)	Max (ms)
1	127.0.0.1:8105	/remote/account/save...	2	4,116	2,776
2	127.0.0.1:8105	/remote/order/save/di...	1	238	238
3	127.0.0.1:8105	/remote/sales/pickup/...	1	922	922
4	127.0.0.1:8105	/remote/sales/remove...	1	1,980	1,980
5	127.0.0.1:8105	/remote/order/save/de...	1	2,639	2,639
6	127.0.0.1:8105	/remote/edu/kill/unit/p...	1	1,519	1,519
7	127.0.0.1:8105	/remote/order/read/de...	1	1,467	1,467
8	127.0.0.1:8105	/remote/order/delete/...	1	485	485
9	127.0.0.1:8105	/remote/sales/kill/emp...	1	2,210	2,210

To see the count, time sum, average time, and such for HTTP calls, select the **HTTP calls** tab. If unnecessary external calls are repeatedly executed, it can be used as analytical information to determine whether the transaction delays are external or internal by finding the cases where the execution time takes longer compared to the low count.

- : You can see the **Transaction statistics** information. If a graph on the chart is selected, the **Analysis** > **Transaction Search** menu appears. You can see the list of transactions for the selected time zone.
- **HTTP**: You can download the HTTP call-related details as a TXT file.

Transaction log

To see the logs of the transaction, select the **Transaction log** tab.

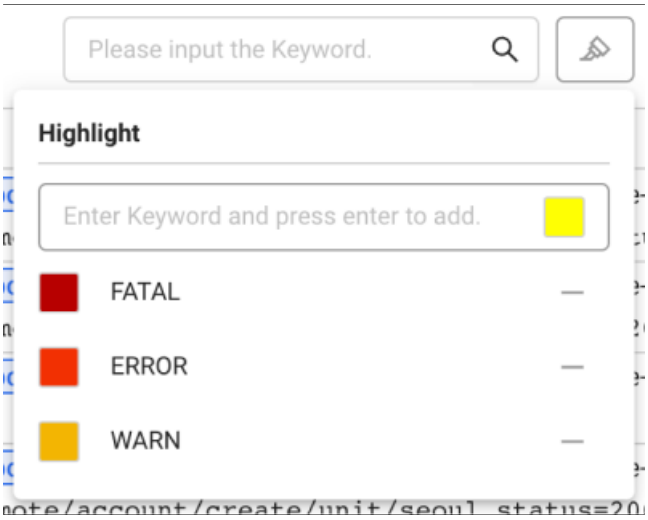


- : You can see the **Transaction statistics** information. If a graph on the chart is selected, the **Analysis** > **Transaction Search** menu appears. You can see the list of transactions for the selected time zone.
- **Log Search**: You can go to Log Search to easily find the desired log data.
- **Keyword search**: Enter the desired text to search in the keyword search field, and then press Enter or select the button. It highlights the keywords that match the entered text.
- **Keyword highlight**: If you set the keyword and color, the texts that match the keyword are automatically highlighted in the log list. For more information, see the following.
- **Column settings**: You can add columns or change the order of columns displayed in the log list table. For more information, see the following.
- **Log view detail setting**: You can configure the items to be displayed in the log list. For more information, see the following.

> - To enable the **Transaction log** tab, apply the agent setting options related to the logs. For more information about the log settings, see the following.
> - The **Transaction log** tab can be accessed by only the members with the **View log** role. For more information about the member roles, see the following.

Setting the keyword highlight

1. To add keywords to highlight, select .



2. Enter a desired keyword.
3. Select the color icon and then select a desired color.
4. Press Enter. The added keywords are added in the list.

Deleting the keyword highlight

To delete the items unused from the highlight list, select —.

Displaying or hiding the table columns

1. Select ▥ on the upper right of the table.
2. If the **Column settings** window appears, select or unselect the desired item in the list on the left.

   You can also hide columns by selecting the ✕ button in the list on the right.
3. After all settings are finished, select **Confirm**. The set content is applied to the table.

Reordering the table columns

1. Select ▥ on the upper right of the table.
2. When the **Column settings** window appears, change the order by dragging the desired items from the list on the right.

3. After all settings are finished, select **Confirm**. The set content is applied to the table.

Setting to display the table content

1. Select ⚙ on the upper right of the table.
2. When the **Log view detail setting** window appears, set the desired options.
   - Select **content** or **Tag** to determine whether or not to display the log content.
   - **Tag management**: Tags are displayed in the log in the order of addition. The order of tags can be changed by dragging. Disabled tags do not appear in the logs.
3. After all settings are finished, click ×. The set content is applied to the table.

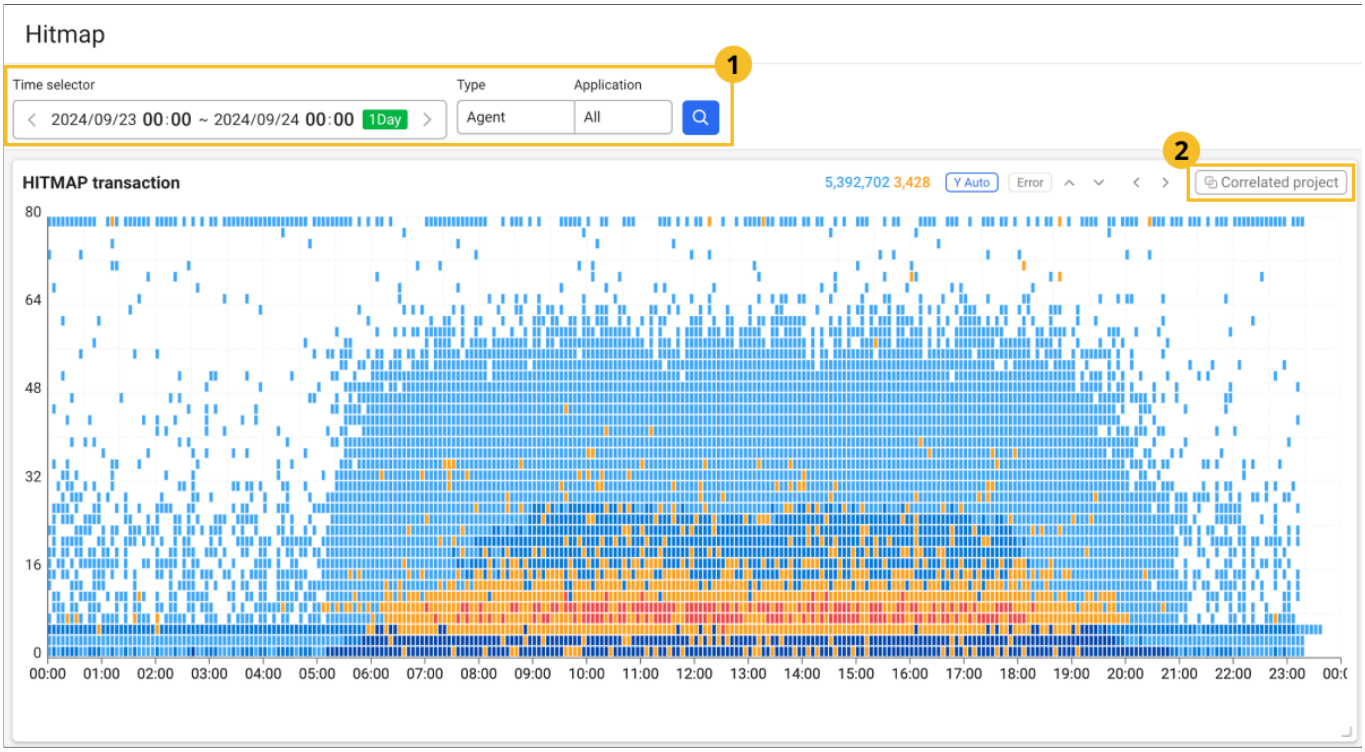
> ⓘ For more information about **Log view detail setting**, see the following.

Additional options

Checking the correlated project metrics

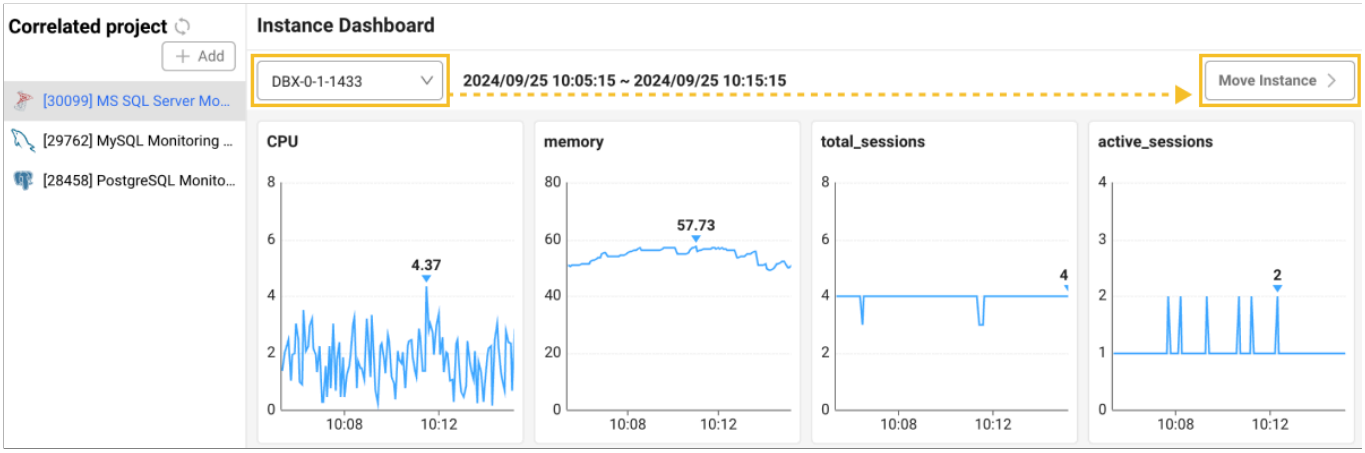
If you have created a database monitoring (DPM) project and an application monitoring (APM) project, you can relate the two projects for monitoring the data collected from DPM in the APM project. You can see performance metrics of the correlated projects while querying transactions via the **Correlated project** feature on the upper right of the **HITMAP transaction** section.

The **Correlated project** feature is useful when analyzing transactions in the APM project. When you found a slow SQL, it is difficult to determine exactly in which DB the problem occurred in the environment using multiple similar DBs.



1. Set the time and **Application** for viewing the transactions, and then select .
2. In the **HITMAP transaction** section, select **Correlated project** on the upper right.
3. A new window appears where you can check the metrics for the correlated project.

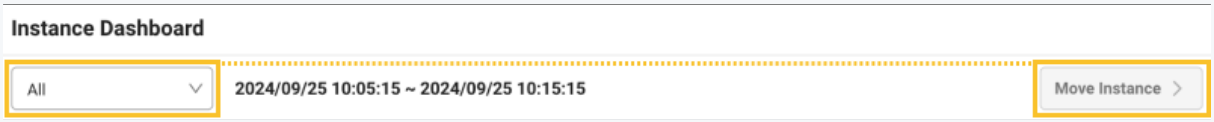
Select the correlated project from the left list and then check the metrics in the **Instance dashboard** section.



To navigate to the dashboard menu for a selected project, select an instance and then click **Move instance** on

the upper right. Go to **Dashboard** > **Monitoring a Database Instance**.

- The **HITMAP transaction** button in the **Correlated project** section appears only when there is an associated project. For more information about adding a correlated project, see the following.
- You can add a project by selecting **Correlated project** at the top of the left **Add** list.
- The **Move instance** button is disabled when **All** is selected.



- Depending on the database product type, you can navigate to the **Analysis** > **Counts Trend** menu.

Viewing HTTP parameters

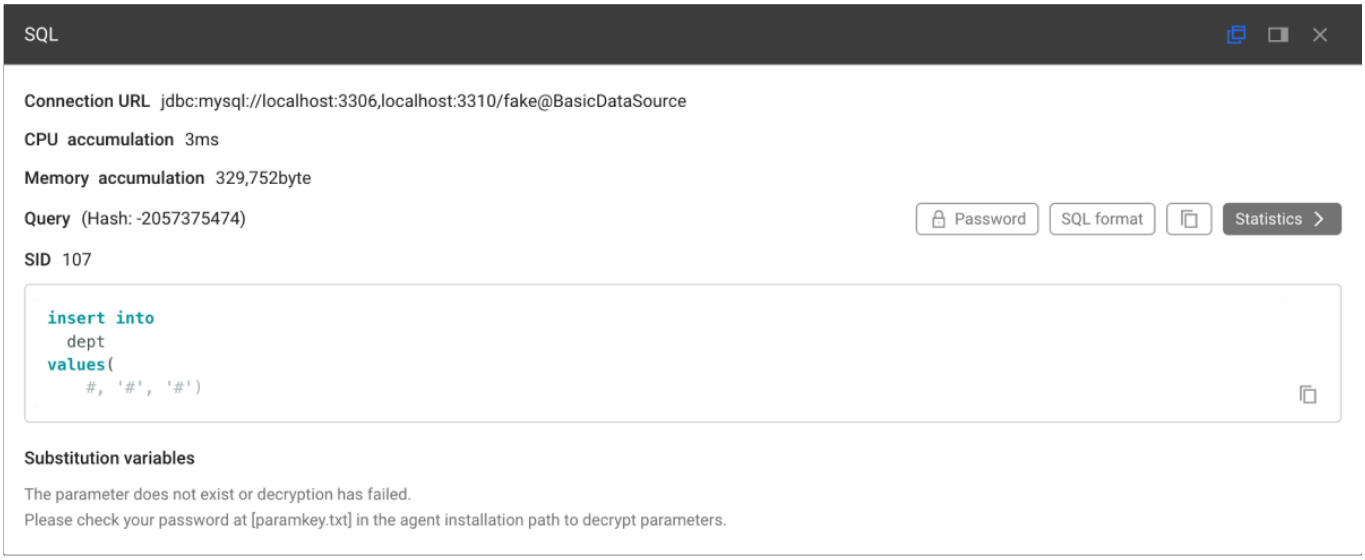
You can view the HTTP parameters of the transaction in the **Table view** tab.

1. Scroll down the page to the final step of the transaction.
2. Select the **HTTP parameters** item.
3. If the **HTTP-PARAMETERS** window appears, select **Password**.
4. Enter the specified Param Key. You can see the hidden parameters.

- For the agent settings for HTTP parameters, see the following.
- **Password**: You can see the decrypted parameter value. The password is a 6-digit string in the `WHATAP_HOME` /security.conf (paramkey.txt) file. You can change it to another string. It is required to view the SQL variables and HTTP queries and to stop the threads. For more information, see the following.

Viewing SQL parameters

Select the SQL step in the **Table view** tab or select in the **Tree view** tab. The **SQL** window appears to display parameters.



- **Password**: You can see the decrypted parameter value. The password is a 6-digit string(`paramkey`) in the `WHATAP_HOME` /security.conf file. You can change it to another string.

  > **Password** is required to view the SQL variables and HTTP queries, and stop the threads. For more information, see the following.

- **SQL format**: You can indent and format your SQL statements for better readability.
- : A SQL statement can be copied to the clipboard.
- **Statistics**: Go to **Statistics** > **SQL**.

To search SQL variables and HTTP queries, add the following options to the agent settings.

- For agent settings related to recording SQL parameters, see the following.
- For agent settings related to recording the HTTP parameters, see the following.

whatap.conf

```
# SQL parameter lookup option: If the option is applied, SQL parameters are collected after being encrypted.
profile_sql_param_enabled=true

// HTTP parameter lookup option: If the option is applied, HTTP query parameters are collected after being encrypted.
```

```
profile_http_parameter_enabled=true
```

Setting the columns in the trace list

You can additionally see the details about the search result by setting the columns in the **TX trace** section.

1. In the **TX Trace** area, click the **Select Column** button.
2. Select or deselect the desired column items. The changes are immediately reflected in the table.
   - **Select all**: It selects all columns.
   - **Deselect all**: All columns are unselected.
   - **Initialize(default)**: The column is set as the default value.
3. Once finished, click the × icon in the upper-right corner.

Filtering the trace list

You can filter the results retrieved from the **TX trace** section using the following options:



- **View all**: You can see all transactions in the viewed result.
- **Active Stack**: Only the transactions that include active stacks can be viewed in the searched result.
- **Multi-transaction**: Only the transactions including multi-transactions can be checked in the searched result.
- **Search URL**: Only transactions containing items that match the entered string are displayed. Enter a desired string and then press `Enter` on the keyboard.
  - **Transaction**
  - **Client IP**
  - **Domain**
  - **Error class**
  - **Error message**

Downloading the viewed list

You can download the results retrieved from the **TX trace** section as a CSV file. The formation of the downloaded CSV file is profile `{project_code}` `YYYYMMDD` .csv.

1. After viewing the transactions, select CSV.
2. Enter a desired value in the **Maximum CSV lines** option.
3. Select **Download**.

Multi Transaction Analysis

Multi-transaction means the transaction involving a different agent or project. **Multiple Transaction Trace** is to trace calls between the application services registered in the WhaTap project.

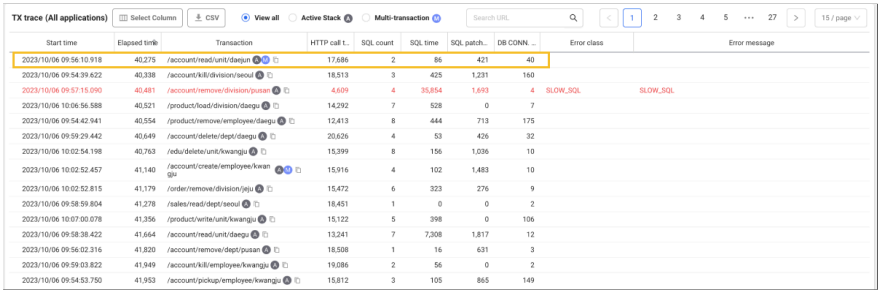
> The Java agent traces multi-transactions with three HTTP header keys (`x-wtap-po` , `x-wtap-mst` , and `x-wtap-sp1`). If HTTP transactions passing through the gateway cannot be traced, check the HTTP header conditions.

> **Enabling the multi-transaction**
>
> To trace multi-transactions, set the `mtrace_enabled` option to `true` in **Management** > **Agent CONFIG.**. For more information about the agent configuration, see the following.

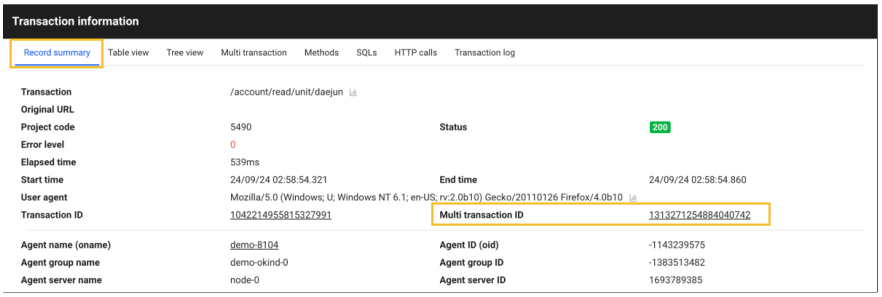
Checking the multi-transaction ID

To use the **Multiple Transaction Trace** menu, **MTID** (Multi Transaction ID) is required. Through the following procedure, you can find the **MTID** value.

1. In **Analysis** > **Hitmap**, drag the chart area.
2. The transaction information in the dragged chart area is displayed in the **TX trace** list below as follows.



3. If you select the trace with the M icon displayed, the **Transaction information** window appears.
4. In the **Record summary** tab, you can see the **Multi transaction ID** value.



- The same **Multi transaction ID** is generated even if the transaction makes an external call. Even if projects are separated by service, all transactions between applications can be found through the initially issued **Multi transaction ID**. For more information about the detailed analysis of transaction trace through the **Transaction information** window, see the following.
- If you select **Multi transaction ID** in the **Transaction information** window, the **Multi transaction** tab appears. The transactions and trace data are provided so that you can see various call relationships at a glance within or between systems, and identify where problems occurred for improvement.

Using the multi-transaction tracing feature

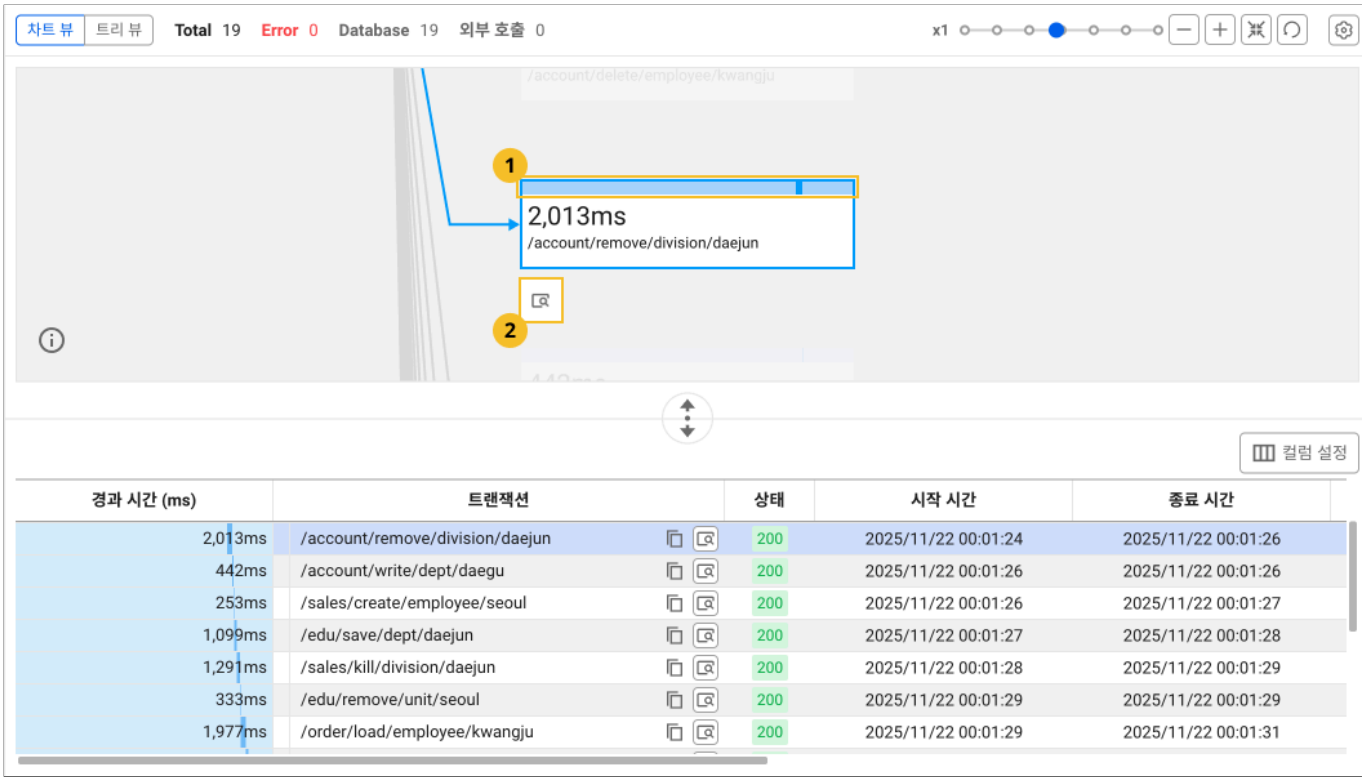
1. Go to **Analysis** > **Multiple Transaction Trace**.
2. Enter the **MTID** value found in the **Transaction information** window into the **Search by MTID / CUSTID** field.
3. Select the date and project you want to query.
4. Click the **Apply** button at the bottom of the screen.
   - The **Apply** button becomes disabled in the following cases:
     - When the same filter values (MTID/CUSTID, date, selected project) as the previous search are set.
     - When none of the filter values (MTID/CUSTID, date, selected project) are entered.
5. The right **Chart** tab displays a diagram and a table that allow you to understand the call relationships of

each transaction.

> The diagram and table areas are divided vertically, and you can adjust their ratio manually.
>
> - Clicking the upward arrow makes the table occupy the entire area. However, if the diagram area was occupying the entire area, the diagram and table will split into equal halves.
> - Clicking the downward arrow makes the chart (diagram) area occupy the entire area. However, if the table area was occupying the entire area, the diagram and table will split into equal halves.

Chart

Chart provides a fast and clear view of the call relationships between transactions.
You can check the individual execution times of transaction services that share the same **Multi transaction ID**.
The elapsed time displayed at the top of each transaction node (1 **Time Bar**) allows you to understand the call flow between transactions.



> You can freely move, zoom in, or zoom out of the chart with your mouse.
>
> - Drag within the chart area to move it.
> - Scroll within the chart area to move vertically or horizontally.
>   - Vertical scroll: move up/down
>   - Horizontal scroll: move left/right
> - Hold `Ctrl` + scroll up/down to zoom in or out.
>   - Pinch zoom on a touchpad is also supported.

When you select a transaction, only the connected nodes remain active, while all unrelated nodes are dimmed.
The **Detail View** button (2) at the bottom of the selected transaction becomes active.
Clicking it opens the **Transaction information** window, where you can view detailed transaction information.
For more details on analyzing transaction traces using the **Transaction information** window, refer to the following document: Transaction Info.

- **Zoom In**
- **Zoom Out**
- **Fit to Screen**: Display all nodes to fit the visible area.
- **Reset Zoom**: Restore the initial chart zoom and position.
- **Chart View Settings**: Show or hide chart elements.
  - **More information**: Show/hide application name, project info, IP, status code.
  - **Database** / **External**: Display other database connections or external HTTP calls as nodes in the chart.

Table

In the **Table** tab, you can see information for each transaction included in a multi-transaction in the table format.

If you select each transaction, the **Transaction information** window for the selected transaction appears, similar to selecting a node on the **Chart** tab. The transaction trace allows you to see the transaction details. For more information about the detailed analysis of transaction trace through the **Transaction information** window, see the following.

Column Settings: You can edit the columns in the table.

Tree

In the **Tree** tab, you can see each transaction and its trace details. The transactions within the overall transaction time or start times and step times of each trace to provide transaction call relationships in a tree format.

- **Single-line View**: You can arrange the tree format at intervals by displaying the texts displayed in the execution data for each section in a single line.
- **Multi-line View**: All the texts displayed in the execution data for each section, appear by wrapping words.
- **Longest Path**: The longest route can appear.
- **Show Time Bar**: The elapsed times are displayed in a bar chart.
- **Show Time**: The timestamps, gaps, and elapsed times for each segment are displayed in text.
  - 8 seconds or more: Excessive delay state is distinguished in red.
  - Between 3 and 8 seconds: Delay state is distinguished in orange.
  - Less than 3 seconds: Normal state is distinguished in blue.

> In case of start and step times, the time differences caused by the transaction call environment are corrected on the screen through traces and mappings in the parent transactions. Accordingly, there may be differences from the actual time data.

- **Hide Time**: The time is hidden.

> If you select button or button, you can see the summary window such as **HTTP call statistics** and **Active stack**.

> To search SQL variables and HTTP queries, add the following options to the agent settings.
>
> - For agent settings related to recording SQL parameters, see the following.
> - For agent settings related to recording the HTTP parameters, see the following.
>
> whatap.conf
>
> ```
> # SQL parameter lookup option: If the option is applied, SQL parameters are collected after being encrypted.
> ```

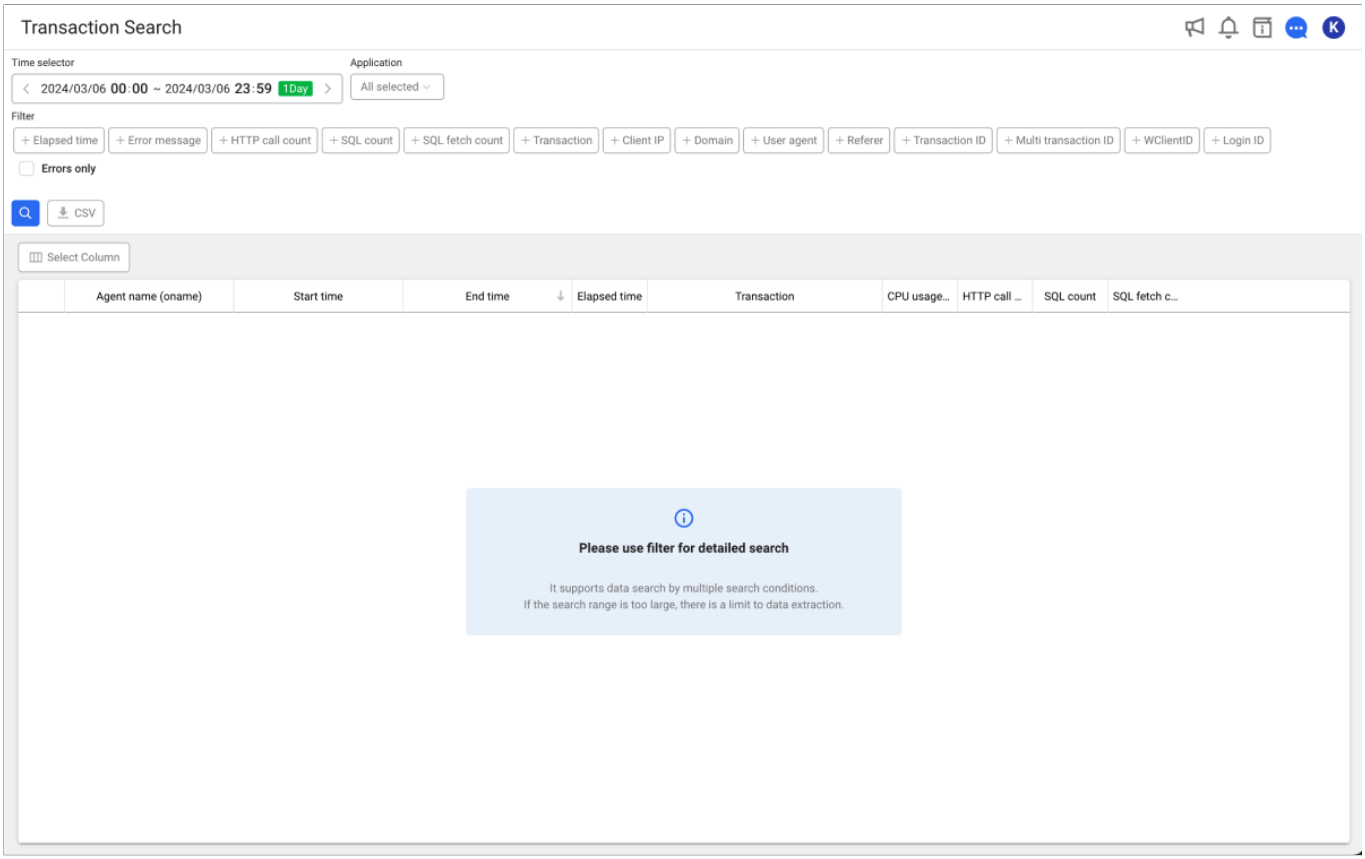
```
profile_sql_param_enabled=true

// HTTP parameter lookup option: If the option is applied, HTTP query parameters are collected after being encrypted.
profile_http_parameter_enabled=true
```

TX Profile

Home > Select Project > **Analysis** > **Transaction Search**

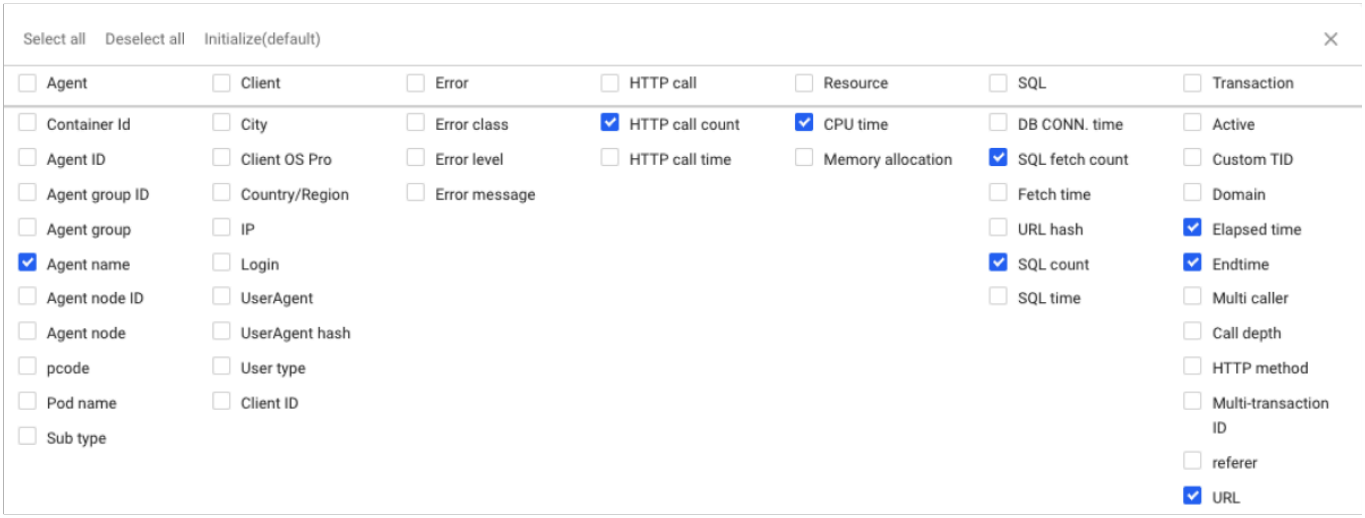
You can view the statistics on the performance properties and execution status for each transaction for a specific period of time. You can view all transactions and quickly check their status (normal or error). Through the provided information, you can find which transactions are busy and resource-consuming. Various filters can be enabled depending on the transaction properties, making it useful in finding transactions with specific conditions.



Set the query time and then select the target from the **Application** option. Select . Then the searched results appear in the table. If you select desired items from the result list, the **Transaction information** window appears. For more information about the detailed analysis of transaction trace through the **Transaction information** window, see the following.

- To search for transactions in error, click **Errors only** and then select .
- It is recommended to set the search time as short as possible because a response delay may occur depending on the search conditions.

Editing columns



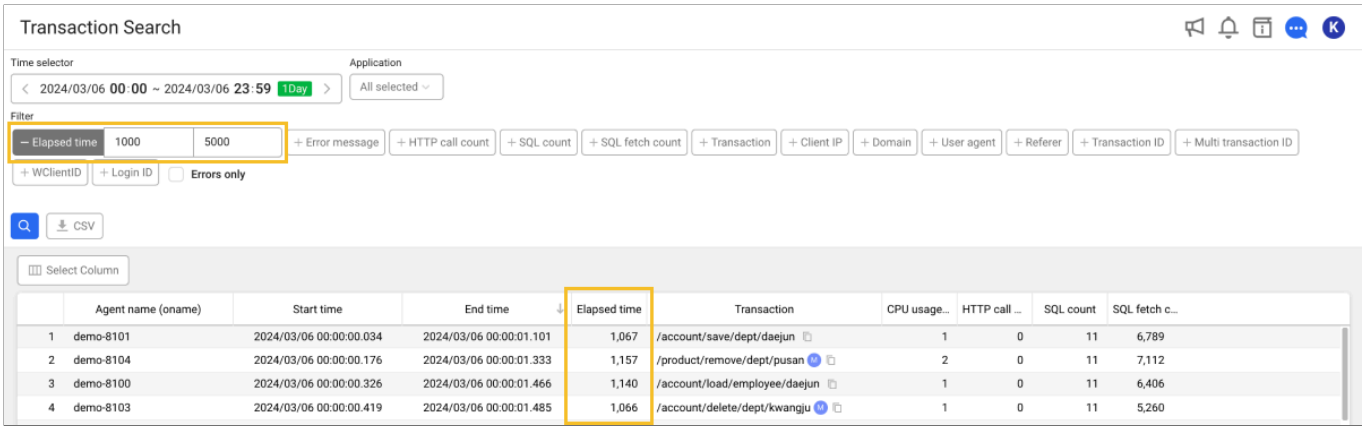
When viewing results, click **Select Column** and then select the columns to be checked.

Applying filters

Use filters for more details. Data exploration is supported using multiple search conditions. After setting a filter(s), select .

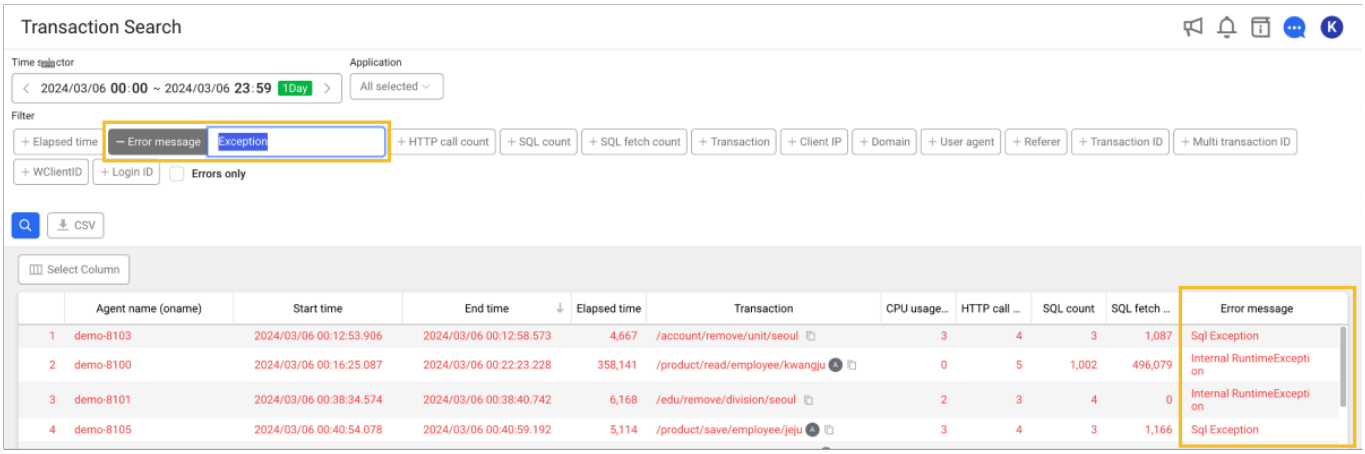
Some filters may not be supported depending on the product.

Elapsed time



Search based on the transaction response time. Search by entering two values above and below the reference.

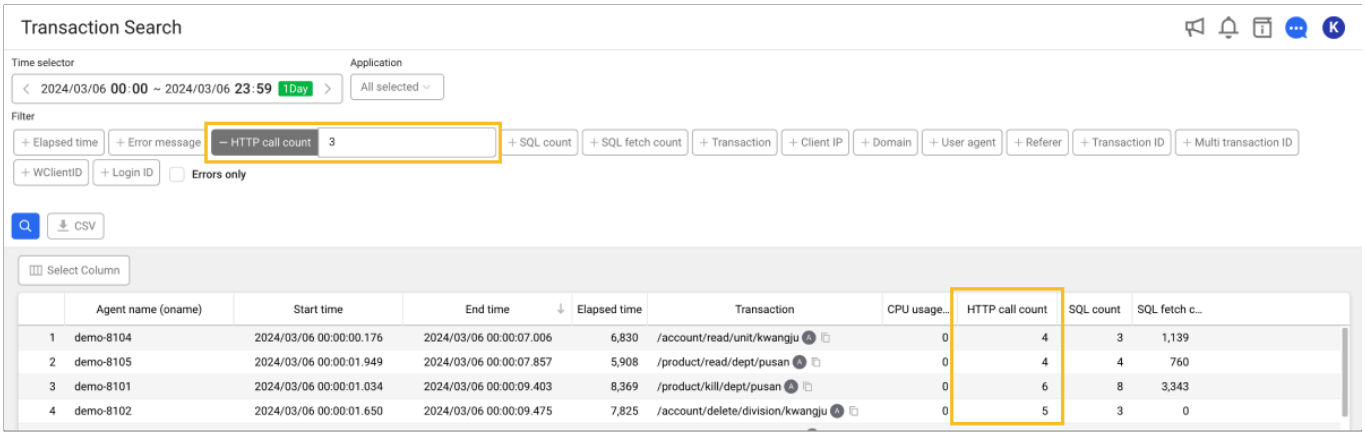
Error message



Search based on the transaction error message. In the default state, error-related columns may not be seen. Select a column.

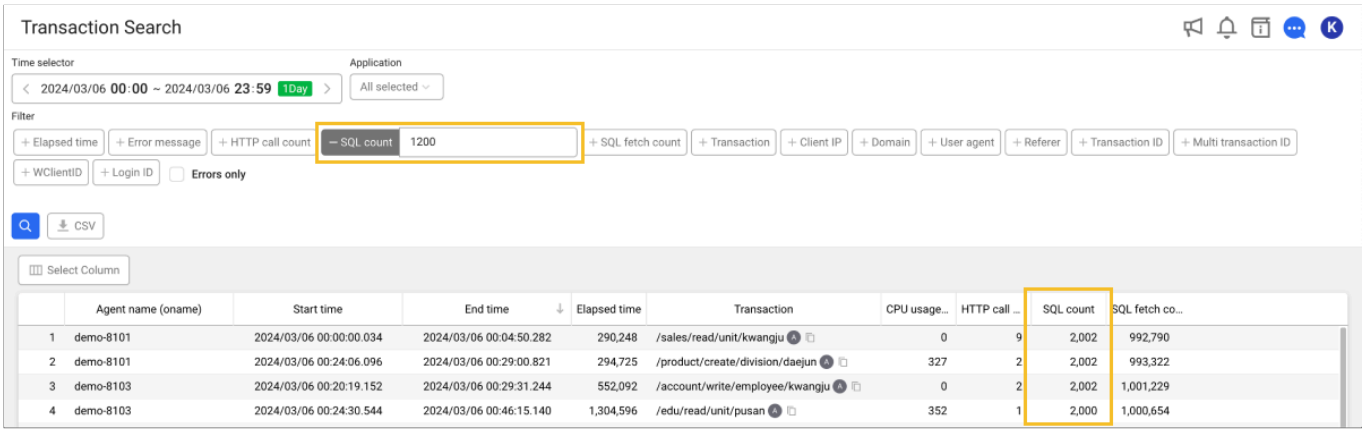
> Transactions in error are displayed in red.

HTTP call count



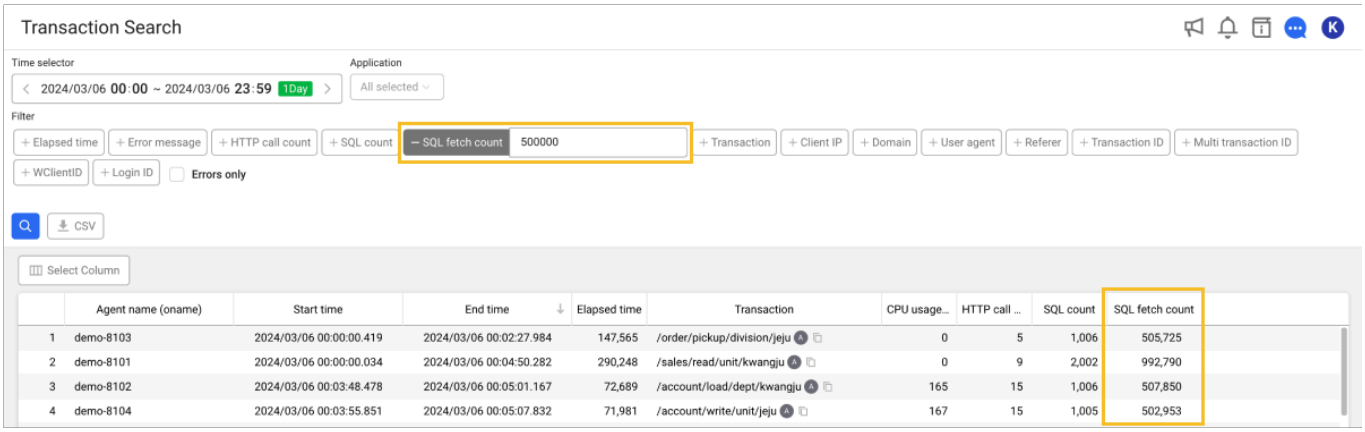
It searches for the transactions that exceed the specified number based on the number of calls for external HTTP services.

SQL count



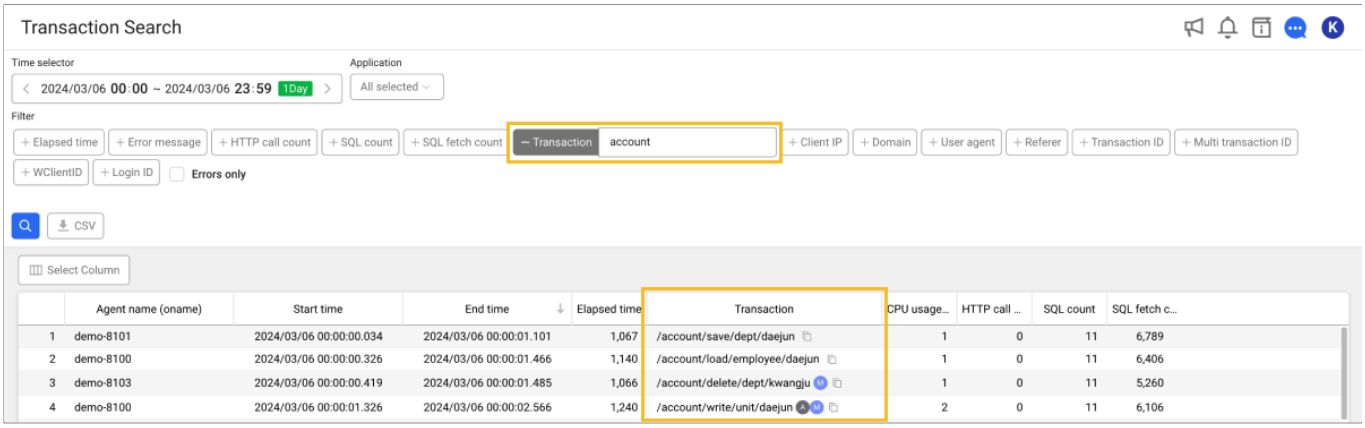
It searches for the transactions where the number of SQL executions per transaction is greater than or equal to the specified number. Exceptionally, this is used to search transactions that call SQLs excessively.

SQL fetch count



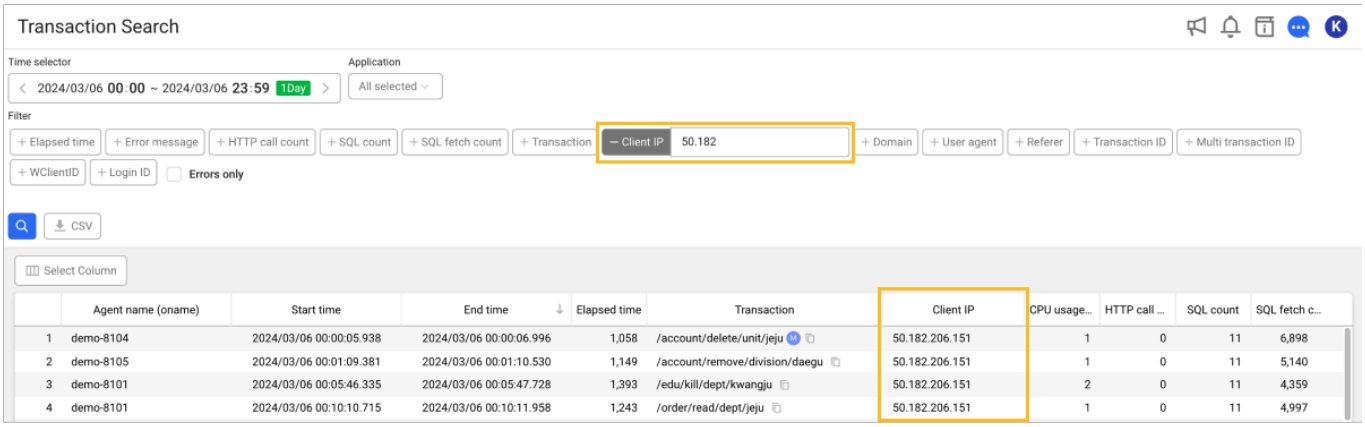
After executing an SQL, it searches for the transactions exceeding the specified number based on the number of DB patches. Use this to search transactions that perform excessive data fetches.

Transaction



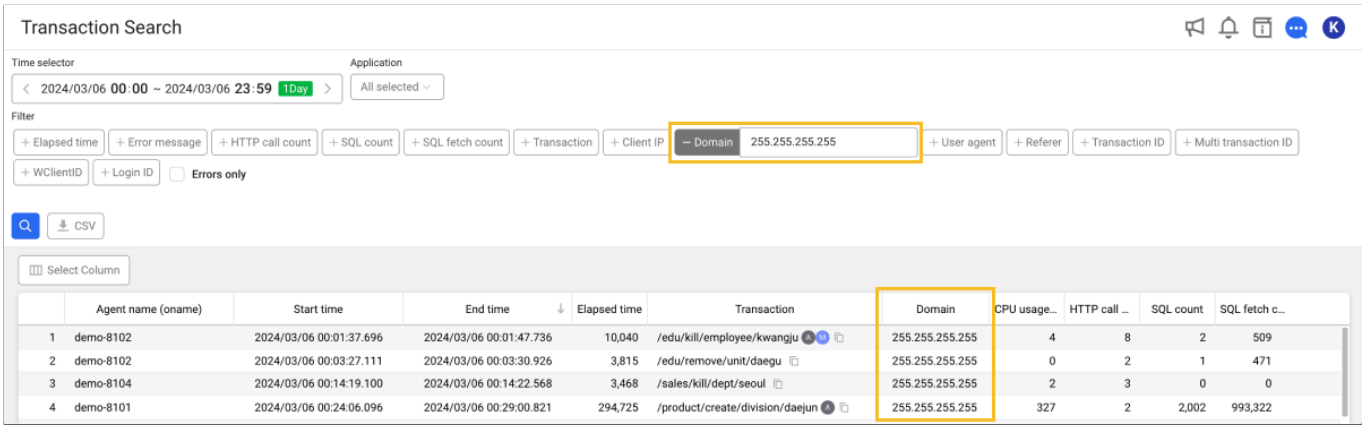
It searches based on the transaction. The entered search string is compared and searched by inclusion (partial matching).

Client IP



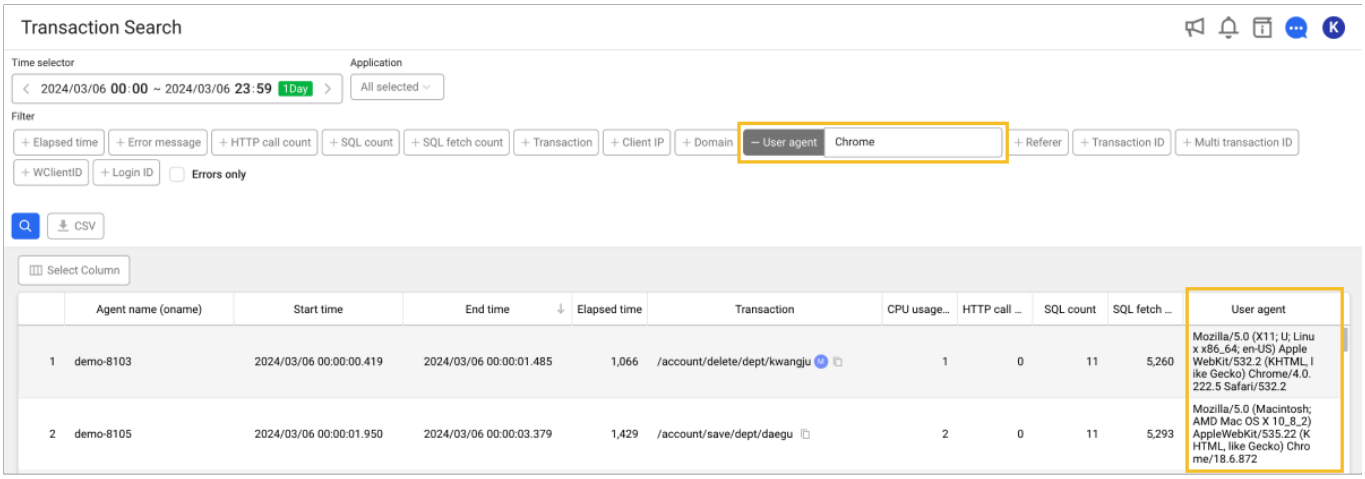
It searches transactions based on the client IP. However, IP strings are searched by prefix comparison (forward matching).

Domain



Retrieve transactions based on the HTTP service domain (host:port). The search conditions must match the domain completely (full match).

User agent



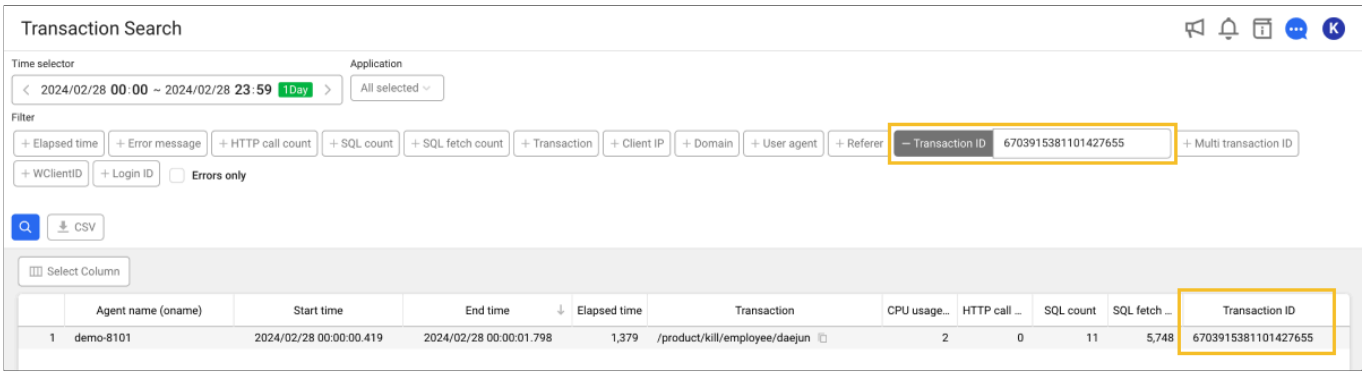
Search transactions based on the user agent (partial match).

Referer

Search transactions based on the referer string.

> Agents in their default settings do not collect referer data. If the referer settings are required, see the following to add the related settings.

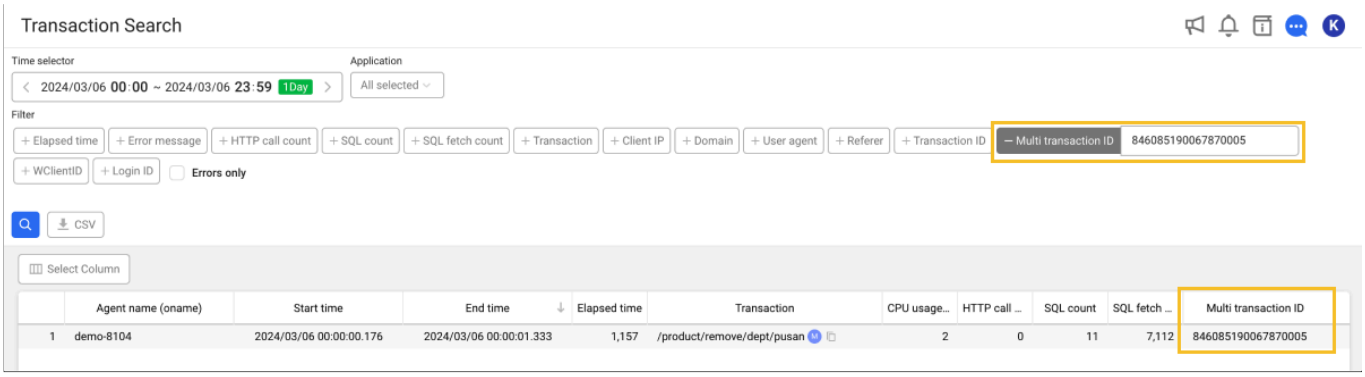
Transaction ID



Search transactions based on their transaction ID (full match). For transaction IDs, the search indexes are created for searching at a high speed.

- The transaction ID can be found in the **Record summary** tab of the **Transaction information** window. For more information, see the following.
- If you are not using the latest version of the agent, a full scan may occur rather than using indexes.

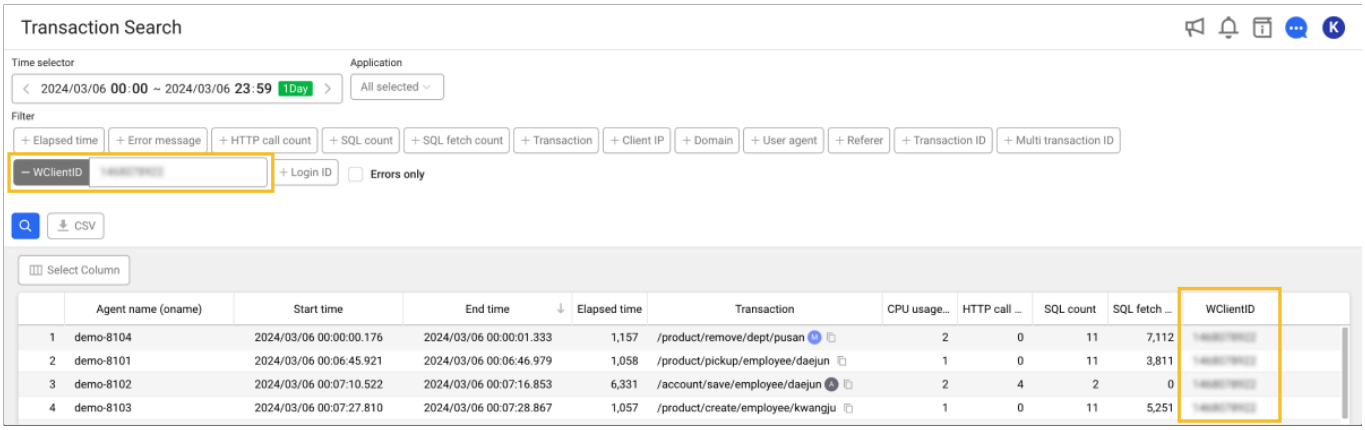
Multi transaction ID



Search transactions based on the multi-transaction ID (full match).

- The multi-transaction ID can be found in the **Record summary** tab of the **Transaction information** window. For more information, see the following.
- If you are not using the latest version of the agent, a full scan may occur rather than using indexes.
- When searching with the **Multi transaction ID** filter applied, only the **Transaction ID** filter can be additionally applied.

WClientID



The client ID (`wClientId`) is an ID given by the WhaTap agent to identify users. Search transactions based on this ID (full match). There is no separate index for searching.

> ⓘ WhaTap basically stores the client-related information.

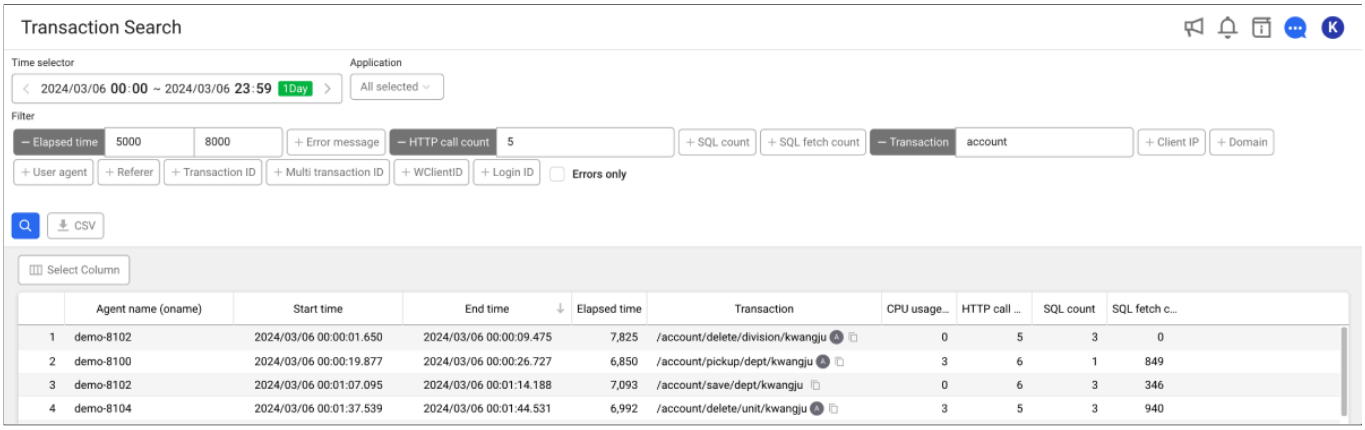
Login ID

Search transactions based on the login ID.

> ⓘ Agents in their default settings do not collect login IDs. For more information about the login ID setting, see the following to add settings.

Applying multiple filters

You can apply multiple filters to see the list of desired transactions. The following case is for the elapsed time between 5 and 8 seconds, 5 or more HTTP calls, and the transaction path including "account."



ⓘ When searching with the **Multi transaction ID** filter applied, only the **Transaction ID** filter can be additionally applied.

Downloading the searched results

The searched results can be downloaded as a CSV file according to the filter conditions set by the user.

1. Click 🔍 to view the searched results.
2. Select CSV.
3. Enter the maximum number of CSV lines. You can save the searched results as many lines as you enter.
4. Select **Download** to download the CSV file.

ⓘ The CSV file name's format is profile_ {pcode} _ YYMMDD .csv.

MSA Analysis

What is the MSA analysis?

The MSA analysis function analyzes the proportion of calls between transactions collected through the WhaTap's **Multiple transaction trace** function. Based on this, dependencies between microservices can be checked at the URL level rather than instance-centric basis.

Microservice Architecture Pattern

For example, there are five microservices: A, B, C, D, and E. If a user calls A, it is assumed that the following three call patterns occur according to the logic.

1. A→B, C calls
2. A→B, C calls + then B→D call
3. A→B, C calls + then B→D, E calls

If these three patterns have been performed once, MSA analysis can check the association analysis data depending on the reference URL.

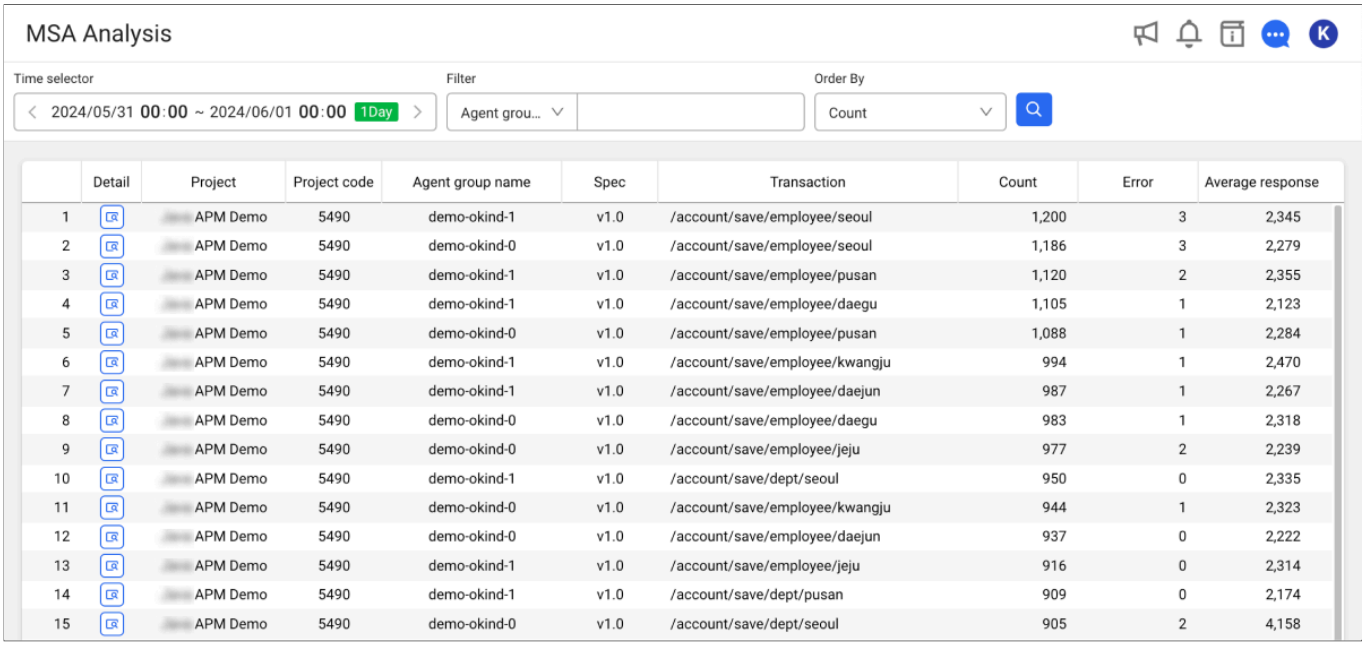
Microservice Architecture Caller & Callee

> For more information about **multiple transaction trace**, see the following.

MSA Transaction Statistics

Home > Select Project > **Sitemap** > **Analysis** > **MSA Analysis**

The **MSA Analysis** menu is a tool that can effectively analyze the performance and stability for each service in the microservice environment. By analyzing the detailed call relationships for transactions, you can optimize the system performance and proactively identify potential problems based on the analysis.



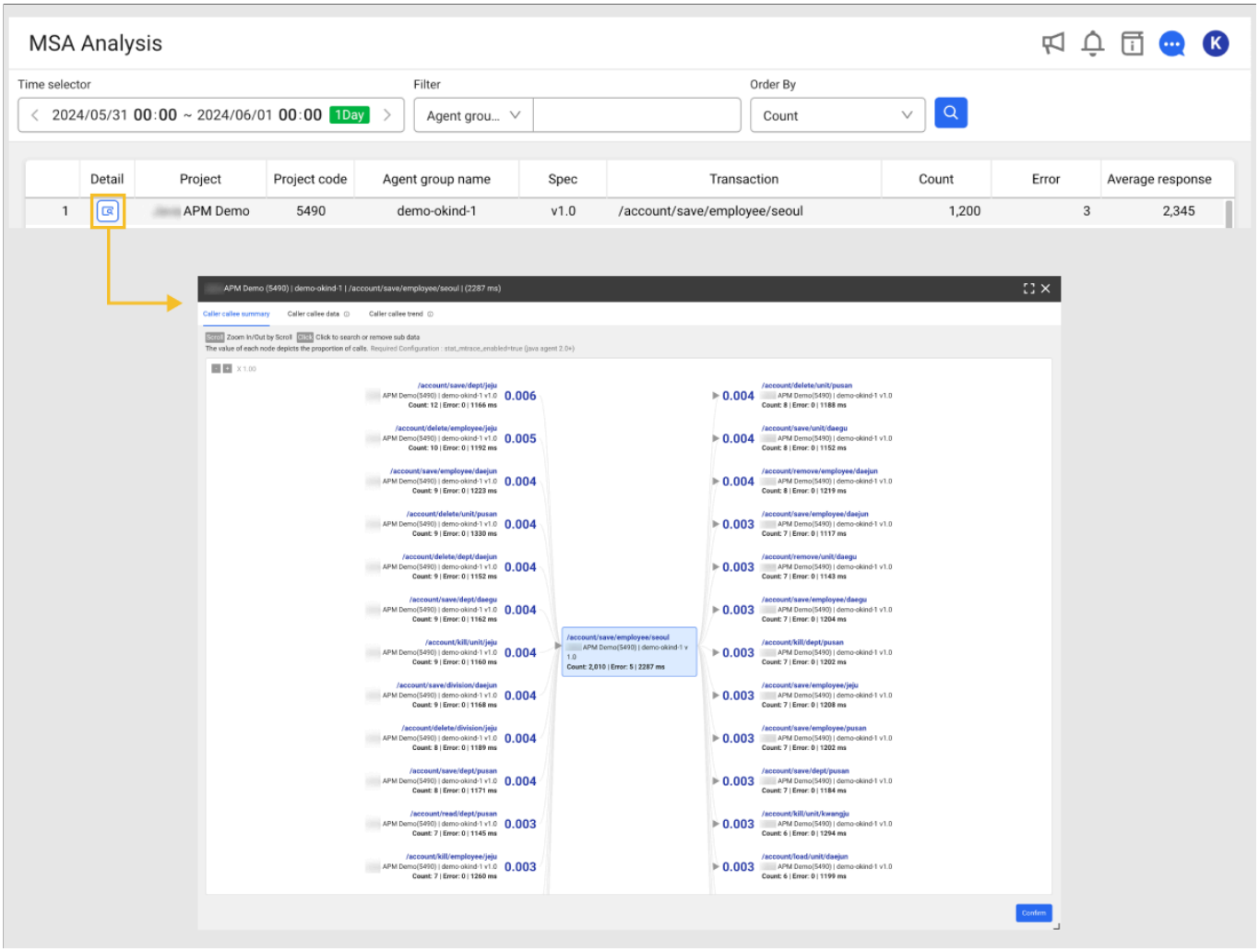
	Detail	Project	Project code	Agent group name	Spec	Transaction	Count	Error	Average response
1		APM Demo	5490	demo-okind-1	v1.0	/account/save/employee/seoul	1,200	3	2,345
2		APM Demo	5490	demo-okind-0	v1.0	/account/save/employee/seoul	1,186	3	2,279
3		APM Demo	5490	demo-okind-1	v1.0	/account/save/employee/pusan	1,120	2	2,355
4		APM Demo	5490	demo-okind-1	v1.0	/account/save/employee/daegu	1,105	1	2,123
5		APM Demo	5490	demo-okind-0	v1.0	/account/save/employee/pusan	1,088	1	2,284
6		APM Demo	5490	demo-okind-1	v1.0	/account/save/employee/kwangju	994	1	2,470
7		APM Demo	5490	demo-okind-1	v1.0	/account/save/employee/daejun	987	1	2,267
8		APM Demo	5490	demo-okind-0	v1.0	/account/save/employee/daegu	983	1	2,318
9		APM Demo	5490	demo-okind-0	v1.0	/account/save/employee/jeju	977	2	2,239
10		APM Demo	5490	demo-okind-1	v1.0	/account/save/dept/seoul	950	0	2,335
11		APM Demo	5490	demo-okind-0	v1.0	/account/save/employee/kwangju	944	1	2,323
12		APM Demo	5490	demo-okind-0	v1.0	/account/save/employee/daejun	937	0	2,222
13		APM Demo	5490	demo-okind-1	v1.0	/account/save/employee/jeju	916	0	2,314
14		APM Demo	5490	demo-okind-1	v1.0	/account/save/dept/pusan	909	0	2,174
15		APM Demo	5490	demo-okind-0	v1.0	/account/save/dept/seoul	905	2	4,158

- **Time selector**: Select a desired search period. You can view a list of transactions including callers and callees on a daily basis.
- **Filter**: By filtering based on **Agent group name**, **Spec**, and **Transaction**, you can perform the search.
- **Order By**: Based on **Count**, **Error**, and **Time avg** (average response time), the list can be sorted.

> - The time range that can be searched in **Time selector** is up to 1 day (24 hours). For more information on how to use the **Time selector** option.
> - **Agent group name** is the group name classified as `whatap.okind` in the agent configuration options.
> - After setting the conditions, select to update the transaction list.

Analyzing in detail

When you select for the item to view in the transaction list, the Display Detail window appears. You can view the caller and callee for the transaction in the Display Detail window.



Each tab provides the following analysis features:

- **Caller callee summary**: It provides a visualized diagram for the relationship between the caller and callee of the transaction.
- **Caller callee data**: It provides the details of the transaction such as call count, error count, and response time.
- **Caller callee trend**: It provides a time series chart for performance changes in the number and rate of transaction calls over time.

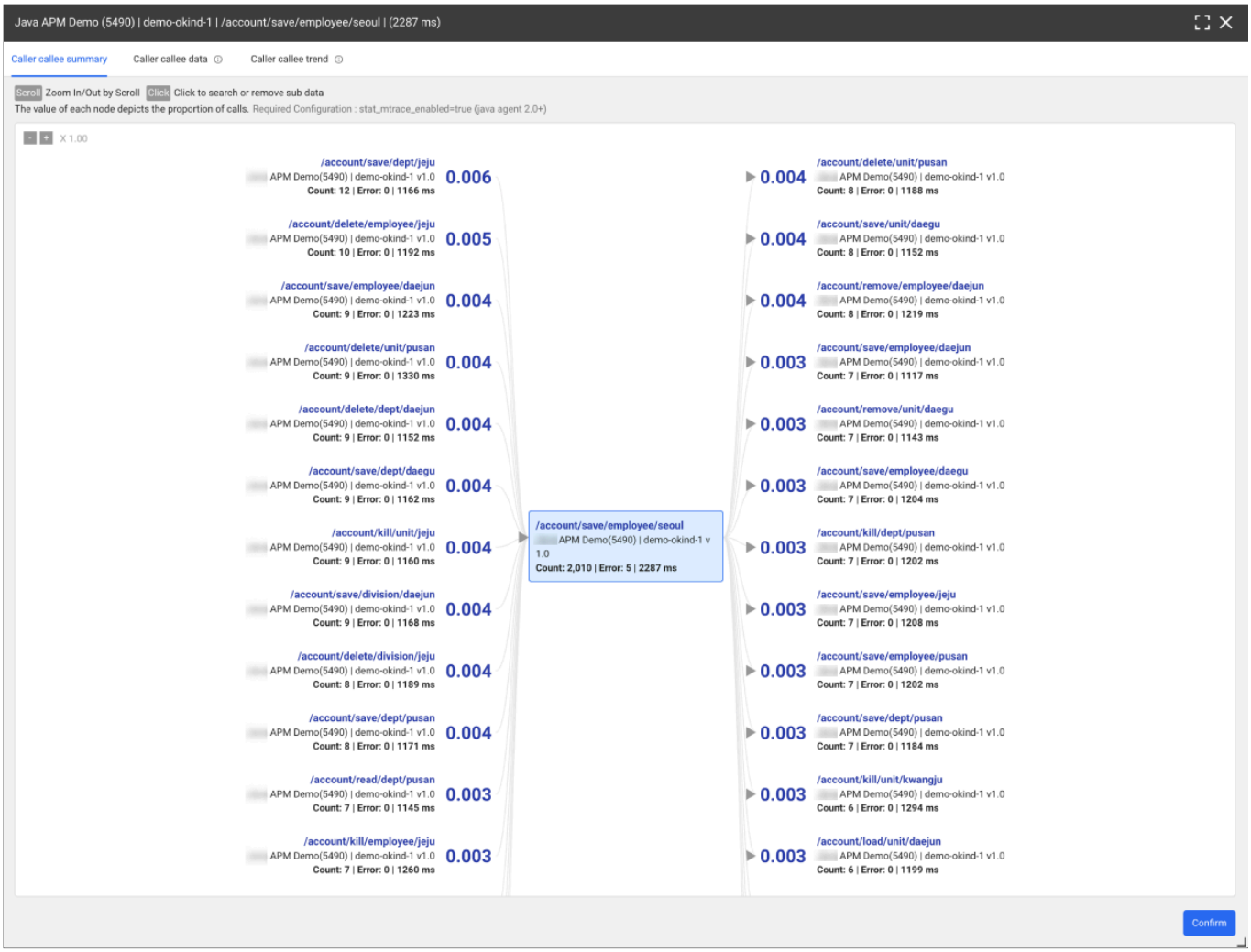
> ⓘ • You can view it in full screen when the ⛶ button is selected on the upper right of the Display Detail window.

- To close the Display Detail window, select **Confirm** at the lower right or press the ESC key.

Caller callee summary

The **Caller callee summary** tab displays the dependency between the caller and callee of a transaction selected from the transaction list in a visualized diagram. You can intuitively understand the call relationships between services. Call paths and dependencies can be visually identified, which is advantageous for analyzing the causes of problems.

Analyzing call relationships



Caller and callee nodes display numerical data whose maximum is 1 such as 0.004 and 0.003, which represents its proportion among all callers. Likewise, the weight of each callee is displayed together on the right.

Through the above screen, you can analyze the dependencies between callers and callees based on the /account/save/employee/seoul transaction. You can visually see from and to which services the transactions are called.

Checking the performance metrics

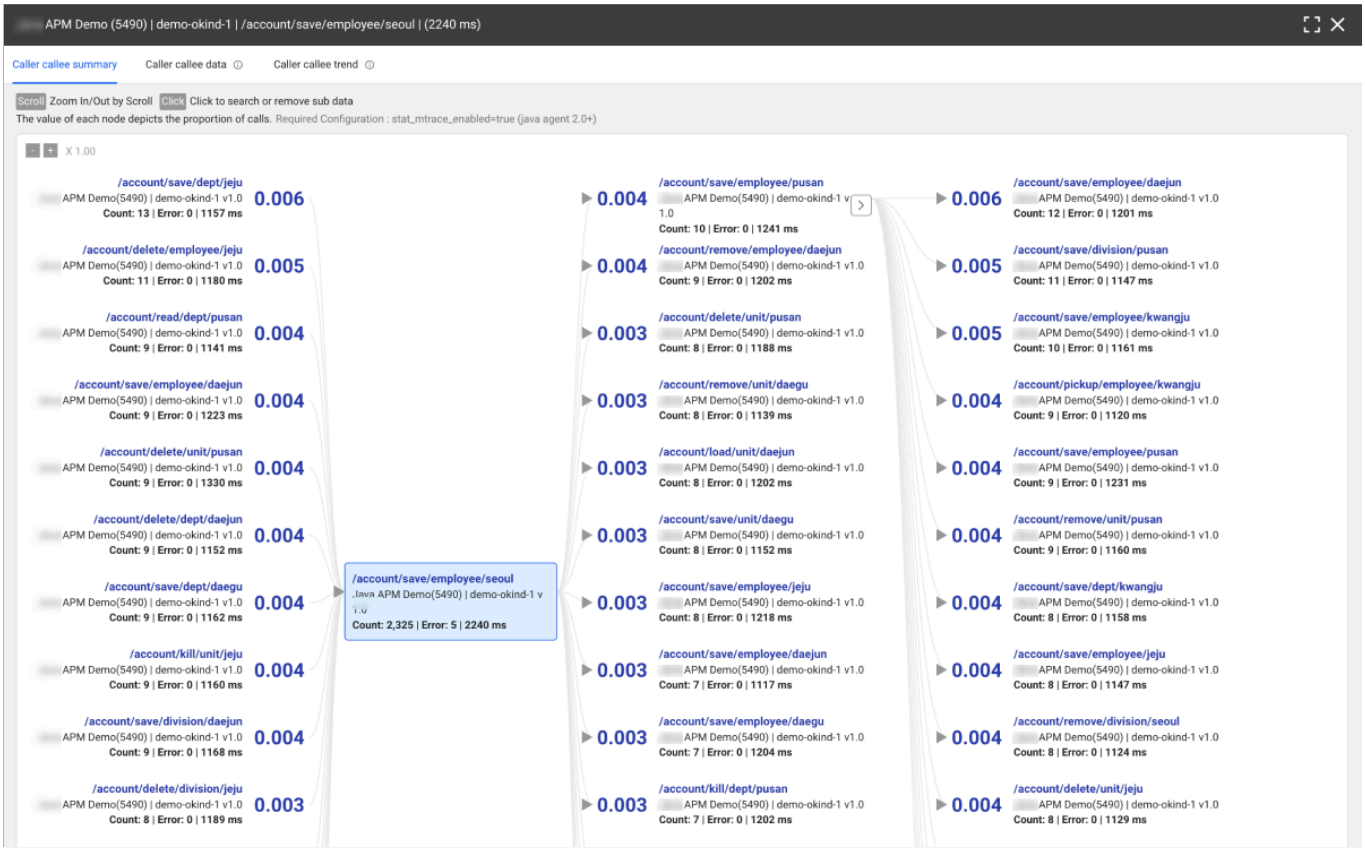


Each node consists of a transaction URL, project name, agent group name, and the following performance data:

- **Count**: Number of calls for the transaction
- **Error**: Number of errors occurred for calls
- Average response time: Average response time for the transaction, in milliseconds (ms)

Finding the transaction flow

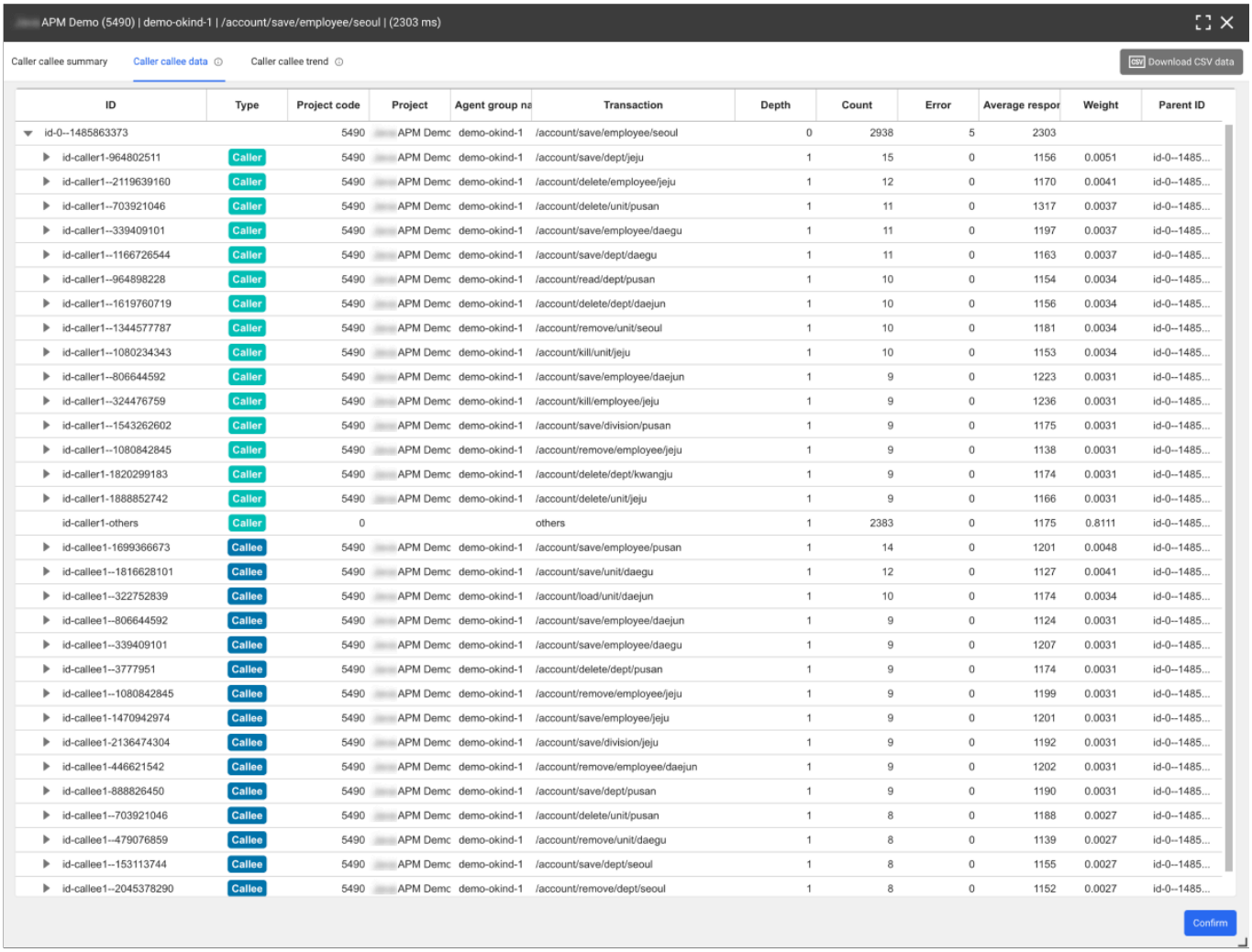
When you hover your mouse over a caller or callee node, the > button appears. If you select this button, it expandes into subnodes. By checking the entire flow of transactions, you can get answers in finding where problems occur.



- To close the subnodes, select > again.
- You can enlarge or reduce the screen using the scroll feature of the mouse. Or click the +/- buttons on the upper left of the screen.

Caller callee data

The **Caller callee data** tab displays the correlation between caller and callee for the reference URL (e.g. /account/save/employee/seoul) in a tabular format. You can specifically check the details of each transaction (call count, error count, average response time, etc.), enabling a precise, data-based analysis.



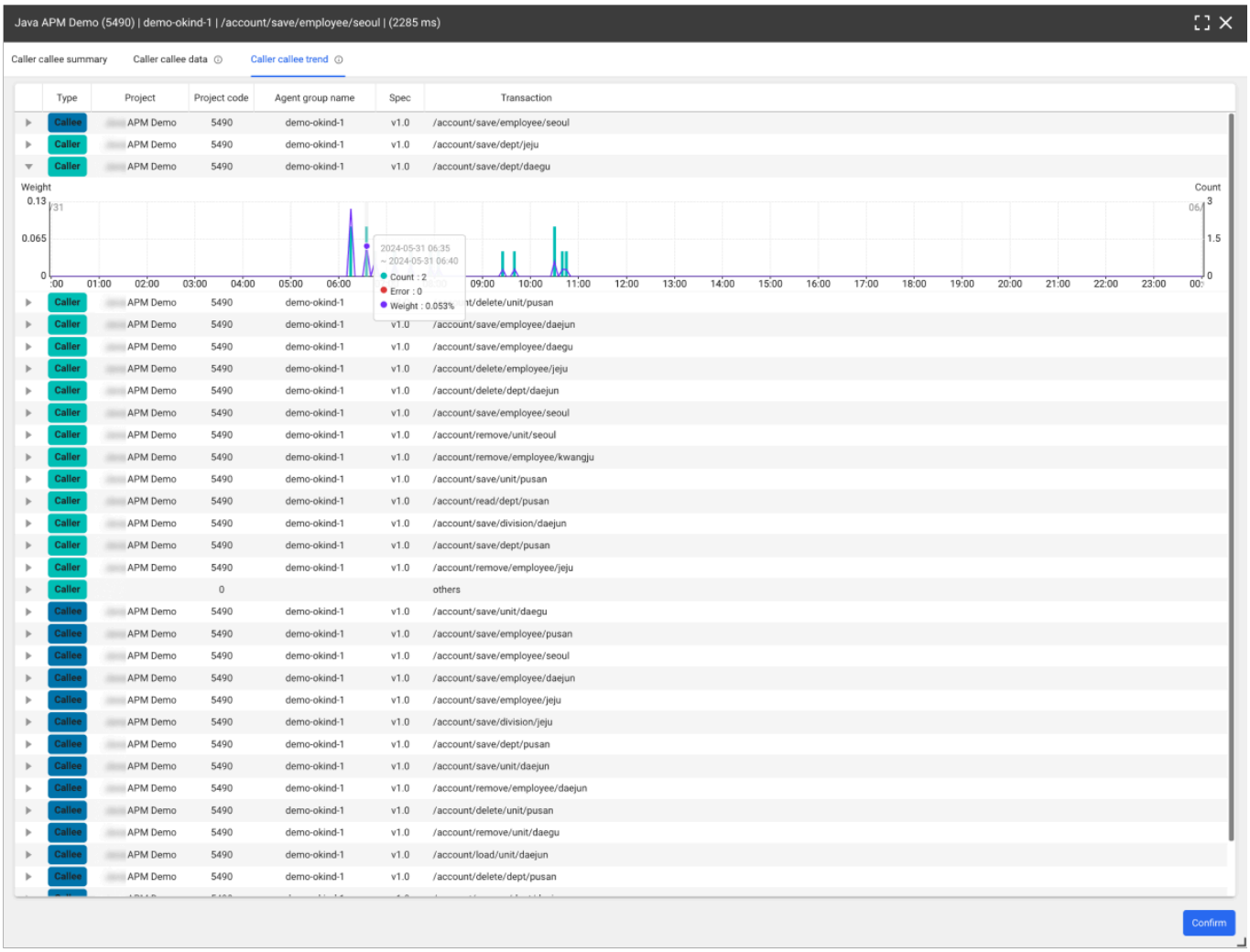
ID	Type	Project code	Project	Agent group na	Transaction	Depth	Count	Error	Average respor	Weight	Parent ID
id-0--1485863373		5490	APM Demc	demo-okind-1	/account/save/employee/seoul	0	2938	5	2303		
id-caller1-964802511	Caller	5490	APM Demc	demo-okind-1	/account/save/dept/jeju	1	15	0	1156	0.0051	id-0--1485...
id-caller1--2119639160	Caller	5490	APM Demc	demo-okind-1	/account/delete/employee/jeju	1	12	0	1170	0.0041	id-0--1485...
id-caller1--703921046	Caller	5490	APM Demc	demo-okind-1	/account/delete/unit/pusan	1	11	0	1317	0.0037	id-0--1485...
id-caller1--339409101	Caller	5490	APM Demc	demo-okind-1	/account/save/employee/daegu	1	11	0	1197	0.0037	id-0--1485...
id-caller1--1166726544	Caller	5490	APM Demc	demo-okind-1	/account/save/dept/daegu	1	11	0	1163	0.0037	id-0--1485...
id-caller1--964898228	Caller	5490	APM Demc	demo-okind-1	/account/read/dept/pusan	1	10	0	1154	0.0034	id-0--1485...
id-caller1--1619760719	Caller	5490	APM Demc	demo-okind-1	/account/delete/dept/daejun	1	10	0	1156	0.0034	id-0--1485...
id-caller1--1344577787	Caller	5490	APM Demc	demo-okind-1	/account/remove/unit/seoul	1	10	0	1181	0.0034	id-0--1485...
id-caller1--1080234343	Caller	5490	APM Demc	demo-okind-1	/account/kill/unit/jeju	1	10	0	1153	0.0034	id-0--1485...
id-caller1--806644592	Caller	5490	APM Demc	demo-okind-1	/account/save/employee/daejun	1	9	0	1223	0.0031	id-0--1485...
id-caller1--324476759	Caller	5490	APM Demc	demo-okind-1	/account/kill/employee/jeju	1	9	0	1236	0.0031	id-0--1485...
id-caller1--1543262602	Caller	5490	APM Demc	demo-okind-1	/account/save/division/pusan	1	9	0	1175	0.0031	id-0--1485...
id-caller1--1080842845	Caller	5490	APM Demc	demo-okind-1	/account/remove/employee/jeju	1	9	0	1138	0.0031	id-0--1485...
id-caller1-1820299183	Caller	5490	APM Demc	demo-okind-1	/account/delete/dept/kwangju	1	9	0	1174	0.0031	id-0--1485...
id-caller1-1888852742	Caller	5490	APM Demc	demo-okind-1	/account/delete/unit/jeju	1	9	0	1166	0.0031	id-0--1485...
id-caller1-others	Caller	0			others	1	2383	0	1175	0.8111	id-0--1485...
id-callee1-1699366673	Callee	5490	APM Demc	demo-okind-1	/account/save/employee/pusan	1	14	0	1201	0.0048	id-0--1485...
id-callee1--1816628101	Callee	5490	APM Demc	demo-okind-1	/account/save/unit/daegu	1	12	0	1127	0.0041	id-0--1485...
id-callee1--322752839	Callee	5490	APM Demc	demo-okind-1	/account/load/unit/daejun	1	10	0	1174	0.0034	id-0--1485...
id-callee1--806644592	Callee	5490	APM Demc	demo-okind-1	/account/save/employee/daejun	1	9	0	1124	0.0031	id-0--1485...
id-callee1--339409101	Callee	5490	APM Demc	demo-okind-1	/account/save/employee/daegu	1	9	0	1207	0.0031	id-0--1485...
id-callee1--3777951	Callee	5490	APM Demc	demo-okind-1	/account/delete/dept/pusan	1	9	0	1174	0.0031	id-0--1485...
id-callee1--1080842845	Callee	5490	APM Demc	demo-okind-1	/account/remove/employee/jeju	1	9	0	1199	0.0031	id-0--1485...
id-callee1-1470942974	Callee	5490	APM Demc	demo-okind-1	/account/save/employee/jeju	1	9	0	1201	0.0031	id-0--1485...
id-callee1-2136474304	Callee	5490	APM Demc	demo-okind-1	/account/save/division/jeju	1	9	0	1192	0.0031	id-0--1485...
id-callee1-446621542	Callee	5490	APM Demc	demo-okind-1	/account/remove/employee/daejun	1	9	0	1202	0.0031	id-0--1485...
id-callee1-888826450	Callee	5490	APM Demc	demo-okind-1	/account/save/dept/pusan	1	9	0	1190	0.0031	id-0--1485...
id-callee1--703921046	Callee	5490	APM Demc	demo-okind-1	/account/delete/unit/pusan	1	8	0	1188	0.0027	id-0--1485...
id-callee1--479076859	Callee	5490	APM Demc	demo-okind-1	/account/remove/unit/daegu	1	8	0	1139	0.0027	id-0--1485...
id-callee1--153113744	Callee	5490	APM Demc	demo-okind-1	/account/save/dept/seoul	1	8	0	1155	0.0027	id-0--1485...
id-callee1--2045378290	Callee	5490	APM Demc	demo-okind-1	/account/remove/dept/seoul	1	8	0	1152	0.0027	id-0--1485...

- It can be used to analyze performance issues by checking the number and rate of calls, number of errors, and average response time of each transaction.
- Through the **Type** column, you can easily distinguish between caller and callee and find the call relationship.
- By selecting ▶ for each transaction, you can additionally check the transactions related to the transaction.
- To save the data list in CSV format, select **Download CSV data** at the upper right.

Caller callee trend

The **Caller callee trend** tab allows you to visually analyze the call relationships and performance data of the reference URL (/account/save/employee/seoul) over time. This can help you find the pattern in the performance

of transactions.



- You can see performance changes at a glance by visualizing the number and ratio of calls for specific transactions on a graph over time.
- Through the **Type** column, you can easily distinguish between caller and callee and find the call relationship.
- When you hover your mouse over the time series chart, you can see the numerical data through the tooltip.
- When you click a specific time zone on the chart, the **Analysis** > **Transaction Search** menu appears. You can view a list of transactions in the selected time zone.

Java Metrics

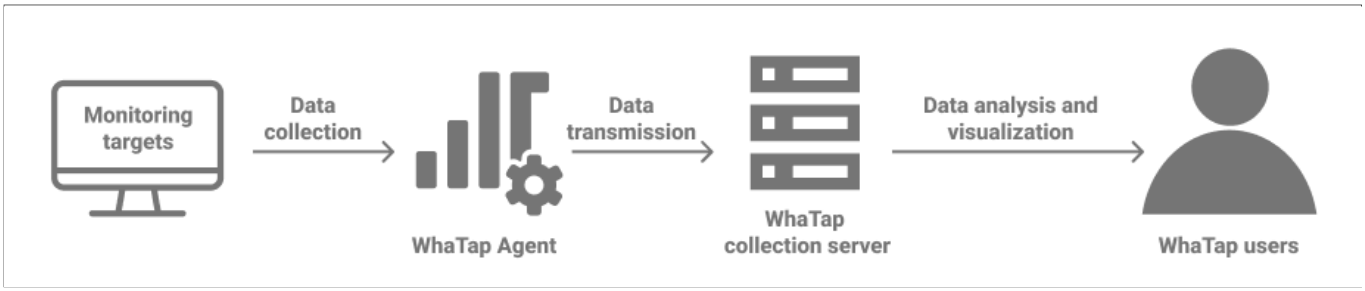
What is a metric?

A **metric** is time-series data that represents the state of a monitoring target as numerical values. It is the result of processing performance data collected by the WhaTap agent from monitoring targets.

Through metrics, you can view key performance indicators such as memory usage per server and DB connection time at a glance. After identifying a problem, you can perform detailed analysis through logs and traces.

Resource usage statistics help you estimate the required amount of resources, which can also be used for cost optimization.

How WhaTap collects metrics



To collect metrics, install the WhaTap agent on the monitoring target. Metrics sent by the agent are stored on the WhaTap collection server.

For example, to monitor **Java applications**, first install the agent. See Application metrics for available metrics.

Metric components

WhaTap **metrics** consist of the following components:

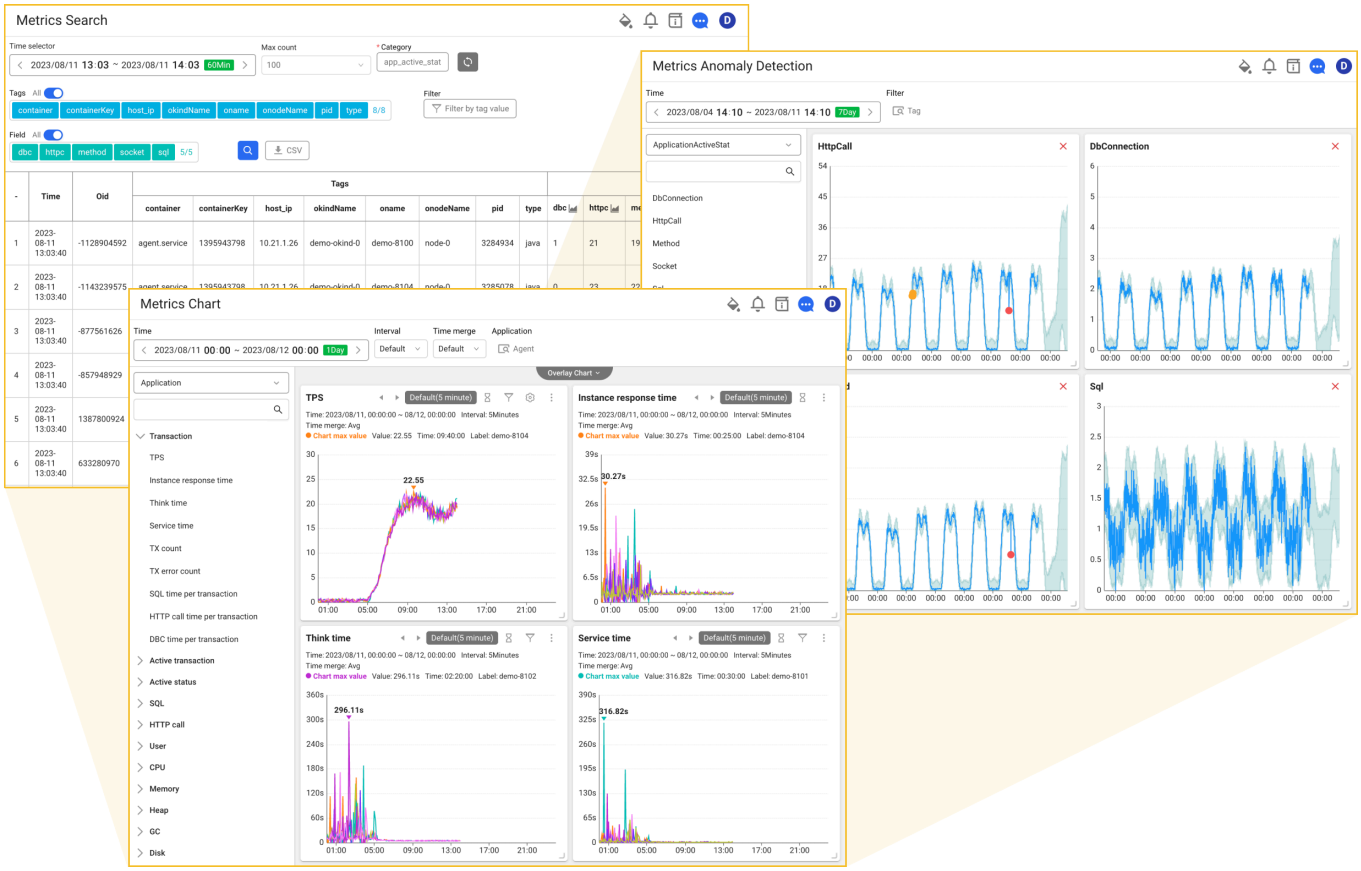
- **Category**: A name that groups related metrics.

- **Tags**: Labels that identify collection targets. Stores rarely-changing unique information such as IP, Oname, and Host. A metric can contain multiple tags.
- **Fields**: Stores actual measured values. A metric can contain multiple fields.
- **Time**: The time when the metric was collected.
- **Oid**: The unique number of the agent that collected the metric.
- **Oname**: The name of the agent that collected the metric.

Viewing and visualizing metrics

WhaTap provides menus to view collected metrics in various ways.

- **Metrics Search**: View the original metric data.
- **Metrics Chart**: View metric data through visualized charts.
- **Metrics Anomaly Detection**: Detect anomalies by comparing with metric patterns learned by AI.


Metrics Search
Time selector
2023/08/11 13:03 ~ 2023/08/11 14:03 60Min
Max count
100
*Category
app_active_stat
Tags All
container containerKey host_ip okindName oname onodeName pid type 8/8
Filter
Filter by tag value
Field All
dbc httpc method socket sql 5/5
CSV
Time
Oid
Tags
container containerKey host_ip okindName oname onodeName pid type dbc httpc
1 2023-08-11 13:03:40 -1128904592 agent.service 1395943798 10.21.1.26 demo-okind-0 demo-8100 node-0 3284934 java 1 21
2 2023-08-11 13:03:40 -1143239575
3 2023-08-11 13:03:40 -877561626
4 2023-08-11 13:03:40 -857948929
5 2023-08-11 13:03:40 1387800924
6 2023-08-11 13:03:40 633280970
Metrics Anomaly Detection
Time
2023/08/04 14:10 ~ 2023/08/11 14:10 7Day
Filter
Tag
ApplicationActiveStat
DbConnection
HttpCall
Method
Socket
HttpCall
DbConnection
Sql
Metrics Chart
Time
2023/08/11 00:00 ~ 2023/08/12 00:00 1Day
Interval
Default
Time merge
Default
Application
Agent
Overlay Chart
Application
Transaction
TPS
Instance response time
Think time
Service time
TX count
TX error count
SQL time per transaction
HTTP call time per transaction
DBC time per transaction
Active transaction
Active status
SQL
HTTP call
User
CPU
Memory
Heap
GC
Disk
TPS
Default(5 minute)
Chart max value Value: 22.55 Time: 09:40:00 Label: demo-8104
22.55
Instance response time
Default(5 minute)
Chart max value Value: 30.27s Time: 00:25:00 Label: demo-8104
30.27s
Think time
Default(5 minute)
Chart max value Value: 296.11s Time: 02:20:00 Label: demo-8102
296.11s
Service time
Default(5 minute)
Chart max value Value: 316.82s Time: 00:30:00 Label: demo-8101
316.82s

Application Metrics

Home > Select Project > **Analysis** > **Metrics Search**

Metrics available in the **Metrics Search** menu. Each metric contains the following information. Some metrics may not have tag information.

- **Tags**: Information to identify the collection target (agent name, IP, etc.)
- **Fields**: Actual measured values (CPU usage, response time, etc.)

---

agent_status_summary

This category collects metrics related to agent status every 10 seconds.

Fields

Field	Type	Unit	Description
inActTime	-	Millisecond(ms)	Amount of time the agent remains inactive
isActive	Boolean	-	Whether the agent is active or not
isRestart	Boolean	-	Whether the agent was restarted recently (`true` / `false`)
lastActTime	-	Millisecond(ms)	Last time when the agent was activated (`0`: Disabled)
oid	-	-	Unique IDs for each agent in the project
startTime	-	Millisecond(ms)	Timestamp when the agent was started.

app_active_stat

- Collects active transaction step metrics
- Collection interval: 5 seconds
- Statistical data: 5 minutes, 1 hour

Tags

Tag	Type	Unit	Description
container	-	-	Container name (unique)
containerKey	-	-	Container key (unique)
host_ip	-	-	Host IP address (unique)
okindName	-	-	Group name categorized as whatap.okind
oname	-	-	Agent name (unique)
onodeName	-	-	Group name categorized as whatap.onode
pid	-	-	Unique ID for the process
type	-	-	Platform type (application development language)

Fields

Field	Type	Unit	Description
dbc	-	Count	Number of transactions in progress for database connection
httpc	-	Count	Number of transactions in progress for HTTP call

Field	Type	Unit	Description
method	-	Count	Number of transactions during execution of the method logic
socket	-	Count	Number of transactions in progress for socket connection
sql	-	Count	Number of transactions in progress for SQL query

---

app_actx_meter

- Collects various activity metrics for active transactions including HTTP requests, SQL queries, and data transfers
- Collection interval: 5 seconds
- Statistical data: 5 minutes, 1 hour

Tags

Tag	Type	Unit	Description
alias	-	-	Agent Alias
container	-	-	Container name
containerKey	-	-	Container's key value
host_ip	-	-	Host IP Address
okindName	-	-	Group name categorized as whatap.okind
oname	-	-	Agent name
onodeName	-	-	Group name categorized as whatap.onode

Fields

Field	Type	Unit	Description
active_httpc	-	Count	Active HTTP client request count
active_sql	-	Count	Active SQL query count
data	-	byte	Data transfer volume
txcaller_group	-	Int	A constant to identify the group of the transaction caller
txcaller_oid	-	Int	A constant to identify the agent ID of the transaction caller
txcaller_unknown	-	Int	A constant to identify the unknown transaction caller

---

app_counter

- Collects various application performance metrics and statistics
- Collection interval: 5 seconds
- Statistical data: 5 minutes, 1 hour

Tags

Tag	Type	Unit	Description
alias	-	-	Alias of the application (unique)
container	-	-	Container name (unique)
containerKey	-	-	Container key (unique)
host_ip	-	-	Host IP address (unique)

Tag	Type	Unit	Description
okindName	-	-	Group name categorized as whatap.okind
oname	-	-	Agent name (unique)
onodeName	-	-	Group name categorized as `whatap.onode`
pid	-	-	Process ID
type	-	-	Platform type (application development language)

Fields

Field	Type	Unit	Description
active_tx_0	-	Count	Number of transactions in the section within 3 seconds
active_tx_3	-	Count	Number of transactions in the section between 3 seconds and 8 seconds
active_tx_8	-	Count	Number of transactions in the section over 8 seconds
active_tx_count	-	Count	Number of all transactions in progress
apdex_satisfied	-	Count	Number of APDEX satisfactions
apdex_tolerated	-	Count	Number of APDEX toleration cases
apdex_total	-	Count	Total number of APDEX transactions
arrival_rate	-	Percent	Transaction request rate
httpc_count	-	Count	Number of HTTP outbound calls
httpc_error	-	Count	Number of HTTP outbound call errors

Field	Type	Unit	Description
httpc_time	-	Millisecond(ms)	Mean time of HTTP outbound calls
metering	-	Core	Number of cores on the host where the applications are running (limit cpu for containers)
resp_time	-	Millisecond(ms)	Average response time
sql_count	-	Count	Number of completed SQL cases
sql_error	-	Count	Number of SQL errors
sql_fetch_count	-	Count	Number of SQL fetches
sql_fetch_time	-	Millisecond(ms)	Execution time of the SQL fetch
sql_time	-	Millisecond(ms)	Mean time for SQL execution
tps	-	Count	Number of transactions per second
tx_count	-	Count	Number of transactions processed
tx_dbc_time	-	Millisecond(ms)	Mean time for DB connection
tx_error	-	Count	Number of transaction errors
tx_httpc_time	-	Millisecond(ms)	Mean time of HTTP calls
tx_sql_time	-	Millisecond(ms)	Average of the sum of SQL execution times per transaction
tx_time	-	Millisecond(ms)	Transaction execution time
tx_time_sqr_sum	-	Count	Used to obtain statistical metrics such as sum of squares of transaction execution times and variance or standard deviation of transaction execution times.

app_counter_okind

- Calculates `app_counter` metrics by `whatap.okind` group
- Collection interval: 5 seconds
- Statistical data: 5 minutes, 1 hour

Tags

Tag	Type	Unit	Description
okindName	-	-	Group name categorized as whatap.okind

Fields

Field	Type	Unit	Description
active_tx_0	-	Count	Number of transactions in the section within 3 seconds
active_tx_3	-	Count	Number of transactions in the section between 3 seconds and 8 seconds
active_tx_8	-	Count	Number of transactions in the section over 8 seconds
active_tx_count	-	Count	Number of all transactions in progress
apdex	-	number	APDEX (Application Customer Satisfaction)
apdex100	-	Percent	Apdex percentage
apdex_satisfied	-	Count	Number of APDEX satisfactions
apdex_tolerated	-	Count	Number of APDEX toleration cases
apdex_total	-	Count	Total number of APDEX transactions

Field	Type	Unit	Description
arrival_rate	-	Percent	Transaction request rate
num_of_agent	-	Count	Number of active agents in the group
realtime_user	-	Count	Number of concurrent users
sql_count	-	Count	Number of completed SQL cases
sql_error	-	Count	Number of SQL errors
sql_fetch_count	-	Count	Number of SQL fetches
sql_fetch_time	-	Millisecond(ms)	Execution time of the SQL fetch
sql_time	-	Millisecond(ms)	Mean time for SQL execution
tps	-	Count	Number of transactions per second
tx_count	-	Count	Number of transactions processed
tx_dbc_time	-	Millisecond(ms)	Mean time for DB connection
tx_error	-	Count	Number of transaction errors
tx_httpc_time	-	Millisecond(ms)	Mean time of HTTP calls
tx_sql_time	-	Millisecond(ms)	Average of the sum of SQL execution times per transaction
tx_time	-	Millisecond(ms)	Transaction execution time

---

app_counter_project

- Calculates `app_counter` metrics on a project basis

- Collection interval: 5 seconds
- Statistical data: 5 minutes, 1 hour

Fields

Field	Type	Unit	Description
active_tx_0	-	Count	Number of transactions in the section within 3 seconds
active_tx_3	-	Count	Number of transactions in the section between 3 seconds and 8 seconds
active_tx_8	-	Count	Number of transactions in the section over 8 seconds
active_tx_count	-	Count	Number of all transactions in progress
apdex	-	number	APDEX (Application Customer Satisfaction)
apdex100	-	Percent	Apdex percentage
apdex_satisfied	-	Count	Number of APDEX satisfactions
apdex_tolerated	-	Count	Number of APDEX toleration cases
apdex_total	-	Count	Total number of APDEX transactions
arrival_rate	-	Percent	Transaction request rate
num_of_agent	-	Count	Number of active agents in the project
realtime_user	-	Count	Number of concurrent users
sql_count	-	Count	Number of completed SQL cases
sql_error	-	Count	Number of SQL errors
sql_fetch_count	-	Count	Number of SQL fetches

Field	Type	Unit	Description
sql_fetch_time	-	Millisecond(ms)	Execution time of the SQL fetch
sql_time	-	Millisecond(ms)	Mean time for SQL execution
tps	-	Count	Number of transactions per second
tx_count	-	Count	Number of transactions processed
tx_dbc_time	-	Millisecond(ms)	Mean time for DB connection
tx_error	-	Count	Number of transaction errors
tx_httpc_time	-	Millisecond(ms)	Mean time of HTTP calls
tx_sql_time	-	Millisecond(ms)	Average of the sum of SQL execution times per transaction
tx_time	-	Millisecond(ms)	Transaction execution time

---

app_host_resource

- Collects application server resource metrics
- Collection interval: 5 seconds
- Statistical data: 5 minutes, 1 hour

Tags

Tag	Type	Unit	Description
alias	-	-	Application's alias
container	-	-	Container name

Tag	Type	Unit	Description
containerKey	-	-	Container's key value
host_ip	-	-	Host IP Address
okindName	-	-	Group name categorized as whatap.okind
oname	-	-	Agent name
onodeName	-	-	Group name categorized as whatap.onode
pid	-	-	Unique ID of a specific process
type	-	-	Platform type (application development language)

Fields

Field	Type	Unit	Description
cpu	-	percent	Host's CPU utilization
cpu_cores	-	Positive number	Number of CPU cores of the host
cpu_irq	-	percent	CPU IRQ utilization of the host
cpu_proc	-	percent	CPU utilization of the Java process
cpu_steal	-	percent	CPU steal utilization of the host
cpu_sys	-	percent	CPU SYS utilization of the host
cpu_usr	-	percent	CPU user utilization of the host
cpu_wait	-	percent	CPU wait utilization of the host

Field	Type	Unit	Description
disk	-	percent	Root file system's disk utilization (JAVA is the disk utilization of the file system path specified by the settings)
mem	-	percent	Memory utilization of the host
swap	-	percent	SWAP utilization of the host

---

app_remote_active_stat

- Measures transaction counts by response time range for DB connections and HTTP calls
- Collection interval: 5 seconds

Tags

Tag	Type	Unit	Description
alias	-	Unique value	Application's alias
container	-	Unique value	Container name
host_ip	-	Unique value	Host IP Address
okindName	-	-	Group name categorized as whatap.okind
oname	-	Unique value	Agent name
onodeName	-	-	Group name categorized as whatap.onode

Fields

Field	Type	Unit	Description
act0	-	count	Number of active transactions with the response time of 3 seconds or less
act3	-	count	Number of active transactions with the response time between 3 seconds and 8 seconds
act8	-	count	Number of active transactions with the response time of 8 seconds or more
actx	-	count	Sum of act0, act3, and act8
count	-	count	Total number of active transactions collected during the monitoring period
error	-	count	Number of erroneous transactions
id	Int32	-	Unique ID to identify active transactions
timeSum	number	-	Total response time for active transactions
type	string	-	db(DB Connection) or httpc(HTTP Call)

---

app_user

- Collects application user metrics
- Collection interval: 10 seconds
- Statistical data: 5 minutes

Tags

Tag	Type	Unit	Description
host_ip	-	-	Host IP address (unique)
oname	-	-	Agent name (unique)
okindName	-	-	Group name categorized as whatap.okind
onodeName	-	-	Group name categorized as whatap.onode

Fields

Field	Type	Unit	Description
logbits	-	byte	User log bit information (unit of data created by user activity events)
realtime_user	-	count	Number of concurrent users

---

db_pool_detail

- Collects database connection pool detail metrics
- Collection interval: 5 seconds
- Statistical data: 5 minutes, 1 hour

Tags

Tag	Type	Unit	Description
$okind	-	-	Group name categorized as whatap.okind

Tag	Type	Unit	Description
$onode	-	-	Group name categorized as whatap.onode
oname	-	-	Agent name
pool	-	-	Type of database pool
class	-	-	Database pooling class

Fields

Field	Type	Unit	Description
act	-	count	Number of active database connections
idle	-	count	Number of idle database connections

---

event_5sec

- Collects events occurring every 5 seconds
- Collection interval: 5 seconds

Tags

Tag	Type	Unit	Description
type	-	-	Event type
no_cache	-	-	Presence of event information caching

Fields

Field	Type	Unit	Description
curtime	-	Millisecond(ms)	Event occurrence time (UNIX epoch time, millisecond)

---

event_cube_done

- Collects events when cube operations are completed
- Collection interval: 5 seconds

Tags

Tag	Type	Unit	Description
type	-	-	Event type
no_cache	-	-	Presence of event information caching

Fields

Field	Type	Unit	Description
cubetime	-	Millisecond(ms)	Cube operation completion time (UNIX epoch time, millisecond)
date	-	date	Convert cubetime to YYYYMMDD format
date9	-	date	Convert date field value to GMT+9 time zone
hhmmss	-	time	Convert cubetime to HHMMSS format
hhmmss9	-	time	Convert hhmmss field value to GMT+9 time zone

visitor

- Collects application user statistics
- Collection interval: 1 hour
- Statistical data: 1 hour

Tags

Tag	Type	Unit	Description
host_ip	-	Unique value	Host IP Address
oname	-	Unique value	Agent name
okindName	-	-	Group name categorized as whatap.okind

Fields

Field	Type	Unit	Description
logbits	-	byte	User log bit information (unit of data created by user activity events)
visit_user	-	count	Number of visitors during the search period

app_proc_counter

A category of the data collected for application process metrics

- Collection interval: 5 seconds
- Statistical data: 5 minutes, 1 hour

Tags

Field	Type	Unit	Description
alias	-	-	Alias of the application (unique)
container	-	-	Container name (unique)
containerKey	-	-	Container key (unique)
host_ip	-	-	Host IP address (unique)
okindName	-	-	Group name categorized as whatap.okind
oname	-	-	Agent name (unique)
onodeName	-	-	Group name categorized as whatap.onode
pid	-	-	Unique ID of a specific process
type	-	-	Platform type (application development language)

Fields

Field	Type	Unit	Description
cputime	-	Millisecond(ms)	Transaction CPU time
db_num_active	-	count	DB Connection Pool Active count
db_num_idle	-	count	Idle DB connection pool count
gc_count	-	count	GC occurrence count
gc_oldgen_count	-	count	Old Generation GC Count

Field	Type	Unit	Description
gc_time	-	Millisecond(ms)	GC execution time
heap_max	-	byte	Maximum size of heap memory (-Xmx value)
heap_perm	-	byte	Memory usage in the Perm area
heap_tot	-	byte	Total amount of allocated heap memory
heap_use	-	byte	Amount of heap memory in use
pending_finalization	-	count	Number of objects waiting to finalize during GC
proc_fd	-	count	Number of file descriptors open by the count process
proc_fd_max	-	count	Maximum number of file descriptors that can be opened by the process
thread_count	-	count	Total number of threads running in the JVM
thread_daemon	-	count	Number of daemon threads running in the JVM
thread_peak_count	-	count	Maximum number of threads observed while the JVM is running
thread_total_started	-	count	Total number of threads created since the start of JVM
threadpool_activeCount	-	count	Number of threads in the thread pool running

---

java_memory

It is a category of data that collects metrics about the memory usage of the JVM while the application is running.

- Collection interval: 5 seconds

- Statistical data: 5 minutes, 1 hour

Tags

Tag	Description	Remarks
host_ip	Host IP Address	Unique value
okindName	Group name categorized as whatap.okind	-
oname	Agent name	Unique value
onodeName	Group name categorized as whatap.onode	-
pid	Unique ID of a specific process	-

Fields

Field	Unit	Description	Remarks
nonheap_committed	byte	Non-heap memory size guaranteed for use by the JVM	-
nonheap_init	byte	Initial non-heap memory size that the JVM requests from the OS at startup	-
nonheap_max	byte	Maximum non-heap memory size that can be used for memory management	-
nonheap_used	byte	Non-heap memory size in use	-
objectPendingFinalizationCount	count	Number of objects waiting to finalize during GC	-

Performance Counters

The WhaTap agent collects various data related to the application performance. The data can be broadly classified into 3 categories:

- **User**: Real-time user or visitor
- **Service**: Transaction, SQL, outbound call count and response, error rate, etc.
- **Resource**: System and process resource usage

User Counter

User is a client using the monitoring system. In the client, users are counted based on the browser.

In the web system performance, user is important because it is the start point that generates the load. For user tracing, it is required to consider what criteria users are classified with and how to count them.

User types

The WhaTap agent provides various options to identify users.

- **Remote IP**

  Remote IP is used by default to identify the user. Remote IP has limitations when distinguishing actual users.

- **Cookie**

  Users are identified using the cookie. UUID is saved in the cookie named WHATAP for all connected clients.

  whatap.conf

  ```
  trace_user_using_ip=false

  # Java agent v2.2.0 or later
  wclient_using_ip=false
  ```

- **Header Key**

Users can be identified using the value passed to the HTTP header.

whatap.conf

```
user_header_ticket=USER

# Java agent v2.2.0 or later
wclient_header_ticket=USER
```

User counting

Depending on the counting method, it is used differently. Real-time users are measured to see how many users are using the system. Daily Active User is a measure for business management on how many users are interested in the service for a day.

- **Real-time user**

  The number of users is counted in the last 5 minutes. Users are counted by shifting every 5 seconds. The numbers counted on each server are merged through the HyperLogLog algorithm.

- **Daily Active User** (**DAU**)

  The users who accessed the system for a day are counted. The number of users logged in for 24 hours is calculated through HyperLogLog.

> WhaTap collects byte blocks for user data by the server to count long-term users. If the data is merged by HyperLogLog, the visitors for more than a month can be theoretically calculated.

Service Counter

Performance metrics for transactions include the number of SQLs or outbound calls used by transactions, the number of response time errors, etc.

- **Transaction Counter**

  A counter starts measuring when any transaction is performed.

- **Count**
- **Response Time**
- **Error Count**

- **Active Transaction Counter**

  It counts the number of transactions in progress.

  - **Count**
  - **Active Status**

    Progress status is fixed into five: METHOD, SQL, HTTPC, DBC, SOCKET.

    - METHOD - Calling a general function
    - SQL - Running DB SQL
    - HTTPC - Calling an HTTP outbound API (service)
    - DBC - DB connection is requrested; it is usually taken from the pool
    - SOCKET - Connecting the TCP session

- **SQL**

  It counts the status of SQL running.

  - **Count**
  - **Response Time**
  - **Error Count**
  - **Patch Count**

- **HTTP Call**

  It counts outbound HTTP calls.

  - **Count**
  - **Response Time**
  - **Error Count**

Resource Counter

It counts the server resources or resource uses inside the node processes.

- **CPU** (sys, usr, wait, steal, irq, cores)

  CPU usage %. The data is collected for each type. Steal is meaningful only in the virtual environment. The number of CPU cores is also collected.

- **Process CPU**

  CPU % used by the Java process.

- **Memory**

  System memory usage (%).

- **Swap**

  Swap memory usage (%).

- **Disk**

  Disk is the Java process utilization (%) of the current directory

- **Heap** (Total, Used, Perm)

  Total, Used, and Perm amounts of the Java heap memory. The data unit is KBytes.

- **GC** (Count, Time)

  Sum of the numbers of GCs and the times that occurred for 5 seconds.

- **ObjectPendingFinalizationCount**

  The number of objects waiting to finalize() during a GC call. As the value increases, the GC time may be delayed.

- **Thread** (start Count, Count, Daemon, Peak Count)

  It collects the numbers of threads from the start of JVM, number of running threads, number of daemons, and the maximum number of threads. You can make judgment about whether the thread pool is required.

- **DB Connection Count** (active,idle)

It collects the idle count and active count in the connection pool.

- **Tomcat Thread Pool** (active, queueSize)

  It collects the performance data once the Tomcat's executor thread pool has been set.

  XML

```
<Executor name="tomcatThreadPool" namePrefix="catalina-exec-"
maxThreads="50" minSpareThreads="2" maxQueueSize="78"/>
```

DB connection pool counter

DB connection pool is called "DataSource." It is usually provided by WAS, but in some cases, any open source pool is used. Usage data is sometimes exposed as JMX, but not always.

WhaTap provides 2 different methods. The direct inquiry is performed by a single BCI and another method uses JMX. The BCI method is used by default and as a secondary method, the JMX method is provided.

The following is the ByteCode Injection class associated with the default values for each option.

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
```

> If the BCI-related option is changed, you have to **restart** the service. Even after a class has been instrumented, it needs to be registered in the tracing table. Accordingly, redefinition does not work during execution.

Importing data sources into JMX

```
If dbcp_pool_enabled==true, tomcat_ds_enabled=false
If weblogic_pool_enabled== true, weblogic_ds_enabled=false
```

1. Check the **option auto determination** details.
2. Declare the option value of the related pool to `false`.
3. Change the DataSource-related option value to `true`. The default value is `false`.

   whatap.conf

   ```
   tomcat_ds_enabled=false
   weblogic_ds_enabled=false
   ```

ThreadPool counter

ThreadPool counters are collected through the JMX Mbean data. This metric may not be collected in some environments, although collection is fine for many use cases such as Tomcat.

Additional settings are required for collection. The related settings are as follows:

whatap.conf

```
jmx_threadpool_objectname=Catalina:type=ThreadPool,name="http-bio-8080"
// Set the name of JMX ThreadPool MBean
jmx_threadpool_atter_activecount=connectionCount
// Set the attribute representing the activecount value.
jmx_threadpool_atter_queuesize=maxThreads
// Set the attribute representing the maxThreads value.
```

Check the Mbean names such as JMX ThreadPool ObjectName and AttributeName through the JMX monitoring tool or see the Mbean documents for each WAS.

> ⓘ See the active WAS and framework-related document and check the Mbean information.

In case of using the Jconsole

- **ObjectName** : Catalina

  type : ThreadPool

  name : "http-bio-8080"

- **ActiveName** : connectionCount

  QueueName : maxThreads

Checking the result

If the setting is finished, you can view the data from the counter as follows:

Performance counter - normal status

Apdex

Apdex (Appliccation Performance Index) is a performance metric that complies with the open standards. Apdex

is based on the response time and is quantified as percentages of satisfied and accepted requests out of total requests. The Apdex graph has been added to the dashboard.

Apdex can be used as a metric for user satisfaction and have a value between 0 and 1.

(Number of satisfied cases + (Number of tolerating cases * 0.5))/Total number of requests

Status	Description
Satisfied (**S**)	No problem with the work ≤ 1.2 secs (**S** as the default)
Tolerating (**T**)	Users feel delay but working is possible ≤ 4.8 secs (**S** * 4)
Frustrated (**F**)	Working is impossible > 4.8 secs (exceeds T and error)

- **whatap.apdex_time** millisecond

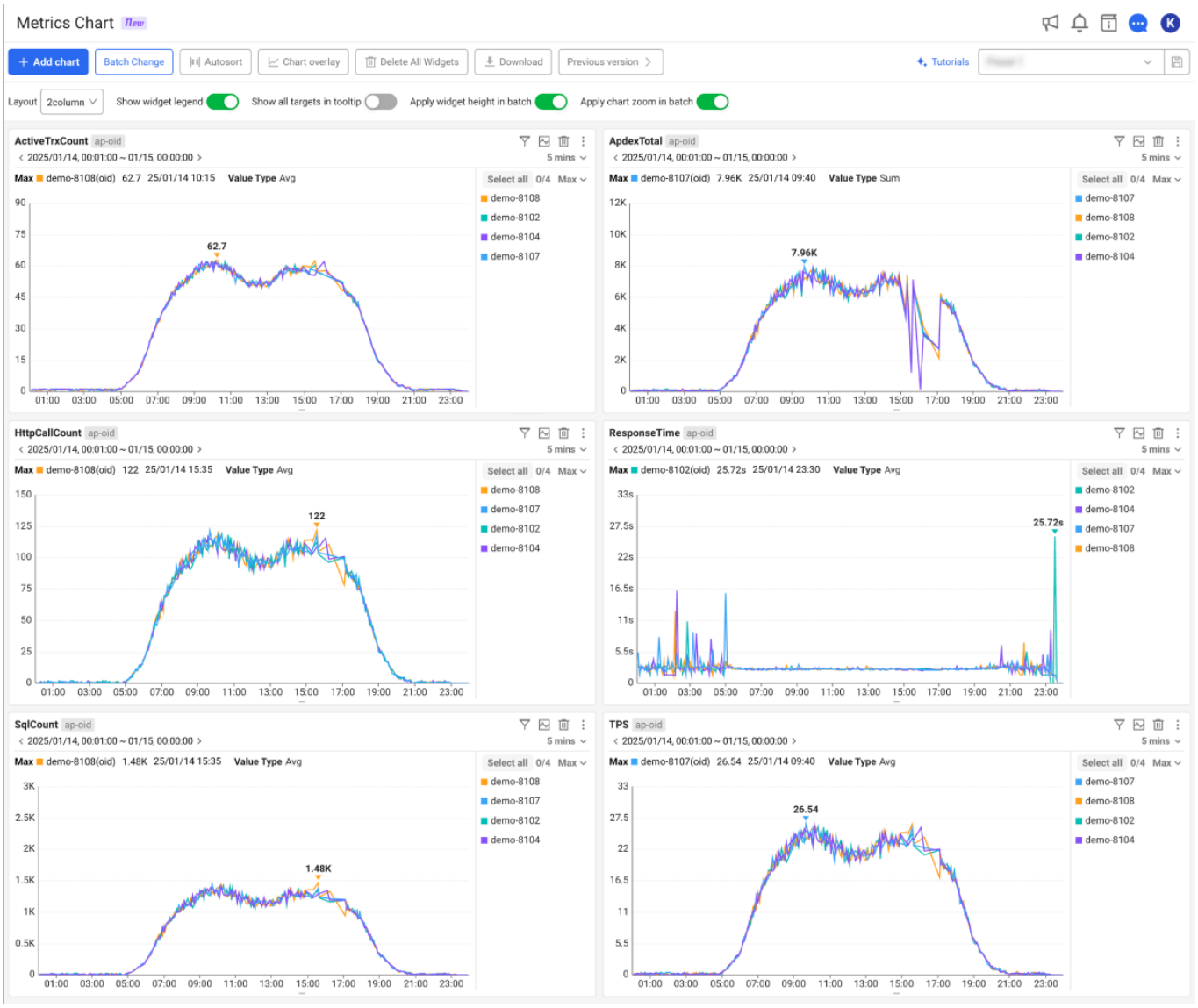
  Default `1200`

  The default value (**S**) can be modified in Agent CONFIG.

Metric Chart V2

Home > Select Project > **Analysis** > **Metrics Chart** *New*

The **Metrics Chart** *New* menu is a core feature of the WhaTap monitoring solution, which provides the data collected from various monitoring targets in a visualized time-series chart. This allows users to effectively monitor the system performance, quickly identify anomalies, and gain insights for resolving issues.



- You can visually check various metrics via time series charts, and analyze multiple metrics upon failure at a time.
- You can select a specific time range to view past data and analyze the performance at that period.
- You can filter data based on specific time zones or conditions for more accurate analysis.
- Users can place their desired charts on the screen and freely adjust the layout.
- Users can arrange desired metrics and save them as presets for loading at any time.

> **Difference from the Metrics Chart menu**
>
> The following are the changes since the Service 2.8.0 update. To check the changes on the screen after the update, select **Tutorials** on the upper right.
>
> - User experience (UX) has been improved so that users can set the time, target, and interval options, and then specify the metrics to monitor and add charts.
> - You can change the time range of a chart view without deleting the widgets placed on the dashboard.
> - It provides a batch **Batch Change** feature that allows you to apply options to all widgets placed on the dashboard at once.
> - The preset feature allows you to save and load the widgets placed on the dashboard.
> - You can adjust the layout of the dashboard and the functionality of the widgets through various options provided.

Adding a chart widget

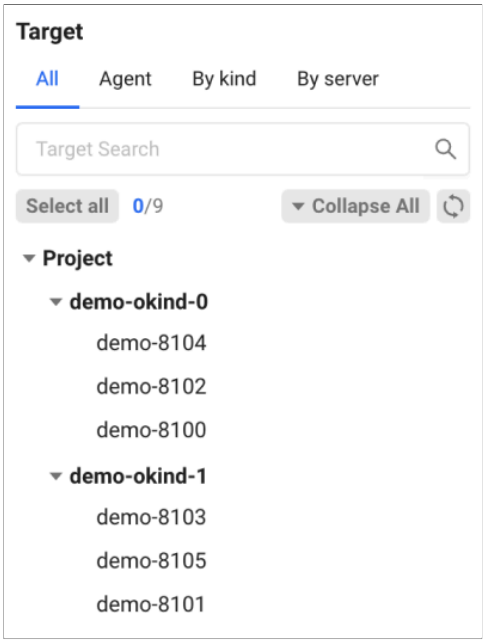
To add a metric as a widget to the dashboard, select **Add chart** on the upper left.

Setting the lookup time

In **Time**, set the date and time to view the metric.

Selecting a target

Select a desired target from the agents in the project. You can select an individual agent or a group classified by the agent configuration.

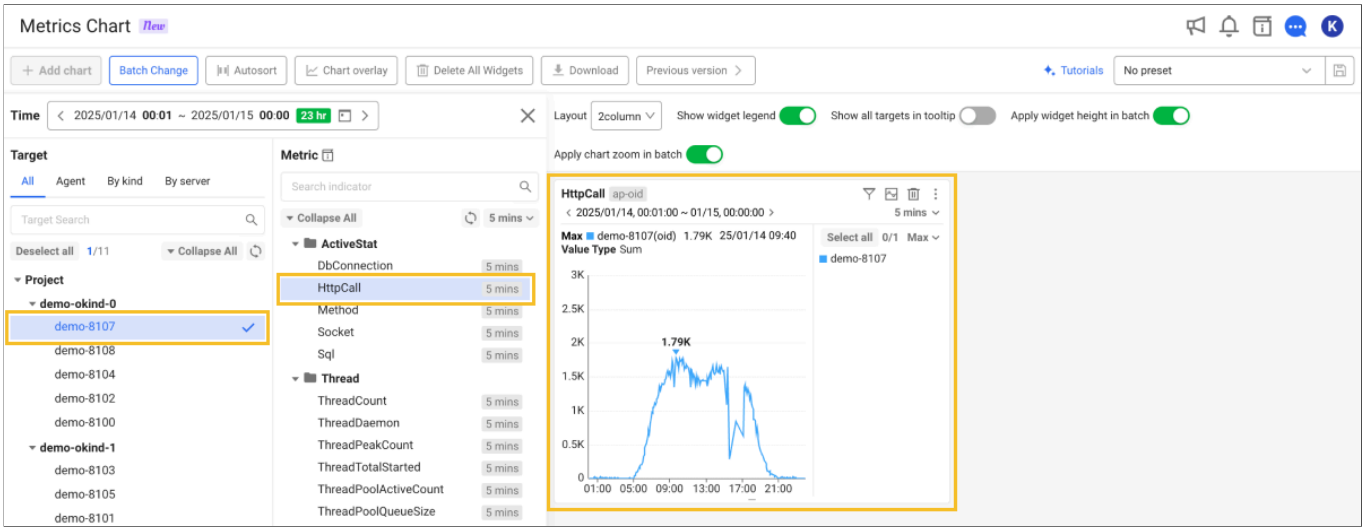


- **Agent**: You can select an individual agent or all agents.
- **By kind**: Agents can be monitored in each group classified by `whatap.okind` in the agent configuration.
- **By server**: Agents can be monitored in each group classified by `whatap.onode` in the agent configuration.

> ⓘ
> - Depending on the set time and date, available targets may change.
> - You can select or deselect targets by clicking in the agent list. You can also select multiple agents at once by using the `Shift` key.
> - By clicking **Select all** or **Deselect all**, you can select or deselect related sub-targets at once.
> - If a desired agent does not appear in the list, select **Refresh**.
> - By selecting **Collapse All** or **Expand All**, you can hide or expand the list by group.

Selecting a metric

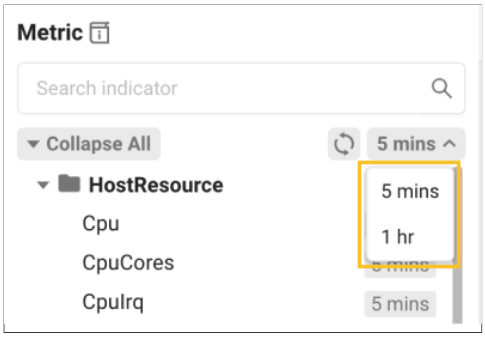
Select a target agent to monitor. Then the list that can be selected on the **Metric** panel appears. Select a desired metric in the list.



A chart widget for the selected metric is placed on the dashboard on the right. You can add chart widgets by selecting additional metrics. To enlarge the chart area, select ✕ on the right of the **Time** option. The **Target** and **Metric** panels disappear, leaving only the chart area visible on the screen.

- If you enter a string in the **Search indicator** field, you can quickly search for a desired metric.
- You can also search by metric category and field name. For more information about the categories and fields, see the following.
- By selecting **Collapse All** or **Expand All**, you can hide or expand the metrics list by category.
- If a desired metric does not appear in the list, select **Refresh**.

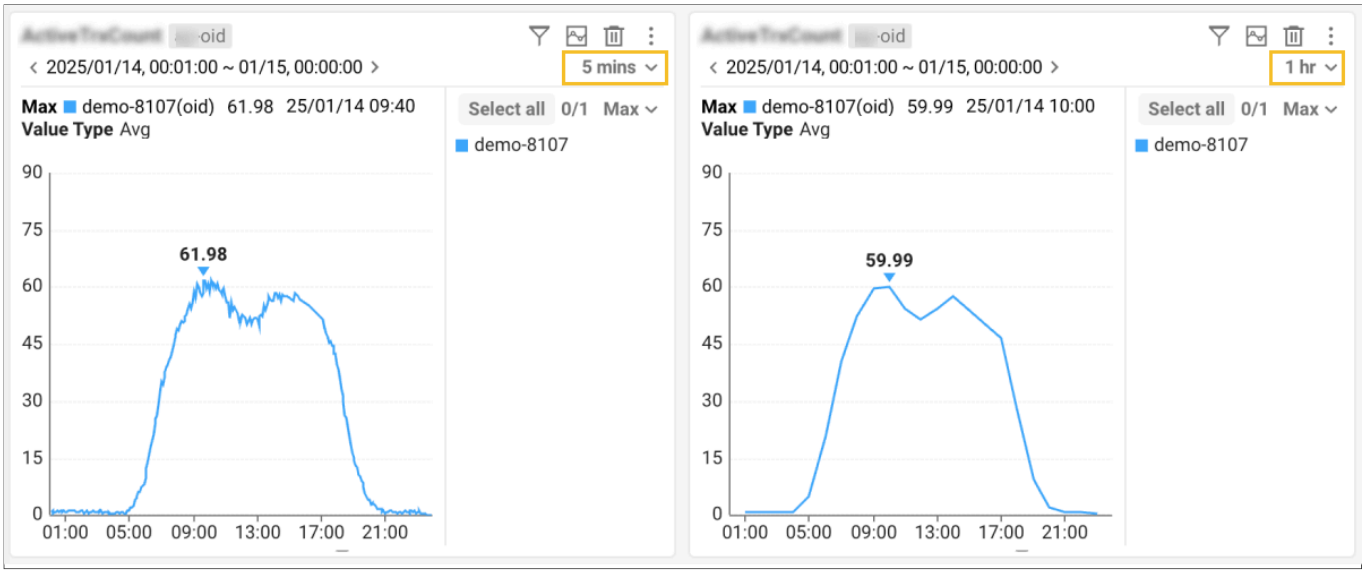
Setting the interval



You can set intervals for each chart, allowing you to adjust the aggregation unit of data to visualize at different time intervals. If you select on the upper right of the **Metric** panel, you can select an interval option.

Each metric has a different default interval time, and the interval time changes depending on the query time range.

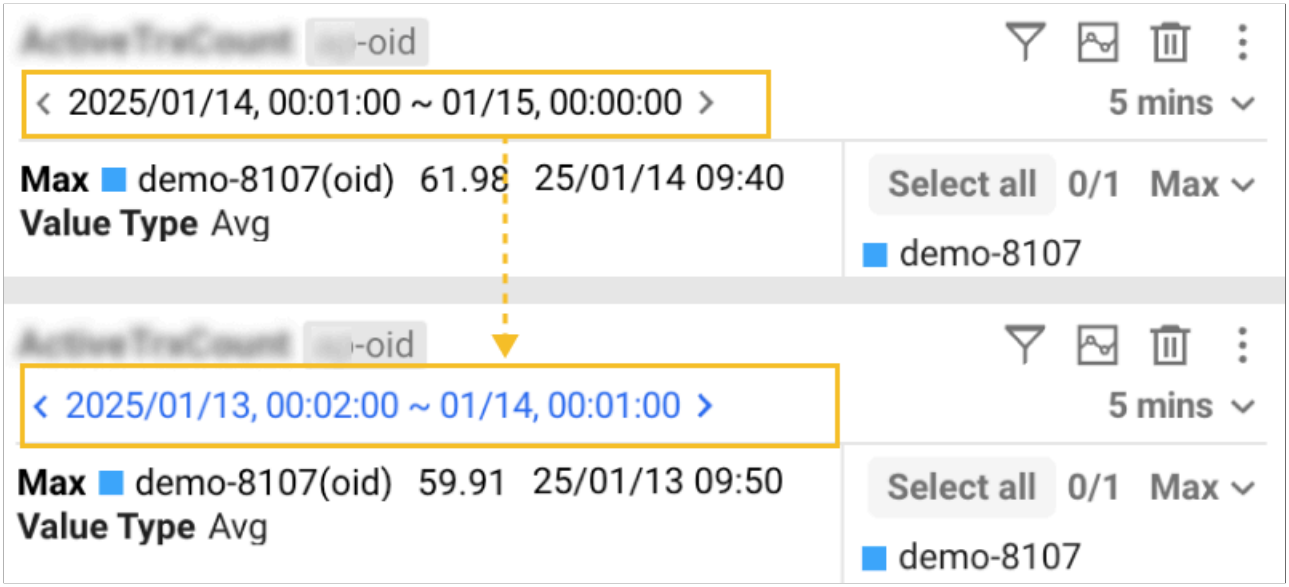
For example, the **5 min** interval displays average values for 5-minute intervals, and the **1 hour** interval displays average values for every one hour.



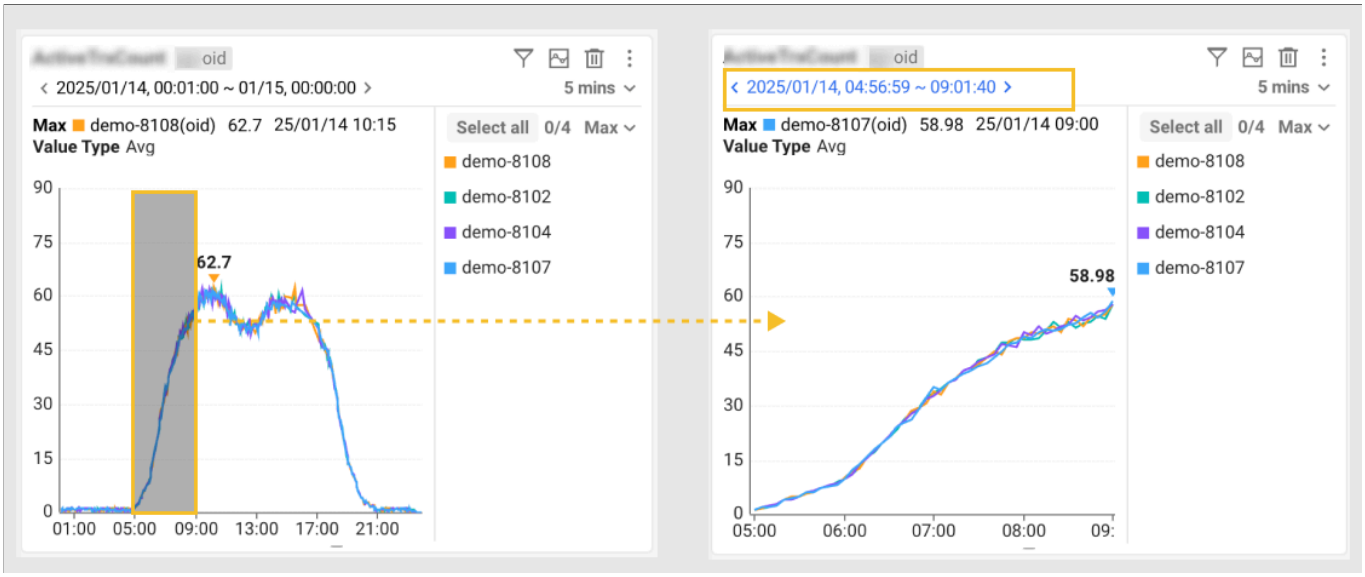
By adjusting the interval option, you can analyze the data from different perspectives, from detailed data over a short time to overall trend over a long time.

Changing the query time

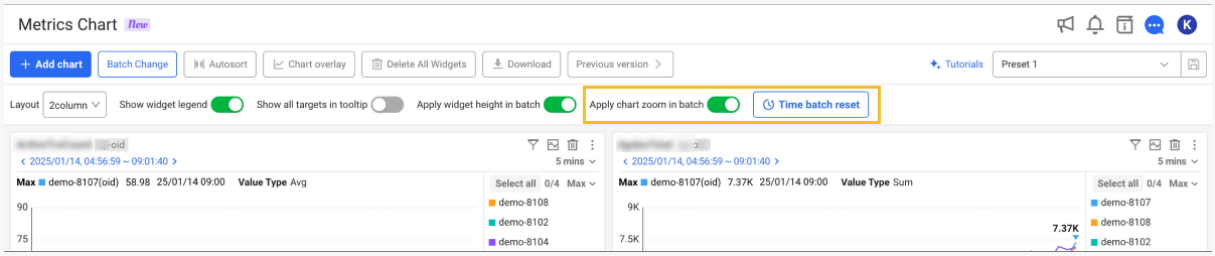
To change the chart query time range for an individual widget, select 〈 or 〉 at the top of the widget. The time change range is the same as the time reference set in the **Time** option. For example, if the **Time** option is set to **1 day** and you select 〈 or 〉, you can change the time to a daily interval.



Alternatively, by dragging a specific area of the chart, you can change the time according to the dragged range for a view.



- You can change the time for all widgets by changing for each widget with the **Apply chart zoom in batch** option enabled at the top of the screen.



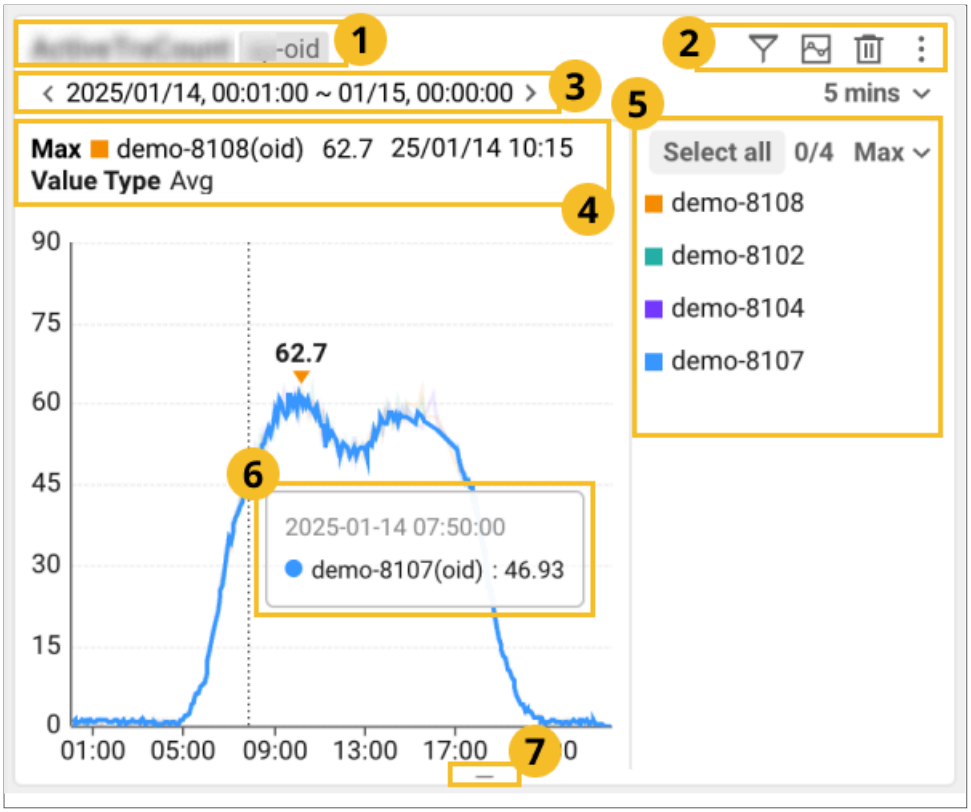
- To change the query time range and then change it back to a recent time, select .

  If the **Apply chart zoom in batch** option is enabled at the top of the screen, the **Time batch reset** button appears. By selecting this button, you can reset the time for all widgets.

By using the **Batch Change** feature, you can change the time for all widgets and change the monitoring target and interval. For more information, see the following.

Chart details

See details for each item displayed in the chart widget.



- 1 **Metric name and category**: It displays the metric name of the chart widget. On the right of the metric, the categories of the query targets are displayed in tag format. The category name may differ depending on the product.
  - `*oid` : It is displayed when an individual agent is selected.
  - `project` : It is displayed when the target is selected on a project basis.
  - `okind` : It is displayed when a group classified as `whatap.okind` is selected in the agent configuration.
  - `onode` : It is displayed when a group classified as `whatap.onode` is selected in the agent configuration.
- 2 **Widget option**: Option menu of the chart widget. For more information, see the following. For more information about the interval options, see the following.
- 3 **Time**: It displays the time range for which data in the chart is viewed. By selecting < or >, you can adjust

the time range for a query.

- 4 **Data**: You can check the maximum value (Max) of the data viewed from the monitoring targets and the data merge method (Value Type).
- 5 **Legend**: It displays the monitoring targets as a legend. For more information about the display of legend, see the following.
- 6 **Chart**: It displays the viewed data in a time series graph. If you hover your mouse pointer over the graph, you can see detailed information through a tooltip. By dragging a specific area of the chart, you can change the time according to the dragged range for a view.
- 7 **Resize**: You can adjust the size by dragging the center of the bottom of the chart widget.

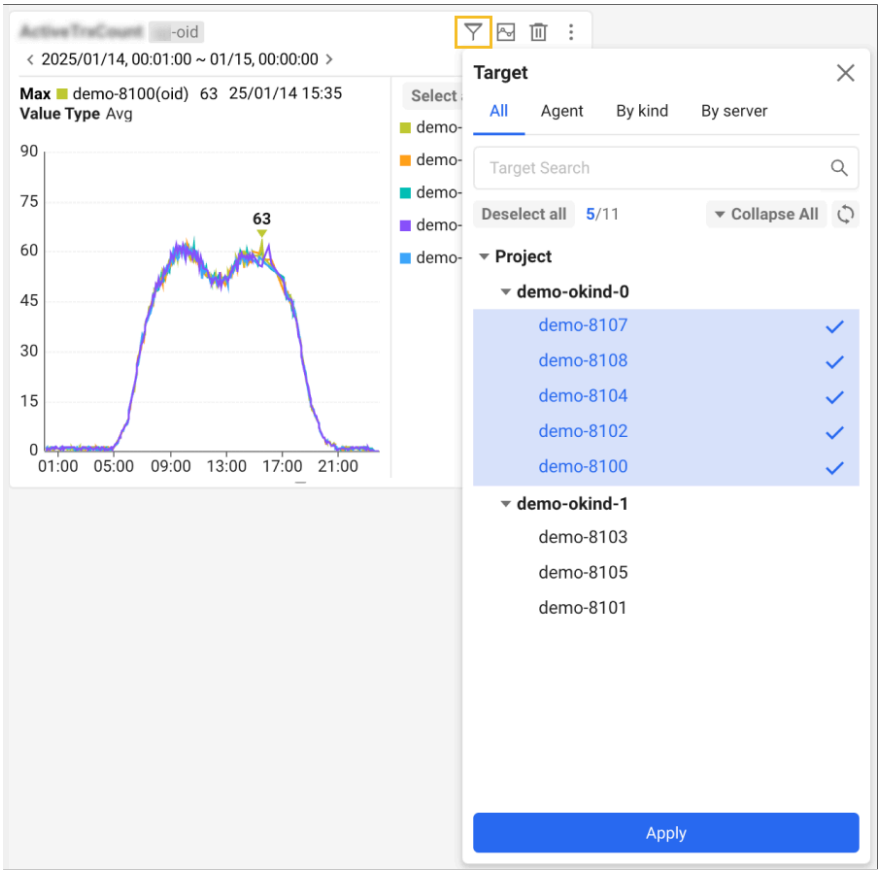
Using the chart widget options

Various features can be used through the option buttons on the upper right of the chart widget.

Filtering targets

You can add or remove monitoring targets to/from the chart widget added to the screen.

1. If you select , a pop-up menu appears for you to select monitoring targets.



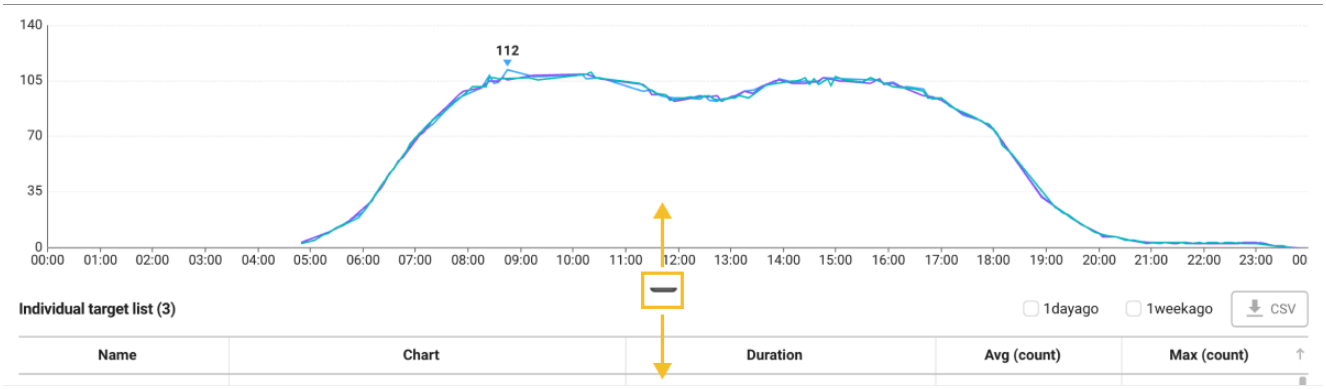
2. Select the desired targets according to the instructions on the screen.
3. Select **Apply**.

Viewing the details

To check the detailed information about the chart data in a wider screen, select . The **View details** window appears. You can view charts, averages, and maximum values for individual targets.



- **1dayago**: You can compare the data with that from the previous day.
- **1weekago**: You can compare the data with that of the previous week.
- **CSV**: The viewed data can be downloaded as a CSV file.
- You can resize the chart by dragging the icon at the bottom of the graph.

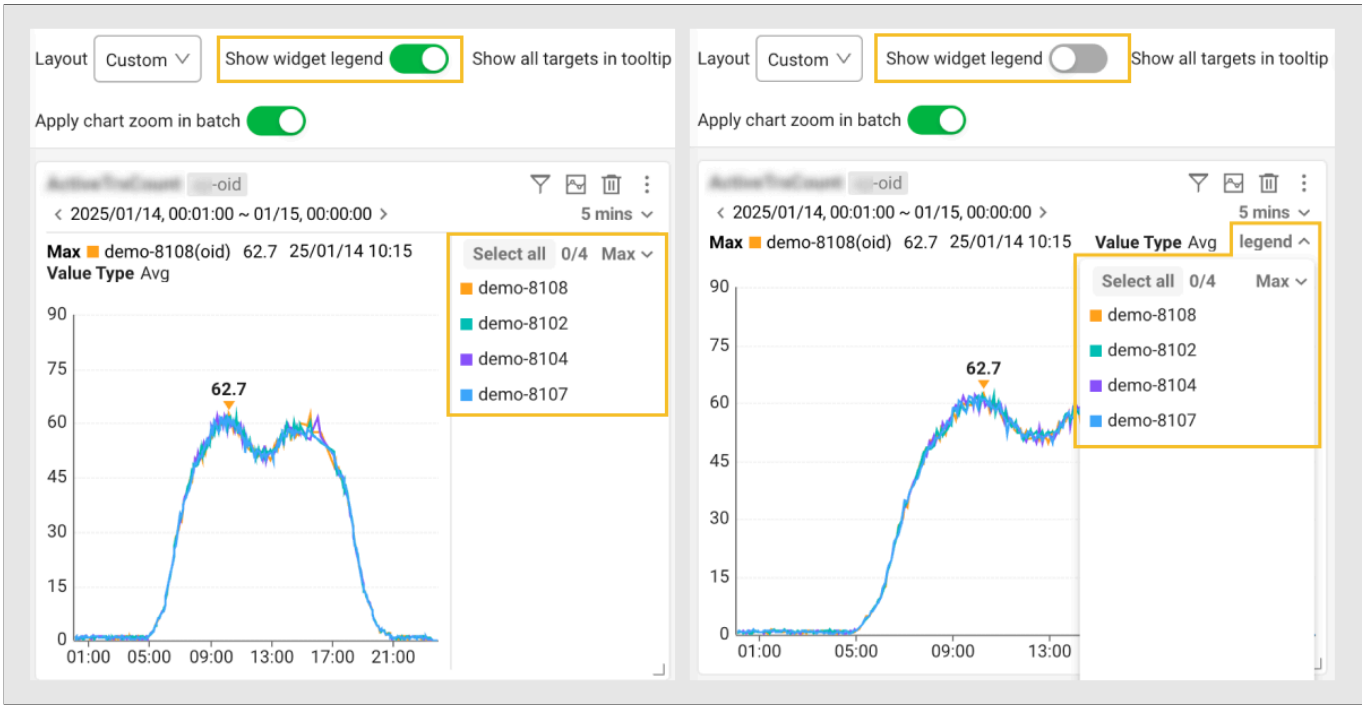


Removing the chart widget

To remove any chart widgets placed on the screen, select .

Using the legend

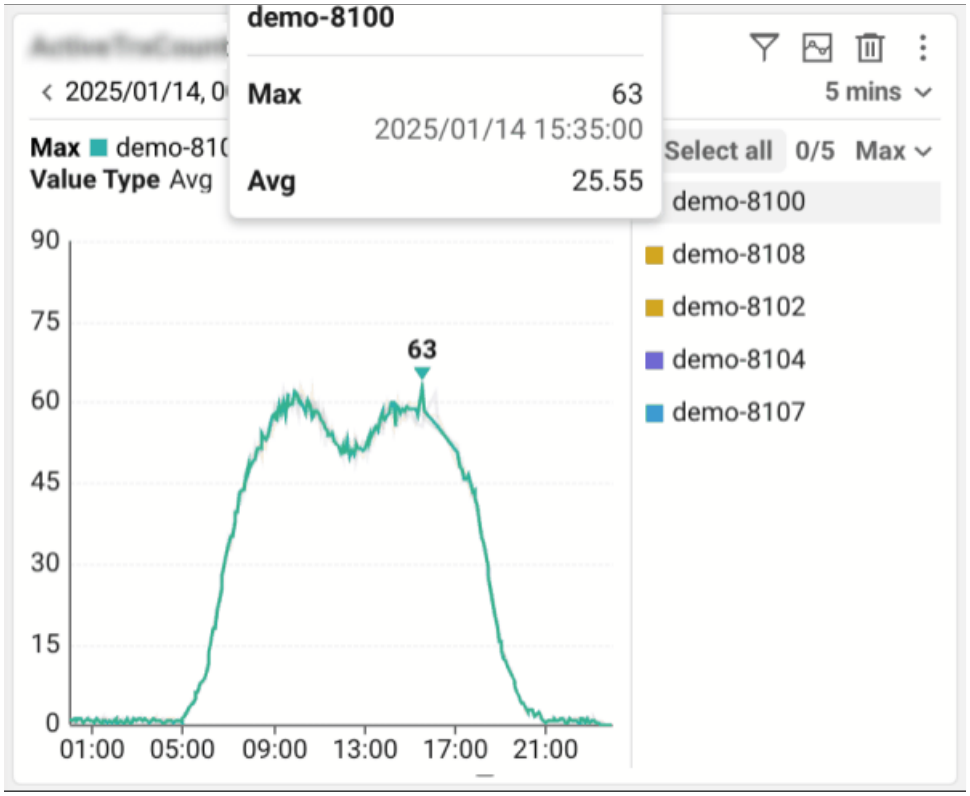
You can check the monitoring targets displayed on the chart through the legend and then sort or filter them by desired criteria. You can show or hide the legend on the right of the widget's chart by turning on or off the **Show widget legend** option at the top of the screen. With the option turned off, you can check and select the monitoring targets by clicking **legend**.



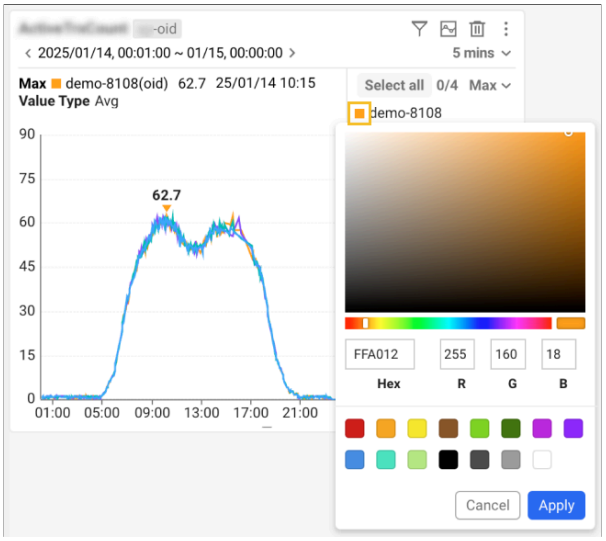
- You can select or unselect monitoring targets by selecting **Select all**/**Deselect all**.
- In the legend list, you can sort monitored objects by name, maximum value, or average value.



- If you hover the mouse over each item in the legend list, the highlighted graph for the monitoring targets appears and a tooltip indicating the specific time at which the maximum value is recorded. Alternatively, you can filter by selecting a target from the legend list.



Changing the legend color



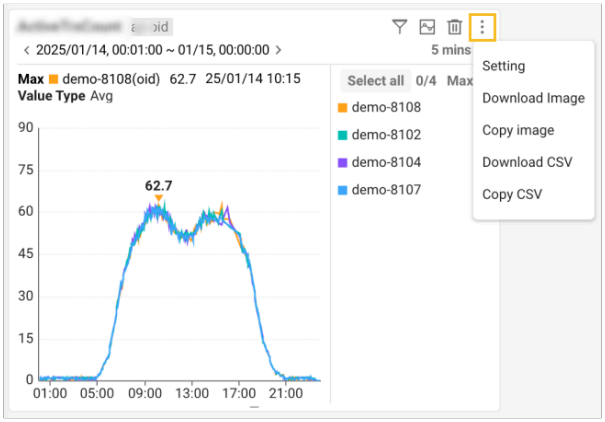
You can change the graph color corresponding to a monitoring target. In the legend list, select the square box

on the left of the target to change its color. Select a desired color and then click **Apply**.

> ⓘ The set colors are not saved via the preset feature. This feature is to be provided in a future update.

Using additional options

If you select ⋮ on the upper right of the chart widget, you can select the following additional options.



- **Setting**
  - **Use Y axis automatically**: The size of the Y-axis can be automatically adjusted within the searched range.
  - **Show maximum value**: You can display the maximum value in the graph area of the chart.
- **Download Image**: You can download the chart as an image file.
- **Copy image**: You can copy the chart to the clipboard as an image format.
- **Download CSV**: You can download the data searched from the chart as a CSV file.
- **Copy CSV**: You can copy the data viewed in the chart to the clipboard in a CSV format.

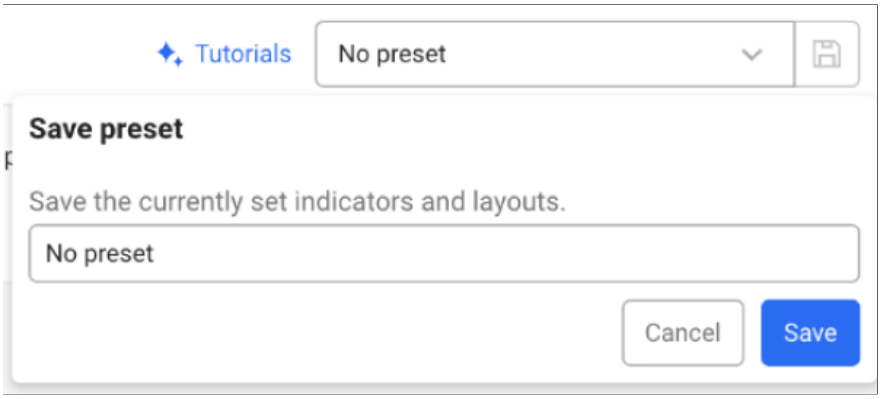
Setting presets

You can save and load the settings and layouts for widgets on the dashboard. You can create a new preset by adjusting its size and placing it on the desired position.

You can save or modify the presets with the **Edit project** role. For more information about the member roles, see the following.

Creating a new preset

1. Place the widgets in the desired format on the dashboard. You can also resize and place only the widgets to be frequently checked.
2. On the upper right of the screen, select .
3. Enter a new preset name.



4. Select **Save**.

You can see the newly saved preset from the preset list.

- If any changes are made to the newly created preset, save the preset again. Select and then save the preset with the same name. Any changes are overwritten in the existing preset.
- If you go to another menu without saving changes on the dashboard, the changes are not saved.
- Presets are saved on a project basis and can be shared with other users.
- The targets saved in the preset are the layout of the dashboard, the widgets placed on the dashboard, and the options (**Target**, **Interval**) modified in the widgets. The **Time** option at the top of the screen is not saved.
- To change the widget query time after loading a preset, use the **Batch Change** feature. For more information, see the following.

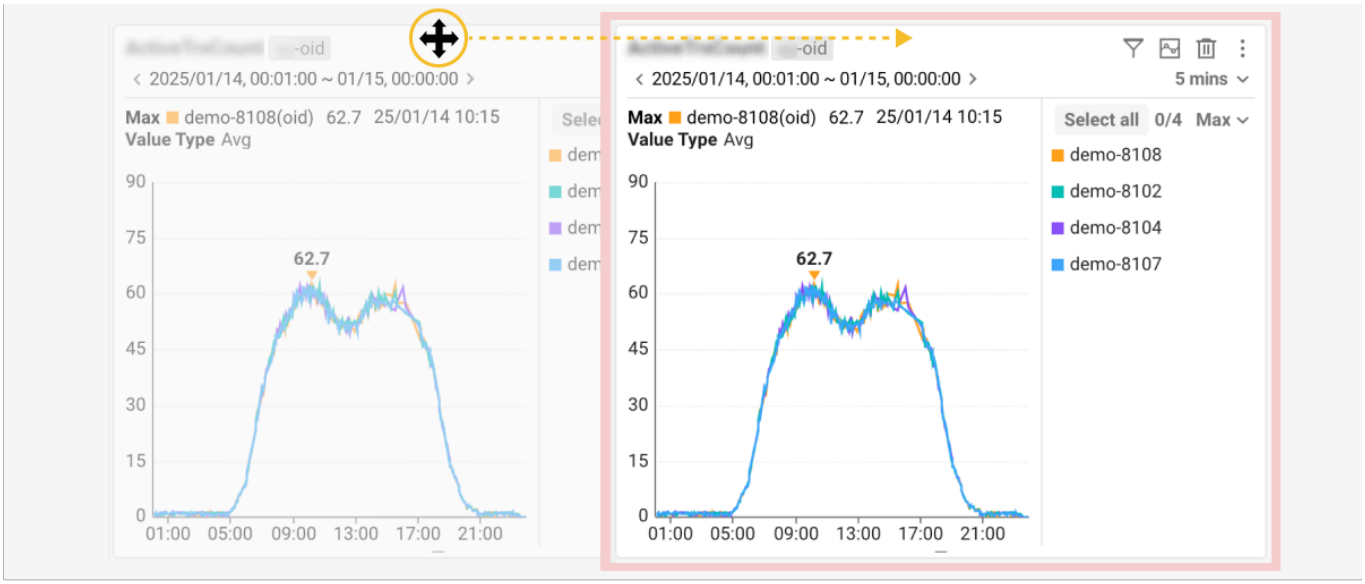
Deleting presets

If you have unused presets, you can delete them from the preset list. Select ✕ on the right of the item to delete from the preset list.

Editing a dashboard widget

The widgets placed on the dashboard can be placed and resized to the user's desired location.

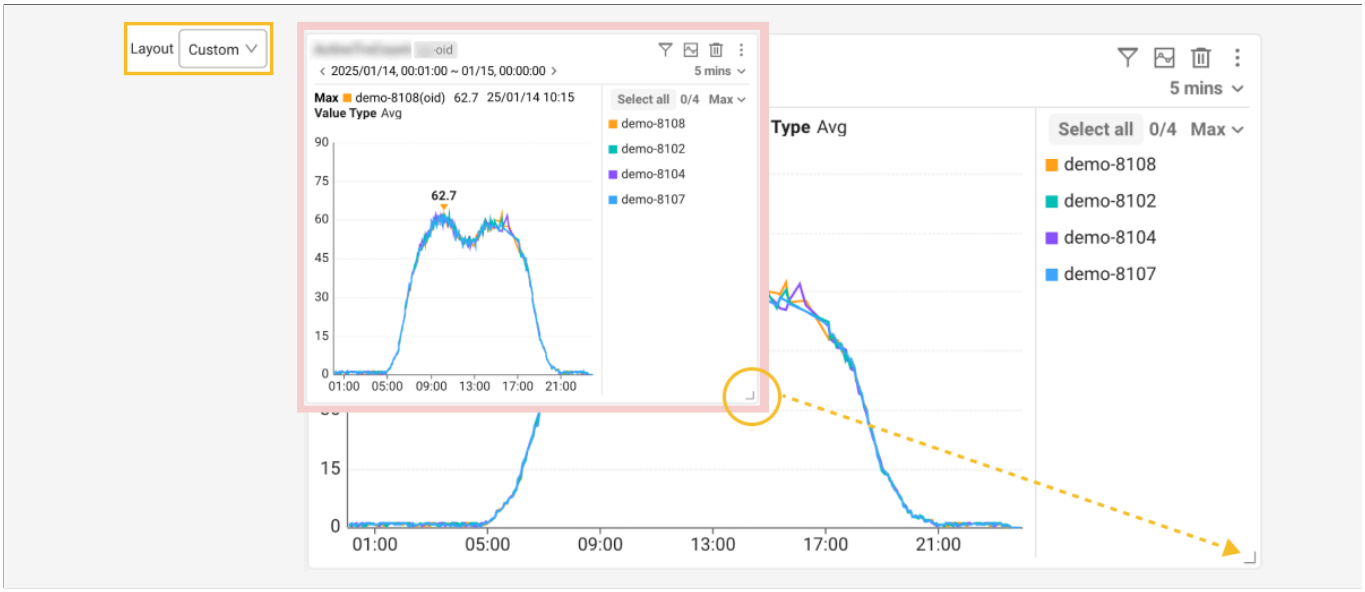
Moving a widget



When you move the top part of the widget by using the mouse pointer, it changes to the ✢ shape. At this time, you can move the widget by dragging with your mouse to the desired position.

Resizing a widget

If **Layout** among the dashboard options is set to **Custom**, you can resize the widget.



Click and hold the mouse on the ⌟ element at the lower right of the widget, and then drag it to the desired size.

> ⓘ If **Layout** is not set to **Custom**, you can adjust only the height of the widget. Resize the widget with the bottom center of the widget clicked with your mouse.

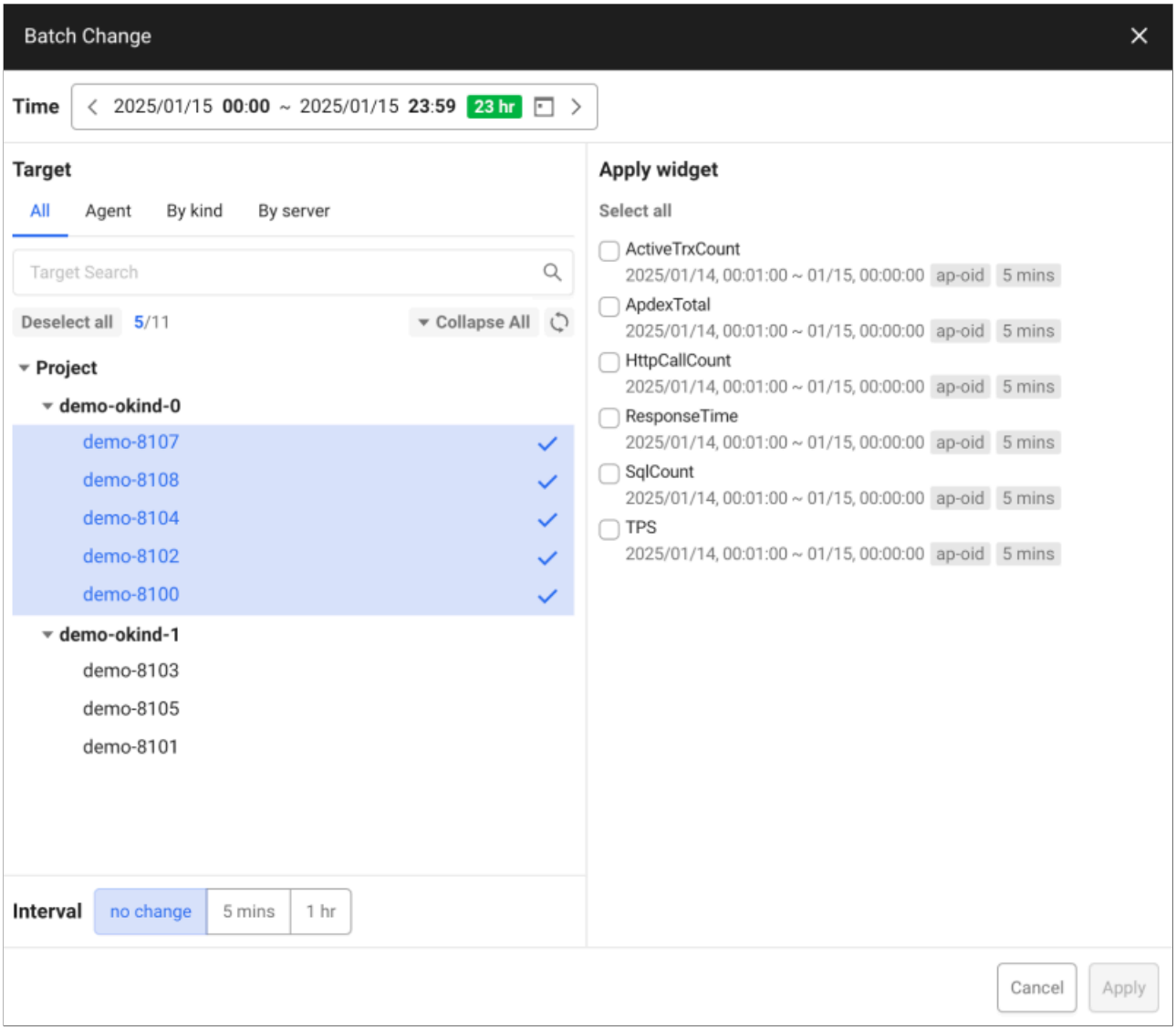
Using the dashboard options

You can apply various settings to the chart widget at once using the Option menu located at the top of the screen. You can also customize the dashboard layout to your desired format.



Batch update of the chart widget options

It provides the feature to apply detailed options to the chart widgets on the screen at once. On the upper left of the screen, select **Batch Change**. If the **Batch Change** window appears, set the desired options and then select **Apply**.



- **Time**: You can set the data query range for the chart.
- **Target**: Select a monitoring target. You can select an individual agent or a group classified by the agent configuration.
  - **Agent**: You can select an individual agent or all agents.
  - **By kind**: Agents can be monitored in each group classified by `whatap.okind` in the agent configuration.
  - **By server**: Agents can be monitored in each group classified by `whatap.onode` in the agent configuration.
- **Interval**: You can adjust the data aggregation unit to various time intervals.

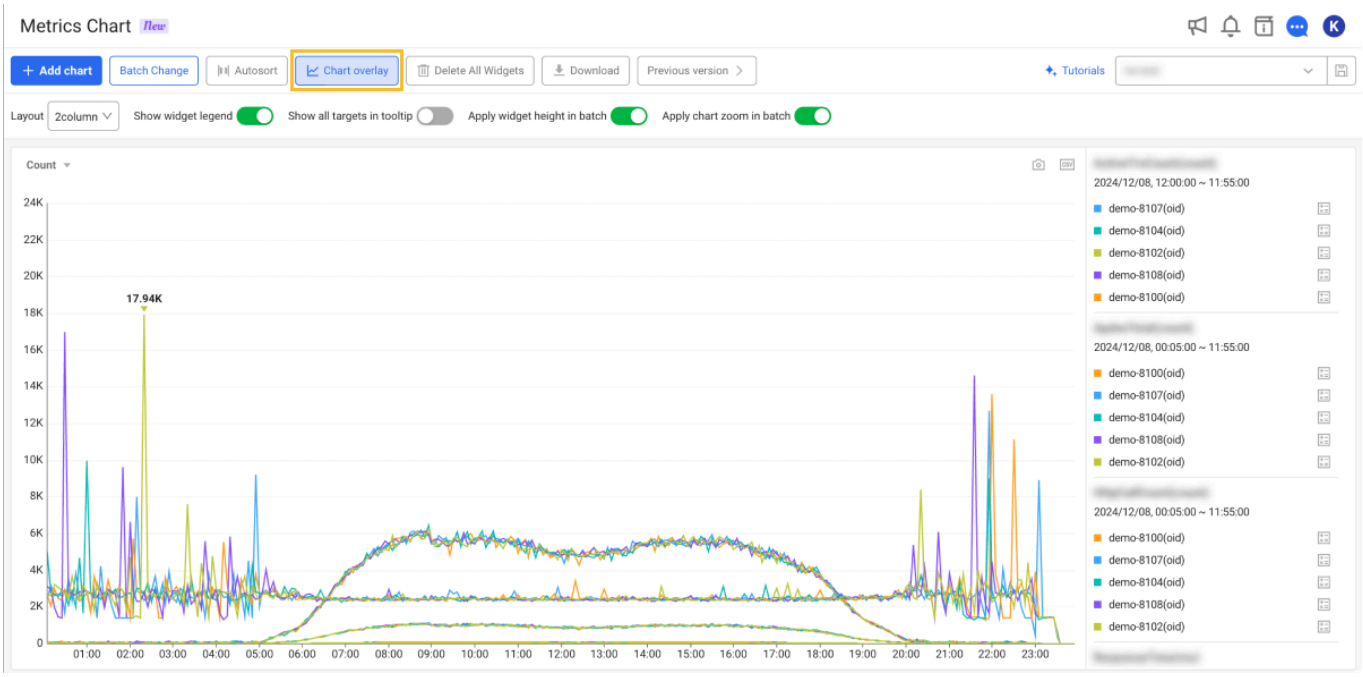
- **Apply widget**: Select a chart widget to apply the options to change. You can apply changed options to the selected widget at once.

Auto-aligning the dashboard

To auto-align the widgets placed on the dashboard, select **Autosort**. It auto-aligns widgets in order of their width and height, from smallest to largest. If the widths are the same, it aligns them in order of height, from smallest to largest.

Comparing multiple charts

This feature allows you to view data from multiple charts placed on the dashboard in a single chart. This feature allows you to compare multiple metrics from the same time period with others. Select **Chart overlay**. You can place overlay charts on the screen.



- If you select **Count** on the upper left of the overlay chart, a list appears where you can change the data criteria for the x-axis chart. You can compare the data of the chart by selecting a desired item.
- : You can download the chart as an image file based on the current view.
- : You can download the chart as a CSV file based on the current view.

- In the right section, you can see a list of agents for each metric. The colors of individual agents match the graph colors on the chart.

Batch deletion of placed widgets

You can delete multiple widgets placed on the dashboard all at once. Select **Delete All Widgets**. When the confirmation message appears, select **Delete**.

Downloading as a CSV file

You can download data from the widgets placed on the dashboard as a CSV file. Select **Download** > **Download CSV**. You can check the data of metrics for each time zone.

Downloading as an image file

You can download all the widgets placed on the dashboard as a single image file. Select **Download** > **Download Image**. You can save a captured image of the dashboard screen as is.

Using the previous version

To use the **Metrics Chart** menu of the previous version, select **Previous version**. Alternatively, you can see in the **Sitemap** > **Analysis** > **Metrics Chart** *Old* path.

> For more information about **Metrics Chart** *Old*, see the following.

Using other options

Other options below allow you to adjust the layout of the dashboard or easily and conveniently batch set the applied options to the widgets.



- **Layout**: You can set the number of columns in the dashboard (or the number of widgets that can be placed in

a row). You can set up to 4 columns. If **Custom** is selected, widgets can be placed regardless of the number of columns.

> ⓘ In **Custom**, you can freely adjust the width and height of the widget. If you select another option, you can only adjust the height of the widget.

- **Show widget legend**: Turning the option on or off shows or hides the legend on the right of the widget's chart. With the option turned off, you can check and select the monitoring targets by clicking **legend**. For more information about the use of legend, see the following.
- **Show all targets in tooltip**: When you hover your mouse pointer over the chart, you can see the value for the point on the graph through a tooltip. Turn on the option to view data for all targets in the time zone where the mouse pointer is located through the tooltip.
- **Apply widget height in batch**: Enabling this option forces all widgets to have the same height.
- **Apply chart zoom in batch**: When changing the data query time of an individual chart widget, turn on this option to also change the chart times for other widgets.

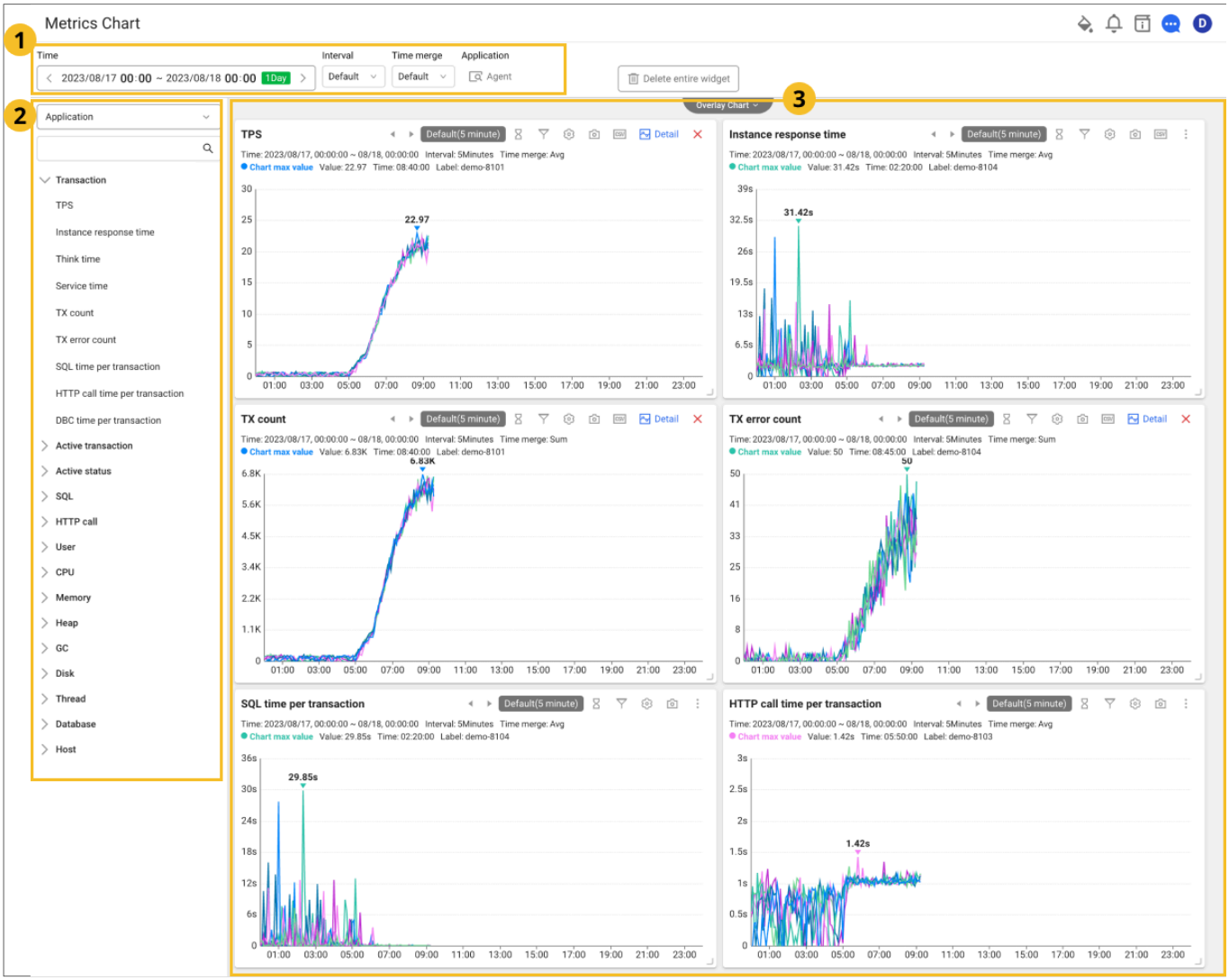
Using the tutorial

When you select **Tutorials** on the upper right of the screen, you can check the changes after the Service 2.8.0 update.

Metric Chart

Sitemap > **Analysis** > **Metrics Chart** *Old*

In **Metrics Chart**, you can view the metrics data collected from the monitoring targets in a chart as follows. Specifying the time and metrics is required.



Options at the top

In the 1 area, you can specify the time range of the chart and the agent to be monitored through the options at the top of **Metrics Chart**.

- **Time**: You can specify the start and end of the X-axis as the total range in time.

  > For more information on how to use the **Time** option.

- **Interval**: You can specify the X-axis data interval as the time interval.
- **Time merge**: As one of the data merging methods, you can merge data within the time specified by intervals.

  For example, **Avg** indicates the average value of data within 1 hour.

- **Agent**: You can specify the agents to query. If not specified, all items are queried.

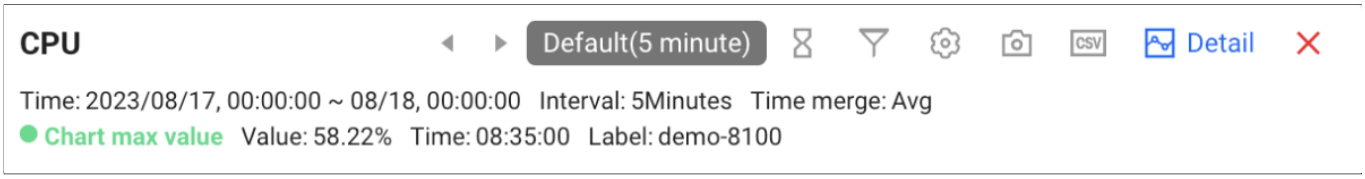
Metrics list

The 2 area is a list of metrics to query for options. First select a **Category**. After querying the metrics under the selected **Category**, select a desired metric. If you select the **Category** and metric, Based on the data within the time range specified in the top menu of the area 1, you can see the chart widgets in the 3 area.

> **Data Merge**
>
> Data merge provides the following methods: **Object Merge** and **Time merge**.
>
> - **Time merge** merges data with the same field values at regular intervals in the original data.
> - **Object Merge** merges data with matching tags among data with different field values.

Chart widget

On the top left of 3, you can see the metric name. On the top right of the chart widget, you can see the following options:



- Time movement: ◂ ▸ Using the left arrow button or right arrow button, it can be moved by -1 or +1 as much as the selected time range.

  For example, if the time range is between 00:00 on February 13 and 00:00 on February 14, when you select the ◂ left arrow button, you can see the data within the time range between 00:00 on February 12 and 00:00 on February 13.

- Interval/Time Merge: In the menu at the top of 1, you can modify the specified interval and time merge.
- By selecting the monitoring target: ▽, you can specify the monitoring targets. If not selected, it searches for all.
- Time comparison: If you select ⧖, you can compare the current trend of the same metric with that of the previous time zone.
- By selecting the snapshot: 📷 icon, you can snapshot charts except for widget options.
- By selecting the CSV: CSV icon, you can download the plot data on the chart as a CSV file.
- **Detail**: By selecting the 🖼 icon, the detailed view can be seen. When there are many monitoring targets, you can see the metrics trends for each monitoring target individually.

> ⓘ If you do not see any options at the top of the metrics chart widget, select the ⋮ icon.

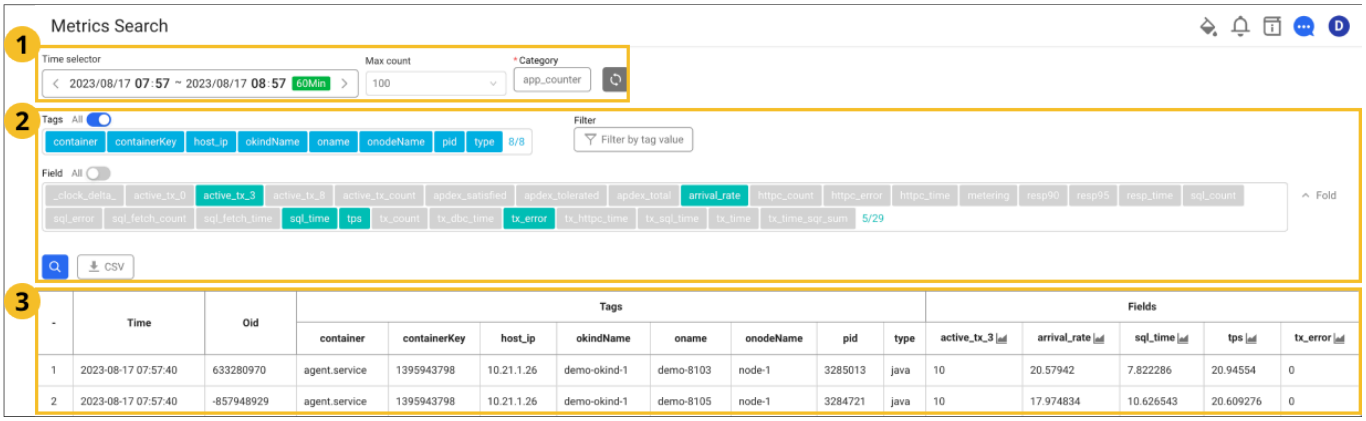
Deleting a widget

To delete all widgets placed on the screen, select 🗑 **Delete All Widgets** on the screen.

Metric Search

Home > Select Project > **Analysis** > **Metrics Search**

You can query specific metrics based on tags in **Metrics Search**.



1 Time and category selection

In the 1 area, you can specify the **Time selector**, **Max count**, and **Category** for the collected metrics. **Time selector** and **Category** must have been specified.

- **Time selector**: You can query by specifying the time when metrics were collected. The default value is `1 hour`. In addition to the lookup time provided as the default option, you can specify the date and time after selecting the **Time selector** tab.

  > ⓘ For more information on how to use the **Time selector** option.

- **Max count**: You can specify the maximum number of metrics to search in the 3 list. You can set the count up to `10`, `50`, `100`, `200`, `300`, `1000`, `2000`, and `3000`.
- **Category**: It is a classification unit for the related metrics. You can set the desired **Category** after selecting the **Category** tab.
- **Refresh categories, tags, and field options**: **Category**, **Tags**, and **Field** options can be fetched.

2 Tag and field selection

In the 2 area, select **Tags** and **Field**. The default setting is "Select All," unless specified separately by user.

- **Tags**: Unique data that can identify the collected targets.
- **Field**: Metrics collected from the monitored targets.
- **Filter**: Select **Filter by tag value** and then set the tag value for filtering.

  For example, you can search filtered data by setting the value of `oname` to `demo-8101`.

- **Search**: If you set the conditions and then select **Search**, you can search the source data of the metric in the 3 area.
- CSV : You can download the source data of the metric as a CSV file.

3 Metrics table

Because the metrics to be collected cannot be specified in advance, it is important to check the source data for all metrics being collected. After setting the desired conditions in the condition area above, you can search the original data of the metric in a table format in the 3 area. Columns in the table are changed when you specify the conditions for **Tags** and **Field**.

> - To search metrics, **Time selector** and **Category** must have been specified.
> - To search metrics, **Tags** and **Field** are optional.

Metric Anomaly Detection

Home > Select Project > **Sitemap**> **Metrics Anomaly Detection**

You can compare the patterns of various metrics with the expected patterns learned by the AI. Anomaly detections that deviate from the expected patterns can be viewed through the graph chart. You can check recurring patterns based on the past data and use them to predict future metrics values.

- The widgets placed on the screen are not saved and reset when you go to another menu.
- It is similar to **Analysis** > **Metrics Chart** except for the pattern and anomaly display.
- The alert, **Anomaly detection** is based on the **Metric anomaly detection**. For more information about the **Anomaly detection** alert setting, see the following.

Checking the widgets



- 1 The bright color areas of the graph are the estimated patterns analyzed by the AI.
- 2 The blue graph is the metrics trend of the project.
- If it deviates from the expected pattern analyzed by the AI, it displays dots on the graph in Orange and Red steps. The values that are far from the expected pattern range appear in red.

Placing the widgets



1. In 1 **Time**, set a desired time interval. You can set the interval up to 1 month.
2. In 2 **Filter**, select a range of the metrics.
3. In the following list, select a metric for monitoring.

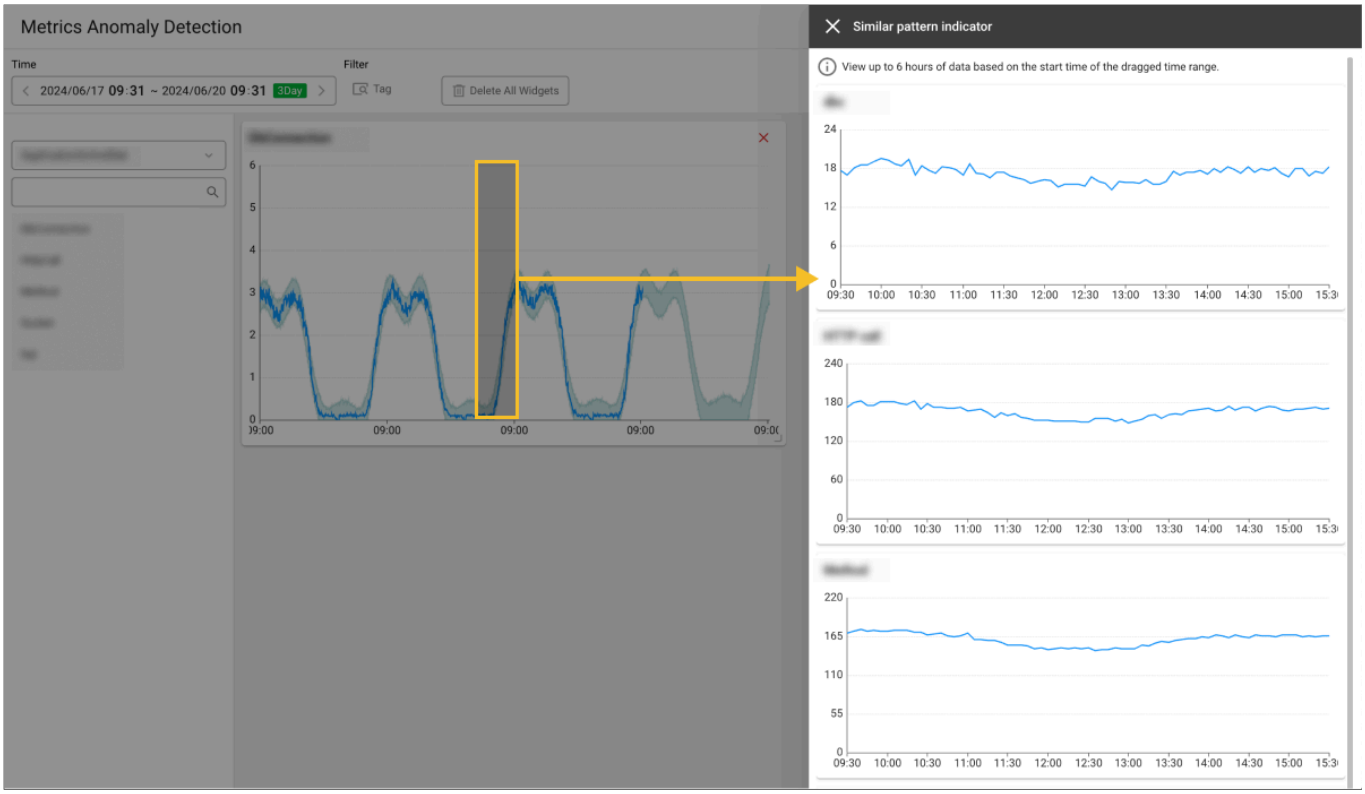
Place the selected metric to the right of the screen.

- Select 2 **Tag** to check the metrics separately by selecting a monitoring target. A tag selection list appears on the right of the screen. Select desired items, and then add widgets.
- The widgets placed on the screen are not reflected in the chart even if the **Time** or **Tag** values have been changed.
- Select ✕ on the upper right of the widget on the screen to delete.
- You can reposition a widget by dragging it while selecting the upper left part of the widget.
- You can resize a widget by selecting the upper right part of the widget and then dragging it.

Deleting a widget

To delete all widgets placed on the screen, select **Delete All Widgets** on the screen.

Checking the related metrics



Drag part of the chart in the widget. The **Similar pattern indicator** window appears on the right of the screen. You can search the related metrics for up to 6 hours based on the start time of the time range you dragged.

Transaction Tracking Overview

What is Transaction?

Transaction means the logical unit of work (LUW) on the server side for processing a user browser request. The process for an individual web service (URL) request is a transaction. A transaction in a web application is the process from receiving an HTTP request for a web service (URL) to returning a response.

The performance of the application can be summarized as the performance of these transactions. Transaction performance includes information such as transaction start-to-end time, response time, resource usage, or attributes of the transaction caller.

Basically, the transaction performance can be analyzed through the response distribution and statistics.

Transaction name

The transaction name is URL. However, the Get parameter (query string) is excluded.

Browser Request

```
http://www.whatap.io/hr/apply.do?name='kim'
```

Transaction Name

```
/hr/apply.do
```

> ⓘ In WhaTap, the web service name and transaction name are used interchangeably. It can be viewed as a module for handling service-specific URLs and requests to them. Because it means a process of a transaction request, the names of both can be called as URL.

Normalizing the transaction name

As the MSA-based systems evolve, instead of URL + ? argument parameter type, it is better to use the methods to put parameters in the URL path.

```
http://www.whatap.io/hr/kim/apply.do
```

If the path parameter is used as the transaction name as it is, it is difficult to analyze the performance from a statistical point of view. You need to normalize this. WhaTap provides options and functions for normalization.

whtap.conf

```
trace_normalize_urls=/hr/{name}/apply.do
```

If set as the above, the transaction name is replaced for collection: /hr/kim/apply.do → /hello/:name/apply.do. To leave the target URL setting as it is and turn off only the function, specify the option as follows. he default value is `true`.

whatap.conf

```
trace_normalize_enabled=false
```

Transaction View

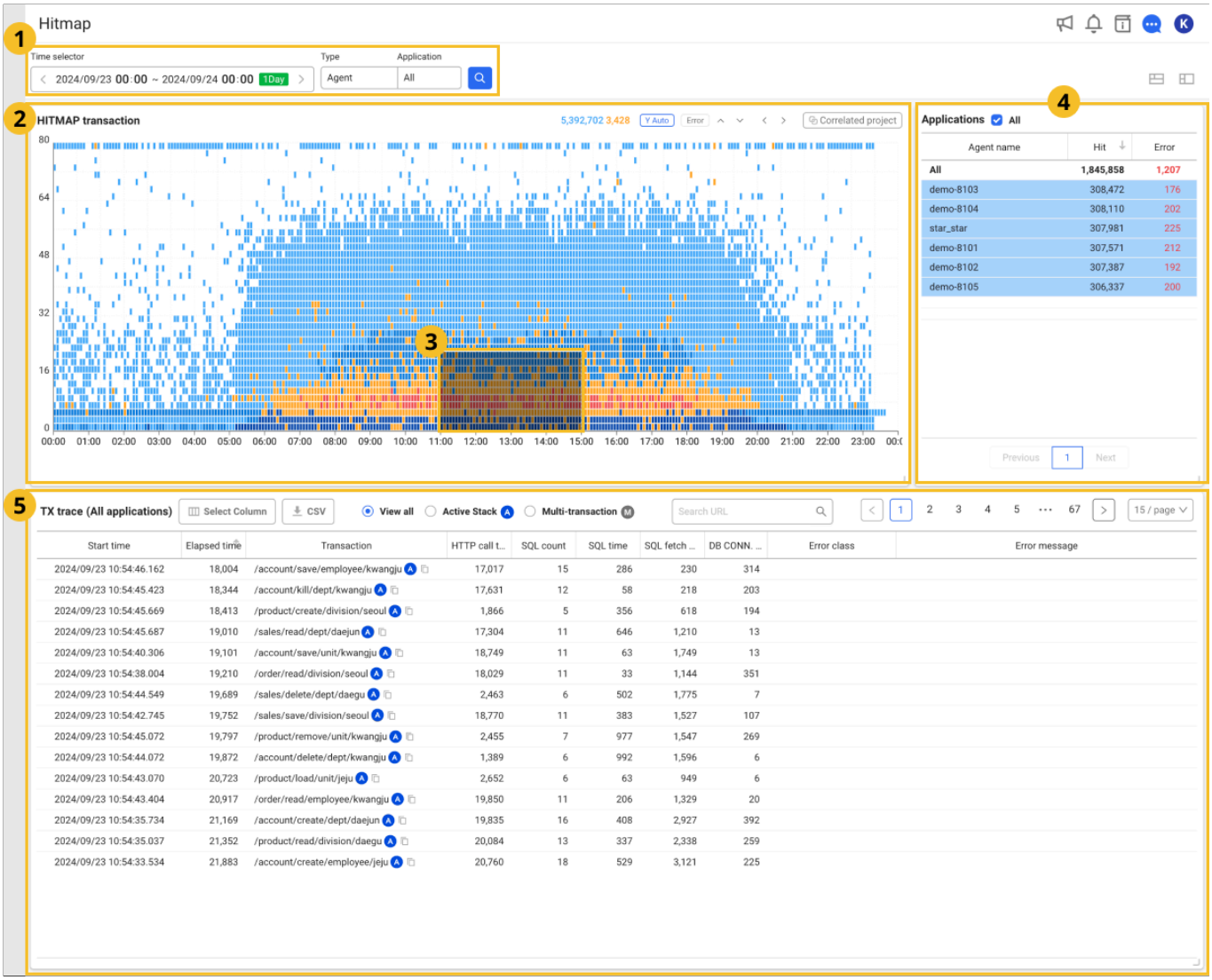
Hitmap

Home > Select Project > **Analysis** > **Hitmap**

On the initial screen of the WhaTap Monitoring service, select a project and then click **Analysis** > **Hitmap**.

> **Hitmap** can be also accessed through the **Hitmap** widget in **Dashboard** > **Application Dashboard**.

Detailed analysis



1 agent area

You can filter by selecting agents linked with the current project. If you select the icon, the **HITMAP transaction** chart can be viewed in the 2 area.

2 HITMAP transaction

If transactions with latency issues are included, they appear in orange or red on the **HITMAP transaction** chart. If only normal transactions are included, they appear in blue.

3 Selection area

If you drag the real-time hitmap chart, the 5 **TX trace** list appears where 4 **Applications** list and transaction information can be viewed in the selected area.

4 Applications

You can check the numbers of transactions and errors included in the selected area on the chart. If you select the desired application in the list, the information details of the application appear in 5 **TX trace**.

5 TX trace

The **TX trace** list contains information details. If you select a desired transaction from the list, the **Transaction information** window appears. In the **Transaction information** window, you can see the detailed trace details for the transaction. For more information, see the following.

Understanding the hitmap patterns

The hitmap is a distribution chart whose X-axis is the transaction end time, and Y-axis is the response time. A normal web application displays a distribution concentrated in a few seconds or less.

hitmap-pattern

Analyzing the hitmap lines

- **Vertical line (LOCK symptom) pattern**

  If a temporary lock (not only DB lock) occurs during transaction processing, the processing is queued. If the lock is released, pending transactions are finished together at a similar time. This creates vertical lines as follows:

  hitmap-lock

Detecting locks in vertical line patterns is a very powerful concept. Especially in the microservice architecture, locks from backend systems can be equally detected.

hitmap-front-api-db

The vertical lines for response patterns in the front applications are also detected when any lock occurs in the DB used by the back-end system.

- **Horizontal line (timeout) pattern**

  In the 10-second timeout condition, if the resources are insufficient, the transactions wait for 10 seconds until a timeout error occurs. At this time, horizontal lines appear near 10 seconds of the hitmap as follows:

  hitmap-timeout

  If there is a logic to retry after timeout, the horizontal lines repeat every 10 seconds as shown in the figure. The following is the hitmap for **failures**.

  hitmap-error

  (1) The response time increased in the section and (2) the red lines in the section are **horizontal line patterns**. (1) ConnectionPool is exhausted due to section load, (2) the secondary **timeout** failure occurred due to insufficient ConnectionPool.

Using the pattern analysis

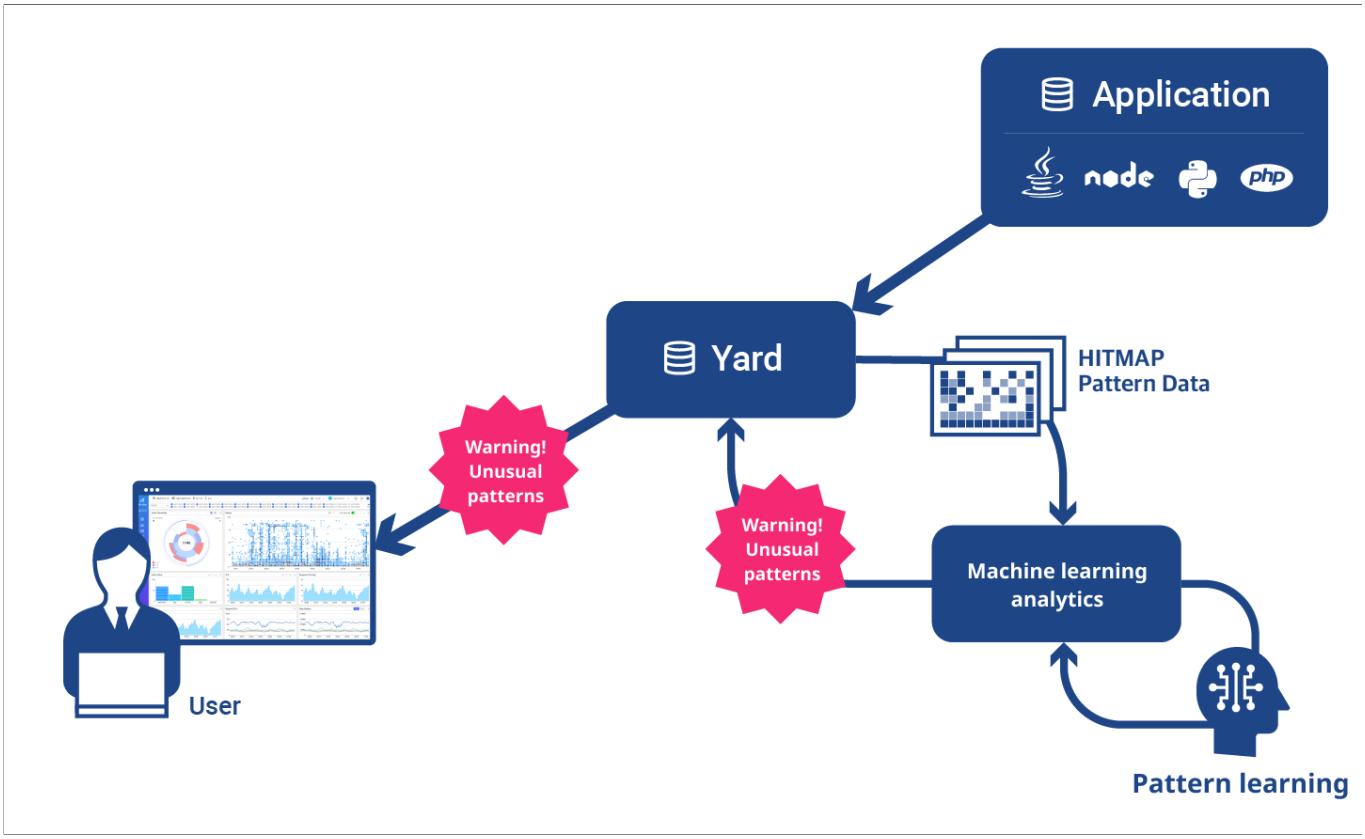
A line in the transaction response distribution indicates a bottleneck. If the lock is temporary, a vertical line is generated. If the bottleneck is release by timeout, a horizontal line is generated.

hitmap-pattern-analysis

When analyzing a problem, you can quickly pinpoint a transaction included in the line for selective analysis.

Analyzing the machine learning based response patterns

This function automatically detects abnormalities after analyzing the hitmap patterns with an aid of machine learning technology and then issues an alert.



Abnormal pattern example

It learns abnormal patterns from hundreds of TB of performance data every month and issues an alarm when a pattern similar to the learned abnormal one is detected.

- **Vertical line pattern**

  hitmap-abnormal1

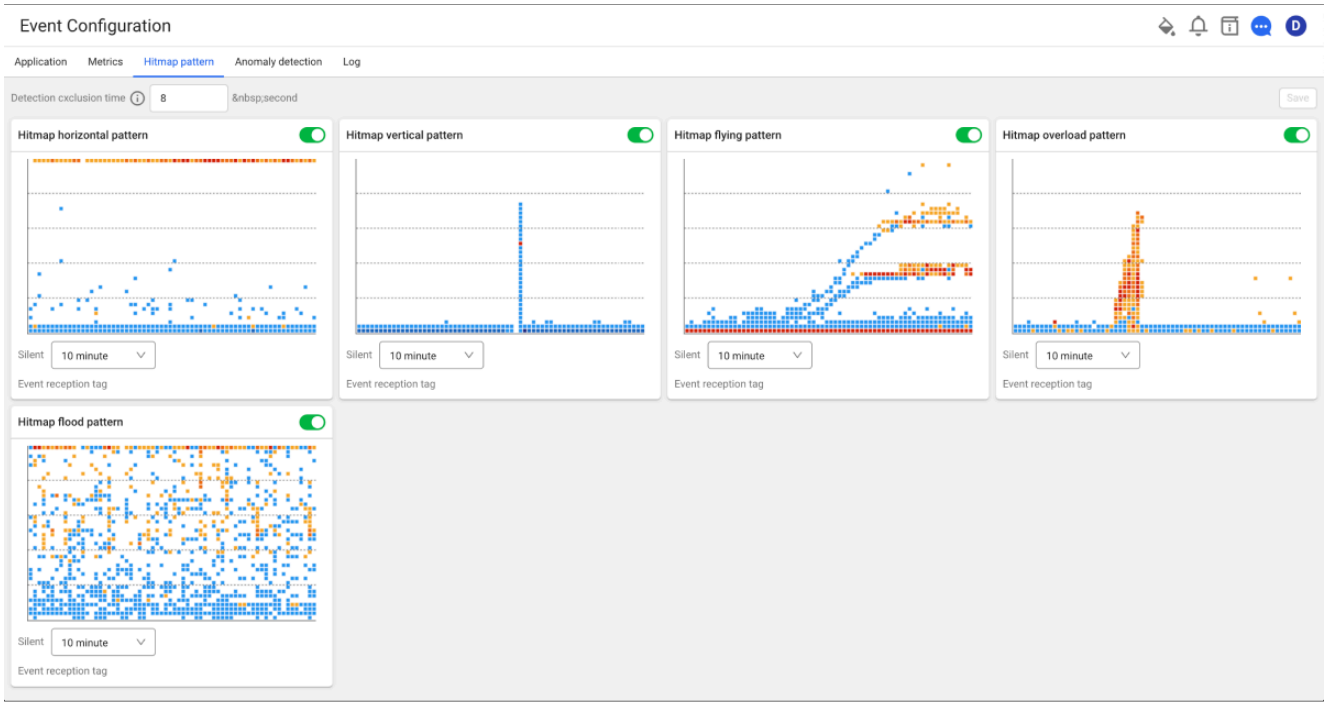
- **Horizontal line pattern**

  hitmap-abnormal2

- **Composite pattern**

  hitmap-abnormal3

- **Hitmap alarm**

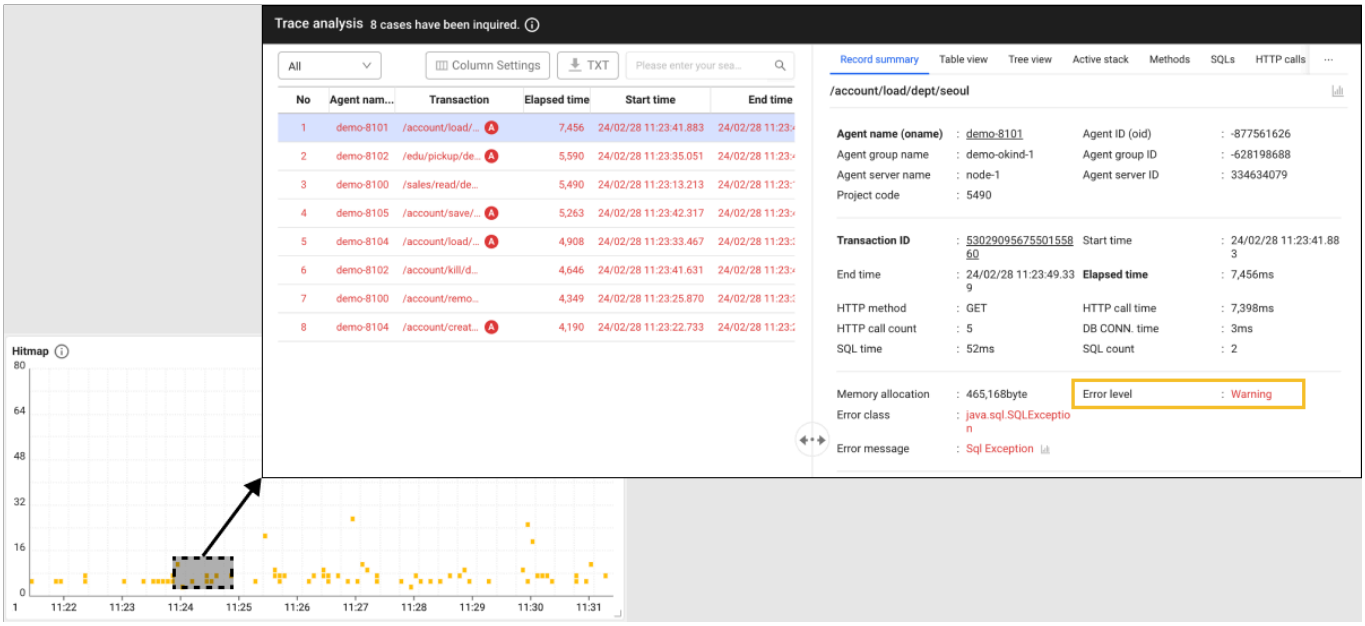


ⓘ For more information about the analysis method of the **HITMAP transaction** chart, see the following.

Criteria for exceptional handling (WARNING) of the Java agent

This guide provides the criteria for the Java agent's exceptional handling for errors that occur in the application environment using the Java language. The following content has been written based on Java Agent v2.2.27.

Error display on the service screen



Most **Error level** that can be seen throuth the **Hitmap** widget and **Trace analysis** window are for the WARNING level.

Error handling when passing the Spring ExceptionHandler

In the Spring framework, passing the `org.springframework.web.servlet.DispatcherServlet.processHandlerException` method is treated as an error. However, if a developer uses the Try-Catch statement, WhaTap does not handle it as an error because it does not pass `ExceptionHandler`.

Status Code 400 or higher error handling

If the HTTP response code is 400 or higher, it is treated as an error even if no exception class occurs.

Error handling when calling the JDBC driver

The errors that occur when making a JDBC connection call or SQL execution call, are treated as errors.

Error handling for HTTP calls

The errors that occur during HTTP calls appear in the `SLOW_HTTPC` or `httpc error` format.

Handling the errors defined in WhaTap

The WhaTap monitoring service handles errors depending on specific conditions.

- **TOO_MANY_RECORDS**: The number of fetches is 10,000 or more.

  For the related agent options, see the following document: `profile_error_jdbc_fetch_max=10000`

- **SLOW_SQL**: SQL time is 30,000 ms or more.

  For the related agent options, see the following document: `profile_error_sql_time_max=30000`

- **SLOW_HTTPC**: The elapsed time for the HTTP call is 10,000 ms or more.

  For the related agent options, see the following document: `profile_error_httpc_time_max=10000`

- **httpc error**: HTTP call status code of 400 or more

  For the related agent options, see the following document: `httpc_status_error_enable=true`

- **status error**: Transaction status code of 400 or more

  For the related agent options, see the following document: `transaction_status_error_enable=true`

Transaction error step INFO processing

Some errors can be displayed as normal or ignored through the agent settings. The level displayed in the **Hitmap** widget is INFO (blue).

- `biz_exceptions` : Displays the record error.

  whatap.conf

  ```
  biz_exceptions=io.home.test.baseapp.base.exception.custom.CustomParameterizedException
  ```

- `ignore_exceptions` : Does not display the record error.

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
  ```

> ⓘ The agent settings can be configured in whatap.conf and they can be viewed in the **Management** > **Agent CONFIG.** menu. For more information, see the following.

Exception handling for methods

If a method set via the `hook_method_patterns` or `hook_service_patterns` option generates an exception during execution, the exception is treated as an error.

If an exception is handled inside a method by using the Try-Catch statement, the Java agent excludes the exception from collection and does not handle it as an error.

Transaction Profile

What is Transaction Tracing?

While the transaction performance means the summary metrics or attributes between the start and end of transaction, the transaction profile traces the steps during transactions. If any transaction is slow or erroneous, the execution records must be traced step by step to find the cause. This is called **transaction tracing**.

The types of steps collected for transaction performance tracing are as follows.

- **DB connection step** START-END

  It includes the performance for connection to RDB. Step information includes the name, response time, and error.

- **SQL Step** START-END

  It includes the performance for JDBC SQL. Step information includes connection data, SQL statements, and errors.

- **HTTP Call step** START-END

  It includes the performance for the outbound HTTP service call. Step information includes URL, host, port, response time, and errors.

- **Message step** ADD

  While collecting traces, the message steps are used to collect the records of all irregular sections.

- **SOCKET Step** ADD

  Step to represent the socket open.

- **METHOD Step** START-END

  The method response time is traced.

- **ACTIVE STACK Step** START-END

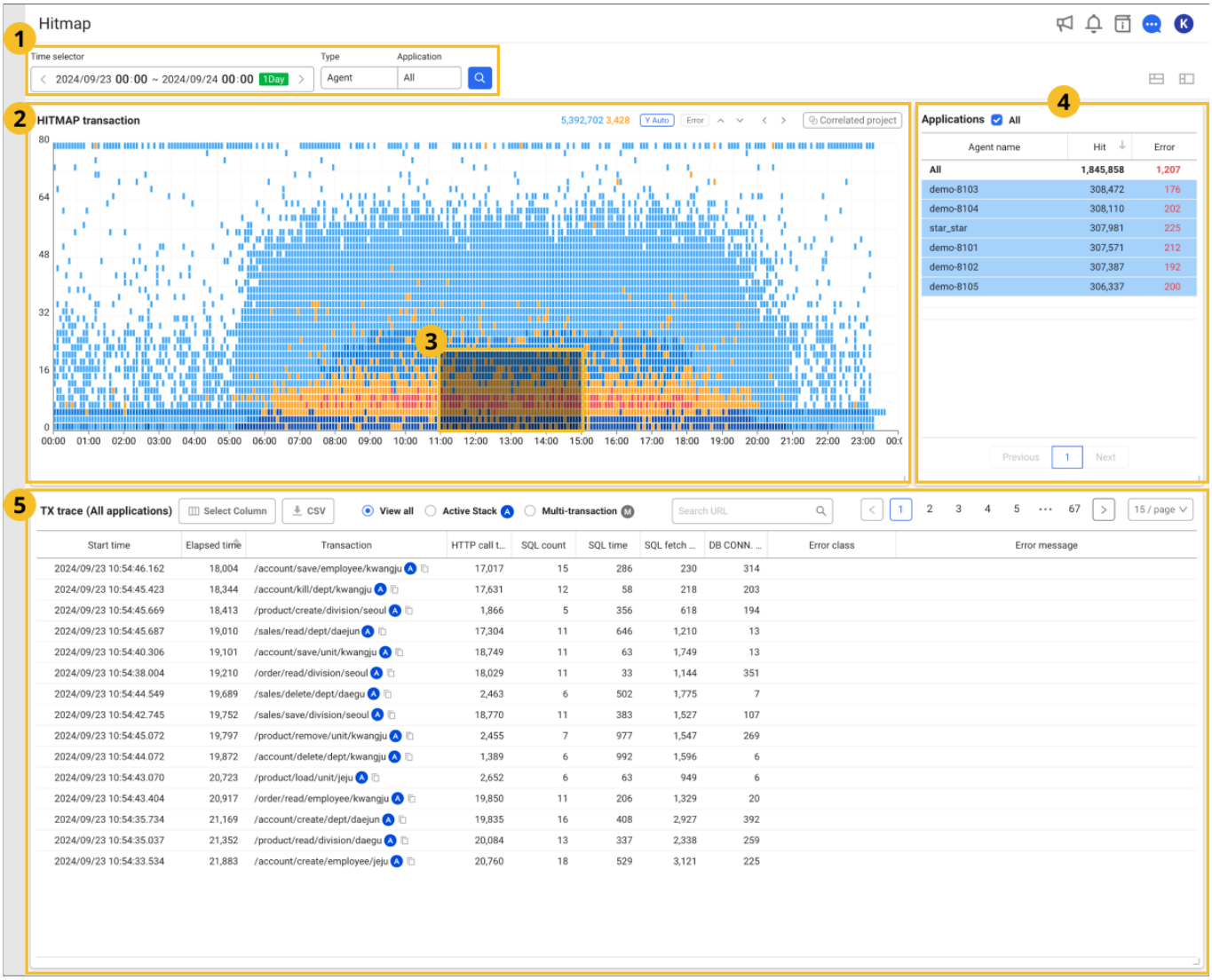
  It includes the active stack information. It is created and collected by a separate thread and it is added to the transaction trace.

If you add a database project through the **Correlated Project Management** menu, you can see the performance metrics of the linked project. For more detailed information about adding and analyzing a linked project, see [the following] (linked-projects).

In the WhaTap monitoring service, **Step** is used as the same meaning as **Span**.

Detailed analysis

Home > Select Project > **Analysis** > **Hitmap**

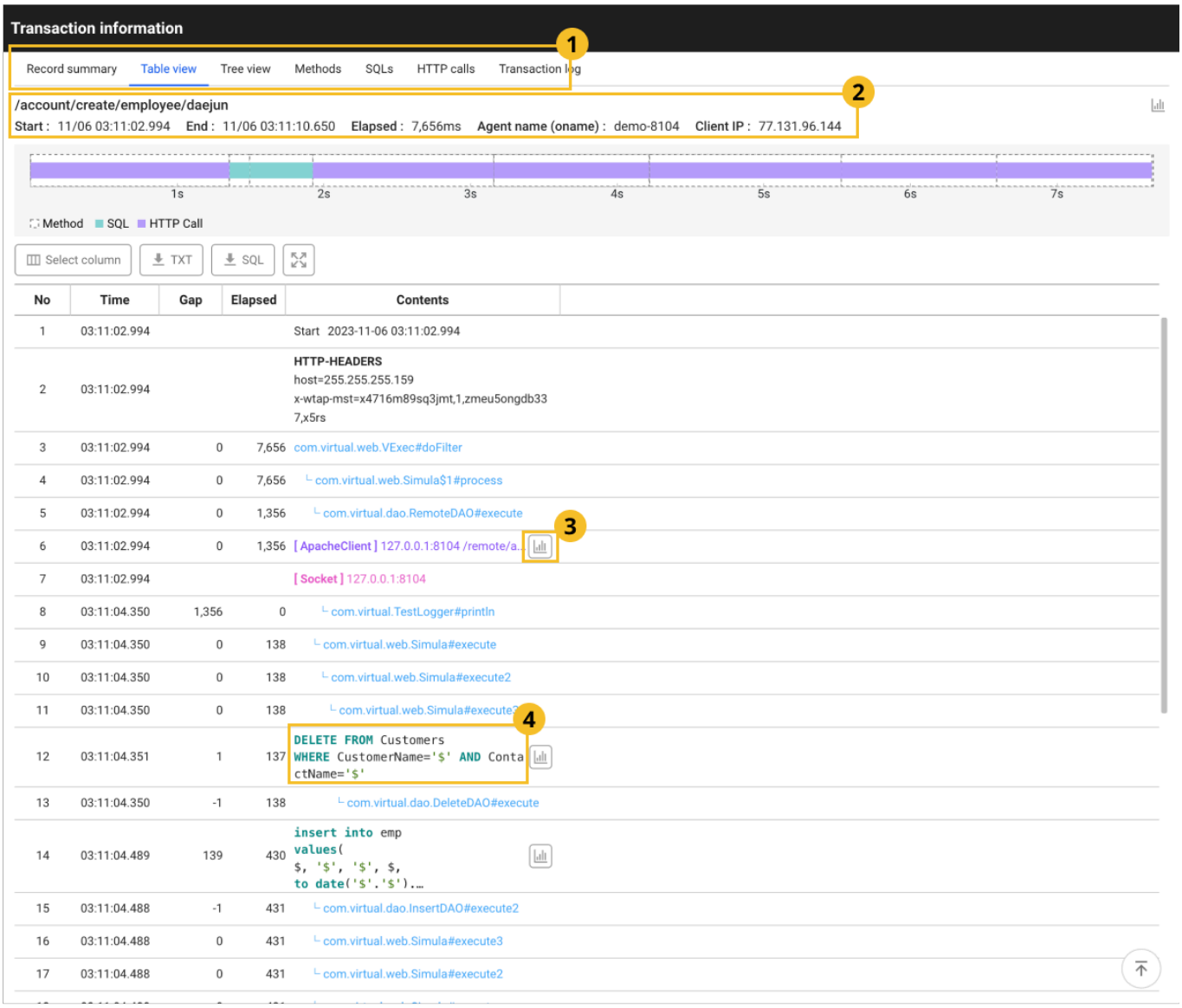


1. In 1 **Time selector**, select a time period to see the hitmap.
2. In 1 **Type** and **Application**, select the targets to view.
3. Select .
4. In 2 **HITMAP transaction**, drag a desired area as in 3.
5. If the list appears in 5 **TX trace**, select a transaction to analyze.

The **Transaction information** window appears where you can check properties such as client information for transaction performance analysis, transaction processing performance, and detailed execution history for each section.

- For more information about the **Hitmap** menu, see the following.
- For more information about the hitmap widgets on the application dashboard, see the following.

Guide to transaction information window



- In the 1 area, the step data can be seen in various formats. Select a desired type of tab.
- In the 2 area, you can see the basic information for transactions.

- 3 📊: The statistical data window for the corresponding URL or each execution section appears. Click a desired time on the graph chart in the statistical data window to go to **Statistics** or **Transaction Search**. You can view the statistical data based on the selected time.

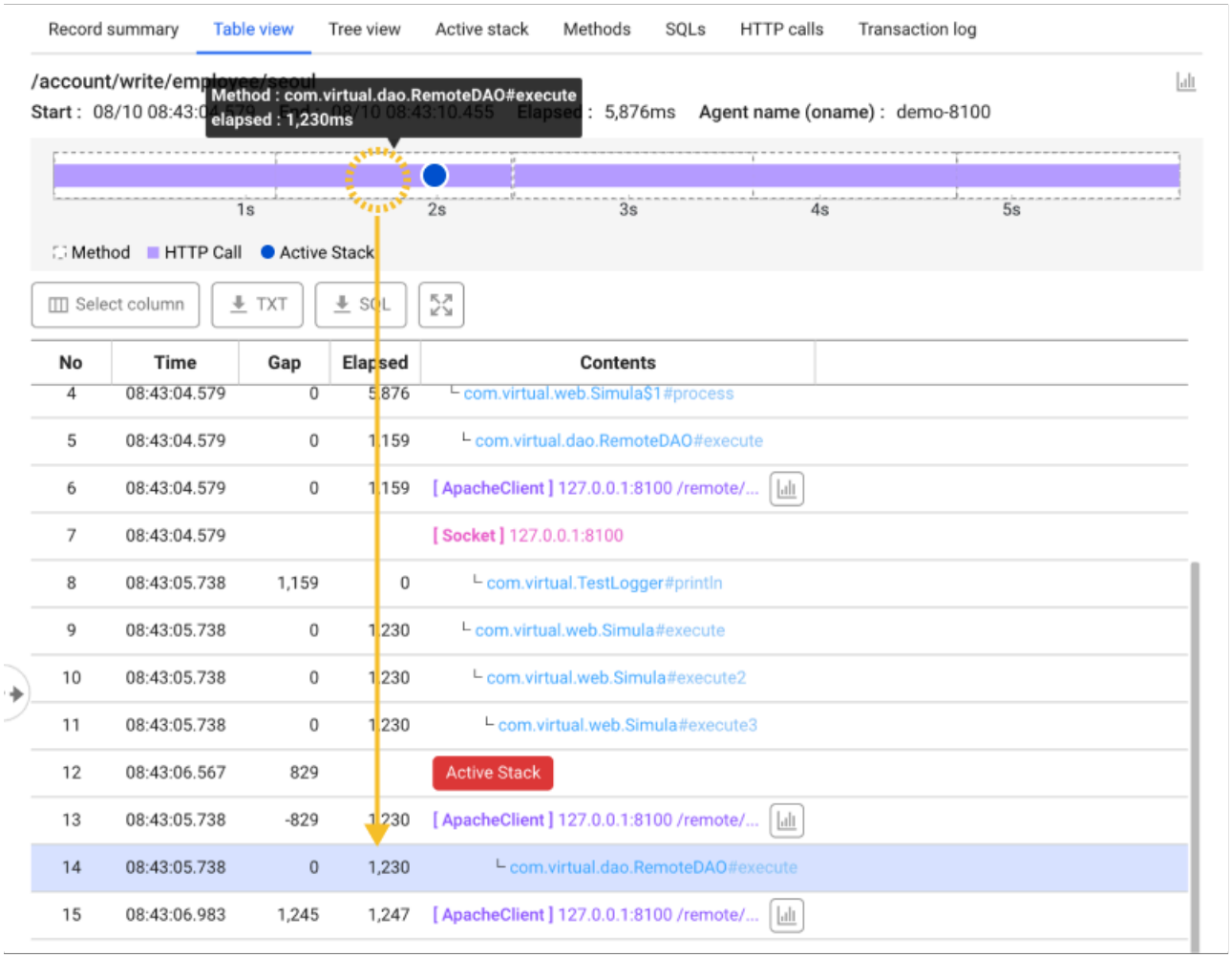
  > ◦ For more information about **Statistics**, see the following.
  >
  > ◦ For more information about the **Transaction Search** menu, see the following.

- If you select 4 SQL step, the SQL variables and HTTP queries can be viewed. For more information, see the following.

Table view

In the **Table view** tab, you can see the transaction execution steps in chronological order.

- You can classify transactions by each execution section and see them in a diagram. Out of all elapsed times, you can quickly find the times spent for each section and the section that took the longest.
- If you select each section in the diagram, the corresponding table list appears where the step is located.



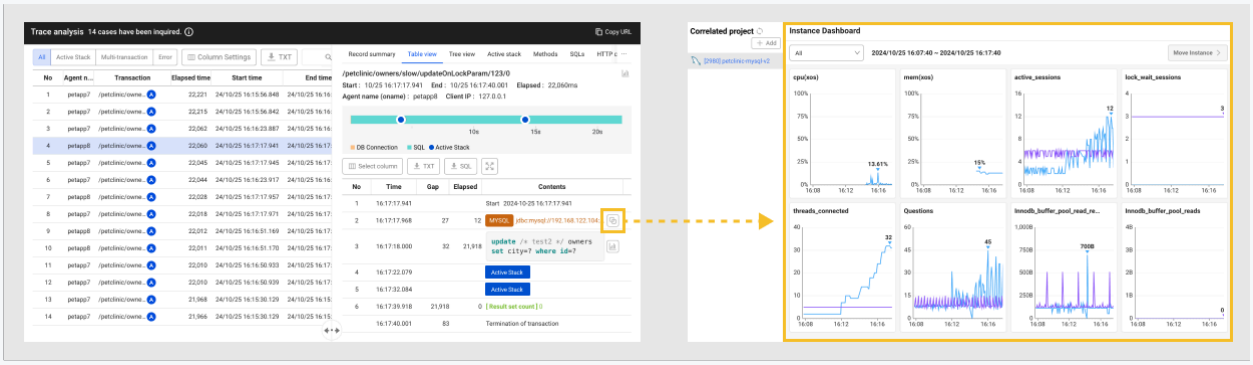
- The area where the blue circle (🔵) is located, is the moment when the active stack has been collected. If you select the blue circle, the table list appears where the **Active Stack** button is located. If you click **Active Stack**, you can check the step performed during the section.

> ⓘ
> - In the transaction list, you can see them from the items marked with the Ⓐ icon.
> - For more information about active stacks, see the following.

- **Select column**: You can add or hide the memory accumulation and CPU accumulation data as columns in the table list. The columns are as follows:
  - **No**: Step occurrence sequence

  - **Time**: Start time of each step
  - **Gap**: The waited time between the start time of the previous step and the current step may have a difference, while there may be a gap from the elapsed time if a delay happens due to external factors.
  - **Elapsed**: Total elapsed time from the start to the end of each method
  - **Contents**: Execution details of the step
- **TXT**: You can download basic transaction data and execution data for each section in a text file.
- **SQL**: The basic transaction information and SQL execution data can be downloaded as a text file.
  - **Download with substitute value**: It downloads the SQL statistics data after replacing the bind variable value with its original one. If a security key (`paramkey`) is set, the security key input window appears. You must enter the security key for downloading. For more information about the agent configuration related to the security key, see the following.
  - **Download without substitute value**: It downloads SQL statistics data without replacing the bind variable value.
- : You can see only the table list, with hiding the transaction summary and diagrams. If you select , the hidden information reappears. This feature is useful when you have a long list of tables.
- If you select the SQL step, the **SQL** window appears to display parameters. For more information, see the following.

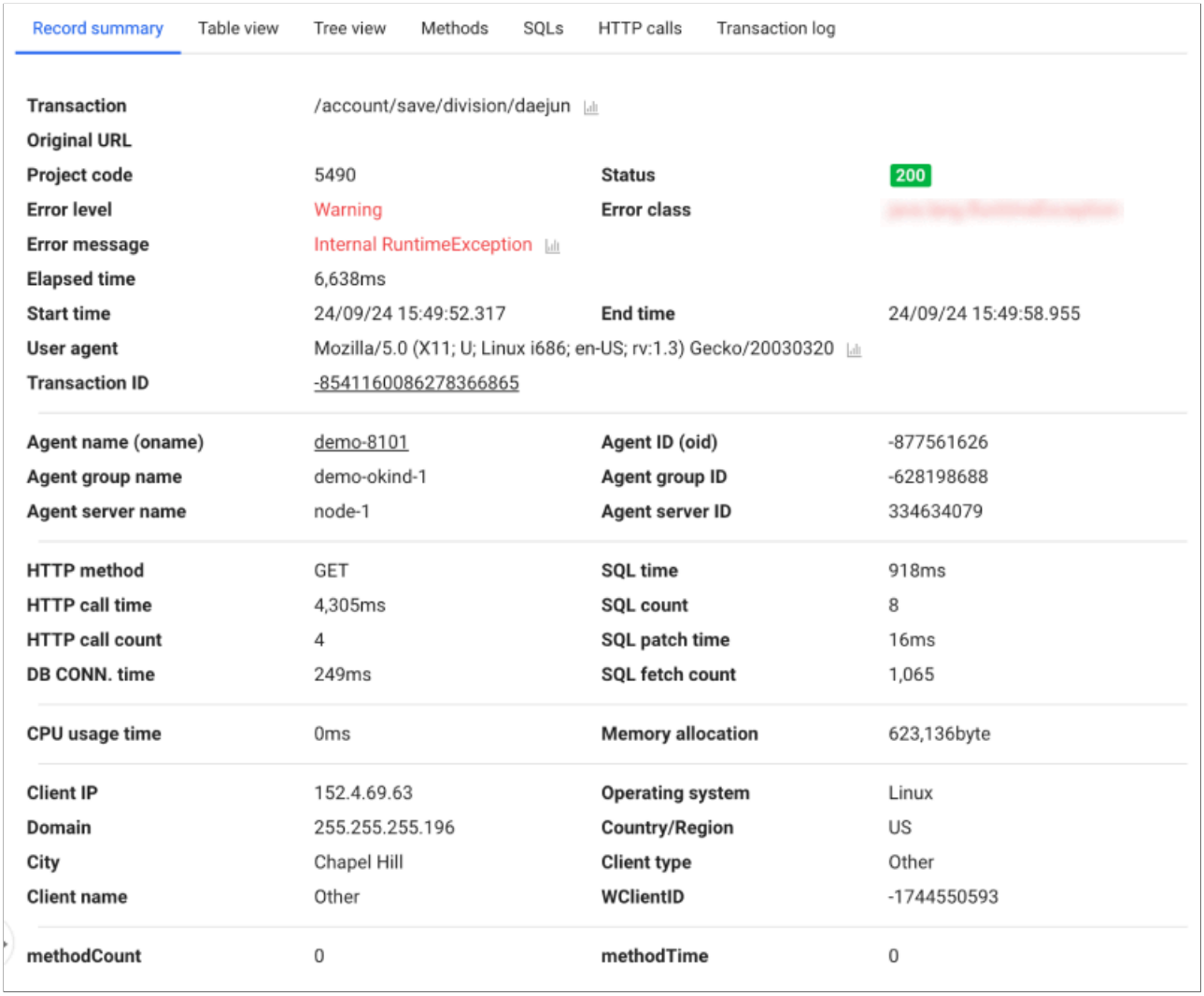
If you add a database project through the **Correlated Project Management** menu, you can see the performance metrics of the project linked in the **DB connection step**.



For more detailed information about adding and analyzing a linked project, see [the following] (linked-projects).

Record summary

To see the basic information for transactions, select the **Record summary** tab. You can see the basic information of the transaction, agent on which the transaction was performed, memory allocation usage, and client information.



Classification	Properties	Description
Transaction	Transaction	API or URL call executed by the application, and the path of the transaction. It converts to a specific format using regular expressions.

Classification	Properties	Description
	Original URL	It is the URL path that was actually called. A concrete request path that does not include variables. If there is an original URL, the original URL appears. If there is no original URL, it is not displayed.
	Project code	Identification code of the project registered in the WhaTap monitoring service.
	Status	HTTP response status code for the transaction.
	Error level	Level of the error that occurred in the transaction. It is displayed as Warning or Critical.
	Error class	Class of the error that occurred in the transaction.
	Error message	Message of the error that occurred in the transaction.
	Elapsed time	Transaction execution time.
	Start time	Transaction start time.
	End time	Transaction end time.
	User agent	Browser used to extract client-related data.
	Referer	Page address or incoming path used by the browser of the client.
	Transaction ID	It is a unique ID of the transaction.
	Multi transaction ID	It is a unique ID of the multi-transaction. It appears when executed in parallel with transactions of other applications.
Agent	Agent name (oname)	Agent name

Classification	Properties	Description
	Agent ID (oid)	It is a unique ID of the agent.
	Agent group name	Agent group name set with the `okind` option.
	Agent group ID	It is a unique ID of the agent group.
	Agent server name	Agent server name set with the `ondoe` option.
	Agent server ID	It is a unique ID of the agent server.
Step	HTTP method	HTTP method. (GET, POST, PUT, HEAD, etc.)
	HTTP call time	Outbound HTTP call time.
	HTTP call count	Number of outbound HTTP calls.
	DB CONN. time	Time taken to connect to the database.
	SQL time	SQL execution time.
	SQL count	SQL execution count.
	SQL patch time	Time taken to search for the record. Nesting may occur or other logic may be included.
	SQL fetch count	Number of the records retrieved from SQL.
Resource	CPU usage	CPU time used for processing transactions.

Classification	Properties	Description
	time	
	Memory allocation	Amount of memory used by the transaction.
Containers	Pod name	It is the name of the Kubernetes Pod that processed the transaction.
	Container Id	It is a unique identifier for a specific container running within a Pod.
Client	Client IP	Client IP address.
	OS	OS environment in which the browser runs.
	Domain	Internet address assigned to the IP address that the client connected to.
	Country/ Region	Client's country.
	City	City in the client's country.
	Client type	Browser type used by the client.
	Client name	Device name used by the client.
	WClientID	This is a unique ID of the client.
Method	methodCount	Number of methods called by transactions.
	methodTime	Time taken for method execution.

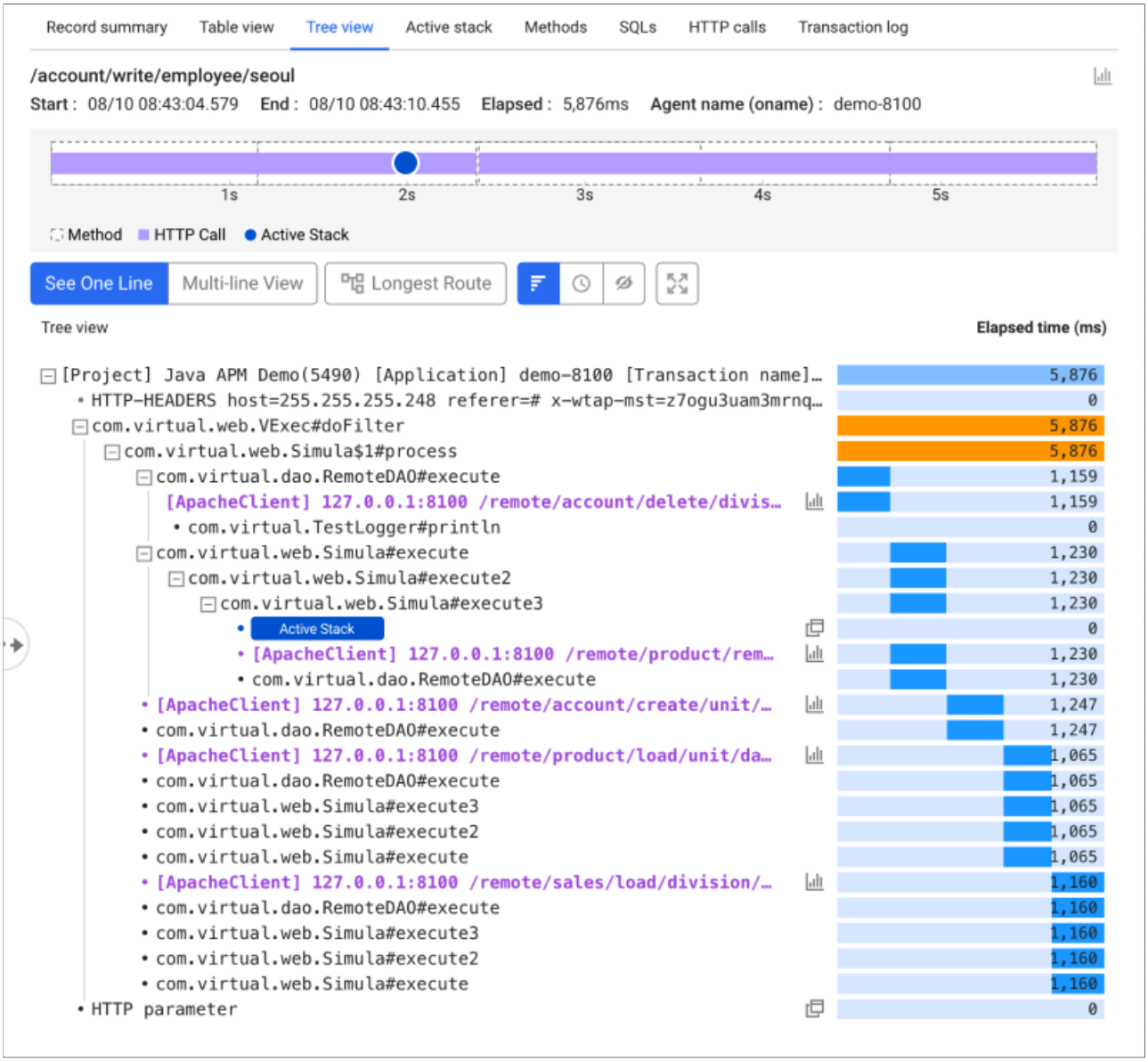
- The collected data may differ depending on the application type, settings, and step type. **Container** entries appear only when the application is running in the container environment.
- Error related items are displayed when an error occurs from the transaction.
- Some transaction properties provide statistical features. If you select the button, a window

appears where you can see the status of transactions occurred during the query time through a time series chart. If you select a bar graph in the chart, you can go to the **Analysis** > **Transaction Search** menu to view details.

- WhaTap basically stores the client-related information. For more information about the agent settings for user data collection, see the following.

Tree view

To see the transaction execution steps in a tree format, select the **Tree view** tab. You can see the details of each transaction and its trace details, trace start time, step time, and call relationship. If you select each section of the diagram, the tree view of the step appears.

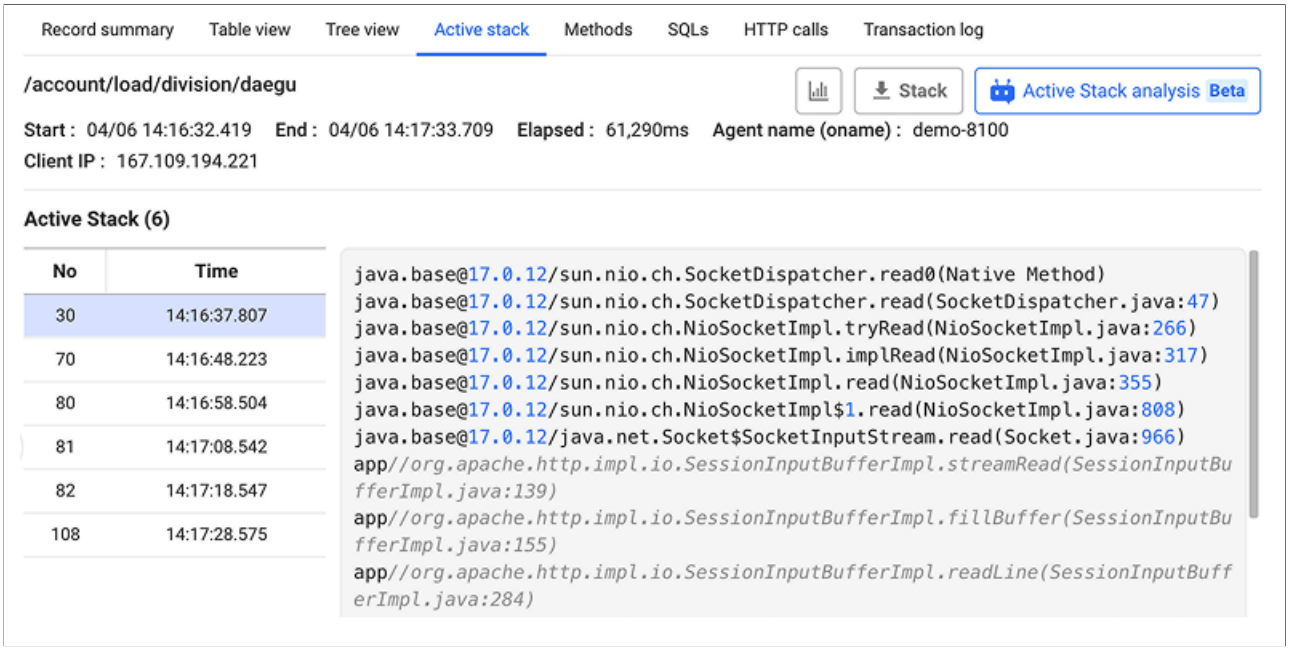


- **Single-line View**: You can arrange the tree format at intervals by displaying the texts displayed in the execution data for each section in a single line.
- **Multi-line View**: All the texts displayed in the execution data for each section, appear by wrapping words.
- **Longest Path**: The longest route can appear.
- **Show Time Bar**: The elapsed times are displayed in a bar chart.
- **Show Time**: The timestamps, gaps, and elapsed times for each segment are displayed in text.

  - 8 seconds or more: Excessive delay state is distinguished in red.
  - Between 3 and 8 seconds: Delay state is distinguished in orange.
  - Less than 3 seconds: Normal state is distinguished in blue.
- **Hide Time**: The time is hidden.
- : You can see only the tree view that hides transaction summary and diagrams. If you select , the hidden information reappears. This feature is useful when the tree view list is long.
- : A window appears where you can view SQL variables and HTTP queries. For more information, see the following.

> In case of start and step times, the time differences caused by the transaction call environment are corrected on the screen through traces and mappings in the parent transactions. Accordingly, there may be differences from the actual time data.

Active Stack



Select the **Active stack** tab to view the list of active stacks collected from the transaction.

- Select a collection point (No, Time) from the **Active Stack** list on the left to display the stack information for that point on the right.
- **Stack**: You can download the active stack details as a TXT file.

> - You can check the active stack from items marked with the A icon in the transaction list.
> - For more information about active stacks, see the following.

AI Active Stack Analysis

Beta

Click the **Active Stack Analysis** button to automatically analyze collected active stacks with AI. It identifies performance issues based on the stack dumps of in-progress transactions and suggests improvements.

The analysis results consist of the following 4 sections:

- **Summary**: Summarizes the number of analyzed stacks and the overall execution flow.
- **Common Patterns**: Displays repeatedly observed execution patterns, including pattern name, occurrence count, and description.
- **Issues Found**: Displays identified performance issues with severity levels (**High**, **Medium**, **Low**) along with the stack information used for the assessment.

Severity	Criteria
High	Deadlock, complete blocking, same blocking repeated 4+ times
Medium	I/O wait, 2-3 lock contentions, slow external calls
Low	Single-occurrence wait, minor optimization opportunities

- **Recommendations**: Suggests specific improvement actions by priority.

Each analysis item displays the collected stack number (`No.N`). Click the stack number to view the original stack.

> - The analysis results are generated by AI and may differ from actual conditions. Use them as

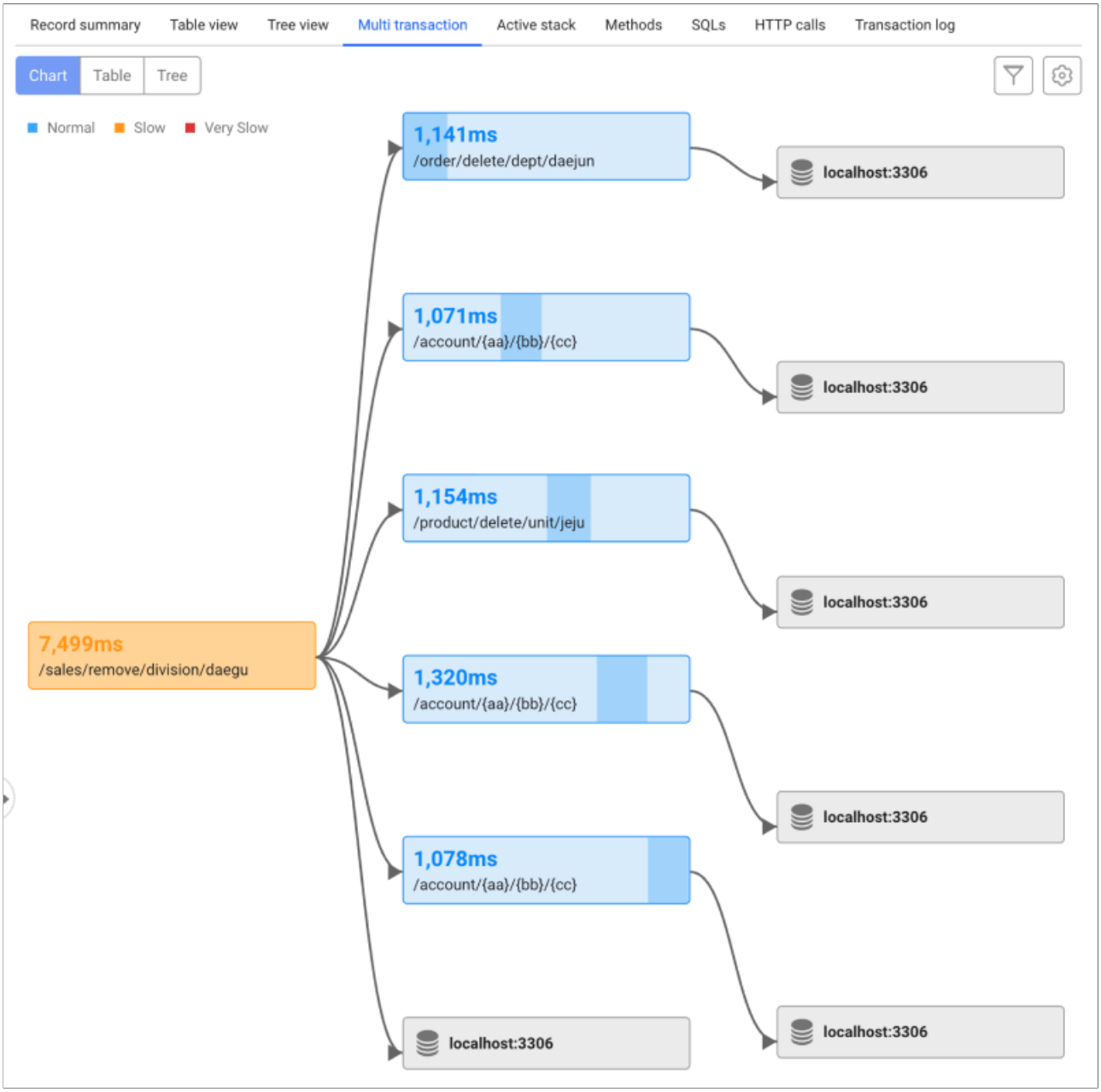
reference material.

- **Member** role or above is required.

Multi-transaction

Multi-transaction indicates the transaction involving other agents or projects. In the **Multi transaction** tab, you can see the call relationships between applications registered in the WhaTap monitoring service.

To trace multi-transactions, set the `mtrace_enabled` option to `true` in **Management** > **Agent CONF.**. For more information about the agent configuration, see the following.



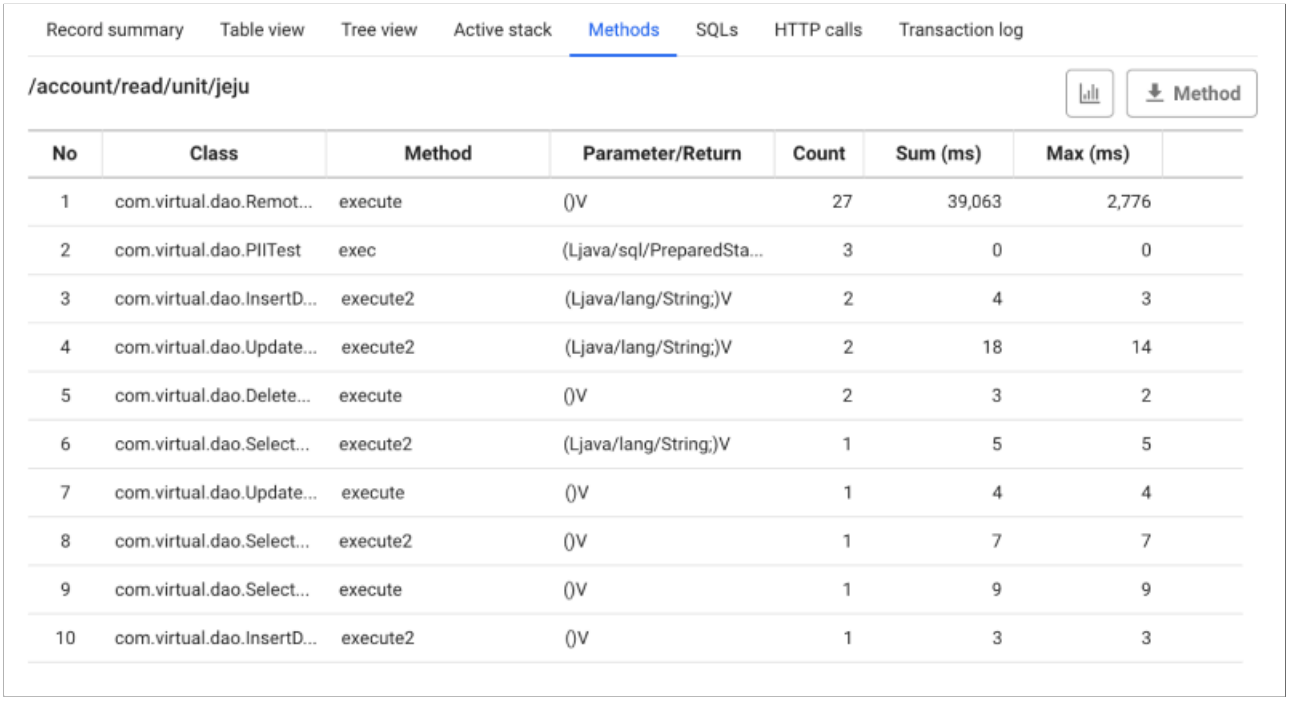
- **Chart**: Provides the call relationships for each transaction in a flowchart. Select a transaction node to see the trace analysis for the transaction node. In **Chart** mode, you can use the mouse to go to a desired position or zoom in or out by scrolling.
  - ⚙ **Chart view settings**: You can display or hide the elements on the chart.
- **Table**: In the form of table, you can check the data for each transaction included in the multi-transaction. You

can select the column selection icon to edit the table header columns. You can see the trace analysis data by selecting each transaction item.

- **Tree**: You can see the call relationships between transactions in a tree format. The related additional functions are the same as those of the **Tree view** tab.
- **Project selection**: You can select or deselect projects to be displayed on the chart.

> - You can see them in the transaction list marked with the M icon.
> - For more information about the multi-transaction, see the following.

Methods



No	Class	Method	Parameter/Return	Count	Sum (ms)	Max (ms)
1	com.virtual.dao.Remot...	execute	()V	27	39,063	2,776
2	com.virtual.dao.PIITest	exec	(Ljava/sql/PreparedSta...	3	0	0
3	com.virtual.dao.InsertD...	execute2	(Ljava/lang/String;)V	2	4	3
4	com.virtual.dao.Update...	execute2	(Ljava/lang/String;)V	2	18	14
5	com.virtual.dao.Delete...	execute	()V	2	3	2
6	com.virtual.dao.Select...	execute2	(Ljava/lang/String;)V	1	5	5
7	com.virtual.dao.Update...	execute	()V	1	4	4
8	com.virtual.dao.Select...	execute2	()V	1	7	7
9	com.virtual.dao.Select...	execute	()V	1	9	9
10	com.virtual.dao.InsertD...	execute2	()V	1	3	3

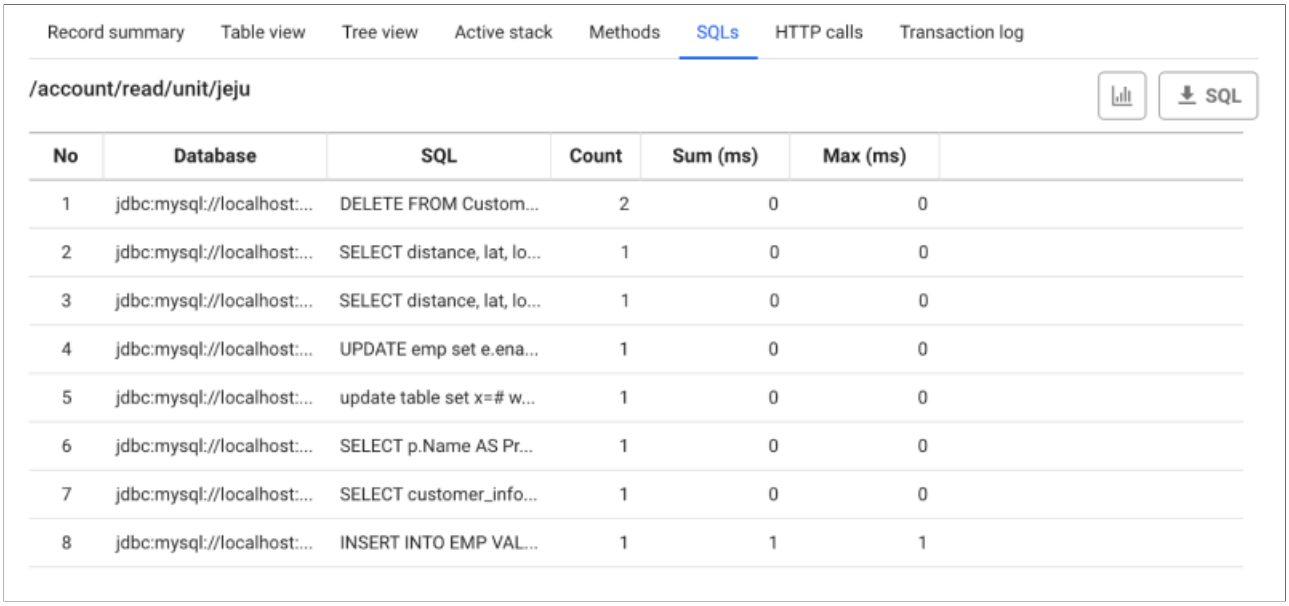
To see only the method data, select the **Methods** tab. It displays the method name and step time for which tracing has been set on the agent. When unnecessary logics are repeatedly executed, it can be used as analysis data to improve the method logic by finding the case where the execution takes longer than the low count.

- : You can see the **Transaction statistics** information. If a graph on the chart is selected, the **Analysis** > **Transaction Search** menu appears. You can see the list of transactions for the selected time zone.

- **Method**: You can download the method-related details as a TXT file.

> For more information about the agent configuration options on methods, see the following. Method tracing support may differ depending on the product in use.

SQLs

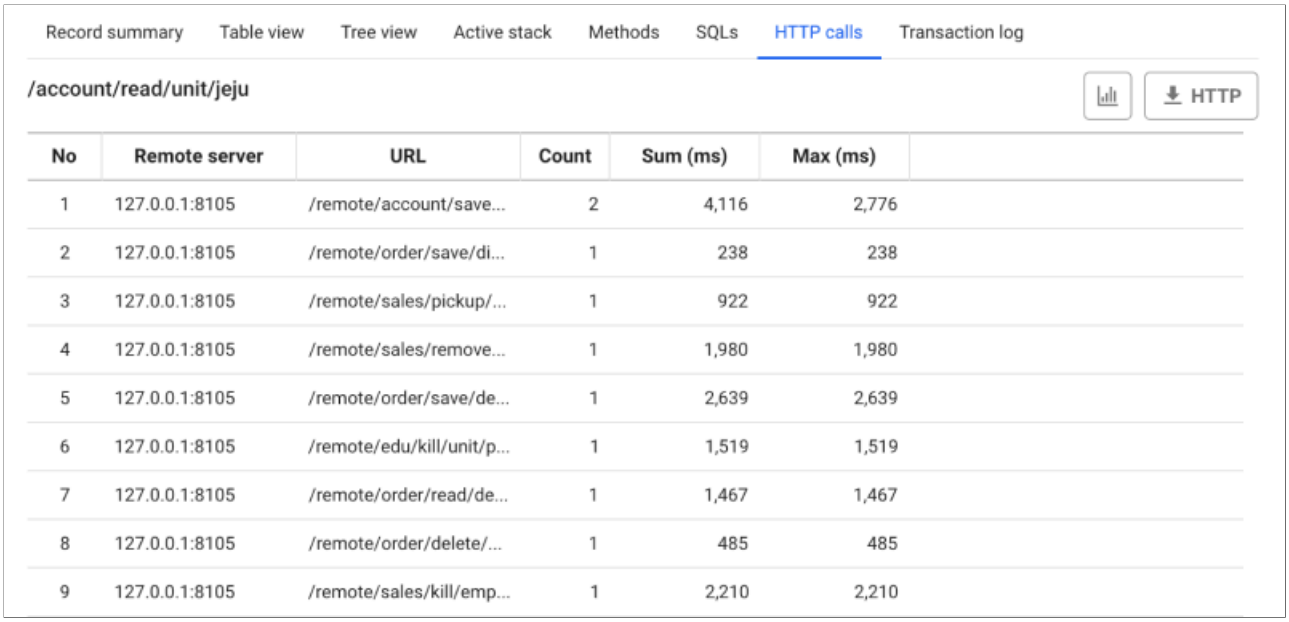


No	Database	SQL	Count	Sum (ms)	Max (ms)
1	jdbc:mysql://localhost:...	DELETE FROM Custom...	2	0	0
2	jdbc:mysql://localhost:...	SELECT distance, lat, lo...	1	0	0
3	jdbc:mysql://localhost:...	SELECT distance, lat, lo...	1	0	0
4	jdbc:mysql://localhost:...	UPDATE emp set e.ena...	1	0	0
5	jdbc:mysql://localhost:...	update table set x=# w...	1	0	0
6	jdbc:mysql://localhost:...	SELECT p.Name AS Pr...	1	0	0
7	jdbc:mysql://localhost:...	SELECT customer_info...	1	0	0
8	jdbc:mysql://localhost:...	INSERT INTO EMP VAL...	1	1	1

To check the SQL statement information, select the **SQLs** tab. In case of unnecessarily repetitive execution, it can be used as analytical data to improve query performance by finding the case where the execution time takes longer compared to the low count.

- : You can see the **Transaction statistics** information. If a graph on the chart is selected, the **Analysis** > **Transaction Search** menu appears. You can see the list of transactions for the selected time zone.
- **SQL**: You can download the SQL-related details as a TXT file.

HTTP calls



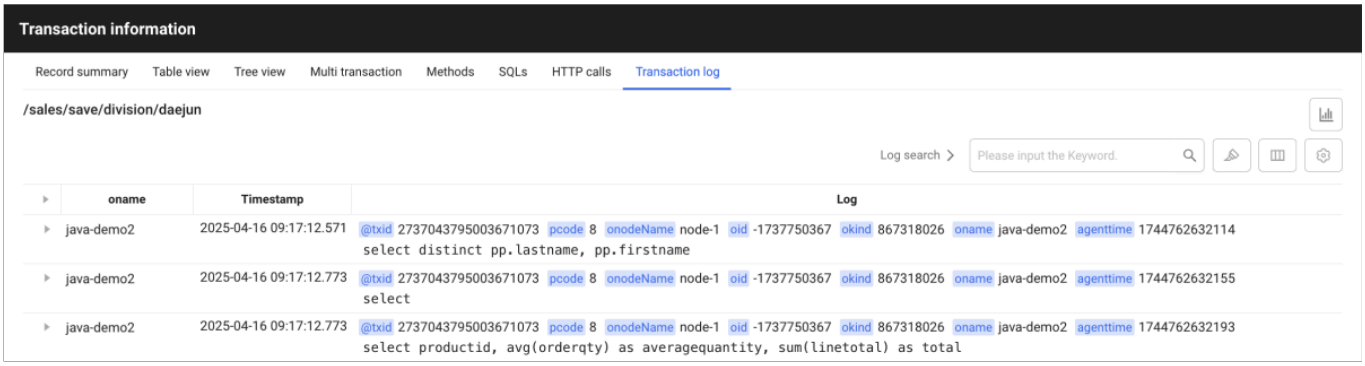
No	Remote server	URL	Count	Sum (ms)	Max (ms)
1	127.0.0.1:8105	/remote/account/save...	2	4,116	2,776
2	127.0.0.1:8105	/remote/order/save/di...	1	238	238
3	127.0.0.1:8105	/remote/sales/pickup/...	1	922	922
4	127.0.0.1:8105	/remote/sales/remove...	1	1,980	1,980
5	127.0.0.1:8105	/remote/order/save/de...	1	2,639	2,639
6	127.0.0.1:8105	/remote/edu/kill/unit/p...	1	1,519	1,519
7	127.0.0.1:8105	/remote/order/read/de...	1	1,467	1,467
8	127.0.0.1:8105	/remote/order/delete/...	1	485	485
9	127.0.0.1:8105	/remote/sales/kill/emp...	1	2,210	2,210

To see the count, time sum, average time, and such for HTTP calls, select the **HTTP calls** tab. If unnecessary external calls are repeatedly executed, it can be used as analytical information to determine whether the transaction delays are external or internal by finding the cases where the execution time takes longer compared to the low count.

- : You can see the **Transaction statistics** information. If a graph on the chart is selected, the **Analysis** > **Transaction Search** menu appears. You can see the list of transactions for the selected time zone.
- **HTTP**: You can download the HTTP call-related details as a TXT file.

Transaction log

To see the logs of the transaction, select the **Transaction log** tab.

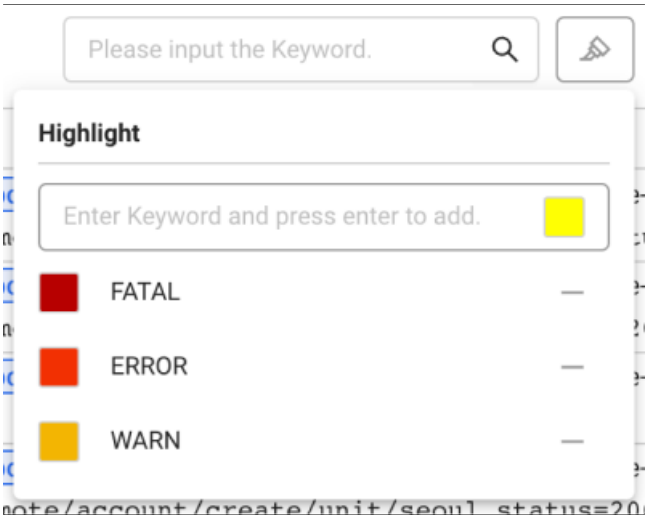


- : You can see the **Transaction statistics** information. If a graph on the chart is selected, the **Analysis** > **Transaction Search** menu appears. You can see the list of transactions for the selected time zone.
- **Log Search**: You can go to Log Search to easily find the desired log data.
- **Keyword search**: Enter the desired text to search in the keyword search field, and then press Enter or select the button. It highlights the keywords that match the entered text.
- **Keyword highlight**: If you set the keyword and color, the texts that match the keyword are automatically highlighted in the log list. For more information, see the following.
- **Column settings**: You can add columns or change the order of columns displayed in the log list table. For more information, see the following.
- **Log view detail setting**: You can configure the items to be displayed in the log list. For more information, see the following.

> - To enable the **Transaction log** tab, apply the agent setting options related to the logs. For more information about the log settings, see the following.
> - The **Transaction log** tab can be accessed by only the members with the **View log** role. For more information about the member roles, see the following.

Setting the keyword highlight

1. To add keywords to highlight, select .



2. Enter a desired keyword.
3. Select the color icon and then select a desired color.
4. Press Enter. The added keywords are added in the list.

Deleting the keyword highlight

To delete the items unused from the highlight list, select —.

Displaying or hiding the table columns

1. Select ▥ on the upper right of the table.
2. If the **Column settings** window appears, select or unselect the desired item in the list on the left.

   You can also hide columns by selecting the ✕ button in the list on the right.
3. After all settings are finished, select **Confirm**. The set content is applied to the table.

Reordering the table columns

1. Select ▥ on the upper right of the table.
2. When the **Column settings** window appears, change the order by dragging the desired items from the list on the right.

3. After all settings are finished, select **Confirm**. The set content is applied to the table.

Setting to display the table content

1. Select ⚙ on the upper right of the table.
2. When the **Log view detail setting** window appears, set the desired options.
   - Select **content** or **Tag** to determine whether or not to display the log content.
   - **Tag management**: Tags are displayed in the log in the order of addition. The order of tags can be changed by dragging. Disabled tags do not appear in the logs.
3. After all settings are finished, click ×. The set content is applied to the table.

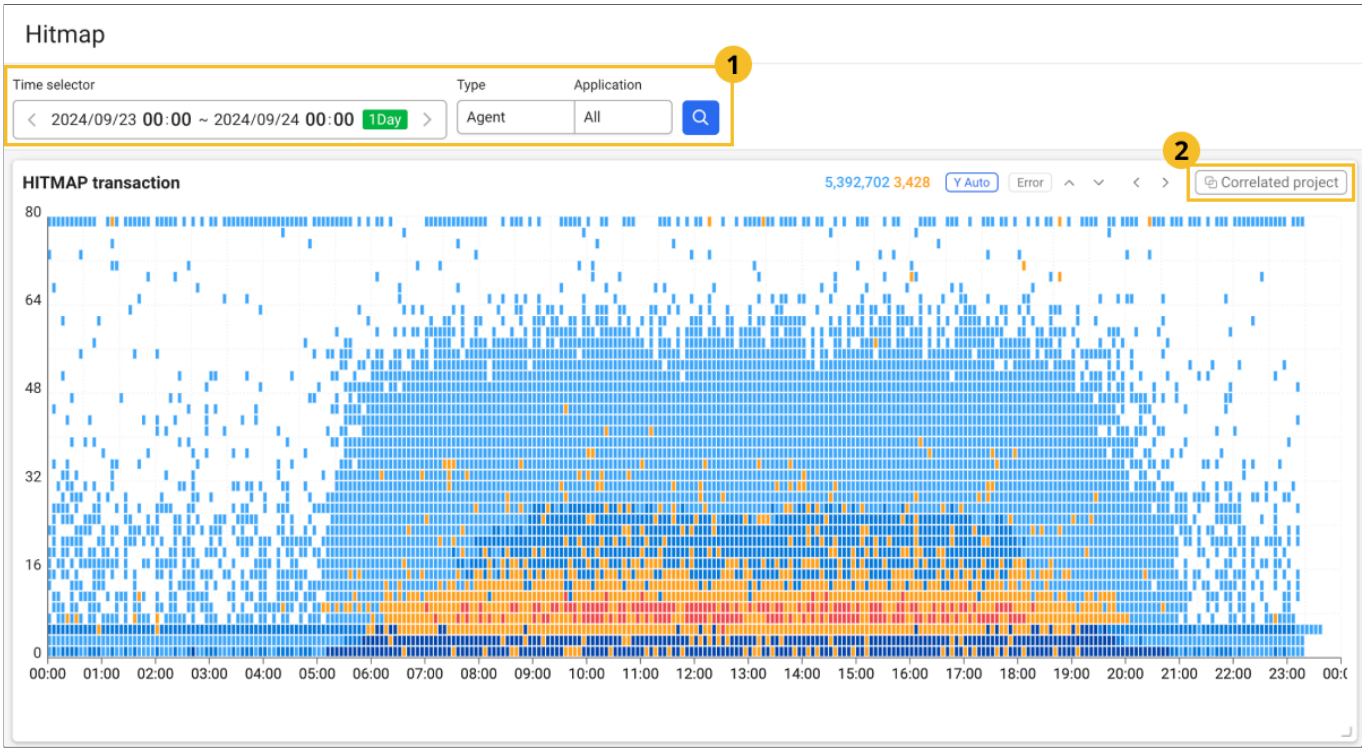
> ⓘ For more information about **Log view detail setting**, see the following.

Additional options

Checking the correlated project metrics

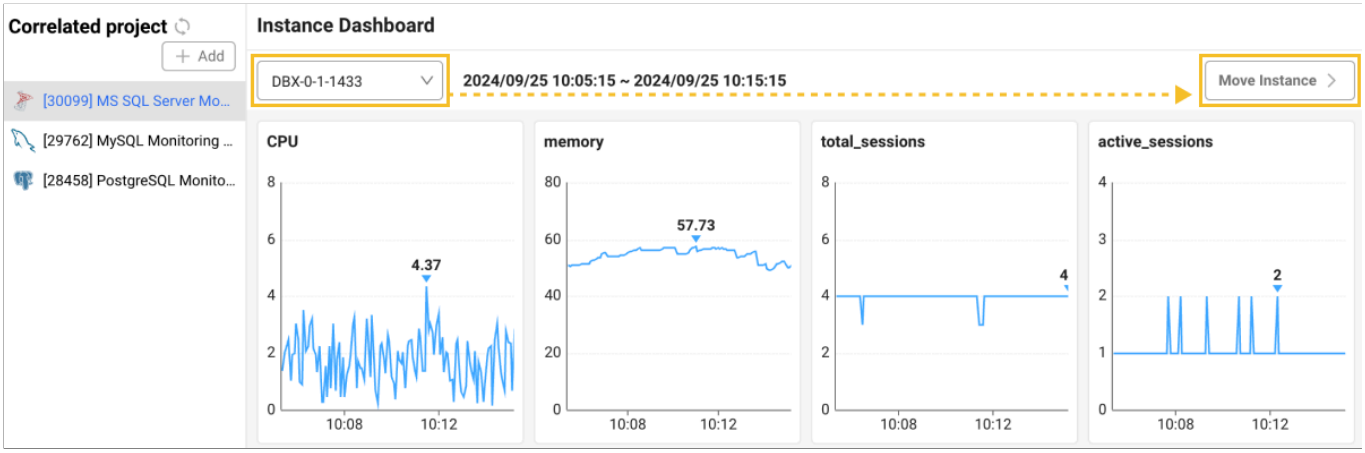
If you have created a database monitoring (DPM) project and an application monitoring (APM) project, you can relate the two projects for monitoring the data collected from DPM in the APM project. You can see performance metrics of the correlated projects while querying transactions via the **Correlated project** feature on the upper right of the **HITMAP transaction** section.

The **Correlated project** feature is useful when analyzing transactions in the APM project. When you found a slow SQL, it is difficult to determine exactly in which DB the problem occurred in the environment using multiple similar DBs.



1. Set the time and **Application** for viewing the transactions, and then select .
2. In the **HITMAP transaction** section, select **Correlated project** on the upper right.
3. A new window appears where you can check the metrics for the correlated project.

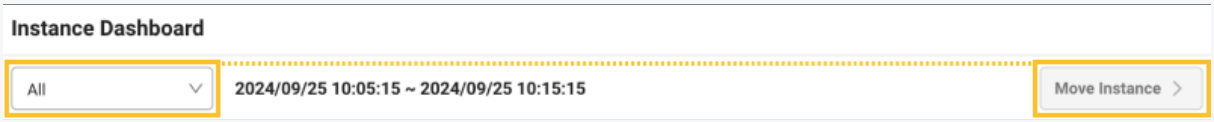
Select the correlated project from the left list and then check the metrics in the **Instance dashboard** section.



To navigate to the dashboard menu for a selected project, select an instance and then click **Move instance** on

the upper right. Go to **Dashboard** > **Monitoring a Database Instance**.

- The **HITMAP transaction** button in the **Correlated project** section appears only when there is an associated project. For more information about adding a correlated project, see the following.
- You can add a project by selecting **Correlated project** at the top of the left **Add** list.
- The **Move instance** button is disabled when **All** is selected.



- Depending on the database product type, you can navigate to the **Analysis** > **Counts Trend** menu.

Viewing HTTP parameters

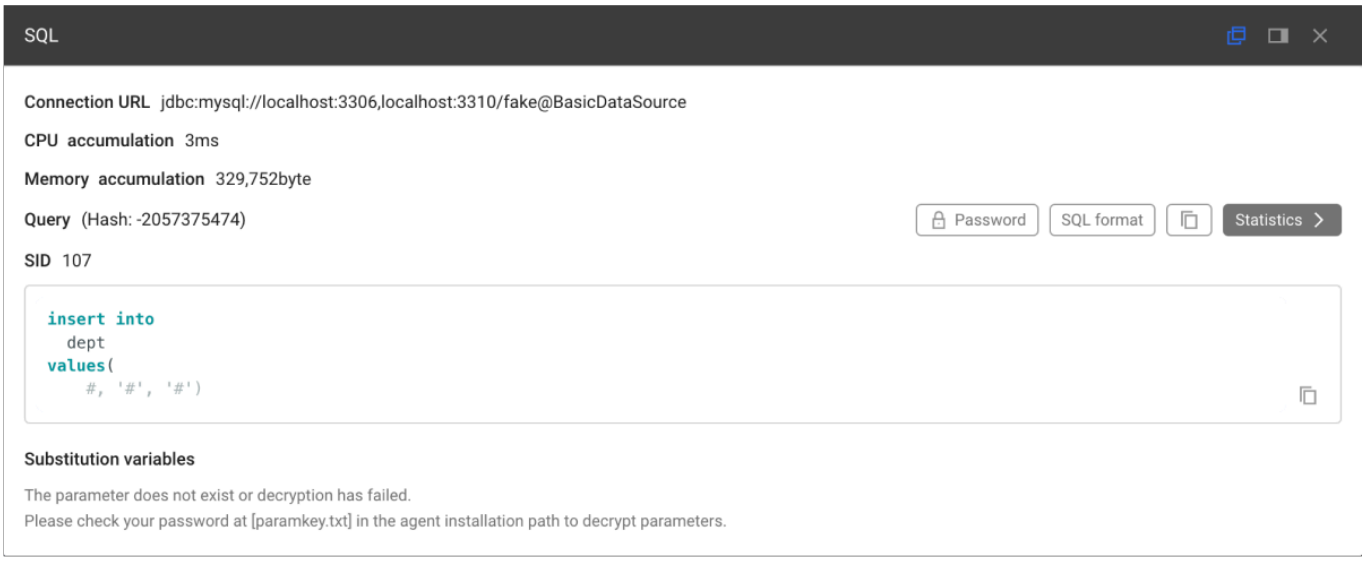
You can view the HTTP parameters of the transaction in the **Table view** tab.

1. Scroll down the page to the final step of the transaction.
2. Select the **HTTP parameters** item.
3. If the **HTTP-PARAMETERS** window appears, select **Password**.
4. Enter the specified Param Key. You can see the hidden parameters.

- For the agent settings for HTTP parameters, see the following.
- **Password**: You can see the decrypted parameter value. The password is a 6-digit string in the `WHATAP_HOME` /security.conf (paramkey.txt) file. You can change it to another string. It is required to view the SQL variables and HTTP queries and to stop the threads. For more information, see the following.

Viewing SQL parameters

Select the SQL step in the **Table view** tab or select in the **Tree view** tab. The **SQL** window appears to display parameters.



- **Password**: You can see the decrypted parameter value. The password is a 6-digit string(`paramkey`) in the `WHATAP_HOME` /security.conf file. You can change it to another string.

  > **Password** is required to view the SQL variables and HTTP queries, and stop the threads. For more information, see the following.

- **SQL format**: You can indent and format your SQL statements for better readability.
- : A SQL statement can be copied to the clipboard.
- **Statistics**: Go to **Statistics** > **SQL**.

To search SQL variables and HTTP queries, add the following options to the agent settings.

- For agent settings related to recording SQL parameters, see the following.
- For agent settings related to recording the HTTP parameters, see the following.

whatap.conf

```
# SQL parameter lookup option: If the option is applied, SQL parameters are collected after being encrypted.
profile_sql_param_enabled=true

// HTTP parameter lookup option: If the option is applied, HTTP query parameters are collected after being encrypted.
```

```
profile_http_parameter_enabled=true
```

Setting the columns in the trace list

You can additionally see the details about the search result by setting the columns in the **TX trace** section.

1. In the **TX Trace** area, click the **Select Column** button.
2. Select or deselect the desired column items. The changes are immediately reflected in the table.
   - **Select all**: It selects all columns.
   - **Deselect all**: All columns are unselected.
   - **Initialize(default)**: The column is set as the default value.
3. Once finished, click the × icon in the upper-right corner.

Filtering the trace list

You can filter the results retrieved from the **TX trace** section using the following options:



- **View all**: You can see all transactions in the viewed result.
- **Active Stack**: Only the transactions that include active stacks can be viewed in the searched result.
- **Multi-transaction**: Only the transactions including multi-transactions can be checked in the searched result.
- **Search URL**: Only transactions containing items that match the entered string are displayed. Enter a desired string and then press `Enter` on the keyboard.
  - **Transaction**
  - **Client IP**
  - **Domain**
  - **Error class**
  - **Error message**

Downloading the viewed list

You can download the results retrieved from the **TX trace** section as a CSV file. The formation of the downloaded CSV file is profile `{project_code}` `YYYYMMDD` .csv.

1. After viewing the transactions, select CSV.
2. Enter a desired value in the **Maximum CSV lines** option.
3. Select **Download**.

Active Transactions

A transaction in progress is called "active transaction." **A stack that has been regularly dumped by an active transaction is called active stack.**

> The WhaTap agent dumps active stacks for active transactions every 10 seconds (optional) and sends them to the server. `active_stack_second=10`

Collecting the compact active stacks

Because the active stack regularly performs thread dumps, it can add lots of overheads to the agent if implemented incorrectly. WhaTap has various options to collect active stacks while minimizing the agent load.

> In **Instance Performance Management** > **Thread List/Dump**, from the thread list, the overhead can be determined by checking the **CPU time** of the WhaTap-ActiveStackDump thread.

Active Stack Example

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(ServletInvocableHandlerMe
org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter.invokeHandlerMethod(RequestMappingHand
```

Optimized data collection

- Dump the stack only for the thread that is performing transactions.
- You can adjust the active stack dump time interval.

  whatap.conf

  ```
  active_stack_second=10
  ```

- It is limited to the maximum line of the active stack. It collects 50 lines from the top line by default.

  whatap.conf

  ```
  trace_active_callstack_depth=50
  ```

- Each hashed line of the active stack is collected. Text is collected only once.
- The maximum number of active stacks collected at a time is limited.

  whatap.conf

  ```
  active_stack_count=100
  ```

Active Stack for Background Threads

Basically, the **active stack** means the stack of the thread in which transactions are being processed. You may need to analyze the stack for some background threads as well. At this time, the ** for the background thread can be obtained using the option. It is available in the Java agent 1.6.2 or later.

- If the value of `async_stack_enabled` is set to `true`, the function is enabled.

  whatap.conf

  ```
  async_stack_enabled=false
  ```

- The stack dump interval depends on the foreground active stack setting.

  whatap.conf

  ```
  active_stack_second=10
  ```

- When setting the target thread name, use `*` to specify the string pattern.

  whatap.conf

  ```
  async_thread_match=http*,abc*
  ```

  It identifies the stack dump target with the thread name. You can specify multiple matches using comma (,) as the delimiter.

- When the Top method of the stack is a class or method registered in `async_thread_parking`, it determines that the thread has been parked and does not generate a dump.

  whatap.conf

  ```
  async_thread_parking_class=sun.misc.Unsafe
  async_thread_parking_method=park
  ```

References

- Active Transaction
- Active transactions that can detect failures quickly

Multi Trace

Multi-transaction means the transaction involving a different agent or project. **Multiple Transaction Trace** is to trace calls between the application services registered in the WhaTap project.

> The Java agent traces multi-transactions with three HTTP header keys (`x-wtap-po` , `x-wtap-mst` , and `x-wtap-sp1`). If HTTP transactions passing through the gateway cannot be traced, check the HTTP header conditions.

> **Enabling the multi-transaction**
>
> To trace multiple transactions, set the `mtrace_enabled` option to `true` in **Management** > **Agent CONF.**. For more information about the agent configuration, see the following.

How to Use Multi-Transaction Trace

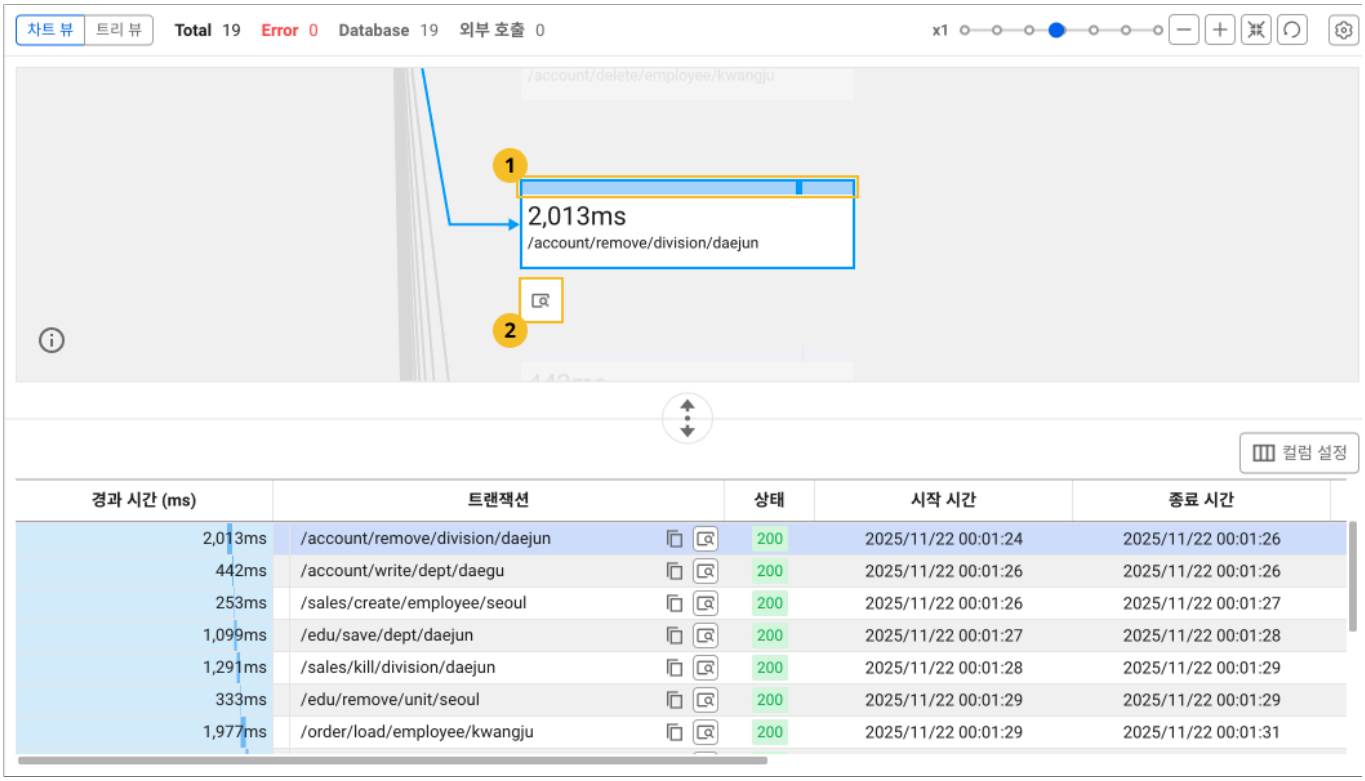
1. Go to **Analysis** > **Multiple Transaction Trace**.
2. Enter the **MTID** value checked in the **Transaction information** window into the **Search by MTID / CUSTID** field.
3. Select the date and project to query.
4. Click the **Apply** button at the bottom of the screen.
    - The **Apply** button will be disabled under the following conditions:
        - When the same filter values (MTID/CUSTID, date, selected project) as the previous search are applied.
        - When none of the filter values (MTID/CUSTID, date, selected project) are entered.
5. The right-side **Chart** tab displays a diagram and a table that show the call relationships between each transaction.

The diagram and table areas are divided vertically, and you can adjust their ratio manually.

- Clicking the upper arrow expands the table to full height.
  If the diagram was previously full height, the layout resets to a 50/50 split.
- Clicking the lower arrow expands the chart area to full height.
  If the table was previously full height, the layout resets to a 50/50 split.

Chart

Chart provides users with a fast and clear view of the call relationships between transactions. You can check the individual execution time of transaction services that share the same **Multi transaction ID**. The elapsed time displayed at the top of each transaction node (1 **Time Bar**) allows you to understand the call flow between transactions.



> You can freely move, zoom in, or zoom out the chart with your mouse.
>
> - Drag within the chart area to move the view.
> - Use scroll inside the chart:
>   - Vertical scroll: move up/down
>   - Horizontal scroll: move left/right
> - Use `Ctrl` + vertical scroll to zoom in/out.
>   (Touchpad pinch gestures are also supported.)

When you select a transaction, only the related nodes remain active, while all unrelated nodes become dimmed.
The **Detail View** button (2) below the selected transaction becomes enabled, and clicking it opens the **Transaction information** window.
There you can view the transaction's detailed trace.
For more information on analyzing transaction traces using the **Transaction information** window, see the Transaction Info Documentation.

- **Zoom In**: Increase zoom level
- **Zoom Out**: Decrease zoom level
- **Fit to Screen**: Adjust all nodes to fit the screen
- **Reset Zoom**: Reset zoom and position to default state
- **Chart View Settings**: Show or hide chart elements
  - **More information**: Toggle application name, project information, IP, and status codes
  - **Database** / **External**: Display database connections or external HTTP Calls as nodes in the chart

Table

In the **Table** tab, you can view each transaction included in the multi-transaction trace in a table format.

When you select a transaction, its corresponding **node** in the **Chart** diagram becomes highlighted, and related nodes are emphasized.
Clicking the detail button in the transaction column will open the **Transaction information** window, where you can inspect the full transaction trace details.
For more information, refer to the Transaction Info Documentation.

- **Column Settings** — Edit visible columns in the table

Tree

When you select the **Tree** button, you can view detailed information for each transaction and its associated traces.
This view visualizes the start time and elapsed time of each sub-transaction or trace within the overall transaction execution time, providing a tree-structured representation of the call hierarchy.

- **Project selection**: Select or deselect the projects displayed in the chart.
- **Single-line View**: Display each execution segment's text in a single line to simplify the tree layout.
- **Multi-line View**: Display each execution segment's text in multiple lines (with line breaks).
- **Longest Path**: Jump to the longest path.
- **Show Time Bar**: Display elapsed time as bar charts.
- **Show Time**: Display timestamps, gaps, and elapsed time for each segment in text form.
  - 8 seconds or more: Displayed in red as an over-delay.
  - 3 to 8 seconds: Displayed in orange as a delay.
  - Less than 3 seconds: Displayed in blue as normal.

> Start time and elapsed time may differ from actual collected values because they are adjusted based on mapping logic that accounts for time discrepancies between parent transactions and their traces.

- **Hide Time**: Hide time information.

> - Selecting the trace's button or button allows you to view detailed information such as **HTTP call statistics**, **Active stack**, or **SQL**.
> - Clicking the button on a SQL step copies the SQL query text.

Endpoint Settings

The end point is the start method of the transaction. In case of the HTTP transaction, `HttpServlet.service()` or `Filter.doFilter()` is the start point of transaction, and this point is called "**transaction end point**."

Non-HTTP tracing

The performance from the start to the end of the method specified as the transaction end point, is called "transaction performance." To trace non HTTP transactions, be sure to specify the **end point**.

> Finding the **end point**
>
> 1. Set the transaction trace for all methods where transactions are supposed to be called.
> 2. After restarting, trigger a transaction for monitoring.
> 3. Turn on the `back stack` option to pinpoint the entry method.

Selecting the target

1. **Method Trace Setting**

   Set the method trace. Be sure to specify a class that is called in transactions. The **JDBC driver** is useful for the programs that use the database.

   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. **Transaction start option setting**

Start a transaction when the method is called. Turn on the transaction start option and the option to dump the stack at the start of a transaction.

whatap.conf

```
trace_auto_transaction_enabled=true
trace_auto_transaction_backstack_enabled=true
```

3. **Trace analysis after restart**

   Restart it. After calling the service, you can see the transaction being traced.

   If you view the transaction, you can see that the methods of all classes starting with `jdbc.*` appear in this transaction. If you view the transaction trace, you can see the message step called **TRANSACTION BACKSTACK**.

   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. **Check Stack Content**

   By checking the stack content, you can infer which method the transaction is departing from.

   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)
   ```

   One of the above three methods can be determined as the start point of a transaction. In this situation, perform reverse compilation to determine the appropriate **transaction end point**.

   Looking at the logic, `while()` runs within `SimulaNonHttp.run` , `SimulaNonHttp.process()` is called, and then `SimulaNonHttp.execute()` is run. We can guess that `process()` is appropriate. This part must be determined by looking at the source.

> The most **important criterion for end point is that it must be ended**. Under normal circumstances, it must terminate immediately without any delay so that the performance evaluation can be made.

Set the transaction endpoint

1. Set the transaction start point as follows:

   whatap.conf

   ```
   hook_service_patterns=com.virtual.web.SimulaNonHttp.process
   ```

2. Restart the application.

The `process()` method becomes the **end point** of a new transaction.

> For the `hook_service_patterns` option, wildcard characters cannot be used. To set multiple values, use comma (,) as the delimiter.

Defining the transaction name

Generally, transactions are identified by the method name.

whatap.conf

```
service_name_mode=[full,class,method,string,arg]
service_name_index=0
```

- **service_name_mode**

  You can specify any of 5 options: `full` , `class` , `method` , `string` , and `arg` .

- `full` : The full class name is used as the service name.
- `class` : The class name is used as the service name.
- `method` : The method name is used as the service name.
- `string` : The first parameter in the text string is used as the service name.
- `arg` : Among parameters, the parameter of the index specified in the `service_name_index` option is used as the service name.

Using the plug-in

> It is recommended to use a plug-in only if you are familiar with it.

Create the plugin folder under the WHATAP_HOME path and move to plugin. Create the AppServiceStart.x file with the `vi` command.

Plugin usage example

If you type `println("test");` and save it, you can see that the string "test" is displayed on the screen. After checking it, extract data from the parameter.

In the following example, `HashMap` is passed as a parameter, and this `HashMap` contains the key called 'url'.

```
Object url =((java.util.HashMap)$point.getArgs()[0]).get("url");
$ctx.service((String)url);
//println("url="+url);
```

If you create a plugin like this, you can change the transaction name.

Collect Stacks

The application monitoring agent provides linear and circular collection methods.

Linear collection

Individual transaction traces cannot collect steps indefinitely. Store trace steps in buffers of limited length for each transaction. For each transaction, the maximum number of steps that can be collected is specified as an option.

- **profile_step_max_count**

  Default `1024`

  Maximum number of steps that can be collected.

- **profile_step_normal_count**

  Default `800`

  Number of steps that are generally collected without any restrictions.

- **profile_step_heavy_count**

  Default `1000`

  If the normal count is exceeded, only the steps with slow response time and the active stack steps are collected.

- **profile_step_heavy_time**

  Default `100`

  The reference time to collect steps within heavy count is `profile_step_heavy_time`.

  step_buffer

  If the number of steps in the profile being collected exceeds the heavy count, only active stacks are collected. Even in this case, the maximum number of steps does not exceed `profile_step_max_count`.

Circular collection

Linear collection collects the front part of the trace and discards the later part, if it exceeds the buffer size. On the other hand, circular collection discards the front step.

```
circular_profile_enabled=true
```

circular_profile

The buffer size is set in `profile_step_max_count` .

```
profile_step_max_count=1024
```

Application Topology

Application topology visually displays the database and external service calls related to the transaction flow. You can clearly see the entire request path—from user request to database calls and external service integrations—making it easy to identify performance bottlenecks.

- Visualize the path from user request → DB/external service
- Filtering based on URL patterns
- Check transaction performance status (slow/error, etc.)
- Identify bottlenecks and view detailed information

Viewing Topology

After selecting the time range and application, configure the search scope using the options below.
Each query opens in a new tab. When hovering over a tab, a tooltip shows the search conditions, and clicking the pop-out button in the tooltip opens the topology in a separate window.

Transaction URL Search

Displays only the domains that called transactions matching a specific URL pattern.
You can use wildcards (`*`) to define patterns.

Examples

- `/api/*` : All URLs starting with `/api/`
- `*/users/*` : All URLs containing `users`
- `/api/v1/users` : Exact match

> ⚠ **Be careful when combining slashes(`/`) and wildcards(`*`).**
>
> - `/api/*` : Does **not** match `/api` ; it requires a path after `/api/` (e.g., 🔵 `/api/users` , ❌ `/api`)
> - `/api*` : Matches `/api` itself

(e.g., 🔵 `/api`, `/api/users`, `/apitest`)

Pattern Limitations

- **Must contain at least one non-wildcard character**
  - ❌ Not allowed: `***`, `**`
  - 🔵 Allowed: `a*`, `*a`, `*a*`, `/api*`
- **Empty string** or null is not matched.
- **Consecutive wildcards (`**`)** behave the same as `*`.
- **Complex patterns** with 3+ `*` are not fully supported.
- **Regular expressions** are not supported.
- **Case sensitive** matching.
- **Cannot escape** `*` to match it literally.

Trace Limit Count

Maximum number of transaction traces (profiles) to retrieve.
Higher limits show more data but increase query time.

- Default: 10,000
- Max: 1,000,000

Even if you increase the limit, you may still not retrieve all matching transactions.
A warning appears when the limit is reached.

Time Limit

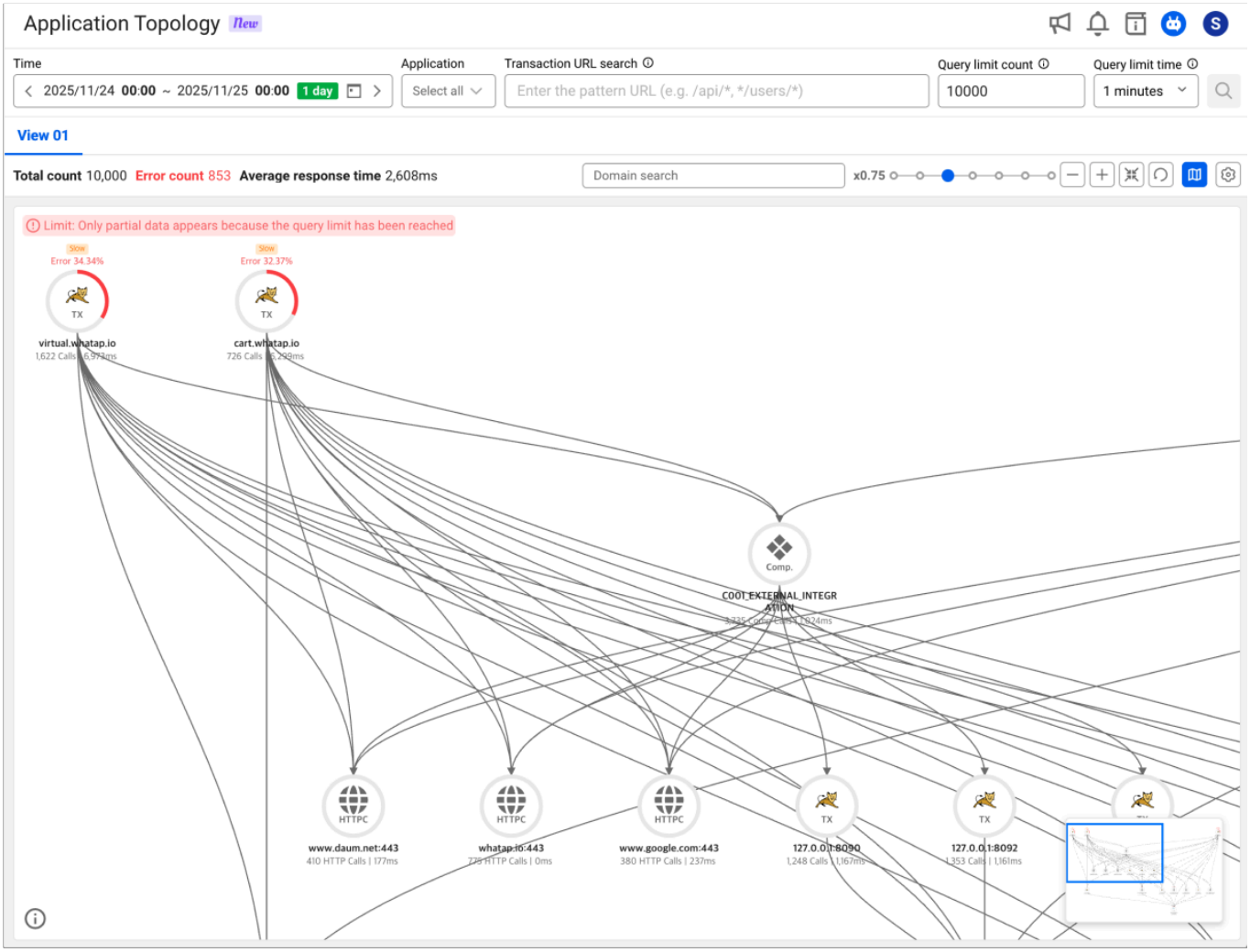
When the time limit is reached, only data up to that point is shown, and a warning appears.

- Range: 30 seconds ~ 5 minutes

- Default: 1 minute

Screen Structure & Visualization

The screen visualizes transaction flow using circular **nodes** and **connecting lines**.



Node Structure

Nodes are arranged vertically based on transaction execution flow: upper nodes call lower nodes.

Node Types

Nodes are arranged in order: User → TX → DB/HTTPC.

Node Type	Description
User	Number of users who invoked the transaction
TX (Transaction)	Application transactions (HTTP domain/URL)
DB (Database)	Database queries and connections
HTTPC (HTTP Call)	External HTTP calls

Node Visual Elements

- **Error Gauge**: Circular error-rate gauge on node border
- **Performance Tags**: "Slow" (orange) / "Very Slow" (red)
- **Hover Effect**: Gray border + highlight connected edges
- **Selected Effect**: Blue border + highlight connected edges

Connecting Lines

Color changes based on average elapsed time:

- **Gray**: Normal
- **Orange**: Slow (≥ 3 sec by default)
- **Red**: Very Slow (≥ 8 sec by default)

Summary Metrics

Displayed at the top:

- **Total Count**

- **Error Count**
- **Average Response Time**

Screen Interaction

Category	Action	Description
Scroll & Zoom	Mouse wheel	Vertical scroll
	Shift + wheel	Horizontal scroll
	Ctrl + wheel	Zoom in/out
Drag	Canvas drag	Move entire topology
	Node drag	Adjust individual node position
Node Interaction	Hover	Highlight connected paths
	Click	Select node + show details
	Re-click	Deselect
Zoom Controls	Zoom in/out	Range 0.1x ~ 2.0x
	Fit to screen	Auto-fit
	Reset	Reset zoom
	Slider	Fine zoom control
Export	PNG Download	Export current chart
Mini-map	Map view	Shows current viewport
	Toggle icon	Show/hide

Category	Action	Description
Domain Search	Input search	Highlight matching nodes

Display Settings

Node Labels

- Call count
- Error count (percentage or number)
- Average elapsed time
- Name (domain, DB URL)
- Type (TX, DB, HTTPC, User)

Edge Labels

- Display mode: Always / On hover or click
- Call count
- Error count / error rate
- Average elapsed time (color-coded)

Other Options

- Edge animation for slow paths
- Show/hide User nodes
- Hide isolated nodes
- Merge identical nodes
- Merge HTTPC and corresponding TX nodes

Slow Thresholds

- **Slow** (orange): default 3 sec
- **Very Slow** (red): default 8 sec

Node Details

Tooltip

- Node type & name
- Call count
- Error count / rate
- Average elapsed time
- For merged nodes: top 10 source-domain stats

Detail Drawer

- Performance metrics
- Merged-node performance table
- Caller transaction list (up to 1,000)
  - Status filter
  - Text search
  - Sorting
  - CSV download
  - Click row → trace popup

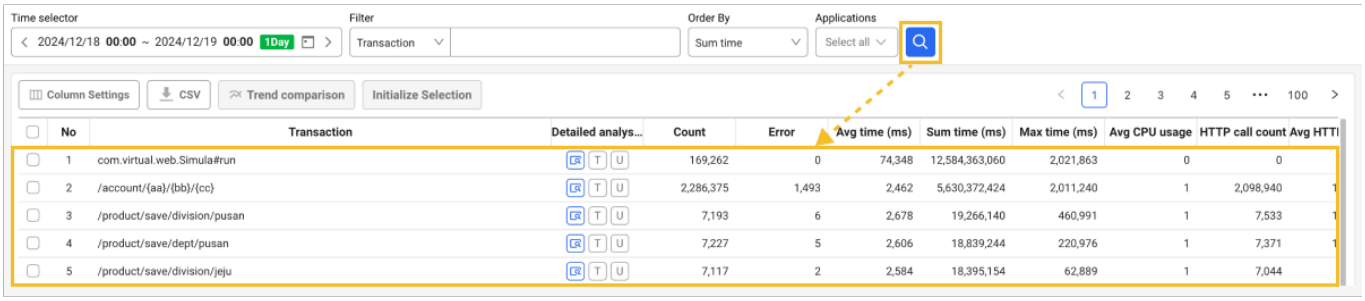
Transaction Analysis

Home > Select Project > **Statistics**

Select a project in the initial screen of the WhaTap monitoring service, and then select a sub menu under **Statistics**. The WhaTap agent collects statistical data for the key traces such as transaction information and SQL. A list is created every 5 minutes and it is sent to the server.

Basic screen guide

Set the **Time selector**, **Filter**, **Order By**, and **Applications** and then select .



- Each **Statistics** tab provides sorting options by **Sum**, **Max**, **Time avg**, and **Count** for viewing the list from your desired perspective.
- You can add or exclude columns to view from the list as well as the default values by using the **Column Settings** button.
- When you select , the CSV button appears above the search list along with the search results. If you select the button, you can download the searched list as a CSV file.

  > If you click CSV without selecting after a statistics list appears, you can download the CSV file with the changed conditions.

- If you select the **Display Detail** icon in the statistics list, you can check the number of cases and average time chart, and additional charts are provided to compare the case count, error count, and response time. If you select the time to search in the chart, the **Transaction Search** menu appears.

- If you select T in the statistics list, the **Top stack of URL** menu appears.
- If you select U in the statistics list, the **Unique stack URL** menu appears.

> ⓘ • You can view the data for up to 31 days. However, you can view the data for up to 7 days in **Client IP**.

Transaction statistics

Home > Select Project > **Statistics** > **Transaction**

It collects statistics for transactions. Every 5 minutes, up to 5,000 URL-specific performance statistics are collected and sent to the server. If the number of URLs for 5 minutes exceeds 5,000, this is ignored.

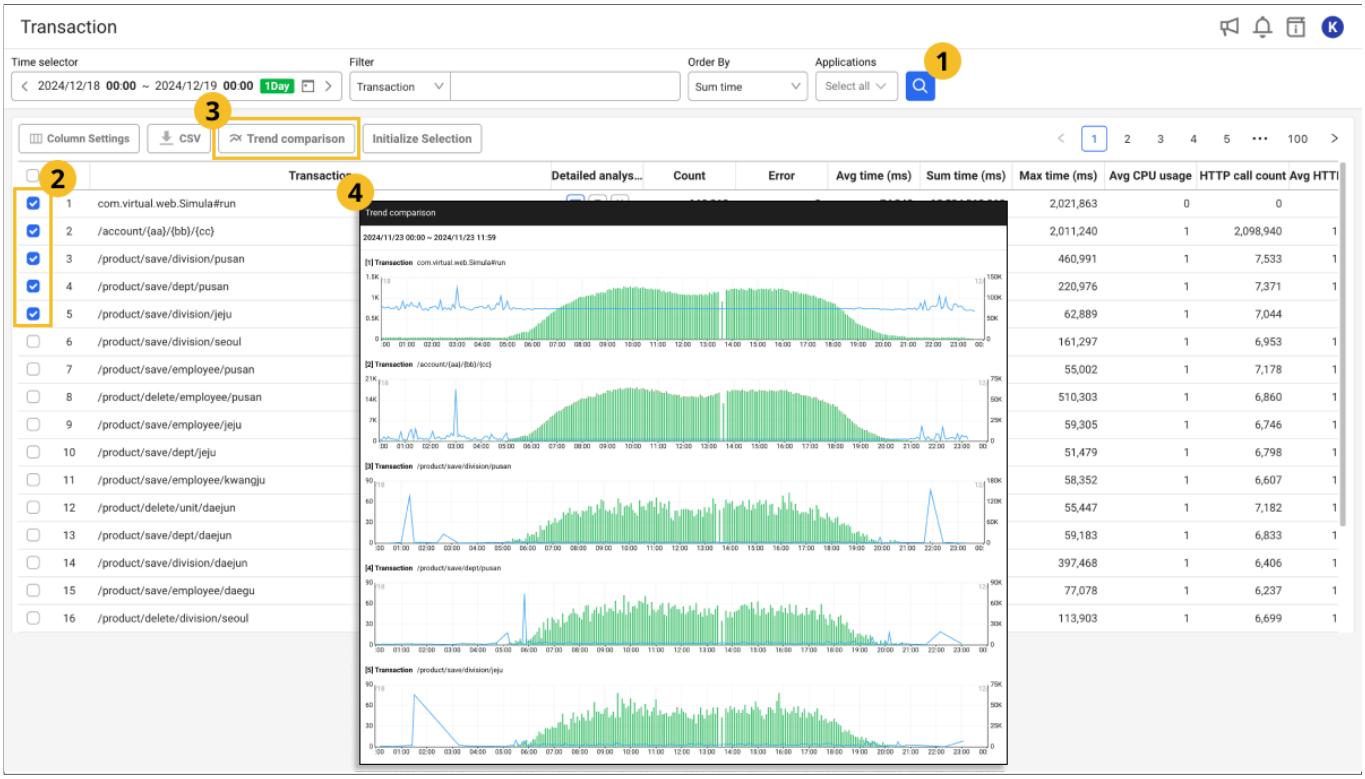
Columns	Description
Transaction	Transaction address
Count	Number of execution cases
Error	Error occurrence count for the executed transactions
Avg time (ms)	Average of the total response time
Sum time (ms)	Sum of response times
Max time (ms)	Maximum response time
Avg CPU usage	Average of the CPU usage time
HTTP call count	HTTP Call Count
Avg HTTP call time (ms)	Total HTTP call time / Number of transactions
Average time per HTTP call (ms)	Total HTTP call time / Total number of HTTP calls in transactions
SQL count	SQL execution count

Columns	Description
Avg SQL time (ms)	Total SQL time / Number of transactions
Average time per SQL (ms)	Total SQL time / Total number of SQL patches in transactions
SQL fetch	SQL Fetch Count
Avg SQL fetch time (ms)	Total SQL patch time / Number of transactions
Average time per SQL patch (ms)	Total SQL patch time / Total number of SQL patches in transactions

> ⓘ **Avg memory allocation** requires to set the `trace_malloc_enabled=true` option in the agent configuration (whatap.conf) for collection. The default value is `false`.

Using the trend comparison feature

You can use the **Trend comparison** feature to compare multiple transaction trends at a glance. This feature is useful for quickly identifying key issues and resolving traffic issues when there are a large number of transactions to check.



1. Select the time and monitoring target for a query, and then select the .
2. Select one or more transactions to compare from the transaction list.
3. Select **Trend comparison**.
4. The **Trend comparison** window appears listing the trend graphs for the selected transactions.

Check the trend graph in the **Trend comparison** window to compare traffic patterns between transactions. You can compare traffic distributions for specific time periods and service paths.

- You can select and view up to 100 transactions.
- If there are a lot of transactions selected, the loading time may increase.
- For accurate analysis, use the **Filter** option appropriately.

Transaction domain statistics

Home > Select Project > **Statistics** > **Domain**

The WhaTap agent can collect transaction statistics for each domain. In the systems where multiple domains are separated and serviced on a server, domain-specific analysis may be required. Set the following option:

whatap.conf

```
stat_domain_enabled=true
```

Through the collected data, you can identify the processing status of URLs for each domain.

Columns	Description
Domain	Service domain
Transaction	Transaction address
Count	Number of execution cases
Error	Error occurrence count for the executed transactions
Time avg (ms)	Average of the total response time
Time sum (ms)	Sum of response times

Transaction caller statistics

Home > Select Project > **Statistics** > **Caller**

If the multi server is connected by a rest call, the relationship statistics between **callers** and **callees** can be collected. To collect this data, set the following options:

whatap.conf

```
mtrace_rate=100
mtrace_spec=v1
stat_mtrace_enabled=true
```

msa-system

In the above architecture, the **Caller & Callee** statistics can be viewed only in api1 and api2. The system called by the user browser cannot see the **Caller** statistics.

However, because the server on the **caller** side must transfer data, the relevant settings must be entered for all servers.

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
```

The statistical data collected is as follows. The data must be viewed from the **callee** side.

Columns	Description
Caller Pcode	Project code corresponding to the caller
Caller agent kind	Group unit to which the agent corresponding to the caller belongs
Caller URL	Address of the caller transaction
Transaction	Callee transaction address
Count	Number of execution cases
Error	Error occurrence count for the executed transactions

Columns	Description
Time avg (ms)	Average of the total response time

ⓘ The maximum number of CSV lines that can be downloaded at one time from the **Caller** menu is 10,000.

IP transaction statistics

Home > Select Project > **Statistics** > **IP Transaction**

It provides the feature to view the URLs and number of calls from applications for each specific IP address. This feature is useful for identifying potential security threats or performance issues by checking IP addresses with unusually high request volumes and analyzing the URLs called from those IP addresses. You can block unnecessary or malicious requests based on this information.

You can filter IP addresses or transaction addresses through the **Filter** option to quickly retrieve data that meets your criteria. This is useful when analyzing specific IPs or transaction address patterns.

The following option needs to be set to collect IP transaction statistics. The default value is `false`.

whatap.conf

```
stat_ipurl_enabled=true
```

Columns	Description
IP	IP address of the client that made the request.
Transaction	Addresses of the transactions called and specific path variables are marked with `{}`.
Count	Total number of requests made from the IP address.
Error	Number of errors occurred for requests.

Columns	Description
Time avg (ms)	Average response time for the request (milliseconds, ms).

- **Abnormal traffic detection**: It can detect IPs with an abnormally high number of calls, analyze transaction address patterns from those IPs, and block potential attacks. For example, if there are excessive requests to the `/account/{aa}/{bb}/{cc}` URL from a specific IP, you can block the IP or take action to limit requests.
- **Response time analysis**: It can detect IPs with abnormally high average response times to identify and improve performance issues on the network or server side.
- **Error occurrence analysis**: When multiple errors occur from a specific IP, you can diagnose whether there is a problem with the requests from the IP or a problem with the system.

Transaction referer statistics

Home > Select Project > **Statistics** > **Referrer**

The WhaTap agent can collect transaction statistics for each referer. Set the following option: The default value is `false`.

whatap.conf

```
stat_referer_enabled=true
```

Through the collected data, you can see the referer data on the statistics.

Columns	Description
Referer	Referer address
Transaction	Transaction address
Count	Number of execution cases

Columns	Description
Error	Error occurrence count for the executed transactions
Time avg (ms)	Average of the total response time
Time sum (ms)	Sum of response times

Error Statistics

Home > Select Project > **Statistics** > **Error**

They are statistics on the service errors that occurred for 5 minutes. They collect different errors and transaction names with the keys. They provide the stacks upon error occurrence. The statistics appear for up to 1,000 errors every 5 minutes.

Columns	Description
Class	Error Class
Transaction	Transaction address
Message	Error Message
Count	Error Count
Agent	Agent name
Agent node	Group unit categorized as `whatap.onode`
Agent kind	Group unit categorized as `whatap.okind`
Initial creation time	Statistical time (5 minutes or 1 hour) when the transaction error was first generated during the query period

SQL execution statistics

Home > Select Project > **Statistics** > **SQL**

The SQL execution statistics for 5 minutes are collected. A maximum of 5,000 SQL statements are allowed in 5 minutes. If a single Java process generates SQL statements that exceed the limit, they are discarded from statistical data.

Columns	Description
Database	Database Type
SQL	SQL query statement
Transaction	Data randomly collected from one of the URLs that called the SQL
CRUD	Create, Read, Update, Delete
Count total	Query execution count
Error count	Number of errors among executed queries
Avg time (ms)	Average of the total query execution time
Sum time (ms)	Sum of query execution times
Std dev	Standard deviation for the query execution time

ⓘ Transaction collects one of the URLs (last call URL) that called the SQL for 5 minutes for analysis.

HTTP call execution statistics

Home > Select Project > **Statistics** > **Remote HTTP Call**

They collect the statistics of calls outside the applications for 5 minutes via the HTTP and HTTPS protocols. A maximum of 5,000 HTTP call statements are allowed in 5 minutes. If a single application process generates external HTTP call statements that exceed the limit, they are discarded from statistical data.

Columns	Description
HTTP call URL	HTTP call URL hash
Transaction	Data randomly collected from one of the URLs that requested the HTTP API
Host	Host or IP address
Port	TCP port
Total	Total execution count
Error	Error Count
Time avg (ms)	Average of the total response time
Time sum (ms)	Sum of response times
Time min (ms)	Minimum response time
Time max (ms)	Maximum response time
Std dev (ms)	Standard deviation for the HTTP call time

Number of calls for each client IP region

Home > Select Project > **Statistics** > **Client IP**

The physical distance between the server and the client is proportional to the data transfer time. It statistically collects call counts for each region of the client IP. The number of different IPs that can be collected per 5 minutes is up to 70,000 per instance.

Columns	Description
Client IP	Client IP address
Country/Region	User access country
City	User access city
Count	Total call count

- The data about the client IP can be viewed for up to 7 days.
- WhaTap basically stores the client-related information. For more information about the agent settings for user data collection, see the following.

Share chart by region

The share chart collects user access locations and presents them for each country and city.

- If a country is selected, the detailed chart appears to display each country's cities and connected user IPs.
- If you select the center circle, it goes to the country selection chart.

Call counts for each user agent

Home > Select Project > **Statistics** > **User-Agent**

The numbers of calls for each hash of the user agent string are collected. There is a maximum of 500 different user agent hashes for each instance that can be collected every 5 minutes.

Columns	Description
UserAgent	User agent of the browser
OS	OS environment in which the browser runs

Columns	Description
Browser	Browser type
Count	Total collection count

Client Browser Statistics

Home > Select Project > **Statistics** > **Client Browser**

By analyzing the user agent strings, it provides the statistical information based on the browser type and OS used by the end users. You can check the market share through a pie chart created based on the collected statistical information.

Columns	Description
UserAgent	User agent of the browser
OS	OS environment in which the browser runs
Browser	Browser type
Count	Total collection count

Java Reports

Home > Select Project > **Report**

Report is a document that reports the service usage status and failure records for individual projects. The persons in charge of service monitoring share the status report with the people of related departments.

The monitoring data analysis is important because it serves as a metric for determining the direction of improvement. However, it is cumbersome to aggregate and document data from multiple dashboards.

WhaTap's **Report** menu supports **automated report creation**. Through the **Report** menu, you can create weekly reports, share reports at the set times, and manage various templates.

> To download, print, or schedule to send **reports** or emails, click the button. Downloading is available in the HTML format.

Difference between the report and integrated report

- **Report** is the menu that generates a report for a single project. You can select a project and then enter the **Report** menu in the side menu.
- **Integrated report** is the menu to generate a report for multiple projects. You can enter it from the side menu on the Home screen without selecting a project. For more information about **Integrated report**, see the following.

Report types

The basic report is a form that is actually used by large enterprises, government agencies, and IT service companies. If you have any desired form, please send a request to support@whatap.io. The requested report template is to be added in the **Report** list.

Add Report Template

- If the requested form is universal, it is additionally provided to the **Report** list.
- Addition is easy when HTML and CSS forms are provided upon request.

Requested form example

When requesting an additional report form, it is recommended to attach the desired HTML and CSS forms as shown in the example. See the following example file for more details.

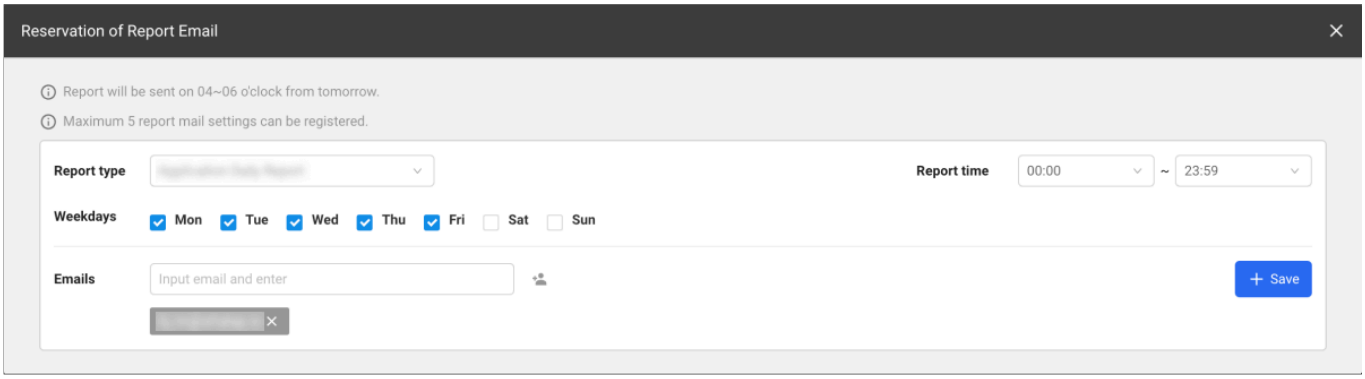
- Example file: After downloading, unzip the ZIP format to see the example file in the HTML format. Instructions to CSS and HTML areas for customization are described in the comments in the file.

Constraints

- If you need to add, modify, or delete each item in the report template provided by WhaTap Monitoring, it can be provided.
- To add data or analyze the results that are not included in the provided report template, review is required by developers.
- Data addition and processing is limited to the data (metrics) collected and stored by WhaTap Monitoring. Other tools and external data cannot be added.
- It is difficult to provide a non-universal report form that meets the needs of a specific client company.
- For more information, contact support@whatap.io or the sales representative.

Email delivery schedule

To receive regular reports via email, select **Reservation of Report Email**. You can replace various service checking tasks performed right after going to work with email confirmation.


Reservation of Report Email
Report will be sent on 04~06 o'clock from tomorrow.
Maximum 5 report mail settings can be registered.
Report type
Report time
00:00
~
23:59
Weekdays
Mon
Tue
Wed
Thu
Fri
Sat
Sun
Emails
Input email and enter
Save

APM Report

Home > Select Project > **Report**

If you click ⌄ in **Report** on the screen, you can see various analysis reports classified by daily, weekly, and monthly. Select a desired report format, and then set the date and time to view and filtering options. You can check the selected report by selecting .

> ⓘ When generating the weekly and monthly reports, it may take some time depending on the scope of the data view.

Daily Report

It provides the report to check changes in application performance for the day. The provided report form as a daily report is as follows:

- **Application Daily Report**
- **Application Peak Performance Comparison**
- **Application User Statistics**
- **Daily Metric Trend Charts(Avg/Max)**
- **Application Daily Custom Report**

Application Daily Report

This report provides a comprehensive analysis of key metrics for application performance for the day.

> ⓘ For more information about key terms and metrics, see the following.

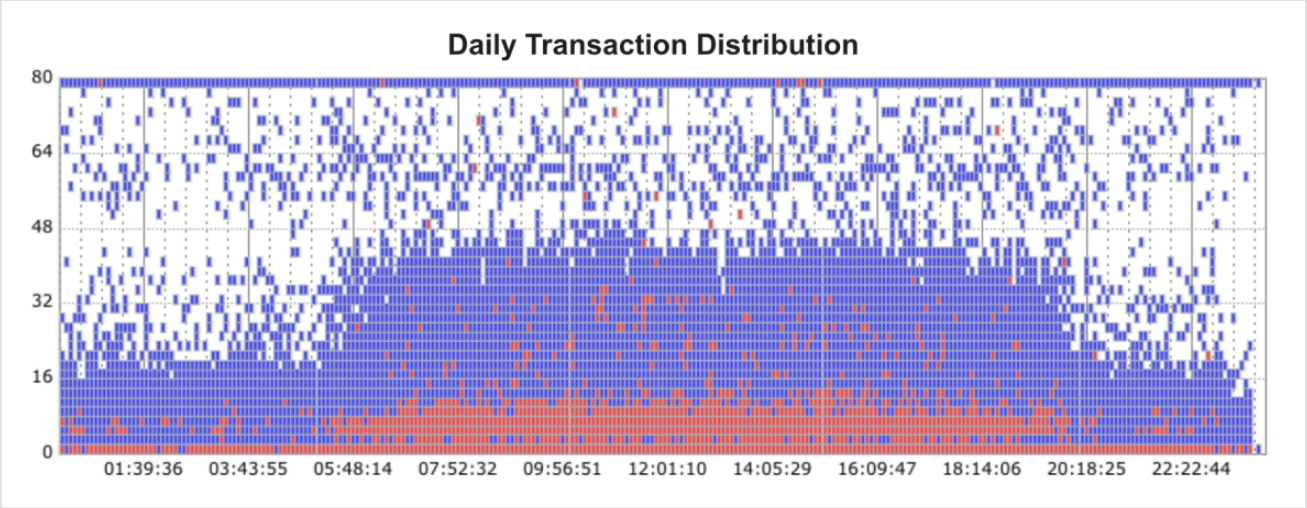
- **Intro**: You can see the key performance metrics of the application.
- **PeakTime Summary**: You can see the time of the day when the throughput (TPS) was the highest and the

metric value of that time.

- **Average/Maximum Value by Metrics**: You can check the average values for each key metric for the day and the maximum value based on the time of occurrence.
- **Trend Chart**: This time series chart visually displays key metrics of the system. You can check the numbers of key metrics over time. It is an important metric in evaluating the system stability and performance.

> ◦ **Application Weekly Report** provides additional daily changes.
>
> ◦ On the upper right of the chart, select CSV to download the data in a CSV file format.

- **Daily Transaction Distribution**: You can see the distribution of transactions and error occurrences by time zone for the day at a glance. The x-axis of the graph represents the end times for each transaction, and the y-axis represents the response times (seconds). Normal transactions are displayed in blue, and erroneous transactions are displayed in red.



- **Application Summary**: You can check the status for each monitored agent. **Active Count** indicates the number of active transactions.
- **Transaction TOP 10**: Based on the **Transaction Count**, **Error Count**, and **Response time average**, you can see the top 10 transactions.
- **Top Stack TOP 10**: Based on the steps on the stack trace, you can view simplified data calculated as call frequency rates between steps.

> In **Analysis** > **Stack** > **Top stack**, you can view the stack information details.

- **SQL Time Average TOP 10**: This is the statistical data sorted by the average response time for the used SQL query statements.

  > - **Increase %**: This figure is calculated for the percentage increase by comparing the average data for three days up to the day before the search and the data of the search.
  >
  >   e.g. `4/25 00:00` to `23:59` data inquiry → comparison with `4/22 00:00` to `4/24 23:59` data
  >   If the figure is greater than 10%, it appears in red, and it is between 5% and 10%, it appears in orange. The growth rates less than 5% are not displayed.
  >
  > - In **Statistics** > **SQL**, you can view the occurrence time and performance for each query.

Application Peak Performance Comparison

For each monitored agent, the application's maximum performance metric appears by comparing with the data for the day (red), the previous day (blue), and the previous week (blue). Each metric displays the maximum value in a specific time zone and the metric value for the corresponding time. This allows you to see performance changes for each application at a glance.

> For more information about key terms and metrics, see the following.

Application User Statistics

This report provides a time series chart of the numbers of concurrent users at intervals of 10 seconds, 5 minutes, and 30 minutes and the number of visitors per hour. This report can provide valuable data to help you understand usage patterns and effectively manage user traffic.

- **10s Concurrent User**: You can closely monitor real-time traffic changes to identify the time zone when the number of users suddenly increased. You can also establish a strategy to provide content and services during the time zone when users are most concentrated.

- **5m / 30m Concurrent User**: Server resources and network bandwidth can be efficiently distributed by analyzing short- to medium-term user patterns. You can also predict the peak time zone and plan to allocate more resources to the period.
- **1h Visitor**: You can analyze the user visiting patterns over time and use them in marketing strategies and operation plans.

Daily Metric Trend Charts(Avg/Max)

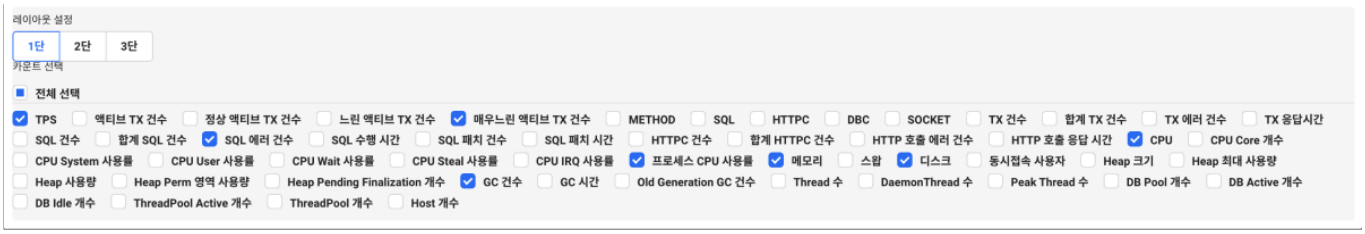
This report provides various performance metrics for applications in a time series chart. The average and maximum values for each metric are visually displayed. This allows you to monitor the system performance trends and use the data needed for performance optimization and troubleshooting.

The title of the chart is in the **Category[Field]** format. For more information about the application performance metrics, see the following.

- **Avg.**: It is distinguished by the blue line. It displays the average values for each time zone.
- **Max.**: It is distinguished by the red line and it represents the maximum values for each time zone.

Application Daily Custom Report

You can create a report by selecting the desired metrics. You can create reports that provide time series charts based on the selected metrics.



1. In **Report** on the screen, select **Application Daily Custom Report**.
2. In **Layout setting**, set the desired number of columns.
3. Select desired metrics. To select all, click **Select all**.
4. Select .

The report is generated.

Application Weekly Report

This report can provide a comprehensive analysis of key metrics for applications during the week.

> ⓘ For more information about key terms and metrics, see the following.

- **Intro**: Provides the total number of visitors and transactions for the week.
- **Operating Rate**: This is the percentage of time when the application ran during a specified period of time. For example, an uptime of 100% means that the application continued to run without a single interruption during the search period. On the other hand, a low uptime rate indicates that the application stopped or abnormal operations happened during the search period.
- **Trend Chart**: This time series chart visually displays key metrics of the system. You can check the numbers of key metrics over time. It is an important metric in evaluating the system stability and performance.

  > ⓘ
  > - It provides daily variations and daily average and maximum values.
  > - On the upper right of the chart, select CSV to download the data in a CSV file format.

- Based on the **Transaction TOP 10 Comparison**: **Transaction Count**, **Error Count**, and **Response time average**, the top 10 transactions can be seen. It provides the number of executions, increases, and decreases for each transaction from a week ago and the week of inquiry.
- **Top Stack TOP 10 Comparison**: A list of top 10 stack traces by comparing the percentage and number of cases from the current week with those from a week ago. You can identify the most frequently called stack.

  > ⓘ In **Analysis** > **Stack** > **Top stack**, you can view the stack information details.

- **SQL Time Average TOP 10 Comparison (Over 100 execute count)**: This list has been created after comparing top 10 SQL statements based on the SQL average response time. For each SQL statement, it provides the number of executions, average response time, and increase/decrease time for a week ago and the week of the search.

- **Errors in this Project(included Error Msg)**: This statistical information provides the error statistics for the week's projects based on the error class and messages.

Application Monthly Report

This report provides a comprehensive analysis of key metrics for the application performance during the period of one month.

> ⓘ For more information about key terms and metrics, see the following.

- **Intro**: You can see the key performance metrics of the application.
- **Application Summary**: You can check the **IP Address**, **OS Information**, and **Operating Rate** for each agent.
- **Trend Chart**: This time series chart visually displays key metrics of the system. You can check the numbers of key metrics over time. It is an important metric in evaluating the system stability and performance.

  > ⓘ On the upper right of the chart, select CSV to download the data in a CSV file format.

- Based on the **Transaction TOP 10 Comparison**: **Transaction Count**, **Error Count**, and **Response time average**, the top 10 transactions can be seen. It provides the number of executions and increases and decreases for each transaction from one month ago and for the current month.
- **Top Stack TOP 10 Comparison**: This is a list comparing the percentage and number of views for the top 10 stack traces from a month ago and for the current month. You can identify the most frequently called stack.

  > ⓘ In **Analysis** > **Stack** > **Top stack**, you can view the stack information details.

- **SQL Time Average TOP 10 Comparison**: A list comparing the top 10 SQL statements based on the SQL average response time. For each SQL statement, it provides the number of executions, average response time, and increase/decrease times for one month ago and the current month.

Guide to key terms and metrics

The following lists the key metrics displayed for each report.

Item	Description
Visitor(DAU) (**DAU**, Daily Active User) / **Total Users**	It is the number of daily active users or number of users who used the application for a day.
Weekly User	Total number of visitors who used the application for a week
Monthly Active Users (**MAU**, Monthly Active User)	Number of unique users who used the application at least once in a month
Concurrent User	Number of concurrent users with duplicates removed
Users / **Realtime Visitor**	Number of users who accessed the application at a specific time zone
Transaction Count	Number of all requested transactions
Total Transaction / **Daily Transaction Count**	Total number of transactions processed for a day
Weekly Transaction Count	Number of transactions requested for a week
Daily Transaction Distribution	This distribution chart allows you to see the distribution of transactions and errors by time zone for a day at a glance.
Error Ratio(%)	Error rate out of the total number of transactions
Event Count	Number of metrics events in Critical level
Error Count	Number of failed transactions due to errors encountered while processing transactions

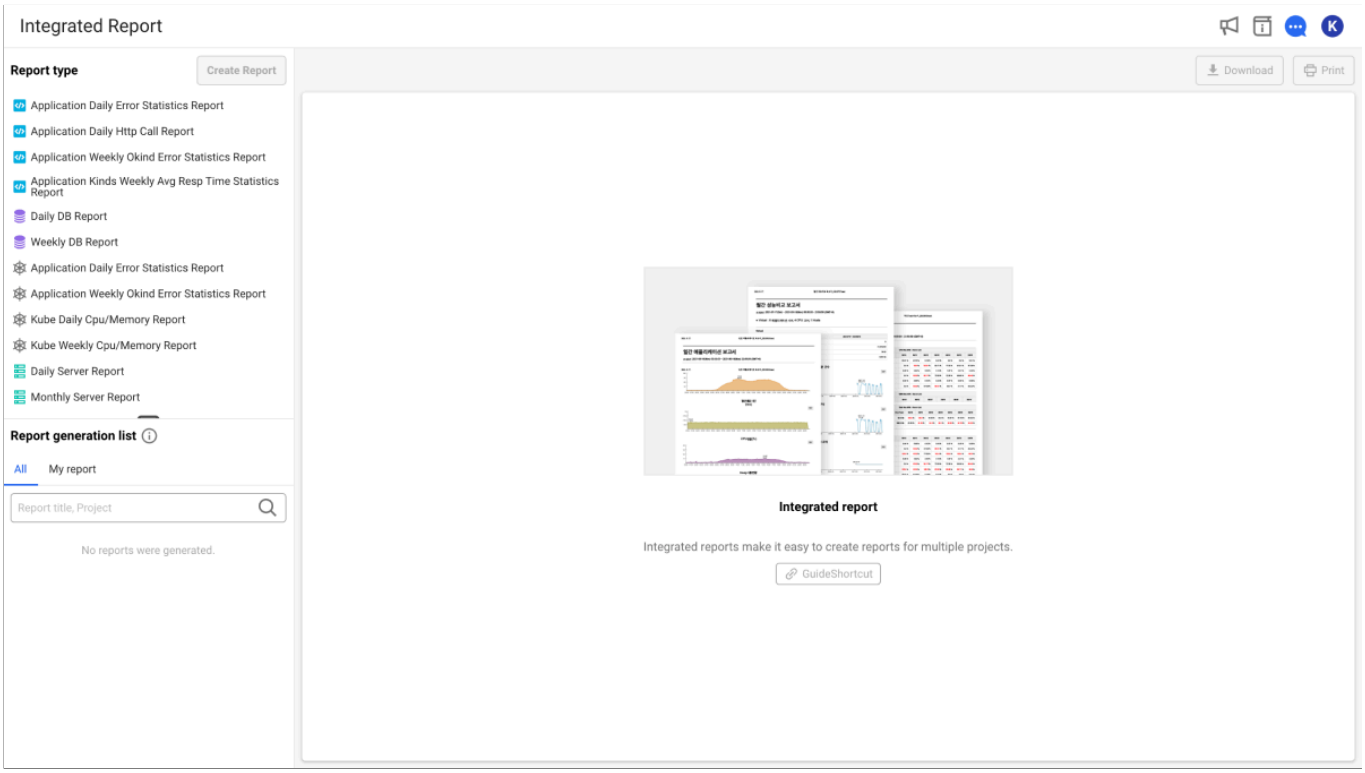
Item	Description
Application Count	Number of applications (number of agents) monitored during the search period
Total Core	Number of CPU cores used during the search period
APDEX*100** (Application Performance Index)	It indicates the customer performance satisfaction with the application for the query period.
TPS (**TPS**, Transaction Per Second)	Number of transactions processed per second
Avg Response(ms) / Response Time	Average response time for transactions
CPU Core	Number of used CPU cores
Visitor Per Hour	Number of visitors per hour during the search period
Server CPU Used Percent	CPU utilization of the server
Server Memory Used Percent	Memory utilization of the server
Active Count / Active TX Count	Number of transactions in progress
CPU	CPU utilization
Heap(Min)	Minimum heap memory usage
Operating Rate	This is the percentage of the normal operation time for the application during a given period of time.

- **Java CPU Used Percent**: CPU utilization of Java applications
- **Java Heap Memory Usage**: Amount of memory allocated in the JVM's heap memory area.

Integrated Report

Home > **Integrated Report**

You can create and share reports that have integrated multiple projects. You can also create a report by selecting a report type and the projects to include, and check the task progress in the creation list. If an error occurs in a specific project, you can see the error details. The reports that have been created can be viewed immediately or shared with the download and print features.



> The **Integrated Report** menu can generate only the reports for projects of **Application**, **Server**, **Kubernetes**, and **Database** products.

Guide to major features

The **Integrated Report** menu provide the following features:

- **Supporting multiple projects**

  It is convenient to be able to generate reports at once by selecting multiple projects. The **Report** menu available in any project can generate reports only for a desired project.

- **Create Report by Kind**

  In the **Create Report** feature, you can select projects by the work unit to save as a template. This feature is useful in the Micro Service Architecture (MSA) environment where projects are usually divided by the work unit.

- **Big Data Optimization**

  With an aid of this, the report creation time has been drastically reduced. Even if an error occurs while creating a project report, you can view the result of the report.

- **Simultaneous operation during report generation**

  You can do other tasks while generating a report. In the **Report** menu available in any project, you must wait until the report generation is completed.

  In **Integrated Report**, you can go to another page while creating a report. You can also see its progress and view the created report from the report result list.

- **Sharing the report work**

  Any user with project permission can view the created reports. You can check whether the related report has already been created, so you can avoid creating duplicates.

Difference between the report and integrated report

The **Report** menu available in any project generates reports for the project. You can access it through **Report** or **Stat/Report** in the side menu after selecting a project on the initial screen.

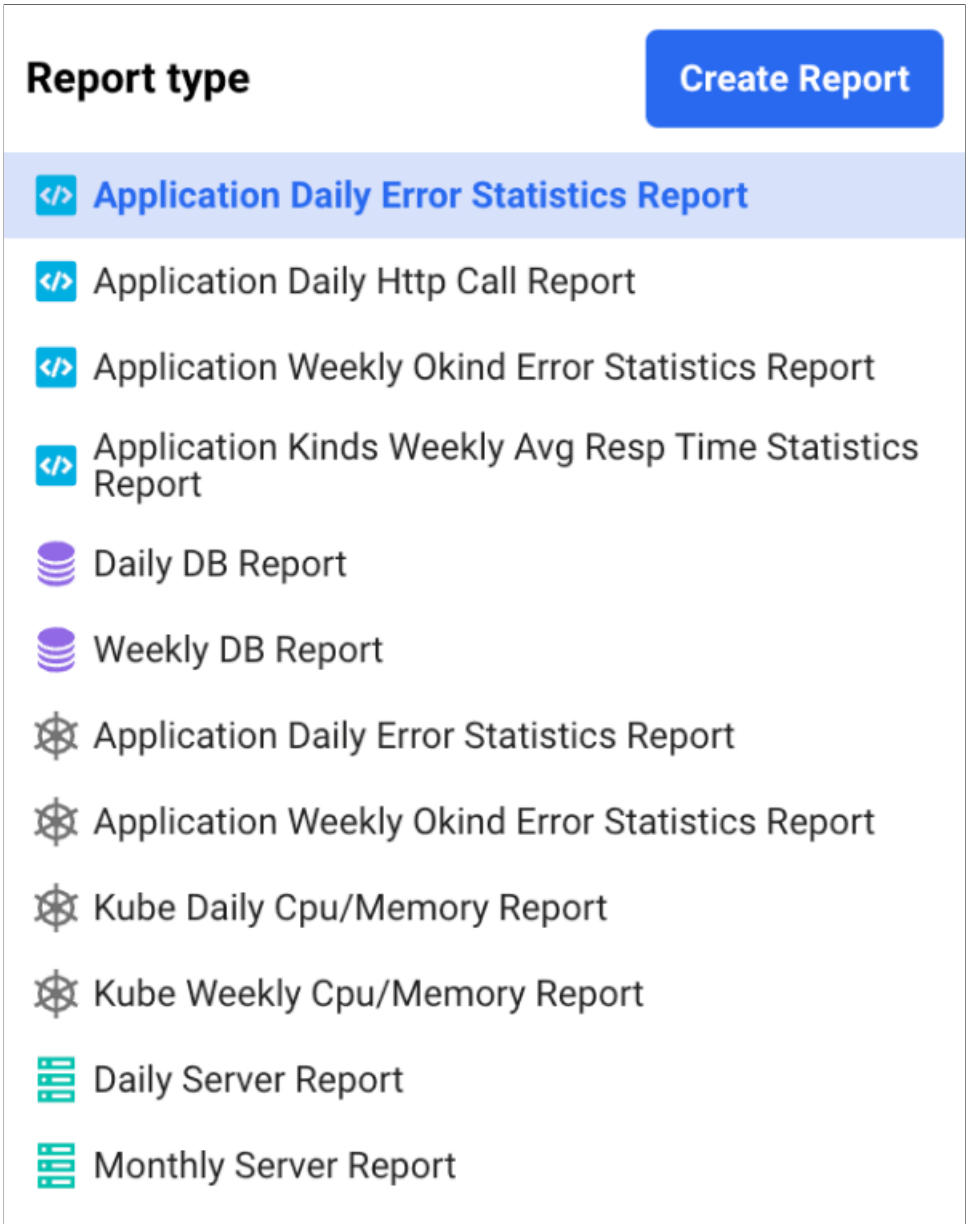
The **Integrated Report** menu can generate reports for multiple projects. It can be accessed from Home screen without selecting a project.

Create Report

Selecting the report type

If you select a desired report type in the **Report type** list, the **Create Report** button is activated. The icon before the report name indicates the project platform that supports report creation.

- indicates the **Application** project platform.
- indicates the Database project platform.
- indicates the **Kubernetes** project platform.
- indicates the **Server** project platform.



Configuring in detail

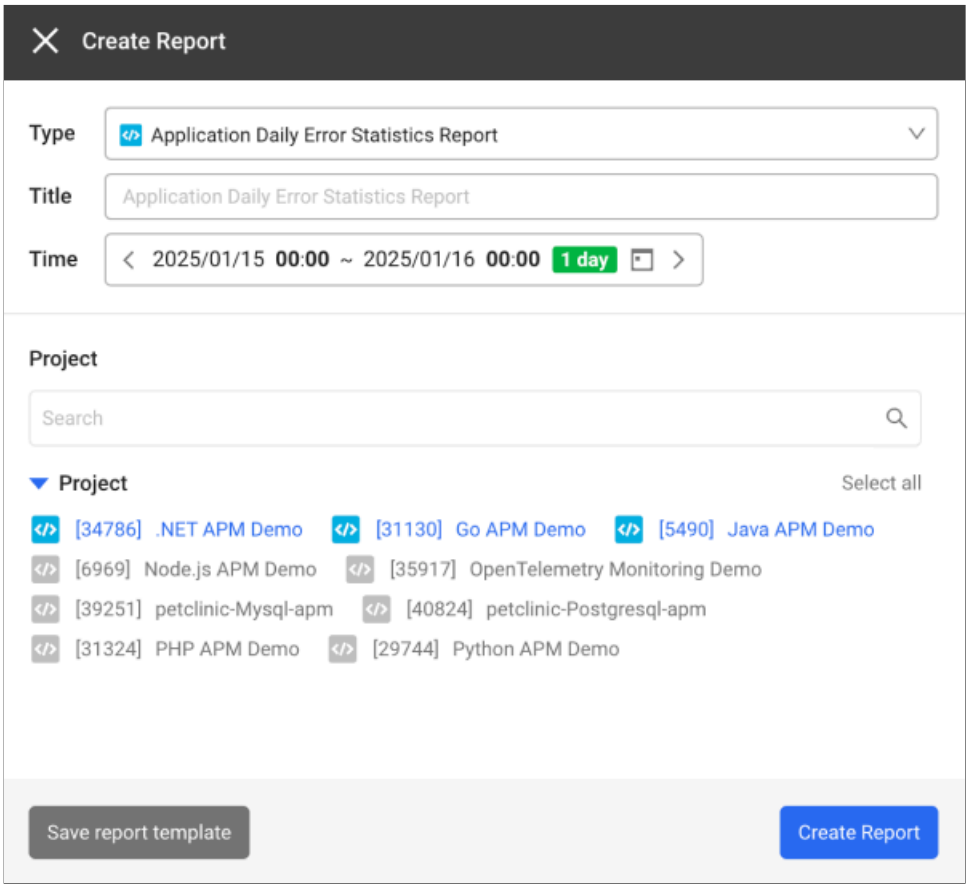
When the **Create Report** window appears, set each item.

- **Title**: If there are multiple reports of the same type, the report names make it easier to distinguish the report results.
- **Time**: Select a period for the data to be used in the report. The

period differs depending on the report type (daily, weekly, monthly).

- **Project**: Select a project to create a report. You can select a group or project. The desired project can be searched via the **Search** field.
- **Save report template**: You can save the settings of the current report as a template. You can load it from the **Type** list when creating a new report.

After configuration, select **Create Report**.



The methods how to set the query period depending on the format of the selected report are as follows:

- **Daily report**: You can select the start/end date and time.

- **Weekly report**: Simply select a start date. Then the end date is selected automatically after calculation.
- **Monthly report**: Simply select a start date. Then the end date is selected automatically after calculation.
  - If the start date is the 1st date, up to the end of the month is automatically selected.
  - If the start date is not the 1st date, up to the date before the next month is automatically selected.

Example 1: If the start date is **January 1**, from **January 1 00:00:00** to **January 31 23:59:59** is automatically selected.

Example 2: If the start date is **January 10**, from **January 10 00:00:00** to **February 9 23:59:59** is automatically selected.

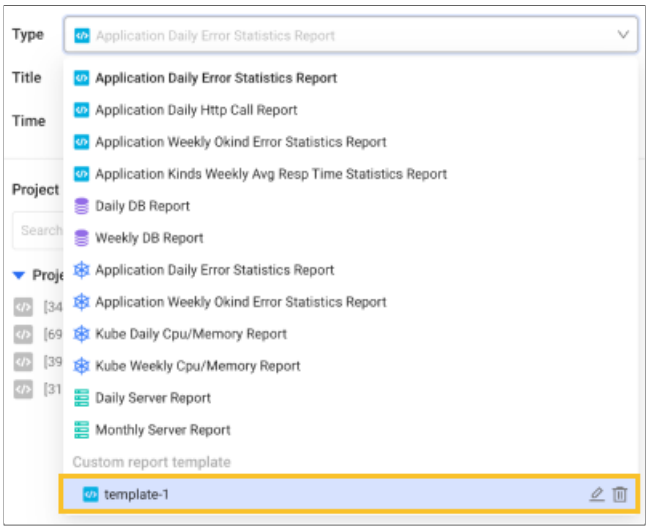
- For more information on how to use the **Time** option, see the following.
- For more information on how to load the saved report forms, see the following.

Loading the report forms

When creating a new report, the form saved with the **Save report template** feature can be checked in **Type** when creating a subsequent report.

Users who frequently use projects as work units can find this highly useful.

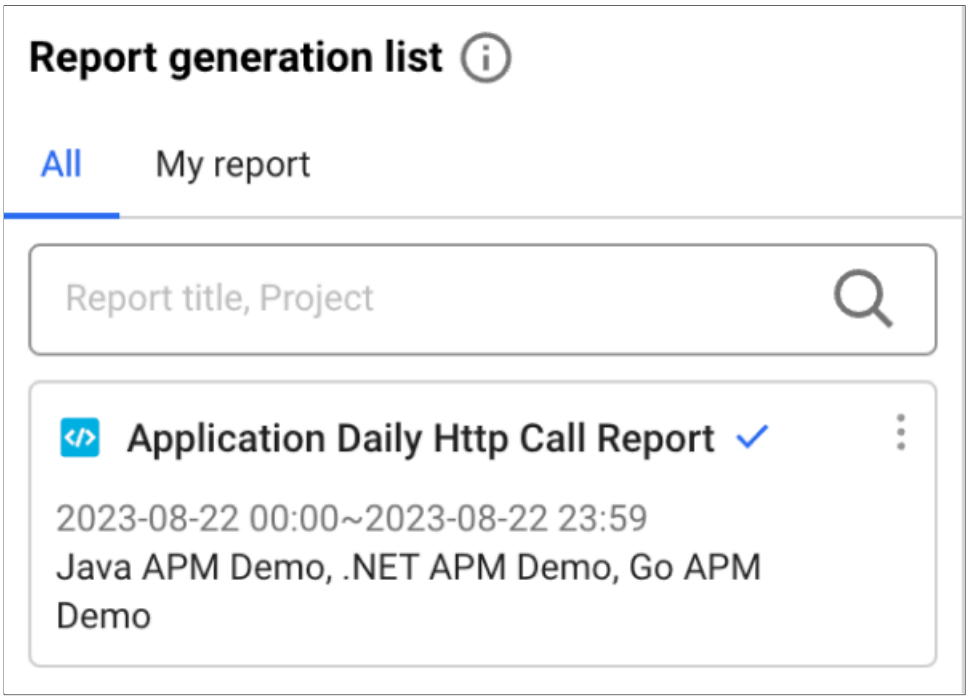
- : You can modify any saved report form.
- : You can delete any saved report form.



Checking the report list

You can see the reports that were created and are being created in **Report generation list**.

- In the **All** tab, you can see the reports created by you and the reports created by users with the same roles.
- In the **My report** tab, you can see only the reports created by you.
- If you select ⋮ on the upper right of a report item, you can use a feature to copy or delete the report.



- A report created with no error has the ✓ icon. In case of a report with any error, the ⚠ warning icon is displayed together. If you select the ⚠ icon, you can see in which project the error occurred.
- The generated reports are stored for 7 days, up to a maximum of 100 records.

Viewing and sharing

- **Viewing on the screen**

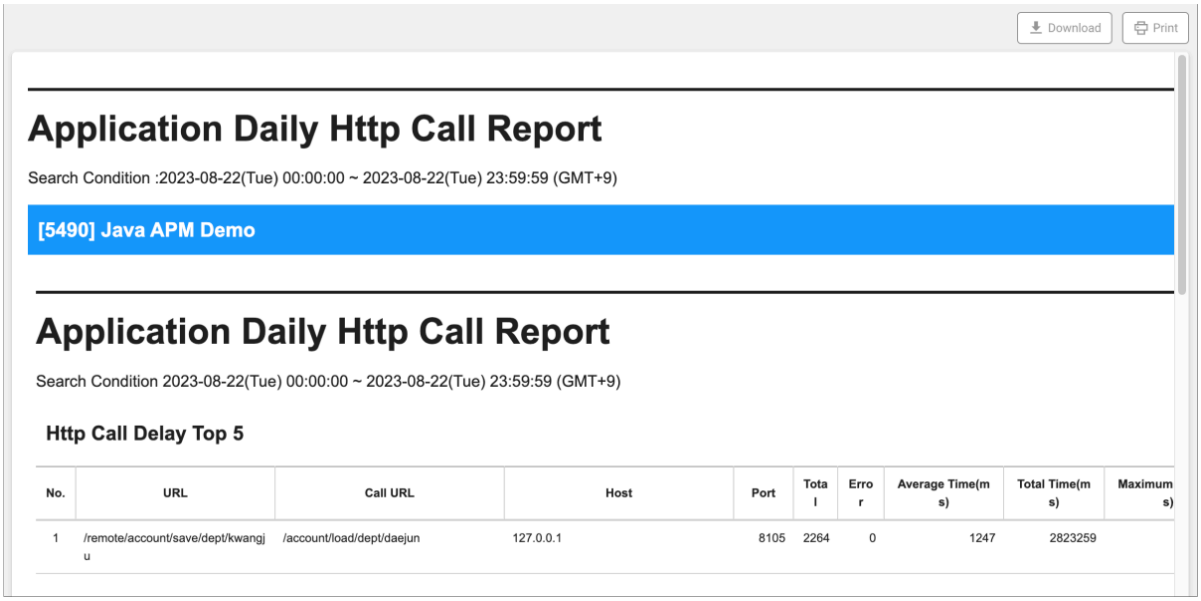
  In the **Report generation list** list, if you select a created report, you can see its details.

- **Download**

  Select **Download** to download the viewed report as an HTML file. You can share the report with users who do not have the monitoring management role.

- **Printing and saving PDF**

  Select **Print** to print the report via a printer or save it as a PDF file.



Java Alert Overview

Home > Select Project > **Alert** > **Event Configuration** > **Transaction** tab

Transaction events provide fine-grained monitoring of transactions based on specific URLs, errors, status codes, and more. When transactions that meet the configured conditions are detected, an event is triggered. Transaction data is analyzed in 1-minute intervals, and notifications are sent when the number of transactions that satisfy the conditions exceeds the defined threshold.

Latest Agent Versions

To use transaction events, you must install an agent that meets or exceeds the minimum supported version.

Language	Minimum Supported Version
Java	2.2.50 or later
PHP	Not specified (use the latest version or contact support)
Node.js	0.5.21 or later
Python	1.7.8 or later
.NET	2.3.5 or later
Go	Not specified (use the latest version or contact support)

Default Options

- **Event Level**: Indicates the severity of the event and is classified into three levels: Info, Warning, and Critical.
- **Search**: Search for events by event name or event reception tag.
- **+ Add Event**: Add a new transaction event.

- **Resize Columns**: Drag the boundary of each column to adjust its width.

Table | Transaction Event List Structure

Item	Description
No.	Event sequence number
	Whether the event is enabled - Enabled: The event is triggered when conditions are met - Disabled: The event does not operate and conditions are not evaluated
	Edit or delete event settings
Event Name	User-defined event name - Up to 255 bytes
Condition Type	Type of event trigger condition (elapsed time, error occurrence, pattern only)
Aggregation Basis	Criteria used to aggregate transaction counts
Target Pattern	Include/exclude patterns to which the event applies (URL, status code, etc.)
Trigger Count	Number of occurrences required to trigger the event - Aggregation period: fixed at 1 minute (based on each minute from :00 to :59 seconds; the period cannot be changed) - The event is triggered when transactions that meet the condition occur at least the configured number of times within that 1-minute period
Pause	Temporarily suppresses repeated occurrences of the same event after an alert is sent - Available durations: Disabled, 5 min, 10 min, 15 min, 20 min, 30 min, 1 hr, 2 hr, 3 hr, 6 hr, 12 hr, 1 day - If recovery notifications are enabled, no alert is sent for the same event during the selected time after a RECOVERED notification is received

Item	Description
Event Recipients	Users or groups that receive the event notification (reception tags)

Event Configuration

Transaction events are configured through the steps of **event conditions, target patterns, and basic information and notification settings**.
With event configuration, you can detect specific transaction patterns and receive alerts.

Add Event

To add a transaction event, go to **Event Configuration** > **Transaction** tab, then click the [+ Add Event] button in the upper-right corner of the screen.

How to Add a Transaction Event
Step 1. Define Event Conditions: Define the criteria that trigger the event.
Step 2. Select Event Target: Select the monitoring targets where the event applies.
Step 3. Basic Information and Notification Settings: Configure the event name, message, and recipients.

Define Event Conditions

Configure the conditions under which a transaction event is triggered.

Condition Type

Select the criteria used to trigger the event.

- **Elapsed Time**: Transactions that take longer than the specified time (ms).
  - **Elapsed Time Threshold**: When the condition type is Elapsed Time, enter the value in milliseconds. An

event is triggered when a transaction takes longer than the configured time (ms).

- **Error Occurred**: Transactions in which an error occurs.
- **Pattern Only**: Transactions that match a specific pattern (such as hitmap patterns).

Aggregation Criteria

Select the unit by which alert conditions are aggregated.

- Aggregation criteria: per agent, per URL of each agent, per error class of each agent, per business of each agent, per URL, per error class, per business, or overall.

Trigger Count (Required)

Configure how many times the condition must be met before a notification is sent.

Example: trigger count **1** sends a notification immediately if even one transaction meets the condition within 1 minute, while trigger count **3** requires at least 3 transactions meeting the condition to accumulate within the same 1 minute before a notification is sent.

Pause

Select a pause duration to prevent excessive alert notifications.

- After an event occurs, the same event will not trigger again during the configured time period.
- Available durations: `1 minute` , `5 minutes` , `10 minutes` , `15 minutes` , `30 minutes` , `1 hour` , `2 hours` , `3 hours` , `6 hours` , `12 hours` , `1 day`

Event Operation Time

Configure the time periods during which the event is active. This allows you to **suppress alerts during non-business hours or maintenance windows**. If no tag is configured, the event runs continuously for 24 hours while enabled.

1. Click **+ Add**.
2. In Event Operation Time, click **+ Create New Tag**.
3. In **Create Event Operation Time Tag**, select the **tag name, days, time, and color**, then click the [Apply]

button.

4. The created tag appears in the tag list. Select the checkbox to apply it.
5. To edit or delete a tag, click the icon and update or remove it in **Edit Event Operation Time Tag**.

> When a tag is deleted, it is removed from all alerts where it is applied. However, tags currently in use by event settings cannot be deleted.

Select Event Target

Configure patterns for transaction filtering.

Pattern Configuration

- You can enter multiple patterns, separated by pressing `Enter`.
- If no pattern is configured, events apply to all transactions in the project.
- Patterns of the same type are applied with an **OR** condition.
- Patterns of different types are applied with an **AND** condition.
- You can use wildcards (`*`) in patterns. Up to two wildcard positions are allowed.
- Each pattern must include at least one character other than the wildcard (`*`) and slash (`/`).

> **Be careful when combining slashes (`/`) and wildcards (`*`)**
>
> - `/api/*` : `/api` alone does not match. A path must follow `/api/` (e.g., ✅ /api/users, ❌ /api)
> - `/api*` : `/api` itself also matches (e.g., ✅ /api, ✅ /api/users, ✅ /apitest)

Match Patterns

Filter data that matches the specified items. After being applied, match patterns are displayed as blue tags.
You can enter up to 10 patterns for each item.

Item	Description
Operation Name	Operation name pattern
URL	Transaction URL pattern - Example: `/api/*`, `*/users/*`
Error Class	Error class name pattern - Example: `*Exception`, `*Error`
Error Message	Error message content pattern
Status Code	HTTP status code pattern - Example: `4**`, `5**`

Exclude Patterns

Exclude data that matches the specified items from the filtering targets. After being applied, exclude patterns are displayed as red tags and use the same format as match patterns. You can enter up to 10 patterns for each item.

Pattern Examples

- **Exact Match**

```
Pattern: "hello"
"hello"    → Match
"Hello"    → No match (case-sensitive)
```

- **Prefix Match**

```
Pattern: "hello*"
"hello"      → Match
"helloworld"  → Match
"hi"         → No match
```

- **Suffix Match**

```
Pattern: "*world"
"world"      → Match
"helloworld" → Match
"hello"      → No match
```

- **Substring Match**

```
Pattern: "*ell*"
"hello"      → Match (h'ell'o)
"excellent"  → Match (exc'ell'ent)
"welcome"    → No match (does not contain "ell")
```

- **Composite Patterns**

```
Pattern: "hello*world"
"hellobeautifulworld" → Match

Pattern: "hello*test*"
"hellomytestcase"     → Match

Pattern: "*test*world"
"mytestcaseworld"     → Match
```

Table | Practical Usage Examples

Pattern Type	Examples
URL Patterns	- `/api/*`: All URLs that start with `/api/` - `/users/*`: All URLs that contain `users` - `/api/v1/users`: Only the URL that matches exactly
Error Class Patterns	- `Exception`: All errors that end with `Exception` - `java.lang.*`: All classes that start with `java.lang` - `Timeout*`: All errors that contain `Timeout`
Status Code Patterns	- `4**`: All 4xx status codes - `404`: Only the 404 status code

Pattern Type	Examples
	- `5**`: All 5xx status codes

Pattern Limitations

- **Required characters**: Each pattern must include at least one character other than the wildcard (`*`). A pattern cannot consist of wildcards only.
  - ❌ Not allowed: `**` , `*`
  - ✅ Allowed: `a*` , `a` , `a*` , `/api*`
- **Empty strings**: Empty strings or null values are not matched.
- **Consecutive wildcards**: Consecutive `*` characters are treated as a single wildcard.
- **Complex patterns**: Patterns containing three or more wildcards are not supported.
- **No regular expressions**: Character classes (`[a-z]`), quantifiers (`+` , `?`), and grouping (`()`) are not supported.
- **Case sensitivity**: All pattern matching is case-sensitive.
- **No escaping**: There is no way to escape characters for literal matching.

Basic Information and Notification Settings

Configure the basic information of the event and how notifications are delivered.

Event Activation

Configure whether notifications are sent when an event occurs.

- **Enabled**: Notifications are sent when the configured conditions are met.
- **Disabled**: Notifications are not sent even if the configured conditions are met.

Event Name (Required)

Enter the event name to be used as the notification title.

Event Level (Required)

Select the event level (Info, Warning, Critical).

Event Recipients

Specify the members who will receive event notifications.

- **Receive All** **Receive by Tag**

Notifications are sent to all targets (members and channels) with event reception enabled in the project.

Notifications are sent to project members who have the selected tag.

1. Select **Receive by Tag**.
2. In the **Receiver Tags** section, click [+ Add Tag] or [+].
3. Click **+ Create New Tag** at the bottom of the **Event Receiver Tags** window.
4. In the **Create Tag** window, enter a tag name, choose a color, and click [Create Tag].
5. To edit or delete a created tag, click the icon in the tag list.
6. In the **Event Receiver Tags** window, select the desired tag from the **Tag List** to apply it.

> **Receiver Tags**
>
> Receiver tags are a feature for managing notification recipients (users or channels) at a group level.
> When you assign receiver tags to an event, notifications are delivered only to users who have those tags.
> If no receiver tag is set, notifications are sent to all users in the project.

Reference. Example of Event Behavior

This is an example of an actual transaction event occurrence.

> **Configuration Conditions**

- Aggregation criteria: By agent
- Occurrence count: 2 times
- Pause: 5 minutes
- URL patterns: `/api/posts/**` , `/delete/*` , `/*test*/`
- Error class patterns: `*TimeoutException*` , `*NullPointerException`
- Status code pattern: `4**`
- Excluded error class: `*IOException`
- Excluded status code: `5**`

Example of a Triggered Notification Message

The transaction information in the event message displays the values from the most recent transaction that matched the conditions. The event message includes both the configured conditions and the details of the actual transaction that triggered the alert, allowing you to clearly identify which conditions caused the notification.

```
(ignored count: 3)
[Per Agent] limit: 2, count: 3
[Condition]
url patterns: /api/posts/**, /delete/*, */test/*
error class patterns: *TimeoutException*, *NullPointerException
status patterns: 4**
exclude error class patterns: *IOException
exclude status patterns: 5**
[Transaction]
url: /api/posts/test/timeout
txid: 408093121470636540
status: 408
error class: java.util.concurrent.TimeoutException
```

Explanation of Event Message Composition

- **(ignored count: n)**: The number of transactions that met the conditions but did not trigger a notification during the pause period. This item is not displayed if there are no ignored transactions during the pause period.
- **Aggregation Information**: Displayed differently depending on the configured aggregation method.

  - **By Agent**: `[Per Agent] limit: n, count: n`
  - **By URL**: `[Per Url] limit: n`, followed by a list of each URL and its occurrence count
  - **By Error Class**: `[Per Error Class] limit: n`, followed by a list of each error class and its occurrence count
  - **By Context**: `[Per Context] limit: n`, followed by a list of each context and its occurrence count
  - **By Agent & URL**: `[Per Agent & Url] limit: n`, followed by a list of each URL and its occurrence count
  - **By Agent & Error Class**: `[Per Agent & Error Class] limit: n`, followed by a list of each error class and its occurrence count
  - **By Agent & Context**: `[Per Agent & Context] limit: n`, followed by a list of each context and its occurrence count
  - **Total**: `[Total] limit: n, count: n`
- **[Condition]**: Displays the conditions that were configured at the time the event was triggered. Only configured patterns are shown.
  - **context patterns**: Context match patterns
  - **url patterns**: URL match patterns
  - **error class patterns**: Error class match patterns
  - **error message patterns**: Error message match patterns
  - **status patterns**: Status code match patterns
  - **exclude context patterns**: Context exclusion patterns
  - **exclude url patterns**: URL exclusion patterns
  - **exclude error class patterns**: Error class exclusion patterns
  - **exclude error message patterns**: Error message exclusion patterns
  - **exclude status patterns**: Status code exclusion patterns
- **[Transaction]**: Detailed information about the most recent transaction that met the conditions.
  - **url**: Transaction URL
  - **txid**: Unique transaction identifier
  - **status**: HTTP response status code (not displayed for non-HTTP transactions)
  - **error class**: Error class (not displayed if no error class exists)
  - **error message**: Error message (not displayed if no error message exists)

Notes

- Transaction events are aggregated and processed in 1-minute intervals.
- If the condition type is **Pattern Only**, at least one match or exclusion pattern must be configured.
- Pattern matching is case-sensitive.
- Notification messages include detailed information about the most recent transaction that met the conditions.

Event Configuration

Home screen > Select project **Alert** > **Event Configuration**

In **Event setting**, you can configure and manage events based on anomaly conditions that may occur in monitored systems or applications. When the configured conditions are met, notifications can be delivered through your preferred channels such as email, SMS, messengers, or app push notifications. You can tailor event configurations according to your monitoring objectives.

Metric Event

Configure metric-based event conditions for Java to trigger alerts.

Composite Metric

Combine multiple metrics to configure composite metric event conditions.

Hitmap Pattern

Configure events that detect abnormal conditions with hitmap-based pattern analysis.

Anomaly Event

Configure events that trigger alerts by automatically detecting anomalies in Java metrics.

Real-time logs

Configure log-based events that trigger alerts based on Java log content.

Composite log

Configure composite log events that trigger alert notifications based on Java log content.

Transaction Event

Configure events that trigger alerts based on Java transaction conditions.

Home Screen > Select Project > **Alert** > **Event Configuration**

Event setting provides features to configure and manage events based on anomaly signals that may occur in monitored systems or applications. When configured conditions are met, you can receive notifications through your preferred channels such as email, SMS, messengers, or app push notifications. You can design event configurations according to your monitoring objectives.

Metric Event

Configure metric-based event conditions for Java to trigger alerts.

Composite Metric

Combine multiple metrics to configure composite metric event conditions.

Hitmap Pattern

Configure events that detect abnormal conditions with hitmap-based pattern analysis.

Anomaly Event

Configure events that trigger alerts by automatically detecting anomalies in Java metrics.

Real-time logs

Configure log-based events that trigger alerts based on Java log content.

Composite log

Configure composite log events that trigger alert notifications based on Java log content.

Transaction Event

Configure events that trigger alerts based on Java transaction conditions.

Reference

- When configuring events, use appropriate threshold values to prevent unnecessary event generation.
- When setting up notifications, consider enabling recovery notifications (alerts sent when issues are resolved) to support effective operational response.

Common Features

All event configurations can be downloaded in **JSON or Excel format**. By sharing the same type of product configuration, you can reduce repetitive event setup work.

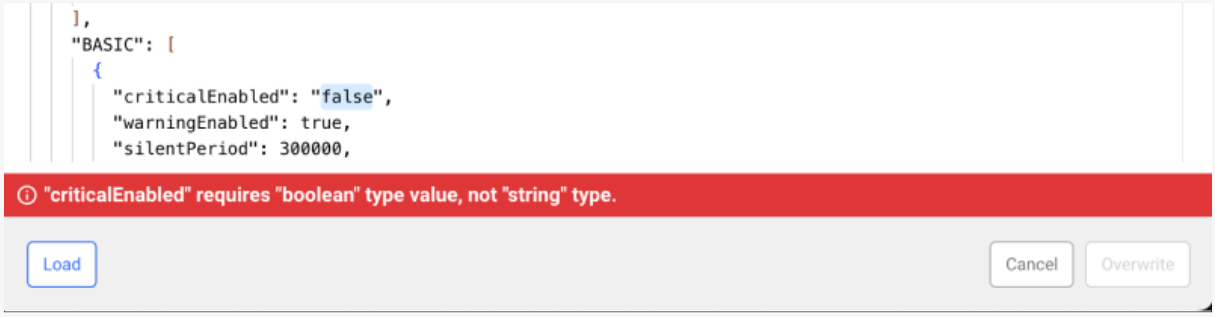
- **Batch JSON Edit**: Edit all event conditions in bulk using JSON format.
- **Batch JSON Download**: Save all currently registered event conditions as a JSON file.
- **Batch Excel Download**: Download the event list as an Excel file for external reporting or backup purposes.

Batch JSON Edit

1. Click the [Batch JSON Edit] button at the top of the screen.
2. Import a JSON file downloaded via the **Batch JSON Download** feature, or edit the content directly.
3. Click the [Overwrite] button to save.
4. When a confirmation message appears, click the [Overwrite] button.

After overwriting, you cannot restore the previous event configuration.

- Arbitrarily modifying the values of the `productType` or `platform` properties may cause events to malfunction in other projects.
- The JSON editor includes built-in validation. If the format or values are invalid, an error message appears below the editor and the [Overwrite] button is disabled.



JSON Data Structure

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

JSON Field	Description
productType	Product type (e.g. APM, Infra)
platform	Platform information (e.g. java, node)
events	Event data
◈ HITMAP	Hitmap pattern

JSON Field	Description
◈ LOG_REALTIME	Real-time log events
◈ METRICS	Metrics
◈ COMPOSITE_LOG	Composite log events
◈ BASIC	Basic events
◈ COMPOSITE_METRICS	Composite metrics
◈ ANOMALY	Anomaly detection

Batch JSON Download

Click the [Batch JSON Download] button at the top of the screen.

- The JSON file name follows the format integrated-event- `YYYYMMDD` .json.
- The structure of the JSON data and the sub-properties under the `events` attribute may vary depending on the product (`platform`) or product type (`productType`).

If you arbitrarily modify the values of the `productType` or `platform` properties in the JSON data, events may not function correctly in other projects.

Batch Excel Download

Click the [Batch Excel Download] button at the top of the screen.

Metric Event Configuration

Home > Select Project > **Alert** > **Event Configuration** > **Metrics** tab

You can view the complete list of all metrics-based events configured for the project. You can review, edit, or delete existing metrics events, or add new events as needed.

Default Options

- **Event Level**: Indicates the severity of an event and is classified into three levels: Normal (Info), Warning (Warning), and Critical (Critical).
- **Duplicate Agent Alert Suppression**: If two or more agents with the same name run simultaneously, events are not triggered for the selected duration (30 seconds, 1 minute, 3 minutes, 5 minutes, 10 minutes).
- **Search**: Search events by event name.
- **JSON Download/Upload**: Edit event rules directly in the JSON editor, or download them as a file, modify them, and upload them again.
- **+ Add Event**: Add a new metrics event.
- **Resize Columns**: Adjust column widths by dragging column borders.

Table | Metrics Event List Structure

Item	Description
No.	Event sequence number
	Event activation status – Enabled: An event is triggered when conditions are met – Disabled: The event does not run and conditions are not evaluated
	Edit or delete event settings
Event Name	User-defined event name – Up to 255 characters

Item	Description
Rule	Event trigger conditions
Target	Monitoring targets to which the event applies
Trigger Count	The time window and number of occurrences required to trigger an event – An event is triggered when conditions are met the configured number of times within the selected duration
Pause	Temporarily suppresses the same event after an alert is triggered – Available durations: Disabled, 5 min, 10 min, 15 min, 20 min, 30 min, 1 hr, 2 hrs, 3 hrs, 6 hrs, 12 hrs, 1 day – If recovery notifications are enabled, alerts are not triggered for the selected duration after a RECOVERED notification is sent
Recovery Notification	Whether to receive a RECOVERED (normal) notification when Critical or Warning events are resolved – Enabled: Sends a RECOVERED notification when the status changes from in progress to normal and allows viewing active and resolved states in event history – Disabled: Triggers a one-time event each time the threshold is exceeded
Event Recipients	Users or groups that receive notifications for the event (recipient tags)

Event Configuration

Metrics events allow you to define events based on conditions that occur in monitored targets and configure notifications so you can respond efficiently.

Add Event

To add a metrics event, go to **Event Configuration** > **Metrics** tab, then click the [+ Add Event] button in the upper-right corner.

How to Add a Metrics Event
Step 1. Select Template: Choose the basic structure of the event.
Step 2. Define Event Conditions: Configure the conditions that determine when the event is triggered.
Step 3. Select Event Target: Select the monitoring targets where the event applies.
Step 4. Basic Information and Notification Settings: Configure the event name, message, and recipients.

Select Template

1. Depending on how you want to configure the event, select one of the three templates on the left side of the screen.
   - **Create New**: Freely configure categories and metrics from scratch. This option is suitable for advanced users who need complex conditions or fine-grained monitoring rules.
   - **Quick Setup**: Provides predefined categories, metrics, and default thresholds. You can apply it quickly by adjusting only the threshold values, which makes it suitable for beginners.
   - **Advanced Setup**: Add conditions or freely modify settings based on predefined categories and metrics.

> ⓘ Metrics events support multiple configuration approaches. The available templates may vary depending on the monitoring platform.

Define Event Conditions

In the **Define Event Conditions** step, configure the metrics that determine when the event is triggered, along with the trigger count, pause settings, recovery notifications, and event active time.

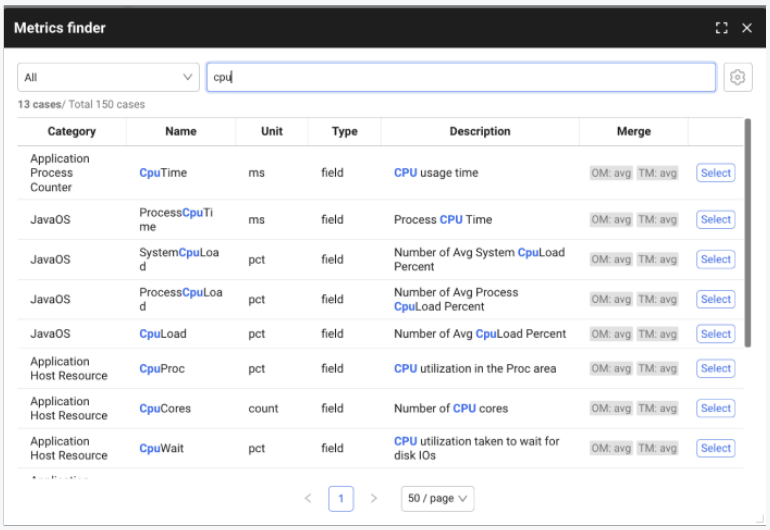
Metric Settings

1. Enter a category in the search field, or select a category (metric) from the category list.
   You must select a category to proceed with metric configuration.

- The category list shows the category name, data collection interval, and key. It displays metrics data collected in the project within the last 3 hours.

2. If the metric you need is not listed, select **Direct Input** and enter the category manually.

> **If you are not sure about categories or metrics, use the Metrics Finder.**
>
> When configuring metrics for the first time, the Metrics Finder helps you locate metrics more quickly.
>
> - Search for a metric in the Metrics Finder to view related categories and detailed information.
> - Click the [Select] button for a metric in the search results to automatically apply the selected value to the event configuration screen.



3. Select the event level (Critical, Warning, Info). You can configure multiple levels for a single metric.

> **Event Level Behavior**
>
> Even when multiple conditions are satisfied, only the highest-priority level (Critical → Warning → Info) triggers an event. For details, see the Level Action Guide.
>
> - Event levels are classified as Critical (Critical), Warning (Warning), and Info (Info).

> ⓘ
> - It is recommended to configure multiple levels for the same metric with stepwise thresholds.

4. Enter the field, operator, and value for the selected event level.

   a. Click **+ Add** to add more conditions.

   - `&&`: An event is triggered only when **all conditions are satisfied simultaneously**
   - `||`: An event is triggered when **at least one condition is satisfied**

   b. Click **—** to remove the condition.

> ⓘ For condition syntax and supported operators, refer to the Condition Settings Guide.

Trigger Count

Configure how many times a condition must be met before an event is triggered. This is useful for detecting persistent issues only.

1. Select the trigger frequency.
   - **Consecutive**: An event is triggered when the condition occurs consecutively for the configured number of times.
   - **Recent**: An event is triggered when the condition occurs the configured number of times within the selected time window.
     - Available durations: `5 seconds`, `10 seconds`, `15 seconds`, `30 seconds`, `1 minute`, `2 minutes`, `3 minutes`, `4 minutes`, `5 minutes`, `10 minutes`, `30 minutes`, `1 hour`
2. Enter the trigger count.
   - The **Interval** value corresponds to the data collection interval of the selected category. (Interval: 10s)

Pause

To prevent excessive event generation, select a pause duration for event execution. After an event occurs, the same event will not be triggered again during the configured time period.

- Available durations: `Disabled` , `5 minutes` , `10 minutes` , `15 minutes` , `20 minutes` , `30 minutes` , `1 hour` , `2 hours` , `3 hours` , `6 hours` , `12 hours` , `1 day`

> **When recovery notifications are enabled**
>
> Even if the same event condition is met again, no notification is sent until the configured time has passed since the **RECOVERED** notification was received.

Resolved Notification

When the resolved notification feature is enabled, you can track state changes after an event occurs (in progress → resolved). The event is shown as an ongoing event in the **Event History** menu.

- Resolved notifications apply only to Critical and Warning levels. Info-level events always operate as one-time events.
- When a Critical or Warning event is resolved, a notification with the RECOVERED (normal) state is sent.
- Click the toggle button to **enable** or **disable** this feature.
  - **Enabled**: Events operate based on state. They enter an in-progress state when the threshold is exceeded and transition to a resolved state when the value recovers below the threshold.
  - **Disabled**: A one-time event is generated each time the threshold is exceeded.

Event Operation Time

Configure tags so that events operate only during specific time periods (such as business hours, non-business hours, or maintenance windows). If no tag is configured, events operate continuously for 24 hours while enabled.

1. Click **+ Add**.
2. In Event Operation Time, click **+ Create New Tag**.

3. In **Create Event Operation Time Tag**, select the **tag name, days, time, and color**, then click the [Apply] button.
4. You can view created tags in the tag list, and apply them by selecting the checkbox.
5. To modify or delete a tag, click the icon and update or delete it in **Edit Event Operation Time Tag**.

> When a tag is deleted, it is removed from all users to whom it was applied. However, tags that are currently in use by event rules cannot be deleted.

Simulation

The simulation feature lets you pretest configured event conditions using historical data. If the expected number of events is too high or too low, you can adjust the conditions to find optimal threshold values.

- **Rule Validation**: Verify how newly created event rules would behave when applied to historical data.
- **Threshold Tuning**: Adjust thresholds to an appropriate level when events occur too frequently or not at all.

Start Simulation

1. After completing event condition settings, click the [Simulation] button.
2. Review the expected number of events and the visualized results.

Understanding Simulation Results

1. **Time-Series Data**

   A graph displays metric changes over time along with threshold lines for each level, allowing you to intuitively identify when events are triggered.

2. **Event Counts by Level**

   The number of events for the Critical, Warning, and Info levels is shown on the right side of the screen. This helps you understand overall event distribution and frequency.

3. **Target-Level Details**

Hover over a specific point on the chart to see the exact metric value and monitoring target at that moment. When multiple targets exist, they are distinguished by color. If patterns differ by target, you can use filtering to analyze them individually.

4. **Peak Interval Identification**

   Identify periods where metrics spike sharply to understand when issues occur most frequently or to recognize recurring patterns.

Simulation Tips

- **Simulate Different Time Ranges**

  Run simulations for weekdays vs. weekends and peak vs. off-peak hours to set optimal thresholds for each time period.

- **Include Actual Incident Periods**

  Include past outage periods in simulations to verify whether current rules accurately detect real incidents.

- **Add Target Filtering**

  When many targets are monitored, simulation results can become complex. Use filters to focus on specific targets for clearer insights.

- **Adjust Thresholds Gradually**

  Instead of making large changes at once, adjust thresholds incrementally, such as in 10 percent steps, to find optimal values.

Select Event Target

If no event target is specified, events are triggered for all targets in the project, which can result in excessive events. It is therefore recommended to clearly define the targets to which the event applies.

If you change the event targets, the number of events may change. Click the **Simulation** button and run it again to check the updated expected event count.

Target Selection

You can specify the targets to which the event applies using either **Select Input** or **Direct Input**.

- Select Input Direct Input

1. Select or enter a tag, operator, and value.
   a. If it is difficult to find a tag, you can search for it using the **Metrics Finder**.
2. Click **+ Add** to add event targets using conditions (`&&` , `||`).

Enter the tag, operator, and value of the target to monitor directly.

- When you choose direct input, a list of available tags (**Tag Select Input**) is provided below the input field.

```
ex. endsWith(okindName, 'example_name') && container == 'prod.billing'
ex. ${4xxErrorType} == '401'
```

> If you change the input method, any content you have entered will be reset.

Basic Information and Notification Settings

Configure the **event name, message, and recipients**. Because this information is displayed as is in event history and actual notification messages, it should be clear and easy to identify.

Event Activation

Configure whether the event is triggered when the defined conditions are met.

- **Enabled**: The event is triggered when the configured conditions are satisfied.
- **Disabled**: Conditions are not evaluated.

Event Name

Enter the event name to be used as the event title. The configured event name is displayed in the **Event List** and **Event History** menus and can be used as a search keyword.

Message

Enter the event message. The message you enter is used in event history and notifications. Click the icon to view previously written message history. You can include variables in the message to insert actual values at the time the event occurs.

- **Variable Usage Rules**
  - Variable format: `${metric_name}`
  - Only metrics within the **same category as the configured category** can be used as variables.
  - You can combine multiple variables to create a more detailed message.
  - The list of available variables can be found in the **Metrics Query** menu.
- **Variable Usage Example**

	Event Configuration	When Event Occurs
Title	Disk Usage Increase	Disk Usage Increase
Message	`Disk = ${disk}%`	`Disk = 89.9%`

Event Recipients

Specify the members who will receive event notifications.

- **Receive All** **Receive by Tag Selection**

Notifications are sent to all targets (members and channels) in the project that have event reception enabled.

Notifications are sent to project members who have the selected tag.

1. Select **Receive by Tag Selection**.
2. In the **Reception Tag** section, click the [+ Add Tag] or [+] button.
3. Under the **Event Reception Tag** window, click **+ Create New Tag**.
4. In the **Create Tag** window, enter a tag name, select a color, and click the [Create Tag] button.
5. You can edit or delete created tags by clicking the icon in the tag list.

6. In the **Event Reception Tag** window, select the desired tag from the **Tag List** to apply it.

> **Reception Tags**
>
> Reception tags are a feature for managing users or channels that receive notifications **at a group level**. When a reception tag is assigned to an event, notifications are delivered only to users associated with that tag. If no reception tag is set, notifications are sent to all users in the project.

Edit/Delete Event

1. Go to **Alert Notifications** > **Event Settings** and open the **Metrics** tab.
2. In the event list, click the icon for the event you want to edit or delete.
3. In the **Edit Event Rule** window, update the options and click the [Save] button.

   a. To delete the selected event, click the [Delete] button in the upper-right corner of the **Edit Event Rule** window.

Event Sharing

You can save metric event settings as a JSON file to share configurations with other users or import settings created by others.

- JSON file name: event-rules-`YYYYMMDD`.json

Export

1. Click the [**JSON**] button at the top right of the screen.
2. When the JSON editor opens, click the [Export] button.
   - If you use the export function after searching for events, only the searched list will be downloaded as a JSON file.
3. After the JSON file is downloaded, share it with other users.

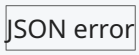
Import

1. Click the [ ] button at the top right of the screen.
2. Select the JSON file downloaded using the **Export** function.
3. When the JSON editor opens, choose either [Add to List] or [Overwrite].

> It is recommended to use this feature between projects of the same product type. You can import event settings from projects of different products, but they may not function correctly.

Edit in JSON Format

1. Click the [**JSON**] button at the top right of the screen.
2. When the editor opens, modify the content according to the JSON format.
3. After completing the edits, click the [Save] button at the bottom of the screen.

> If the modified content does not conform to the JSON format, an error message will be displayed at the bottom of the screen and the content cannot be saved. The error message may vary depending on the type of formatting issue.
>
> 
>

JSON Data Structure

```
{
  "version": 2,
  "eventId": "zcef7s7f27rum1",
  "enabled": true,
  "stateful": false,
  "title": "Active Transaction",
  "message": "Active Transaction = ${active_tx_count}",
  "category": "app_counter",
  "alertLabel": [
    "oid"
  ],
  "repeatCount": 1,
  "repeatDuration": 0,
```

```
  "silent": 300000,
  "receiver": [],
  "timeTag": [],
  "selectString": "",
  "conditions": [
   {
    "level": 20,
    "enabled": true,
    "rule": "active_tx_count > 100"
   }
  ],
  "createTime": 1763632674345,
  "lastModifiedTime": 1763632674347,
  "lastModifiedUser": "support@whatap.io",
  "basic": true,
  "metaId": "java004",
  "selectCondition": {
        "oid": [
          "-1010758404",
          "-250906941"
        ]
  }
}
```


Table | JSON Data Structure

JSON Field	Type	Description	Notes
version	Integer	Version of the event rule	Restricted
eventId	String	Unique identifier of the event rule	
enabled	boolean	Whether the event is enabled	
stateful	boolean	Whether the event is state-based	
title	String	Event rule name defined by the user	
message	String	Event rule message defined by the	

JSON Field	Type	Description	Notes
		user	
category	String	Data category	
alertLabel	`List<String>`	Default identifier (primary key) value assigned per category for event state management	Restricted
repeatCount	Integer	Number of repetitions	
repeatDuration	Long	Duration for repetition	
silent	Integer	Pause duration	
receiver	`List<String>`	List of recipient tag keys	
timeTag	`List<String>`	List of operation-time tag keys	
selectString	String	Target selection	
conditions	`List<Object>`	List of trigger conditions by level	
◈ level	byte	Level value	`Critical`: 30, `Warning`: 20, `Info`: 10
◈ enabled	boolean	Whether the level is enabled	
◈ rule	String	Trigger condition for the level	
createTime	Long	Initial creation time of the event rule	
lastModifiedTime	Long	Last modification time of the event rule	
lastModifiedUser	String	Account that last modified the event rule	

JSON Field	Type	Description	Notes
basic	boolean	Whether the event was created using Quick Setup	Restricted
metaId	String	Unique identifier of the Quick Setup template	Restricted: This value exists only for events created via Quick Setup (basic=true).
selectCondition	Object	Target selection value for Quick Setup	This value exists only for events created via Quick Setup (basic=true).

Fields marked as “Restricted” may affect event behavior, so it is recommended not to modify their values.

Appendix

Trigger Conditions and Target Selection Guide

Metric alert trigger conditions and event target selection use the same syntax. However, trigger conditions use the field key as the variable, while target selection uses the tag key as the variable.

Level Behavior Guide

Event levels are categorized into Critical, Warning, and Info.

1. **Priority-based Event Triggering**

   When multiple level conditions are met at the same time, only the event with the highest priority is triggered.

   Example

```
Configuration:
- Warning: CPU > 70%
- Critical: CPU > 90%

Current state: CPU 95%
→ Result: Only the Critical event is triggered (Warning is suppressed)
```

2. **Level Escalation**

   For events that maintain an in-progress state, when a higher-level condition is met while a lower-level event is already in progress, both levels enter the in-progress state.

   Example

```
Configuration:
- Warning: CPU > 70%
- Critical: CPU > 90%

Scenario:
1) CPU 80% → Warning triggered (in progress)
2) CPU 95% → Warning (in progress), Critical (in progress)
```

3. **Level De-escalation**

   For events with an in-progress state, when the level transitions from a higher level to a lower level, the higher-level event is resolved while the lower-level event remains active.

   Example

```
Configuration:
- Warning: CPU > 70%
- Critical: CPU > 90%

Scenario:
1) CPU 97% → Critical triggered
2) CPU 85% → Critical resolved, Warning remains
3) CPU 65% → Warning resolved
```

4. **Execution Flow**

```
- Warning: CPU > 70%
- Critical: CPU > 90%

Metric value: 60%  →  75%  →  92%  →  85%  →  60%
State:      Info      Warning     Critical     Warning     Info
                   (triggered)  (triggered)  (maintained)  (resolved)
                                Warning
                              (maintained)
```

Events that have an in-progress state remain active as long as their conditions are met and continue until those conditions are cleared.

Condition Configuration Guide

If a metric name starts with a number or contains special characters, you must wrap it in the `${metric_name}` format.

- Brackets `()`, `[]`
- Operators `+`, `-`, `*`, `/`, `%`
- Delimiters `:`, `@`, `#`, `,`, (space)
- Other special characters `!`, `^`, `&`, `|`, `~`, `` ` ``, `=`

Examples using `${}`:

```
${cpu(xos)} > 50
${mem[0]} >= 100
${cpu-usage} > 80
${namespace:cpu} > 70
${metric name} > 60
${4xx_error} > 10
```

Usable without `${}`: Metric names that contain only alphabetic characters, underscores (`_`), or dots (`.`)

```
cpu > 80
cpu_usage > 50
CPUUtilization.Average > 0.8
```

1. Comparison Operators

Numeric Comparison

```
Greater than:          cpu > 80
Greater than or equal:   cpu >= 80
Less than:             memory < 1000
Less than or equal:      memory <= 1000
Equal to:              status == 200
Not equal to:          error != 0
```

String Comparison

```
status == 'OK'
region == "us-east-1"
```

2. Arithmetic Operators

- Basic arithmetic operations

```
Addition:       cpu + 10 >= 90
Subtraction:    memory - 100 >= 500
Multiplication: cpu * 2 >= 100
Division:       disk / 1024 >= 100
Modulo:         value % 10 == 0
```

- Controlling precedence with parentheses

```
(cpu + memory) * 2 >= 200
(disk - used) / total >= 0.2
((cpu + memory) / 2) >= 50
```

- Negative values

```
cpu > -100
```

3. **Logical Operators**

   - AND operator (&&)

   ```
   cpu > 80 && memory > 1000
   ${cpu(xos)} > 50 && ${mem(xos)} > 60
   ```

   - OR operator (||)

   ```
   cpu > 90 || memory > 90
   disk < 10 || network > 1000
   ```

   - Combined logical operations

   ```
   (cpu > 50 && memory > 50) || disk < 20
   cpu > 80 && (memory > 1000 || disk > 500)
   ```

4. **Pattern Matching Operators**

   - LIKE operator

   ```
   oname like 'prod-*'
   url like '*error*'
   message like 'WARN%'
   ```

   - NOT LIKE operator

   ```
   oname not like 'test-*'
   url not like '*debug*'
   ```

5. **Built-in Functions**

   - Null check functions

```
isNull(value)           # Checks whether the value is null
isNotNull(value)        # Checks whether the value is not null
nvl(value, 0)           # Returns a default value if null
isEmpty(str)            # Checks whether the string is empty
isNotEmpty(str)         # Checks whether the string is not empty
```

- Aggregate functions

```
sum(cpu, memory, disk)            # Sum
avg(cpu, memory)                  # Average
max(cpu, memory, disk)            # Maximum
min(cpu, memory, disk)            # Minimum
count(value1, value2, value3)     # Count
```

- Math functions

```
round(cpu, 2)           # Rounds to 2 decimal places
```

- String functions

```
length(str)             # String length
startsWith(str, 'prefix')  # Checks prefix
endsWith(str, 'suffix')    # Checks suffix
indexOf(str, 'search')     # String index
substring(str, 0, 10)      # Substring
trim(str)               # Trims whitespace
replace(str, 'old', 'new') # String replacement
hasStr(str, 'search')      # Checks string containment
```

- Conditional functions
  - **if**: A conditional function that returns the `trueValue` if the condition evaluates to `true`, or the `falseValue` if it evaluates to `false`

    Syntax

```
if(condition, trueValue, falseValue)
```

Example

```
if(value >= 90, 'High', 'Medium') == 'High'
```

- **decode**: Compares a value against conditions and returns the result of the matching condition, or a default value if no match is found

Syntax

```
decode(value, condition1, result1, condition2, result2, ..., conditionN, resultN, defaultValue)
```

Example

```
decode(value, 1, 'Low', 2, 'Medium', 3, 'High', 'Unknown') == 'Unknown'
```

- **in**: Returns `true` if the value matches any of the specified candidates; otherwise returns `false`

Syntax

```
in(value, candidate1, candidate2, ..., candidateN)
```

Example

```
in(status, 200, 201, 204) == false
```

Writing Guidelines

- Recommended
  - Use simple variable names without `${}` (e.g. cpu, memory)
  - Always use `${}` when special characters are included
  - Use parentheses to clearly separate complex expressions
  - Enter function names exactly as defined (case-sensitive)
- Avoid

- Incomplete expressions (e.g. `cpu >`, `memory &&`)
- Mismatched parentheses (e.g. `(cpu > 80 && cpu > 70)`)
- Consecutive or invalid operators (e.g. `cpu >> 80`, `cpu >< 80`)

Composite Metric Event Configuration

Home > Select Project > **Alert** > **Event Configuration** > **Composite metrics** tab

You can create events and send alert notifications using complex rules.

Default Options

- **Search**: Search for specific events by selecting an event name or an event recipient tag
- **JSON Download/Upload**: Edit event rules directly in the JSON editor, or download them as a file, modify them, and upload them again
- **Alert Message Settings**: Select detailed fields for alert messages and edit the content for each field
- **+ Add Event**: Add a new composite metrics event

> **Permissions**
>
> To configure **Composite metrics** events, you must have **Alert Settings** permission. You can grant this permission by clicking the icon for the member in **Home > Select Project > Management > Integrated Member Management**.

Table | Composite Metrics Event List Layout

Item	Description
Event Title	User-defined event name - Up to 255 bytes
Condition	Event trigger conditions
Notify on Event Recovery	Whether to receive RECOVERED (normal) notifications when Critical or Warning events are resolved – Enabled: Sends a RECOVERED notification when the state changes from in-progress to normal, and allows you to distinguish in-progress and resolved states in event history – Disabled: Triggers a one-time event each time the threshold is exceeded

Item	Description
Event Recipient Tag	Users or groups that receive notifications for the event
	Edit or delete the event
	Event enablement status – Enabled: Event is triggered when conditions are met – Disabled: Event does not run and conditions are not evaluated

Event Configuration

Composite metrics events define event conditions based on **MXQL**, the metric data query language. The **Create from Chart** feature provides a combo box to support MXQL auto-completion.

Add Event

1. Click the **Composite metrics** tab from **Alert** > **Event Configuration**.
2. Click the [+ Add Event] button on the right side of the screen.
3. In the **Composite metrics** window, click **Create from Chart**.

   a. You can use the provided templates.
4. In the **Event Configuration** screen, first query the event data, then configure the event conditions.

Query Event Data

Query event data to build charts. In the **Event Configuration** screen, select either the **Widget** or **Text** option under **Query event data** on the right to configure events.

- **Widget** **Text**

Using the options for configuring a time series chart, you can automatically complete the **MXQL** used for event configuration.

1. Select the date range for querying data.
2. Select the **Category** (target) for querying data.
3. Click **+ Add Filter** under **Filter** to create event conditions.

   a. In the filter window, select an expression, tag, and filter value.
   b. Click **+ Add** to add additional conditions.
   c. To delete a condition, click the **–** icon next to it.
4. Under **Grouping**, select the grouped metric data. Multiple selections are supported.
5. Under **Time Unit**, set the time interval for grouped data by selecting `seconds`, `minutes`, or `hours`.
6. Under **Field**, select the fields to be used for event trigger conditions. Multiple selections are supported.

> MXQL entered in text mode is not compatible with widget mode.

An editor appears where you can directly modify **MXQL**.

1. Edit **MXQL** directly and click the **Query** button.
2. After selecting a date, you can view the results as a chart or graph.

Notification

Enter the basic information for alert notification settings.

- **Enable Event**: When the toggle is enabled, the event becomes active.
- **Level**: Select the event level (Critical, Warning, Info).
  - **Notify on Event Recovery**: When the toggle is enabled, a notification is sent when the event is resolved and enters the normal (RECOVERED) state.
- **Title**: Enter a title for the event rule.
- **Message**: Enter the content of the alert message delivered when the event is triggered. You can use variables in the message to include actual values at the time the event occurs.

  Message example

Variable Usage Rules

- Variable format: `${metricName}`
- Only metrics within the **same category as the configured category** can be used as variables.
- You can combine multiple variables to compose a detailed message.
- The list of available variables can be checked in **Metric Query**.

Event Trigger Conditions

- **Data Query Range**: Select the real-time data query window.
  - Available ranges: `30 seconds`, `1 minute`, `2 minutes`, `5 minutes`, `10 minutes`, `30 minutes`, `1 hour`, `2 hours`, `3 hours`
- **Condition**: Enter the event trigger condition. Only fields included in the event data query can be used.

Additional Information

- **Interval**: The alert condition is evaluated at the selected interval (`1 minute`, `5 minutes`, `10 minutes`, `30 minutes`, `1 hour`, `12 hours`).
- **Silence**: After an event is triggered, the same event is suppressed for the selected duration (`unset`, `1 minute`, `5 minutes`, `10 minutes`, `30 minutes`, `1 hour`, `12 hours`).
- **Event Active Time**: Configure tags to control when the event runs, such as working hours, non-working hours, or maintenance windows. If no tag is set, the event runs continuously for 24 hours while it is enabled.

  1. Click **+ Add**.
  2. Under **Event Active Time**, click **+ Create New Tag**.
  3. In **Create Event Active Time Tag**, select a **Tag Name**, **Days**, **Time**, and **Color**, then click the [Apply] button.
  4. Created tags appear in the tag list and can be applied by selecting the checkbox.
  5. To edit or delete a tag, click the icon and modify or remove it in **Edit Event Active Time Tag**.

When a tag is deleted, it is removed from all users to whom it is applied. However, tags that are

currently in use by event rules cannot be deleted.

- **Event Recipient Tags**: When recipient tags are configured, notifications are sent to project members who have those tags. If no recipient tag is set, notifications are sent to all users in the project.
    1. Click **+ Add Tag** or the **+** icon.
    2. Click **+ Create New Tag** at the bottom of the **Event Recipient Tags** window.
    3. In the **Create Tag** window, enter a tag name, select a color, and click the [Create Tag] button.
    4. To edit or delete a created tag, click the icon in the tag list.
    5. Select the desired tag from the **Tag List** in the **Event Recipient Tags** window to apply it.

When configuring events, selecting event recipient tags allows notifications to be sent to project members and third-party plugins associated with those tags. You can assign tags to project members and third-party plugins separately via **Alert** > **Notifications**.

Test Event Rules

You can test the configured conditions against historical data to estimate how many events would have been triggered.

1. Select the date for testing the event rules and click the [Confirm] button.
2. Click the [Run] button to check the number of triggered events.

Edit/Delete Event

1. Go to the **Alert** > **Notifications** > **Composite metrics** tab.
2. In the composite metrics list, click the icon of the event you want to edit or delete.
3. In the **Notifications** screen, update the options and click the [Save] button.

    a. To delete the selected event, click the [Delete] button in the upper right corner of the **Notifications** screen.

Event Sharing

You can save metric event settings as a JSON file to share configurations with other users or import settings created by others.

- JSON file name: event-rules-`YYYYMMDD`.json

Export

1. Click the [**JSON**] button at the top right of the screen.
2. When the JSON editor opens, click the [Export] button.
    - If you use the export function after searching for events, only the searched list will be downloaded as a JSON file.
3. After the JSON file is downloaded, share it with other users.

Import

1. Click the [ ] button at the top right of the screen.
2. Select the JSON file downloaded using the **Export** function.
3. When the JSON editor opens, choose either [Add to List] or [Overwrite].

> It is recommended to use this feature between projects of the same product type. You can import event settings from projects of different products, but they may not function correctly.

Edit in JSON Format

1. Click the [**JSON**] button at the top right of the screen.
2. When the editor opens, modify the content according to the JSON format.
3. After completing the edits, click the [Save] button at the bottom of the screen.

> If the modified content does not conform to the JSON format, an error message will be displayed at the

bottom of the screen and the content cannot be saved. The error message may vary depending on the type of formatting issue.

JSON error

Alert Message Settings

You can select detailed fields for the messages to be received and edit the content for each field.

Appendix

MXQL Syntax Guide

Composite metrics operates based on MXQL queries. For detailed syntax, refer to the MXQL Syntax Guide.

Hitmap Pattern Event Configuration

Home > Select Project > **Alert** > **Event Configuration** > **Hitmap pattern** tab

Hitmap pattern automatically detects anomalies based on time-based pattern changes in application execution ranges, such as response time distribution and call density.

Without manually configuring complex conditions or thresholds, the system analyzes Hitmaps collected in real time and automatically triggers alert notifications when patterns deviate from normal behavior. Hitmap pattern alerts are especially useful in the following situations.

- When a specific transaction becomes excessively concentrated during certain time periods compared to normal behavior
- When traffic distribution or response patterns differ from the usual baseline
- When pattern-based anomaly detection is required beyond what condition-based rules can capture

Event Configuration

Configure the options for the event patterns to be detected.

Activation

When the toggle is turned on, an alert notification is sent when the corresponding pattern is detected.

- **Enabled**: An event is triggered when conditions are met.
- **Disabled**: No event is triggered, and conditions are not evaluated.

Silence

After an event is triggered, the event is suppressed for the selected duration and is not recorded in **Event History**.

Event Recipient Tags

When recipient tags are configured, notifications are sent to project members who have those tags. If no recipient tag is set, notifications are sent to all users in the project.

1. Click **+ Add Tag** or the **+** icon.
2. Click **+ Create New Tag** at the bottom of the **Event Recipient Tags** window.
3. In the **Create Tag** window, enter a tag name, select a color, and click the [Create Tag] button.
4. To edit or delete a created tag, click the icon in the tag list.
5. Select the desired tag from the **Tag List** in the **Event Recipient Tags** window to apply it.

Event Active Time

Configure tags to control when events run, such as working hours, non-working hours, or maintenance windows. If no tag is set, events run continuously for 24 hours while they are enabled.

1. Click **+ Add**.
2. Under **Event Active Time**, click **+ Create New Tag**.
3. In **Create Event Active Time Tag**, select a **Tag Name**, **Days**, **Time**, and **Color**, then click the [Apply] button.
4. Created tags appear in the tag list and can be applied by selecting the checkbox.
5. To edit or delete a tag, click the icon and modify or remove it in **Edit Event Active Time Tag**.

> When a tag is deleted, it is removed from all users to whom it is applied. However, tags that are currently in use by event rules cannot be deleted.

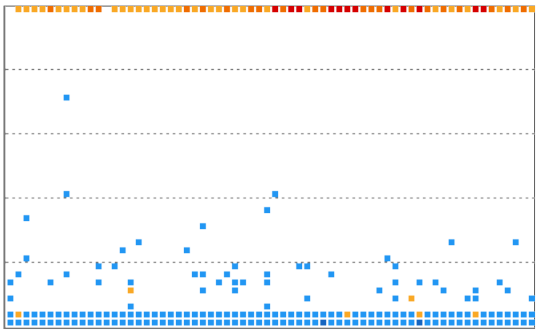
Hitmap Pattern Types

When you turn on the toggle for each pattern, an event is triggered when the corresponding pattern is detected.

Hitmap Horizontal Line Pattern

This pattern appears when multiple transactions complete within a similar response time range. Requests cluster around a specific latency band, which indicates increasing service delay or unstable responses from external resources.

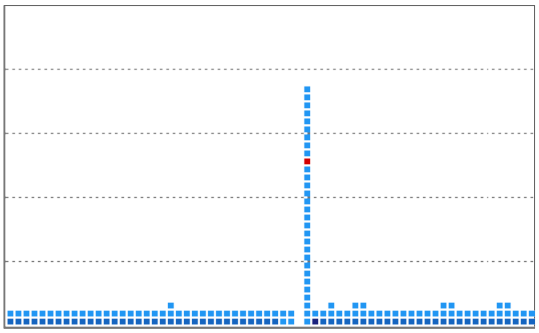
- Timeouts or delays occurring in external API, DB, or HTTPS calls
- Requests waiting for a certain duration before being processed all at once
- Repeated latency in shared I/O or network paths



Hitmap Vertical Line Pattern

In this pattern, transactions start at different times but finish simultaneously. Multiple requests are processed at the same moment, which indicates a bottleneck in a shared resource.

- Increased wait time due to shared resources such as DB locks or file locks
- Concurrent processing delays in queues or thread pools
- Temporary request suspension followed by simultaneous resumption due to system pauses such as GC pauses

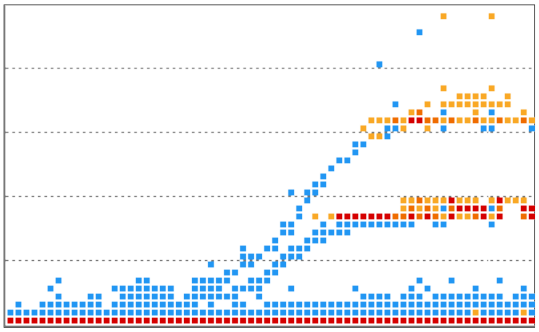


A Hitmap vertical line pattern is a signal that lock contention or bottleneck resources should be inspected.

Hitmap Flying Pattern

This pattern appears when response times rise and fall in waves at regular intervals due to shortages in shared resources such as specific system components or logs.

- Shortages in shared resources such as GC, connection pools, or file handles
- Periodic latency factors such as log flushing or storage I/O
- Batch jobs or background jobs that periodically consume resources

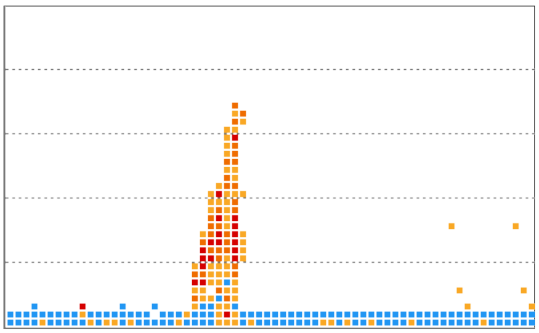


A Hitmap flying pattern strongly indicates periodic bottlenecks or resource shortages.

Hitmap Overload Pattern

This pattern occurs when transactions become abnormally concentrated within a specific response time range. It appears when requests fail to distribute normally and instead pile up in a narrow band due to temporary load spikes or early signs of failure.

- Transactions accumulate in a queue due to temporary failures or delays and are then processed in bulk
- Backend service response latency
- Insufficient processing capacity caused by traffic concentration during specific time periods

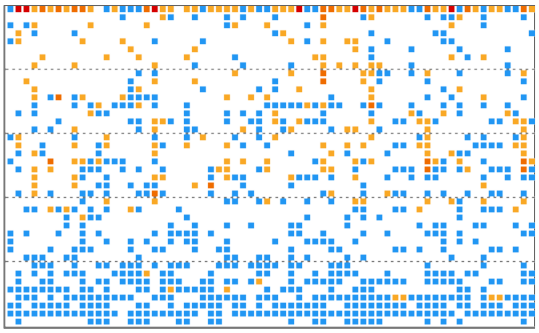


An overload pattern indicates temporary service quality degradation or load balancing failure.

Hitmap Flood Pattern

This pattern indicates an overall increase in response times caused by excessive requests or load. It reflects a broad performance degradation across the service.

- Sudden spikes in user traffic or automated traffic
- System resource shortages such as CPU, memory, or GC pressure
- Overall response quality degradation due to backend bottlenecks



A flood pattern requires immediate investigation.

Anomaly Detection Event Configuration

Home > Select Project > **Alert** > **Event Configuration** > **Anomaly detection** tab

Anomaly detection identifies abnormal rising or falling patterns in metric data and triggers events when movements that differ from the usual behavior are detected.

Default Options

- **Search**: Search for specific events by selecting an event name or an event recipient tag.
- **+ Add Event Rule**: Add a new anomaly detection event.

Table | Anomaly Detection Event List Layout

Item	Description
Event Message	Message sent when an event is triggered
Category	Unit used to classify metric data and serve as the basis for metric event configuration
Field	Metric field used to detect anomalies
Filter	Conditions that limit the data range subject to anomaly detection
Rising/Falling Pattern	Sensitivity of rising or falling patterns
Event Recipient Tag	Users or groups that receive notifications for the event
	Edit event settings
	Delete event
	Event enablement status - Enabled: Event is triggered when conditions are met - Disabled: Event does not run and conditions are not evaluated

Event Configuration

You can configure alert notifications to be sent when metric values rise or fall in a way that differs from their usual behavior.

Add Event Rules

1. Click the [+ Add Event Rule] button on the right side of the screen.
2. Configure **Event Basic Information**, **Metrics**, **Anomaly Detection**, and **Additional Information** in order.

Event Basic Information

1. Select the event level (Critical, Warning, Info).
   - This determines the event priority and alert level.
2. Enter the alert message to be delivered when the event is triggered.
   - You can use variables in the message to include actual values at the time the event occurs.

> **Available Variables**
>
> - `$pcode` : Project code
> - `$category` : Category
> - `$field` : Field
> - `$value` : Triggered value
> - `$oname` : Event target

Metrics

1. Select a **Category**.
   - This category serves as the baseline for the metric data used to detect anomalies.

2. Select the **Field** to be used for the event trigger conditions. Multiple selections are supported.
3. Select filters that define the event condition targets used to narrow the event scope. Multiple selections are supported.

Anomaly Detection

1. Select the sensitivity levels for **Rising Pattern** and **Falling Pattern**.
2. Use the toggle buttons to enable or disable each pattern.
   - To detect only rising patterns: Rising Pattern **ON**, Falling Pattern **OFF**
   - To detect only falling patterns: Falling Pattern **ON**, Rising Pattern **OFF**

Additional Information

Event Active Time

Configure tags to control when an event runs, such as working hours, non-working hours, or maintenance windows. If no tag is set, the event runs continuously for 24 hours while it is enabled.

1. Click **+ Add**.
2. Under **Event Active Time**, click **+ Create New Tag**.
3. In **Create Event Active Time Tag**, select a **Tag Name**, **Days**, **Time**, and **Color**, then click the [Apply] button.
4. Created tags appear in the tag list and can be applied by selecting the checkbox.
5. To edit or delete a tag, click the icon and modify or remove it in **Edit Event Active Time Tag**.

> When a tag is deleted, it is removed from all users to whom it is applied. However, tags that are currently in use by event rules cannot be deleted.

Event Recipient Tags

When event recipient tags are configured, notifications are sent to project members who have those tags. If no recipient tag is set, notifications are sent to all users in the project.

1. Click **+ Add Tag** or the **+** icon.
2. Click **+ Create New Tag** at the bottom of the **Event Recipient Tags** window.
3. In the **Create Tag** window, enter a tag name, select a color, and click the [Create Tag] button.
4. To edit or delete a created tag, click the icon in the tag list.
5. Select the desired tag from the **Tag List** in the **Event Recipient Tags** window to apply it.

> When configuring events, selecting event recipient tags allows notifications to be sent to project members and third-party plugins associated with those tags. You can assign tags to project members and third-party plugins separately via **Alert** > **Notifications**.

Edit Event

1. Go to the **Alert** > **Event Configuration** > **Anomaly detection** tab.
2. In the **Anomaly Detection** list, click the icon on the right of the event you want to edit.
3. In the **Anomaly Detection** window, update the options and click the [Save] button.

Delete Event

1. Go to the **Alert** > **Event Configuration** > **Anomaly detection** tab.
2. In the **Anomaly Detection** list, click the icon on the right of the event you want to delete.
3. In the delete confirmation window, click the [Delete] button.

Real-time Log Events

Home > Select Project > **Alert** > **Event Configuration** > **Real-time Log** tab

You can set alert notifications by filtering collected log data against conditions. Log monitoring must be activated to use log events. For more information on **activating log monitoring**, refer to the log documentation for each product.

- **Real-time log event**: An event is triggered when the configured event conditions are met in real-time collected logs.

Default options

- **Search**: Search for events by selecting an event name or event notification tag
- **JSON download/upload**: Event rules can be directly edited through the JSON editor, or downloaded as a file, modified, and re-uploaded
- **Notification message settings**: Select detailed items for the message to receive and edit the content for each item
- **+ Add event**: Add a new log event

Table | Real-time log event list configuration

Item	Description
Event name	Event name used as the notification title - Up to 255 bytes
Category	Log classification name (log folder name)
Search key	The field to search for in log data - Example: To access an HTTP response status code, use the search key `status`
Search value	The actual value corresponding to the search key. An alert is sent when logs contain the specified value

Item	Description
	- Example: If the search key is `status` and the search value is `200`, an alert is triggered when logs containing HTTP status code 200 are collected
Pause event notifications	Pauses the same event notification for a specified time after an event notification is triggered - Available times: Disabled, 5 min, 10 min, 15 min, 20 min, 30 min, 1 hour, 2 hours, 3 hours, 6 hours, 12 hours, 1 day - If resolved notifications are enabled, no notification is sent for the same event during the selected time after receiving a normal (RECOVERED) status notification
Event target filtering	Conditions that define the targets to which the event applies - If no condition is set, alerts are evaluated for all collected data - Select input: Build conditions by selecting tags, operators, and values - Direct input: Manually enter conditions to specify targets
Event notification tags	User or group (notification tag) to receive the event notification
	Edit and delete events
	Whether the event is active - Enabled: Event occurs when conditions are met - Disabled: Event does not operate and conditions are not checked

Event settings

Log events allow you to configure alert notifications when logs are collected in real time or when logs that meet specific conditions accumulate.

Add real-time log event

1. Click the **Real-time Log** tab in **Alert** > **Event Configuration**.

2. Click the [+ Add event] button on the right side of the screen.
3. Set the log event conditions in the **Add event rule** window.
   - **Event name**: Enter a name to use as the event title.
   - **Event activation**: When the toggle button is activated, an event occurs when conditions are met.
   - **Level**: Select the event level (Critical, Warning, Info).
   - **Message**: Enter the notification message to receive when the event is triggered. You can use variables in the message to include actual values at the time of event occurrence.

     > **Variable usage rules**
     > - Variable format: `${metricName}`
     > - Only metrics within the **same category as the configured category** can be used as variables.
     > - You can combine multiple variables to compose a detailed message.
     > - A list of available variables can be checked in **Metrics Query**.

   - **Category**: Select the log classification name (log folder name) from the list or enter it directly. You must select a category before entering a search key and search value.
   - **Search key**: Enter the field you want to search for in log data.
     - Example: To access an HTTP response status code, use the search key `status`.
   - **Search value**: Enter the actual value corresponding to the **search key**. An alert is sent when logs contain the specified value.
     - Example: If the **search key** is `status` and the **search value** is `200`, an alert is triggered when logs containing HTTP status code 200 are collected.
   - **Event target filtering**: Enter conditions to define the targets to which the event applies. If no value is entered, alerts are evaluated for all collected data.
     - **Select input**: Select a tag, operator, and value to set the target conditions.

       > For condition expression syntax and supported operators, refer to the Condition

> ⓘ settings guide.

  - **Direct input**: Enter conditions directly.

- **Pause event notifications**: Select a duration to temporarily suppress alerts. After the event is triggered, the same event will not occur during the selected time.
  - Available times: `Disabled` , `1 min` , `2 min` , `3 min` , `5 min` , `10 min` , `20 min` , `30 min` , `1 hour` , `3 hours` , `6 hours` , `12 hours`
- **Event operation time**: Set tags to activate the event during specific time periods (work/non-work/maintenance hours, etc.). If no tag is set, the event operates 24 hours a day when activated.
  1. Click **+ Add**.
  2. Click **+ Create new tag** in **Event operation time**.
  3. In **Create event operation time tag**, select the **tag name, day of week, time, and color**, then click the [Apply] button.
  4. Created tags can be viewed in the tag list and are applied by selecting the checkbox.
  5. Click the ✎ icon for the tag to modify or delete, and edit or delete the tag in **Edit event operation time tag**.

> ⚠ Deleting a tag removes the tag for all users to whom it has been applied. However, tags in use by event rules cannot be deleted.

- **Event notification tags**: If notification tags are set for an event, notifications are sent to project members with those tags. If no recipient tags are set, notifications are sent to all users in the project.
  1. Click **+ Add tag** or **+**.
  2. Click **+ Create new tag** at the bottom of the **Event notification tags** window.
  3. In the **Create tag** window, enter the tag name, select a color, and click the [Create tag] button to create the tag.

4. Edit or delete created tags by clicking the icon in the tag list.
5. Select the desired tag from the **tag list** in the **Event notification tags** window to apply it.

> When configuring events, you can send notifications to project members and 3rd-party plugins with the selected event notification tags. You can assign tags to project members and 3rd-party plugins respectively in **Alert notifications > Event notification settings**.

4. When configuration is complete in the **Add event rule** window, click the [Save] button.

Edit/delete events

1. Go to the **Real-time log** tab in **Alert** > **Event Configuration**.
2. Click the icon for the event to edit or delete in the log event list.
3. Modify options in the **Event settings** screen and click the [Save] button.

   a. To delete the selected event, click the [Delete] button in the upper right of the **Event settings** screen.

Event sharing

You can save metric event settings as a JSON file to share configurations with other users or import settings created by others.

- JSON file name: event-rules-`YYYYMMDD`.json

Export

1. Click the [**JSON**] button at the top right of the screen.
2. When the JSON editor opens, click the [Export] button.
   - If you use the export function after searching for events, only the searched list will be downloaded

as a JSON file.

3. After the JSON file is downloaded, share it with other users.

Import

1. Click the [ ] button at the top right of the screen.
2. Select the JSON file downloaded using the **Export** function.
3. When the JSON editor opens, choose either [Add to List] or [Overwrite].

> It is recommended to use this feature between projects of the same product type. You can import event settings from projects of different products, but they may not function correctly.

Edit in JSON Format

1. Click the [**JSON**] button at the top right of the screen.
2. When the editor opens, modify the content according to the JSON format.
3. After completing the edits, click the [Save] button at the bottom of the screen.

> If the modified content does not conform to the JSON format, an error message will be displayed at the bottom of the screen and the content cannot be saved. The error message may vary depending on the type of formatting issue.
>
> JSON error

Appendix

Trigger Conditions and Target Selection Guide

Metric alert trigger conditions and event target selection use the same syntax. However, trigger conditions use the field key as the variable, while target selection uses the tag key as the variable.

Level Behavior Guide

Event levels are categorized into Critical, Warning, and Info.

1. **Priority-based Event Triggering**

   When multiple level conditions are met at the same time, only the event with the highest priority is triggered.

   Example

   ```
   Configuration:
   - Warning: CPU > 70%
   - Critical: CPU > 90%

   Current state: CPU 95%
   → Result: Only the Critical event is triggered (Warning is suppressed)
   ```

2. **Level Escalation**

   For events that maintain an in-progress state, when a higher-level condition is met while a lower-level event is already in progress, both levels enter the in-progress state.

   Example

   ```
   Configuration:
   - Warning: CPU > 70%
   - Critical: CPU > 90%

   Scenario:
   1) CPU 80% → Warning triggered (in progress)
   2) CPU 95% → Warning (in progress), Critical (in progress)
   ```

3. **Level De-escalation**

   For events with an in-progress state, when the level transitions from a higher level to a lower level, the higher-level event is resolved while the lower-level event remains active.

   Example

```
Configuration:
- Warning: CPU > 70%
- Critical: CPU > 90%

Scenario:
1) CPU 97% → Critical triggered
2) CPU 85% → Critical resolved, Warning remains
3) CPU 65% → Warning resolved
```

4. **Execution Flow**

```
- Warning: CPU > 70%
- Critical: CPU > 90%

Metric value: 60%  →  75%  →  92%  →  85%  →  60%
State:      Info      Warning     Critical     Warning     Info
                    (triggered)  (triggered)  (maintained)  (resolved)
                                  Warning
                                (maintained)
```

> Events that have an in-progress state remain active as long as their conditions are met and continue until those conditions are cleared.

Condition Configuration Guide

> If a metric name starts with a number or contains special characters, you must wrap it in the `${metric_name}` format.
>
> - Brackets `()`, `[]`
> - Operators `+`, `-`, `*`, `/`, `%`
> - Delimiters `:`, `@`, `#`, `,`, (space)
> - Other special characters `!`, `^`, `&`, `|`, `~`, `` ` ``, `=`
>
> **Examples using `${}`:**

```
${cpu(xos)} > 50
${mem[0]} >= 100
${cpu-usage} > 80
${namespace:cpu} > 70
${metric name} > 60
${4xx_error} > 10
```

Usable without `${}` : Metric names that contain only alphabetic characters, underscores (`_`), or dots (`.`)

```
cpu > 80
cpu_usage > 50
CPUUtilization.Average > 0.8
```

1. **Comparison Operators**

   Numeric Comparison

   ```
   Greater than:          cpu > 80
   Greater than or equal:   cpu >= 80
   Less than:             memory < 1000
   Less than or equal:      memory <= 1000
   Equal to:              status == 200
   Not equal to:          error != 0
   ```

   String Comparison

   ```
   status == 'OK'
   region == "us-east-1"
   ```

2. **Arithmetic Operators**

   - Basic arithmetic operations

     ```
     Addition:       cpu + 10 >= 90
     Subtraction:    memory - 100 >= 500
     Multiplication: cpu * 2 >= 100
     ```

```
Division:       disk / 1024 >= 100
Modulo:         value % 10 == 0
```

- Controlling precedence with parentheses

```
(cpu + memory) * 2 >= 200
(disk - used) / total >= 0.2
((cpu + memory) / 2) >= 50
```

- Negative values

```
cpu > -100
```

3. Logical Operators

- AND operator (&&)

```
cpu > 80 && memory > 1000
${cpu(xos)} > 50 && ${mem(xos)} > 60
```

- OR operator (||)

```
cpu > 90 || memory > 90
disk < 10 || network > 1000
```

- Combined logical operations

```
(cpu > 50 && memory > 50) || disk < 20
cpu > 80 && (memory > 1000 || disk > 500)
```

4. Pattern Matching Operators

- LIKE operator

```
oname like 'prod-*'
url like '*error*'
```

```
message like 'WARN%'
```

- NOT LIKE operator

```
oname not like 'test-*'
url not like '*debug*'
```

5. Built-in Functions

- Null check functions

```
isNull(value)           # Checks whether the value is null
isNotNull(value)        # Checks whether the value is not null
nvl(value, 0)           # Returns a default value if null
isEmpty(str)            # Checks whether the string is empty
isNotEmpty(str)         # Checks whether the string is not empty
```

- Aggregate functions

```
sum(cpu, memory, disk)            # Sum
avg(cpu, memory)                  # Average
max(cpu, memory, disk)            # Maximum
min(cpu, memory, disk)            # Minimum
count(value1, value2, value3)     # Count
```

- Math functions

```
round(cpu, 2)           # Rounds to 2 decimal places
```

- String functions

```
length(str)             # String length
startsWith(str, 'prefix')  # Checks prefix
endsWith(str, 'suffix')    # Checks suffix
indexOf(str, 'search')     # String index
substring(str, 0, 10)      # Substring
```

```
trim(str)               # Trims whitespace
replace(str, 'old', 'new') # String replacement
hasStr(str, 'search')     # Checks string containment
```

- Conditional functions
  - **if**: A conditional function that returns the `trueValue` if the condition evaluates to `true`, or the `falseValue` if it evaluates to `false`

    Syntax

    ```
    if(condition, trueValue, falseValue)
    ```

    Example

    ```
    if(value >= 90, 'High', 'Medium') == 'High'
    ```

  - **decode**: Compares a value against conditions and returns the result of the matching condition, or a default value if no match is found

    Syntax

    ```
    decode(value, condition1, result1, condition2, result2, ..., conditionN, resultN, defaultValue)
    ```

    Example

    ```
    decode(value, 1, 'Low', 2, 'Medium', 3, 'High', 'Unknown') == 'Unknown'
    ```

  - **in**: Returns `true` if the value matches any of the specified candidates; otherwise returns `false`

    Syntax

    ```
    in(value, candidate1, candidate2, ..., candidateN)
    ```

    Example

```
in(status, 200, 201, 204) == false
```

Writing Guidelines

- Recommended
  - Use simple variable names without `${}` (e.g. cpu, memory)
  - Always use `${}` when special characters are included
  - Use parentheses to clearly separate complex expressions
  - Enter function names exactly as defined (case-sensitive)
- Avoid
  - Incomplete expressions (e.g. `cpu >` , `memory &&`)
  - Mismatched parentheses (e.g. `(cpu > 80 && cpu > 70)`)
  - Consecutive or invalid operators (e.g. `cpu >> 80` , `cpu >< 80`)

Composite Log Event Settings

Home > Select Project > **Alert** > **Event Configuration** > **Composite Log** tab

You can set alert notifications by filtering collected log data against conditions. Log monitoring must be activated to use log events. For more information on **activating log monitoring**, refer to the log documentation for each product.

- **Composite log event**: An event is triggered when the number of logs that meet the log search conditions within the configured **data query range** reaches or exceeds the threshold count.

> **Permission**
>
> **Alert configuration** permission is required to configure composite metrics events. Alert configuration permission can be set by clicking the icon for the member to grant permission in **Home Screen > Select Project > Management > Integrated Member Management**.

Default options

- **Search**: Search for events by selecting an event name or event notification tag
- **JSON download/upload**: Event rules can be directly edited through the JSON editor, or downloaded as a file, modified, and re-uploaded
- **Notification message settings**: Select detailed items for the message to receive and edit the content for each item
- **+ Add event**: Add a new log event

Table | Composite log event list configuration

Item	Description
Event name	Event name used as the notification title
Template	Composite log template

Item	Description
Category	Log classification name (log folder name)
Rule	Conditions that trigger the event
Additional notification when event is resolved	Whether to receive notifications in normal (RECOVERED) status when Critical and Warning level events are resolved - Enabled: Sends a RECOVERED notification when transitioning from in-progress → normal status, and the in-progress/resolved status can be distinguished in event history - Disabled: A one-time event occurs each time the threshold is exceeded
Pause event notifications	Pauses the same event notification for a specified time after an event notification is triggered - Available times: Disabled, 5 min, 10 min, 15 min, 20 min, 30 min, 1 hour, 2 hours, 3 hours, 6 hours, 12 hours, 1 day - If resolved notifications are enabled, no notification is sent for the same event during the selected time after receiving a normal (RECOVERED) status notification
Interval	Time range for querying log data
Event notification tags	User or group (notification tag) to receive the event notification
	Edit and delete events
	Whether the event is active - Enabled: Event occurs when conditions are met - Disabled: Event does not operate and conditions are not checked

Event settings

The composite log event feature triggers alert notifications when the number of logs matching the log search conditions within the configured **data query range** reaches or exceeds the threshold count.

Add composite log event

1. Click the **Composite Log** tab in **Alert** > **Event Configuration**.
2. Click the [+ Add event] button on the right side of the screen.
3. Set the log event conditions in the **Add event rule** window. For details, refer to Basic settings, Detail settings, and Additional settings.

   Add composite log event
4. When configuration is complete in the **Add event rule** window, click the [Save] button.

Basic settings

- **Event name**: Enter a name to use as the event title.
- **Event activation**: When the toggle button is activated, an event occurs when conditions are met.
- **Level**: Select the event level (Critical, Warning, Info).

Detail settings

Configure Template (preset), Message, Category, **Filter**, and Event trigger conditions.

Template (preset)

Selecting a predefined preset automatically configures the **filter, query type, and event trigger conditions**. You can modify the settings after selecting a preset.

Composite log template types

Preset	Query type	Default rule
None	none	`rows > 10`
2xx status code count	count	`count > 10`

Preset	Query type	Default rule
3xx status code count	count	`count > 10`
4xx status code count	count	`count > 10`
5xx status code count	count	`count > 10`
Normal status code (2xx, 3xx) count	count_v2	`include_minus_exclude_count > 10`
Error status code (4xx, 5xx) count	count_v2	`include_minus_exclude_count > 10`
Error received count	count	`count > 10`
Exception received count	count	`count > 10`
Log count per second	rps	`rps > 100`
Field aggregation (Aggregation)	aggr	`avg > 1000`
Rate	rate	`rate > 10`
Unique value count (Cardinality)	cardinality	`cardinality > 100`
Percentile	percentile	`p99 > 3000`

Message

Enter the notification message to receive when an event occurs. You can use variables in the message to compose dynamic notifications.

- **Example**: Message `${host}` error `${count}` occurred → Notification message web-01 error 15 occurred

Table | Message variables

Variable	Description	Usage condition
`${count}`	Number of logs matching the condition	Always available
`${groupField}`	Grouping field name	Grouped notifications only
`${groupValue}`	Grouping field value	Grouped notifications only
`${fieldName}`	Grouping field value (referenced directly by field name)	Grouped notifications only

> The `${count}` variable can be used in all notifications regardless of grouping. The `${groupField}`, `${groupValue}`, and `${fieldName}` variables are only substituted when a grouping field is configured.

Category

Select the log classification name (log folder name) from the list or enter it directly. You can enter the search key and search value only after selecting a category.

Filter

Enter log search conditions using Lucene query syntax. Two input modes are supported.

- **Filter mode**: Build a query by selecting fields and values.
- **Code mode**: Enter Lucene queries directly.

Table | Lucene query examples

Query	Description
`level:ERROR`	Logs where the level field is ERROR
`status_nxx:status_4xx OR status_nxx:status_5xx`	4xx or 5xx status codes
`appender:accessLog`	Logs from the accessLog appender
`response_time.n:*`	Logs where the response_time.n field exists

Query	Description
`host:web-01 AND level:ERROR`	Error logs from host web-01

Event trigger conditions

Configure by query type. Additional settings are displayed depending on the selected template (preset) or query type.

Table | Event trigger condition key meanings

Key	Meaning
`rows` / `count`	Number of logs matching the filter condition
`rps`	Log count per second (count / interval in seconds)
`avg`	Average value of the target field
`sum`	Sum of the target field values
`min`	Minimum value of the target field
`max`	Maximum value of the target field
`rate`	Numerator count / denominator count × 100 (%)
`cardinality`	Number of unique values in the target field
`p99` / `p95` / `p90` / `p75` / `p50`	Percentile values of the target field
`include_minus_exclude_count`	Include filter count − exclude filter count

- **Count (count / none)**: Counts the number of logs matching the configured filter conditions.

  Table | Count settings

Setting	Description
Filter	Log search condition (Lucene query)
Event trigger condition	`count` or `matching logs` `{operator}` `{threshold}` (e.g., `count > 10`)

- **Log count per second (rps)**: Calculates the number of logs per second (Requests Per Second) matching the configured filter conditions. The total count during the interval is divided by the interval in seconds.

Table | Log count per second settings

Setting	Description
Filter	Log search condition (Lucene query)
Event trigger condition	`RPS` `{operator}` `{threshold}` (e.g., `rps > 0.5`)

- **Field aggregation (aggr)**: Calculates the aggregated value (average, sum, min, max) of the specified numeric field. The aggregation method can be selected from a dropdown in the event trigger condition.

Table | Field aggregation settings

Setting	Description
Filter	Log search condition (Lucene query)
Target field	Numeric field to aggregate (e.g., `response_time.n`)
Event trigger condition	`avg` / `sum` / `min` / `max` `{operator}` `{threshold}` (e.g., `avg > 1000`)

- **Rate**: Calculates the percentage (%) ratio of the numerator filter count to the denominator filter count. Useful for monitoring the ratio of specific conditions against all logs.

Table | Rate settings

Setting	Description
Filter (numerator)	Log condition for the numerator (e.g., `level:ERROR`)
Denominator filter	Log condition for the denominator (e.g., `level:*`)
Event trigger condition	`Rate (%)` `{operator}` `{threshold}` (e.g., `rate > 10`)

  - **Example**: Filter `level:ERROR`, denominator filter `level:*`, rule `rate > 10` → Alert triggered when the error rate among all logs exceeds 10%

- **Unique value count (cardinality)**: Counts the number of unique values in the specified field. Useful for detecting an abnormally large variety of values.

Table | Unique value count settings

Setting	Description
Filter	Log search condition (Lucene query)
Target field	Field to calculate unique values for (e.g., `host`)
Event trigger condition	`Unique value count` `{operator}` `{threshold}` (e.g., `cardinality > 100`)

- **Percentile**: Calculates the percentile value of the specified numeric field. Useful for monitoring tail latency in response times. Changing the percentile selection automatically updates the key in the event trigger condition.

Table | Percentile settings

Setting	Description
Filter	Log search condition (Lucene query)
Target field	Numeric field to calculate percentile for (e.g., `response_time.n`)
Percentile	Select from P99, P95, P90, P75, P50

Setting	Description
Event trigger condition	`P99` / `P95` /... `{operator}` `{threshold}` (e.g., `p99 > 3000`)

Table | Percentile meanings

Percentile	Meaning
P50	Median. 50% of all data is at or below this value
P75	75% of all data is at or below this value
P90	90% of all data is at or below this value
P95	95% of all data is at or below this value
P99	99% of all data is at or below this value

- **Include/exclude count (count_v2)**: Calculates the net count by subtracting the exclude filter count from the include filter count.

Table | Include/exclude count settings

Setting	Description
Filter (include)	Log conditions to include (e.g., `status_nxx:status_4xx OR status_nxx:status_5xx`)
Filter exclude	Log conditions to exclude (e.g., `status_nxx:status_4xx`)
Denominator filter	Denominator for ratio calculation (optional)
Event trigger condition	`Include-exclude count` `{operator}` `{threshold}` (e.g., `include_minus_exclude_count > 10`)

  - **Example**: Filter `status_nxx:status_4xx OR status_nxx:status_5xx`, filter exclude `status_nxx:status_4xx` → Alert triggered when the net 5xx count (4xx+5xx minus 4xx) exceeds 10

Group by field (Group By)

When a group by field is configured, alert conditions are evaluated independently for each value of that field. Select the field to group by from the dropdown. Clicking the dropdown shows a list of available fields for the current category with autocomplete.

- **Example**: Filter `level:ERROR`, group by field `host`, rule `rows > 3`

Table | Grouping comparison

Item	Without grouping	With grouping
Evaluation unit	All logs aggregated as one	Independent aggregation per group value
Alert occurrence	1 alert	One per group meeting the condition
Alert title	`Event title`	`[Group value] Event title`
Stateful	State transition based on all logs	Independent state transition per group

  - `[web-01] Error alert` — 8 errors from web-01 → alert triggered
  - `[web-02] Error alert` — 5 errors from web-02 → alert triggered
  - `[api-01] Error alert` — 2 errors from api-01 → condition not met, no alert

> ⓘ When **Additional notification when event is resolved** is enabled, the ABNORMAL → RECOVERED state transition occurs independently per group. For example, when errors from `web-01` are resolved, a `[web-01] RECOVERED` notification is triggered, with no effect on other hosts.

Additional settings

- **Interval**: Check alert conditions at the selected time interval (`unset`, `1 min`, `5 min`, `10 min`, `30 min`, `1 hour`, `12 hours`).
- **Mute**: After an event occurs, the same event is not triggered during the selected time (`unset`, `1 min`, `5 min`, `10 min`, `30 min`, `1 hour`, `12 hours`).

- **Event operation time**: Set tags to activate the event during specific time periods (work/non-work/maintenance hours, etc.). If no tag is set, the event operates 24 hours a day when activated.

    1. Click **+ Add**.
    2. Click **+ Create new tag** in **Event operation time**.
    3. In **Create event operation time tag**, select the **tag name, day of week, time, and color**, then click the [Apply] button.
    4. Created tags can be viewed in the tag list and are applied by selecting the checkbox.
    5. Click the icon for the tag to modify or delete, and edit or delete the tag in **Edit event operation time tag**.

    > Deleting a tag removes the tag for all users to whom it has been applied. However, tags in use by event rules cannot be deleted.

- **Event notification tags**: If notification tags are set for an event, notifications are sent to project members with those tags. If no recipient tags are set, notifications are sent to all users in the project.

    1. Click **+ Add tag** or **+**.
    2. Click **+ Create new tag** at the bottom of the **Event notification tags** window.
    3. In the **Create tag** window, enter the tag name, select a color, and click the [Create tag] button to create the tag.
    4. Edit or delete created tags by clicking the icon in the tag list.
    5. Select the desired tag from the **tag list** in the **Event notification tags** window to apply it.

    > When configuring events, you can send notifications to project members and 3rd-party plugins with the selected event notification tags. You can assign tags to project members and 3rd-party plugins respectively in **Alert notifications** > **Event notification settings**.

Simulation

You can simulate using historical data before saving the alert to check the expected number of alert occurrences.

1. After completing the alert configuration, select a time range at the bottom.
2. Click the [Simulation] button.
3. The chart displays data by time period and the number of alert occurrences.

   > You can adjust the simulation time range to check alert occurrence patterns across different time periods. Data from the last 1–2 minutes may not be available due to index build delays.

4. When configuration is complete in the **Add event rule** window, click the [Save] button.

Create alert from Log Explorer

You can create alerts directly from Log Explorer.

1. Navigate to **Log** > **Log Search** menu.
2. Configure the category and search query.
3. Click the [Add event] button in the upper right of the screen.
4. The alert configuration form opens in a drawer, with the current category and search query automatically filled in.
5. Complete additional settings and click the [Save] button.
6. After saving, click the [Go] button to navigate to the alert list.

Edit/delete events

1. Go to the **Composite log** tab in **Alert** > **Event Configuration**.
2. Click the icon for the event to edit or delete in the log event list.
3. Modify options in the **Event settings** screen and click the [Save] button.

   a. To delete the selected event, click the [Delete] button in the upper right of the **Event settings** screen.

Event sharing

You can save metric event settings as a JSON file to share configurations with other users or import settings created by others.

- JSON file name: event-rules-`YYYYMMDD`.json

Export

1. Click the [**JSON**] button at the top right of the screen.
2. When the JSON editor opens, click the [Export] button.
    - If you use the export function after searching for events, only the searched list will be downloaded as a JSON file.
3. After the JSON file is downloaded, share it with other users.

Import

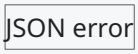
1. Click the [ ] button at the top right of the screen.
2. Select the JSON file downloaded using the **Export** function.
3. When the JSON editor opens, choose either [Add to List] or [Overwrite].

> It is recommended to use this feature between projects of the same product type. You can import event settings from projects of different products, but they may not function correctly.

Edit in JSON Format

1. Click the [**JSON**] button at the top right of the screen.
2. When the editor opens, modify the content according to the JSON format.
3. After completing the edits, click the [Save] button at the bottom of the screen.

If the modified content does not conform to the JSON format, an error message will be displayed at the bottom of the screen and the content cannot be saved. The error message may vary depending on the type of formatting issue.



Transaction Event Configuration

Home > Select Project > **Alert** > **Event Configuration** > **Transaction** tab

Transaction events provide fine-grained monitoring of transactions based on specific URLs, errors, status codes, and more. When transactions that meet the configured conditions are detected, an event is triggered. Transaction data is analyzed in 1-minute intervals, and notifications are sent when the number of transactions that satisfy the conditions exceeds the defined threshold.

Latest Agent Versions

To use transaction events, you must install an agent that meets or exceeds the minimum supported version.

Language	Minimum Supported Version
Java	2.2.50 or later
PHP	Not specified (use the latest version or contact support)
Node.js	0.5.21 or later
Python	1.7.8 or later
.NET	2.3.5 or later
Go	Not specified (use the latest version or contact support)

Default Options

- **Event Level**: Indicates the severity of the event and is classified into three levels: Info, Warning, and Critical.
- **Search**: Search for events by event name or event reception tag.
- **+ Add Event**: Add a new transaction event.

- **Resize Columns**: Drag the boundary of each column to adjust its width.

Table | Transaction Event List Structure

Item	Description
No.	Event sequence number
	Whether the event is enabled - Enabled: The event is triggered when conditions are met - Disabled: The event does not operate and conditions are not evaluated
	Edit or delete event settings
Event Name	User-defined event name - Up to 255 bytes
Condition Type	Type of event trigger condition (elapsed time, error occurrence, pattern only)
Aggregation Basis	Criteria used to aggregate transaction counts
Target Pattern	Include/exclude patterns to which the event applies (URL, status code, etc.)
Trigger Count	Number of occurrences required to trigger the event - Aggregation period: fixed at 1 minute (based on each minute from :00 to :59 seconds; the period cannot be changed) - The event is triggered when transactions that meet the condition occur at least the configured number of times within that 1-minute period
Pause	Temporarily suppresses repeated occurrences of the same event after an alert is sent - Available durations: Disabled, 5 min, 10 min, 15 min, 20 min, 30 min, 1 hr, 2 hr, 3 hr, 6 hr, 12 hr, 1 day - If recovery notifications are enabled, no alert is sent for the same event during the selected time after a RECOVERED notification is received

Item	Description
Event Recipients	Users or groups that receive the event notification (reception tags)

Event Configuration

Transaction events are configured through the steps of **event conditions, target patterns, and basic information and notification settings**.
With event configuration, you can detect specific transaction patterns and receive alerts.

Add Event

To add a transaction event, go to **Event Configuration** > **Transaction** tab, then click the [+ Add Event] button in the upper-right corner of the screen.

How to Add a Transaction Event
Step 1. Define Event Conditions: Define the criteria that trigger the event.
Step 2. Select Event Target: Select the monitoring targets where the event applies.
Step 3. Basic Information and Notification Settings: Configure the event name, message, and recipients.

Define Event Conditions

Configure the conditions under which a transaction event is triggered.

Condition Type

Select the criteria used to trigger the event.

- **Elapsed Time**: Transactions that take longer than the specified time (ms).
  - **Elapsed Time Threshold**: When the condition type is Elapsed Time, enter the value in milliseconds. An

event is triggered when a transaction takes longer than the configured time (ms).

- **Error Occurred**: Transactions in which an error occurs.
- **Pattern Only**: Transactions that match a specific pattern (such as hitmap patterns).

Aggregation Criteria

Select the unit by which alert conditions are aggregated.

- Aggregation criteria: per agent, per URL of each agent, per error class of each agent, per business of each agent, per URL, per error class, per business, or overall.

Trigger Count (Required)

Configure how many times the condition must be met before a notification is sent.

Example: trigger count **1** sends a notification immediately if even one transaction meets the condition within 1 minute, while trigger count **3** requires at least 3 transactions meeting the condition to accumulate within the same 1 minute before a notification is sent.

Pause

Select a pause duration to prevent excessive alert notifications.

- After an event occurs, the same event will not trigger again during the configured time period.
- Available durations: `1 minute` , `5 minutes` , `10 minutes` , `15 minutes` , `30 minutes` , `1 hour` , `2 hours` , `3 hours` , `6 hours` , `12 hours` , `1 day`

Event Operation Time

Configure the time periods during which the event is active. This allows you to **suppress alerts during non-business hours or maintenance windows**. If no tag is configured, the event runs continuously for 24 hours while enabled.

1. Click **+ Add**.
2. In Event Operation Time, click **+ Create New Tag**.
3. In **Create Event Operation Time Tag**, select the **tag name, days, time, and color**, then click the [Apply]

button.

4. The created tag appears in the tag list. Select the checkbox to apply it.
5. To edit or delete a tag, click the icon and update or remove it in **Edit Event Operation Time Tag**.

> When a tag is deleted, it is removed from all alerts where it is applied. However, tags currently in use by event settings cannot be deleted.

Select Event Target

Configure patterns for transaction filtering.

Pattern Configuration

- You can enter multiple patterns, separated by pressing `Enter`.
- If no pattern is configured, events apply to all transactions in the project.
- Patterns of the same type are applied with an **OR** condition.
- Patterns of different types are applied with an **AND** condition.
- You can use wildcards (`*`) in patterns. Up to two wildcard positions are allowed.
- Each pattern must include at least one character other than the wildcard (`*`) and slash (`/`).

> **Be careful when combining slashes (`/`) and wildcards (`*`)**
>
> - `/api/*` : `/api` alone does not match. A path must follow `/api/` (e.g., ✅ /api/users, ❌ /api)
> - `/api*` : `/api` itself also matches (e.g., ✅ /api, ✅ /api/users, ✅ /apitest)

Match Patterns

Filter data that matches the specified items. After being applied, match patterns are displayed as blue tags.
You can enter up to 10 patterns for each item.

Item	Description
Operation Name	Operation name pattern
URL	Transaction URL pattern - Example: `/api/*`, `*/users/*`
Error Class	Error class name pattern - Example: `*Exception`, `*Error`
Error Message	Error message content pattern
Status Code	HTTP status code pattern - Example: `4**`, `5**`

Exclude Patterns

Exclude data that matches the specified items from the filtering targets. After being applied, exclude patterns are displayed as red tags and use the same format as match patterns. You can enter up to 10 patterns for each item.

Pattern Examples

- **Exact Match**

```
Pattern: "hello"
"hello"    → Match
"Hello"    → No match (case-sensitive)
```

- **Prefix Match**

```
Pattern: "hello*"
"hello"      → Match
"helloworld"  → Match
"hi"         → No match
```

- **Suffix Match**

```
Pattern: "*world"
"world"      → Match
"helloworld" → Match
"hello"      → No match
```

- **Substring Match**

```
Pattern: "*ell*"
"hello"      → Match (h'ell'o)
"excellent"  → Match (exc'ell'ent)
"welcome"    → No match (does not contain "ell")
```

- **Composite Patterns**

```
Pattern: "hello*world"
"hellobeautifulworld" → Match

Pattern: "hello*test*"
"hellomytestcase"     → Match

Pattern: "*test*world"
"mytestcaseworld"     → Match
```

Table | Practical Usage Examples

Pattern Type	Examples
URL Patterns	- `/api/*`: All URLs that start with `/api/` - `/users/*`: All URLs that contain `users` - `/api/v1/users`: Only the URL that matches exactly
Error Class Patterns	- `Exception`: All errors that end with `Exception` - `java.lang.*`: All classes that start with `java.lang` - `Timeout*`: All errors that contain `Timeout`
Status Code Patterns	- `4**`: All 4xx status codes - `404`: Only the 404 status code

Pattern Type	Examples
	- `5**`: All 5xx status codes

Pattern Limitations

- **Required characters**: Each pattern must include at least one character other than the wildcard (`*`). A pattern cannot consist of wildcards only.
  - ❌ Not allowed: `**` , `*`
  - ✅ Allowed: `a*` , `a` , `a*` , `/api*`
- **Empty strings**: Empty strings or null values are not matched.
- **Consecutive wildcards**: Consecutive `*` characters are treated as a single wildcard.
- **Complex patterns**: Patterns containing three or more wildcards are not supported.
- **No regular expressions**: Character classes (`[a-z]`), quantifiers (`+` , `?`), and grouping (`( )`) are not supported.
- **Case sensitivity**: All pattern matching is case-sensitive.
- **No escaping**: There is no way to escape characters for literal matching.

Basic Information and Notification Settings

Configure the basic information of the event and how notifications are delivered.

Event Activation

Configure whether notifications are sent when an event occurs.

- **Enabled**: Notifications are sent when the configured conditions are met.
- **Disabled**: Notifications are not sent even if the configured conditions are met.

Event Name (Required)

Enter the event name to be used as the notification title.

Event Level (Required)

Select the event level (Info, Warning, Critical).

Event Recipients

Specify the members who will receive event notifications.

- **Receive All** **Receive by Tag**

Notifications are sent to all targets (members and channels) with event reception enabled in the project.

Notifications are sent to project members who have the selected tag.

1. Select **Receive by Tag**.
2. In the **Receiver Tags** section, click [+ Add Tag] or [+].
3. Click **+ Create New Tag** at the bottom of the **Event Receiver Tags** window.
4. In the **Create Tag** window, enter a tag name, choose a color, and click [Create Tag].
5. To edit or delete a created tag, click the icon in the tag list.
6. In the **Event Receiver Tags** window, select the desired tag from the **Tag List** to apply it.

> **Receiver Tags**
>
> Receiver tags are a feature for managing notification recipients (users or channels) at a group level.
> When you assign receiver tags to an event, notifications are delivered only to users who have those tags.
> If no receiver tag is set, notifications are sent to all users in the project.

Reference. Example of Event Behavior

This is an example of an actual transaction event occurrence.

> **Configuration Conditions**

- Aggregation criteria: By agent
- Occurrence count: 2 times
- Pause: 5 minutes
- URL patterns: `/api/posts/**` , `/delete/*` , `/*test*/`
- Error class patterns: `*TimeoutException*` , `*NullPointerException`
- Status code pattern: `4**`
- Excluded error class: `*IOException`
- Excluded status code: `5**`

Example of a Triggered Notification Message

The transaction information in the event message displays the values from the most recent transaction that matched the conditions. The event message includes both the configured conditions and the details of the actual transaction that triggered the alert, allowing you to clearly identify which conditions caused the notification.

```
(ignored count: 3)
[Per Agent] limit: 2, count: 3
[Condition]
url patterns: /api/posts/**, /delete/*, */test/*
error class patterns: *TimeoutException*, *NullPointerException
status patterns: 4**
exclude error class patterns: *IOException
exclude status patterns: 5**
[Transaction]
url: /api/posts/test/timeout
txid: 408093121470636540
status: 408
error class: java.util.concurrent.TimeoutException
```

Explanation of Event Message Composition

- **(ignored count: n)**: The number of transactions that met the conditions but did not trigger a notification during the pause period. This item is not displayed if there are no ignored transactions during the pause period.
- **Aggregation Information**: Displayed differently depending on the configured aggregation method.

  - **By Agent**: `[Per Agent] limit: n, count: n`
  - **By URL**: `[Per Url] limit: n`, followed by a list of each URL and its occurrence count
  - **By Error Class**: `[Per Error Class] limit: n`, followed by a list of each error class and its occurrence count
  - **By Context**: `[Per Context] limit: n`, followed by a list of each context and its occurrence count
  - **By Agent & URL**: `[Per Agent & Url] limit: n`, followed by a list of each URL and its occurrence count
  - **By Agent & Error Class**: `[Per Agent & Error Class] limit: n`, followed by a list of each error class and its occurrence count
  - **By Agent & Context**: `[Per Agent & Context] limit: n`, followed by a list of each context and its occurrence count
  - **Total**: `[Total] limit: n, count: n`
- **[Condition]**: Displays the conditions that were configured at the time the event was triggered. Only configured patterns are shown.
  - **context patterns**: Context match patterns
  - **url patterns**: URL match patterns
  - **error class patterns**: Error class match patterns
  - **error message patterns**: Error message match patterns
  - **status patterns**: Status code match patterns
  - **exclude context patterns**: Context exclusion patterns
  - **exclude url patterns**: URL exclusion patterns
  - **exclude error class patterns**: Error class exclusion patterns
  - **exclude error message patterns**: Error message exclusion patterns
  - **exclude status patterns**: Status code exclusion patterns
- **[Transaction]**: Detailed information about the most recent transaction that met the conditions.
  - **url**: Transaction URL
  - **txid**: Unique transaction identifier
  - **status**: HTTP response status code (not displayed for non-HTTP transactions)
  - **error class**: Error class (not displayed if no error class exists)
  - **error message**: Error message (not displayed if no error message exists)

Notes

- Transaction events are aggregated and processed in 1-minute intervals.
- If the condition type is **Pattern Only**, at least one match or exclusion pattern must be configured.
- Pattern matching is case-sensitive.
- Notification messages include detailed information about the most recent transaction that met the conditions.

Application Event Configuration (Classic)

Home > Select project > **Alert** > **Event Configuration** *classic*

You can configure alert notifications based on conditions such as **server resource usage (CPU, memory, disk)**, **number of active transactions**, **transaction errors**, and **increased response time**.

> The *classic* event configuration is a separated menu that allows continued use of the legacy event configuration method.
> For more advanced features, use Event configuration.

Default options

Inactive agent alert

An event occurs when no data is received from an agent for the configured period. You can enable or disable this feature using the toggle button.

- **Enabled**: An inactive agent event occurs when no data is received for a certain period.
- **Disabled**: No event occurs regardless of data reception status.

> The legacy inactive agent check interval can be configured under **Management** > **Agent list**.

Send notification when event is resolved

After an event occurs due to threshold conditions, you can choose whether to send a **Recovered** notification when the state returns to normal. Use the toggle button to enable or disable this feature.

- **Enabled**: Events operate in a state based manner. When thresholds are exceeded, the status becomes ongoing, and when values recover below thresholds, the status changes to recovered.
- **Disabled**: A one time event occurs each time the threshold is exceeded.

Duplicate agent alert

If two or more agents with the same name run simultaneously, normal monitoring may be affected, so an event is generated. However, duplicate agent events do not occur again within the configured time window.

- To check whether agent names are duplicated, review the related options in the **Event Configuration** menu.
- Available durations: `30 seconds` , `1 minute` , `3 minutes` , `5 minutes` , `10 minutes`

Conditions

You can configure event trigger conditions per item. Input formats vary depending on the item.

- **Server CPU / Server memory / Server disk**
- **Active transactions / Transaction errors**
- **Transaction response time**

Condition (%)

Set two threshold levels: Warning and Critical.
Alerts are triggered when thresholds are exceeded, with Critical taking priority.

- Left value: Warning threshold (Warning)
- Right value: Critical threshold (Critical)

Condition (count)

Enter the count used as the event trigger threshold. If the count exceeds the configured value for the specified duration, an event occurs.

Condition (count)

Configure the condition. An event is triggered when the number of transactions exceeding the configured response time reaches or exceeds the specified count.

Condition (response time)

Select the baseline response time.

- Available times: `1 second` , `2 seconds` , `3 seconds` , `5 seconds` , `10 seconds` , `15 seconds` , `30 seconds` , `1 minute` , `2 minutes` , `3 minutes` , `5 minutes` , `10 minutes` , `30 minutes` , `1 hour`

- Example: If the count threshold is **10**, the response time threshold is **1 second**, and the duration is **5 seconds**, then an event occurs when 10 or more transactions exceed 1 second and the condition persists for 5 seconds.

Duration

An event occurs when the configured condition persists for the selected duration.

- Available durations: `5 seconds` , `10 seconds` , `15 seconds` , `30 seconds` , `1 minute` , `2 minutes` , `3 minutes` , `5 minutes` , `10 minutes` , `30 minutes` , `1 hour`

Silence

After an event occurs, no additional events with the same condition occur during the selected silence period. Setting an appropriate silence duration helps prevent excessive alerts.

- Available durations: `1 minute` , `2 minutes` , `3 minutes` , `5 minutes` , `10 minutes` , `30 minutes` , `1 hour` , `2 hours` , `3 hours` , `6 hours` , `12 hours` , `1 day`

Event enablement

You can control event behavior using the toggle button.

- **Enabled**: An event occurs when the configured conditions are met.
- **Disabled**: Conditions are not evaluated.

Event configuration

You can configure events for the following items to monitor system status in real time. After configuring each item, click the **Save** button at the top right of the screen.

Server CPU / Server memory / Server disk

An event occurs when server resource usage exceeds the configured threshold (%). You can set different thresholds for Warning and Critical levels.

- Example: If CPU usage is set to Warning: 80% and Critical: 90%, a Critical event occurs when usage exceeds 90%.

Active transactions

Monitors the number of transactions currently being processed. An event occurs when the count exceeds the configured threshold and persists for the configured **duration**. This is useful for detecting situations with a large number of concurrent transactions.

- Default occurrence count: 100

Transaction errors

Transactions that do not complete normally are treated as errors. An event occurs when the error count exceeds the configured threshold and persists.

- Collection interval: 5 seconds

Transaction response time

Events are configured based on the time required for each transaction to complete (response time). You can detect service slowdowns relative to expected response time.

- Example: If the response time threshold is 10 seconds and the count threshold is 10
  “An event occurs when 10 or more transactions exceed 10 seconds and the condition persists for 5 seconds.”

Event Notification Settings

Home > Select Project > **Alert** > **Notifications**

Event Notification Settings per User

> Information about other members in **Event notification settings per-user** can only be viewed by members with the **Member Management** permission. For all other members, required information is masked.
>
> - Mobile device management for bulk event notification receipt settings and access control can be managed in **Account Information**.
> - For detailed information about member permissions, refer to the following document.

Notification Methods

Alerts are available via **Email**, **SMS**, **WhatsApp**, and **Mobile**.

In the **Event notification settings per-user** section, select the checkbox for the desired notification method to receive alerts. If you clear the checkbox for a notification method, alerts will not be sent through that channel.

> All users except the project super administrator can change only their own notification settings.

Email Alerts

Email alerts are sent to the email address provided during sign-up.

SMS Alerts

You can receive alerts via SMS.

> The SMS alert feature is available only for paid projects.

1. Click your **profile icon** at the top right of the screen, then go to **Manage Accounts**.
2. In **Account Information**, click the [General Mobile Phone] button in the **User Phone Number** section.
3. Enter the **phone number** that will receive the verification code.
    - Only **Korean mobile phone numbers** can be registered to receive SMS alerts.
4. Click the [Save] button.

> **Change or delete the registered phone number**
>
> - To change the phone number, click the [Change] button. SMS verification is required.
> - To delete the phone number, click the [Delete] button. After deletion, you will no longer receive alerts via SMS.

WhatsApp Alerts

You can receive alerts via WhatsApp.

> The WhatsApp alert feature is available only for paid projects.

1. Click your **profile icon** at the top right of the screen and go to **Manage Accounts**.
2. In **Account Information**, click the [WhatsApp] button in the **User Phone Number** section.
3. Enter the **phone number** that will receive the verification code.
4. Click the [Send Verification Code] button.
5. Enter the 6-digit verification code sent to your WhatsApp application.
    - If you do not receive the verification code within 10 minutes, click the [Resend Verification Code] button.
6. Click the [Verify] button.

Mobile Alerts

Users who need to receive mobile alerts must install the WhaTap application on their mobile device and log in.

- Download the WhaTap application – **Android**
- Download the WhaTap application – **iOS**

Repeated Alerts (Escalation) Settings

If the alert condition is not resolved after an alert is triggered, you can configure repeated alert intervals based on the time of the initial alert. This setting applies only to **Critical** level alerts.

For example, to repeat alerts immediately (0 minutes), 1 hour later, and 1 day later after the initial alert, enter `0,1H,1D` in the **Repeated Alerts (Escalation)** column.

1. Go to **Alert** > **Notifications** and navigate to the **Event notification settings per-user** section.
2. Configure the settings in the **Repeated Alerts (Escalation)** field.
   - **M**: minutes, **H**: hours, **D**: days. If the unit is omitted, the value is interpreted as minutes.
   - Enter a number or a number with a unit (**M**, **H**, **D**). If the input format is invalid, an error message is displayed.
3. Click the [Save] button. The settings will not be applied unless they are saved.

Event Reception Tags

When configuring events, you can select **event reception tags** to send alerts to project members and 3rd-party plugins associated with those tags. If no event reception tag is set, alert notifications are sent to **all project members**.

Add / Edit / Delete Event Reception Tags

> Event reception tags that are **currently applied to an event** cannot be deleted.

1. Go to **Alert** > **Notifications**.

2. In the user list under the **Event notification settings per-user** section, click **+ Add Tag** or **+**.
3. In the **Event reception tag** window, click **+ Create New Tag**.
4. In the **Create Tag** dialog, enter a tag name, select a color, and click the [Create Tag] button.
5. To edit or delete a created tag, click the icon in the tag list.
6. Select the desired tag from the **Tag List** in the **Event reception tag** window to apply it.

Remove Event Reception Tags

1. Go to **Alert** > **Notifications** and open the **Event notification settings per-user** section.
2. Click **+** in the **Event reception tag** field for the user whose event reception tag you want to remove.
3. In the **Event reception tag** window, click the **X** on the tag you want to remove. The tag will be removed.

Receive Alerts Without Reception Tags

When configuring events, you can select event reception tags to send alerts to project members and 3rd-party plugins associated with those tags.

- To receive alerts **without event reception tags**, enable the **Receive alerts without reception tags** option.
- If you want to receive **only alerts that have event reception tags**, disable this option.

> To stop receiving all alerts, disable this option and make sure that no **Event reception tag** is selected.

Selecting the event reception mode

In the **Event notification settings per-user** section, you can select the event alert reception scope per member or third-party channel from the following three modes. The reception mode is selected in the **Event reception tags** column of each member or channel row.

Mode	Description
Receive all alerts	Receive all alerts regardless of the event reception tags
Receive alerts for selected tags	Receive only alerts that have the selected event reception tags. Select at least

Mode	Description
only	one tag
Receive no alerts	Receive no event alerts. The selected tags are preserved without being deleted

- In the **Receive alerts for selected tags only** mode, select at least one tag to receive alerts in the tag selection area.
- To also receive alerts that have no event reception tag specified, select the **Unclassified** virtual tag in the tag selection area. The **Unclassified** tag can be selected or deselected like a normal tag, but the tag itself cannot be edited or deleted.

> ⓘ In the **Receive no alerts** mode, the corresponding member or channel receives no event alerts at all. The selected tags are preserved without being deleted, so if you switch back to the **Receive alerts for selected tags only** mode, you can use the previously selected tag settings as is.

Select Notification Language

You can choose the language in which you receive alert notifications.

1. Go to **Alert** > **Notifications**.
2. In the user list under the **Event notification settings per-user** section, click the ▶ button at the far left of the user whose language you want to configure.
3. Select the **Notification Language**.

Day and Time-based Notification Settings

You can configure whether a user receives notifications by day of the week and time of day.

1. Go to **Alert** > **Notifications**.
2. In the user list under the **Event notification settings per-user** section, click the ▶ button at the far left of the user whose day and time settings you want to configure.

3. Select the days and/or enter the time ranges during which you want to receive alerts.
    - Settings can be configured separately for each notification method.

3rd-party Plugins

You can receive alert notifications through external applications.

- **Supported applications**: Slack, Telegram, Teams, Teams Workflows, Jandi, AlertNow, Webhook, Webhook JSON, Opsgenie, PagerDuty, LINE, ilert

> ⓘ Internal messengers that are not included in WhaTap Labs' supported scope can be integrated via standard **Webhook** or **Webhook JSON**.

Table | 3rd-party Plugin List

Item	Description
Plugin Name	Name of the plugin
Auth Key	Authentication key
Auth Value	Authentication value
Receive Level	Notification level: Warning, All, or Critical
Repeated Alerts (Escalation)	Re-notification interval from the initial alert time if the issue is not resolved - Applies only to Critical alerts
Smart Message (AI)	AI-powered notifications available when adding the Telegram plugin - Supported languages: Korean, English, Japanese
Event Reception Tag	Whether to receive alerts for specific members or teams in the project
Delete	Delete the 3rd-party plugin

Add a 3rd-party Plugin

1. Go to **Alert** > **Notifications**.
2. In the **3rd-party Plugins** section, click **Add**.
3. In the **Add 3rd-party Plugin** window, select the desired service tab.
4. Follow the instructions for the selected service (see How to add 3rd-party plugin notifications).
5. Click **Test Receive** to verify the configuration.
6. When finished, click **Add** or **Register**.

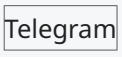
How to Add 3rd-party Plugin Notifications

Slack

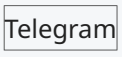
In the **Add 3rd-party Plugin** window, click the [**Add To Slack**] button on the **Slack** tab to register **WhaTapNotiBot** to your Slack team and channel.

Telegram

1. In **Telegram**, search for **@WhaTapApmNotiBot**.

2. Select the bot and click the [Start] button.
3. In the chat window, enter the `/license` command followed by the access key of the project for which you want to receive alerts.

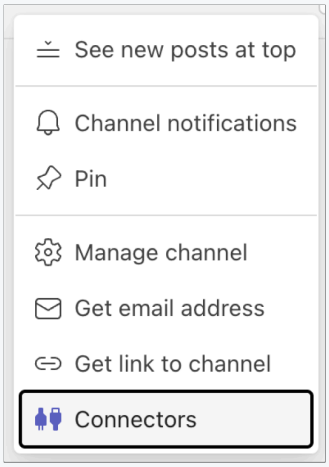
Available commands

- `/help` : View the list of commands
- `/license` : Enter the access key of the project to receive alerts

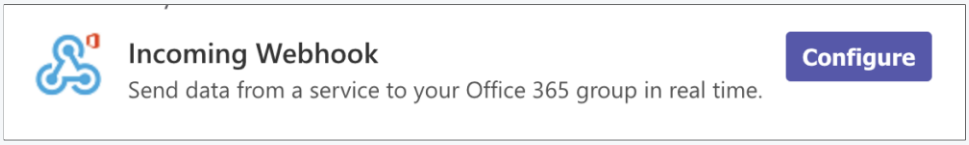
- `/remove` : Enter the project code for which you want to stop receiving alerts
- `/list` : View the list of project codes currently receiving alerts

Teams

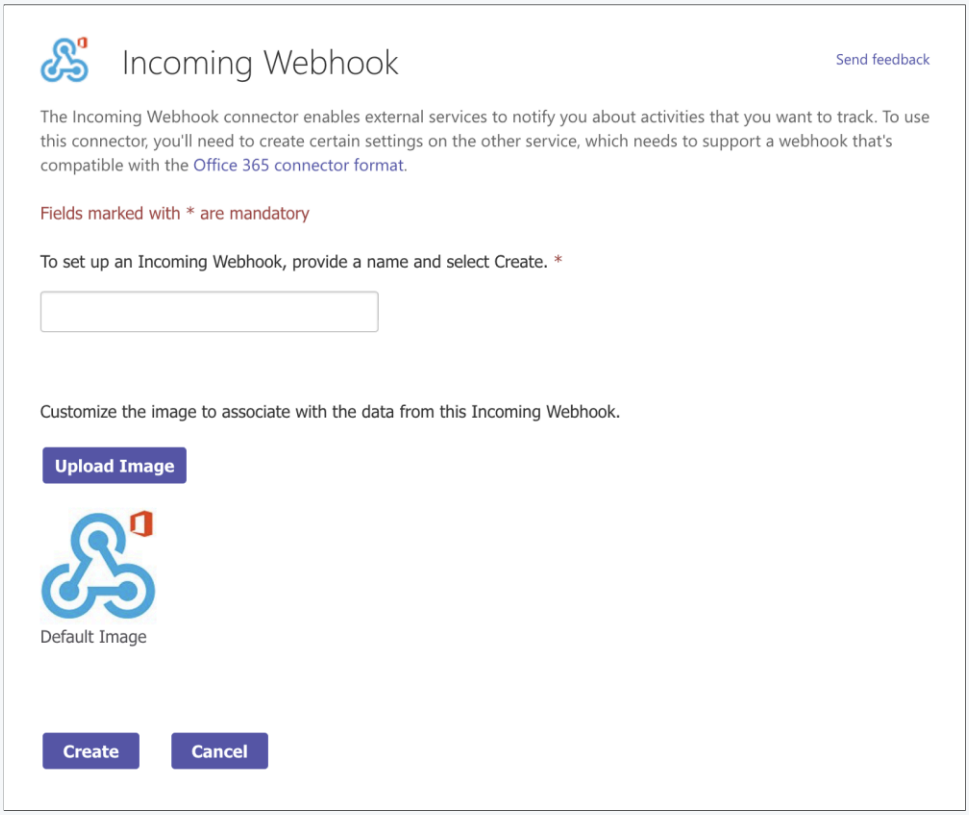
1. Go to the **connector** of the channel where you want to receive alerts.



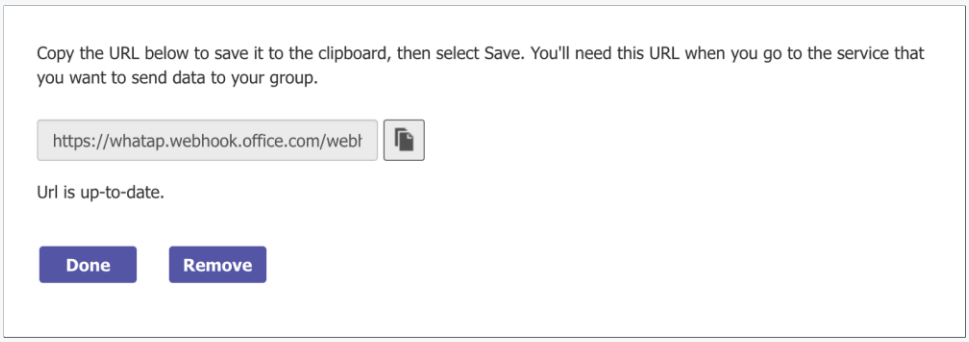
2. Add a **Webhook** and click the [Configure] button.



3. Enter a webhook name and click the [Create] button.



4. Copy the generated **Webhook URL**.

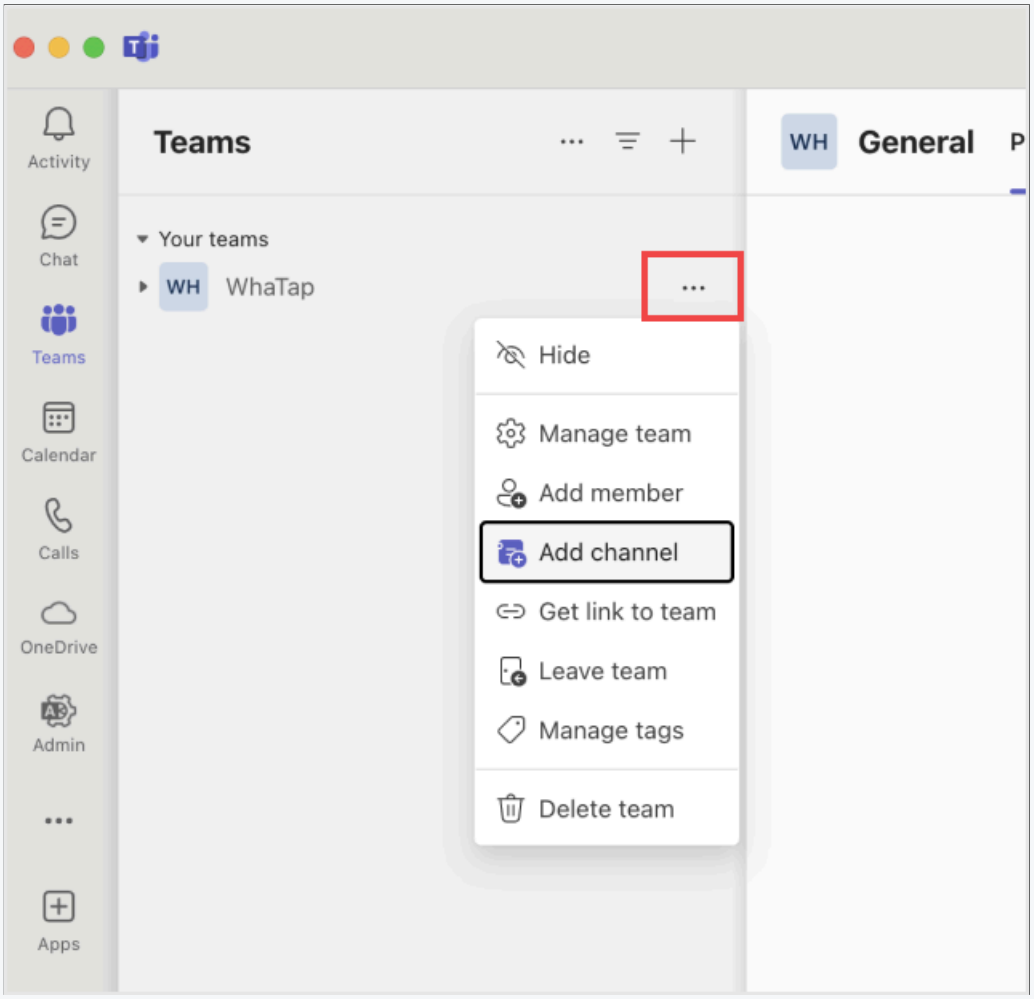


5. Return to the **Add a third-party plug-in** window in the WhaTap Monitoring service, and enter values for **Channel Name** and **Webhook URL** at the bottom of the **Teams** tab.
6. Click the [**Registration**] button.

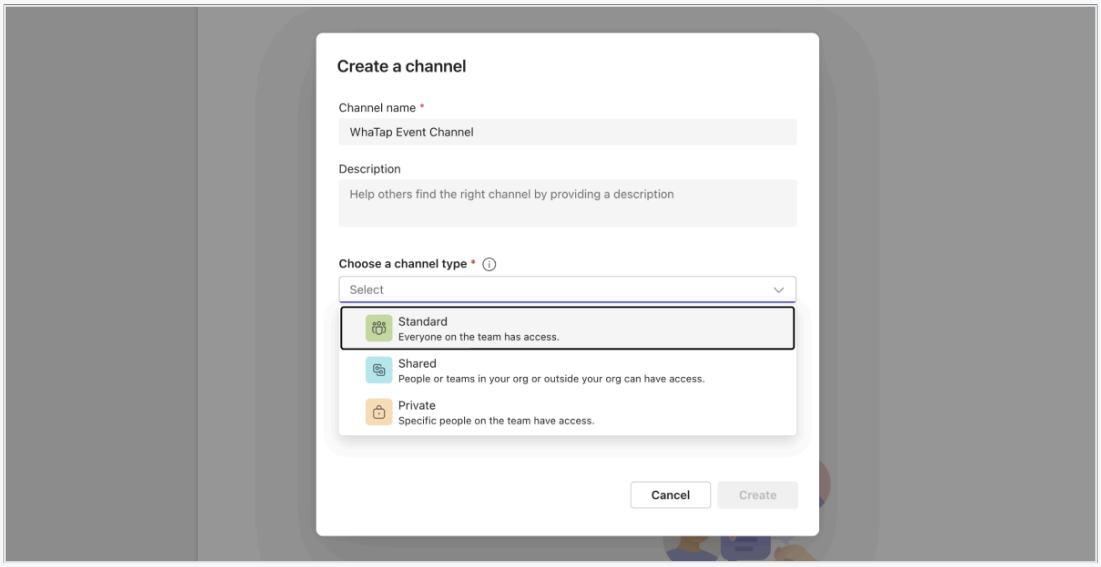
Teams Workflows

To use workflows, you need an account that can access **Microsoft Teams** and a **Business plan subscription**.

1. In the **Teams** menu, click **More actions (...)** > **Add channel**.

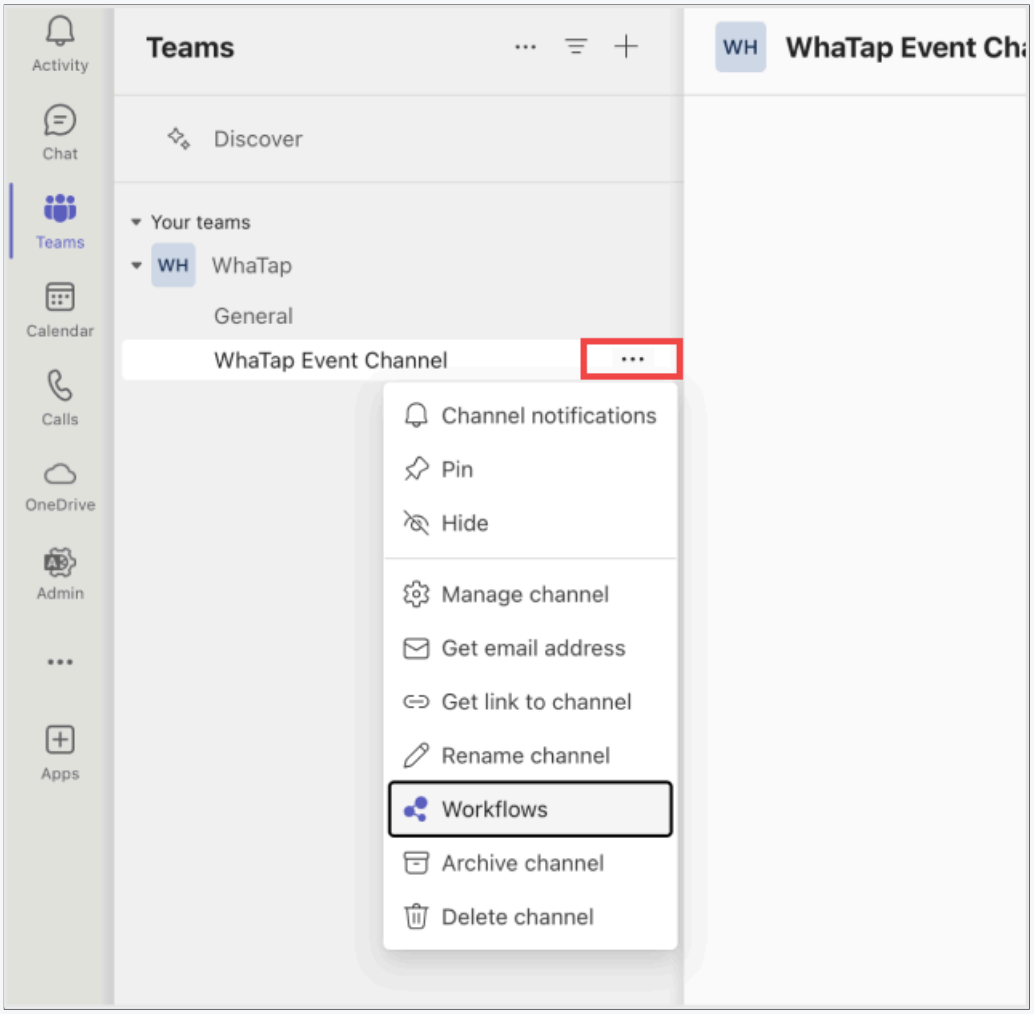


2. In the **Create a channel** window, enter the **Channel name** and **Description**. For **Channel type**, select **Standard**, then click the [Create] button.

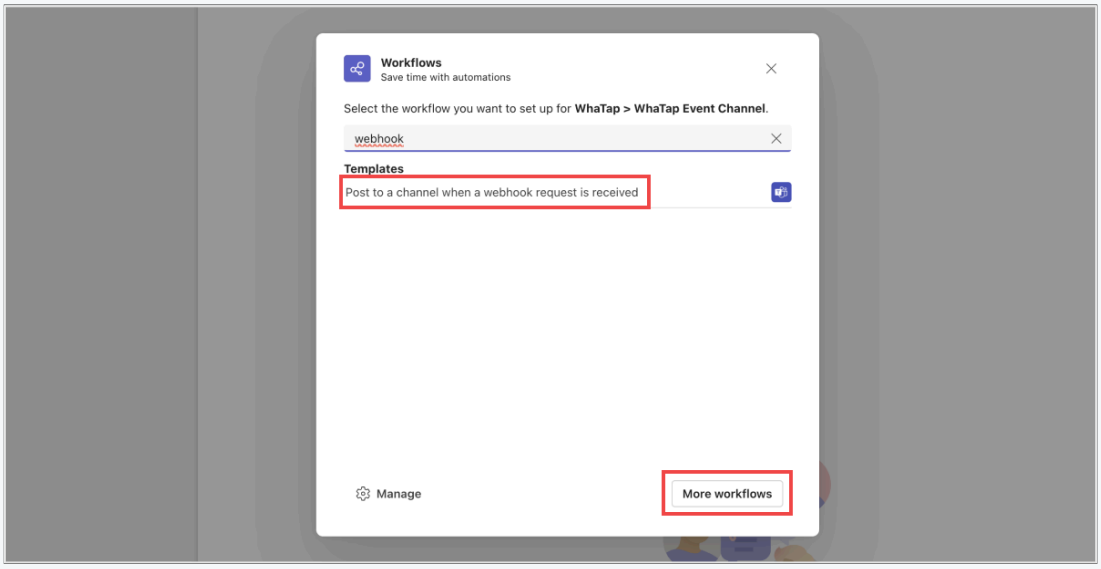


If you select **Shared** as the channel type, workflows cannot be used. If you select **Private**, workflows can be added but notifications cannot be received.

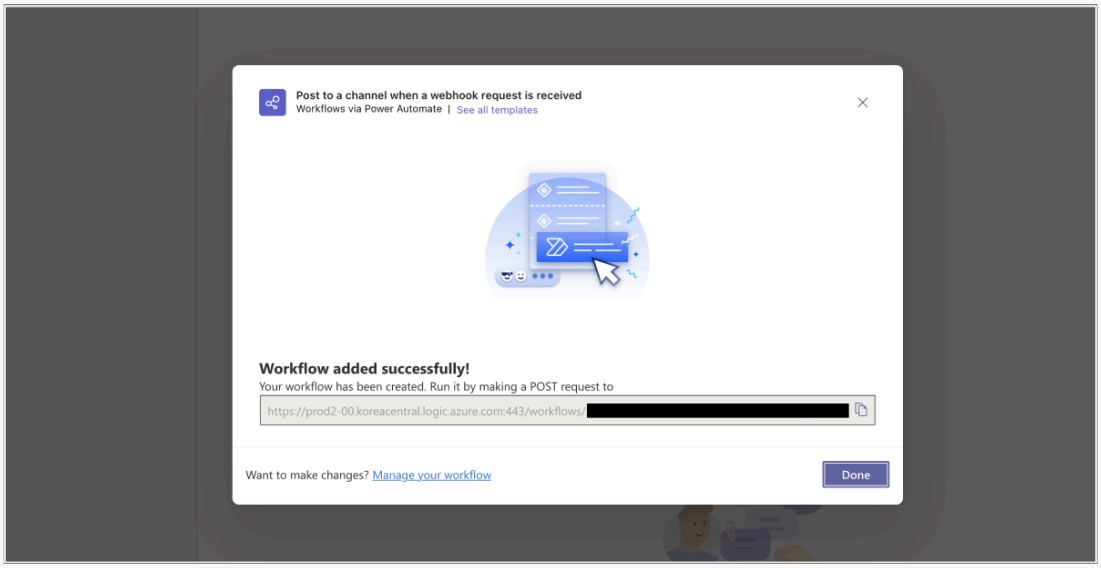
3. Click **More actions (...)** of the created channel, then select **Workflows**.



4. In the **Workflows** dialog, select **Post to a channel when a webhook request is received** from the workflow list, then click the [Add workflow] button.



5. Check the **Name** and **Connection** status, then click the [Next] button.
6. Review the **Details**, then click the [Add workflow] button.
7. The workflow has been added. Copy the generated URL.

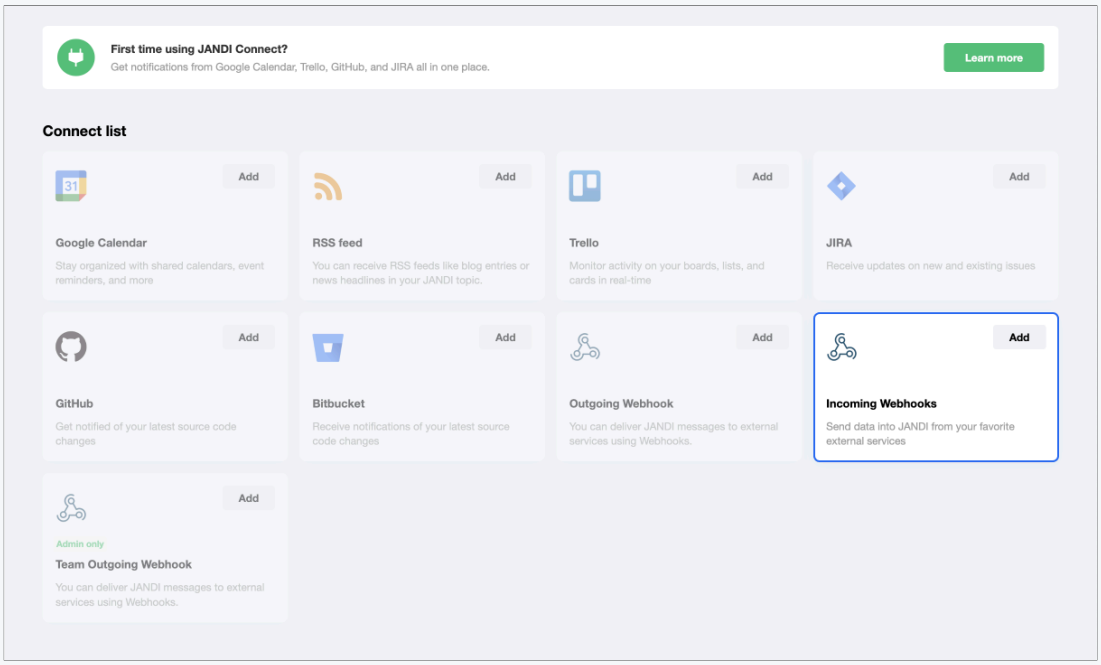


8. Return to the **Add a third-party plug-in** window in the WhaTap Monitoring service, and enter values for **Channel Name** and **Webhook URL** at the bottom of the **Teams Workflows** tab.

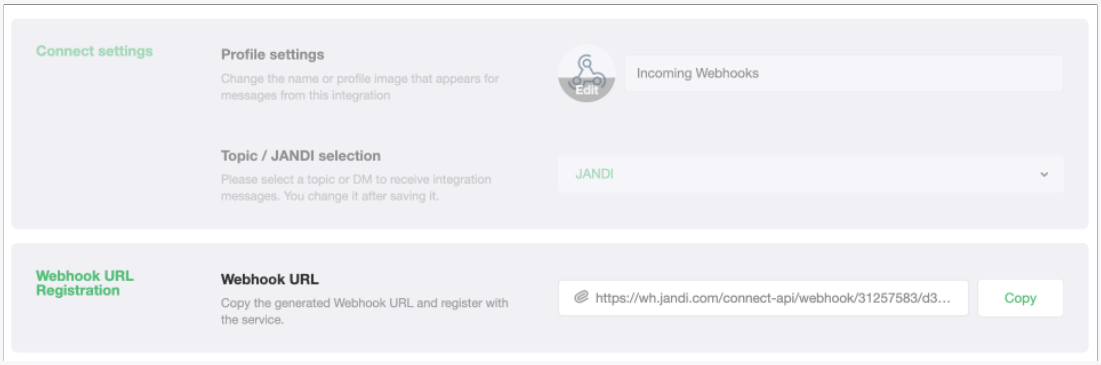
9. Click the [**Registration**] button.

Jandi

1. Go to **Jandi Connect settings** in the topic or chat where you want to receive notifications.
2. In the **Connect list**, find **Webhook Receiver (Incoming Webhook)** and click the [Connect] button.



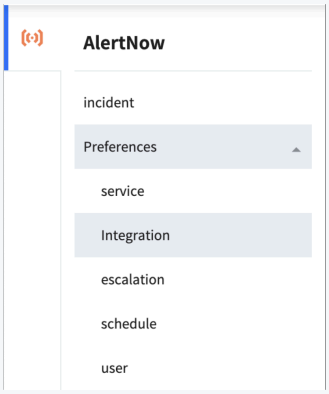
3. Click the [Add Webhook Receiver] button.
4. Copy the Webhook URL from the **Register Webhook URL** section.



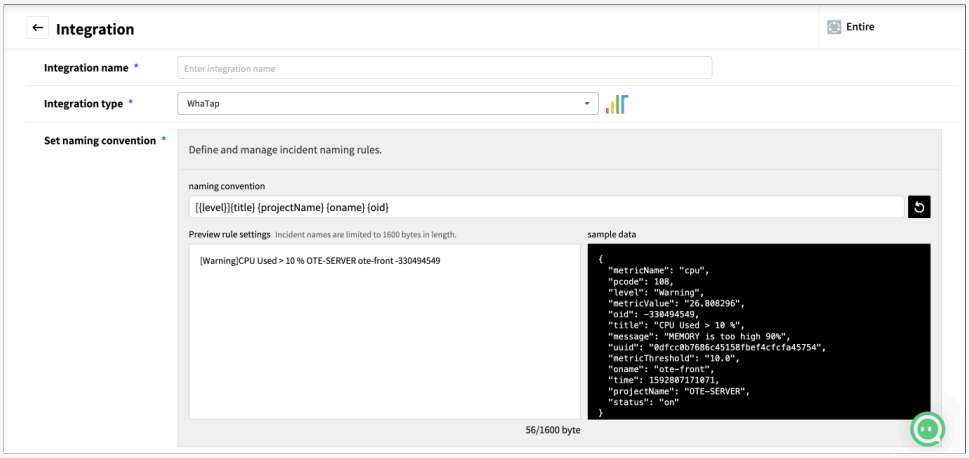
5. Return to the **Add a third-party plug-in** window in the WhaTap Monitoring service, and enter values for **Channel Name** and **Webhook URL** at the bottom of the **Jandi** tab.
6. Click the [**Registration**] button.

AlertNow

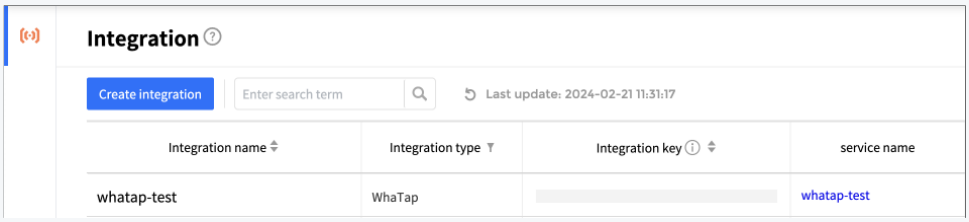
1. Access the **AlertNow** service, then go to **Settings** > **Integration**.



2. On the **Integration** screen, click the [Create Integration] button.
3. From the list of integration types, select **WhaTap**.
4. On the **Create Integration** screen, configure the **Name** and **Naming Rules**.



5. Click the [Confirm] button to create the integration.
6. Check the list of created integrations and select the newly created integration.



7. On the details screen, click the copy icon on the right side of the URL to copy it.



8. Return to the **Add a third-party plug-in** window in the WhaTap Monitoring service, and enter values for **Channel Name** and **Webhook URL** at the bottom of the **AlertNow** tab.
9. Click the [**Registration**] button.

Webhook

1. Register the Webhook that will receive notifications. The alert `title` and `text` are sent to the registered URL via the **POST** method.

```
{
  "title":"Alert Title",
  "text": "Sample warning message"
}
```

2. Enter values for **Webhook Name** and **Webhook URL**.
3. Click the [**Registration**] button.

Webhook JSON

1. Register the Webhook that will receive notifications. Alert data is sent to the registered URL in **JSON format** via the **POST** method. For substitution variables that can be included in the JSON payload, see **Available Variables**.
   - `${something}` represents a substitution value provided by WhaTap. Refer to **Available Variables** for the full list and descriptions.
   - The event status (`status`) is categorized as **event triggered** (`on`) or **event recovered** (`off`).
   - The `level` is categorized as `None` , `Info` , `Warning` , or `Critical` .

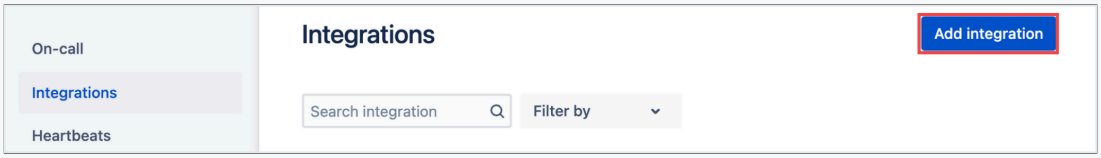
```
{
  "pcode": "${pcode}",
  "projectName": "${projectName}",
  "time": "${time}",
  "oid": "${oid}",
  "oname": "${oname}",
  "title": "${title}",
  "message": "${message}",
  "level": "${level}",
  "status": "${status}",
```

```
  "metricName": "${metricName}",
  "metricValue": "${metricValue}",
  "metricThreshold": "${metricThreshold}",
  "uuid": "${uuid}",
  "okind": "${okind}",
  "onode": "${onode}"
}
```

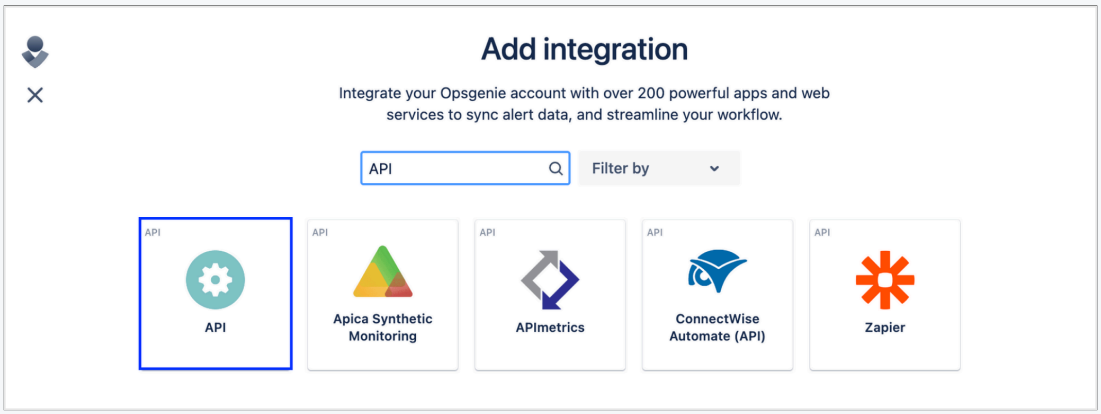
2. Enter values for the **Webhook Name** and **Webhook URL** fields.
3. To add a header value, click **+ Add** in **Add user header value**, then enter the key and value.
4. Click the [**Registration**] button.

Opsgenie

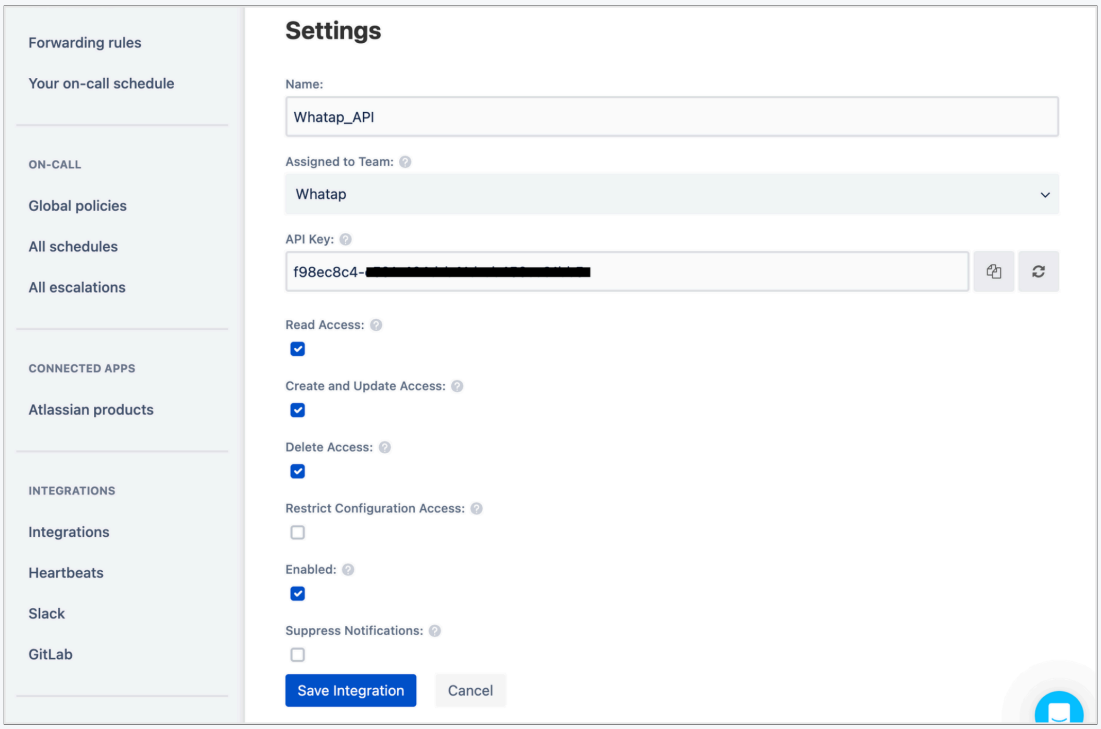
1. In the **Teams** menu, select the team that will receive notifications.
2. Go to the **Integration** menu and click the [Add Integration] button.



3. In the integration list, search for and select **API**.



4. Set the API name, assign a team, configure the required permissions, and then click the [Save Integration] button.



5. Return to the **Add a third-party plug-in** window in the WhaTap monitoring service, and enter values for **Name** and **API Key** at the bottom of the **Opsgenie** tab.
6. Click the [**Registration**] button.

PagerDuty

1. Sign up for a PagerDuty account and log in.
2. In PagerDuty, go to **Services** > **Service Directory**, then click the [+ New Service] button.

   PagerDuty
3. Enter the service information (**Name**) and click the [Next] button.

   PagerDuty
4. Configure the **Escalation Policy**, then click the [Next] button.

PagerDuty

5. Configure **Alert Grouping** and **Transient Alerts**, then click the [Next] button.

   PagerDuty

6. In **Integrations**, select **Event API V2**, then click the [Create Service] button.

   PagerDuty

7. On the created service page, copy the **Integration Name** and **Integration Key** values.

   PagerDuty

8. Return to the **Add a third-party plug-in** screen in the WhaTap monitoring service and enter the **Integration Name** and **Integration Key** values.
9. Click the [Register] button.

> If you do not receive messages or the plugin is not registered, check whether the **Integration Name** or **Integration Key** values are correct.

LINE

1. Install the LINE app.
2. Scan the QR code below to add the official **WhaTapLabs** channel as a friend.

   Add LINE 3rd-party plugin

3. Use the following commands to link your project with LINE and configure notifications.

   Commands

```
/help # Show available commands
/list # View the list of linked projects
/license [ACCESS_KEY] # Add a new project linkage
/remove [PROJECT_CODE] # Remove a linked project
/language [ko|ja|en] # Set notification language (Korean/Japanese/English)
```

- You can find the project access key under **Management > Project Management**.
- Using the `/language` command in LINE applies **only to the current LINE chat room**. It does not change the notification language for other 3rd-party plugins or projects.

Examples

```
/license XXXXXXXXXX # Link a project
/language ko # Set language to Korean
/list # Check linked projects
```

- **Project access key**: Available under **Management > Project Management** in the WhaTap project.
- **Language settings**: The `/language` command applies only within the LINE chat room and does not affect other notification channels.

ilert

1. In the ilert console, click [+ Create alert source] at the top right of the **Alert sources** page.
2. In the **Integration type** step, enter `WhaTap` in the **Filter integrations** search field at the bottom to search.
3. From the search results, select the **WhaTap** tile and click the [Next] button.
4. Enter the alert source name and assign a team, then click the [Next] button.
   - You can create a new escalation policy or select an existing one.
5. Select the alert grouping method and click the [Continue setup] button.
   - The initially selected **Do not group alerts** option can be changed later.
6. On the next screen, you can configure additional settings such as **alert templates and priorities**. For now, click the [Finish setup] button to complete the setup.
7. When the setup is complete, an **API key** or **Webhook URL** will be generated on the completion screen.

8. Copy the generated **Webhook URL** from the **Webhook JSON** tab and paste it into the **Webhook URL** input field below.
9. Click the [Register] button.

Preventing Alert Storms

If a large number of alerts occur within a short period, alert notifications are temporarily paused for a specified duration. For example, if 20 events occur within 5 minutes, alert notifications will be suspended for the next 5 minutes. Once the configured **Pause Duration** has elapsed, the alert-storm prevention feature is automatically released.

1. Go to **Alert** > **Notifications** > the **Avoids a large number of alerts** section.
2. Click the toggle button to **Enable** or **Disable** the feature.
   - **Enabled**: Temporarily pause alerts when an alert storm is detected
   - **Disabled**: Resume alert notifications
3. Configure the **Detection Time**, **Detection Count**, and **Pause Duration**.
   - If the number of events exceeds the **Detection Count** within the **Detection Time**, alert notifications are paused for the configured **Pause Duration**.

When alerts are blocked due to an alert storm, a notification banner is displayed. To immediately release the alert-storm block, click the **Stop Alert Storm Prevention** button in the banner.

Event Notification Format

Common Alert Template

The common template can be applied in most cases. It allows you to receive alerts in a consistent format across various environments.

- Product Type: Application, Database, Kubernetes
- Event Type: Application Alerts, Database Alerts, Metrics Alerts
- Event Channel: SMS, Mobile, 3rd-party plugins, Plugins

Event Title

Event title format

```
[Level][Platform][ProjectName][ApplicationName][EventTitle]
```

Event title example

```
[Info][JAVA][Application Project][TC-0-1-8081][CRITICAL_HIGH_MEMORY]
```

> If ApplicationName is not set, it is omitted from the event title.
>
> - Platform is displayed as one of the following.
>   - 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

Event Message

This section describes the information that can be included in the event message. If `optional` is set to `false`, the item is always included in the message. If `optional` is set to `true`, the item is displayed only when the data is available.

Event message example

```
Project Name : Application Project
Project Code : 3
Agent Name : TC-0-1-8081
Message : RECOVERED: Memory is too high. less than 10%
```

```
Event 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
```

Table | Event Message Components

En	Ko	Supported Alert Types	Description
Project Name	Project name	All	-
Project Code	Project code	All	-
Agent Name	Agent name	All (optional)	oname
Message	Event message	All	-
Alert Type	Alert type	All	See the AlertType table below
Event Time	Event occurrence time	All	In `2022-04-13 10:40:49 +0900`, `+0900` indicates the GMT offset
Event Off Time	Event recovery time	All (optional)	In `2022-04-13 10:40:49 +0900`, `+0900` indicates the GMT offset
Metric Name	Metric name	All (optional)	Name of the metric used to evaluate the event condition
Metric Value	Metric value	All (optional)	Included when the metric value exceeds the threshold
Metric Threshold	Metric threshold	All	Threshold value used to determine the event

En	Ko	Supported Alert Types	Description
		(optional)	condition
Stateful	Recovered-event notification	All (optional)	`true` if recovered-event notification is enabled; otherwise `false`
Event Rule	Event trigger condition	Metrics alerts	-
Target Filter	Event target filter	Metrics alerts	Evaluates event conditions only for metrics collected from specific targets
Repeat Count	Event repeat count	Metrics alerts	The event is triggered when the condition is met the specified number of times within the repeat duration
Repeat Duration	Event repeat duration	Metrics alerts	Time window used to evaluate the repeat count
Receiver	Receiver	Metrics alerts	-
Query	MXQL query	Composite metrics alerts	-
Rule	Event trigger condition	Composite metrics alerts	-
Query Period	Query period	Composite metrics alerts	-
Query Interval	Query interval	Composite	-

En	Ko	Supported Alert Types	Description
		metrics alerts	
Silent Time	Silent time	Composite metrics alerts	-
Query	URL	Exception alerts	URL of the request that caused the exception
TXID	Transaction ID	Exception alerts	-
Class	Error class name	Exception alerts	-
Log Message	Log message	Server – file log alerts	-
Log File	Log file path	Server – file log alerts	-
IP	IP address	All server alerts	-
CPU	CPU usage	All server alerts	Snapshot at the time the event occurred
CPU_load1	CPU load (1 min)	All server alerts	Snapshot at the time the event occurred
CPU_loadPerCore	CPU load per core	All server	Snapshot at the time the event occurred

En	Ko	Supported Alert Types	Description
		alerts	
Memory	Memory usage	All server alerts	Snapshot at the time the event occurred
Swap	Swap usage	All server alerts	Snapshot at the time the event occurred
Disk Name | Used Percent | Free Size | IO Percent	Disk performance	All server alerts	Snapshot at the time the event occurred
Name | Bps | Pps	Network traffic performance	All server alerts	Snapshot at the time the event occurred
Message | Time | Name	Acknowledgement information	All server alerts	-

This section displays as much information as possible that can be provided for the event.

- **AlertType** is displayed as one of the following values.

AlertType	Description
APPLICATION_CPU	Application CPU alert
APPLICATION_MEMORY	Application memory alert
APPLICATION_DISK	Application disk alert
APPLICATION_ACTIVE_TRANSACTION	Application active transaction alert
APPLICATION_ERROR_TRANSACTION	Application error transaction alert

AlertType	Description
APPLICATION_SLOW_TRANSACTION	Application slow transaction (response time) alert
METRICS	Metrics alert
COMPOSITE_METRICS	Composite metrics alert
ANOMALY	Anomaly detection alert
LOG_REALTIME	Real-time log alert
COMPOSITE_LOG	Composite log alert
SERVER_REBOOT	Server reboot alert
SERVER_NO_DATA	Server no-data alert
SERVER_PORT	Server port alert
SERVER_NETWORK_IOPS	Server network IOPS alert
SERVER_NETWORK_BPS	Server network BPS alert
SERVER_DISK_IO	Server disk I/O alert
SERVER_DISK_QUOTA	Server disk usage alert
SERVER_DISK_INODE	Server inode alert
SERVER_CPU	Server CPU alert
SERVER_MEMORY	Server memory alert
SERVER_CPU_STEAL	Server CPU steal alert

AlertType	Description
SERVER_MEMORY_SWAP	Server swap memory alert
SERVER_LOG_FILE	Server log file alert
SERVER_WINDOW_EVENT	Server Windows event alert
SERVER_OFF	Server alert OFF notification
SERVER_ACKNOWLEDGE	Server acknowledge notification
SERVER_PROCESS_COUNT	Server process count alert
SERVER_PROCESS_CPU	Server process CPU alert
SERVER_PROCESS_MEMORY	Server process memory alert
SERVER_PROCESS_OFF	Server process alert OFF notification
AGENT_ACTIVE	Agent activated alert
AGENT_INACTIVE	Agent deactivated alert
AGENT_REACTIVATED	Agent reactivated alert
URL	URL alert
TOO_MANY_EVENT	Too many events alert
CLOUD_WATCH	CloudWatch alert
EXCEPTION	Exception alert

Application Alert Notifications

Application alerts provide both an Event Title and an Event Message. When using the **Additional notifications when the event state is resolved.** feature, an Event Off Message is sent when the event trigger condition is cleared.

Event Type	Event Level	Event Title	Event Message	Event Off Message
Server CPU	Warning	HIGH_CPU	CPU is high. `${value}% (>= ${threshold}%)`	RECOVERED: CPU is high. less than `${threshold}%`
Server CPU	Critical	CRITICAL_HIGH_CPU	CPU is too high. `${value}% (>= ${threshold}%)`	RECOVERED: CPU is too high. less than `${threshold}%`
Server Memory	Warning	HIGH_MEMORY	Memory is high. `${value}% (>= ${threshold}%)`	RECOVERED: Memory is high. less than `${threshold}%`
Server Memory	Critical	CRITICAL_HIGH_MEMORY	Memory is too high `${value}% (>= ${threshold}%)`	RECOVERED: Memory is too high. less than `${threshold}%`
Server Disk	Warning	HIGH_DISK	Disk is high `${value}% (>= ${threshold}%)`	RECOVERED: Disk id high. less than `${threshold}%`
Server Disk	Critical	CRITICAL_HIGH_DISK	Disk is too high `${value}% (>= ${threshold}%)`	RECOVERED: Disk is too highf. less than `${threshold}%`
Normal Transactions	Warning	HIGH_ACTIVE_TRANSACTION	Active Transaction Count is over `${value} (>= ${threshold})`	RECOVERED: Active Transaction Count is less than `${threshold}`
Error	Warning	HIGH_ERROR_TRANSACTION	Error Transaction Count	RECOVERED: Error

Event Type	Event Level	Event Title	Event Message	Event Off Message
Transactions			is over ${threshold} (${value})	Transaction Count is less than ${threshold}
Slow Transactions	Warning	TOO_MANY_SLOW_TX	Too many delayed transactions (${value}, above ${time} ms)	RECOVERED: Too many delayed transactions. less than ${threshold}

Event History And Monitoring

Home > Select Project > **Alert** > **Event History**

Event History allows you to view events that have occurred in the project and analyze their detailed information. It supports Elasticsearch DSL queries, enabling you to find events that match specific conditions quickly and accurately.

Viewing Event History

Click the **event title** in the search results to move to a screen where you can analyze detailed data for the time period when the event occurred. This allows you to comprehensively understand the situation before and after the event.

1. Go to **Alert** > **Event History**.
2. In the **Event History** screen, use **Time Selection** to set the period for viewing event records.
3. Enter a query in the filter input field and click the [Magnifying Glass] button.
    - If no query is entered, all events are displayed.

Column Settings

You can choose which columns to display or hide.

1. In the **Event History** screen, click the [Column Settings] button at the top left.
2. In the **Column Settings** window, select the columns you want to show or hide.
    - You can change the column order in **Displayed Items**.
    - You can search for specific columns using the **Column Name Search** field.

Column Item	Description
Event Name	The title of the event

Column Item	Description
Message	The event message or snapshot information
Event Status	The current status of the event
Agent Name	The name (Oname) of the agent where the event occurred - Displayed as blank if the event is not associated with a specific agent
Agent Group	The agent type (OkindName) where the event occurred - Displayed as blank if the event is not associated with a specific agent
Agent Node	The server (OnodeName) of the agent where the event occurred - Displayed as blank if the event is not associated with a specific agent
Event Occurred Time	The time when the event occurred
Event Resolved Time	The time when the event was resolved - Displayed as `-` for events without a resolved state

3. After completing the settings, click the [Confirm] button to apply the column configuration.

CSV Download

You can download the query results as a CSV file.

1. Go to **Alert** > **Event History** and click the [CSV] button.
2. The list of events that match the current filter conditions is saved as a CSV file.

View Only Events in Progress

You can view only events that are currently in progress.

1. Go to **Alert** > **Event History** and click the [View Only Events in Progress] button.
2. The following condition is automatically added to the filter query and applied to the search.

```
Stateful: true and Status : "ON"
```

Event Search

- **Precise Search**: Specify exact conditions to retrieve only the events you need
- **Composite Conditions**: Search across multiple fields at once to handle complex conditions in a single query
- **Flexible Patterns**: Supports various search methods such as wildcards and partial matches
- **High Performance**: Fast searches powered by indexed data

Searchable Fields

Field names and values are not case sensitive.

- **Field Name Tag**: Event types that support the field
- **Field Type**: The data type of the field
  - #: Number
  - T: String
  - : Boolean
  - : Date

Event Information Fields

Field Name	Type	Description	Example Value
Title	String	Event title	"Database Connection Error"
OffTitle	String	Event recovery title	"RECOVERED: Database Connection Error"
Message	String	Event message	"Connection timeout occurred"
OffMessage	String	Event recovery message	"RECOVERED: Connection timeout occurred"

Field Name	Type	Description	Example Value
Level	String	Current event level	`Critical`, `Warning`, `Info`
OriginLevel	String	Original event level	`Critical`, `Warning`, `Info`
Status	String	Event status	`ON`, `OFF`, `CANCEL`, `ACKNOWLEDGE`, `MAINTENANCE`, `DISABLED`
ActCount	Number	Number of occurrences	2
MetricName	String	Metric name	"memory"
MetricValue	String	Metric value	"85.5"
OffValue	String	Recovery value	"72.8"
MetricThreshold	String	Threshold value	"80"
alertType	String	Event type	"BASIC", "METRICS", "TRANSACTION", etc.
alertId	String	Unique identifier of the event rule	"zf3uojer0fv4v7"

Type-specific Supported Fields

> ⓘ These fields are supported only for specific event types. If the event type does not match, the value may be empty.

Field Name	Type	Event Type	Description	Example Value
eventRule	String	Basic, Metrics	Event trigger rule	"memory ≥ 80"
field	String	Real-	Log search	"content"

Field Name	Type	Event Type	Description	Example Value
		time Log	key	
keyword	String	Real-time Log	Log search value	"Error"
logCategory	String	Real-time Log	Log category	"AppLog"
logContent	String	Real-time Log	Log content	"00:00000:00009:2025/04/06 23:03:52.55 server Error: 1601, Severity: 17, State: 3\n00:00000:00009:2025/04/06 23:03:52.55 server There are not enough 'user connections' available to start a new process."

Target Fields

Field Name	Type	Description	Example Value
Oid	Number	Agent unique identifier	-98765432
Oname	String	Agent name	"web-server-01"
Okind	Number	Agent type unique identifier	867318026
OkindName	String	Agent type name	"web-server"
Onode	Number	Agent node unique identifier	334634079
OnodeName	String	Agent node name	"production-node-1"

Unique Event Fields

Field Name	Type	Description	Example Value
Id	Number	Unique identifier of the event	5768121
UUID	String	Unique event identifier	"550e8400-e29b-41d4-a716"

State / Flag Fields

Field Name	Type	Description	Example Value
Stateful	Boolean	Whether the event is state based	true, false
Disabled	Boolean	Whether the event is disabled	true, false
Escalation	Boolean	Whether escalation is applied	true, false
SystemEvent	Boolean	Whether it is a system event	true, false

Query Syntax by Search Type

> ⓘ For the basic search syntax, refer to the WhaTap log search query syntax documentation.

1. Keyword Search

Events that contain the string "Database Connection"

```
title: "Database Connection"
```

2. Multiple Value Search

Retrieves events that include any one of the specified values.

```
title: Database Connection
```

Events whose current event level is Info or Warning

```
level: info warning
```

3. Pattern Search

You can search using patterns with wildcards.

Find events that occurred on specific agents using a name pattern

```
oname: web-*-prod
```

タイトルが Connection で終わるイベントを検索

```
title: *Connection
```

4. Complex Condition Search

You can combine multiple conditions to perform precise searches.

Events with a Critical level and a title containing Database

```
level: critical and title: Database
```

Warning or Critical events from a specific agent

```
oname: web-server-01 and level: Warning Critical
```

5. OR Condition Search

Retrieves events that satisfy **at least one** of the specified conditions.

Events where the title or message contains a specific keyword

```
title: Connection or message: Connection
```

6. Exclusion Conditions

You can exclude specific conditions from the search.

All events except those with an Info level

```
not level: info
```

Real-time Notifications

The **Alert status** feature displays pop-up notifications when events configured in **Event Configuration** occur. You can receive alerts immediately when an event is triggered, allowing you to quickly recognize issues and respond. You can also review notifications that have already occurred or were missed, helping you understand and resolve situations where real-time response was difficult.

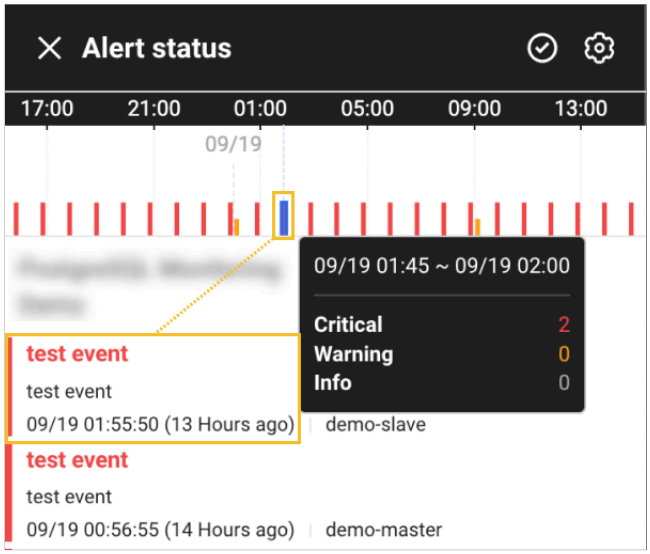
Viewing Real-time Notifications

Real-time notifications appear as pop-ups in the upper-right corner of the screen. You can check the time, title, and message of each notification.

- To close an individual notification, click the ✕ button in the upper-right corner of the message.
- To close all pop-up notifications displayed on the screen, click **Close All Notifications**.

Viewing Notification History

1. Click the 🔔 icon at the top of the screen after selecting a project.
2. The **Alert status** window appears.
    - You can view up to 100 notifications from the past 24 hours for the current project.
    - Selecting an individual event item navigates you to the related menu.
    - In the time-based chart within the **Alert status** window, selecting a specific time range takes you to the corresponding event items.



- To view the full notification history, click **View All Notification History**. This navigates to **Alert** > **Event History**.

> If the **Alert status** list is empty, the **View All Notification History** button is displayed. Disable the **Receive notifications without reception tags** option.

Resolving Events in Bulk

You can resolve all ongoing events at once from the **Alert status** list. The bulk resolution process may take approximately 30 seconds to 1 minute to complete.

- When using the **Add Acknowledge** feature, a large number of recovery (resolved) notifications may be generated for currently ongoing events.
- Even after using the **Add Acknowledge** feature, events may occur again if the event rules remain enabled and the data still exceeds the threshold.
- Recovery notifications can be disabled by editing the event in **Alert** > **Event Configuration**. Disable **Send additional notification when the event is resolved** or **Recovered notification**.

How to use:

1. Click the icon at the top of the screen after selecting a project.
2. When the **Real-time Notifications** window appears, click the icon in the upper-right corner.
3. When the message `Resolve all ongoing events in the real-time notification list.` appears, click the **Resolve** button.

Configuring Real-time Notifications

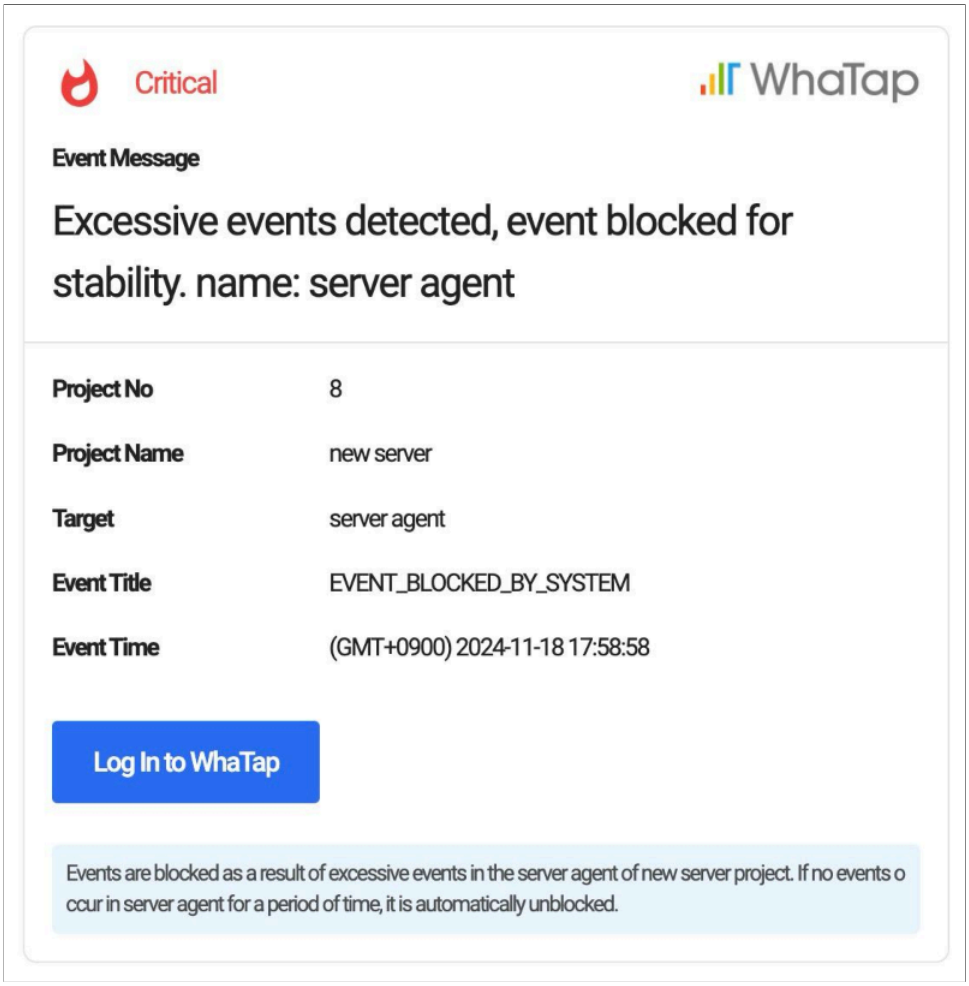
1. Click the icon at the top of the screen after selecting a project.
2. When the **Alert status** window appears, click the icon in the upper-right corner.
3. Configure each option in the real-time notification settings window.
   - **Popup notifications**: Displays notifications as pop-ups even when the **Alert status** window is closed.
   - **Show resolved events**: Allows resolved events to be displayed in the **Alert status** list.
   - **Preview notification sound**: Lets you preview browser notification sounds. Click the button next to Critical or Warning. If you cannot hear the sound, check your browser settings.

System Events

WhaTap Monitoring provides common system events that are automatically detected and generated across the platform. This document explains the types of events generated by the system, their trigger conditions, and recommended response actions.

EVENT_BLOCKED_BY_SYSTEM

If an excessive number of events occurs within a short period of time in a project, the WhaTap system detects this situation and blocks further events. When events are blocked, the system delivers the following message to the user. If you encounter this event, review and adjust your event configuration settings.



Blocking Criteria

Events are blocked based on the following criteria:

1. **Agent-level blocking**

   If **10 events** occur within **1 minute** from a single agent, all subsequent events from that agent are blocked.

2. **Project-wide blocking triggered by multiple agents**

   If **3 or more agents** are blocked within **1 minute** according to condition **1**, events from the entire project are blocked.

3. **Project-level blocking**

   If **100 events** occur within **1 minute** in a single project, all events from that project are blocked.

How to Release the Block

When events are blocked by the system, no events must occur from the affected agent or project for **1 minute**.

- Disable all configured events from **Alert > Event Configuration**. **Review your event settings and adjust thresholds to prevent excessive event generation.**
- Before configuring events, use the **Simulation** feature to estimate how many events may occur.

Instance Performance Analysis

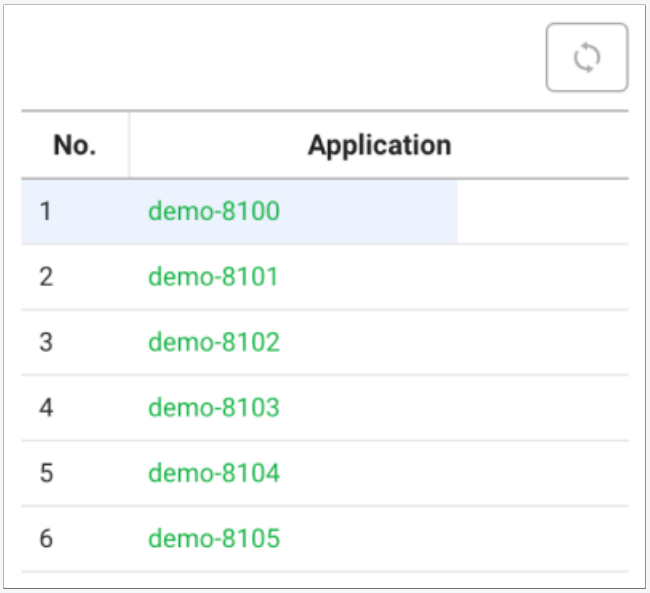
Home > Select Project > **Instance Performance Management**

You can see the application environment along with the performance-related settings. You can see detailed information of agents and applications.

- You can filter the desired items through the text field on the upper right of the screen.
- If changes to agents and applications located on the monitored server are not automatically applied, select **Refresh**.

Agent list

You can check the list of agents assigned to the project in the application list on the left of the screen. If you select an individual agent, you can view information for environment variables, agent settings, and performances set on the right of the screen.

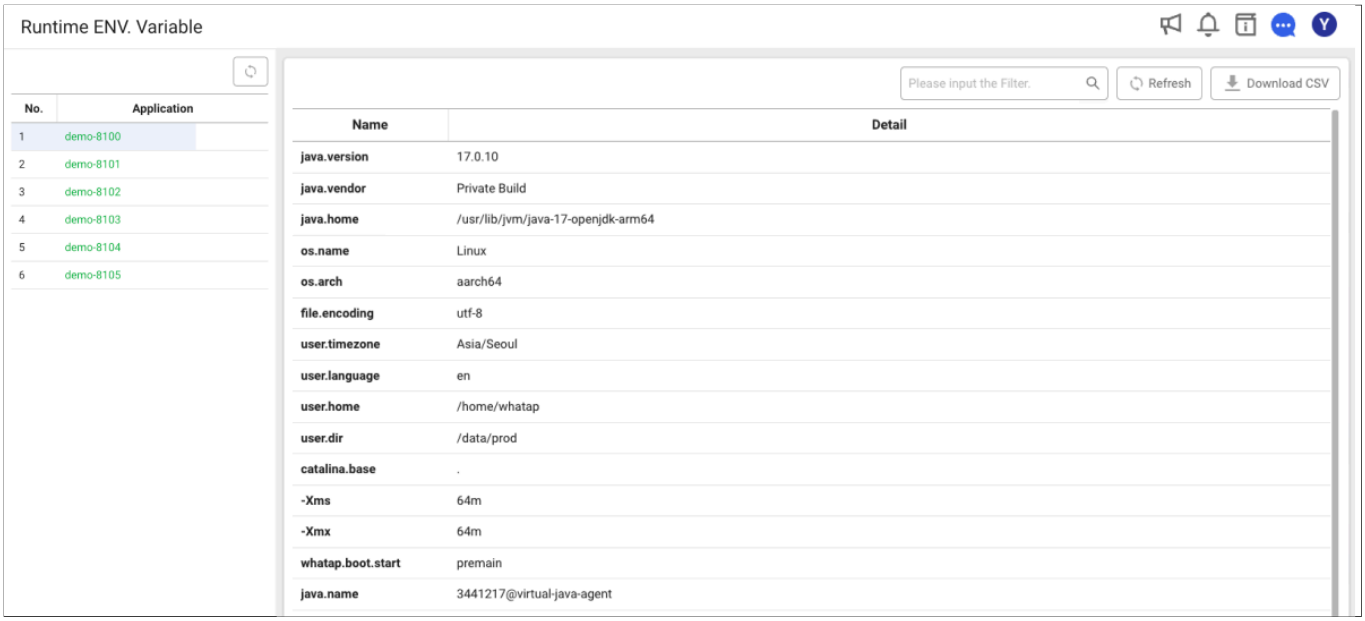


No.	Application
1	demo-8100
2	demo-8101
3	demo-8102
4	demo-8103
5	demo-8104
6	demo-8105

After updating the agent list or restarting a disabled agent, if it is not automatically displayed in the list, select .

Runtime ENV. Variable

Home > Select Project > **Instance Performance Management** > **Runtime ENV. Variable**

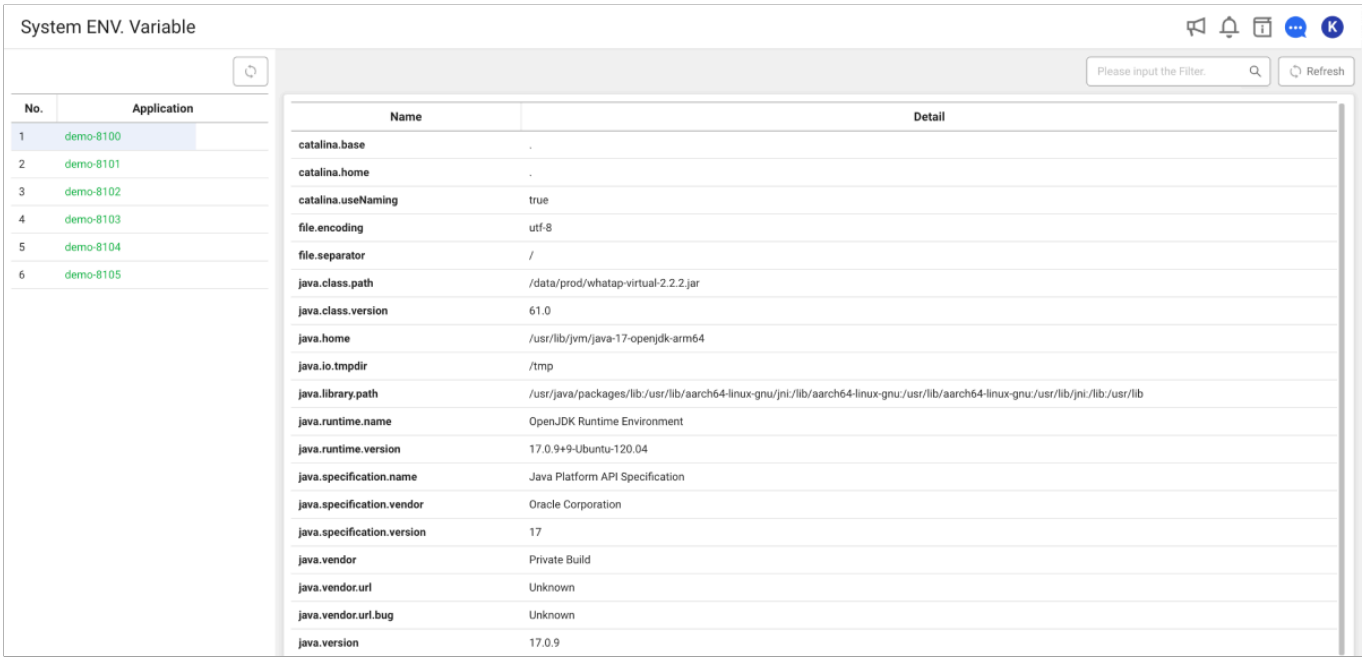


You can view environment variables for agent execution. Important information among environment variables or application status can be stored on the WaTap server so that they can be retrieved even if the agent is down. You can see the version, installation path, name, and IP address of the agent. You can download environment variables by clicking **Download CSV**. The following are key points to check:

- `(Application).version` : You can check the application version.
- `file.encoding` : You can check the file encoding format such as `utf-8` .
- `os.name` : You can check the current OS.
- `user.language` : You can check the user language.
- `-Xms` , `-Xms` : You can check environment variables that are related to heap memory settings.
- `(Application).start` : You can check the application start time.

System ENV. Variable

Home > Select Project > **Instance Performance Management** > **System ENV. Variable**

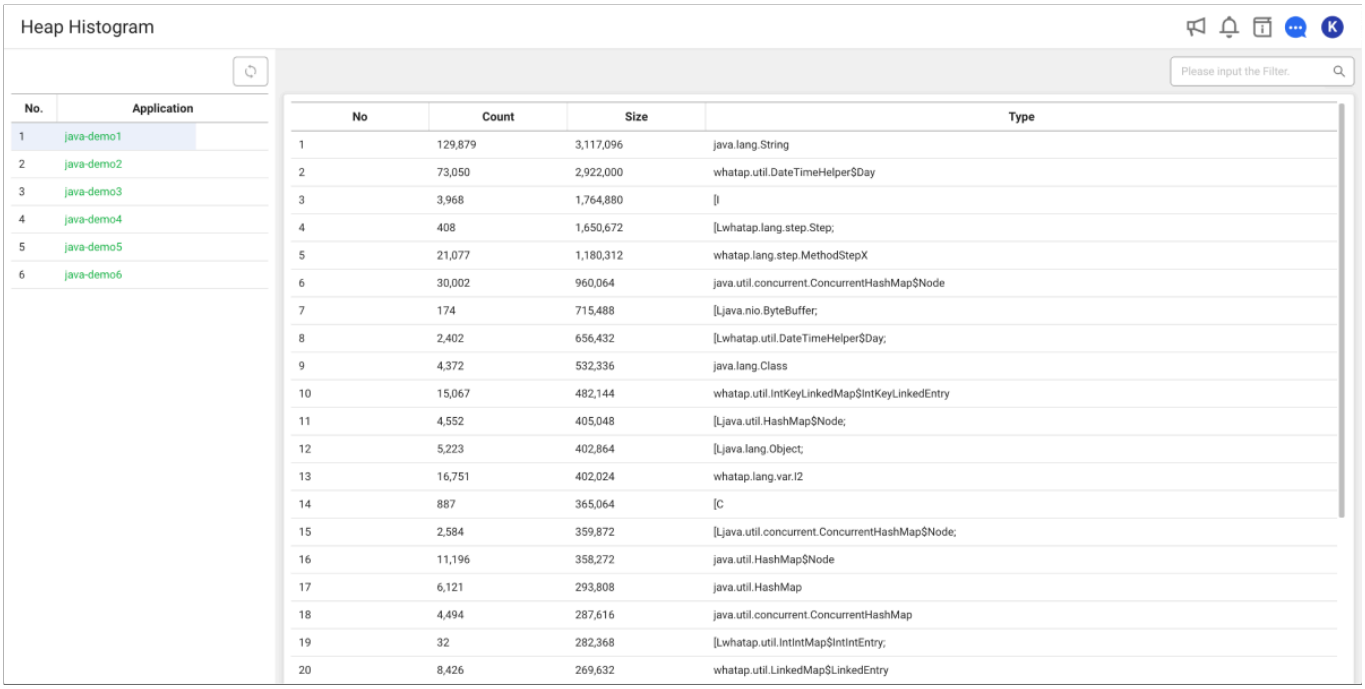


You can view environment variables for application execution. The **Boot environment** menu collects and provides environment variables at the time of query. It uses `System.getProperty()` to retrieve the value of the key passed as a parameter. The following are key points to check:

- `(Application).version` : You can check the application version.
- `file.encoding` : You can check the file encoding format such as `utf-8` .
- Application state: For example, if the `catalina.home` environment variable is retrieved, you can see that it is the tomcat environment.
- OS environment variables: For example, they are provided in the suffix `(env)` format, such as `(env) LANG` environment variable.

Heap Histogram

Home > Select Project > **Instance Performance Management** > **Heap Histogram**



You can view the status of heap occupied objects (sizes for each object in heap memory) in the memory of the JVM (Java Virtual Machine). It retrieves the pure size of the object. Through the **Count** column, it is easy to check the number of main objects.

JVM Option

Java versions 6 to 8 support basic functions without the JVM options. However, for some Java versions, the JVM options must be applied as follows:

- Java 9 to Java 15

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

example

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

- Java 16 or later

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

example

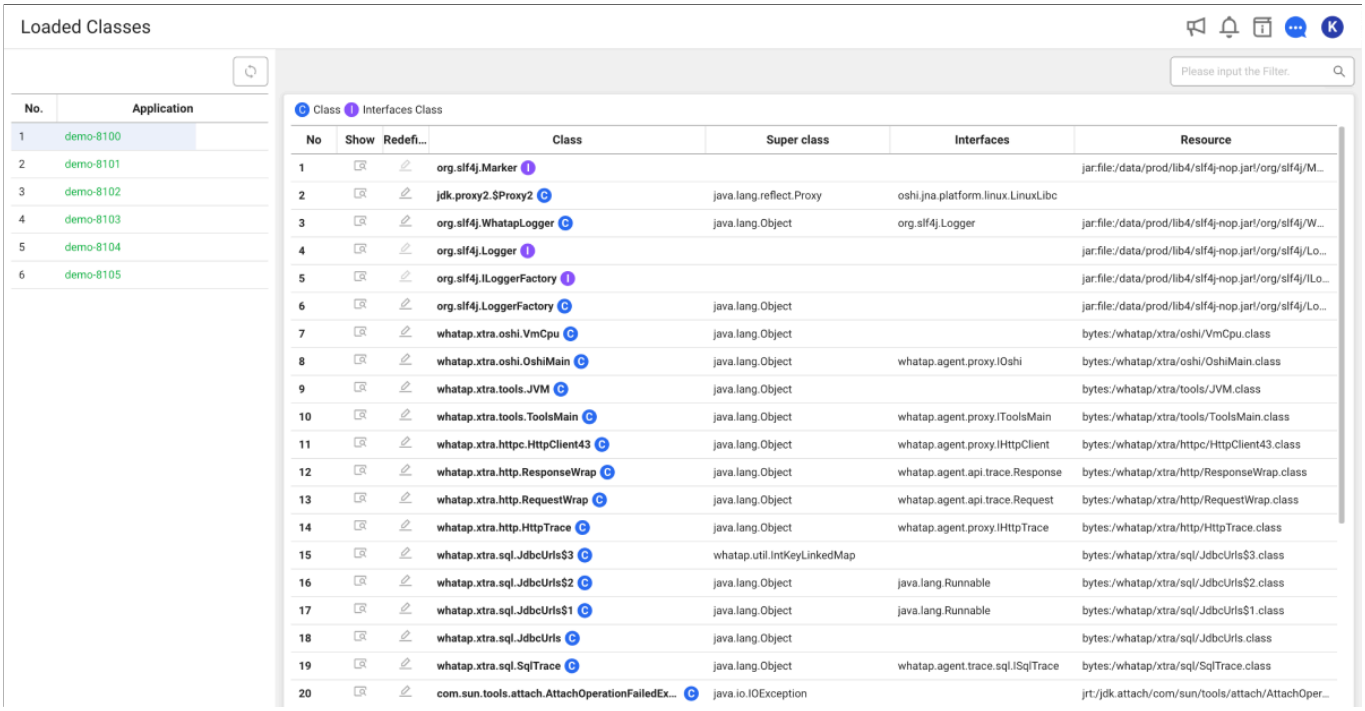
```
java -javaagent:{WHATAP_HOME}/whatap.agent-X.Y.Z.jar -Djdk.attach.allowAttachSelf=true --add-opens=java.base/
java.lang=ALL-UNNAMED --add-opens=jdk.attach/sun.tools.attach=ALL-UNNAMED -jar {application.jar}
```

- It does not support Java 5 or earlier and IBM Java.

> ⓘ Because it retrieves the number of objects that are kept until GC, they appear in the list even if they are not actually used. For an accurate query, it is recommended running System GC and then checking the list. For more information about System GC, see the following.

Loaded Classes

Home > Select Project > **Instance Performance Management** > **Loaded Classes**



You can see class information loaded in Java applications. The **Loaded Classes** menu allows you to retrieve the list of classes loaded into your application and their detailed data, and to change the behaviors of classes without interruption.

- **Class details**: When you click the icon in the **Class details** column, you can see **Class details**. The **Class details** provides information about methods and inheritance relationships within the class.
- **Redefine**: If you click the icon in the **Redefine** column, the class can be redefined. You can change the behaviors of the class that has been loaded at runtime without stopping the application by reloading it and then using the byte code injection (BCI).
- **Resource**: You can check which physical file the class was loaded from. Due to the complex multi-layered nature of the application, we need to check the **Resource** information to determine which jar file a class was loaded from.

> For more information about using the **Redefine** feature and **Resource**, see the following.

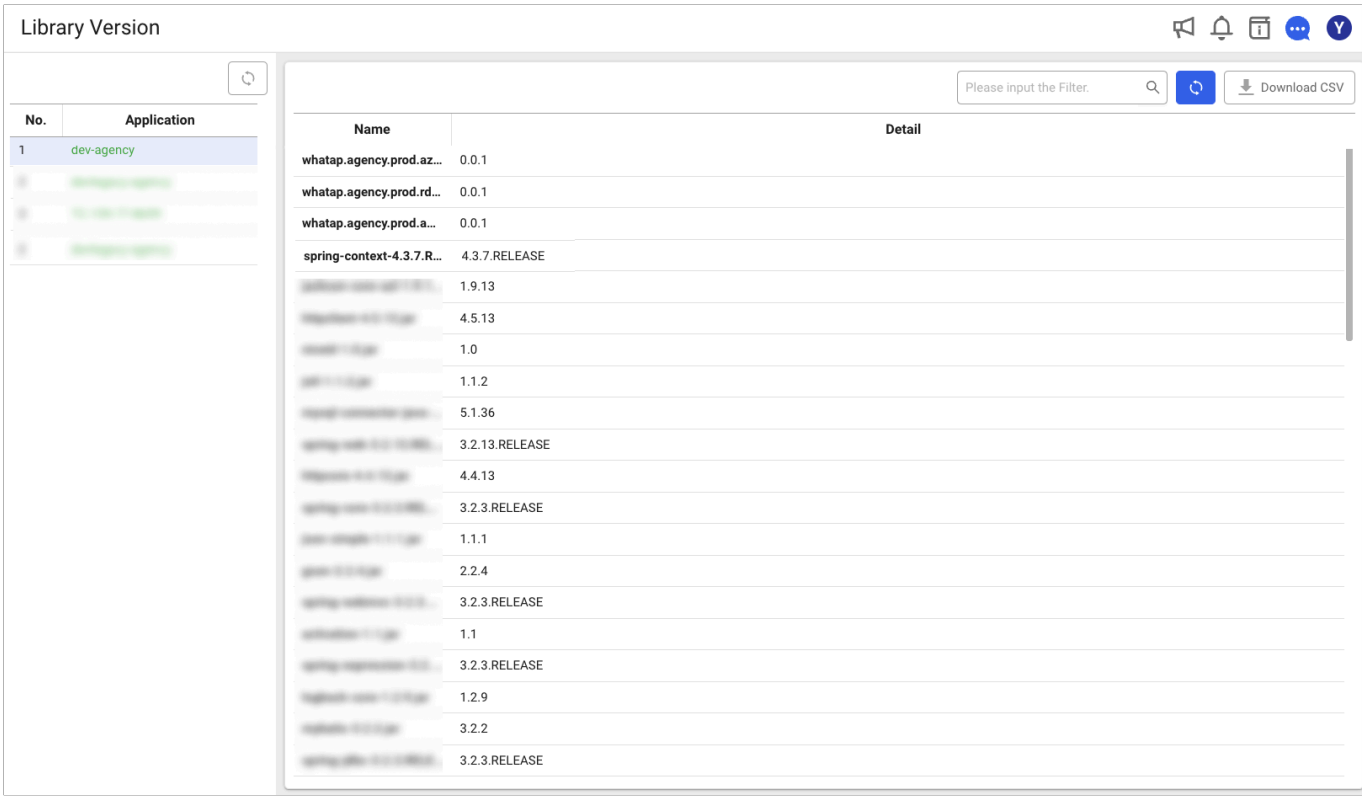
> **BCI(ByteCode Instrumentation)**
>
> Java bytecode is a platform-independent intermediate code that runs on the JVM. **BCI** (byte code

injection), is a technique to implement additional functionality by dynamically changing the bytecode at runtime or build time without modifying the source file. For more information, see the following document.

- Java SE Specifications
- Java Package Summary

Library Version

Home > Select Project > **Instance Performance Management** > **Library Version**

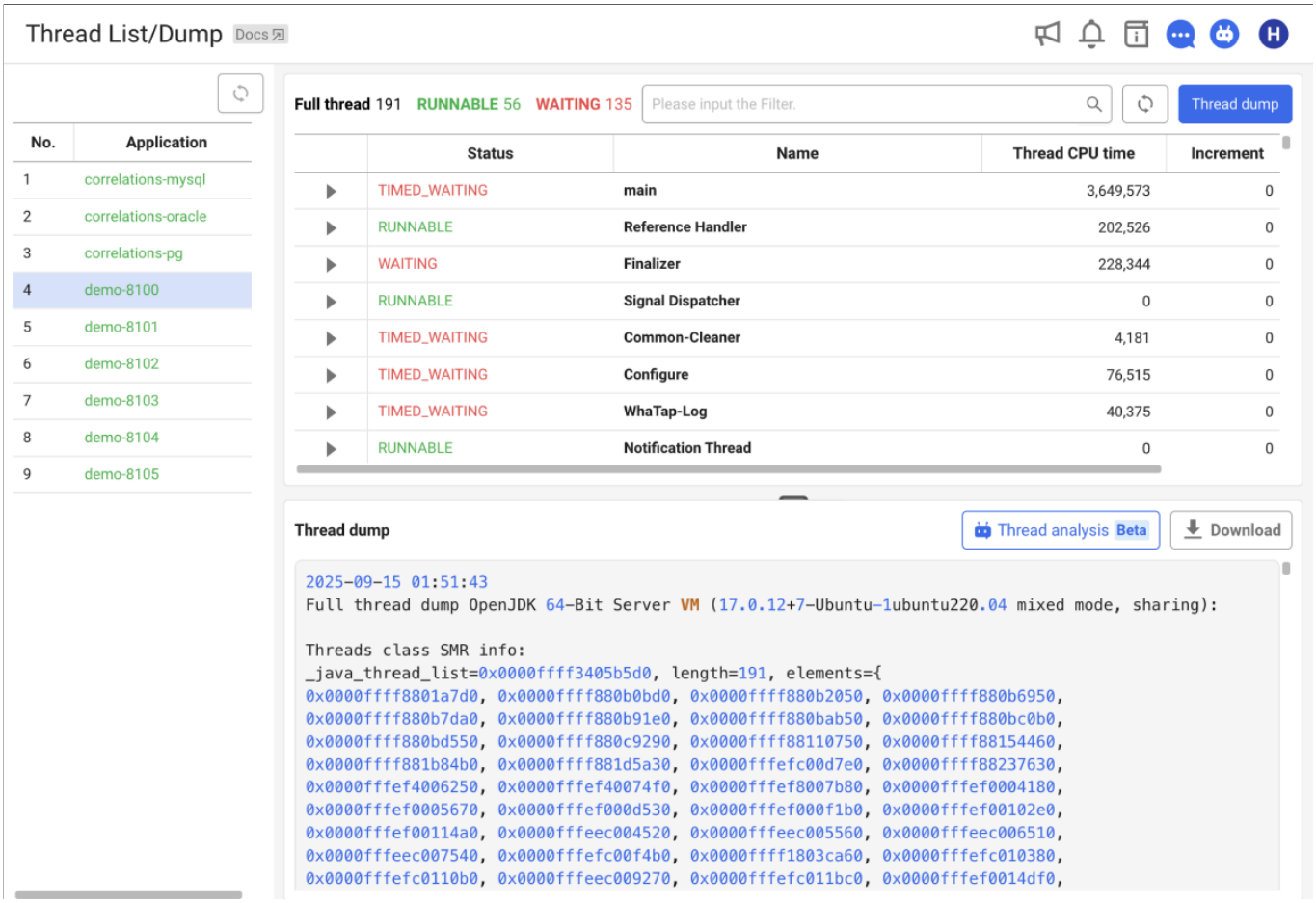


You can view a list of components deployed to the Java application. In the environments with lots of open source libraries, you can quickly check and resolve compatibility issues by checking the version of the jar file. Additionally, checking the version makes it easy to manage security, such as finding and patching versions with security vulnerabilities.

- Component list Info: It provides **Name** and version for the corresponding jar file in **Detail**.
- **Download CSV**: You can download the version information by clicking Download.

Thread List/Dump

Home > Select Project > **Instance Performance Management** > **Thread List/Dump**



It can provide snapshot information of all threads in the Java process. **Thread List/Dump** is a key menu for CPU optimization. You can manage CPU resources more finely by checking the status at the thread level rather than the process level. For example, you can identify performance issues by checking the thread that is using the most CPU at the time of query and then checking the stack information for the thread.

- Thread status: You can see the **Number of all threads** and the number of threads in RUNNABLE and WAITING

status through the status summary at the top of the list.

- **Thread dump**: You can create a full thread dump by clicking Thread Dump on the upper right. After generating a full thread dump, you can check the information at the bottom of the thread list.
- Individual thread dump: You can see individual thread dump information by selecting the ▶ icon for each thread. You can view the details and stack traces about the thread state and execution path, in **Stack** and **Stack trace**. The list can be checked after refreshing it. For more information, see the following.
- In **Thread CPU time**: It indicates the total accumulated time that the thread has occupied the CPU. It uses the time in milliseconds.
- **Increment**: If you click the Refresh icon at the top of the thread list, the difference between the CPU time of the existing thread and the CPU time of the current thread is added.

> **Thread CPU time** is the total accumulated time that the thread has occupied the CPU. Therefore, to identify the thread that is using the most CPU at the query time, it is required to check the **Increment** value. For more information, see the following.

> **CPU usage spike types**
>
> - Heap memory is full: CPU usage spikes due to frequent GC.
> - There are lots of requests: CPU usage spikes due to processing a large number of client requests.
> - Wrong code logic: Inefficient algorithms or infinite loops can cause spikes in CPU usage.

Thread Dump Details

It guides you to individual thread dumps that can be checked in **Thread list/dump**.

JVM Option

To use the **Thread dump** feature for each Java version, additional configuration for JVM options is required.

Java versions 6 to 8 support basic functions without the JVM options. However, for some Java versions, the JVM options must be applied as follows:

- Java 9 to Java 15

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

example

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

- Java 16 or later

```
-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}
```

- It does not support Java 5 or earlier and IBM Java.

Thread status

The states of the thread are declared as enumerated types named `State` in the `java.lang.Thread` class.

- `NEW` : The thread has been generated but not run.
- `RUNNABLE` : The CPU has been occupied and the tasks are being performed. The resource allocation by the OS may result in `WAITING` state.
- `WAITING` : It is waiting using the `wait()` , `join()` , and `park()` methods.
- `TIMED_WAITING` : It is waiting using the `sleep()` , `wait()` , `join()` , and `park()` methods. The difference from the `WAITING` state is that it is waiting for a given time. The waiting state can be released by the time as well as external environment change.
- `BLOCKED` : It is waiting for the locked object to be released.
- `TERMINATED` : Execution has been completed.

Thread dump information

You can check individual thread dumps by selecting the ▸ button for each thread in the thread list.



Stack

Item	Properties	Description
Thread ID	threadId	Unique ID allocated to the thread
Lock owner Id	lockOwnerId	Thread ID that holds the monitor lock state on the object on which the threads are blocked.
Waited count	waitedCount	Total number of WAITING and TIMED_WAITING states of the thread
Blocked count	blockedCount	Total number of Blocked states of the thread
Waited time	waitedTime	Elapsed time (milliseconds) that the thread has been in the WAITING state. It returns -1 when the thread contention monitoring is disabled.
Thread CPU time	threadCpuTime	It displays the total CPU time (nanoseconds) of the thread calculated in milliseconds. It returns -1 when the CPU time measurement is disabled.
Status	threadstate	Thread status
Blocked time	blockedTime	Elapsed time (milliseconds) that the thread has been in the BLOCKED state. It returns -1 when the thread contention monitoring is disabled.
Thread	threadName	Unique name of the thread

Item	Properties	Description
name		
Lock name	lockName	String indicating that the thread's input is blocked or a monitor lock prompts for notification via the Object.wait method.
Lock owner name	lockOwnerName	Name of the thread that holds the monitor lock on the objects on which the thread is blocked.
Thread user time	threadUserTime	The CPU time (nanoseconds) for which the thread executed in user mode that is calculated in milliseconds.

> For more information about the thread information, see the following link.

Stack trace

It displays the stack trace when any exception occurs. The stack frame list that contains information about the methods called by the code.

AI Thread Analysis

In the thread dump detail screen, click the **Thread analysis** button.

AI-based analysis results are provided in a drawer.

Analysis Result Items

- Summary
- Total number of threads
- Distribution by thread state
- Issues

- Recommendations

Retry Thread Analysis

Click the button at the top of the drawer to analyze the same thread dump again.

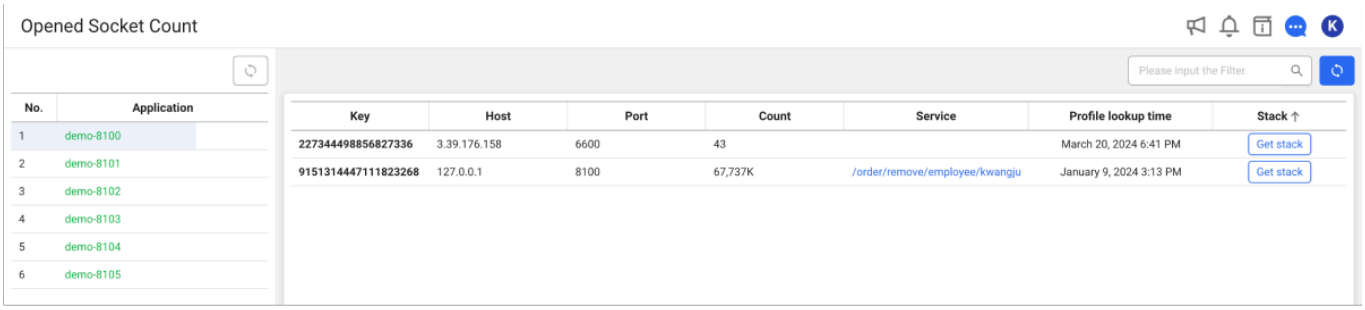
The analysis results are displayed in the language set on the WhaTap page.
In some cases, results may be displayed in English.

Each project can be analyzed up to 20 times per day, and the count resets daily at 00:00:00 based on the project time zone.

- Criteria for analysis count deduction
  - One analysis is deducted when analyzing a new thread dump
    - Application selection state changes
      - Selecting another application from the application list
    - Time value changes
      - Selecting another application (both selection state and time change)
      - Selecting the same application again (selection state remains, but time changes and the thread dump is updated)
  - One analysis is deducted even when retrying analysis on the same thread dump inside the drawer

Opened Socket Count

Home > Select Project > **Instance Performance Management** > **Opened Socket Count**



You can check the socket information that the Java application opened to perform TCP features. This can be used to view the network dependency, which is the external system that the application instance depends on, such as database. You can also trace who it is making outbound calls to. This information is important for optimizing the efficiency, reliability, and performance of TCP communications.

- **Count**: It indicates the number of connection attempts to external systems.
- **Stack**: By selecting **Get stack**, the stacks acquired on the next connections can be viewed and deleted.

It can sort by **Count** and then check **Host** and **Port** to find which services or systems have the most connections. Afterwards, obtain **Stack** to view the detailed content.

> ⓘ **Database and connection pool**
>
> For example, if the **Count** of 3306 **Port** increases, it is possible that MySQL is not managing connections properly, such as not using the connection pool or frequently disconnecting and recreating the connection pool. To check whether the 3306 **Port** is being used in which code path, you can obtain **Stack** for tracing.

Method Performance

Home > Select Project > **Instance Performance Management** > **Method Performance**

It provides the performance statistics about the method executed in the Java application.

- **Refresh**: It recalls the data.
- **Reset**: It initializes the statistical data for the registered methods.

> ⓘ To use the **Method Performance** menu, add the following custom options in **Agent CONFIG.**. In the reboot or **Loaded Classes** menu, perform **Redefine**.
>
> `hook_method_stat_patterns=io.home.test.baseapp.app.post.mapper.PostMapperImpl.*`

DataSource Status

Home > Select Project > **Instance Performance Management** > **DataSource Status**

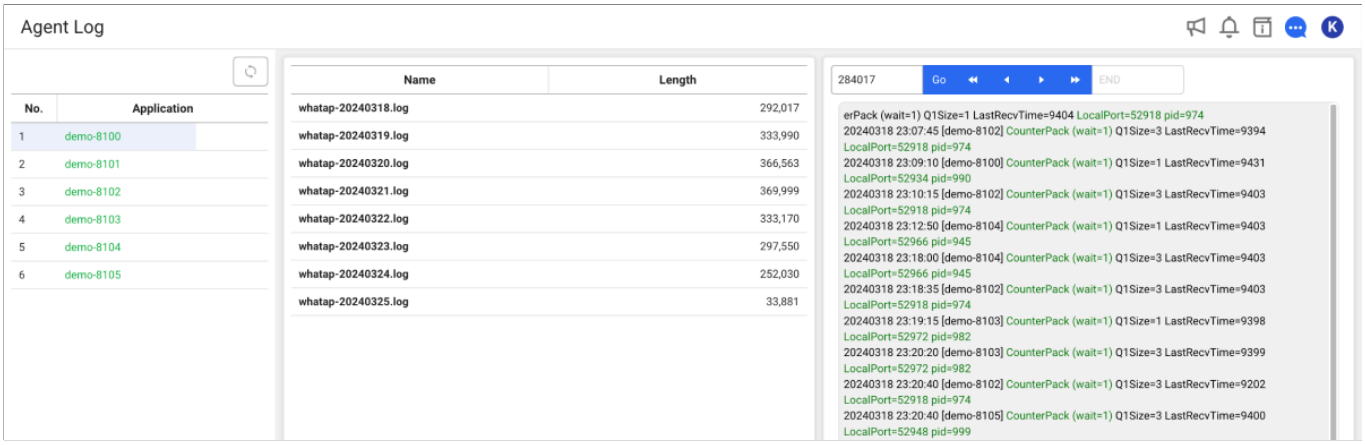


You can see the status of DataSource. It provides information for current status and configuration of the database connection pool used by the application, connection settings, connection state, and configuration mismatches.

> Because the monitoring starts as soon as a connection pool object is created, it may appear in the data source status even if the object is not being used.

Agent log

Home > Select Project > **Instance Performance Management** > **Agent Log**



You can view the agent logs stored on the monitored server. The name of the log file is in the format of whatap-`YYYYMMDD`.log. You can select each log to access information about the errors and events captured in the logs.

> For the agent settings for the log, see the following.

System GC

Home > Select Project > **Instance Performance Management** > **System GC**



By calling `System.gc()`, you can collect and remove memory objects (garbage) that are no longer needed for the dynamically allocated memory in the Heap area of the JVM (Java Virtual Machine). If you select **Run GC**, the GC process is performed and you can see the memory capacities before and after execution.

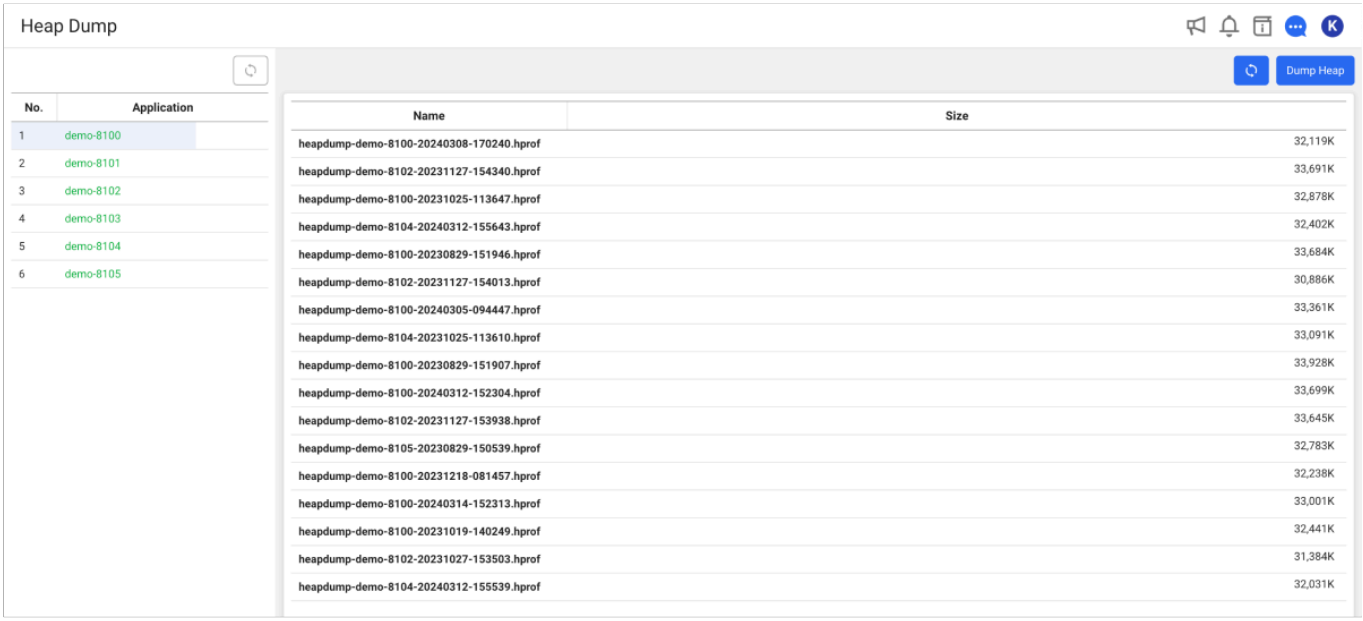
> If the option has been set to disable explicit GC requests, such as `-XX:+DisableExplicitGC`, **System GC** does not work.

> **System GC and memory leak**
>
> When the heap memory usage continues to rise, the System GC is required to accurately identify and diagnose the possibility of memory leak.
>
> - **Memory requested or not**: Check if the application is continuously requesting memory. Check if memory requests increased more than expected or memory is being freed correctly.
> - **Checking memory usage**: Check if excessive memory usage is happening. Run GC to free objects that are no longer required and observe changes in memory usage.

- **GC cycle**: Check if GC is not running even if there is sufficient free memory. Check the GC cycle or settings.

Heap Dump

Home > Select Project > **Instance Performance Management** > **Heap Dump**



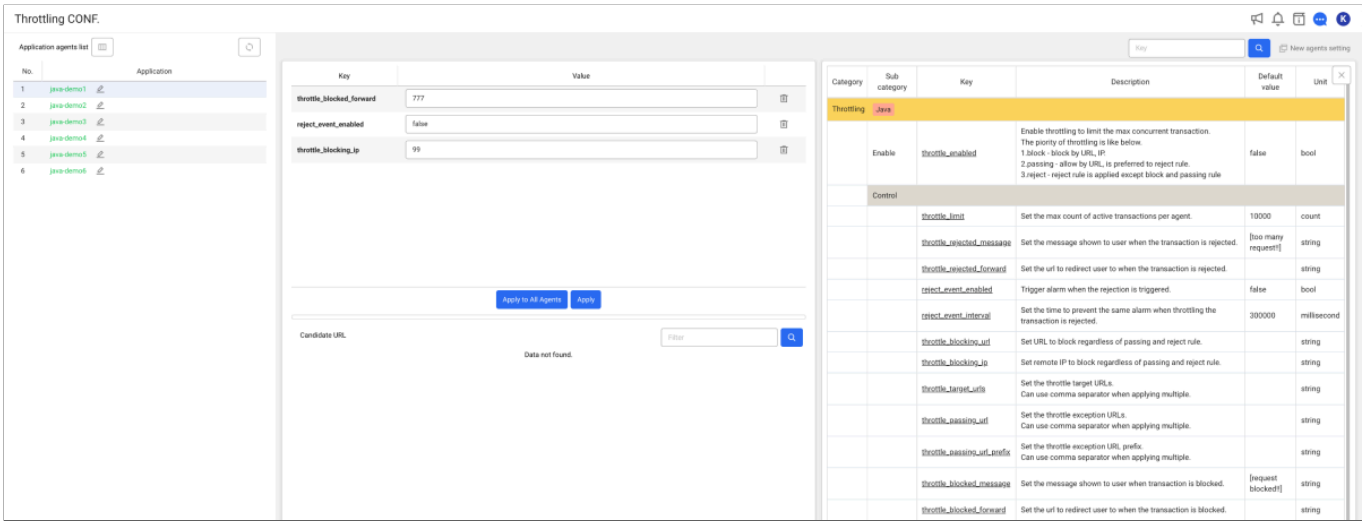
If any problem occurs such as a memory leak while running the Java application, you can generate its dump file for analysis of the problem. Select **Dump Heap**. The generated dump file is stored on the application server. You can see it in the dump directory under the path where you installed the agent.

This function is supported in Java agent 1.5.2 or later. Set the `heapdump_enabled` option to `true` in the agent configuration. Select **Dump Heap** to enable the function to create a heap dump file. It does not automatically generate a heap dump each time, and this option does not affect the performance. However, dump acquisition may affect the performance.

Obtaining a heap dump requires a significant **burden**. Separate from the burden, because the execution of the application code is stopped (stop the world) until all heap dumps are extracted, the perceived performance difference by the user may be more severe.

Throttling CONF.

Home > Select Project > **Instance Performance Management** > **Throttling CONF.**



You can see the agent settings for load control of Java applications, and then set the related options. For more information about the agent settings for load control, see the following.

Agent CONF. *Old*

Home > Select Project > **Instance Performance Management** > **Agent CONF.** *Old*

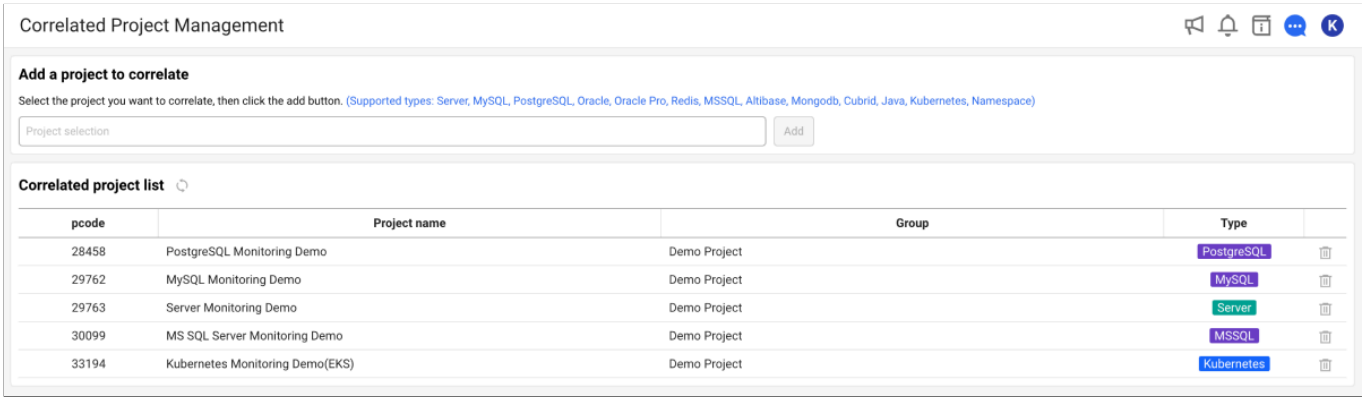
You can add, modify, or delete the agent configuration options without directly modifying the whatap.conf file located on the monitored server. When you click **Download Config File**, the whatap.conf file of the selected agent is downloaded.

> ⓘ For the agent configuration, it is recommended to use the **Management** > **Agent CONF.** menu that has improved the usability and functionality. On the upper right of the screen, select **New agents setting**. For more information, see the following.

Linked Projects

Home > Select Project > **Management** > **Correlated Project Management**

If you have created a database monitoring (DPM) project and an application monitoring (APM) project, you can relate the two projects for monitoring the data collected from DPM in the APM project.



The **Correlated Project Management** feature allows you to go beyond product-/equipment-centric monitoring and view monitoring data across multiple projects. This is useful for analyzing how different system components such as applications and databases, interact together. In particular, when performance degradation occurs, you can quickly determine whether it is a hardware problem or an individual application problem, which can significantly reduce the troubleshooting time. Through **Correlated Project Management**, you can comprehensively analyze performance data at the user system level and obtain more visible insights.

Through the following video clip, see the content details.

Notes before you begin

- The database platforms that support the **Correlated project** feature are as follows:

  PostgreSQL, Oracle, MySQL, SQL Server, CUBRID, Altibase, Redis, MongoDB

- The **Correlated DB session** feature requires Java agent 2.2.33 or later. Database platforms that support this feature are as follows:

  PostgreSQL, Oracle, Oracle Pro, MySQL

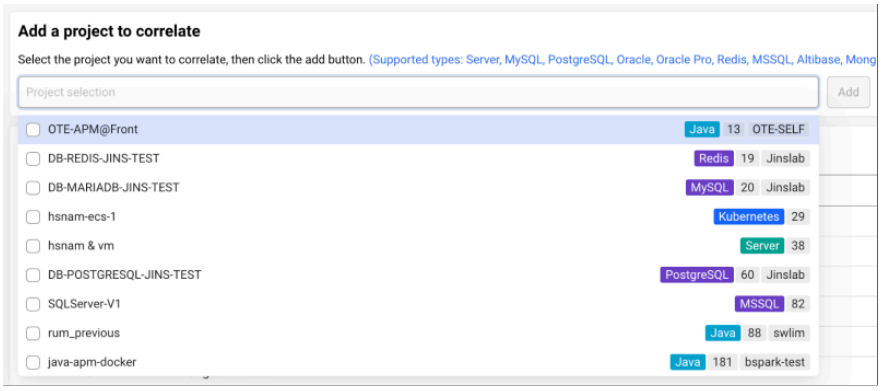
Role configuration

For Oracle and Oracle Pro platforms, the following roles are required to add linked projects.

```
grant select on sys.v_$session to DB access account;
# ex) grant select on sys.v_$session to petapp;
```

Adding any linked project

1. Go to **Management** > **Correlated Project Management**.
2. In the **Add a project to correlate** section, select the **Project selection** input field.



3. If you see a list of projects, select a project to link to. You can also enter a string to search for matching projects.
4. Select one or more projects and then select **Add** that is active.

The selected project is added in **Correlated project list**.

You need the **Edit** role on the project to add a linked project. If you have the **Edit** role, the **Add** button is enabled in the **Correlated Project Management** menu.

Setting the agent options

For the database project and associated analysis, set the options in the agent configuration file (whatap.conf) as follows. Or configuration is possible in **Management** > **Agent CONF.**.

whatap.conf

```
dbc_sid_enabled=true
hook_db_native_enabled=true
debug_dbc_sid_enabled=true
```

- **dbc_sid_enabled** bool

  Default `false`

  This option is used to issue an SID. Set the value to `true` for associated analysis. It is the **Required** option.

- **hook_db_native_enabled** bool

  Default `false`

  This option is used to extract the DB type used in the transaction. Set the value to `true` for associated analysis. It is the **Required** option.

- **debug_dbc_sid_enabled** bool

  Default `false`

  It is optional to write the SID-related information to the Java agent log.

> ⓘ **Management** > **Agent CONF.**



Checking the metrics of the linked project

Metrics related to the added project can be checked in the following menu path.

- **Dashboard** > **Application Dashboard**
- **Dashboard** > **Active Transaction**
- **Analysis** > **Hitmap**

- For more information about the **Application Dashboard** menu, see the following.
- For more information about the **Active Transaction** menu, see the following.
- For more information about **Hitmap**, see the following.

Application Dashboard

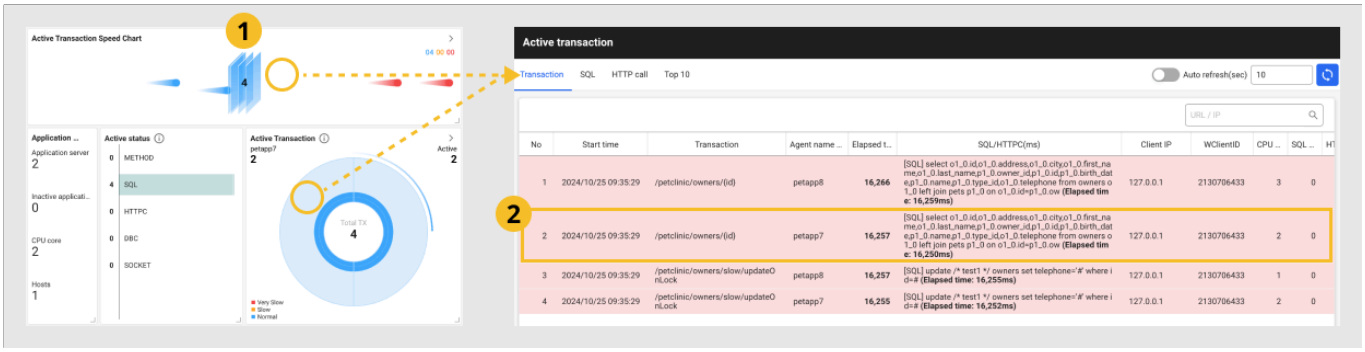
You can check the linked projects through the following widget in the **Application Dashboard** menu.

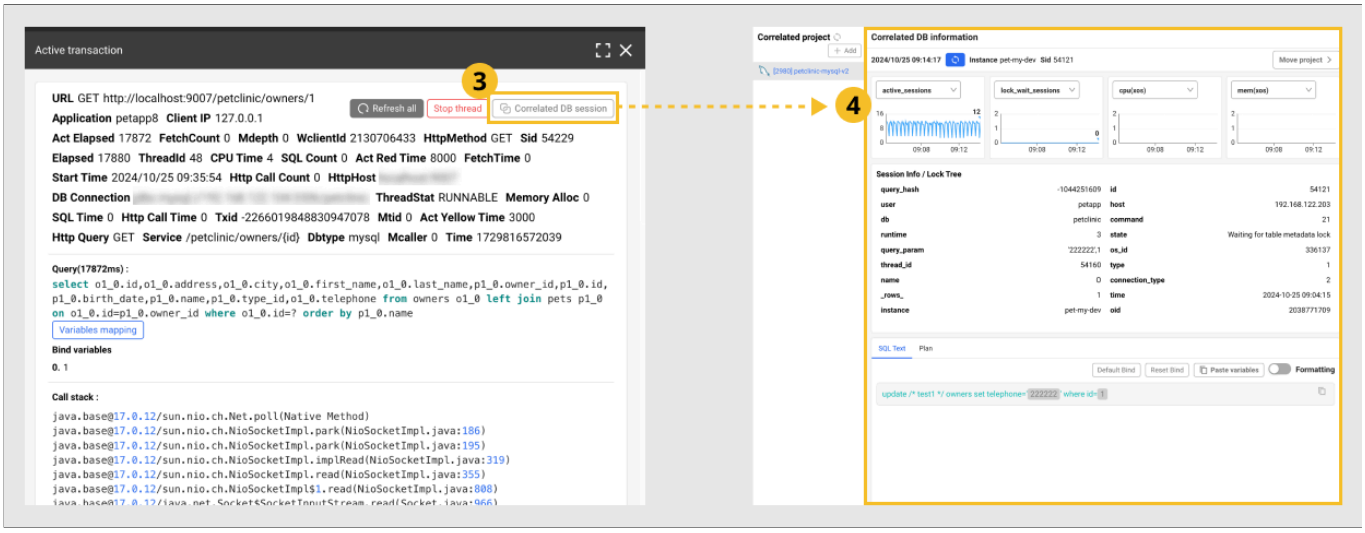
- **Active Transaction Speed Chart**
- **Active Transaction**

- **Hitmap**

Active Transaction Speed Chart / Active Transaction widget

You can query the SQL performance of the database associated with active transactions. Go to **Dashboard** > **Application Dashboard**.





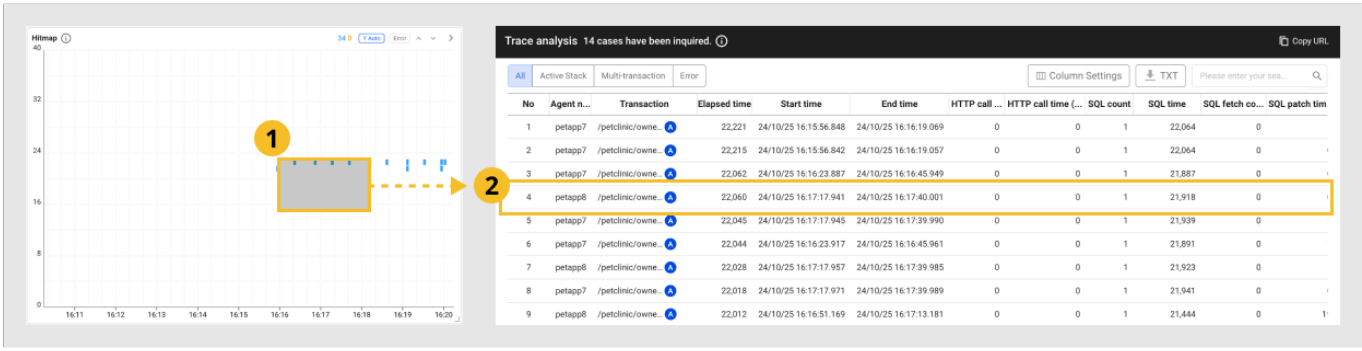
1. On the dashboard, click the chart area of the **Active Transaction Speed Chart** or **Active Transaction** widget.
2. When the **Active transaction** window appears, select transactions.
3. When the transaction details window appears, select **Correlated DB session** on the upper right.
4. When the **Correlated DB information** window appears, check the related metrics.

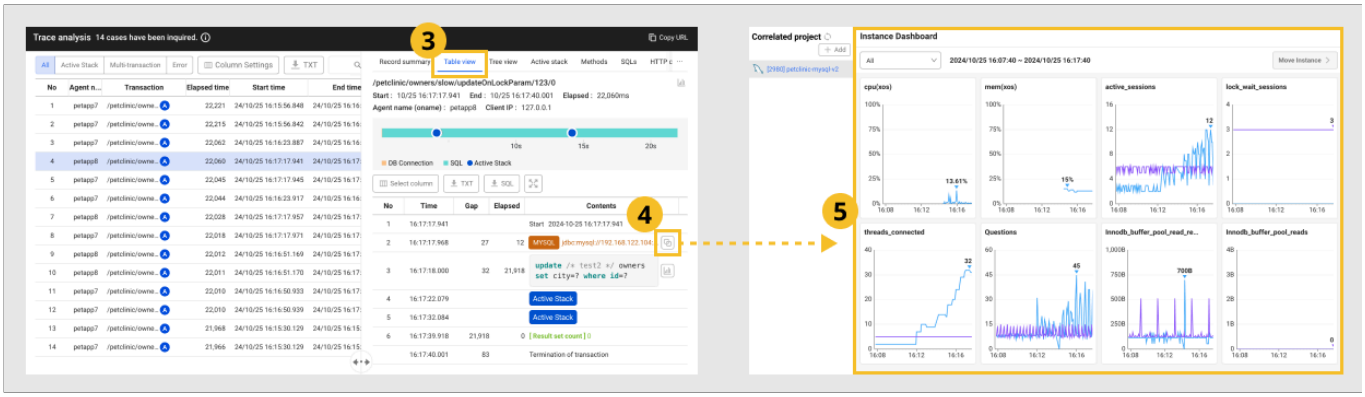
- The **Correlated DB session** button appears in the transaction details window only when the DB of the database project registered for linked analysis has executed the SQL of the selected transaction.
- For more information about the **Correlated DB information** window, see the following.
- In **the versions before Java Agent 2.2.40**, the **ThreadStat** values appear in numeric format on the **Active transaction** window. To display in text format, install Java Agent 2.2.40 or later.
- The databases that support this feature are **PostgreSQL**, **Oracle**, **Oracle Pro**, and **MySQL** platforms.

Hitmap Widget

You can view performance metrics of the database associated with transactions from the past. Go to **Dashboard** > **Application Dashboard**.

Checking in the DB connection step



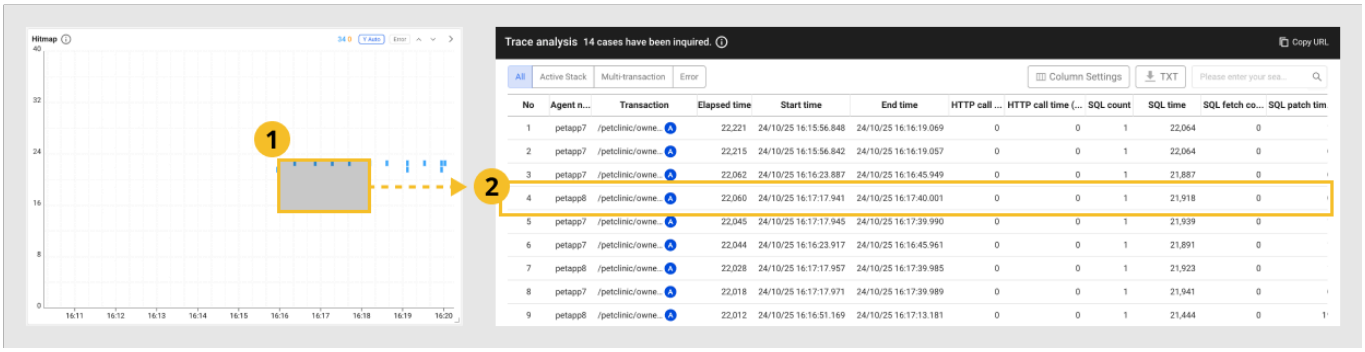


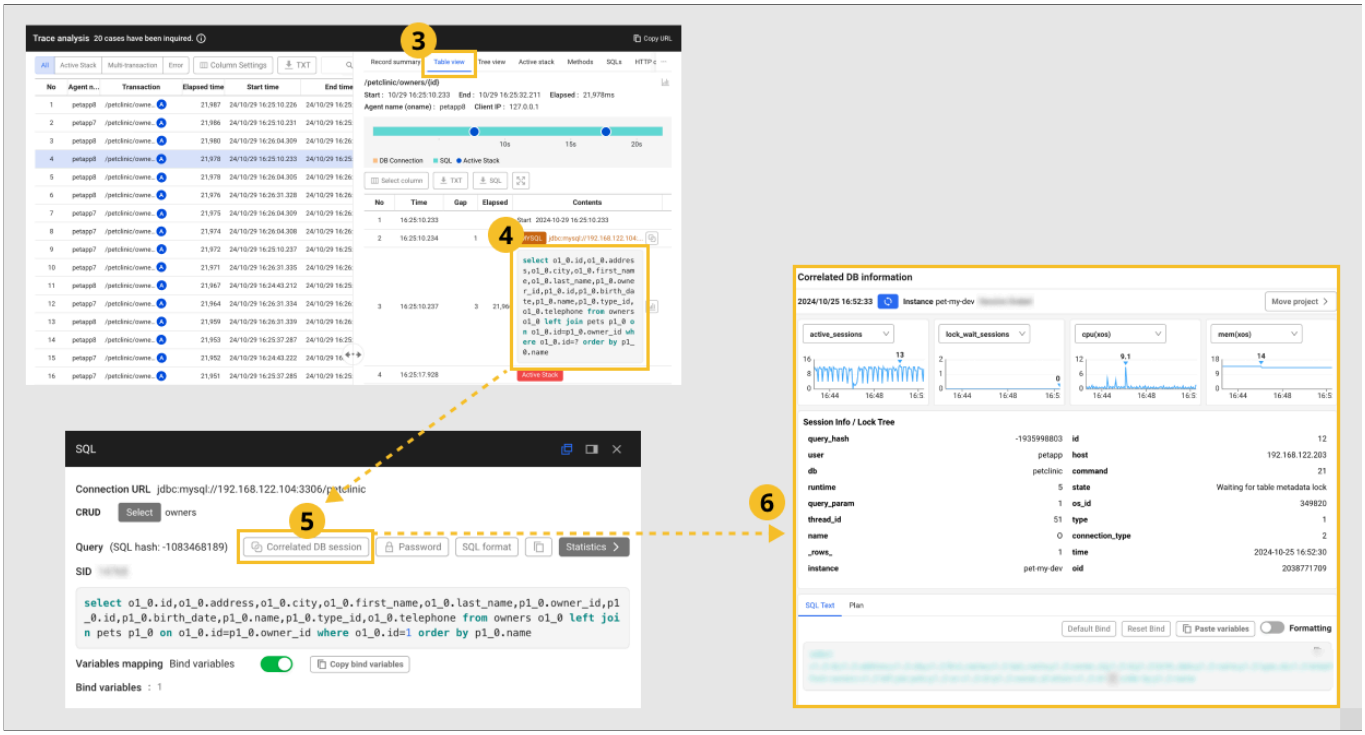
1. Drag the chart area of the **Hitmap** widget on the dashboard.

2. When the **Trace analysis** window appears, select a transaction to view.
3. When the transaction details appear on the right, select the **Table view** tab.
4. In the DB Connection step, select .
5. When the **Instance dashboard** window of the linked project appears, check the related metrics.

- The button appears on the **Table view** tab only when the DB of the database project registered for linked analysis executes the SQL statement of the selected transaction.
- For more information about the **Instance dashboard** window, see the following.
- For more information about the **Trace analysis** window, see the following.
- The databases that support this feature are **PostgreSQL**, **Oracle**, **Oracle Pro**, and **MySQL** platforms.

Checking in the SQL step

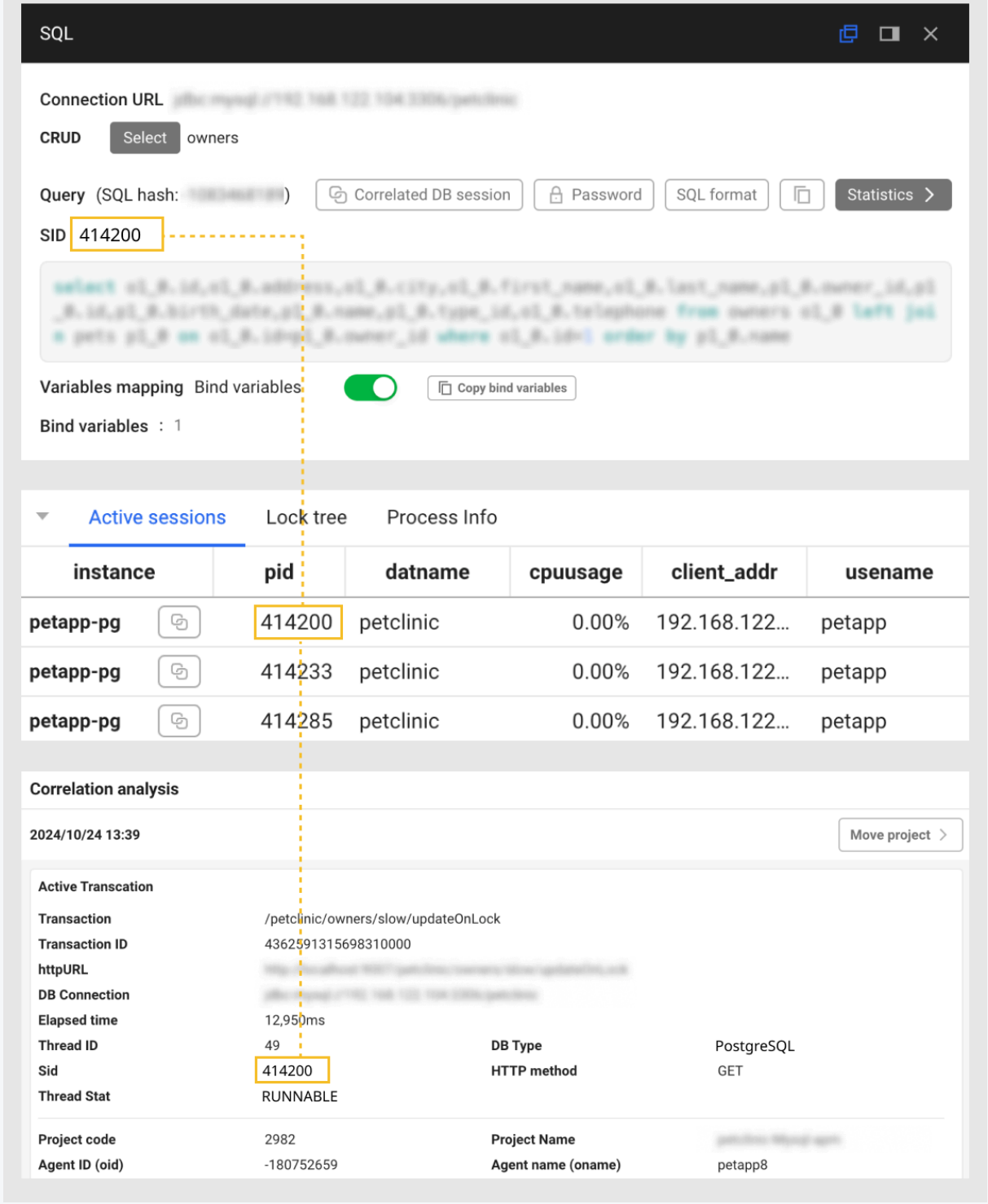




1. Drag the chart area of the **Hitmap** widget on the dashboard.
2. When the **Trace analysis** window appears, select a transaction to view.
3. When the transaction details appear on the right, select the **Table view** tab.
4. Select a SQL statement in the SQL step.
5. If the **SQL** window appears, select **Correlated DB session**.
6. When the **Correlated DB information** window appears, check the related metrics.

- The **Correlated DB session** button appears in the **SQL** window only when the DB of the database project registered for linked analysis executes the SQL statement of the selected transaction.
- The **Copy bind variables** feature allows you to copy variables and paste them into the **SQL Text** tab of the **Correlated DB information** window. This feature is available when you select **Password**, enter the security key, and then display the value of the bind variable. For more information, see the following.
- For more information about the **Correlated DB information** window, see the following.
- The databases that support this feature are **PostgreSQL**, **Oracle**, **Oracle Pro**, and **MySQL** platforms.

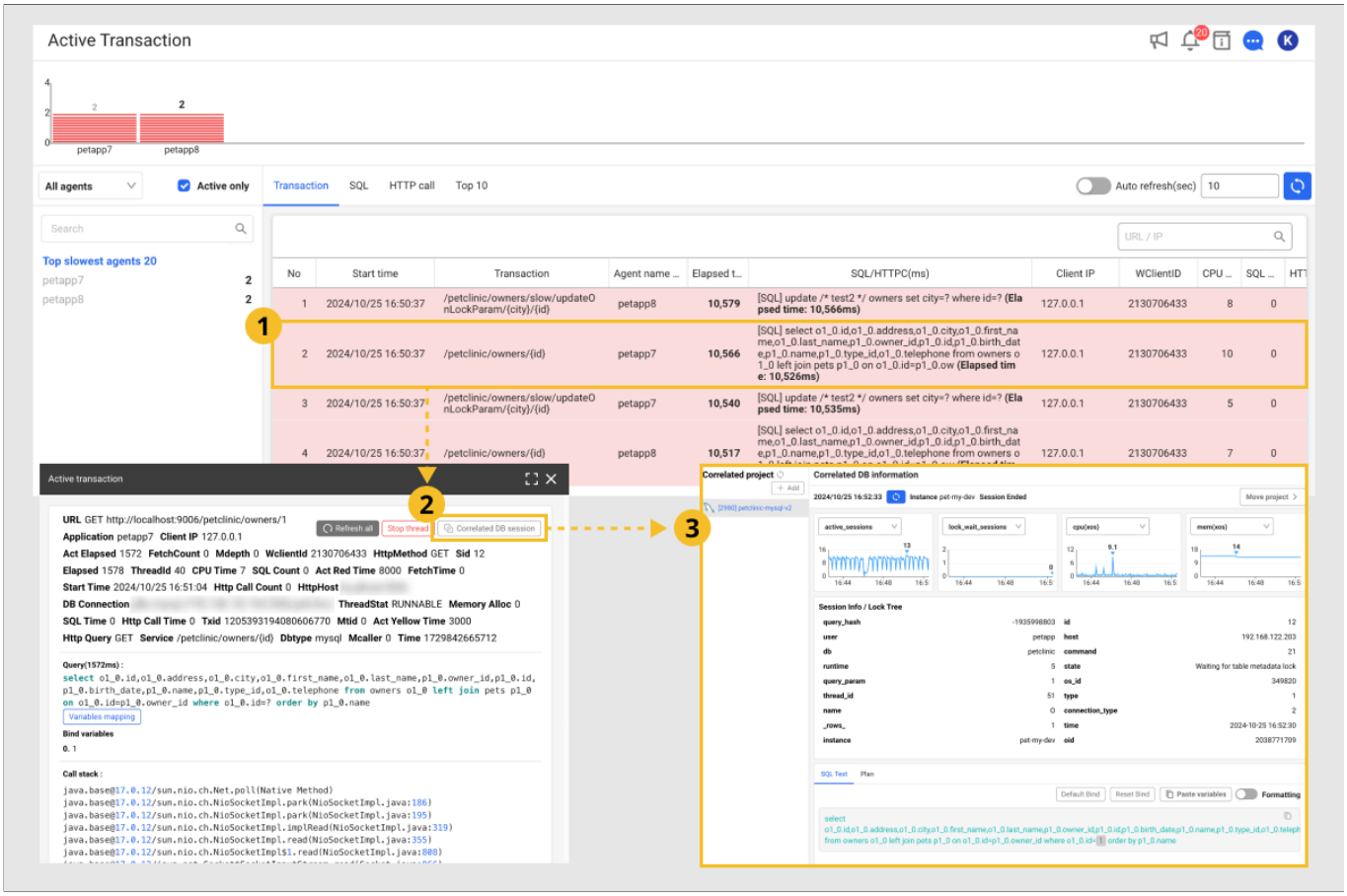
- On the **PostgreSQL** platform, the **PID** entry in the **Active sessions** table is the same as the **SID** entry.


SQL
Connection URL
CRUD
Select
owners
Query (SQL hash:
Correlated DB session
Password
SQL format
Statistics
SID
414200
Variables mapping
Bind variables
Copy bind variables
Bind variables : 1
Active sessions
Lock tree
Process Info
instance
pid
datname
cpuusage
client_addr
usename
petapp-pg
414200
petclinic
0.00%
192.168.122...
petapp
petapp-pg
414233
petclinic
0.00%
192.168.122...
petapp
petapp-pg
414285
petclinic
0.00%
192.168.122...
petapp
Correlation analysis
2024/10/24 13:39
Move project
Active Transcation
Transaction
/petclinic/owners/slow/updateOnLock
Transaction ID
4362591315698310000
httpURL
DB Connection
Elapsed time
12,950ms
Thread ID
49
DB Type
PostgreSQL
Sid
414200
HTTP method
GET
Thread Stat
RUNNABLE
Project code
2982
Project Name
Agent ID (oid)
-180752659
Agent name (oname)
petapp8

> In **the versions before Java Agent 2.2.40**, the **ThreadStat** values appear in numeric format on the **Active transaction** window. To display in text format, install Java Agent 2.2.40 or later.

Active Transaction

You can query the SQL performance of the database associated with active transactions. Go to **Dashboard** > **Active Transaction**.

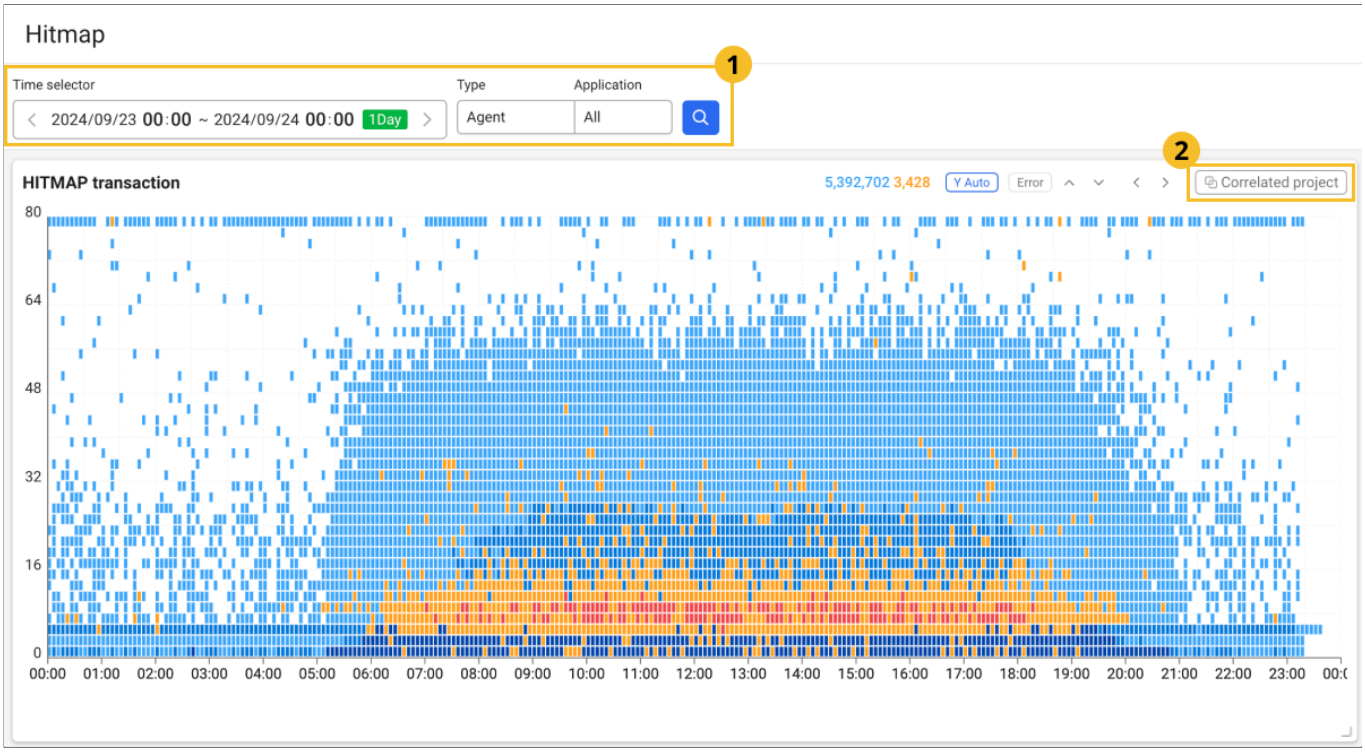


1. Select a desired transaction to view from the transaction list.
2. When the **Active transaction** window appears, select **Correlated DB session** on the upper right.
3. When the **Correlated DB information** window appears, check the related metrics.

- The **Correlated DB session** button appears in the **Active transaction** window only when the DB of the database project registered for linked analysis executes the SQL of the selected transaction.
- For more information about the **Correlated DB information** window, see the following.
- In **the versions before Java Agent 2.2.40**, the **ThreadStat** values appear in numeric format on the **Active transaction** window. To display in text format, install Java Agent 2.2.40 or later.
- The databases that support this feature are **PostgreSQL**, **Oracle**, **Oracle Pro**, and **MySQL** platforms.

Hitmap

You can also see performance metrics for the linked project while viewing transactions. This feature is useful when analyzing transactions in the APM project. When you found a slow SQL, it is difficult to determine exactly in which DB the problem occurred in the environment using multiple similar DBs.



1. Set the time and **Application** for viewing the transactions, and then select .
2. In the **HITMAP transaction** section, select **Correlated project** on the upper right.

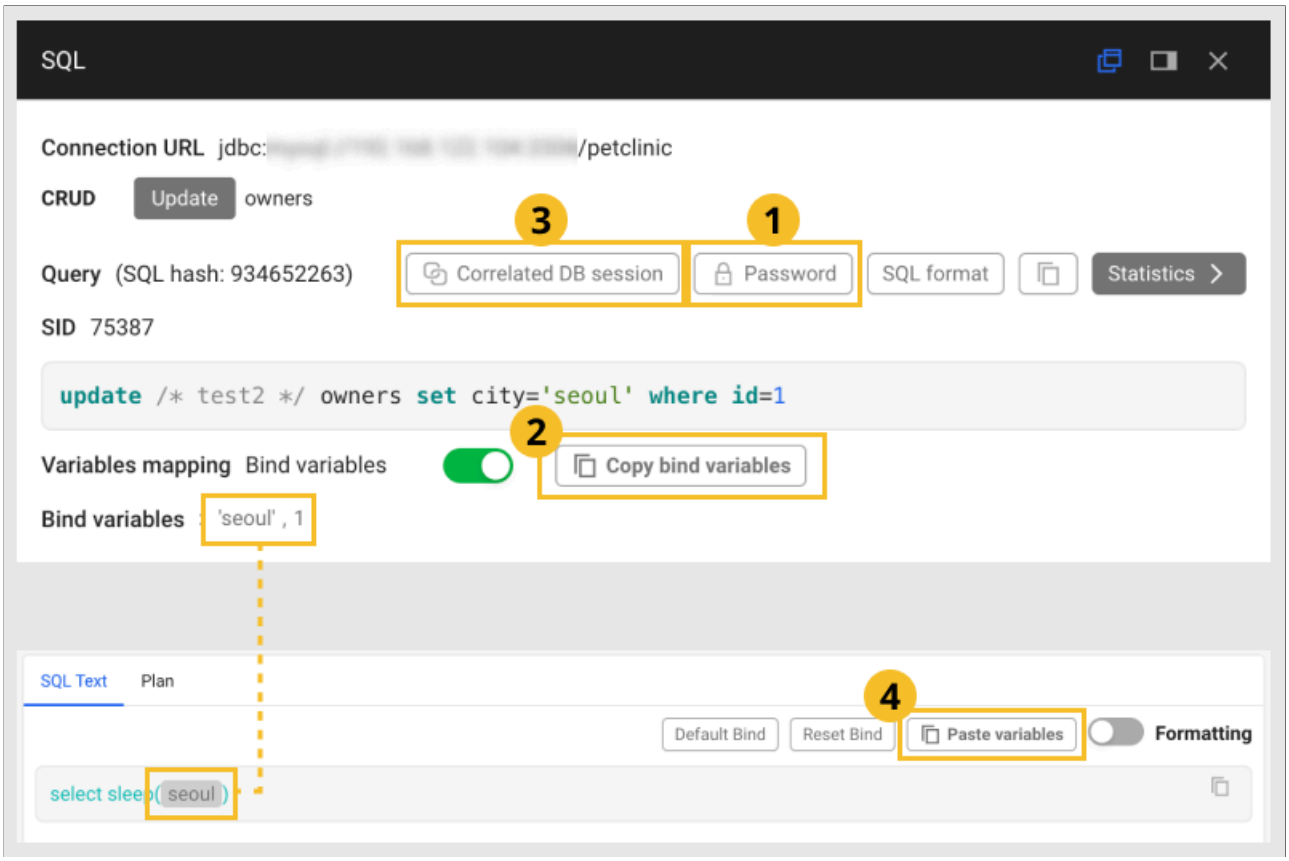
3. A new window appears where you can check the metrics for the correlated project.

Select the correlated project from the left list and then check the metrics in the **Instance dashboard** section.

> ⓘ For more information about the **Instance dashboard** window, see the following.

Copying/pasting the bind variable

You can copy the bind variable value displayed in the SQL step and then paste it into the **SQL Text** tab of the **Correlated DB information** window. Check the SQL step in the **Trace analysis** window.



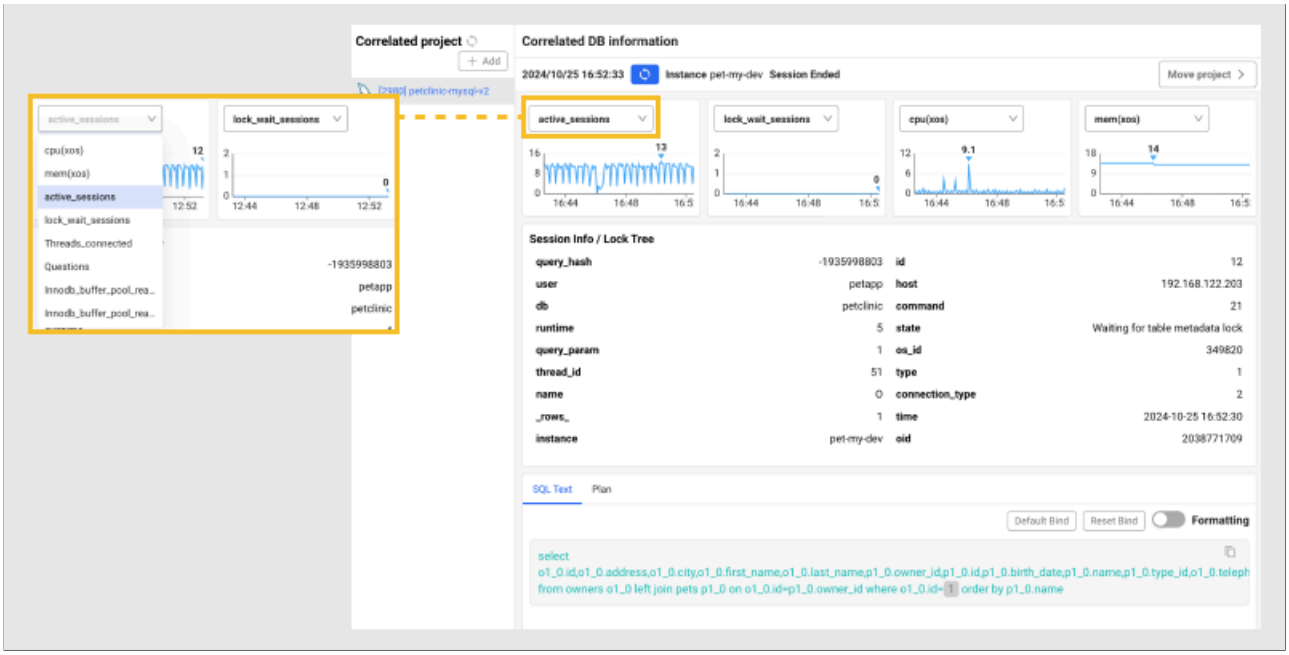
1. In the **SQL** window, select **Password** and then enter the security key.
2. Check the bind variable and then select **Copy bind variables**.
3. Select **Correlated DB session**.

4. When the **Correlated DB information** window appears, select **Paste variables** on the **SQL Text** tab.

> • For more information on how to view the SQL step in the **Correlated DB information** window, see the following.
> • For more information about the agent configuration related to the security key, see the following.

Correlated DB information

Correlated DB information allows you to check the SQL performance of the database associated with the current transaction in the application.



- The charts at the top of the screen provide charts for key metrics of the database. You can also select other metrics by selecting the checkbox.
- The Session Info / Lock Tree section provides various details for the database session.

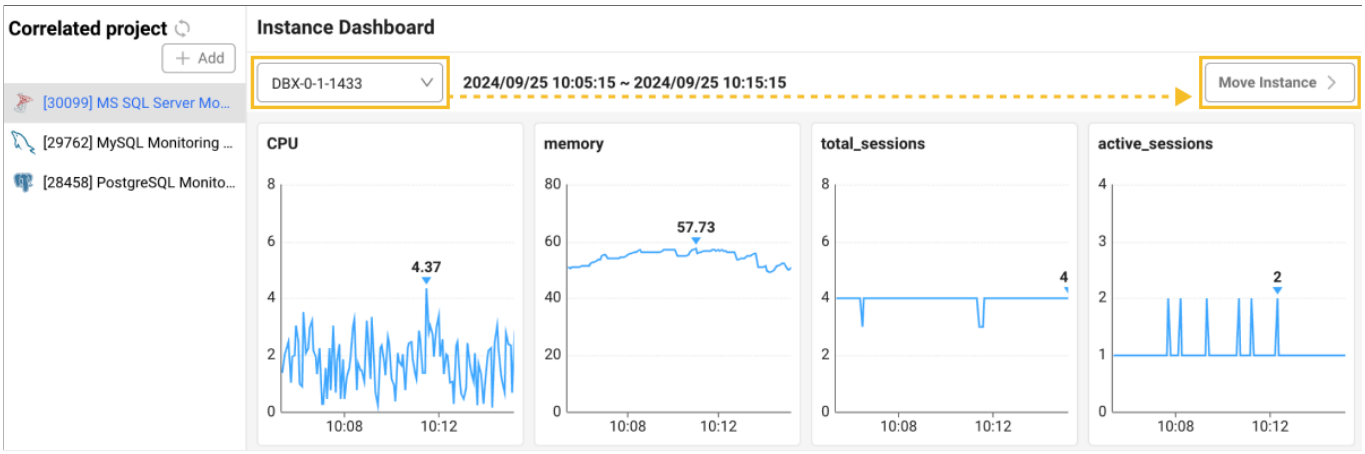
> Session Info / Lock Tree

- query_hash: Unique hash value for the SQL query in progress.
- user: User name connected to the database.
- db: Name of the connected database.
- runtime: Time (in milliseconds) that the session has been run.
- query_param: Parameter used in the query.
- thread_id: Unique identifier of the thread connected to this session.
- name: Name associated with the session.
- rows: Number of rows processed in the executed query.
- state: State of the session.
- id: Unique identifier of the current session.
- host: IP address of the server in which the database is running.
- command: It indicates the currently executing command or task.
- os_id: ID of the session used in the operating system.
- type: Type of the session.
- connection_type: Database connection type.
- time: Started time of the session.
- oid: Unique ID of the instance (agent) to which the session belongs.

- The SQL Text tab displays the actual content of the executed SQL query.
  - Default Bind: You can bind the part set as a variable in the query sentence to the default value.
  - Reset Bind: You can release the bound default value and check the variable.
  - **Paste variables**: You can paste the copied bind variable value. For more information, see the following.
  - Formatting: You can improve readability by applying indentation and formatting to a SQL query statement.
- Plan: Plan information can be checked. Enter **DB Name**, **User Name**, and **Password**, and then select .

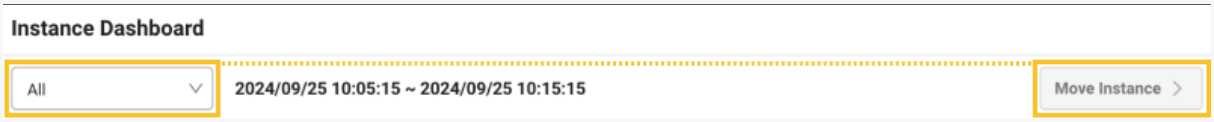
Instance dashboard

Instance dashboard allows you to view key performance metrics of the database associated with the transactions executed in the application.



To navigate to the dashboard menu for a selected project, select an instance and then click **Move instance** on the upper right. Go to **Dashboard** > **Monitoring a Database Instance**.

- The **HITMAP transaction** button in the **Correlated project** section appears only when there is an associated project. For more information about adding a correlated project, see the following.
- You can add a project by selecting **Correlated project** at the top of the left **Add** list.
- The **Move instance** button is disabled when **All** is selected.



- Depending on the database product type, you can navigate to the **Analysis** > **Counts Trend** menu.

Removing any linked project

1. Go to **Management** > **Correlated Project Management**.

2. To delete a project to unlink from the **Correlated project list** section, select on the utmost right of the list.
3. When the confirmation message appears, select **Delete**.

The deleted project is excluded from the **Correlated project list** section.

Log Monitoring

Logs are the files that have recorded the events and messages that occurred in applications and systems. It is important to check the logs to find anomalies for preventing system deterioration or detecting the failure causes.

Modern IT service building environments are changing to MSA or Kubernetes. As a result, the targets to be managed are increasing. In typical log monitorings, the `tail` command or editor is used for checking after connecting the servers. However, in this case, it is difficult to use simple methods such as accessing individual servers one by one. How can you view the logs from hundreds or thousands of servers in severe cases?

Through the WhaTap log monitoring service, you can easily manage numerous logs.

> ⓘ The Linux `tail` command is mainly used to check the logs whose content is added over time.

Key features

- **Central integrated management**

  WhaTap can manage a large number of logs from the center. It is very convenient because you can check the log details without accessing individual servers.

- **Collection of all logs**

  While collecting the logs selectively, important data may be missing. In general, WhaTap collects all logs. The collected logs can be checked in real time through the Live Tail menu.

- **Visibility**

  You can check the logs through various charts provided by WhaTap. Through this visibility, you can increase access to errors and issues and detect failures as soon as possible.

- **Flexible capacity management**

  While loading logs on individual servers, the logs may cause an excessive file system capacity being occupied. However, if the logs are collected through WhaTap, there is no need to maintain log files that occur on individual servers, and only the log maintenance period can be managed centrally.

- **Various analysis perspectives**

  To find and predict failures, you have to analyze logs from various perspectives. WhaTap can check the log count trends with specific tags or the logs filtered by specific tags. For frequently used patterns, you can save them as charts so that you can search them at any time.

- **Pattern notification**

  Once you have found a pattern of failures, you can set this as an alert to recognize or prevent problems as soon as possible. WhaTap provides Real-time Log Alert to receive notifications when specific keywords are included on an individual log basis and Composite Log Alert to receive notifications based on the count trend of logs containing specific tags.

Analyzing logs

Live Tail

Through Live Tail, the logs collected in real time can be viewed as they flow through the screen, just like using the `tail` command. For more information about Live Tail, see the following.

Log Trend

Through Log Trend, you can see all the collected logs or the log count trend with specific tags. If the log counts are closely related to the times of failure occurrence and resolution, the failure cause analysis and response can be accelerated through the log occurrence trend. For more information about the log trend, see the following.

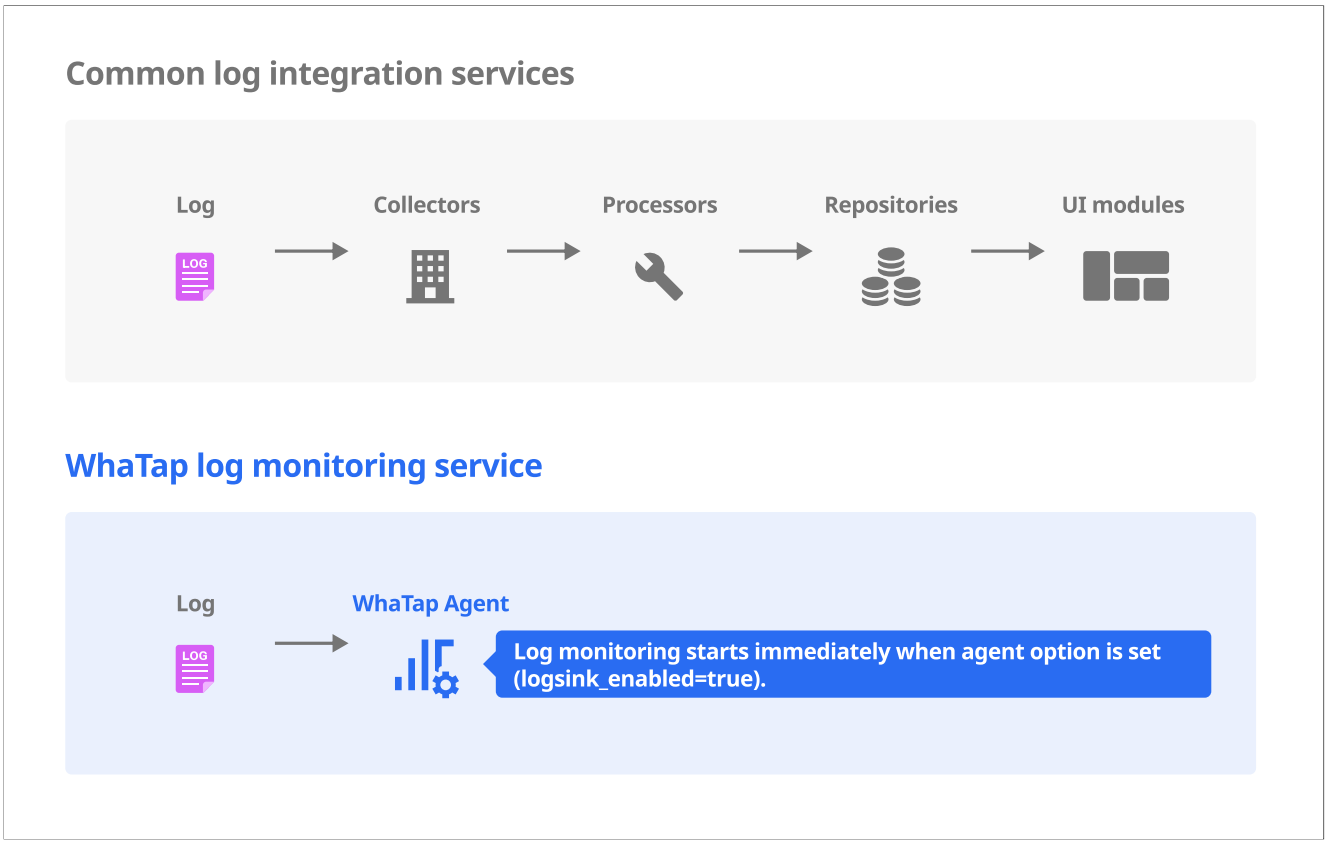
Log Search

Through Log Search, you can view all the collected logs or the logs with specific tags. The logs generated in a specific time period or specific server can be viewed based on tags. The Adjacent Log function that checks the logs that occurred before and after a selected log, can be used to see the situation in which a specific error or exception occurred. For more information about Log Search, see the following.

Applying The Log Monitoring

The following guides you to the basic application method for using the WhaTap log monitoring service. WhaTap log monitoring does not require configuring additional agents or applying a parser according to the logs. A simple configuration gets you started quickly.

Operating principle of log monitoring



A typical log integration service consists of the collector, processor, repository, and UI module. The process of building each module is cumbersome and additional costs are incurred because it requires step-by-step configuration and structuring.

WhaTap log monitoring is easy to apply. Because the previous monitoring agent acts as a collector, the log monitoring can be started simply by turning on the agent option.

- Java PHP Python Go Server Kubernetes

  - It directly collects the logs that are forwarded to the log library of your Java application rather than reading the output file.
  - Because the logs are collected directly, it does not cause file I/O so the performance impact on the system is very low.
  - You can check logs from the transaction traces by obtaining the connection traceability for transaction traces and logs.

  > **Java log library**
  >
  > Representative Java log libraries are Apache Log4j and Logback.

  > They are available in Java Agent 2.1.1 or later.

  - The log collection function has been added to the existing agent. The monitoring agent collects by using the output logs in addition to log files.

  > They are available in PHP Agent 2.3.2 or later.

  - The log collection function has been added to the existing agent. The monitoring agent collects by using the output logs in addition to log files.
  - If the transaction ID is output to the log, you can check logs from the transaction traces by obtaining the connection traceability for transaction traces and logs.

  > They are available in Python Agent 1.2.2 or later.

  - The log collection function has been added to the existing agent. The monitoring agent collects by using the output logs in addition to log files.
  - The log collection function has been added to the existing agent. The monitoring agent collects by using the output logs in addition to log files.

> They are available in Server Agent 2.1.2 or later.

- Logs can be collected in the Kubernetes container.
- You can collect logs from the applications in Kubernetes containers.

> They are available in Kubernetes Agent 1.1.35 or later.

Applying the log monitoring

The application method according to the used application is as follows. Before application of Log Monitoring, check the supported version.

1. Check the version of the supported agent and perform update.
2. Set the options for Log Monitoring.
3. **Enable** the log monitoring.

Java

Learn how to collect logs from your Java application.

PHP

Learn how to collect logs from your PHP application.

Node.js

Learn how to collect logs from your Node.js application.

Python

Learn how to collect logs from your Python application.

Go

Learn how to collect logs from your Go application.

Server

Learn how to collect logs from your server application.

Kubernetes

The following guides you through collecting logs from Kubernetes containers and the applications in Kubernetes containers.

Java

To collect logs from your Java application, complete the following 3 steps:

Agent Update

It is available from Java agent 2.1.1. For more information about update, see the following.

Checking the agent settings

Home > Select Project > **Log** > **Log Setting**

Select a project in the initial screen of the WhaTap monitoring service, and then select **Log** > **Log Setting** under Project Menu. Proceed with the process according to the instructions in the **Check agent settings** tab of the **Getting started with log monitoring** section.

1. Under Project Menu, select **Management** > **Agent CONFIG.**.
2. In **Write options**, select **Direct Input**. Add the `logsink_enabled=true` option to the input field as follows:
3. Restart the application to apply the log monitoring.

Key options

- **hooklog_enabled** Boolean

  Default `false`

  It enables the log monitoring by hooking the log library.

  > Before running the application, this option must have been enabled in whatap.conf to dynamically turn on or off the monitoring through the `logsink_enabled` setting. If the `logsink_enabled` option is set to `true` before running the application, log monitoring is possible without a separate setting.

> If there is a possibility to use the log monitoring, it is recommended to set this option in advance.

- **hooklog_custom_methods**

  Register a custom log. An arbitrary log framework content is delivered. Use this to trace logs of the log module created on the website.

  Java

```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

  Default `false`

  It turns on or off Log Monitoring.

> Before running the application, if in whatap.conf, the `hooklog_enabled` option is set, this option can dynamically turn on or off the log monitoring.

- **logsink_trace_enabled** Boolean

  Default `false`

  Set whether or not to expose the log tab of the transaction trace by inserting the transaction ID into the logs.

Enabling the log monitoring

Home > Select Project > Log > Log Setting

You can enable or disable the WhaTap log monitoring with the toggle button in the Activate log monitoring tab of the Starting the log monitoring section.

- If you turn on  , the log monitoring is enabled. You can try for free for 15 days from the activation date.
- If you turn off  , the log monitoring is disabled. Logs are no longer saved.

> **Role**
>
> After agent installation, you can enable the log monitoring settings only when you have the edit permission on the project. For more information about the roles, see the following.

PHP

To collect logs by your PHP application, check the following:

Agent Update

It is available from PHP agent 2.3.2. For more information about update, see the following.

Checking the agent settings

Home > Select Project > Log > Log Setting

Select a project in the initial screen of the WhaTap monitoring service, and then select Log > Log Setting under Project Menu. Proceed with the process according to the instructions in the Check Agent Settings tab of the Start Log Monitoring section.

Checking the configuration file path

Check the **configuration file path** of the file for log monitoring. You can check it in the CLI environment or on the web.

Checking in the CLI environment

```
$ php -i | grep 'Scan'
Scan this dir for additional .ini files => /etc/php8.0.d
```

> ⓘ If the path is "(None)," the configuration file path is /usr/whatap/php.

Checking on the web

If the PHP settings of the CLI and Apache environments are different, check the results of the `phpinfo()`

function on the web. .

Entering the command

Log monitoring starts when you enter the command including the configuration file path.

```
export LOGFILES=/some/path/file1,/some/other/file2
echo "whatap.logsink_enabled=true" | sudo tee -a [configuration file path]/whatap.ini
echo "whatap.logsink.files=$LOGFILES" | sudo tee -a [configuration file path]/whatap.ini
```

Enabling the log monitoring

Home > Select Project > Log > Log Setting

You can enable or disable the WhaTap log monitoring with the toggle button in the Activate log monitoring tab of the Starting the log monitoring section.

- If you turn on  , the log monitoring is enabled. You can try for free for 15 days from the activation date.
- If you turn off  , the log monitoring is disabled. Logs are no longer saved.

> **Role**
>
> After agent installation, you can enable the log monitoring settings only when you have the edit permission on the project. For more information about the roles, see the following.

Node.js

To collect logs by your Node.js application, check the following.

Agent Update

It is available from Node.js agent 0.5.1. For more information about the update, see the following.

Checking the agent settings

Home > Select Project > **Log** > **Log Setting**

Select a project in the initial screen of the WhaTap monitoring service, and then select **Log** > **Log Setting** under Project Menu. Proceed with the process according to the instructions in the **Check agent settings** tab of the **Getting started with log monitoring** section.

1. Under Project Menu, select **Management** > **Agent CONFIG.**.
2. In **Write options**, select **Direct Input**. Add the `logsink_enabled=true` option to the input field as follows:
3. Restart the application to apply the log monitoring.

Key options

- **logsink_enabled** Boolean

  Default `false`

  Set whether or not to enable the log monitoring.

- **logsink_trace_txid_enabled** Boolean

  Default `true`

  Set whether or not to expose the log tab of the transaction trace by inserting the transaction ID into the logs.

- **logsink_limit_content_enabled** Boolean

Default `true`

Set whether or not to limit the maximum size for log messages.

- **logsink_limit_content_length** Number

  Default `10,000`

  Set the maximum length of the log message. This value is applied only when the `logsink_limit_content_enabled` option is set to `true`.

Enabling the log monitoring

Home > Select Project > Log > Log Setting

You can enable or disable the WhaTap log monitoring with the toggle button in the Activate log monitoring tab of the Starting the log monitoring section.

- If you turn on  , the log monitoring is enabled. You can try for free for 15 days from the activation date.
- If you turn off  , the log monitoring is disabled. Logs are no longer saved.

> **Role**
>
> After agent installation, you can enable the log monitoring settings only when you have the edit permission on the project. For more information about the roles, see the following.

Python

To collect logs by your Python application, check the following.

Agent Update

It is available from Python agent 1.2.2. For more information about update, see the following.

Checking the agent settings

Home > Select Project > Log > Log Setting

Select a project in the initial screen of the WhaTap monitoring service, and then select Log > Log Setting under Project Menu. Proceed with the process according to the instructions in the Check Agent Settings tab of the Start Log Monitoring section.

Enabling the collection of logs

If you enter the command including the configuration file path (WHATAP_HOME), log collection starts.

```
export LOGFILES={Log file's full path},...
echo "logsink_enabled=true" | sudo tee -a {Configuration file path}/whatap.conf
echo "logsink.files=$LOGFILES" | sudo tee -a {Configuration file path}/whatap.conf
```

Linking the log and web transaction

Version 1.3.6 or later

Starting from the version 1.3.6, transactions and logs can be integrated using the following methods. Currently, WhaTap supports the Python's logging and loguru libraries. Configure whatap.conf according to your Python log libraries.

- logging module

logging

```
echo "trace_logging_enabled=true" | sudo tee -a {configuration file path}/whatap.conf
```

- loguru module

loguru

```
echo "trace_loguru_enabled=true" | sudo tee -a {configuration file path}/whatap.conf
```

Version 1.3.6 or earlier

The transaction ID `{txid}` is output to the log so that the logs generated by each transaction can be viewed separately. WhaTap Monitoring automatically injects `{txid}` into Python LogRecord so that `{txid}` can be output to the log file when setting the formatter.

settings.py

```
...
LOGGING = {
...
    'formatters': {
        ...
    },
    'handlers': {
        ...
    },
    'loggers': {
        ...
        '{logger_name}': {
            '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'][{Logger name}]['handlers'].append('whatap')
except:
    pass
...
```

Enabling the log monitoring

Home > Select Project > Log > Log Setting

You can enable or disable the WhaTap log monitoring with the toggle button in the Activate log monitoring tab of the Starting the log monitoring section.

- If you turn on , the log monitoring is enabled. You can try for free for 15 days from the activation date.
- If you turn off , the log monitoring is disabled. Logs are no longer saved.

> **Role**
>
> After agent installation, you can enable the log monitoring settings only when you have the edit permission on the project. For more information about the roles, see the following.

Go

To collect logs by your Go application, check the following.

Checking the agent settings

Home > Select Project > Log > Log Setting

Select a project in the initial screen of the WhaTap monitoring service, and then select Log > Log Setting under Project Menu. Proceed with the process according to the instructions in the Check Agent Settings tab of the Start Log Monitoring section.

Enabling the collection of logs

If you enter the following command, log collection is started immediately.

```
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
```

Enabling the log monitoring

Home > Select Project > Log > Log Setting

You can enable or disable the WhaTap log monitoring with the toggle button in the Activate log monitoring tab of the Starting the log monitoring section.

- If you turn on  , the log monitoring is enabled. You can try for free for 15 days from the activation date.
- If you turn off  , the log monitoring is disabled. Logs are no longer saved.

> **Role**
>
> After agent installation, you can enable the log monitoring settings only when you have the edit

permission on the project. For more information about the roles, see the following.

Server

To collect logs by your server application, check the following:

Agent Update

It is available from Server agent 2.1.2. For more information about update, see the following.

Checking the agent settings

Home > Select Project > **Log** > **Log Settin_**

Select a project in the initial screen of the WhaTap monitoring service, and then select **Log** > **Log Setting** under Project Menu. In the OS selection tab at the top of **Check Agent CONF.**, select a server OS (**Linux (shell)**, **Windows (Powershell)**). See **Basic Installation** or **Install with Category** for the process.

> ⓘ The following configuration requires restart of the agent.

Linux Shell

- Basic installation

  Linux Shell

  ```
  export LOGFILES=/some/path/file1,/some/other/file2
  echo "logsink.files=$LOGFILES" | sudo tee -a /usr/whatap/infra/conf/whatap.conf
  ```

- Install with Category

  Linux Shell

  ```
  1  cd /usr/whatap/infra
  ```

```
sudo mkdir extension

cat <<EOL | sudo tee extension/logsink.conf > /dev/null
[[inputs.logsink]]
 category = "serverlog"
 ## Whether to transfer the log generation statistics as separate data
 stats_enabled = true
 ## Log generation statistics category
 stats_category = "logsink_stats"
 ##  Set the log file name to be excluded if the log file path contains asterisk (*).
 excludeNames = [ ".gz",".zip" ]
 [[inputs.logsink.file]]
  ## When setting a log file, you can set the date pattern (strftime.org).
  path = "/some/path/%Y-%m-%d/.log"
  disabled = false
  encoding = "utf-8"

 [[inputs.logsink.file]]
  path = "/some/other/log"
  disabled = false
  encoding = "utf-8"

 [[inputs.logsink.file]]
  ## When searching for the keyword in the line-by-line log, merge it with the previous log.
  nowrap_keywords = ["Caused by:", "Test"]

 [[inputs.logsink.file]]
 ## When entering the project code and project license, the logs can be transferred to a desiged project in the same region.
 # pcode = Project code
 # license = "Project license"

 [[inputs.logsink.file]]
 # whatap_host_ip = "Collection server IP1/Collection server IP2"
 # whatap_host_port = 6600

 [[inputs.logsink.file]]
 path = "/tmp/mixed.log"
 disabled = false
```

```
  encoding = "utf-8"
  ## The [{"A":1,"B":2},{"C":3,"D":4},...] format can be divided into individual logs.{"A":1,"B":2} {"C":3,"D":4}
  multiline_pattern = "simple_json_array"
EOL

sudo service whatap-infra restart
```

Windows Powershell

- Basic installation

  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"
```

- Install with Category

  Windows Powershell

```
# The administrator role required
New-Item -type "Directory" -Path "C:\Program Files\WhatapInfra\extension"

$contentToAdd = @"
[[inputs.logsink]]
category = "serverlog"
## Whether to transfer the log generation statistics as separate data
stats_enabled = true
## Log generation statistics category
stats_category = "logsink_stats"
## Set the log file name to be excluded when the log file path contains asterisk (*).
excludeNames = [ ".gz",".zip" ]
[[inputs.logsink.file]]
## The date pattern (strftime.org) can be entered when setting the log file.
path = "c:\\whatap\\logs%Y-%m-%d_.log"
disabled = false
encoding = "euc-kr"

```

```
[[inputs.logsink.file]]
## When searching for the keyword in the line-by-line logs, it merges with the previous log.
nowrap_keywords = ["Caused by:", "Test"]

[[inputs.logsink.file]]
## When entering the project code and project license, logs can be transferred to a desired project in the same region.
# pcode = Project code
# license = "Project license"

[[inputs.logsink.file]]
# whatap_host_ip = "Collection server IP1/Collection server IP2"
# whatap_host_port = 6600

[[inputs.logsink.file]]
path = "/tmp/mixed.log"
disabled = false
encoding = "utf-8"
## The [{"A":1,"B":2},{"C":3,"D":4},...] format can be divided into individual logs.{"A":1,"B":2} {"C":3,"D":4}
multiline_pattern = "simple_json_array"

"@

New-Item -path "C:\Program Files\WhatapInfra\extension" -name "logsink.conf" -type "file" -value $contentToAdd -Force

Restart-Service "Whatap Infra"
```

Configuring options

- `stats_enabled` : Set whether or not to collect the collection status data. The default value is `false` . The value must be set to `true` . If set to `true` , statistical data is generated in the category (`logsink_stats`) set in the following `stats_category` .
- `stats_category` : Set the metrics category to store the collection status data in. The value must be set to `logsink_stats` . Statistical data fields are as follows:
  - `file`

  - checkInterval
  - encoding
  - filepos
  - checkedLocalTime
  - lastupdatedLocalTime
  - fileSize
  - error
  - firstCheck
  - transferBytes
- excludeNames : By including asterisks (*) in the log file path, you can set the file name to exclude non log files. You can set multiple values using comma (,) as a separator.

  Example

  ```
  excludeNames = [ ".gz",".zip" ]
  ```

- nowrap_keywords : While searching logs by line, if any keyword is detected by a specified option value, it is merged with the previous log.
- pcode : Enter the project code of the same region to which you want to send the logs.
- license : Enter the license code of the project in the same region to which you want to send the logs.

> ⓘ In case of the pcode and license options, it is supported from Server Agent 2.7.4 or later.

Configuring the log options for Windows events

When collecting Windows event logs, you can set the options as follows:

```
# The administrator role required
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, ForwardedEvents
      file = "Application"
      #1: Information, 2: Warning 3: Error 4: Audit Success 5 Audit Fail
      #event_type =
      #event id
      #event_id =
      #event source name
      #source_name = ""
      #When entering the project code and project license, logs can be transferred to the desired project in the same region.
      #pcode = Project code
      #license = "Project license"
      #whatap_host_ip = "Collection server IP1/Collection server IP2"
      #whatap_host_port = 6600

"@

New-Item -path "C:\Program Files\WhatapInfra\extension" -name "win_eventlog.conf" -type "file" -value $contentToAdd -Force

Restart-Service "Whatap Infra"
```

- It is required to specify the `category`.

  e.g. `win_event_log`

- It is required to specify the statistics category (`stats_category`).

  e.g. `win_event_log_stats`

- The statistics category On/Off (`stats_enabled`) is required.

  e.g. `true` or `false`

- The collection On/Off (`enabled`) is required.

e.g. `true` or `false`

- The file-based collection On/Off (`enabled`) is required.

  e.g. `true` or `false`

- The file (`file`) is required.

  e.g. `Application` , `Security` , `Setup` , `System` , `Forwarded`

- The event type (`event_type`) is optional.

  e.g. `1` , `2` , `3` , `4` , `5`

> **Event type**
>
> 1. Information
> 2. Warning
> 3. Error
> 4. Audit Success
> 5. Audit Fail

- The event ID (`event_id`) is optional.
- The event source name (`source_name`) is optional.
- Project code (`pcode`) optional
- Project license (`license`) optional

> - For Windows event log collection, the Server Agent 2.5.2 or later is supported.
> - Among the Windows event log options, `pcode` and `license` are supported in the Server Agent 2.7.4 or later.

Specifying and separating multiple files

Specifying multiple files

To categorize multiple log files under the same category, add each log file path into the `[[inputs.logsink.file]]` entry and specify the file path by using the `path` setting. Additionally, you can configure additional settings, such as log generation statistics and options such as `category`, `stats_enabled`, and `stats_category`. The example is where the /root/test1 and /root/test2 files are assigned the same category (`serverlog`).

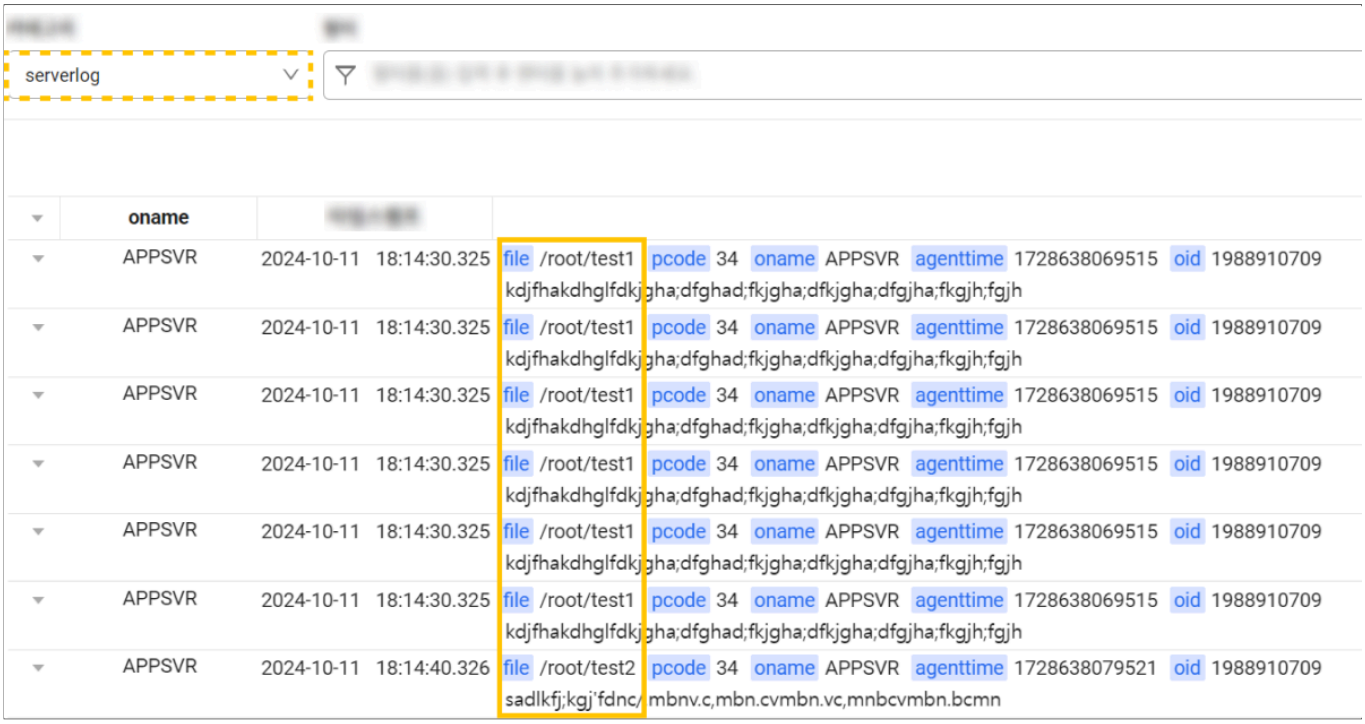
```
[[inputs.logsink]]
 category = "serverlog"
 stats_enabled = true
 stats_category = "logsink_stats"
 excludeNames = [ ".gz",".zip" ]

[[inputs.logsink.file]]
 path = "/root/test1"
 disabled = false
 encoding = "utf-8"

[[inputs.logsink.file]]
 path = "/root/test2"
 disabled = false
 encoding = "utf-8"
```

> ⓘ For information about options such as `category`, `stats_enabled`, `stats_category`, and `nowrap_keywords`, see the guidance above.

In this example, you can see the following in **Live Tail**.



File separation

To specify different categories for log files, create separate logsink{**number**}.conf files and then set different `category` values in each file. The following example shows how to set different log categories through logsink.conf and logsink**1**.conf.

logsink.conf

```
# cat logsink.conf
[[inputs.logsink]]
 category = "serverlog1"
 stats_enabled = true
 stats_category = "logsink_stats"
 excludeNames = [ ".gz",".zip" ]

[[inputs.logsink.file]]
 path = "/root/test1"
 disabled = false
 encoding = "utf-8"
```

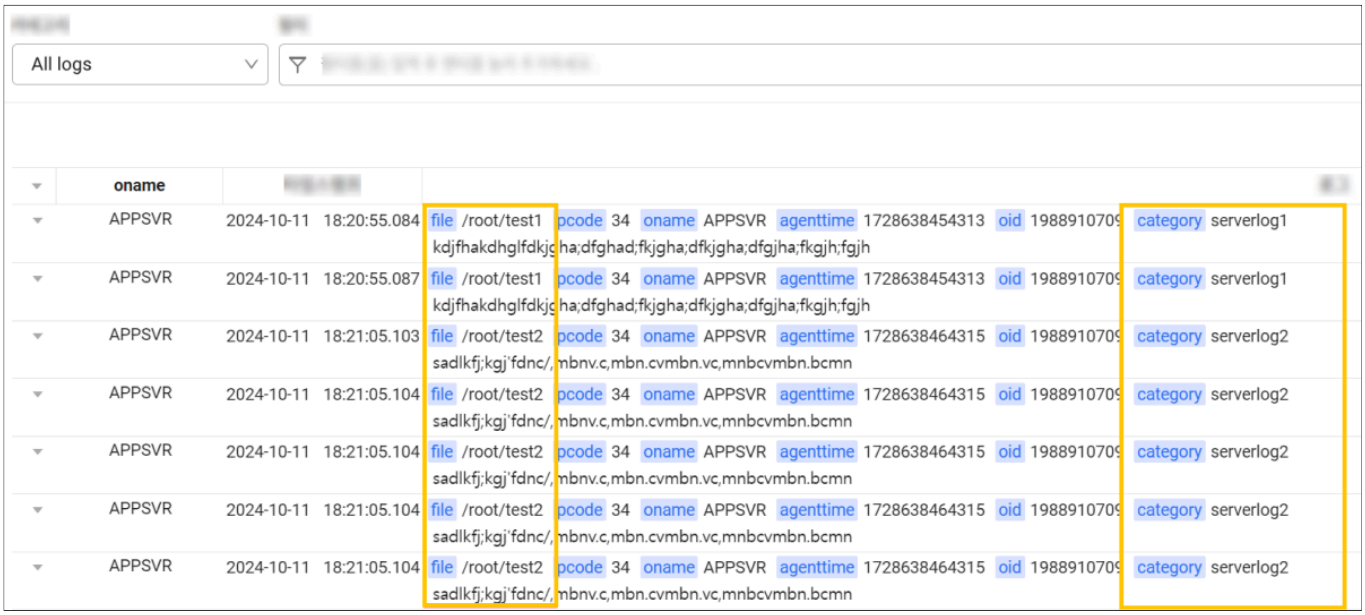
logsink1.conf

```
#cat logsink1.conf
[[inputs.logsink]]
 category = "serverlog2"
 stats_enabled = true
 stats_category = "logsink_stats"
 excludeNames = [ ".gz",".zip" ]

[[inputs.logsink.file]]
 path = "/root/test2"
 disabled = false
 encoding = "utf-8"
```

ⓘ For information about options such as `category`, `stats_enabled`, `stats_category`, and `nowrap_keywords`, see the guidance above.

In this example, you can see the following in **Live Tail**.



Enabling the log monitoring

Home > Select Project > **Log** > **Log Setting**

You can enable or disable the WhaTap log monitoring with the toggle button in the **Activate log monitoring** tab of the **Starting the log monitoring** section.

- If you turn on  , the log monitoring is enabled. You can try for free for 15 days from the activation date.
- If you turn off  , the log monitoring is disabled. Logs are no longer saved.

> **Role**
>
> After agent installation, you can enable the log monitoring settings only when you have the edit permission on the project. For more information about the roles, see the following.

Kubernetes

To collect logs from Kubernetes containers and the applications in Kubernetes containers, check the following.

Agent Update

It is available from Kubernetes agent 1.1.35. For more information about update, see the following.

Checking the agent settings

Home > Select Project > Log > Log Setting

Select a project in the initial screen of the WhaTap monitoring service, and then select Log > Log Setting under Project Menu. Proceed with the process according to the instructions in the Check Agent CONF. tab of the Set up the agent and enable log monitoring section.

Enabling the container log collection

To collect logs in Kubernetes containers, select Apply log settings in the Check agent configuration tab.

Enabling application log collection in the container

It is available from Java 2.1.1 and Python 1.2.2. To collect logs of the applications running on Kubernetes containers, see the following.

1. On the lower part of Check agent settings, select 1 Agent CONFIG..
2. Go to Application Agent CONFIG., and then select Direct Input in the 2 Write Options window.
3. Enter the key, `logsink_enabled` and the value, `true` of the following agent configuration command in the 3 field.

```
logsink_enabled=true
```

4. Select 4 Apply. You can collect application logs in the Kubernetes containers.

> ⓘ
> - For more information about the collection of Java application logs, see the following.
> - For more information about the collection of Python application logs, see the following.
> - For more information about the collection of Go application logs, see the following.

Enabling the log monitoring

Home > Select Project > Log > Log Setting

You can enable or disable the WhaTap log monitoring with the toggle button in the Activate log monitoring tab of the Starting the log monitoring section.

- If you turn on, the log monitoring is enabled. You can try for free for 15 days from the activation date.
- If you turn off, the log monitoring is disabled. Logs are no longer saved.

> ⓘ **Role**
>
> After agent installation, you can enable the log monitoring settings only when you have the edit permission on the project. For more information about the roles, see the following.

Kubernetes log monitoring category guide

You can check various logs related to Kubernetes. The monitoring categories provided by WhaTap Kubernetes are as follows. Depending on the settings, duplicate log contents may be stored. Be sure to check for duplicates.

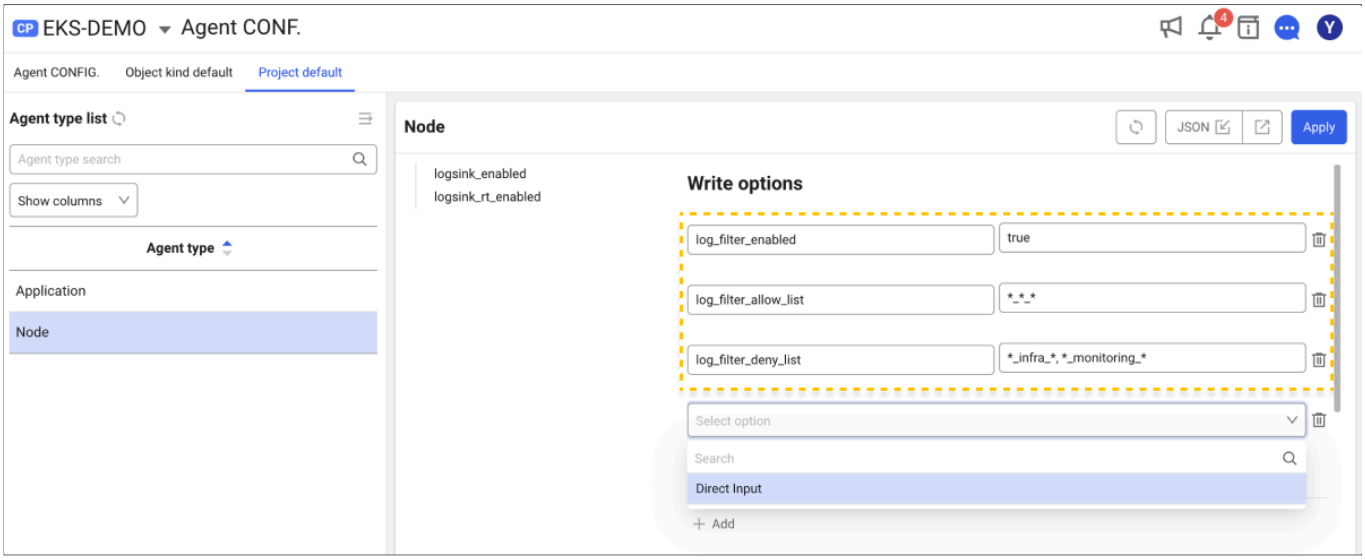
Category	Description
#K8sEvent	• Logs that store the events that occur in Kubernetes • Generation by default regardless of user settings

Category	Description
`#WhatapEvent`	• Logs that store the events that occur by WhaTap event settings • Generation by default regardless of user settings
`containerStdout`	• Container Standard Out logs • Generation by user settings • When `logsink_enabled=true` setting is added to the node agent
`AppLog`	• Application logs in the container • Generation by user settings • When `logsink_enabled=true` setting is added to the application agent

Setting the containerStdOut log filter

Home > Select Project > **Management** > **Agent CONF.**

The following guides you to filtering options to select information needed in the Kubernetes logs. You can set which logs to allow or exclude through filtering. In **Agent CONF.**, select **Direct Input** in the **Write options** manner.



Log naming conventions

The Kubernetes containerStdOut logs are generated under the /var/log/containers/ path on the node. They keep the following naming conventions.

```
<podName>_<namespace>_<containerName>-<containerId>.log
```

Log filtering options

- **log_filter_enabled** bool

  Default `false`

  It sets whether to use log filters.

- **log_filter_allow_list** list

  It sets the list of logs to allow. It indicates the items to be included for filtering.

- **log_filter_deny_list** list

  It sets the list of logs to exclude. It indicates the items to be excluded for filtering.

Log filtering operation

Log filters work only when the `log_filter_enabled` option is active. You can selectively collect logs via `log_filter_allow_list` and `log_filter_deny_list` when the value of the option is set to `true`.

Log filters work on a blacklist basis. For example, if you enable `log_filter_enabled` and do not set the items to be added to the `log_filter_allow_list` option, the log collection for all containers is blocked by default.

Additionally, in the log filters, deny rules take precedence over allow rules. If the same items are set in both allow and deny rules, the deny rule takes precedence and logs are not collected.

Log filter example

Collecting logs for a single namespace

- Allow **log_filter_allow_list** `*_infra_*`

All containers that exist in the `infra` namespace are targeted for collection.

Collecting multiple namespace logs

- Allow **log_filter_allow_list** `*_infra_*, *_monitoring_*`

All containers that exist in the `infra` and `monitoring` namespaces are specified as collection targets.

Pods containing specific terms are excluded.

- Allow **log_filter_allow_list** `*_*_*`
- Deny **log_filter_deny_list** `*prod*_*_*`

All containers are targeted for collection, but the logs that contain `prod` in `podName` are excluded.

Pods containing specific terms are collected.

- Allow **log_filter_allow_list** `*prod*_*_*`

Only the logs that contain `prod` in `podName` are collected.

Multi-namespace logs are collected and Pods containing specific terms are excluded.

- Allow **log_filter_allow_list** `*_infra_*, *_monitoring_*`
- Deny **log_filter_deny_list** `*prod*_*_*`

The logs for the `infra` and `monitoring` namespaces are collected, and the logs that contain `prod` in `podName` are excluded.

The container logs that start with a specific term in the container name are excluded.

- Allow **log_filter_allow_list** `*_*_db-*`

The container logs whose `containerName` starts with `db` in all namespaces are excluded.

Live Tail

> ⓘ Without the **Log Read** role, you cannot enter the menu.

Home > Select Project > **Log** > **Live Tail**

In the **Live Tail** menu, you can directly check the log data stream on the monitoring screen without connecting to the server console. Even with large volumes of complex logs, you can filter and quickly identify them using the **Filter** and **Highlight** features. The log data retrieval cycle is 2 seconds.

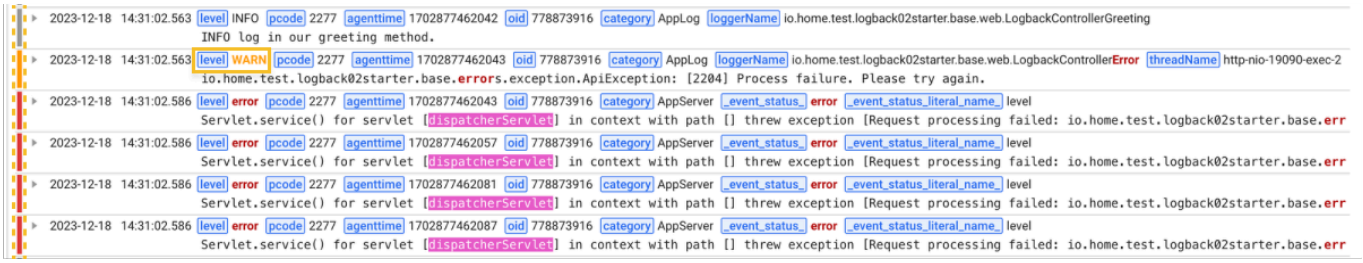
The major terms are as follows:

- **Category**: Unit for log collection and inquiry
- **Content**: Log messages
- **Search Key**: It is generated through the log parser settings.
- **Tag**: Key to search the collected logs

> ⓘ You can change the column width by dragging an edge of the log table column.

Agent option

If the agent option is set, the log levels are collected and the colors for each log level appear as follows:



> ⓘ **Setting the agent options**

- The agent options are as follows:

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

  - In Java agent 2.2.22 or later, it can be used when setting log4j-2.17 or logback-1.2.8 in the weaving setting. It is required to restart the agent.
  - The log level is determined based on the `level` and `type` among the parsed keywords. When there are keys parsed as `level` and `type`, and the parsed values have FATAL, CRITICAL, ERROR, WARN, WARNING, and INFO, the log level color appears.

1 Filter area

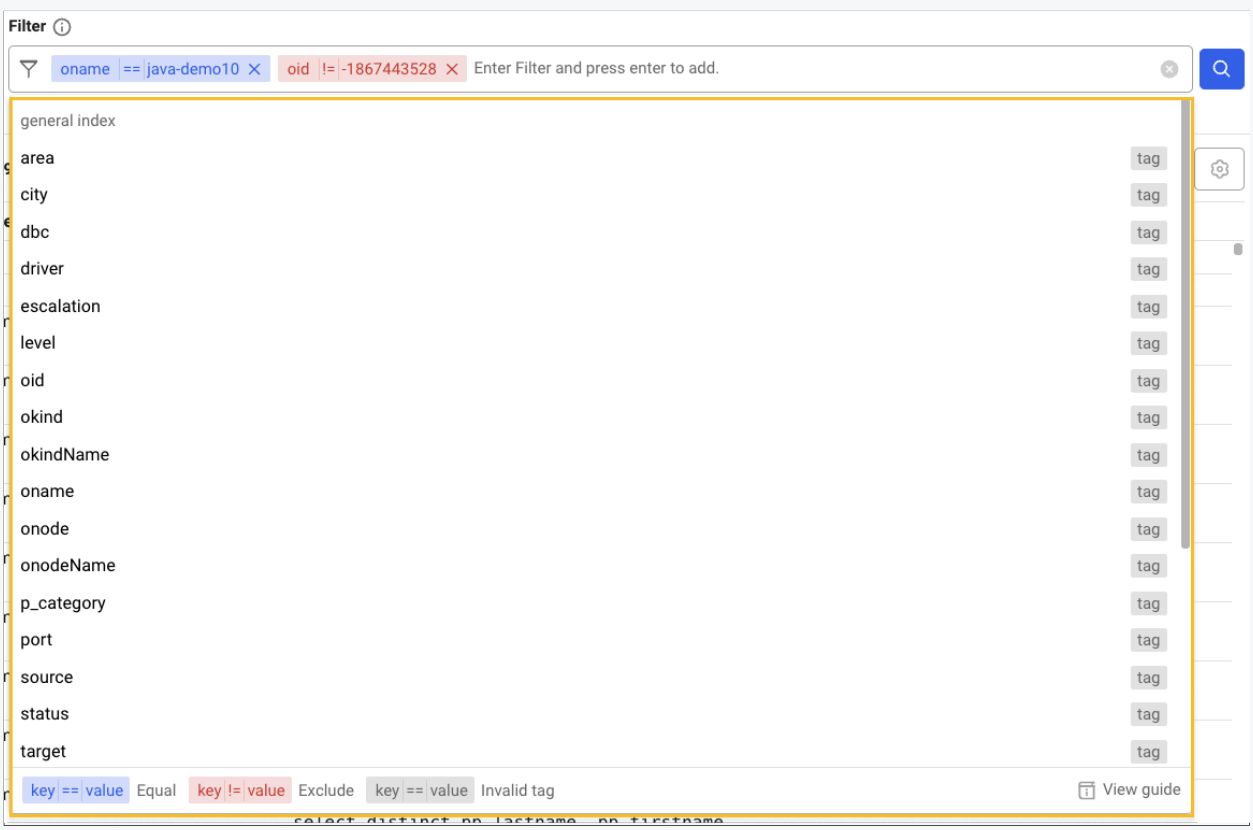
Apply Filter

When a filter is applied, the logs that match the specified conditions are filtered. You can enter multiple filters. OR (`||`) if the filters have the same tags, otherwise AND (`&&`).

You can specify a filter by directly entering a value in the input field or clicking the **Filter** field. Filter tags are entered in the following order: `Search key`, `Operator`, and `Search value`. If you select **Search**, the data with the filter applied can be viewed log list area.

Guide UI

The **Guide UI** is provided under the input field as follows: You can close the **Guide UI** via the ESC key when your mouse cursor is over the input field or filter tag.



Entry of the search key, operator, and search value

- Upon entry of the `search key`, recommended values are provided by distinguishing for general index, reserved word index, and numeric index.
- When entering an `operator`, for a general index search key, `==` and `!=` options are provided at the bottom. In case of the numeric index, the `>`, `<`, `<=`, `>=`, `==`, and `!=` options are provided.
- Upon entry of the `search value` for matched search (`>`, `<`, `<=`, `>=`, `==`), it is highlighted in blue, In case of excluded search (`!=`), it is highlighted in red. Upon entry of the `search value`, you can search using the case-sensitive option.

- If the filter tags exceed 2 lines, you can fold it by selecting the ^ **Fold** icon.
- After entering a filter tag, you can use the Shift and Tab keys simultaneously in the input field to go to the previous filter tag.

- After entering a filter tag, you can use the Tab key in the input field to go to the next filter tag.

Adding filter tags

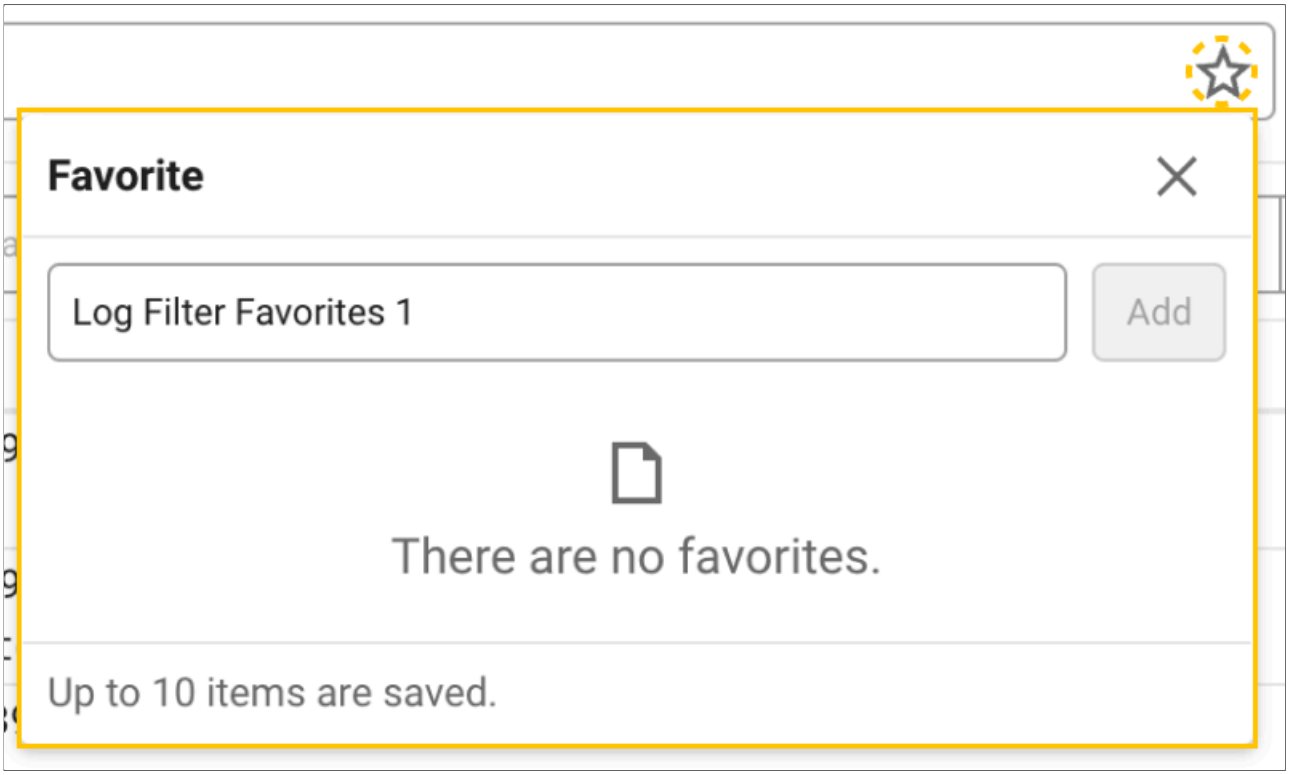
To add a filter tag, type the desired text in the input field and press **Enter** or **Tab**.
You can also select a recommended value from the **Guide UI** below the input field by clicking it or navigating with the arrow keys.

Deleting filter tags

To remove a single tag, press **Backspace** or click the **X** icon on the tag. To clear all tags at once, click the **X** icon located on the right side of the input field.

Filter favorites

The filter **favorites** feature is provided. You can select the ☆ **Favorites** icon on the right of the input window to **add**, **delete**, and select desired filter search conditions in the favorites. **Favorites** can store up to 50 cases.



Exceptional cases for application of filters

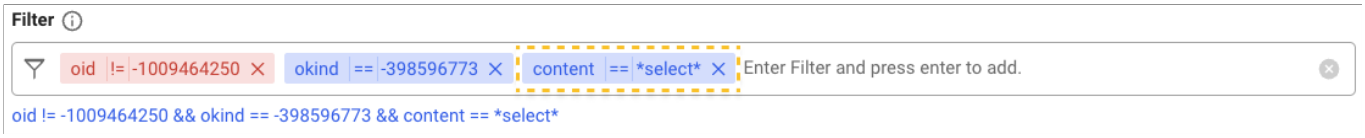
- In the tag with a numeric index (`search key` that ends with `.n`), only numbers can be input for the `search value` .
- Duplicate 'search key' and 'search value' cannot be entered.
- Searching is not possible if there is a tag whose `search key` or `search value` is missing. Invalid tags appear in gray.

> You cannot enter a `category` with the **Live Tail** `search key` .
> The expression under the entered filter value is a preview of the filter expression for querying the log data.

Applying non-parsing keyword filters

You can search the logs containing keywords that have not been parsed, or for which no index has been created.

In this case, all logs in the specified range are fully scanned. Therefore, the search speed may be slower compared to the indexed key. For standardized log data, it is recommended to search using the index key through **Log parser setting**.

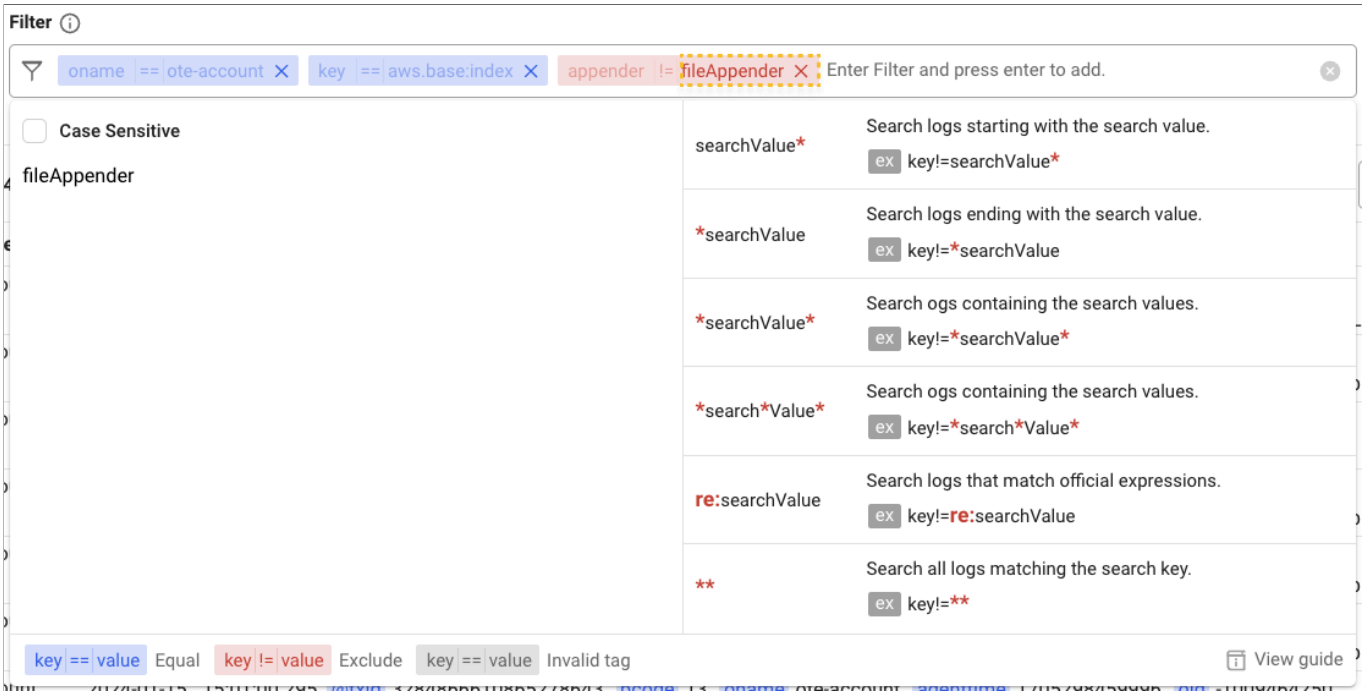


1. Select **Category**.
2. In the **Filter** field, enter a desired keyword for a search after leaving spaces on the `content` basis.
   e.g. `content *select*`
3. Click **Search** to view the logs.

> - In case of **Live Tail**, all logs can be viewed. Thus, the category is not required to be specified.
> - For more information about the parser configuration, see the following.

Modifying the filter

Enter a value in the filter and then click the specified value to modify it.



- You can modify the filter by entering text again in the input field.
- You can modify the recommended values by selecting them in the **Guide UI** below the input field.

Search Key

In the following image, the blue boxed part is the search key for parsing. In **Log Configuration**, you can register and set the parsing logic in **Log parser setting** tab with the search key.

Search key

Filter input syntax

A tag consists of a search key and a search value. In the following example, the search key is `exception` and the search value is `UnknownHostException`. The example searches for the log data that includes an exception (`UnknownHostException`) that is generated when the server cannot connect the host because the IP address and domain address do not match among the collected log data.



Search key types

Search key types	Search key format	Meaning	Example for Search key and search value	Seach example
String keyword	keyword	File Name	- Key: fileName - Value: /data/whatap/logs/yard.log	fileName:/data/whatap/logs/yard.log
Numeric keyword	keyword.n	Response Time	- Key: response_time.n - Value: 2945	response_time.n>=2945
Reserved keyword (predefined keyword)	@keyword	Transaction ID	- Key: @txid - Value: 85459614215434144	-
Log body's keyword	content	Log body	- Key: content - Value: user input	content: `*ERROR*`

Content search key

- The content search key searches the bodies of unindexed logs. For example, if you enter `content *ERROR*` , the logs containing `ERROR` are searched among the log bodies.
- If you do not know which keywords need to be used for indexing, use the content search keys to identify the logs containing problematic keywords. Afterwards, you can improve the search performance by setting a parser with the keyword through the log parser settings in **Log Configuration** and creating indexes.

Common syntaxes

Syntax type	Description	Example
`==searchValue`	Searches for the logs that match the search value.	`exception==RuntimeExceptionexception`

Syntax type	Description	Example
`!=searchValue`	Searches for the logs that exclude the search value.	`exception!=RuntimeException`
`*searchValue`	Searches for the logs ending with the search value.	`word==*hello`
`searchValue*`	Searches for the logs starting with the search value.	`word==hello*`
`*searchValue*`	Searches for the logs that contain the search value in the middle.	`word==*hello*`
`*search*Value*`	Searches for the logs that contain the search value.	`word==*he*llo*`
`re:{regex}`	Searches for the logs that match a regular expression.	`caller==re:^i\.w\.a\.w\.s\.v\.r\.`
`**`	Searches all logs that match the search key.	

Syntax in case the search key has a numeric value (keyword.n)

The following syntax is supported only when the search key is in `keyword.n` format.

- Search values must only be numbers.
- No prefix is attached to the value of the `.n` keyword. All keywords other than `.n` must be attached with a prefix.

  For example, `+>searchValue` is invalid.

Syntax type	Description	Example
`>searchValue`	Searches for the logs containing values greater than the search value.	`response_time.n>3000`

Syntax type	Description	Example
>=searchValue	Searches for the logs containing values greater than or equal to the search value.	response_time.n>=3000
==searchValue	Searches for the logs containing the same search value.	response_time.n==3000
!=searchValue	Searches for the logs that contain values different from the search value.	response_time.n!=3000
<searchValue	Searches for the logs containing values less than the search value.	response_time.n<3000
<=searchValue	Searches for the logs containing values less than or equal to the search value.	response_time.n<=3000

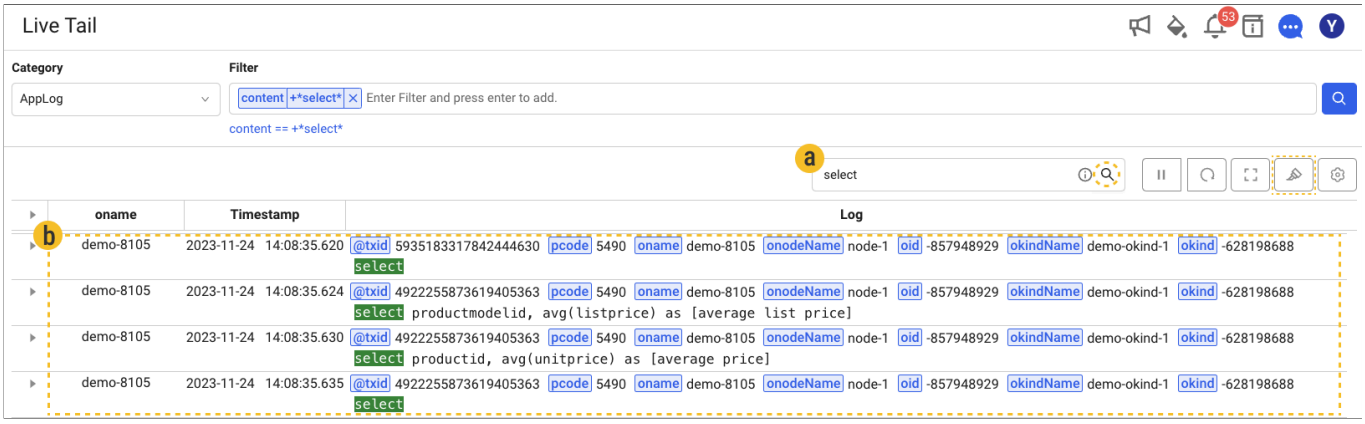
Viewing personal information

If you have the **log personal information query** role, you can display de-identified personal information by enabling the **Display personal information** toggle button.

- For more information about the **log personal information query** role, see the following.
- For more information about **De-identification of personal information**, see the following.

2 Content highlight area

The highlight function allows you to easily identify desired keywords in log content. You can filter with single or multiple keywords.



1. Enter the keyword you want to highlight in the keyword input field.
   - Example: `select`
2. The searched keyword will be highlighted in the Content of the log list.
   - If you click the **Log full screen** icon, you can view **Log** and **Timestamp** in fullscreen.
   - You can navigate between highlighted keywords using shortcuts.
     - `Enter` : Move to the next row
     - `Shift+Enter` : Move to the previous row
     - ∧ / ∨ icons: Move between rows
     - `Cmd` + `F` (Mac) / `Ctrl` + `F` (Windows): Search keywords

Multi-keyword condition

When highlighting with multiple keywords, write as follows:

String to input	Description	Result
`a b c`	Separate each keyword with a space.	a, b, c
`"Whatap is good."`	To include spaces in keywords, wrap them in `''` or `""`.	Whatap is good.
`"Whatap\\ is good."`	If `\` is included in the keyword surrounded by "", enter it as `\\`.	Whatap\ is good.

Setting the highlight color

By selecting **Highlight**, you can set the keyword and its color for highlighting.

- In addition, enter the keyword to set the color for.
- If you click Color on the left of the input field, available colors appear.
- Basically, different highlighting by the level (WARN, ERROR, FATAL) is applied.
- The settings are saved in **Project Rules**.

Table settings

Column settings

Right of the 2 area, select **Column settings**. You can add columns or set order.

- **Adding columns**

  You can add columns to the table by selecting tags. If you unselect the **log** column, you cannot see **Log view detail setting**. Be sure to select at least one column.

- **Setting the column order**

  When you add a column, the column is added to **Display columns**. Drag a desired column to change the order of the columns.

Log view detail setting

On the right of 2, select **Log view detail setting**. **content** and **Tag** are all checked and two items are also displayed. Be sure to select any of **content** and **Tag**.

Unchecked items are not displayed in the table. If you deselect the **Tag** as follows, the log's **Tag** does not appear in the table.

Log table tag invisible

When you add tags in the **Tag management** list, the tags in the log are listed in the order you added. You can change the order of the tags by dragging them. f you disable an added tag, the disabled tag does not appear.

- **Column settings** and **Log view detail setting** can be used in **Live Tail**, **Log Search**, and **Log Trend**.
- In the same project, the **Live Tail**, **Log Search**, and **Log Trend** menus share **Column settings** and **Log view detail setting**.

Log Explorer

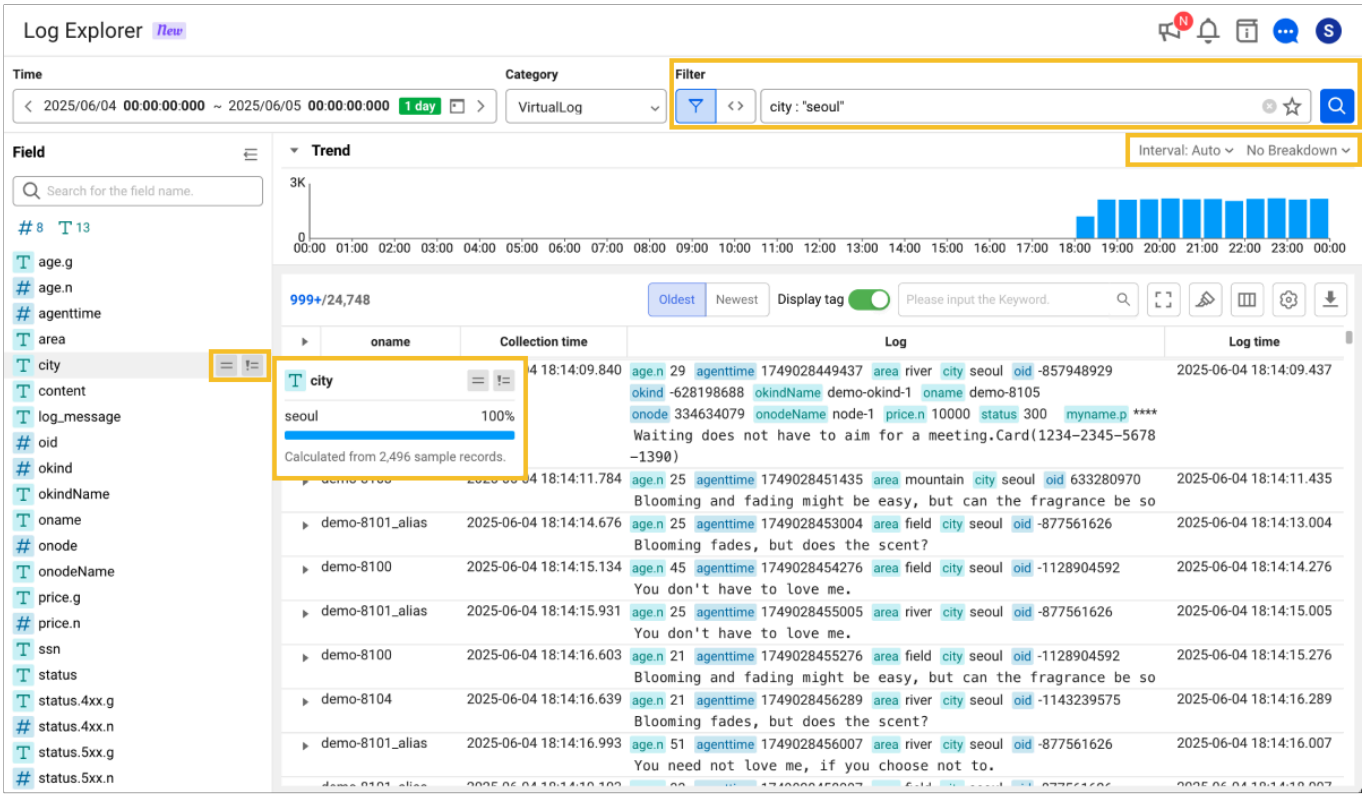
> ⓘ Without the **Log Read** role, you cannot enter the menu.

Home screen > Select Project > **Log** > **Log Explorer** *New*

The **Log Explorer** feature allows you to quickly search, filter, and analyze log data. You can understand the field structure, perform query-based filtering to efficiently search for logs, and visualize the results. This feature is based on the Lucene index and offers the following key functions:

1. **Log trend analysis and search**: Enables both log trend visualization and search, helping you understand log patterns over time.
2. **Fast search and filtering**: Search for problematic logs quickly using simple keywords. Supports both WhaTap log search query and Lucene query syntax for defining various search conditions.
3. **Autocomplete functionality**: Helps users write and learn queries easily through auto-suggestions.
4. **Data visualization**: Uses stacked graphs to visualize the volume of logs over time. Drill-down capability allows deep analysis of specific logs.
5. **Favorites and sharing**: Save frequently used filters as favorites and share them easily via URL.

Basic Screen Overview



Time Range Selection

Set the time range for the log search using the **Time Selector** in the top option bar.

Category

Select the type of log you want to explore in **Select Category** on the top option bar (e.g., AppLog).

Filter

In the **Filter** on the top option bar, select the query language and enter the search conditions to apply the filter. The search query syntax supports both WhaTap Log Search Query and Lucene Query.

- : WhaTap Log Search Query
- : Lucene Query

Search Query Syntax

WhaTap log search query

WhaTap log search queries support autocomplete and allow quick log searches with simple keyword input. You can define various search conditions using operators like `and`, `or`, etc.

- Example: Entering `error` in the search filter will search within the log content.
    1. In the top option bar, click the filter icon and select a field from the query search box (fields are shown based on selected time and category).
    2. Choose an operator:

        a. Default: `:`, `:*`, `and`, `or`

        b. If the field is of type `Number`: Default + `<=`, `>=`, `<`, `>`

        c. If the field is of type `String`: Default only

    3. Select a value (values shown based on selected time, category, and field).

Search Index Field Types

Only three field types are supported: `Number`, `String`, and `Boolean`.

- `Number`: Numeric values like long, double, float
- `String`: Full text such as email body
- `Boolean`: A value of either True or False

> For more details about WhaTap log search query syntax, refer to the documentation.

Lucene

Use the standard syntax of Apache Lucene version 8.11.1. Autocomplete is not supported.

> For more information on the Lucene QueryParser Classic package, see the official documentation.

Field List

Can be used for drill-down analysis.

Apply Field Info to Filter

Check field names in the **Field List** and click the `=` or `!=` buttons to apply them as filter conditions.

- Hovering over a field shows the `=` and `!=` buttons, which can be clicked to apply filters.
- Use the field search function to quickly find the desired field.

Trend Chart

Use **Trend Chart** to view logs in graph form with **interval** and **breakdown** applied.

- **Graph Interaction**: Zoom in using mouse drag or the button, and click bars to view logs in that area
- **Interval**: Choose from Auto, 1s, 5s, 30s, 1m, 5m, 10m, 30m, 1h
- **Breakdown**: Visualize distribution of Top 3 values + Others for a specific field, and apply drill-down filters

Breakdown

Breakdown is a visualization feature that analyzes log count trends by a specific field.

- Displays a stacked graph separating the Top 3 values and Others for a given field
- Useful for drill-down analysis. You can click parts of the legend or stacked graph to highlight specific values or add filter conditions.

View Logs

You can view logs that match your search conditions. In the **Field List**, field data and log tags are displayed in matching highlight colors.

Content Highlight

The highlight function allows you to easily identify desired keywords in log content. You can filter with single or multiple keywords.

1. Enter the keyword you want to highlight in the keyword input field.
   - Example: `select`
2. The searched keyword will be highlighted in the Content of the log list.
   - If you click the ⛶ **Log full screen** icon, you can view **Log** and **Timestamp** in fullscreen.
   - You can navigate between highlighted keywords using shortcuts.
     - `Enter` : Move to the next row
     - `Shift+Enter` : Move to the previous row
     - ∧ / ∨ icons: Move between rows
     - `Cmd` + `F` (Mac) / `Ctrl` + `F` (Windows): Search keywords

Multi-keyword condition

When highlighting with multiple keywords, write as follows:

String to input	Description	Result
`a b c`	Separate each keyword with a space.	a, b, c
`"Whatap is good."`	To include spaces in keywords, wrap them in `''` or `""`.	Whatap is good.
`"Whatap\\ is good."`	If `\` is included in the keyword surrounded by "", enter it as `\\`.	Whatap\ is good.

Setting the highlight color

By selecting **Highlight**, you can set the keyword and its color for highlighting.

- In addition, enter the keyword to set the color for.
- If you click Color on the left of the input field, available colors appear.
- Basically, different highlighting by the level (WARN, ERROR, FATAL) is applied.
- The settings are saved in **Project Rules**.

Column settings

Click the button to add columns or change their order.

- **Add Columns**

  In **Column Settings**, select tags to add them as columns in the table.

  > If no log columns are selected, **Log view detail setting** will not be visible. At least one column must be selected.

- **Reorder Columns**

  Added columns will appear in **Display columns**. Drag them to change the display order.

Log view detail setting

Click the button to configure log display options. By default, both **content** and **Tag** are selected and shown.

- Either **content** or **Tag** must be selected.
- Tags added in **Tag management** appear in order in the log display. You can reorder tags by dragging. Disabled tags are not shown.
- If an item is unchecked, it is not shown in the table. For example, if **Tag** is unchecked, log tags are not displayed.

log table contents remove tag sc

> **Scope of Column and Log Display Settings**
>
> - **Column settings** and **Log view detail setting** can be used in **Live Tail**, **Log Search**, and **Log Trend**.
> - In the same project, **Live Tail**, **Log Search**, and **Log Trend** share **Column settings** and **Log view detail setting**.

Download Log File

You can download the collected logs that match your search results in CSV or TXT file formats.

1. Click the **Download** button at the right center of the screen.
2. Click **Download CSV** or **Download Log** to download the file in your preferred format.

WhaTap Log Search Query Syntax

This document describes the WhaTap log search query syntax used in the Log Explorer.

1. **Equal**
   - `field : value`
2. **Not**
   - `not field : value`
3. **Wildcard** (Only applicable to `String`-type fields; do not use `"` around the value)
   - `field : val*`
4. **Exist**
   - `field : *`
5. **Range** (Only applicable to `Number`-type fields)
   - Operators supported: `<`, `>`, `<=`, `>=`
   - Example: `field < 30 and field >= 10`
6. **Combining Multiple Queries**
   - AND example: `field < 30 and field >= 10`
   - OR example: `field < 30 or field >= 10`
7. **Grouping**
   - Example: `(field < 30 or field >= 10) and other : 70`
8. **Escaping Special Characters**
   - Special characters supported by Lucene escape: `&&` `||` `!` `(` `)` `{` `}` `^` `"` `~` `*` `?` `:` `\` `/`
   - Use `\` before the character to escape
   - Example: To search `(1+1):2`, use `\\(1\\+1\\)\\:2`

Search Index Field Types

Supported field types: `Number` and `String`

- `Number` : Numeric values such as long, double, float
  - Use syntax like `field : "value"`
- `String` : Full text such as email body
  - Use syntax like `field : value`
- `Boolean` : True or False
  - Use as `field : true` or `field : false`

Log Trend

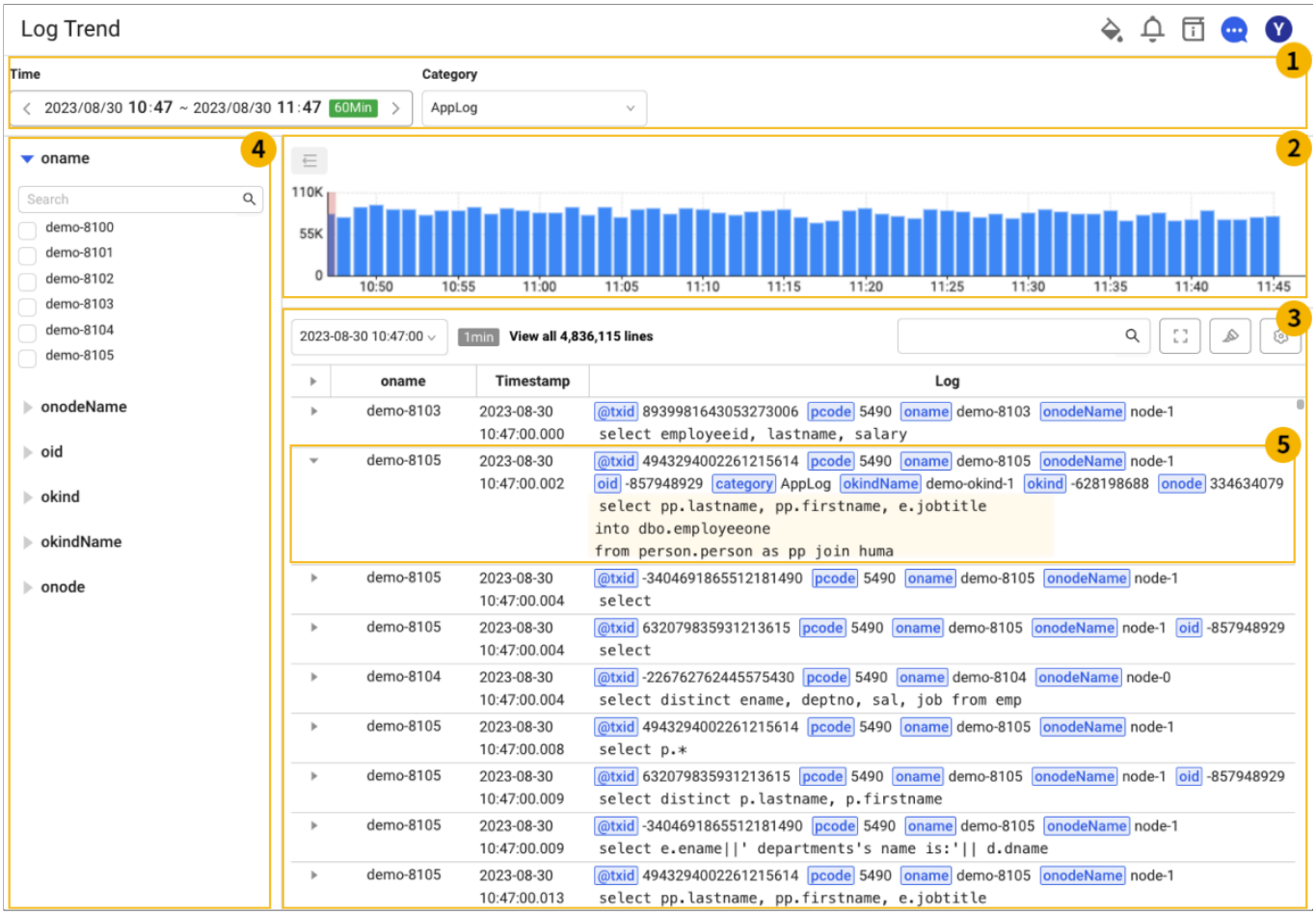
> ⓘ Without the **Log Read** role, you cannot enter the menu.

Home > Select Project > **Log** > **Log Trend**

In **Log Trend**, you can check the occurrence pattern of a specific error type through the occurrence count trend of the logs classified by type and see the log data details by time. You can quickly see the desired log through the highlighting feature. You can also check the trends of the collected logs by category.

The major terms are as follows:

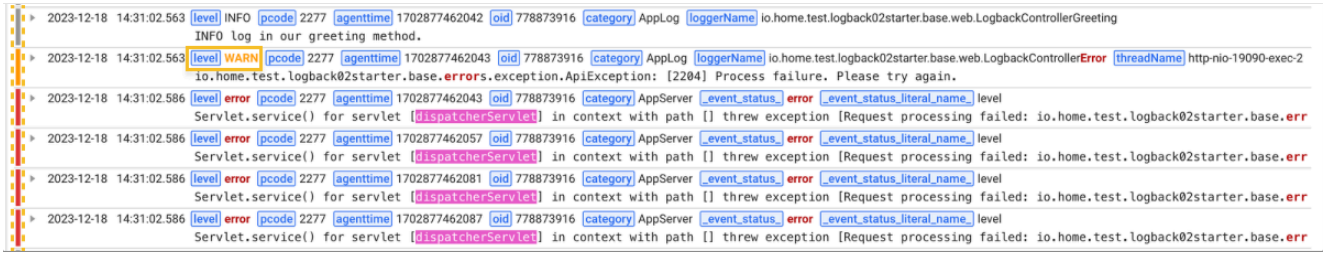
- **Category**: Unit for log collection and inquiry
- **Content**: Log messages
- **Search Key**: It is generated through the log parser settings.
- **Tag**: Key to search the collected logs



Request Data

- If the scroll reaches the bottom, the next data is retrieved.
- In 1, specify the **time** and collection unit, **category**, of the log data to be searched.
- If the **category** is changed, the logs that correspond to the selected category are searched. You can see the logs in 2 bar chart and 3 log table.
- If you click a bar in 2 bar chart, the logs are viewed that correspond to the time range of the bar.
- 3 You can see the total number of logs viewed from the upper left of the log table.
- On the upper right of the 3 log table, select the ⛶ **full screen** icon, you can see the **log** and **collection time** in the full screen.

- 4 In Side menu, you can see the logs filtered by tags. Up to two search keys can be selected, and multiple search values can be selected.
- **If the agent option is set**, the log levels are collected and the colors for each log level appear as follows:



> ⓘ **Setting the agent options**
>
> - The agent options are as follows:
>
> ```
> # whatap.conf
> weaving=log4j-2.17
> weaving=logback-1.2.8
> ```
>
> - In Java agent 2.2.22 or later, it can be used when setting log4j-2.17 or logback-1.2.8 in the weaving setting. It is required to restart the agent.
> - The log level is determined based on the `level` and `type` among the parsed keywords. When there are keys parsed as `level` and `type`, and the parsed values have FATAL, CRITICAL, ERROR, WARN, WARNING, and INFO, the log level color appears.

> ⓘ You can change the column width by dragging an edge of the log table column.

Viewing personal information

If you have the **log personal information query** role, you can display de-identified personal information by enabling the **Display personal information** toggle button.

- For more information about the **log personal information query** role, see the following.
- For more information about **De-identification of personal information**, see the following.

Checking the log content

> **What is content?**
>
> Content means the log messages.

- The first line of the log column is the log's parsed key and value, and the second line is the log's content.
- 3 For each row (log) of the log table, the ▸ **More** button exists. If you select the ▸ **More** button, you can check the entire content of the log like 5.

Configure Highlight

- The highlight function is provided to easily identify the desired keywords.
- 3 In the input field on the right, enter the keyword to highlight. You can check the highlighted keywords like this.

Content Highlight

The highlight function allows you to easily identify desired keywords in log content. You can filter with single or multiple keywords.

1. Enter the keyword you want to highlight in the keyword input field.
   - Example: `select`
2. The searched keyword will be highlighted in the Content of the log list.
   - If you click the ⛶ **Log full screen** icon, you can view **Log** and **Timestamp** in fullscreen.
   - You can navigate between highlighted keywords using shortcuts.
     - `Enter` : Move to the next row
     - `Shift+Enter` : Move to the previous row
     - ∧ / ∨ icons: Move between rows
     - `Cmd` + `F` (Mac) / `Ctrl` + `F` (Windows): Search keywords

Multi-keyword condition

When highlighting with multiple keywords, write as follows:

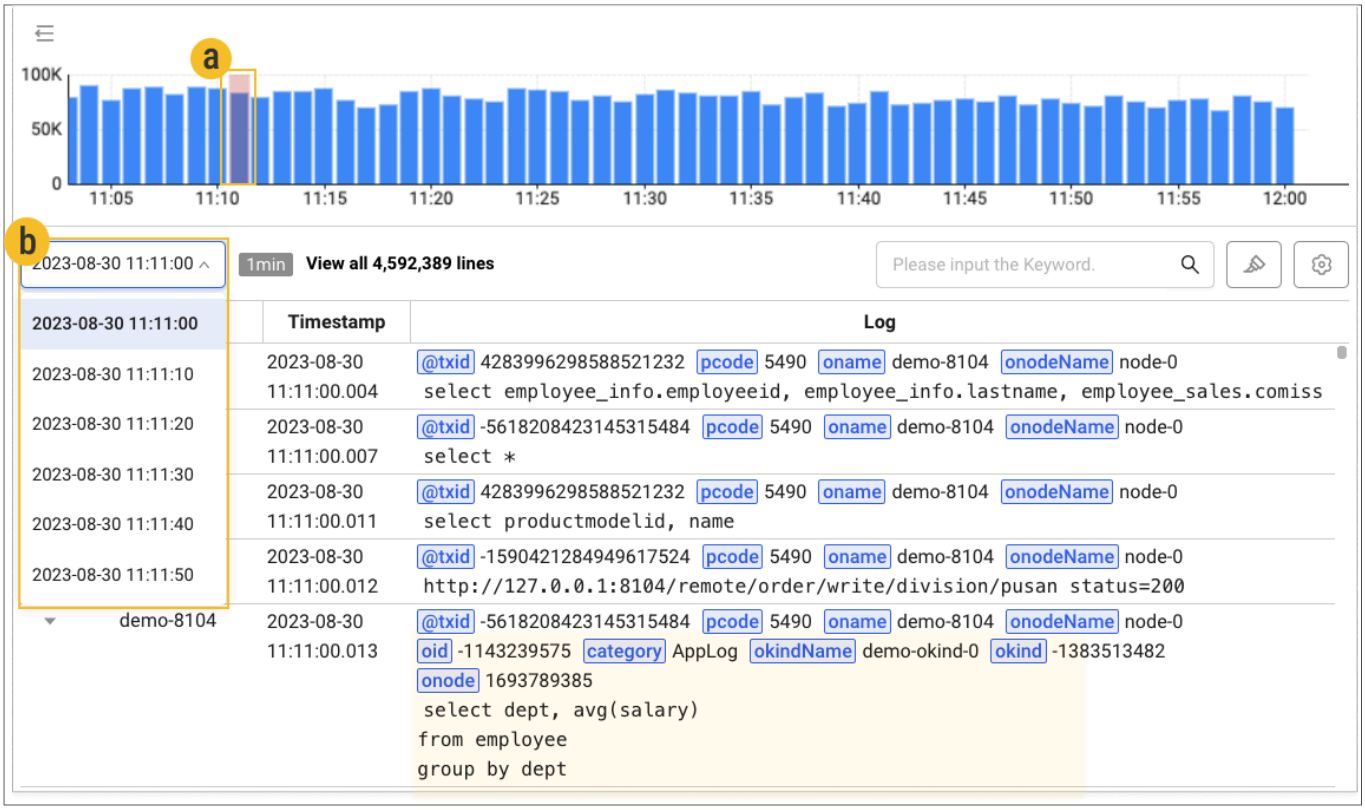
String to input	Description	Result
`a b c`	Separate each keyword with a space.	a, b, c
`"Whatap is good."`	To include spaces in keywords, wrap them in `''` or `""`.	Whatap is good.
`"Whatap\\ is good."`	If `\` is included in the keyword surrounded by "", enter it as `\\`.	Whatap\ is good.

Setting the highlight color

By selecting **Highlight**, you can set the keyword and its color for highlighting.

- In addition, enter the keyword to set the color for.
- If you click Color on the left of the input field, available colors appear.
- Basically, different highlighting by the level (WARN, ERROR, FATAL) is applied.
- The settings are saved in **Project Rules**.

Viewing the logs with a chart



- You can click or drag the desired time range bar on the bar chart (a) to view the logs for the time range.
- You can use the b time selection option on the upper left of the log table under the bar chart to further refine your log search in the selected time zone:
  - 1min: interval (spacing between bars on the chart)
  - Select a time option: Time zone whose time range is divided into 6 segments.

Setting the table

Column settings

Right of the 3 area, select **Column settings**. You can add columns or set order.

- **Adding columns**

  You can add columns to the table by selecting tags. If you unselect the **log** column, you cannot see **Log view detail setting**. Be sure to select at least one column.

- **Setting the column order**

  When you add a column, the column is added to **Display columns**. Drag a desired column to change the order of the columns.

Log view detail setting

On the right of 3, select **Log view detail setting**. **content** and **Tag** are all checked and two items are also displayed. Be sure to select any of **content** and **Tag**.

Unchecked items are not displayed in the table. If you deselect the **Tag** as follows, the log's **Tag** does not appear in the table.

Log table tag invisible

When you add tags in the **Tag management** list, the tags in the log are listed in the order you added. You can change the order of the tags by dragging them. f you disable an added tag, the disabled tag does not appear.

> - **Column settings** and **Log view detail setting** can be used in **Live Tail**, **Log Search**, and **Log Trend**.
> - In the same project, the **Live Tail**, **Log Search**, and **Log Trend** menus share **Column settings** and **Log view detail setting**.

Log Search

> ⓘ Without the **Log Read** role, you cannot enter the menu.

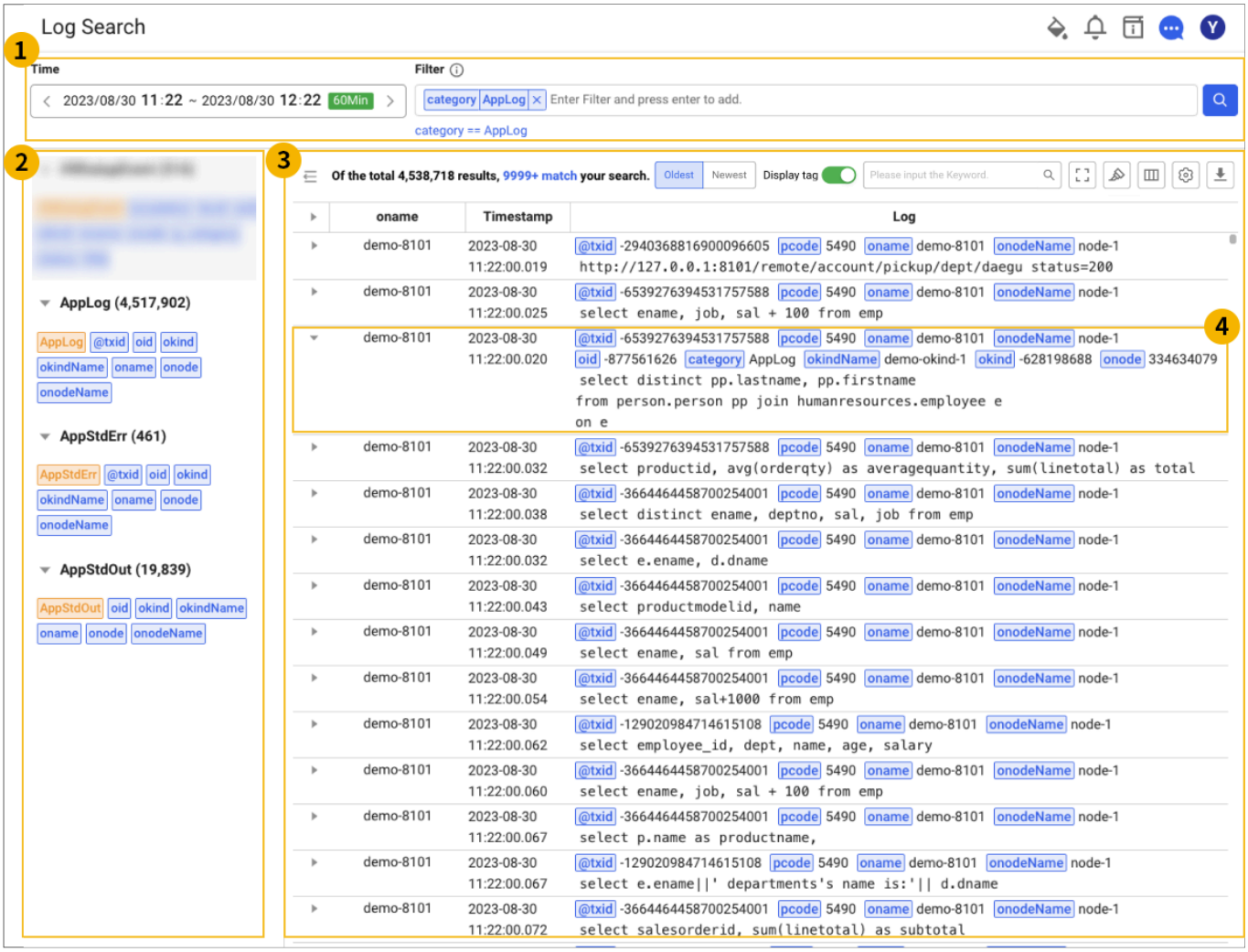
Home > Select Project > **Log** > **Log Search**

In **Log Search**, you can search a large number of integrated logs under various conditions and specify the desired logs. Multiple search conditions can be specified with parsed keys and values, so only the log data that matches the desired conditions are extracted.

The searched log data is imported to the dynamic page line by line. If the scroll reaches the bottom, the next data is automatically retrieved and displayed.

The major terms are as follows:

- **Category**: Unit for log collection and inquiry
- **Content**: Log messages
- **Search Key**: It is generated through the log parser settings.
- **Tag**: Key to search the collected logs

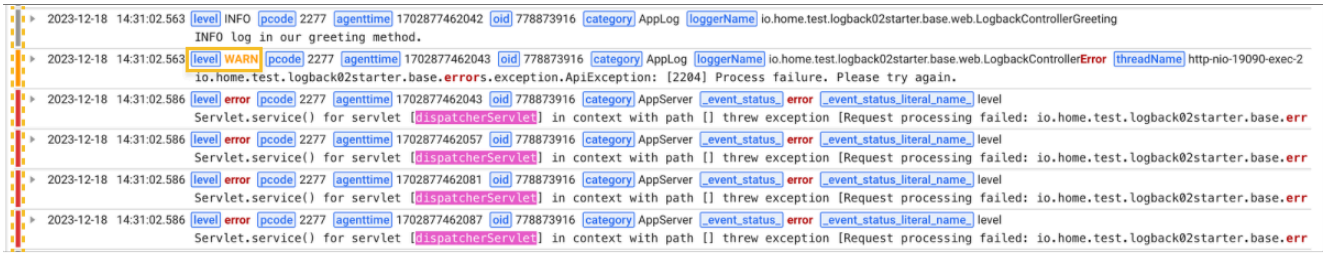


Request Data

- If the scroll reaches the bottom, the next data is retrieved. You can view 10,000 logs at a time.
- 3 You can see the total number of logs viewed from the upper left of the log table.
- Log data can be queried in chronological order. On the upper right of the 3 log table, select a desired search method among **Oldest** and **Newest**.
- After specifying the time range, select **Apply** to finish the search time setting, and then select **Search** to search for data.
- On the upper right of the 3 log table, select the **Log full screen** icon, you can see the **Log** and

Timestamp in the full screen.

- **If the agent option is set**, the log levels are collected and the colors for each log level appear as follows:



> **Setting the agent options**
>
> - The agent options are as follows:
>
> ```
> # whatap.conf
> weaving=log4j-2.17
> weaving=logback-1.2.8
> ```
>
> - In Java agent 2.2.22 or later, it can be used when setting log4j-2.17 or logback-1.2.8 in the weaving setting. It is required to restart the agent.
> - The log level is determined based on the `level` and `type` among the parsed keywords. When there are keys parsed as `level` and `type`, and the parsed values have FATAL, CRITICAL, ERROR, WARN, WARNING, and INFO, the log level color appears.

> You can change the column width by dragging an edge of the log table column.

Checking the log content

> **What is content?**
>
> Content means the log messages.

- The first line of the log column is the log's parsed key and value, and the second line is the log's content.

- 3 For each row (log) of the log table, the ▸ **More** button exists. If you select the ▸ **More** button, you can check the entire content of the log like 4.
- If you select a tag in the log, a drop-down menu appears that allows you to perform **Copy**, **Search**, **Exclusion search**, and **Adjacent Log**.

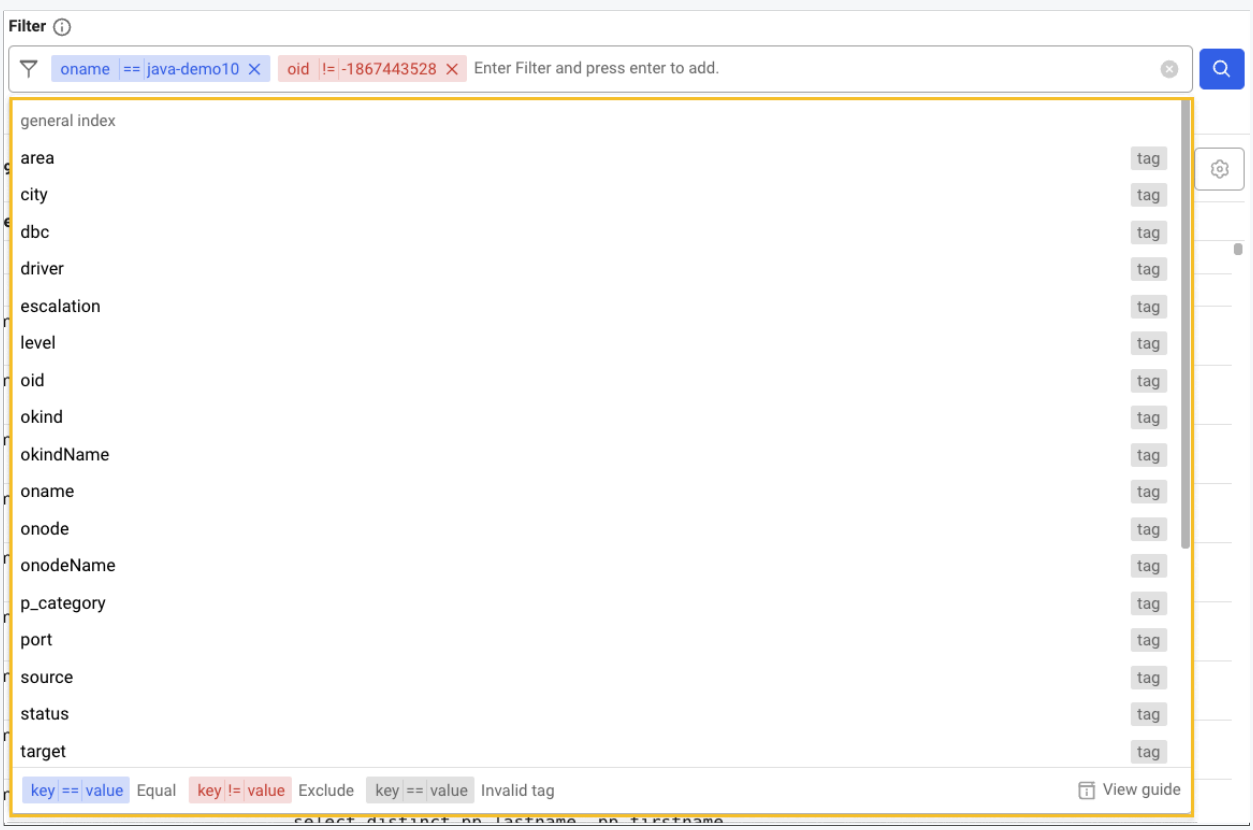
Filter

Apply Filter

In **Time selector** on the left of 1, you can specify the time range. On the right, When a filter is applied, the logs that match the specified conditions are filtered. You can enter multiple filters. OR (`||`) if the filters have the same tags, otherwise AND (`&&`).

You can specify a filter by directly entering a value in the input field or clicking the **Filter** field. Filter tags are entered in the following order: `Search key` , `Operator` , and `Search value` . If you select **Search**, the data with the filter applied can be viewed log list area.

> **Guide UI**
>
> The **Guide UI** is provided under the input field as follows: You can close the **Guide UI** via the ESC key when your mouse cursor is over the input field or filter tag.



Entry of the search key, operator, and search value

- Upon entry of the `search key`, recommended values are provided by distinguishing for general index, reserved word index, and numeric index.
- When entering an `operator`, for a general index search key, `==` and `!=` options are provided at the bottom. In case of the numeric index, the `>`, `<`, `<=`, `>=`, `==`, and `!=` options are provided.
- Upon entry of the `search value` for matched search (`>`, `<`, `<=`, `>=`, `==`), it is highlighted in blue, In case of excluded search (`!=`), it is highlighted in red. Upon entry of the `search value`, you can search using the case-sensitive option.

- If the filter tags exceed 2 lines, you can fold it by selecting the ^ **Fold** icon.
- After entering a filter tag, you can use the Shift and Tab keys simultaneously in the input field to go to the previous filter tag.

- After entering a filter tag, you can use the Tab key in the input field to go to the next filter tag.

Adding filter tags

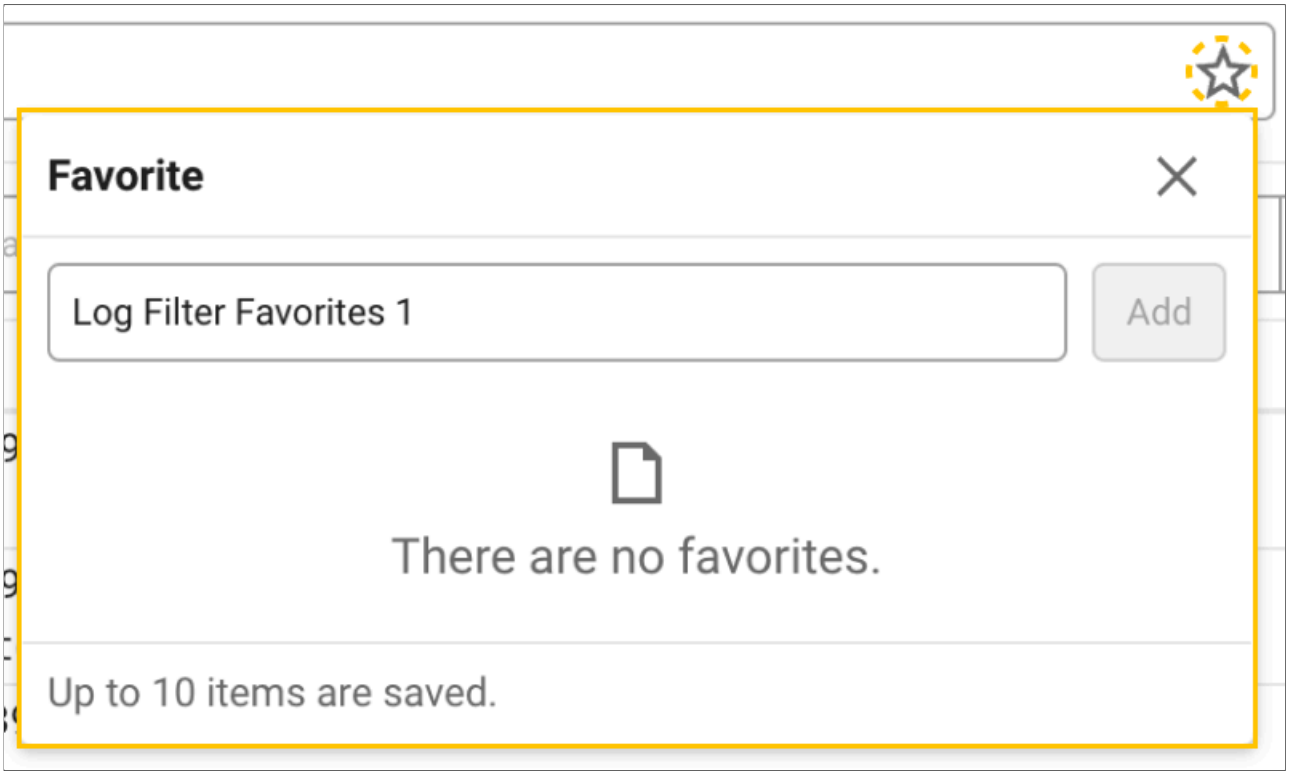
To add a filter tag, type the desired text in the input field and press **Enter** or **Tab**.
You can also select a recommended value from the **Guide UI** below the input field by clicking it or navigating with the arrow keys.

Deleting filter tags

To remove a single tag, press **Backspace** or click the **X** icon on the tag. To clear all tags at once, click the **X** icon located on the right side of the input field.

Filter favorites

The filter **favorites** feature is provided. You can select the ☆ **Favorites** icon on the right of the input window to **add**, **delete**, and select desired filter search conditions in the favorites. **Favorites** can store up to 50 cases.

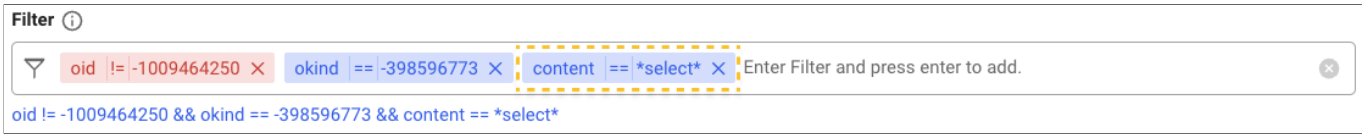


Exceptional cases for application of filters

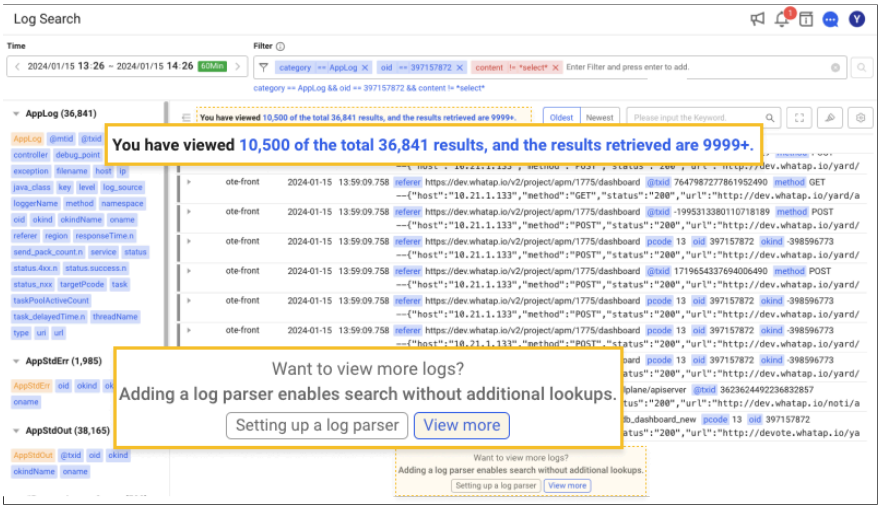
- In the tag with a numeric index (`search key` that ends with `.n`), only numbers can be input for the `search value` .
- Duplicate 'search key' and 'search value' cannot be entered.
- Searching is not possible if there is a tag whose `search key` or `search value` is missing. Invalid tags appear in gray.

Applying non-parsing keyword filters

You can search the logs containing keywords that have not been parsed, or for which no index has been created. In this case, all logs in the specified range are fully scanned. Therefore, the search speed may be slower compared to the indexed key. For standardized log data, it is recommended to search using the index key through **Log parser setting**.



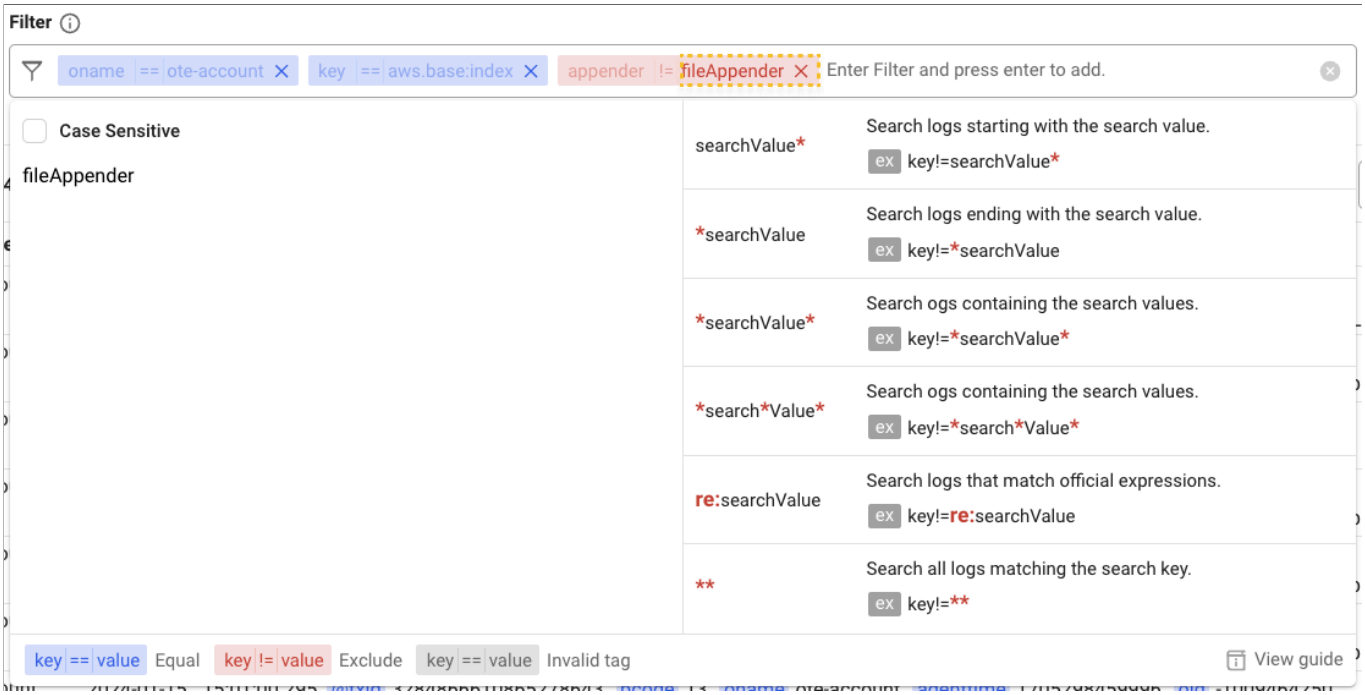
1. Select **Category**. It is required to set the category.
2. In the **Filter** field, enter a desired keyword for a search after leaving spaces on the `content` basis.
   e.g. `content *select*`
3. Click **Search** to view the logs. Several logs are viewed first. You can view a maximum of 10,000 logs per each search.
4. You can make additional inquiries by scrolling down and selecting **View more** at the bottom.



- Among all logs, 10,000 logs are viewed for a server search range. The default range for a server search is 200,000, but the ratio may differ depending on the total log volume.
- For more information about the parser configuration, see the following.

Modifying the filter

Enter a value in the filter and then click the specified value to modify it.



- You can modify the filter by entering text again in the input field.
- You can modify the recommended values by selecting them in the **Guide UI** below the input field.

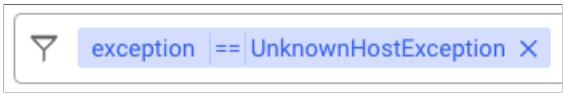
Search Key

The search key Indicates an identifier for accessing a desired value within the log data. The actual data for the search key is the search value. The tags on the left (2) area are the search keys parsed for each category. You can enter filters by selecting tags. The orange tag is the category, and the blue tag is the search key.

For example, AppLog and AppStdOut in (2) are categories. The tags below such as oid are search keys parsed. In **Log** > **Log Configuration**, you can register and set the parsing logic in **Log parser setting** tab with the search key.

Filter input syntax

A tag consists of a search key and a search value. In the following example, the search key is `exception` and the search value is `UnknownHostException`. The example searches for the log data that includes an exception (`UnknownHostException`) that is generated when the server cannot connect the host because the IP address and domain address do not match among the collected log data.



Search key types

Search key types	Search key format	Meaning	Example for Search key and search value	Seach example
String keyword	keyword	File Name	- Key: fileName - Value: /data/whatap/logs/yard.log	fileName:/data/whatap/logs/yard.log
Numeric keyword	keyword.n	Response Time	- Key: response_time.n - Value: 2945	response_time.n>=2945
Reserved keyword (predefined keyword)	@keyword	Transaction ID	- Key: @txid - Value: 85459614215434144	-

Common syntaxes

Syntax type	Description	Example
`==searchValue`	Searches for the logs that match the search value.	`exception==RuntimeExceptionexception`
`!=searchValue`	Searches for the logs that exclude the search value.	`exception!=RuntimeException`
`*searchValue`	Searches for the logs ending with the search value.	`word==*hello`
`searchValue*`	Searches for the logs starting with the search value.	`word==hello*`
`*searchValue*`	Searches for the logs that contain the search value	`word==*hello*`

Syntax type	Description	Example
	in the middle.	
`*search*Value*`	Searches for the logs that contain the search value.	`word==*he*llo*`
`re:{regex}`	Searches for the logs that match a regular expression.	`caller==re:^i\.w\.a\.w\.s\.v\.r\.`
`**`	Searches all logs that match the search key.	

Syntax in case the search key has a numeric value (keyword.n)

The following syntax is supported only when the search key is in `keyword.n` format.

- Search values must only be numbers.
- No prefix is attached to the value of the `.n` keyword. All keywords other than `.n` must be attached with a prefix.

  For example, `+>searchValue` is invalid.

Syntax type	Description	Example
`>searchValue`	Searches for the logs containing values greater than the search value.	`response_time.n>3000`
`>=searchValue`	Searches for the logs containing values greater than or equal to the search value.	`response_time.n>=3000`
`==searchValue`	Searches for the logs containing the same search value.	`response_time.n==3000`
`!=searchValue`	Searches for the logs that contain values different from the search value.	`response_time.n!=3000`
`<searchValue`	Searches for the logs containing values less than the search value.	`response_time.n<3000`

Syntax type	Description	Example
`<=searchValue`	Searches for the logs containing values less than or equal to the search value.	`response_time.n<=3000`

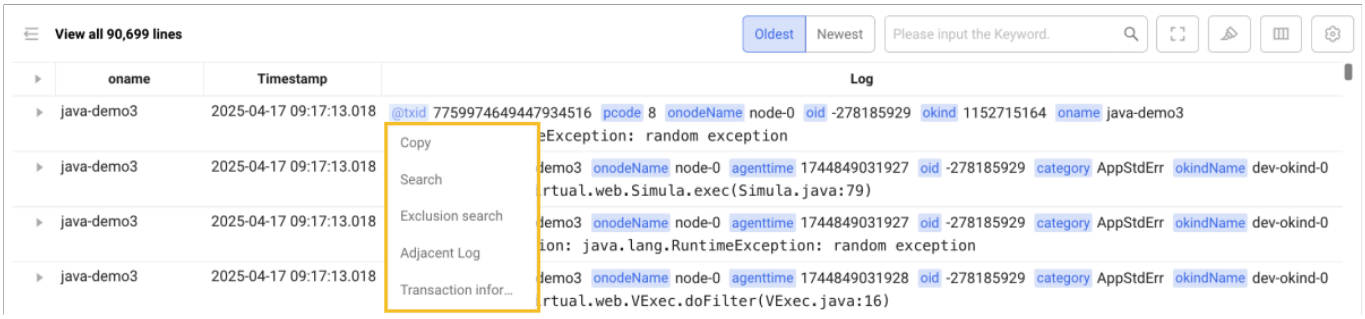
Viewing personal information

If you have the **log personal information query** role, you can display de-identified personal information by enabling the **Display personal information** toggle button.

- For more information about the **log personal information query** role, see the following.
- For more information about **De-identification of personal information**, see the following.

Log tagging options

When you select a log tag, the following drop-down menu appears. You can find the **Search**, **Exclusion search**, **Adjacent Log** and **Transaction information** options.



- Copy
- **Search**

  If you select the **Search** option, the tags that correspond to **Filter** are entered with the inclusion (`==`) condition.

- **Exclusion search**

  If you select the **Exclusion search** option, the tags that correspond to **Filter** are entered with the exclusion (`!=`) condition.

- **Adjacent Log**

  If you select the **Adjacent Log** option, the **Adjacent Log** detail window appears. Search for the logs of the target server and the logs in time zones adjacent to the selected log. You can click **Time selector** to see the logs in adjacent time zones. Reference logs appear with a blue background.

- Transaction

  If the log contains the `@txid` and `@mtid` tags, you can see the information in the transaction Information details window by clicking the transaction information option.

Content highlighting

The highlight function allows you to easily identify desired keywords in log content. You can filter with single or multiple keywords.

1. Enter the keyword you want to highlight in the keyword input field.
   - Example: `select`
2. The searched keyword will be highlighted in the Content of the log list.
   - If you click the ⛶ **Log full screen** icon, you can view **Log** and **Timestamp** in fullscreen.
   - You can navigate between highlighted keywords using shortcuts.
     - `Enter` : Move to the next row
     - `Shift+Enter` : Move to the previous row
     - ∧ / ∨ icons: Move between rows
     - `Cmd` + `F` (Mac) / `Ctrl` + `F` (Windows): Search keywords

Multi-keyword condition

When highlighting with multiple keywords, write as follows:

String to input	Description	Result
`a b c`	Separate each keyword with a space.	a, b, c

String to input	Description	Result
`"Whatap is good."`	To include spaces in keywords, wrap them in `'` or `""`.	Whatap is good.
`"Whatap\\ is good."`	If `\` is included in the keyword surrounded by "", enter it as `\\`.	Whatap\ is good.

Setting the highlight color

By selecting **Highlight**, you can set the keyword and its color for highlighting.

- In addition, enter the keyword to set the color for.
- If you click Color on the left of the input field, available colors appear.
- Basically, different highlighting by the level (WARN, ERROR, FATAL) is applied.
- The settings are saved in **Project Rules**.

Setting the table

Column settings

Right of the 3 area, select **Column settings**. You can add columns or set order.

- **Adding columns**

  You can add columns to the table by selecting tags. If you unselect the **log** column, you cannot see **Log view detail setting**. Be sure to select at least one column.

- **Setting the column order**

  When you add a column, the column is added to **Display columns**. Drag a desired column to change the order of the columns.

Log view detail setting

On the right of 3, select **Log view detail setting**. **content** and **Tag** are all checked and two items are also displayed. Be sure to select any of **content** and **Tag**.

Unchecked items are not displayed in the table. If you deselect the **Tag** as follows, the log's **Tag** does not appear in the table.

Log table tag invisible

When you add tags in the **Tag management** list, the tags in the log are listed in the order you added. You can change the order of the tags by dragging them. f you disable an added tag, the disabled tag does not appear.

- **Column settings** and **Log view detail setting** can be used in **Live Tail**, **Log Search**, and **Log Trend**.
- In the same project, the **Live Tail**, **Log Search**, and **Log Trend** menus share **Column settings** and **Log view detail setting**.

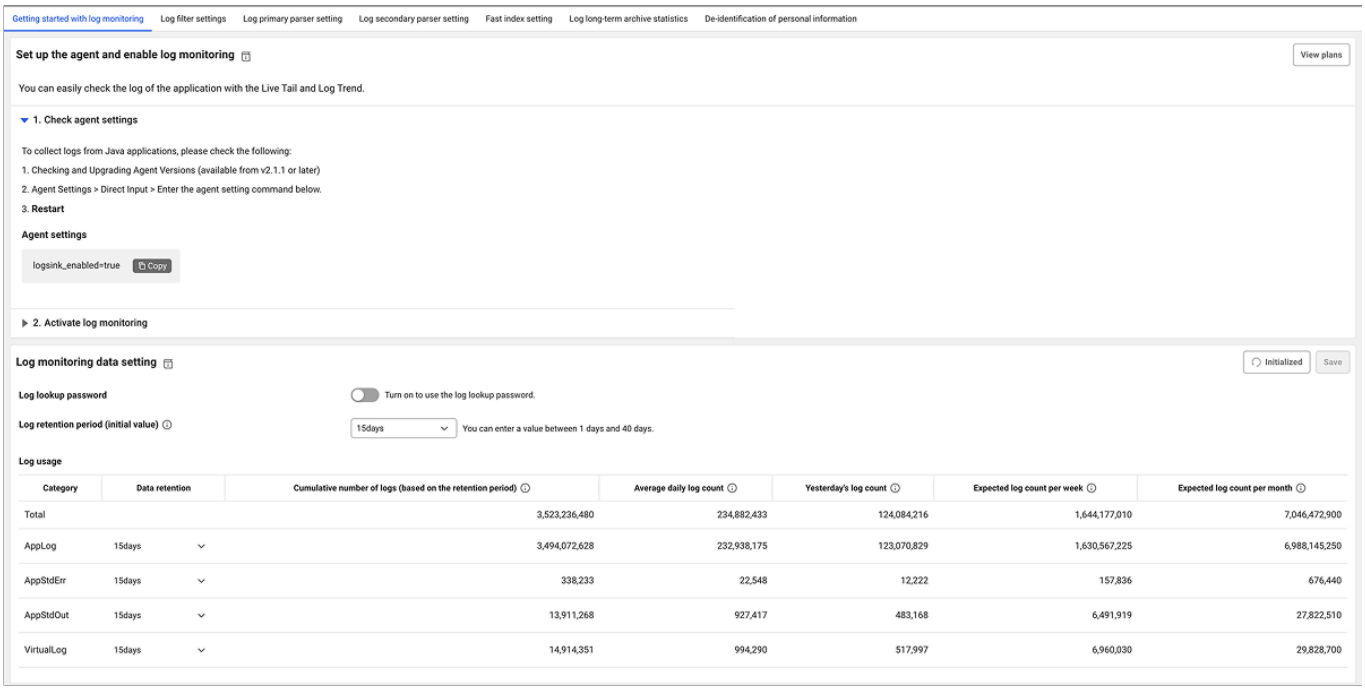
Log Setting

Home > Select Project > **Log** > **Log Configuration**

You can configure the log monitoring related settings in **Log Configuration**. Using the tab at the top, you can use menus for checking the agent settings, determining whether to enable log monitoring, setting the log data retention period and lookup password, registering the log parser, and setting the quick indexes.

- To use the **Activate log monitoring** function, the **Edit project** role is required.
- The **Log edit** role allows you to modify the **Log Configuration** menu other than **Activate log monitoring**.

Starting the log monitoring



At the top, select the **Getting started with log monitoring** tab. If you select the **View guide** icon and the **View plans** button, the corresponding guide screen appears.

Set up the agent and enable log monitoring

In the 1 area, check the agent settings, and then click the **Activate log monitoring** toggle button to set whether or not to enable the log monitoring.

Check agent configuration

Check the agent version and settings to start the log monitoring. Select **Check Agent Configuration** to follow the onscreen instructions.

- **Server Monitoring**

  Under **Apply**, check the application guides for each application. See the following guides: Java, PHP, Node.js, Python, Go.

- **DB Monitoring**

  Under **Apply**, check each individual application guide. See the following. According to the instructions, add the settings for the log monitoring target file in whatap.conf.

- **Kubernetes Monitoring**

  Under **Apply**, check each individual application guide. See the following.

Activate log monitoring

Select **Activate log monitoring** to set whether or not to enable the log monitoring.

- If you turn on the toggle button, the log monitoring is enabled. You can try it for free for 15 days from the activation date.
- If you turn off the toggle button, the log monitoring is disabled. Logs are no longer saved.

Log monitoring data setting

In the 2 area, you can see the **Log usage**. You can also change the **Log retention period** and **Log lookup password** settings.

Log retention period

This is the default data retention period to be applied commonly. If not specified, the default value is 1 day. In addition to the option to select a log retention period, you can enter a desired period of time. If not set separately in the **Log usage** list, this log data retention period is applied by default. If you set the data retention periods for each category in the **Log usage** list and select **Initialized**, the default data retention period is reset.

Log usage amount

You can specify the log data retention periods for each category in the **Log usage** list. **Log Count** means the number of logs stacked for the period. For example, **Daily log count** is the number of logs accumulated during the day, and **Expected log count** is the number of logs of the today's log count multiplied by the days of data retention.

You can specify the log data retention period as follows. You can free the space by deleting old data according to the specified period.

- **Trial Project**

  You can select 1, 2, or 3 days for the data retention period.

- **Paid Project**

  For the data retention period, you can select one of 1 day, 2 days, 3 days, 4 days, 5 days, 6 days, 7 days, 10 days, 30 days, and 40 days.

- **Data size-based billing**

  The price differs depending on the data retention period.

  > For example, if an average of 2 million logs per day accumulates and the data retention period is set to 3 days, an average of 6 million logs is kept on the collection server and subject to billing.

Log lookup password

To enhance security, set the **Log lookup password**. The log lookup password is optional. If you are using a log lookup password, enter the password to go to the log screen.

> ⓘ **In case you forgot the password**

If you have the **Edit Log** role, you can modify it with a new password in **Log Configuration**.

Log Filter Settings

Log Filter Settings allow you to store only the logs that match the filter conditions specified by the user.
By storing only the necessary logs, you can reduce costs.

> **Permission**
>
> Only users with log edit permissions can configure these settings.

Add a Log Filter Setting

This section explains how to add a new filter setting. When configuring log filter settings, **Filter Name, Show in Live Tail, Category, and Filter Condition** are required fields.

Even logs that are not stored by the filter can still be viewed in **Live Tail** if the **Show in Live Tail** toggle is enabled.

> Only one filter can be set per category.

1. Go to **Logs** > **Log Settings** > **Log Filter Settings** tab.
2. Click the **+ Add** button to create a new filter setting.
3. In **Add Log Filter Setting**, enter **Filter Name, Show in Live Tail, Category, and Filter Condition**.
    - **Filter Name**: A name to identify the filter.
    - **Show in Live Tail**: Select whether to display in Live Tail regardless of storage.
    - **Category**: The log category to which the filter is applied **(multiple selection allowed)**.
    - **Filter Condition**: Conditions for logs to be stored by the filter.

      > For more details on filter conditions, see WhaTap log search query syntax.

    - **Example**

- **Include**: `oid : 123`
  - Collect only logs whose oid is 123.
- **Exclude**: `not oid : 123`
  - Collect only logs whose oid is **not** 123.
- **Include all**: leave the filter condition empty
- **Exclude all**: `not content : *`

4. Click **Add** to review and edit the filter settings in the **Preview Results** screen.
5. Once confirmed, click **Apply** to save the log filter settings.

Simulation

You can preview the log results based on the entered category and filter conditions before applying the log filter settings.

Click the **Simulation** button to display the **Log Search** screen with the applied filter settings.

Edit a Log Filter Setting

Registered log filters can be modified by clicking the edit icon. The **Edit Log Filter Setting** screen shows the existing values, which you can change and apply as needed.

> When editing log filter settings, only one category can be selected.

Delete a Log Filter Setting

Registered filters can be deleted by clicking the delete icon. After deletion, make sure to click the **Save** button.

> If changes are not saved, the filter will not be deleted.

Import/Export Log Filter Settings as JSON

Log filter settings can be imported or exported in JSON format.
This is useful if you want to apply the same log filter settings to another project.

Export JSON

1. Click the JSON button at the top right of the screen.
2. In the **Export JSON** window, click the **Export** button.
3. The log filter settings will be downloaded as a JSON file.

Import JSON

1. Click the button at the top right of the screen.
2. Select the JSON file to import.
3. In the **Import JSON** window, click either **Add to List** or **Overwrite** to import the log filter settings.
4. Click the **Save** button to apply the changes.

Validation Rules

The following validation rules are applied when saving settings:

- **type**: Only `GENERAL` is supported.
- **category**: Required, supports `string` type.
- **condition**
  - **language**: Only `jsonDsl` and `lucene` are supported.
  - **description**: Invalid filter conditions cannot be saved.
- **showLiveTail**: Only `boolean` type is supported.
- **name**: Required, supports `string` type.

Parsing The Logs

The log parser helps you transform the irregular logs into a structured form. WhaTap Log Monitoring provides two types of parsers.

- **GROK parser**: The logs collected in arbitrary forms are parsed using regular expressions and GROK syntax.
- **JSON parser**: The logs collected in the JSON form are parsed.

> ⓘ **Common precautions**
>
> - If multiple parsers are registered in the same category, only the first matching parser is applied.
> - **WhaTap has the role to disable the parsers that may affect the stability of the WhaTap service.**

GROK parser

If the logs are collected in an irregular form, you can use the GROK parser to parse them. GROK syntax provides named regular expressions, allowing you to use regular expressions more easily.

Starting GROK

GROK provides two types of syntaxes.

1. `%{SYNTAX:SEMANTIC}` : Syntax provided in the GROK library. Using the **named regular expressions**, you can extract tags. For usage examples, see the following.
   - **SYNTAX**: Specify the named regular expressions provided by GROK.
   - **SEMANTIC**: Specify the name for the matched value.

   > ⓘ **named regular expressions**
   >
   > The syntax is provided by GROK. This function provided by GROK allows you to assign names to complex regular expressions.

name	regular expression
WORD	`\b\w+\b`
SPACE	`\s*`
NOTSPACE	`\S+`
UUID	`[A-Fa-f0-9]{8}-(?:[A-Fa-f0-9]{4}-){3}[A-Fa-f0-9]{12}`

To see all named regular expressions provided by WhaTap, see the following link.

2. `(?<SEMANTIC>REGX)` : The **named capturing group** syntax for regular expressions. You can use regular expressions to extract tags according to the user's intent. For usage examples, see the following.

- **SEMANTIC**: Specify the name for matched values.
- **REGX**: Enter regular expressions for matching.

named capturing group

It is the syntax provided by regular expressions.

- capturing group: This indicates the function to bundle multiple tokens into one as a single matching unit.
- named capturing group: Name assigned to the capturing group.
- Let's look at the string matching example. dev@whatap.io
  - Example 1 `(\w+)@(\w+\.\w+)`
  - Example 2: When the entire email is matched and the username and domain are matched additionally, `(?<username>\w+)@(?<domain>\w+\.\w+)`

%{SYNTAX:SEMANTIC} Usage example

The following is an example that uses the %{SYNTAX:SEMANTIC} syntax.

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
```

You can look at the sample log to infer what each word means. When replacing each part with semantic words, it can be expressed as follows:

semantic replace

```
[date][logLevel][caller] projectCode logCategory dexBuildStartTime {area=areaEnum, city=cityEnum} dexBuildElapsed
```

Any semantic word can be replaced with a regular expression. The GROK parser allows you to use the predefined named regular expressions. TIMESTAMP_ISO8601 , LOGLEVEL , and DATA used here are named regular expressions provided by GROK. These values are matched after replacing each of them with the following regular expression.

- 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)?
- name: DATA
  - regular expression: .*?

GROK parsing pattern

```
\[%{TIMESTAMP_ISO8601:date}\sGMT\]\[%{LOGLEVEL:level}\s\]\[%{DATA:caller}\]
```

If you parse with the above syntax, you can extract tags as follows. In this way, GROK's %{SYNTAX:SEMANTIC} syntax helps you apply complex and long regular expressions easily and concisely.

Tag extraction

```
- date : 2023-08-08 02:02:30,101
- caller : i.w.y.l.c.LogSinkDexScheduleThread.realProcess(159)
- level : LEVEL
```

(?<SEMANTIC>REGX) Usage example

The parts that do not match with the named regular expressions can be parsed using the (?<SEMANTIC>REGX) pattern. In the sample log above, the parts that cannot be parsed with the %{SYNTAX:SEMANTIC} syntax are as follows.

Unparsed area

```
8 VirtualLog 20230808 02:01:00.000 {area=4, city=5} 56ms
```

When replacing each part of the log with semantic words, it can be expressed as follows.

semantic replace

```
projectCode logCategory dexBuildStartTime {area=areaEnum, city=cityEnum} dexBuildElapsed
```

These irregularly strings can be parsed using the (?<SEMANTIC>REGX) syntax.

Regular expressions that match for each sample log parsing keyword

Parsing keyword	(?<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)

Parsing keyword	(?<SEMANTIC>REGX)
56ms	(?<dexBuildElapsed>\d{2})ms

Basic regular expression syntax

Syntax rule	Meaning	Alias
?	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	-
[0-9&&[^2468]]	subtraction	-
a{2,}	2 or more	-
a{1,3}	between 1 and 3	-

Syntax rule	Meaning	Alias
a+?	match as few as possible	-
{2,3}?	match as few as possible	-
(abc)	capturing group (processes multiple strings as a single unit)	-
\d	digit	[0-9]
\D	non-digit	[^0-9]
\W	non-word character	-
\S	non-white space	-

By connecting the parsed keywords with space (\s) and special character escapes (\{ , \, , \}), you can apply the pattern as follows.

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),\scity=(?<cityEnum>\d)\}\s(?<dexBuildElapsed>\d{2})ms
```

If you parse with the above syntax, you can extract tags as follows.

Tag extraction

```
- projectCode : 8
- logCategory : VirtualLog
- dexBuildStartTime : 20230808 02:01:00.000
- areaEnum : 4
- cityEnum : 5
- dexBuildElapsed : 56
```

Applying GROK

Log Configuration > **Log primary parser setting**

1. To apply the GROK pattern parser, go to the **Log primary parser setting** tab in **Log Configuration**.
2. Select + **Add** and then select the GROK parser in the **Parser** field.
3. When you select **Register pattern**, the pattern registration and simulation window appears on the right.
4. Enter **Pattern** and **Log**, and then click **Simulation** to check whether parsing is successful with the pattern to apply.

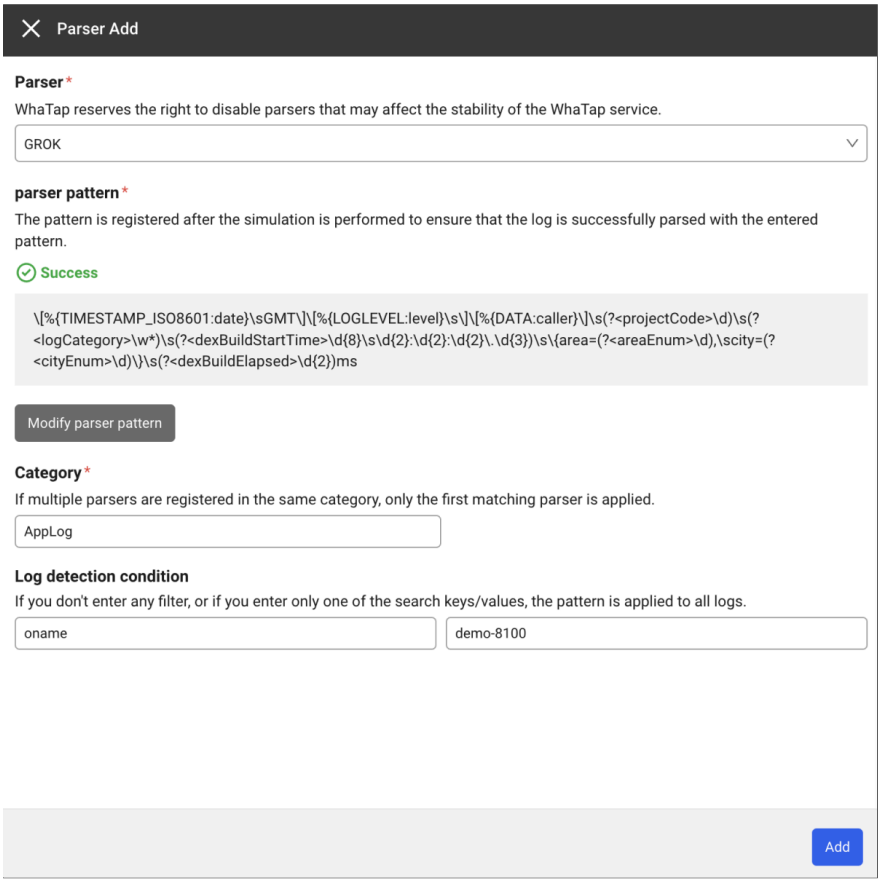
   Pattern e.g.

   ```
   [%{TIMESTAMP_ISO8601:date}\sGMT][%{LOGLEVEL:level}\s][%{DATA:caller}]\s(?<projectCode>\d)\s(?<logCatego
   d{2}:\d{2}:\d{2}\.\d{3})\s\{area=(?<areaEnum>\d),\scity=(?<cityEnum>\d)\}\s(?<dexBuildElapsed>\d{2})ms
   ```

   Log example:

   ```
   [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
   ```

5. If the simulation is successful, you can view **Simulation result** and **Performance measurement results**.
6. When you click **Apply pattern** after simulation, the pattern that has been entered for the selected parser is applied.
7. After applying the pattern, enter values for **Category**, **Log detection condition**, and **Pattern**.



- **Category**

  Select a log category. **Category** is required.

- **Log detection condition**

  - Only the logs that meet the conditions are applied to the parser.
  - Select or enter values for **Search key** and **Search value**.
  - **Log detection condition** are applied before all parsers are run. Accordingly, you cannot use the **Tag** that have been added for the parser.

8. Select **Add** to register a parser.

> - In the log parser list, you can change the **Priority** or **Enable**, **Edit**, or **Delete** the parser.
> - After **parser simulation**, you can register a pattern.

GROK parser precautions

- GROK parser supports two patterns: `%{SYNTAX:SEMANTIC}` and `%{SYNTAX:SEMANTIC}`.
- When using the `%{SYNTAX:SEMANTIC}` pattern, `SEMANTIC` must be entered.
- When using the `%{SYNTAX:SEMANTIC}` pattern, `SEMANTIC` must be unique in a parser.
- When using the `(?<SEMANTIC>REGX)` pattern, `SEMANTIC` can only contain characters (a-z, A-Z), numbers (0-9), and specified special characters (`.` , `_` , `-`).
- `SEMANTIC` must start with a character (a-z, A-Z).
- `SEMANTIC` must end with a character (a-z, A-Z) or number (0-9).

Selecting additional GROK patterns

During GROK parser simulation, in addition to the basic patterns (`WORD` , `NUMBER` , `IP` , etc.), you can select **additional patterns** that match a specific log format. 32 pattern types, such as Java, Nginx, and MySQL, are provided grouped into 11 categories.

Additional pattern category dropdown

1. In the parser simulation window, click the dropdown in the **Additional patterns (optional)** area.
2. From the list of pattern types grouped by category, select the pattern type you need. You can also search by entering a pattern type name or description in the search box.
3. The selected pattern type is loaded together with the basic patterns during simulation.

The basic patterns (Core) are always included automatically. Selecting additional patterns is optional.

Pattern type list by category

Category	Pattern type
Web Servers	haproxy, httpd, iis, nginx, squid
Application Servers & Frameworks	java, log4j, rails, ruby, springboot

Category	Pattern type
Databases	mongodb, mysql, oracle, postgresql, redis
Cloud	aws
System & Infrastructure	docker, kubernetes, linux-syslog
Messaging & Search	elasticsearch, kafka
Network & Security	bind, bro, firewalls, junos, zeek
Mail	exim, postfix
Monitoring & Backup	bacula, mcollective, nagios
Build Tools	maven

Usage example

To parse a Tomcat Catalina 8 log, select the **java** pattern type additionally and then use the `%{CATALINA8_LOG}` pattern.

Sample log

```
31-Jul-2020 16:40:38 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Server version name: Apache Tomcat/8.5.57
```

GROK parsing pattern

```
%{CATALINA8_LOG}
```

Tag extraction

```
- log_timestamp : 31-Jul-2020 16:40:38
- log.level : INFO
- java.log.origin.thread.name : main
- java.log.origin.class.name : org.apache.catalina.startup.VersionLoggerListener
```

```
- log.origin.function : log
- log_message : Server version name: Apache Tomcat/8.5.57
```

Viewing pattern details

You can check which patterns are included in the selected additional pattern type in the details window. You can look up the pattern definitions (regular expressions) and sample logs and apply them directly to the simulation.

Viewing pattern details

After selecting an additional pattern type, click the selected pattern type tag to open the details window on the right. The details window is displayed side by side without hiding the simulation form, so you can proceed with the simulation while checking the pattern definitions.

You can use the following functions in each pattern item.

Button	Function	Description
Sample	Enter a sample log	Automatically enters an example log that matches the pattern into the **Simulation parsing log** input box.
Select	Insert a pattern	Inserts it into the **Pattern** input box in the `%{pattern_name}` format. The corresponding pattern type is also selected automatically.
Copy	Copy to clipboard	Copies the `%{pattern_name}` format to the clipboard.

- When you click a pattern item, its pattern definition (regular expression) expands. You can expand multiple patterns at the same time to compare their definitions.
- In the `%{PATTERN:field}` references included in a pattern definition, the pattern name, field name, and literal characters are displayed distinguished by different colors.
- You can run the **simulation** directly with the log entered by the **Sample** button to check the parsing result.

Saving simulation sample logs

The log used in the simulation is saved automatically together with the parser settings. When you edit a parser, the sample log used previously is restored automatically to the simulation window, so you do not need to enter the log again each time.

JSON parser

If logs are collected in JSON format, they can be easily parsed using the JSON parser.

Applying the JSON

Log Configuration > **Log primary parser setting**

1. To apply the JSON pattern parser, go to the **Log primary parser setting** tab in **Log Configuration**.
2. Select **+ Add** and then select the **JSON** parser in the **Parser** field.
3. When you select **Register pattern**, the pattern registration and simulation window appears on the right.
4. Enter **Pattern** and **Log**, and then click **Simulation** to check whether parsing is successful with the pattern to apply.

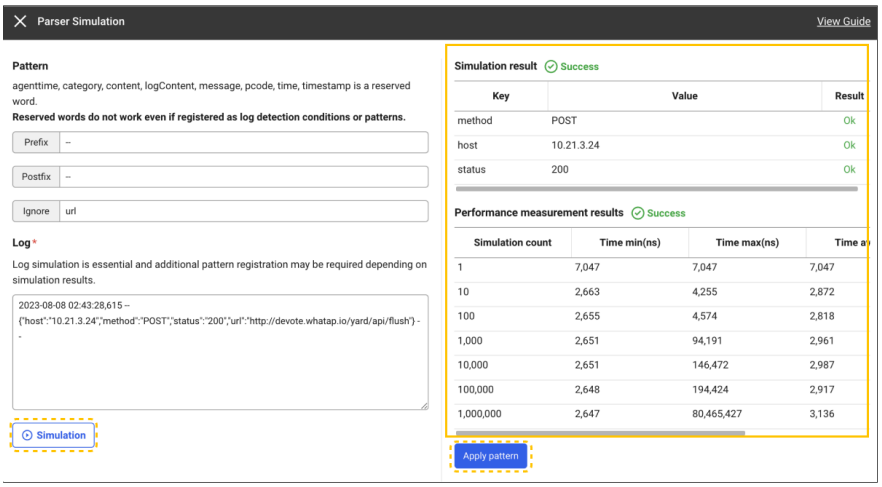
   Example

   ```
   2023-08-08 02:43:28,615 -- {"host":"10.21.3.24","method":"POST","status":"200","url":"http://dev.whatap.io/yard/api/flush"} --
   ```

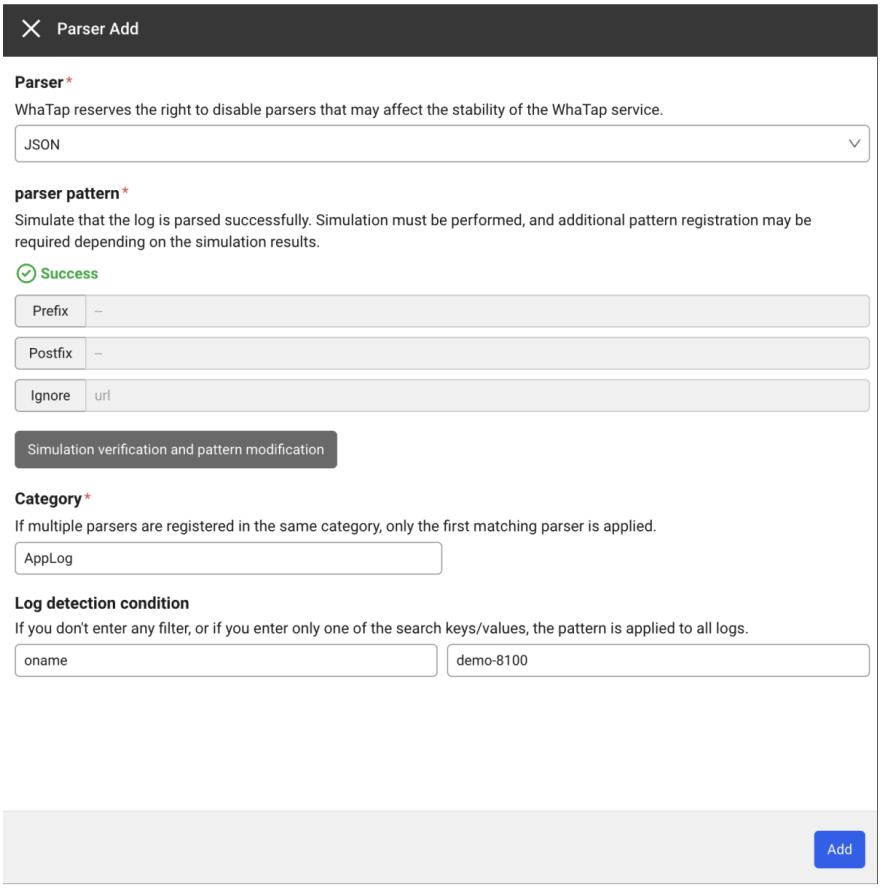
   In the example log, if **Prefix** and **Postfix** are specified as `--` and `url` is specified in **Ignore**, only three **Tag** (`host` , `method` , `status`) are created.

   - **Prefix** Specify the position where the JSON format begins in the log. If the entire log is in JSON format, set an empty value.
   - **Postfix** Specify the position where the JSON format ends in the log. If the entire log is in JSON format, set an empty value.
   - **Ignore** Specify the key for which tags are not created during JSON formatting.

5. If the simulation is successful, you can view **Simulation result** and **Performance measurement results**.



6. When you click **Apply pattern** after simulation, the pattern that has been entered for the selected parser is applied.
7. After applying the pattern, enter values for **Category**, **Log detection condition**, and **Pattern**.



- **Category**

  Select a log category. **Category** is required.

- **Log detection condition**

  - Only the logs that meet the conditions are applied to the parser.
  - Select or enter values for **Search key** and **Search value**.
  - **Log detection condition** are applied before all parsers are run. Accordingly, you cannot use the **Tag** that have been added for the parser.

8. Select **Add** to register a parser.

- In the log parser list, you can change the **Priority** or **Enable**, **Edit**, or **Delete** the parser.
- Before registering a parser, you can check whether the pattern to register is normal through

> **Simulation**. It is the same as the GROK parser registration simulation process. See the following.

Example of using JSON format

Sample log

```
{"host":"10.21.3.24","method":"POST","status":"200","url":"http://dev.whatap.io/yard/api/flush"}
```

If the above sample log has been collected, select the **JSON** parser in the **Parser Add** window. Without having to write a complex parsing logic, you can extract the **Tag** for log analysis as follows:

Tag extraction

```
- host : 10.21.3.24
- method : POST
- status : 200
- url : http://dev.whatap.io/yard/api/flush
```

Usage example when configuring part of the JSON format

Some JSON format sample log

```
2023-08-08 02:43:28,615 -- {"host":"10.21.3.24","method":"POST","status":"200","url":"http://dev.whatap.io/yard/api/flush"} --
```

If part of the log is in JSON format as in the example, specify **Prefix** and **Postfix**. WhaTap log monitoring recognizes and parses the area between **Prefix** and **Postfix** in JSON format.

Tag extraction

```
- host : 10.21.3.24
- method : POST
- status : 200
- url : http://dev.whatap.io/yard/api/flush
```

Log Fast Index Setting

Fast Index Setting

Select the **Fast index setting** tab at the top of **Log Configuration**. Collecting a large number of logs can significantly decrease the log search performance. The frequently used search conditions are created as **index**, you can improve the log search performance for quick search. The setting items are as follows:

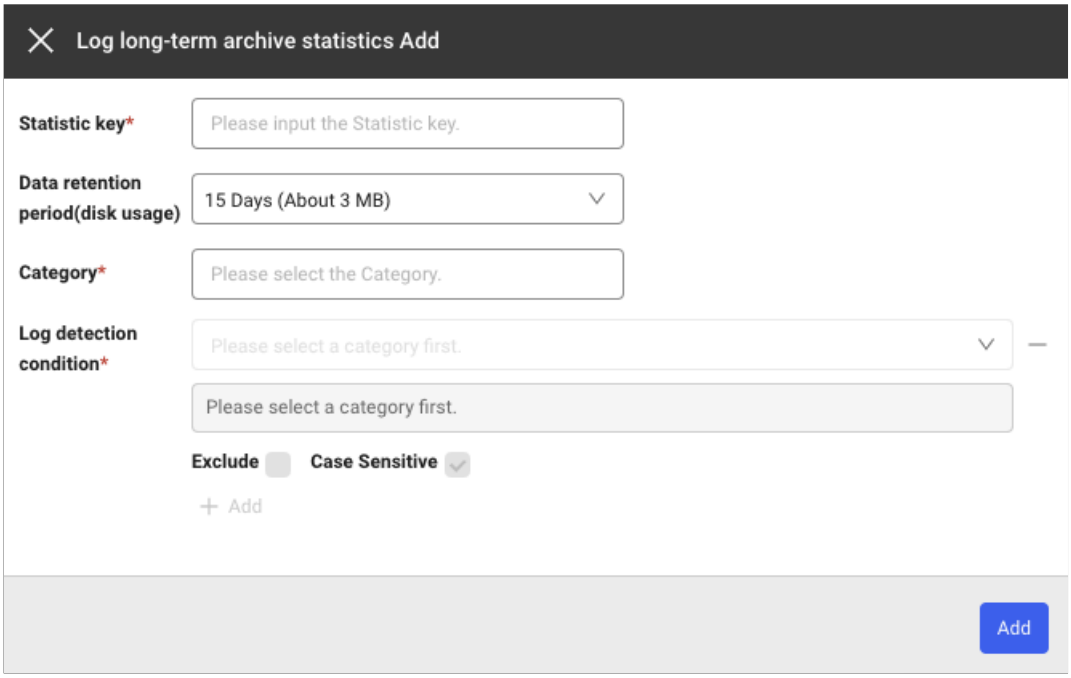
Set value	Required	Description
Category	Mandatory	Category to be set as fast index
Search Key	Mandatory	Search key for fast index setting
Case insensitive	Option	Whether to be case sensitive
Rule	Mandatory	`*` must be included at least one.
Enabled	Mandatory	Active or inactive (default value is `true`)

Log Long-term Archive Statistics Setting

Log long-term archive statistics

Select the **Log long-term archive statistics** tab at the top of **Log Configuration**. Log data is so large and difficult to retain for a long time. Using the Set log statistics data function, you can **save information on how many logs that meet specific conditions are collected every 5 minutes**. Even if actual log data has been deleted for a long time, you can check the trend of how many logs that meet the conditions are collected.

Adding the log long-term archive statistics



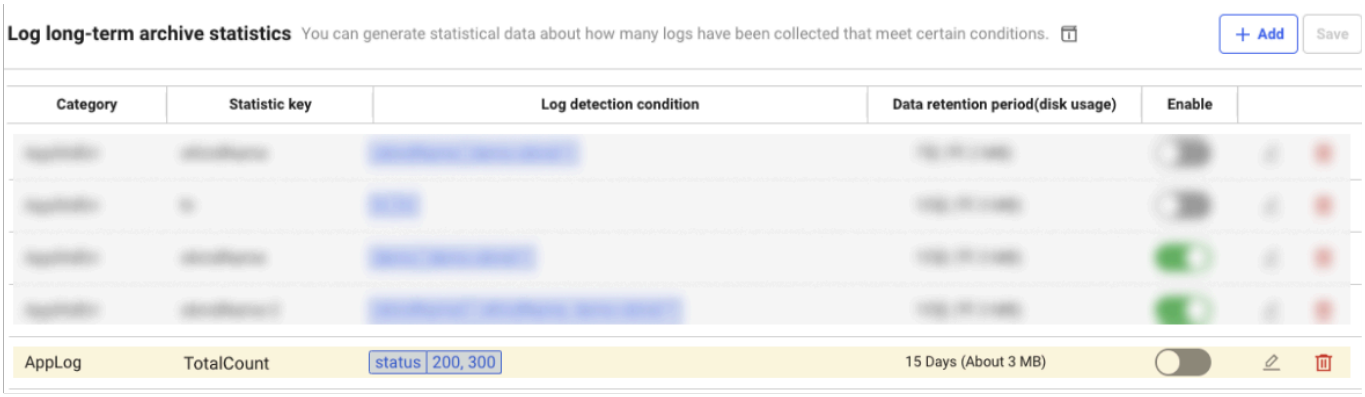
If you select **+ Add** under the **Log long-term archive statistics** tab, the **Log long-term archive statistics Add** window appears. You can add rules by using **+ Add** or delete the created rules by using the **-** icon.

Setting item

Field	Description
Category	Category to apply the rule to.
Statistic Key	The same key cannot be set twice to save when a log that meets the rule is generated.
Log detection condition	Condition for generating log statistical data. Statistical data is generated based on how many logs that meet this condition are collected.
Exclude	If checked, statistical data is generated with values that do not correspond to the entered conditions.
Case Sensitive	Specify case sensitivity for the values of the entered log detection conditions.
Enabled	Active or inactive (default value is `true`)

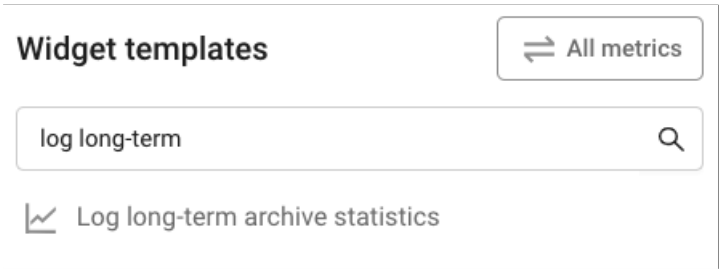
Example

If a setting is added as follows, statistical data is generated with a key value of **TotalCount** for the logs whose status is `200` or `300`.

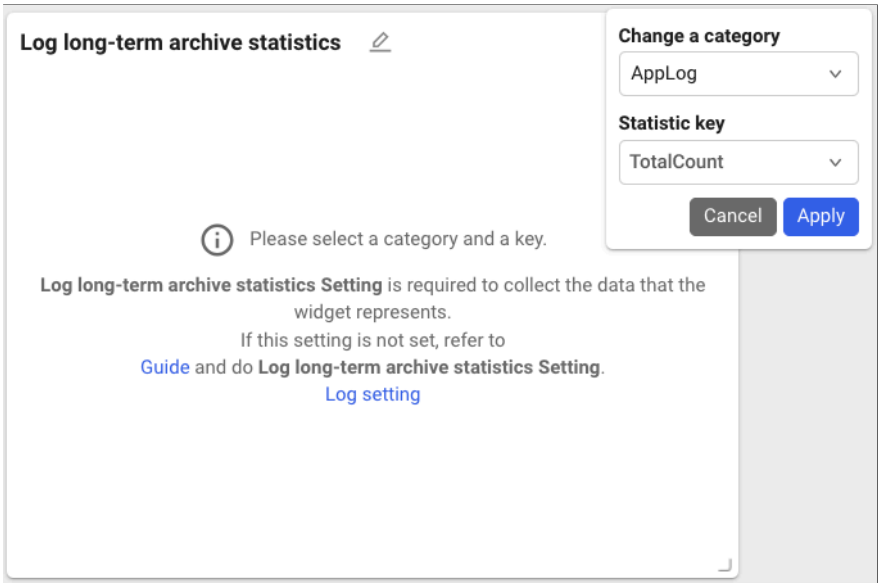


Data Search

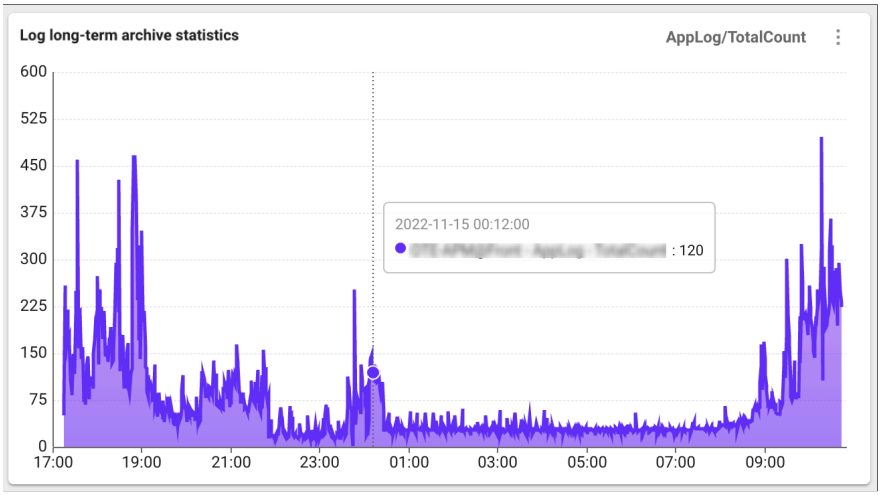
1. Create a widget by searching **Log long-term archive statistics** in **Widget templates** of **Integrated Flex Board**.



2. Enter the **Category** and **Statistic key** to view and then select **Apply**.



3. With the added settings, you can check the **Log long-term archive statistics** data as follows.


Log long-term archive statistics
AppLog/TotalCount
600
525
450
375
300
225
150
75
0
17:00
19:00
21:00
23:00
01:00
03:00
05:00
07:00
09:00
2022-11-15 00:12:00
: 120

Log Personal Information De-identification

Personal information de-identification

In **De-identification of personal information**, you can mask personal information in log data or replace it with safe values.

- Encryption: Specified sensitive data is encrypted and stored in the database.
- Masking: When displayed on the log screen, sensitive data is masked (`***`) by default.
- Substitution: Sensitive data is replaced with a user-specified value that is safe.

Personal information classification

The items classified as personal information are as follows: If there are any items that need to be treated as sensitive data in addition to personal information, mask or replace the desired items in the log data by using the **De-identification of personal information** feature.

- National ID number
- Passport number
- Driver's license number
- Foreigner registration number
- Credit card number
- Bank account number
- Biometric information

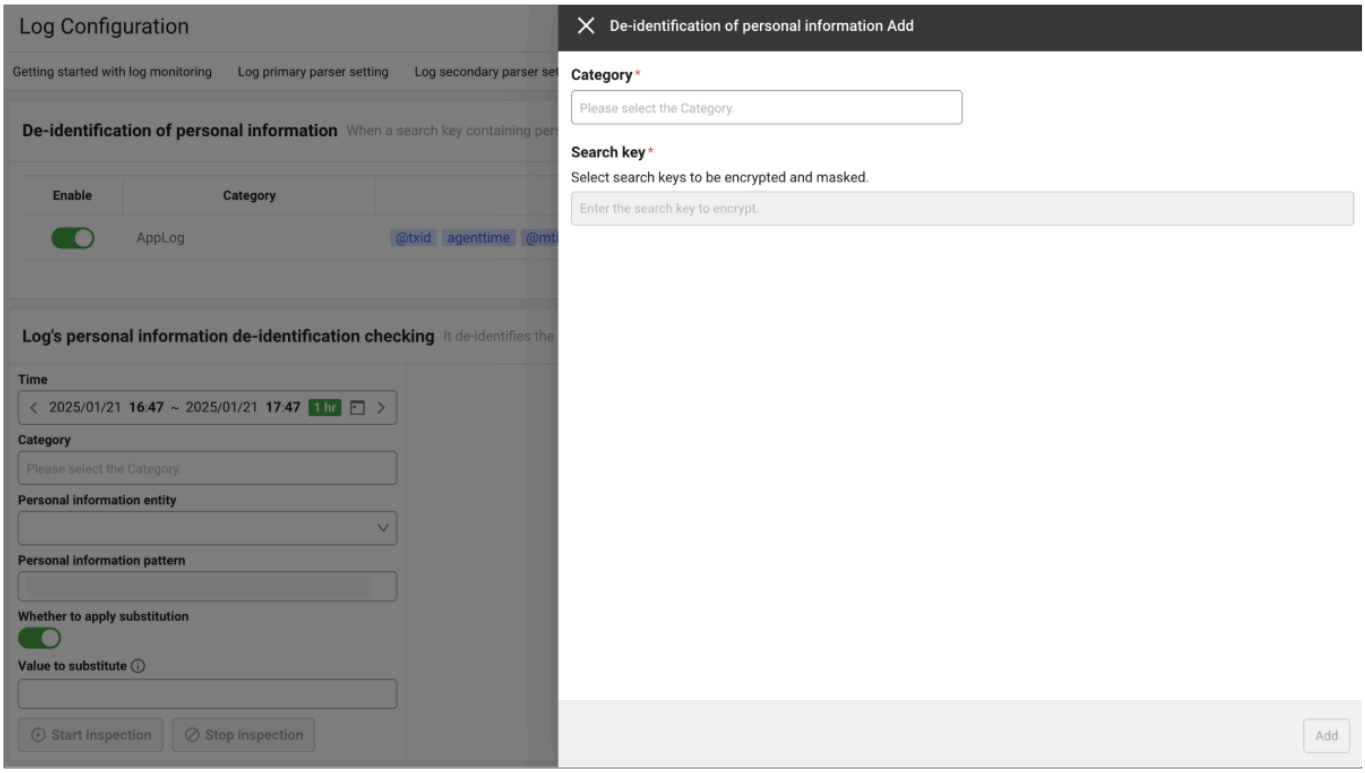
Required roles

De-identified personal information is displayed as masked. Only users with the **log personal information query** role can view original data through the **Display personal information** toggle. Adding or modifying de-identification target search keys requires the **log editing** role.

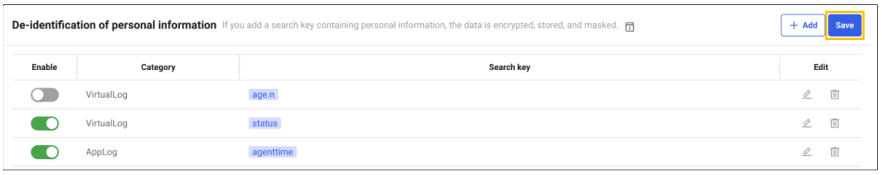
> For more information about the **log personal information query** and **log editing** roles, see the following.

Setting the personal information de-identification

You can protect sensitive information by applying masking to new log data.



1. In the **De-identification of personal information** tab, select **Add**.
2. In the **De-identification of personal information Add** pane on the right, select **Category** and enter the **Search key** to mask.
3. Click **Add** to add the setting.
4. After checking the de-identified items in the **De-identification of personal information** list, click **Save** on the upper right to save the current settings.

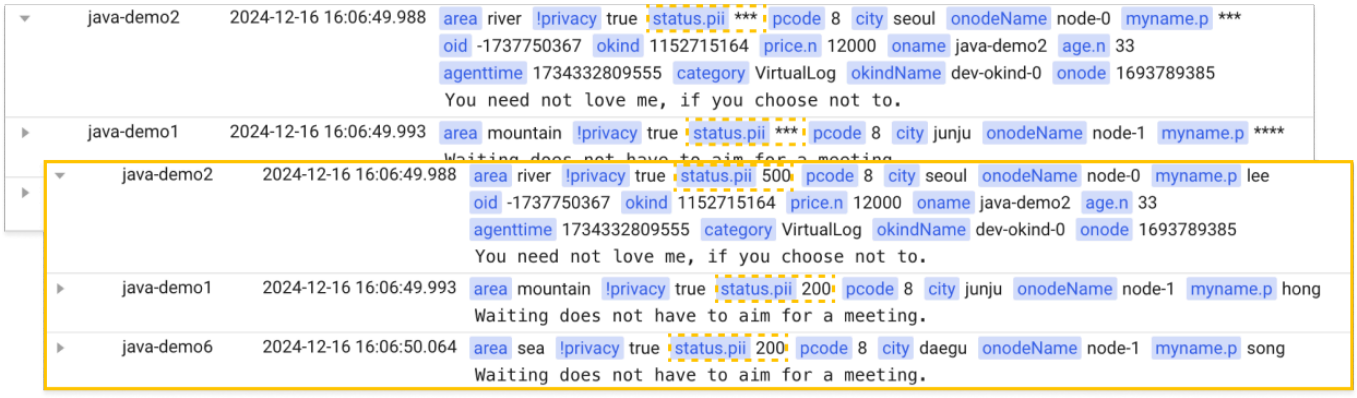


- The de-identified search keys are output in the format of `search key.pii` in the log content.
- You can perform **Enable** for de-identified items in the **Enable** column of the **De-identification of personal information** list.
- You can modify and remove in the **edit** column of the **De-identification of personal information** list.

> If the search key name is specified as `search-key.p` in the **Log primary parser setting** tab, it is automatically registered as a de-identification target. You can check or disable auto-registered items in **Specified search key** after selecting the **Category** in the **De-identification of personal information** tab.
>
> We recommend configuring de-identification directly in the **De-identification of personal information** tab rather than using the `.p` suffix in parser settings.

Personal information display

De-identified data is displayed as masked in the **Live Tail**, **Log Trend**, and **Log Search** menus. Users with the **log personal information query** role can click the **Display personal information** button in the log list to view the original data.



> The search key specified in the **De-identification of personal information** tab appears in the log content of the Log menu in the format of `Search key.pii`.
> `Example of status.pii` `***`

Checking the log's personal information de-identification

> If you replace the existing data using the **Log's personal information de-identification checking** feature, the data cannot be recovered.

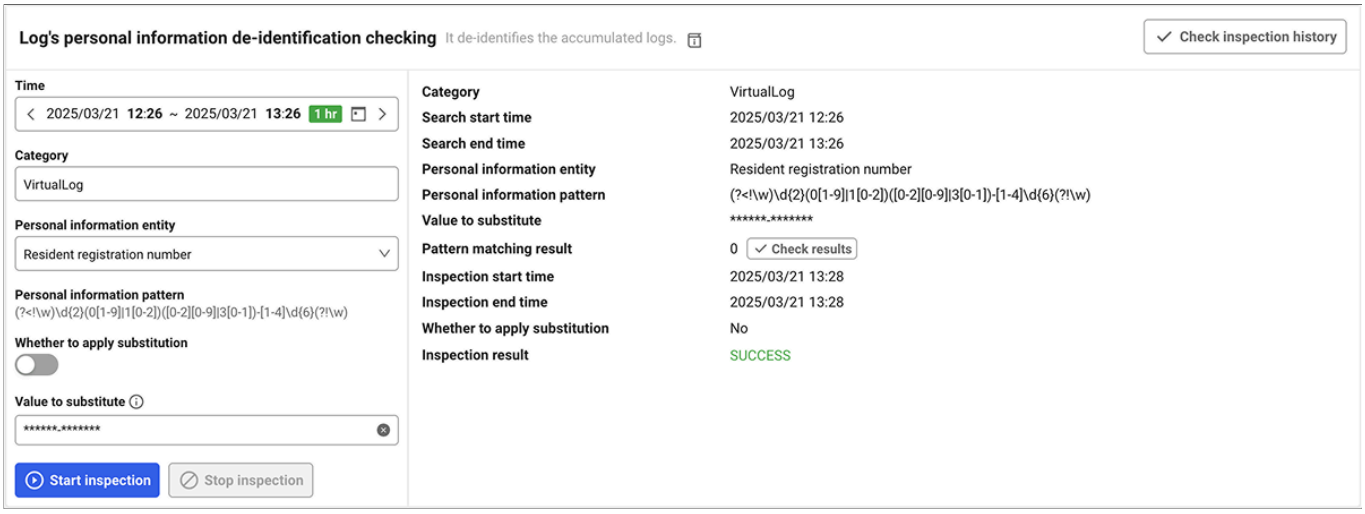
Log's personal information de-identification checking identifies personal information in existing log data and replaces it with the **Value to substitute** specified by the user.

Replacement by target key

If you specify a target key, only the value stored in that key is replaced, and the rest of the log content is kept as is. The target key applies to both the `field` value and the `tag` value of a log. If you do not specify a target key, personal information is identified and replaced across the entire log content, the same as before.

> Replacement applies only to the stored log data. The original value may remain in the search index even after replacement.

You specify the target key in the **Target key** input field of the log personal information de-identification check form. On the screen, `field` and `tag` are not distinguished and are entered as a single target key.



1. Specify the start time and end time.
2. Set values for **Category** .
3. Select **Personal information entity** from the examples or **Direct Input**.

   > When selecting **Direct Input**, directly enter a regular expression pattern to identify personal information.

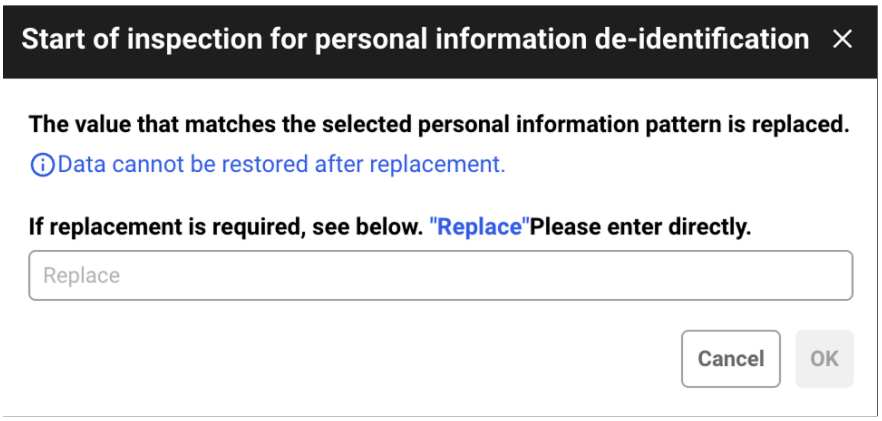
4. Select whether to replace personal information, if it is identified via the **Whether to apply substitution** toggle.
5. Enter **Value to substitute** to change the string identified by the regular expression pattern in the existing log.

   > - **Value to substitute** must not be longer than the original value.
   > - When enabling the **Whether to apply substitution** toggle, be sure to enter **Value to substitute**.

6. Enter the conditions and then select **Start inspection**.

   > You can stop this step at any time by clicking **Stop inspection** during the checkup.

7. When applying substitution, enter **Replace** in the **Start of inspection for personal information de-identification** confirmation window, and then click the **OK** button.



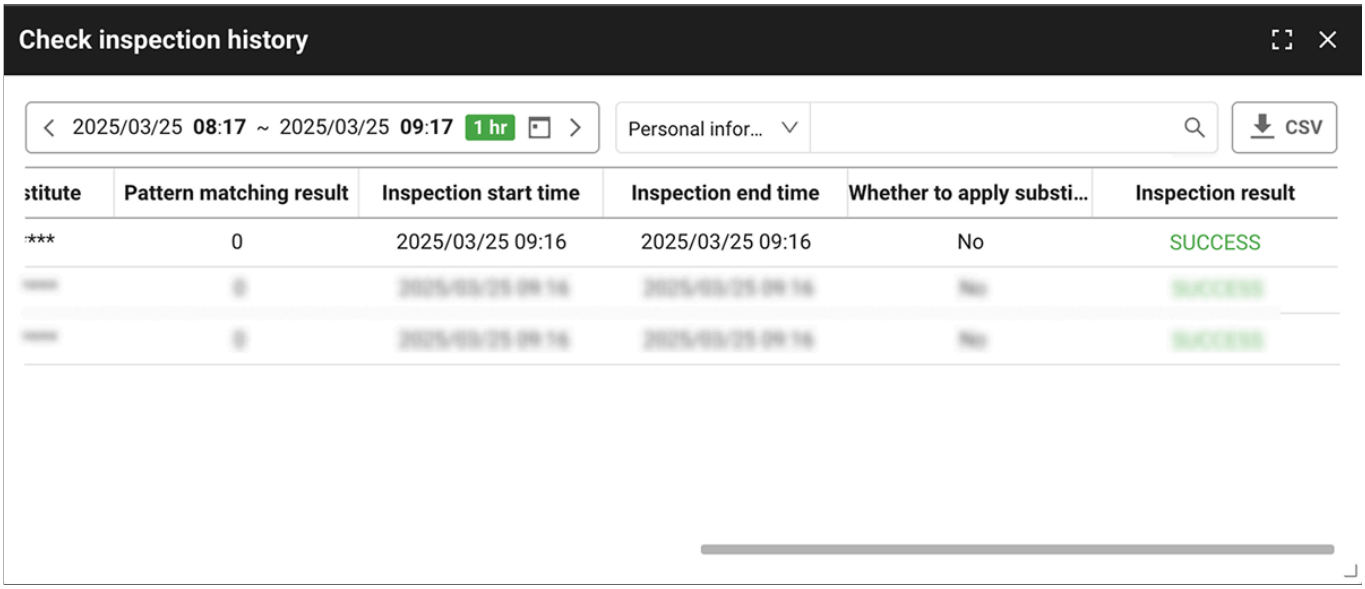
8. After the inspection is finished, you can see **Inspection result** on the right as in the example image.

   - **Category**: It displays the categories checked.
   - **Search start time** and **Search end time**: It displays the time range for the checkup.
   - **Personal information entity**: It displays the personal items that have been checked.
   - **Personal information pattern**: It displays the regular expression pattern used.
   - **Value to substitute**: It displays the applied replacement value.
   - **Pattern matching result**: The number of personal data identified is displayed. You can check the sample logs by clicking **Check results**.
   - **Inspection result**: It displays the result where personal information has been identified.

     `SUCCESS` (check success), `FAILED` (check failure), `STOP` (check stop), `TIMED_OUT` (check ended due to timeout)

Check inspection history

To view the results of past checks after performing de-identification check for the log's personal information, click the **Check inspection history** button at the upper right. You can trace the substitution history, inspection time, and applied personal information items. You can also download the list information in CSV file format by clicking the CSV button at the upper right of the **Check inspection history** window.



Personal information pattern

Examples of regular expression patterns used to identify personal information in the existing log are as follows:

Personal information pattern	Personal information example	Regular expression pattern	Replacement value
National ID number	123456-1234567	(?<!w)d{2}(0[1-9]|1[0-2])([0-2][0-9]|3[0-1])-[1-4]d{6}(?!w)	`******_*******`
Passport number	M1234567	([MSRODTC][0-9]{7}\b)	`********`
Driver's license number	01-23-456789-00	(?<!w)d{2}-d{2}-d{6}-d{2}(?!w)	`**-**-******-**`
Foreigner registration number	987654-1234567	(?<!w)d{6}-?[5-8]d{6}(?!w)	`******_*******`

Personal information pattern	Personal information example	Regular expression pattern	Replacement value
Credit card number	1234-5678-9012-3456	(?<!w)d{4}-d{4}-d{4}-d{4}(?!w)	`****-****-****-****`
Mobile phone number	010-1234-5678	(?<!w)(+82-?1[016789]-?d{3,4}-?d{4}|01[016789]-?d{3,4}-?d{4})(?!w)	`***-****-****`
Email address	`example@domain.com`	([a-zA-Z0-9][a-zA-Z0-9._%+-]*@[a-zA-Z0-9.-]+.[a-zA-Z]{2,})	`***@***.***`
Bank account number	123456-01-654321	(?<!w)d{6}-d{2}-d{6}(?!w)	`******-**-******`
Business registration number	111-11-01111	(?<!w)d{3}-d{2}-d{5}(?!w)	`***-**-*****`
Corporate registration number	111111-1111111	(?<!w)d{6}-?d{7}(?!w)	`******-*******`

Importing And Exporting Log Settings

Importing/exporting log settings

You can save parser settings, Fast index setting, and filter settings in JSON file format, and apply them by importing a JSON file from other projects. You can reduce the hassle of creating the settings repeatedly for each project.

1. Add parser settings, Fast index setting, and filter settings to one project.
2. Select **JSON** on the upper right of each configuration tab.
3. On the upper right of the **Export JSON** window, select **Export**.
4. The JSON configuration file is saved on your PC.
5. Move to another project and then go to the **Log** > **Log setting** menu.
6. Select the configuration tab where you have exported the JSON configuration file before, and then select .
7. If the file selection window appears, select the JSON configuration file saved on your PC.
8. If the **Import JSON** window appears, check the configuration file and then select **Add to list** or **Overwrite**.
9. Select **Save** on the upper right of the screen.

> After importing the JSON configuration file, you cannot save the imported settings unless you select **Save**.

Alert Notifications

This feature delivers notifications to users based on predefined conditions when issues occur in monitored systems or applications. Notifications can be sent through various channels such as email, SMS, messengers, and mobile push notifications.

For example, you can configure alert thresholds for key performance indicators such as CPU usage, memory usage, disk I/O, network traffic, transaction count, error occurrences, and response time latency. When these thresholds are exceeded or abnormal behavior is detected, notifications are automatically sent.

With WhaTap's AI-based alerting, you do not need to define complex conditions manually. The AI learns normal application behavior patterns and automatically detects and notifies anomalies. Additionally, by using tag-count alerts, you can define conditions on values aggregated by specific tags, enabling precise detection even in complex scenarios.

Key Features

- **Event Configuration**: Configure threshold-based alert conditions and notification channels
- **Event Notification Settings**: Configure notification channels and time schedules per user
- **Event History**: View the history of alert occurrences

Permissions

To add or configure events, the **Alert Configuration** permission is required.

- **View**: To view metrics event configurations, the **View & Analyze** permission is required.
- **Add/Edit/Delete**: To modify configurations, the **Alert Configuration** permission is required.
- **Receive Notifications**: To receive notifications, the **Alert Receive** permission is required.

> ⓘ **Reference**
>
> For more details, see Member Permissions.

Event Configuration

Home screen > Select project **Alert** > **Event Configuration**

In **Event setting**, you can configure and manage events based on anomaly conditions that may occur in monitored systems or applications. When the configured conditions are met, notifications can be delivered through your preferred channels such as email, SMS, messengers, or app push notifications. You can tailor event configurations according to your monitoring objectives.

Metrics

Metrics events allow you to monitor system and application performance indicators based on real-time data.

Real-time logs

Sends alert notifications when event conditions configured in real-time collected logs are detected.

Composite log

Configure composite log events that trigger alert notifications based on log content.

Home Screen > Select Project > **Alert** > **Event Configuration**

Event setting provides features to configure and manage events based on anomaly signals that may occur in monitored systems or applications. When configured conditions are met, you can receive notifications through your preferred channels such as email, SMS, messengers, or app push notifications. You can design event configurations according to your monitoring objectives.

Metrics

Metrics events allow you to monitor system and application performance indicators based on real-time data.

Real-time logs

Sends alert notifications when event conditions configured in real-time collected logs are detected.

Composite log

Configure composite log events that trigger alert notifications based on log content.

Reference

- When configuring events, use appropriate threshold values to prevent unnecessary event generation.
- When setting up notifications, consider enabling recovery notifications (alerts sent when issues are resolved) to support effective operational response.

Common Features

All event configurations can be downloaded in **JSON or Excel format**. By sharing the same type of product configuration, you can reduce repetitive event setup work.

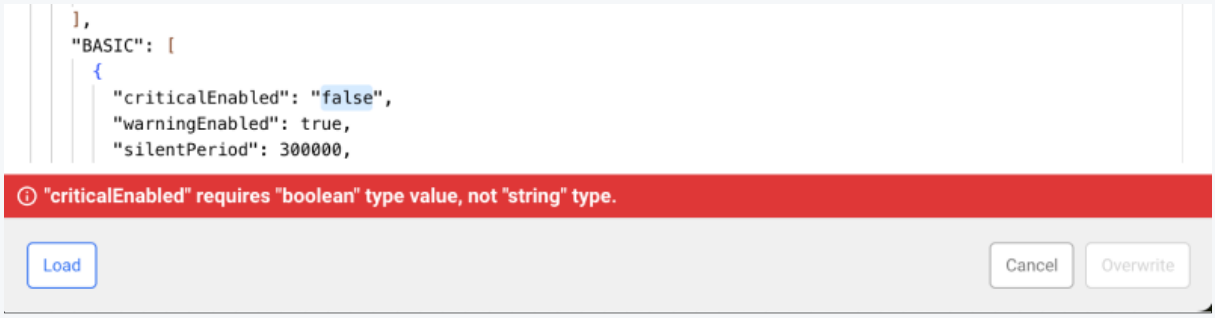
- **Batch JSON Edit**: Edit all event conditions in bulk using JSON format.
- **Batch JSON Download**: Save all currently registered event conditions as a JSON file.
- **Batch Excel Download**: Download the event list as an Excel file for external reporting or backup purposes.

Batch JSON Edit

1. Click the [Batch JSON Edit] button at the top of the screen.
2. Import a JSON file downloaded via the **Batch JSON Download** feature, or edit the content directly.
3. Click the [Overwrite] button to save.
4. When a confirmation message appears, click the [Overwrite] button.

After overwriting, you cannot restore the previous event configuration.

- Arbitrarily modifying the values of the `productType` or `platform` properties may cause events to malfunction in other projects.
- The JSON editor includes built-in validation. If the format or values are invalid, an error message appears below the editor and the [Overwrite] button is disabled.



JSON Data Structure

```
{
 "productType": "{productType}",
 "platform": "{platform}",
 "events": {
  "HITMAP": {
   ...
  },
  "LOG_REALTIME": [
   {...}
  ],
```



```
    "METRICS": [
      {...}
    ],
    "COMPOSITE_LOG": [
      {...}
    ],
    "BASIC": [
      {...}
    ],
    "COMPOSITE_METRICS": [
      {...}
    ],
    "ANOMALY": [
      {...}
    ]
  }
}
```

JSON Field	Description
productType	Product type (e.g. APM, Infra)
platform	Platform information (e.g. java, node)
events	Event data
◈ HITMAP	Hitmap pattern
◈ LOG_REALTIME	Real-time log events
◈ METRICS	Metrics
◈ COMPOSITE_LOG	Composite log events
◈ BASIC	Basic events
◈ COMPOSITE_METRICS	Composite metrics

JSON Field	Description
ANOMALY	Anomaly detection

Batch JSON Download

Click the [Batch JSON Download] button at the top of the screen.

- The JSON file name follows the format integrated-event- `YYYYMMDD` .json.
- The structure of the JSON data and the sub-properties under the `events` attribute may vary depending on the product (`platform`) or product type (`productType`).

> If you arbitrarily modify the values of the `productType` or `platform` properties in the JSON data, events may not function correctly in other projects.

Batch Excel Download

Click the [Batch Excel Download] button at the top of the screen.

Metrics

Home > Select Project > **Alert** > **Event Configuration** > **Metrics** tab

You can view the complete list of all metrics-based events configured for the project. You can review, edit, or delete existing metrics events, or add new events as needed.

Default Options

- **Event Level**: Indicates the severity of an event and is classified into three levels: Normal (Info), Warning (Warning), and Critical (Critical).
- **Duplicate Agent Alert Suppression**: If two or more agents with the same name run simultaneously, events are not triggered for the selected duration (30 seconds, 1 minute, 3 minutes, 5 minutes, 10 minutes).
- **Search**: Search events by event name.
- **JSON Download/Upload**: Edit event rules directly in the JSON editor, or download them as a file, modify them, and upload them again.
- **+ Add Event**: Add a new metrics event.
- **Resize Columns**: Adjust column widths by dragging column borders.

Table | Metrics Event List Structure

Item	Description
No.	Event sequence number
	Event activation status – Enabled: An event is triggered when conditions are met – Disabled: The event does not run and conditions are not evaluated
	Edit or delete event settings
Event Name	User-defined event name – Up to 255 characters

Item	Description
Rule	Event trigger conditions
Target	Monitoring targets to which the event applies
Trigger Count	The time window and number of occurrences required to trigger an event – An event is triggered when conditions are met the configured number of times within the selected duration
Pause	Temporarily suppresses the same event after an alert is triggered – Available durations: Disabled, 5 min, 10 min, 15 min, 20 min, 30 min, 1 hr, 2 hrs, 3 hrs, 6 hrs, 12 hrs, 1 day – If recovery notifications are enabled, alerts are not triggered for the selected duration after a RECOVERED notification is sent
Recovery Notification	Whether to receive a RECOVERED (normal) notification when Critical or Warning events are resolved – Enabled: Sends a RECOVERED notification when the status changes from in progress to normal and allows viewing active and resolved states in event history – Disabled: Triggers a one-time event each time the threshold is exceeded
Event Recipients	Users or groups that receive notifications for the event (recipient tags)

Event Configuration

Metrics events allow you to define events based on conditions that occur in monitored targets and configure notifications so you can respond efficiently.

Add Event

To add a metrics event, go to **Event Configuration** > **Metrics** tab, then click the [+ Add Event] button in the upper-right corner.

How to Add a Metrics Event
Step 1. Select Template: Choose the basic structure of the event.
Step 2. Define Event Conditions: Configure the conditions that determine when the event is triggered.
Step 3. Select Event Target: Select the monitoring targets where the event applies.
Step 4. Basic Information and Notification Settings: Configure the event name, message, and recipients.

Select Template

1. Depending on how you want to configure the event, select one of the three templates on the left side of the screen.
   - **Create New**: Freely configure categories and metrics from scratch. This option is suitable for advanced users who need complex conditions or fine-grained monitoring rules.
   - **Quick Setup**: Provides predefined categories, metrics, and default thresholds. You can apply it quickly by adjusting only the threshold values, which makes it suitable for beginners.
   - **Advanced Setup**: Add conditions or freely modify settings based on predefined categories and metrics.

> ⓘ Metrics events support multiple configuration approaches. The available templates may vary depending on the monitoring platform.

Define Event Conditions

In the **Define Event Conditions** step, configure the metrics that determine when the event is triggered, along with the trigger count, pause settings, recovery notifications, and event active time.

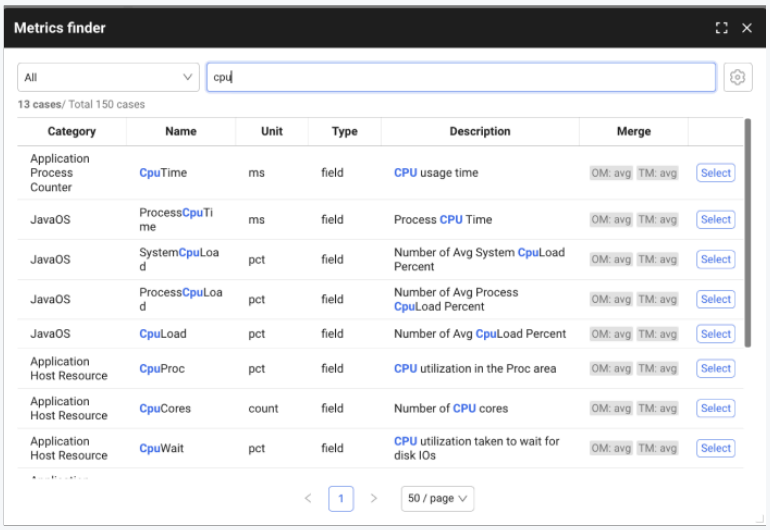
Metric Settings

1. Enter a category in the search field, or select a category (metric) from the category list.
   You must select a category to proceed with metric configuration.

- The category list shows the category name, data collection interval, and key. It displays metrics data collected in the project within the last 3 hours.

2. If the metric you need is not listed, select **Direct Input** and enter the category manually.

> **If you are not sure about categories or metrics, use the Metrics Finder.**
>
> When configuring metrics for the first time, the Metrics Finder helps you locate metrics more quickly.
>
> - Search for a metric in the Metrics Finder to view related categories and detailed information.
> - Click the [Select] button for a metric in the search results to automatically apply the selected value to the event configuration screen.



3. Select the event level (Critical, Warning, Info). You can configure multiple levels for a single metric.

> **Event Level Behavior**
>
> Even when multiple conditions are satisfied, only the highest-priority level (Critical → Warning → Info) triggers an event. For details, see the Level Action Guide.
>
> - Event levels are classified as Critical (Critical), Warning (Warning), and Info (Info).

> ⓘ
> - It is recommended to configure multiple levels for the same metric with stepwise thresholds.

4. Enter the field, operator, and value for the selected event level.

   a. Click **+ Add** to add more conditions.

   - `&&`: An event is triggered only when **all conditions are satisfied simultaneously**
   - `||`: An event is triggered when **at least one condition is satisfied**

   b. Click **—** to remove the condition.

> ⓘ For condition syntax and supported operators, refer to the Condition Settings Guide.

Trigger Count

Configure how many times a condition must be met before an event is triggered. This is useful for detecting persistent issues only.

1. Select the trigger frequency.
   - **Consecutive**: An event is triggered when the condition occurs consecutively for the configured number of times.
   - **Recent**: An event is triggered when the condition occurs the configured number of times within the selected time window.
     - Available durations: `5 seconds`, `10 seconds`, `15 seconds`, `30 seconds`, `1 minute`, `2 minutes`, `3 minutes`, `4 minutes`, `5 minutes`, `10 minutes`, `30 minutes`, `1 hour`
2. Enter the trigger count.
   - The **Interval** value corresponds to the data collection interval of the selected category. (Interval: 10s)

Pause

To prevent excessive event generation, select a pause duration for event execution. After an event occurs, the same event will not be triggered again during the configured time period.

- Available durations: `Disabled` , `5 minutes` , `10 minutes` , `15 minutes` , `20 minutes` , `30 minutes` , `1 hour` , `2 hours` , `3 hours` , `6 hours` , `12 hours` , `1 day`

> **When recovery notifications are enabled**
>
> Even if the same event condition is met again, no notification is sent until the configured time has passed since the **RECOVERED** notification was received.

Resolved Notification

When the resolved notification feature is enabled, you can track state changes after an event occurs (in progress → resolved). The event is shown as an ongoing event in the **Event History** menu.

- Resolved notifications apply only to Critical and Warning levels. Info-level events always operate as one-time events.
- When a Critical or Warning event is resolved, a notification with the RECOVERED (normal) state is sent.
- Click the toggle button to **enable** or **disable** this feature.
  - **Enabled**: Events operate based on state. They enter an in-progress state when the threshold is exceeded and transition to a resolved state when the value recovers below the threshold.
  - **Disabled**: A one-time event is generated each time the threshold is exceeded.

Event Operation Time

Configure tags so that events operate only during specific time periods (such as business hours, non-business hours, or maintenance windows). If no tag is configured, events operate continuously for 24 hours while enabled.

1. Click **+ Add**.
2. In Event Operation Time, click **+ Create New Tag**.

3. In **Create Event Operation Time Tag**, select the **tag name, days, time, and color**, then click the [Apply] button.
4. You can view created tags in the tag list, and apply them by selecting the checkbox.
5. To modify or delete a tag, click the icon and update or delete it in **Edit Event Operation Time Tag**.

> When a tag is deleted, it is removed from all users to whom it was applied. However, tags that are currently in use by event rules cannot be deleted.

Simulation

The simulation feature lets you pretest configured event conditions using historical data. If the expected number of events is too high or too low, you can adjust the conditions to find optimal threshold values.

- **Rule Validation**: Verify how newly created event rules would behave when applied to historical data.
- **Threshold Tuning**: Adjust thresholds to an appropriate level when events occur too frequently or not at all.

Start Simulation

1. After completing event condition settings, click the [Simulation] button.
2. Review the expected number of events and the visualized results.

Understanding Simulation Results

1. **Time-Series Data**

   A graph displays metric changes over time along with threshold lines for each level, allowing you to intuitively identify when events are triggered.

2. **Event Counts by Level**

   The number of events for the Critical, Warning, and Info levels is shown on the right side of the screen. This helps you understand overall event distribution and frequency.

3. **Target-Level Details**

Hover over a specific point on the chart to see the exact metric value and monitoring target at that moment. When multiple targets exist, they are distinguished by color. If patterns differ by target, you can use filtering to analyze them individually.

4. **Peak Interval Identification**

   Identify periods where metrics spike sharply to understand when issues occur most frequently or to recognize recurring patterns.

Simulation Tips

- **Simulate Different Time Ranges**

  Run simulations for weekdays vs. weekends and peak vs. off-peak hours to set optimal thresholds for each time period.

- **Include Actual Incident Periods**

  Include past outage periods in simulations to verify whether current rules accurately detect real incidents.

- **Add Target Filtering**

  When many targets are monitored, simulation results can become complex. Use filters to focus on specific targets for clearer insights.

- **Adjust Thresholds Gradually**

  Instead of making large changes at once, adjust thresholds incrementally, such as in 10 percent steps, to find optimal values.

Select Event Target

If no event target is specified, events are triggered for all targets in the project, which can result in excessive events. It is therefore recommended to clearly define the targets to which the event applies.

If you change the event targets, the number of events may change. Click the **Simulation** button and run it again to check the updated expected event count.

Target Selection

You can specify the targets to which the event applies using either **Select Input** or **Direct Input**.

- Select Input Direct Input

1. Select or enter a tag, operator, and value.
   a. If it is difficult to find a tag, you can search for it using the **Metrics Finder**.
2. Click **+ Add** to add event targets using conditions (`&&` , `||`).

Enter the tag, operator, and value of the target to monitor directly.

- When you choose direct input, a list of available tags (**Tag Select Input**) is provided below the input field.

```
ex. endsWith(okindName, 'example_name') && container == 'prod.billing'
ex. ${4xxErrorType} == '401'
```

> If you change the input method, any content you have entered will be reset.

Basic Information and Notification Settings

Configure the **event name, message, and recipients**. Because this information is displayed as is in event history and actual notification messages, it should be clear and easy to identify.

Event Activation

Configure whether the event is triggered when the defined conditions are met.

- **Enabled**: The event is triggered when the configured conditions are satisfied.
- **Disabled**: Conditions are not evaluated.

Event Name

Enter the event name to be used as the event title. The configured event name is displayed in the **Event List** and **Event History** menus and can be used as a search keyword.

Message

Enter the event message. The message you enter is used in event history and notifications. Click the icon to view previously written message history. You can include variables in the message to insert actual values at the time the event occurs.

- **Variable Usage Rules**
  - Variable format: `${metric_name}`
  - Only metrics within the **same category as the configured category** can be used as variables.
  - You can combine multiple variables to create a more detailed message.
  - The list of available variables can be found in the **Metrics Query** menu.
- **Variable Usage Example**

	Event Configuration	When Event Occurs
Title	Disk Usage Increase	Disk Usage Increase
Message	`Disk = ${disk}%`	`Disk = 89.9%`

Event Recipients

Specify the members who will receive event notifications.

- **Receive All** **Receive by Tag Selection**

Notifications are sent to all targets (members and channels) in the project that have event reception enabled.

Notifications are sent to project members who have the selected tag.

1. Select **Receive by Tag Selection**.
2. In the **Reception Tag** section, click the [+ Add Tag] or [+] button.
3. Under the **Event Reception Tag** window, click **+ Create New Tag**.
4. In the **Create Tag** window, enter a tag name, select a color, and click the [Create Tag] button.
5. You can edit or delete created tags by clicking the icon in the tag list.

6. In the **Event Reception Tag** window, select the desired tag from the **Tag List** to apply it.

> **Reception Tags**
>
> Reception tags are a feature for managing users or channels that receive notifications **at a group level**. When a reception tag is assigned to an event, notifications are delivered only to users associated with that tag. If no reception tag is set, notifications are sent to all users in the project.

Edit/Delete Event

1. Go to **Alert Notifications** > **Event Settings** and open the **Metrics** tab.
2. In the event list, click the icon for the event you want to edit or delete.
3. In the **Edit Event Rule** window, update the options and click the [Save] button.

   a. To delete the selected event, click the [Delete] button in the upper-right corner of the **Edit Event Rule** window.

Event Sharing

You can save metric event settings as a JSON file to share configurations with other users or import settings created by others.

- JSON file name: event-rules-`YYYYMMDD`.json

Export

1. Click the [**JSON**] button at the top right of the screen.
2. When the JSON editor opens, click the [Export] button.
   - If you use the export function after searching for events, only the searched list will be downloaded as a JSON file.
3. After the JSON file is downloaded, share it with other users.

Import

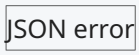
1. Click the [ ] button at the top right of the screen.
2. Select the JSON file downloaded using the **Export** function.
3. When the JSON editor opens, choose either [Add to List] or [Overwrite].

It is recommended to use this feature between projects of the same product type. You can import event settings from projects of different products, but they may not function correctly.

Edit in JSON Format

1. Click the [**JSON**] button at the top right of the screen.
2. When the editor opens, modify the content according to the JSON format.
3. After completing the edits, click the [Save] button at the bottom of the screen.

If the modified content does not conform to the JSON format, an error message will be displayed at the bottom of the screen and the content cannot be saved. The error message may vary depending on the type of formatting issue.



JSON Data Structure

```
{
  "version": 2,
  "eventId": "zcef7s7f27rum1",
  "enabled": true,
  "stateful": false,
  "title": "Active Transaction",
  "message": "Active Transaction = ${active_tx_count}",
  "category": "app_counter",
  "alertLabel": [
    "oid"
  ],
  "repeatCount": 1,
  "repeatDuration": 0,
```

```
  "silent": 300000,
  "receiver": [],
  "timeTag": [],
  "selectString": "",
  "conditions": [
   {
    "level": 20,
    "enabled": true,
    "rule": "active_tx_count > 100"
   }
  ],
  "createTime": 1763632674345,
  "lastModifiedTime": 1763632674347,
  "lastModifiedUser": "support@whatap.io",
  "basic": true,
  "metaId": "java004",
  "selectCondition": {
        "oid": [
          "-1010758404",
          "-250906941"
        ]
  }
}
```


Table | JSON Data Structure

JSON Field	Type	Description	Notes
version	Integer	Version of the event rule	Restricted
eventId	String	Unique identifier of the event rule	
enabled	boolean	Whether the event is enabled	
stateful	boolean	Whether the event is state-based	
title	String	Event rule name defined by the user	
message	String	Event rule message defined by the	

JSON Field	Type	Description	Notes
		user	
category	String	Data category	
alertLabel	`List<String>`	Default identifier (primary key) value assigned per category for event state management	Restricted
repeatCount	Integer	Number of repetitions	
repeatDuration	Long	Duration for repetition	
silent	Integer	Pause duration	
receiver	`List<String>`	List of recipient tag keys	
timeTag	`List<String>`	List of operation-time tag keys	
selectString	String	Target selection	
conditions	`List<Object>`	List of trigger conditions by level	
◈ level	byte	Level value	`Critical`: 30, `Warning`: 20, `Info`: 10
◈ enabled	boolean	Whether the level is enabled	
◈ rule	String	Trigger condition for the level	
createTime	Long	Initial creation time of the event rule	
lastModifiedTime	Long	Last modification time of the event rule	
lastModifiedUser	String	Account that last modified the event rule	

JSON Field	Type	Description	Notes
basic	boolean	Whether the event was created using Quick Setup	Restricted
metaId	String	Unique identifier of the Quick Setup template	Restricted: This value exists only for events created via Quick Setup (basic=true).
selectCondition	Object	Target selection value for Quick Setup	This value exists only for events created via Quick Setup (basic=true).

Fields marked as “Restricted” may affect event behavior, so it is recommended not to modify their values.

Appendix

Trigger Conditions and Target Selection Guide

Metric alert trigger conditions and event target selection use the same syntax. However, trigger conditions use the field key as the variable, while target selection uses the tag key as the variable.

Level Behavior Guide

Event levels are categorized into Critical, Warning, and Info.

1. **Priority-based Event Triggering**

   When multiple level conditions are met at the same time, only the event with the highest priority is triggered.

   Example

```
Configuration:
- Warning: CPU > 70%
- Critical: CPU > 90%

Current state: CPU 95%
→ Result: Only the Critical event is triggered (Warning is suppressed)
```

2. **Level Escalation**

   For events that maintain an in-progress state, when a higher-level condition is met while a lower-level event is already in progress, both levels enter the in-progress state.

   Example

```
Configuration:
- Warning: CPU > 70%
- Critical: CPU > 90%

Scenario:
1) CPU 80% → Warning triggered (in progress)
2) CPU 95% → Warning (in progress), Critical (in progress)
```

3. **Level De-escalation**

   For events with an in-progress state, when the level transitions from a higher level to a lower level, the higher-level event is resolved while the lower-level event remains active.

   Example

```
Configuration:
- Warning: CPU > 70%
- Critical: CPU > 90%

Scenario:
1) CPU 97% → Critical triggered
2) CPU 85% → Critical resolved, Warning remains
3) CPU 65% → Warning resolved
```

4. **Execution Flow**

```
- Warning: CPU > 70%
- Critical: CPU > 90%

Metric value: 60%  →  75%  →  92%  →  85%  →  60%
State:      Info      Warning     Critical     Warning     Info
                   (triggered)  (triggered)  (maintained)  (resolved)
                                Warning
                              (maintained)
```

Events that have an in-progress state remain active as long as their conditions are met and continue until those conditions are cleared.

Condition Configuration Guide

If a metric name starts with a number or contains special characters, you must wrap it in the `${metric_name}` format.

- Brackets `()`, `[]`
- Operators `+`, `-`, `*`, `/`, `%`
- Delimiters `:`, `@`, `#`, `,`, (space)
- Other special characters `!`, `^`, `&`, `|`, `~`, `` ` ``, `=`

Examples using `${}`:

```
${cpu(xos)} > 50
${mem[0]} >= 100
${cpu-usage} > 80
${namespace:cpu} > 70
${metric name} > 60
${4xx_error} > 10
```

Usable without `${}`: Metric names that contain only alphabetic characters, underscores (`_`), or dots (`.`)

```
cpu > 80
cpu_usage > 50
CPUUtilization.Average > 0.8
```

1. Comparison Operators

Numeric Comparison

```
Greater than:          cpu > 80
Greater than or equal:   cpu >= 80
Less than:             memory < 1000
Less than or equal:      memory <= 1000
Equal to:              status == 200
Not equal to:          error != 0
```

String Comparison

```
status == 'OK'
region == "us-east-1"
```

2. Arithmetic Operators

- Basic arithmetic operations

```
Addition:       cpu + 10 >= 90
Subtraction:    memory - 100 >= 500
Multiplication: cpu * 2 >= 100
Division:       disk / 1024 >= 100
Modulo:         value % 10 == 0
```

- Controlling precedence with parentheses

```
(cpu + memory) * 2 >= 200
(disk - used) / total >= 0.2
((cpu + memory) / 2) >= 50
```

- Negative values

```
cpu > -100
```

3. **Logical Operators**

   - AND operator (&&)

   ```
   cpu > 80 && memory > 1000
   ${cpu(xos)} > 50 && ${mem(xos)} > 60
   ```

   - OR operator (||)

   ```
   cpu > 90 || memory > 90
   disk < 10 || network > 1000
   ```

   - Combined logical operations

   ```
   (cpu > 50 && memory > 50) || disk < 20
   cpu > 80 && (memory > 1000 || disk > 500)
   ```

4. **Pattern Matching Operators**

   - LIKE operator

   ```
   oname like 'prod-*'
   url like '*error*'
   message like 'WARN%'
   ```

   - NOT LIKE operator

   ```
   oname not like 'test-*'
   url not like '*debug*'
   ```

5. **Built-in Functions**

   - Null check functions

```
isNull(value)           # Checks whether the value is null
isNotNull(value)        # Checks whether the value is not null
nvl(value, 0)           # Returns a default value if null
isEmpty(str)            # Checks whether the string is empty
isNotEmpty(str)         # Checks whether the string is not empty
```

- Aggregate functions

```
sum(cpu, memory, disk)            # Sum
avg(cpu, memory)                  # Average
max(cpu, memory, disk)            # Maximum
min(cpu, memory, disk)            # Minimum
count(value1, value2, value3)     # Count
```

- Math functions

```
round(cpu, 2)           # Rounds to 2 decimal places
```

- String functions

```
length(str)             # String length
startsWith(str, 'prefix')  # Checks prefix
endsWith(str, 'suffix')    # Checks suffix
indexOf(str, 'search')     # String index
substring(str, 0, 10)      # Substring
trim(str)               # Trims whitespace
replace(str, 'old', 'new') # String replacement
hasStr(str, 'search')      # Checks string containment
```

- Conditional functions
  - **if**: A conditional function that returns the `trueValue` if the condition evaluates to `true`, or the `falseValue` if it evaluates to `false`

    Syntax

```
if(condition, trueValue, falseValue)
```

Example

```
if(value >= 90, 'High', 'Medium') == 'High'
```

- **decode**: Compares a value against conditions and returns the result of the matching condition, or a default value if no match is found

Syntax

```
decode(value, condition1, result1, condition2, result2, ..., conditionN, resultN, defaultValue)
```

Example

```
decode(value, 1, 'Low', 2, 'Medium', 3, 'High', 'Unknown') == 'Unknown'
```

- **in**: Returns `true` if the value matches any of the specified candidates; otherwise returns `false`

Syntax

```
in(value, candidate1, candidate2, ..., candidateN)
```

Example

```
in(status, 200, 201, 204) == false
```

Writing Guidelines

- Recommended
  - Use simple variable names without `${}` (e.g. cpu, memory)
  - Always use `${}` when special characters are included
  - Use parentheses to clearly separate complex expressions
  - Enter function names exactly as defined (case-sensitive)
- Avoid

- Incomplete expressions (e.g. `cpu >`, `memory &&`)
- Mismatched parentheses (e.g. `(cpu > 80 && cpu > 70)`)
- Consecutive or invalid operators (e.g. `cpu >> 80`, `cpu >< 80`)

Real-time Log Events

Home > Select Project > **Alert** > **Event Configuration** > **Real-time Log** tab

You can set alert notifications by filtering collected log data against conditions. Log monitoring must be activated to use log events. For more information on **activating log monitoring**, refer to the log documentation for each product.

- **Real-time log event**: An event is triggered when the configured event conditions are met in real-time collected logs.

Default options

- **Search**: Search for events by selecting an event name or event notification tag
- **JSON download/upload**: Event rules can be directly edited through the JSON editor, or downloaded as a file, modified, and re-uploaded
- **Notification message settings**: Select detailed items for the message to receive and edit the content for each item
- **+ Add event**: Add a new log event

Table | Real-time log event list configuration

Item	Description
Event name	Event name used as the notification title - Up to 255 bytes
Category	Log classification name (log folder name)
Search key	The field to search for in log data - Example: To access an HTTP response status code, use the search key `status`
Search value	The actual value corresponding to the search key. An alert is sent when logs contain the specified value

Item	Description
	- Example: If the search key is `status` and the search value is `200`, an alert is triggered when logs containing HTTP status code 200 are collected
Pause event notifications	Pauses the same event notification for a specified time after an event notification is triggered - Available times: Disabled, 5 min, 10 min, 15 min, 20 min, 30 min, 1 hour, 2 hours, 3 hours, 6 hours, 12 hours, 1 day - If resolved notifications are enabled, no notification is sent for the same event during the selected time after receiving a normal (RECOVERED) status notification
Event target filtering	Conditions that define the targets to which the event applies - If no condition is set, alerts are evaluated for all collected data - Select input: Build conditions by selecting tags, operators, and values - Direct input: Manually enter conditions to specify targets
Event notification tags	User or group (notification tag) to receive the event notification
	Edit and delete events
	Whether the event is active - Enabled: Event occurs when conditions are met - Disabled: Event does not operate and conditions are not checked

Event settings

Log events allow you to configure alert notifications when logs are collected in real time or when logs that meet specific conditions accumulate.

Add real-time log event

1. Click the **Real-time Log** tab in **Alert** > **Event Configuration**.

2. Click the [+ Add event] button on the right side of the screen.
3. Set the log event conditions in the **Add event rule** window.
   - **Event name**: Enter a name to use as the event title.
   - **Event activation**: When the toggle button is activated, an event occurs when conditions are met.
   - **Level**: Select the event level (Critical, Warning, Info).
   - **Message**: Enter the notification message to receive when the event is triggered. You can use variables in the message to include actual values at the time of event occurrence.

     > **Variable usage rules**
     > - Variable format: `${metricName}`
     > - Only metrics within the **same category as the configured category** can be used as variables.
     > - You can combine multiple variables to compose a detailed message.
     > - A list of available variables can be checked in **Metrics Query**.

   - **Category**: Select the log classification name (log folder name) from the list or enter it directly. You must select a category before entering a search key and search value.
   - **Search key**: Enter the field you want to search for in log data.
     - Example: To access an HTTP response status code, use the search key `status`.
   - **Search value**: Enter the actual value corresponding to the **search key**. An alert is sent when logs contain the specified value.
     - Example: If the **search key** is `status` and the **search value** is `200`, an alert is triggered when logs containing HTTP status code 200 are collected.
   - **Event target filtering**: Enter conditions to define the targets to which the event applies. If no value is entered, alerts are evaluated for all collected data.
     - **Select input**: Select a tag, operator, and value to set the target conditions.

       > For condition expression syntax and supported operators, refer to the Condition

> ⓘ settings guide.

  - **Direct input**: Enter conditions directly.

- **Pause event notifications**: Select a duration to temporarily suppress alerts. After the event is triggered, the same event will not occur during the selected time.
  - Available times: `Disabled` , `1 min` , `2 min` , `3 min` , `5 min` , `10 min` , `20 min` , `30 min` , `1 hour` , `3 hours` , `6 hours` , `12 hours`
- **Event operation time**: Set tags to activate the event during specific time periods (work/non-work/maintenance hours, etc.). If no tag is set, the event operates 24 hours a day when activated.
  1. Click **+ Add**.
  2. Click **+ Create new tag** in **Event operation time**.
  3. In **Create event operation time tag**, select the **tag name, day of week, time, and color**, then click the [Apply] button.
  4. Created tags can be viewed in the tag list and are applied by selecting the checkbox.
  5. Click the ✎ icon for the tag to modify or delete, and edit or delete the tag in **Edit event operation time tag**.

> ⚠ Deleting a tag removes the tag for all users to whom it has been applied. However, tags in use by event rules cannot be deleted.

- **Event notification tags**: If notification tags are set for an event, notifications are sent to project members with those tags. If no recipient tags are set, notifications are sent to all users in the project.
  1. Click **+ Add tag** or **+**.
  2. Click **+ Create new tag** at the bottom of the **Event notification tags** window.
  3. In the **Create tag** window, enter the tag name, select a color, and click the [Create tag] button to create the tag.

4. Edit or delete created tags by clicking the icon in the tag list.
5. Select the desired tag from the **tag list** in the **Event notification tags** window to apply it.

> When configuring events, you can send notifications to project members and 3rd-party plugins with the selected event notification tags. You can assign tags to project members and 3rd-party plugins respectively in **Alert notifications > Event notification settings**.

4. When configuration is complete in the **Add event rule** window, click the [Save] button.

Edit/delete events

1. Go to the **Real-time log** tab in **Alert** > **Event Configuration**.
2. Click the icon for the event to edit or delete in the log event list.
3. Modify options in the **Event settings** screen and click the [Save] button.

   a. To delete the selected event, click the [Delete] button in the upper right of the **Event settings** screen.

Event sharing

You can save metric event settings as a JSON file to share configurations with other users or import settings created by others.

- JSON file name: event-rules-`YYYYMMDD`.json

Export

1. Click the [**JSON**] button at the top right of the screen.
2. When the JSON editor opens, click the [Export] button.
   - If you use the export function after searching for events, only the searched list will be downloaded

as a JSON file.

3. After the JSON file is downloaded, share it with other users.

Import

1. Click the [ ] button at the top right of the screen.
2. Select the JSON file downloaded using the **Export** function.
3. When the JSON editor opens, choose either [Add to List] or [Overwrite].

> It is recommended to use this feature between projects of the same product type. You can import event settings from projects of different products, but they may not function correctly.

Edit in JSON Format

1. Click the [**JSON**] button at the top right of the screen.
2. When the editor opens, modify the content according to the JSON format.
3. After completing the edits, click the [Save] button at the bottom of the screen.

> If the modified content does not conform to the JSON format, an error message will be displayed at the bottom of the screen and the content cannot be saved. The error message may vary depending on the type of formatting issue.
>
> JSON error

Appendix

Trigger Conditions and Target Selection Guide

Metric alert trigger conditions and event target selection use the same syntax. However, trigger conditions use the field key as the variable, while target selection uses the tag key as the variable.

Level Behavior Guide

Event levels are categorized into Critical, Warning, and Info.

1. **Priority-based Event Triggering**

   When multiple level conditions are met at the same time, only the event with the highest priority is triggered.

   Example

   ```
   Configuration:
   - Warning: CPU > 70%
   - Critical: CPU > 90%

   Current state: CPU 95%
   → Result: Only the Critical event is triggered (Warning is suppressed)
   ```

2. **Level Escalation**

   For events that maintain an in-progress state, when a higher-level condition is met while a lower-level event is already in progress, both levels enter the in-progress state.

   Example

   ```
   Configuration:
   - Warning: CPU > 70%
   - Critical: CPU > 90%

   Scenario:
   1) CPU 80% → Warning triggered (in progress)
   2) CPU 95% → Warning (in progress), Critical (in progress)
   ```

3. **Level De-escalation**

   For events with an in-progress state, when the level transitions from a higher level to a lower level, the higher-level event is resolved while the lower-level event remains active.

   Example

```
Configuration:
- Warning: CPU > 70%
- Critical: CPU > 90%

Scenario:
1) CPU 97% → Critical triggered
2) CPU 85% → Critical resolved, Warning remains
3) CPU 65% → Warning resolved
```

4. **Execution Flow**

```
- Warning: CPU > 70%
- Critical: CPU > 90%

Metric value: 60%  →  75%  →  92%  →  85%  →  60%
State:      Info      Warning     Critical     Warning     Info
                    (triggered)  (triggered)  (maintained)  (resolved)
                                  Warning
                                (maintained)
```

> Events that have an in-progress state remain active as long as their conditions are met and continue until those conditions are cleared.

Condition Configuration Guide

> If a metric name starts with a number or contains special characters, you must wrap it in the `${metric_name}` format.
>
> - Brackets `()`, `[]`
> - Operators `+`, `-`, `*`, `/`, `%`
> - Delimiters `:`, `@`, `#`, `,`, (space)
> - Other special characters `!`, `^`, `&`, `|`, `~`, `` ` ``, `=`
>
> **Examples using `${}`:**

```
${cpu(xos)} > 50
${mem[0]} >= 100
${cpu-usage} > 80
${namespace:cpu} > 70
${metric name} > 60
${4xx_error} > 10
```

Usable without `${}` : Metric names that contain only alphabetic characters, underscores (`_`), or dots (`.`)

```
cpu > 80
cpu_usage > 50
CPUUtilization.Average > 0.8
```

1. **Comparison Operators**

   Numeric Comparison

   ```
   Greater than:          cpu > 80
   Greater than or equal:   cpu >= 80
   Less than:             memory < 1000
   Less than or equal:      memory <= 1000
   Equal to:              status == 200
   Not equal to:          error != 0
   ```

   String Comparison

   ```
   status == 'OK'
   region == "us-east-1"
   ```

2. **Arithmetic Operators**

   - Basic arithmetic operations

     ```
     Addition:       cpu + 10 >= 90
     Subtraction:    memory - 100 >= 500
     Multiplication: cpu * 2 >= 100
     ```

```
Division:       disk / 1024 >= 100
Modulo:         value % 10 == 0
```

- Controlling precedence with parentheses

```
(cpu + memory) * 2 >= 200
(disk - used) / total >= 0.2
((cpu + memory) / 2) >= 50
```

- Negative values

```
cpu > -100
```

3. Logical Operators

- AND operator (&&)

```
cpu > 80 && memory > 1000
${cpu(xos)} > 50 && ${mem(xos)} > 60
```

- OR operator (||)

```
cpu > 90 || memory > 90
disk < 10 || network > 1000
```

- Combined logical operations

```
(cpu > 50 && memory > 50) || disk < 20
cpu > 80 && (memory > 1000 || disk > 500)
```

4. Pattern Matching Operators

- LIKE operator

```
oname like 'prod-*'
url like '*error*'
```

```
message like 'WARN%'
```

- NOT LIKE operator

```
oname not like 'test-*'
url not like '*debug*'
```

5. Built-in Functions

- Null check functions

```
isNull(value)           # Checks whether the value is null
isNotNull(value)        # Checks whether the value is not null
nvl(value, 0)           # Returns a default value if null
isEmpty(str)            # Checks whether the string is empty
isNotEmpty(str)         # Checks whether the string is not empty
```

- Aggregate functions

```
sum(cpu, memory, disk)            # Sum
avg(cpu, memory)                  # Average
max(cpu, memory, disk)            # Maximum
min(cpu, memory, disk)            # Minimum
count(value1, value2, value3)     # Count
```

- Math functions

```
round(cpu, 2)           # Rounds to 2 decimal places
```

- String functions

```
length(str)             # String length
startsWith(str, 'prefix')  # Checks prefix
endsWith(str, 'suffix')    # Checks suffix
indexOf(str, 'search')     # String index
substring(str, 0, 10)      # Substring
```

```
trim(str)                # Trims whitespace
replace(str, 'old', 'new') # String replacement
hasStr(str, 'search')     # Checks string containment
```

- Conditional functions
  - **if**: A conditional function that returns the `trueValue` if the condition evaluates to `true`, or the `falseValue` if it evaluates to `false`

    Syntax

    ```
    if(condition, trueValue, falseValue)
    ```

    Example

    ```
    if(value >= 90, 'High', 'Medium') == 'High'
    ```

  - **decode**: Compares a value against conditions and returns the result of the matching condition, or a default value if no match is found

    Syntax

    ```
    decode(value, condition1, result1, condition2, result2, ..., conditionN, resultN, defaultValue)
    ```

    Example

    ```
    decode(value, 1, 'Low', 2, 'Medium', 3, 'High', 'Unknown') == 'Unknown'
    ```

  - **in**: Returns `true` if the value matches any of the specified candidates; otherwise returns `false`

    Syntax

    ```
    in(value, candidate1, candidate2, ..., candidateN)
    ```

    Example

```
in(status, 200, 201, 204) == false
```

Writing Guidelines

- Recommended
  - Use simple variable names without `${}` (e.g. cpu, memory)
  - Always use `${}` when special characters are included
  - Use parentheses to clearly separate complex expressions
  - Enter function names exactly as defined (case-sensitive)
- Avoid
  - Incomplete expressions (e.g. `cpu >` , `memory &&`)
  - Mismatched parentheses (e.g. `(cpu > 80 && cpu > 70)`)
  - Consecutive or invalid operators (e.g. `cpu >> 80` , `cpu >< 80`)

Composite Log Event Settings

Home > Select Project > **Alert** > **Event Configuration** > **Composite Log** tab

You can set alert notifications by filtering collected log data against conditions. Log monitoring must be activated to use log events. For more information on **activating log monitoring**, refer to the log documentation for each product.

- **Composite log event**: An event is triggered when the number of logs that meet the log search conditions within the configured **data query range** reaches or exceeds the threshold count.

> **Permission**
>
> **Alert configuration** permission is required to configure composite metrics events. Alert configuration permission can be set by clicking the icon for the member to grant permission in **Home Screen > Select Project > Management > Integrated Member Management**.

Default options

- **Search**: Search for events by selecting an event name or event notification tag
- **JSON download/upload**: Event rules can be directly edited through the JSON editor, or downloaded as a file, modified, and re-uploaded
- **Notification message settings**: Select detailed items for the message to receive and edit the content for each item
- **+ Add event**: Add a new log event

Table | Composite log event list configuration

Item	Description
Event name	Event name used as the notification title
Template	Composite log template

Item	Description
Category	Log classification name (log folder name)
Rule	Conditions that trigger the event
Additional notification when event is resolved	Whether to receive notifications in normal (RECOVERED) status when Critical and Warning level events are resolved - Enabled: Sends a RECOVERED notification when transitioning from in-progress → normal status, and the in-progress/resolved status can be distinguished in event history - Disabled: A one-time event occurs each time the threshold is exceeded
Pause event notifications	Pauses the same event notification for a specified time after an event notification is triggered - Available times: Disabled, 5 min, 10 min, 15 min, 20 min, 30 min, 1 hour, 2 hours, 3 hours, 6 hours, 12 hours, 1 day - If resolved notifications are enabled, no notification is sent for the same event during the selected time after receiving a normal (RECOVERED) status notification
Interval	Time range for querying log data
Event notification tags	User or group (notification tag) to receive the event notification
	Edit and delete events
	Whether the event is active - Enabled: Event occurs when conditions are met - Disabled: Event does not operate and conditions are not checked

Event settings

The composite log event feature triggers alert notifications when the number of logs matching the log search conditions within the configured **data query range** reaches or exceeds the threshold count.

Add composite log event

1. Click the **Composite Log** tab in **Alert** > **Event Configuration**.
2. Click the [+ Add event] button on the right side of the screen.
3. Set the log event conditions in the **Add event rule** window. For details, refer to Basic settings, Detail settings, and Additional settings.

   Add composite log event
4. When configuration is complete in the **Add event rule** window, click the [Save] button.

Basic settings

- **Event name**: Enter a name to use as the event title.
- **Event activation**: When the toggle button is activated, an event occurs when conditions are met.
- **Level**: Select the event level (Critical, Warning, Info).

Detail settings

Configure Template (preset), Message, Category, **Filter**, and Event trigger conditions.

Template (preset)

Selecting a predefined preset automatically configures the **filter, query type, and event trigger conditions**. You can modify the settings after selecting a preset.

Composite log template types

Preset	Query type	Default rule
None	none	`rows > 10`
2xx status code count	count	`count > 10`

Preset	Query type	Default rule
3xx status code count	count	`count > 10`
4xx status code count	count	`count > 10`
5xx status code count	count	`count > 10`
Normal status code (2xx, 3xx) count	count_v2	`include_minus_exclude_count > 10`
Error status code (4xx, 5xx) count	count_v2	`include_minus_exclude_count > 10`
Error received count	count	`count > 10`
Exception received count	count	`count > 10`
Log count per second	rps	`rps > 100`
Field aggregation (Aggregation)	aggr	`avg > 1000`
Rate	rate	`rate > 10`
Unique value count (Cardinality)	cardinality	`cardinality > 100`
Percentile	percentile	`p99 > 3000`

Message

Enter the notification message to receive when an event occurs. You can use variables in the message to compose dynamic notifications.

- **Example**: Message `${host}` error `${count}` occurred → Notification message web-01 error 15 occurred

Table | Message variables

Variable	Description	Usage condition
`${count}`	Number of logs matching the condition	Always available
`${groupField}`	Grouping field name	Grouped notifications only
`${groupValue}`	Grouping field value	Grouped notifications only
`${fieldName}`	Grouping field value (referenced directly by field name)	Grouped notifications only

> The `${count}` variable can be used in all notifications regardless of grouping. The `${groupField}`, `${groupValue}`, and `${fieldName}` variables are only substituted when a grouping field is configured.

Category

Select the log classification name (log folder name) from the list or enter it directly. You can enter the search key and search value only after selecting a category.

Filter

Enter log search conditions using Lucene query syntax. Two input modes are supported.

- **Filter mode**: Build a query by selecting fields and values.
- **Code mode**: Enter Lucene queries directly.

Table | Lucene query examples

Query	Description
`level:ERROR`	Logs where the level field is ERROR
`status_nxx:status_4xx OR status_nxx:status_5xx`	4xx or 5xx status codes
`appender:accessLog`	Logs from the accessLog appender
`response_time.n:*`	Logs where the response_time.n field exists

Query	Description
`host:web-01 AND level:ERROR`	Error logs from host web-01

Event trigger conditions

Configure by query type. Additional settings are displayed depending on the selected template (preset) or query type.

Table | Event trigger condition key meanings

Key	Meaning
`rows` / `count`	Number of logs matching the filter condition
`rps`	Log count per second (count / interval in seconds)
`avg`	Average value of the target field
`sum`	Sum of the target field values
`min`	Minimum value of the target field
`max`	Maximum value of the target field
`rate`	Numerator count / denominator count × 100 (%)
`cardinality`	Number of unique values in the target field
`p99` / `p95` / `p90` / `p75` / `p50`	Percentile values of the target field
`include_minus_exclude_count`	Include filter count − exclude filter count

- **Count (count / none)**: Counts the number of logs matching the configured filter conditions.

  Table | Count settings

Setting	Description
Filter	Log search condition (Lucene query)
Event trigger condition	`count` or `matching logs` `{operator}` `{threshold}` (e.g., `count > 10`)

- **Log count per second (rps)**: Calculates the number of logs per second (Requests Per Second) matching the configured filter conditions. The total count during the interval is divided by the interval in seconds.

Table | Log count per second settings

Setting	Description
Filter	Log search condition (Lucene query)
Event trigger condition	`RPS` `{operator}` `{threshold}` (e.g., `rps > 0.5`)

- **Field aggregation (aggr)**: Calculates the aggregated value (average, sum, min, max) of the specified numeric field. The aggregation method can be selected from a dropdown in the event trigger condition.

Table | Field aggregation settings

Setting	Description
Filter	Log search condition (Lucene query)
Target field	Numeric field to aggregate (e.g., `response_time.n`)
Event trigger condition	`avg` / `sum` / `min` / `max` `{operator}` `{threshold}` (e.g., `avg > 1000`)

- **Rate**: Calculates the percentage (%) ratio of the numerator filter count to the denominator filter count. Useful for monitoring the ratio of specific conditions against all logs.

Table | Rate settings

Setting	Description
Filter (numerator)	Log condition for the numerator (e.g., `level:ERROR`)
Denominator filter	Log condition for the denominator (e.g., `level:*`)
Event trigger condition	`Rate (%)` `{operator}` `{threshold}` (e.g., `rate > 10`)

  - **Example**: Filter `level:ERROR`, denominator filter `level:*`, rule `rate > 10` → Alert triggered when the error rate among all logs exceeds 10%

- **Unique value count (cardinality)**: Counts the number of unique values in the specified field. Useful for detecting an abnormally large variety of values.

Table | Unique value count settings

Setting	Description
Filter	Log search condition (Lucene query)
Target field	Field to calculate unique values for (e.g., `host`)
Event trigger condition	`Unique value count` `{operator}` `{threshold}` (e.g., `cardinality > 100`)

- **Percentile**: Calculates the percentile value of the specified numeric field. Useful for monitoring tail latency in response times. Changing the percentile selection automatically updates the key in the event trigger condition.

Table | Percentile settings

Setting	Description
Filter	Log search condition (Lucene query)
Target field	Numeric field to calculate percentile for (e.g., `response_time.n`)
Percentile	Select from P99, P95, P90, P75, P50

Setting	Description
Event trigger condition	`P99` / `P95` /... `{operator}` `{threshold}` (e.g., `p99 > 3000`)

Table | Percentile meanings

Percentile	Meaning
P50	Median. 50% of all data is at or below this value
P75	75% of all data is at or below this value
P90	90% of all data is at or below this value
P95	95% of all data is at or below this value
P99	99% of all data is at or below this value

- **Include/exclude count (count_v2)**: Calculates the net count by subtracting the exclude filter count from the include filter count.

Table | Include/exclude count settings

Setting	Description
Filter (include)	Log conditions to include (e.g., `status_nxx:status_4xx OR status_nxx:status_5xx`)
Filter exclude	Log conditions to exclude (e.g., `status_nxx:status_4xx`)
Denominator filter	Denominator for ratio calculation (optional)
Event trigger condition	`Include-exclude count` `{operator}` `{threshold}` (e.g., `include_minus_exclude_count > 10`)

  - **Example**: Filter `status_nxx:status_4xx OR status_nxx:status_5xx`, filter exclude `status_nxx:status_4xx` → Alert triggered when the net 5xx count (4xx+5xx minus 4xx) exceeds 10

Group by field (Group By)

When a group by field is configured, alert conditions are evaluated independently for each value of that field. Select the field to group by from the dropdown. Clicking the dropdown shows a list of available fields for the current category with autocomplete.

- **Example**: Filter `level:ERROR`, group by field `host`, rule `rows > 3`

Table | Grouping comparison

Item	Without grouping	With grouping
Evaluation unit	All logs aggregated as one	Independent aggregation per group value
Alert occurrence	1 alert	One per group meeting the condition
Alert title	`Event title`	`[Group value] Event title`
Stateful	State transition based on all logs	Independent state transition per group

  - `[web-01] Error alert` — 8 errors from web-01 → alert triggered
  - `[web-02] Error alert` — 5 errors from web-02 → alert triggered
  - `[api-01] Error alert` — 2 errors from api-01 → condition not met, no alert

> ⓘ When **Additional notification when event is resolved** is enabled, the ABNORMAL → RECOVERED state transition occurs independently per group. For example, when errors from `web-01` are resolved, a `[web-01] RECOVERED` notification is triggered, with no effect on other hosts.

Additional settings

- **Interval**: Check alert conditions at the selected time interval (`unset`, `1 min`, `5 min`, `10 min`, `30 min`, `1 hour`, `12 hours`).
- **Mute**: After an event occurs, the same event is not triggered during the selected time (`unset`, `1 min`, `5 min`, `10 min`, `30 min`, `1 hour`, `12 hours`).

- **Event operation time**: Set tags to activate the event during specific time periods (work/non-work/maintenance hours, etc.). If no tag is set, the event operates 24 hours a day when activated.

    1. Click **+ Add**.
    2. Click **+ Create new tag** in **Event operation time**.
    3. In **Create event operation time tag**, select the **tag name, day of week, time, and color**, then click the [Apply] button.
    4. Created tags can be viewed in the tag list and are applied by selecting the checkbox.
    5. Click the icon for the tag to modify or delete, and edit or delete the tag in **Edit event operation time tag**.

    > Deleting a tag removes the tag for all users to whom it has been applied. However, tags in use by event rules cannot be deleted.

- **Event notification tags**: If notification tags are set for an event, notifications are sent to project members with those tags. If no recipient tags are set, notifications are sent to all users in the project.

    1. Click **+ Add tag** or **+**.
    2. Click **+ Create new tag** at the bottom of the **Event notification tags** window.
    3. In the **Create tag** window, enter the tag name, select a color, and click the [Create tag] button to create the tag.
    4. Edit or delete created tags by clicking the icon in the tag list.
    5. Select the desired tag from the **tag list** in the **Event notification tags** window to apply it.

    > When configuring events, you can send notifications to project members and 3rd-party plugins with the selected event notification tags. You can assign tags to project members and 3rd-party plugins respectively in **Alert notifications** > **Event notification settings**.

Simulation

You can simulate using historical data before saving the alert to check the expected number of alert occurrences.

1. After completing the alert configuration, select a time range at the bottom.
2. Click the [Simulation] button.
3. The chart displays data by time period and the number of alert occurrences.

   > You can adjust the simulation time range to check alert occurrence patterns across different time periods. Data from the last 1–2 minutes may not be available due to index build delays.

4. When configuration is complete in the **Add event rule** window, click the [Save] button.

Create alert from Log Explorer

You can create alerts directly from Log Explorer.

1. Navigate to **Log** > **Log Search** menu.
2. Configure the category and search query.
3. Click the [Add event] button in the upper right of the screen.
4. The alert configuration form opens in a drawer, with the current category and search query automatically filled in.
5. Complete additional settings and click the [Save] button.
6. After saving, click the [Go] button to navigate to the alert list.

Edit/delete events

1. Go to the **Composite log** tab in **Alert** > **Event Configuration**.
2. Click the icon for the event to edit or delete in the log event list.
3. Modify options in the **Event settings** screen and click the [Save] button.

   a. To delete the selected event, click the [Delete] button in the upper right of the **Event settings** screen.

Event sharing

You can save metric event settings as a JSON file to share configurations with other users or import settings created by others.

- JSON file name: event-rules-`YYYYMMDD`.json

Export

1. Click the [**JSON**] button at the top right of the screen.
2. When the JSON editor opens, click the [Export] button.
    - If you use the export function after searching for events, only the searched list will be downloaded as a JSON file.
3. After the JSON file is downloaded, share it with other users.

Import

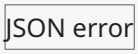
1. Click the [ ] button at the top right of the screen.
2. Select the JSON file downloaded using the **Export** function.
3. When the JSON editor opens, choose either [Add to List] or [Overwrite].

> It is recommended to use this feature between projects of the same product type. You can import event settings from projects of different products, but they may not function correctly.

Edit in JSON Format

1. Click the [**JSON**] button at the top right of the screen.
2. When the editor opens, modify the content according to the JSON format.
3. After completing the edits, click the [Save] button at the bottom of the screen.

If the modified content does not conform to the JSON format, an error message will be displayed at the bottom of the screen and the content cannot be saved. The error message may vary depending on the type of formatting issue.



Event Notification Settings

Home > Select Project > **Alert** > **Notifications**

Event Notification Settings per User

> Information about other members in **Event notification settings per-user** can only be viewed by members with the **Member Management** permission. For all other members, required information is masked.
>
> - Mobile device management for bulk event notification receipt settings and access control can be managed in **Account Information**.
> - For detailed information about member permissions, refer to the following document.

Notification Methods

Alerts are available via **Email**, **SMS**, **WhatsApp**, and **Mobile**.

In the **Event notification settings per-user** section, select the checkbox for the desired notification method to receive alerts. If you clear the checkbox for a notification method, alerts will not be sent through that channel.

> All users except the project super administrator can change only their own notification settings.

Email Alerts

Email alerts are sent to the email address provided during sign-up.

SMS Alerts

You can receive alerts via SMS.

> The SMS alert feature is available only for paid projects.

1. Click your **profile icon** at the top right of the screen, then go to **Manage Accounts**.
2. In **Account Information**, click the [General Mobile Phone] button in the **User Phone Number** section.
3. Enter the **phone number** that will receive the verification code.
    - Only **Korean mobile phone numbers** can be registered to receive SMS alerts.
4. Click the [Save] button.

> **Change or delete the registered phone number**
>
> - To change the phone number, click the [Change] button. SMS verification is required.
> - To delete the phone number, click the [Delete] button. After deletion, you will no longer receive alerts via SMS.

WhatsApp Alerts

You can receive alerts via WhatsApp.

> The WhatsApp alert feature is available only for paid projects.

1. Click your **profile icon** at the top right of the screen and go to **Manage Accounts**.
2. In **Account Information**, click the [WhatsApp] button in the **User Phone Number** section.
3. Enter the **phone number** that will receive the verification code.
4. Click the [Send Verification Code] button.
5. Enter the 6-digit verification code sent to your WhatsApp application.
    - If you do not receive the verification code within 10 minutes, click the [Resend Verification Code] button.
6. Click the [Verify] button.

Mobile Alerts

Users who need to receive mobile alerts must install the WhaTap application on their mobile device and log in.

- Download the WhaTap application – **Android**
- Download the WhaTap application – **iOS**

Repeated Alerts (Escalation) Settings

If the alert condition is not resolved after an alert is triggered, you can configure repeated alert intervals based on the time of the initial alert. This setting applies only to **Critical** level alerts.

For example, to repeat alerts immediately (0 minutes), 1 hour later, and 1 day later after the initial alert, enter `0,1H,1D` in the **Repeated Alerts (Escalation)** column.

1. Go to **Alert** > **Notifications** and navigate to the **Event notification settings per-user** section.
2. Configure the settings in the **Repeated Alerts (Escalation)** field.
   - **M**: minutes, **H**: hours, **D**: days. If the unit is omitted, the value is interpreted as minutes.
   - Enter a number or a number with a unit (**M**, **H**, **D**). If the input format is invalid, an error message is displayed.
3. Click the [Save] button. The settings will not be applied unless they are saved.

Event Reception Tags

When configuring events, you can select **event reception tags** to send alerts to project members and 3rd-party plugins associated with those tags. If no event reception tag is set, alert notifications are sent to **all project members**.

Add / Edit / Delete Event Reception Tags

> Event reception tags that are **currently applied to an event** cannot be deleted.

1. Go to **Alert** > **Notifications**.

2. In the user list under the **Event notification settings per-user** section, click **+ Add Tag** or **+**.
3. In the **Event reception tag** window, click **+ Create New Tag**.
4. In the **Create Tag** dialog, enter a tag name, select a color, and click the [Create Tag] button.
5. To edit or delete a created tag, click the icon in the tag list.
6. Select the desired tag from the **Tag List** in the **Event reception tag** window to apply it.

Remove Event Reception Tags

1. Go to **Alert** > **Notifications** and open the **Event notification settings per-user** section.
2. Click **+** in the **Event reception tag** field for the user whose event reception tag you want to remove.
3. In the **Event reception tag** window, click the **X** on the tag you want to remove. The tag will be removed.

Receive Alerts Without Reception Tags

When configuring events, you can select event reception tags to send alerts to project members and 3rd-party plugins associated with those tags.

- To receive alerts **without event reception tags**, enable the **Receive alerts without reception tags** option.
- If you want to receive **only alerts that have event reception tags**, disable this option.

> To stop receiving all alerts, disable this option and make sure that no **Event reception tag** is selected.

Selecting the event reception mode

In the **Event notification settings per-user** section, you can select the event alert reception scope per member or third-party channel from the following three modes. The reception mode is selected in the **Event reception tags** column of each member or channel row.

Mode	Description
Receive all alerts	Receive all alerts regardless of the event reception tags
Receive alerts for selected tags	Receive only alerts that have the selected event reception tags. Select at least

Mode	Description
only	one tag
Receive no alerts	Receive no event alerts. The selected tags are preserved without being deleted

- In the **Receive alerts for selected tags only** mode, select at least one tag to receive alerts in the tag selection area.
- To also receive alerts that have no event reception tag specified, select the **Unclassified** virtual tag in the tag selection area. The **Unclassified** tag can be selected or deselected like a normal tag, but the tag itself cannot be edited or deleted.

> ⓘ In the **Receive no alerts** mode, the corresponding member or channel receives no event alerts at all. The selected tags are preserved without being deleted, so if you switch back to the **Receive alerts for selected tags only** mode, you can use the previously selected tag settings as is.

Select Notification Language

You can choose the language in which you receive alert notifications.

1. Go to **Alert** > **Notifications**.
2. In the user list under the **Event notification settings per-user** section, click the ▶ button at the far left of the user whose language you want to configure.
3. Select the **Notification Language**.

Day and Time-based Notification Settings

You can configure whether a user receives notifications by day of the week and time of day.

1. Go to **Alert** > **Notifications**.
2. In the user list under the **Event notification settings per-user** section, click the ▶ button at the far left of the user whose day and time settings you want to configure.

3. Select the days and/or enter the time ranges during which you want to receive alerts.
    - Settings can be configured separately for each notification method.

3rd-party Plugins

You can receive alert notifications through external applications.

- **Supported applications**: Slack, Telegram, Teams, Teams Workflows, Jandi, AlertNow, Webhook, Webhook JSON, Opsgenie, PagerDuty, LINE, ilert

> ⓘ Internal messengers that are not included in WhaTap Labs' supported scope can be integrated via standard **Webhook** or **Webhook JSON**.

Table | 3rd-party Plugin List

Item	Description
Plugin Name	Name of the plugin
Auth Key	Authentication key
Auth Value	Authentication value
Receive Level	Notification level: Warning, All, or Critical
Repeated Alerts (Escalation)	Re-notification interval from the initial alert time if the issue is not resolved - Applies only to Critical alerts
Smart Message (AI)	AI-powered notifications available when adding the Telegram plugin - Supported languages: Korean, English, Japanese
Event Reception Tag	Whether to receive alerts for specific members or teams in the project
Delete	Delete the 3rd-party plugin

Add a 3rd-party Plugin

1. Go to **Alert** > **Notifications**.
2. In the **3rd-party Plugins** section, click **Add**.
3. In the **Add 3rd-party Plugin** window, select the desired service tab.
4. Follow the instructions for the selected service (see How to add 3rd-party plugin notifications).
5. Click **Test Receive** to verify the configuration.
6. When finished, click **Add** or **Register**.

How to Add 3rd-party Plugin Notifications

Slack

In the **Add 3rd-party Plugin** window, click the [**Add To Slack**] button on the **Slack** tab to register **WhaTapNotiBot** to your Slack team and channel.

Telegram

1. In **Telegram**, search for **@WhaTapApmNotiBot**.

   Telegram

2. Select the bot and click the [Start] button.
3. In the chat window, enter the `/license` command followed by the access key of the project for which you want to receive alerts.

   Telegram

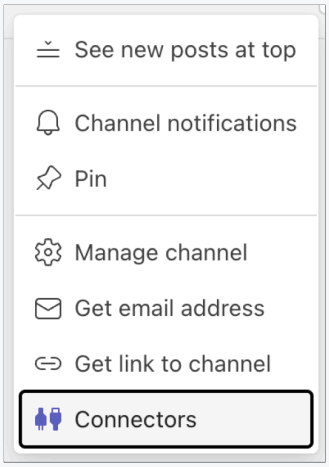
Available commands

- `/help` : View the list of commands
- `/license` : Enter the access key of the project to receive alerts

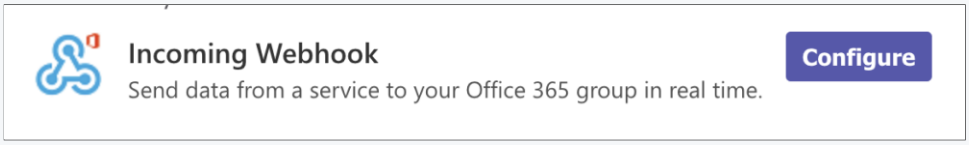
- `/remove` : Enter the project code for which you want to stop receiving alerts
- `/list` : View the list of project codes currently receiving alerts

Teams

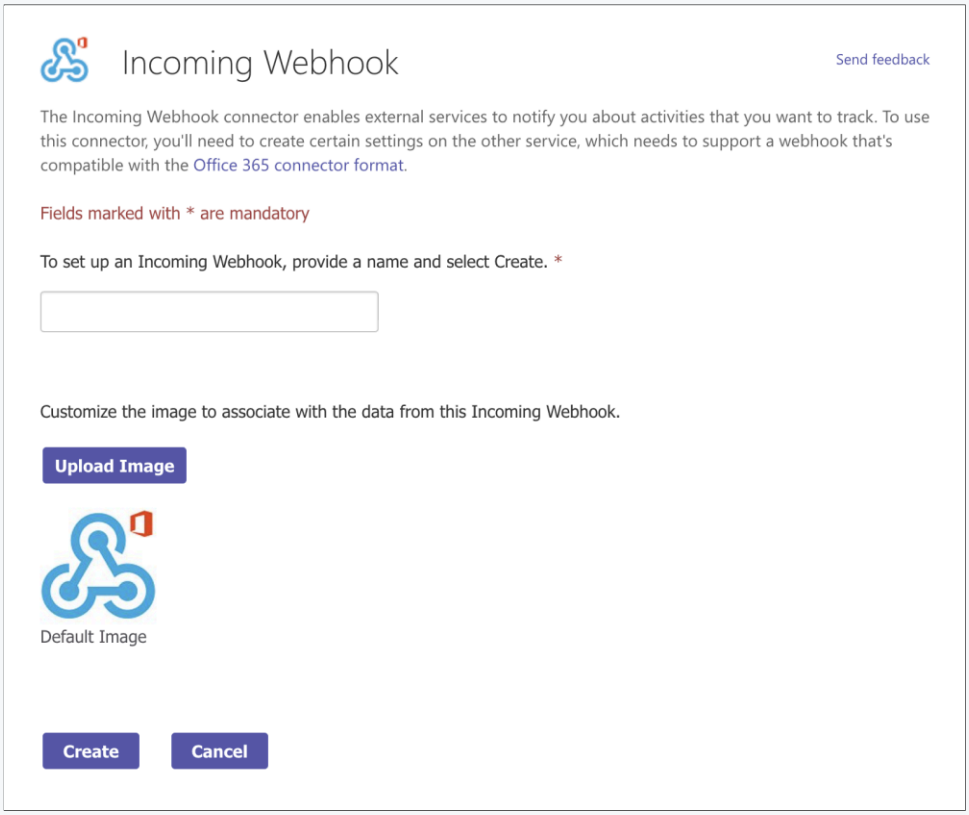
1. Go to the **connector** of the channel where you want to receive alerts.



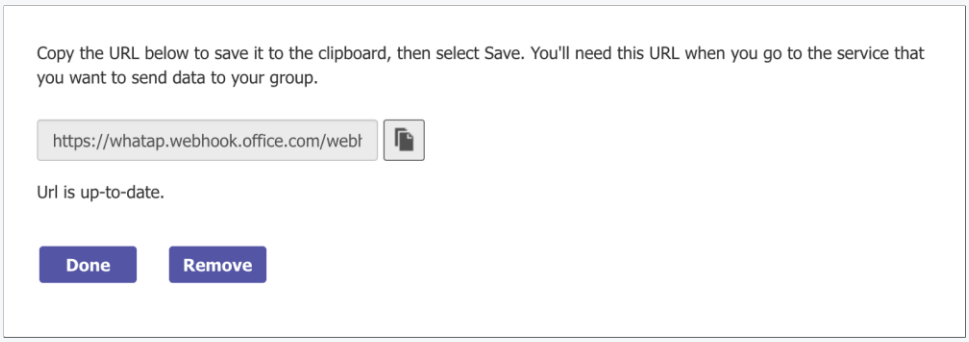
2. Add a **Webhook** and click the [Configure] button.



3. Enter a webhook name and click the [Create] button.



4. Copy the generated **Webhook URL**.

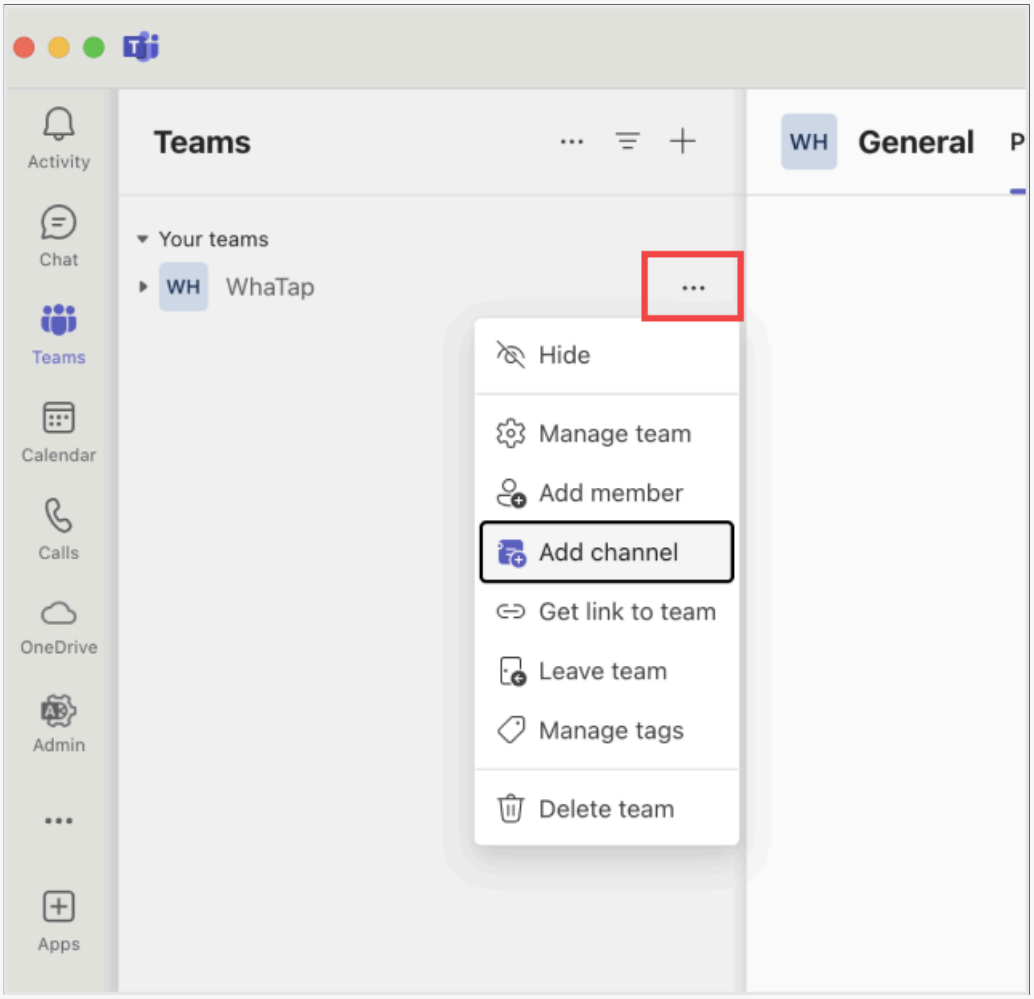


5. Return to the **Add a third-party plug-in** window in the WhaTap Monitoring service, and enter values for **Channel Name** and **Webhook URL** at the bottom of the **Teams** tab.
6. Click the [**Registration**] button.

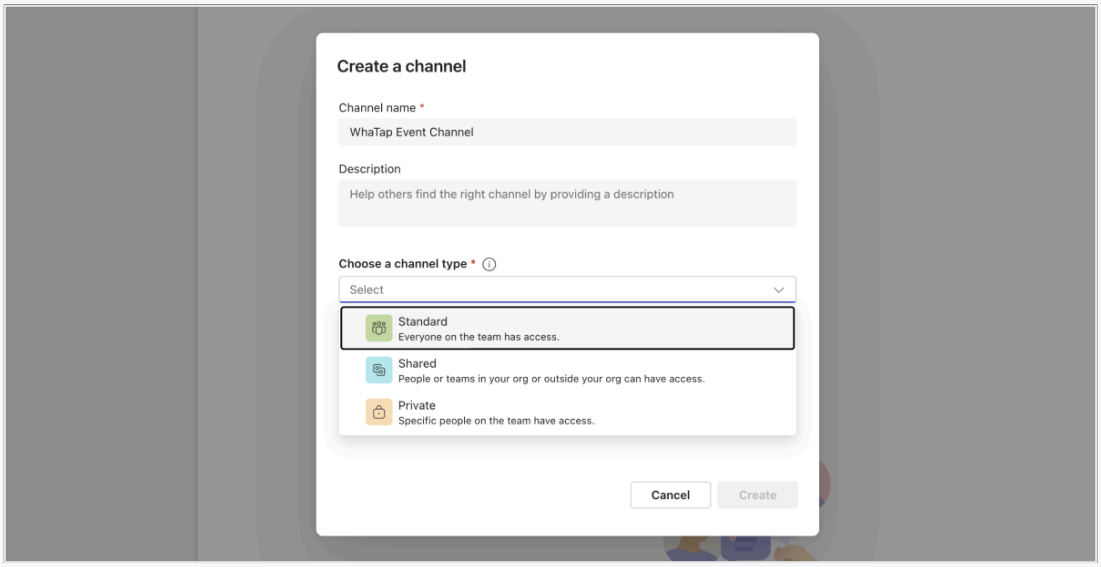
Teams Workflows

To use workflows, you need an account that can access **Microsoft Teams** and a **Business plan subscription**.

1. In the **Teams** menu, click **More actions (...)** > **Add channel**.

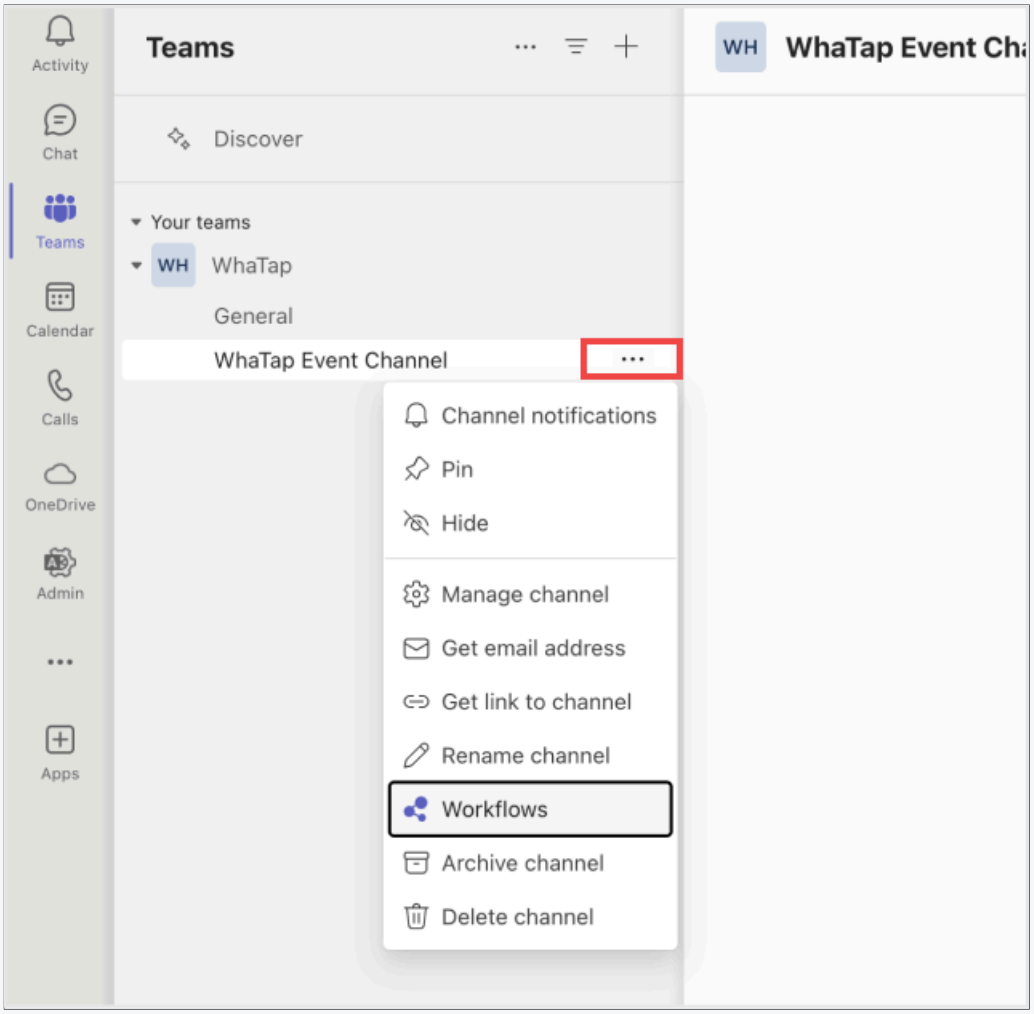


2. In the **Create a channel** window, enter the **Channel name** and **Description**. For **Channel type**, select **Standard**, then click the [Create] button.

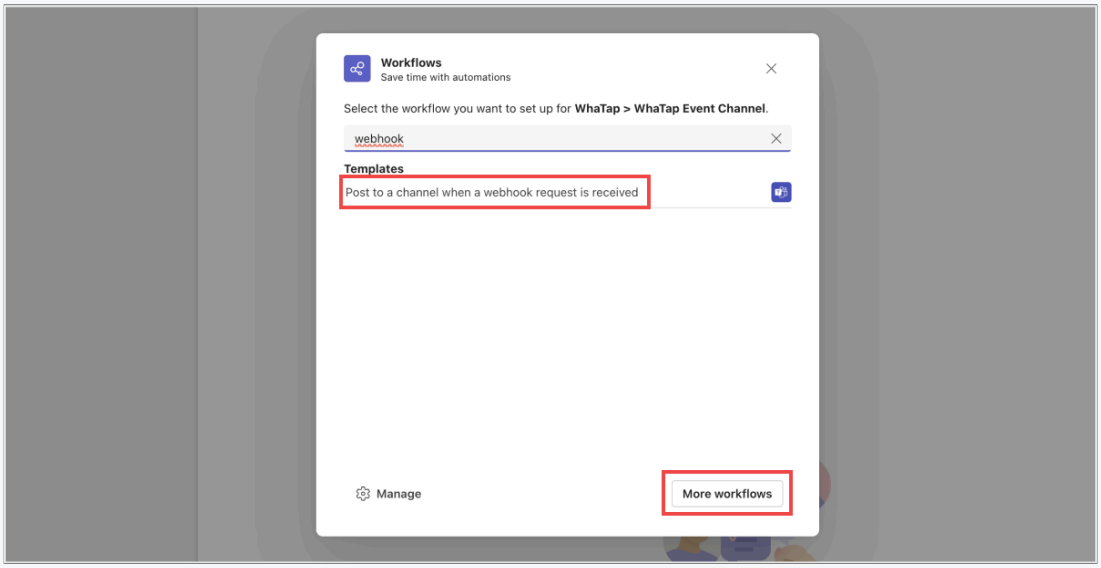


If you select **Shared** as the channel type, workflows cannot be used. If you select **Private**, workflows can be added but notifications cannot be received.

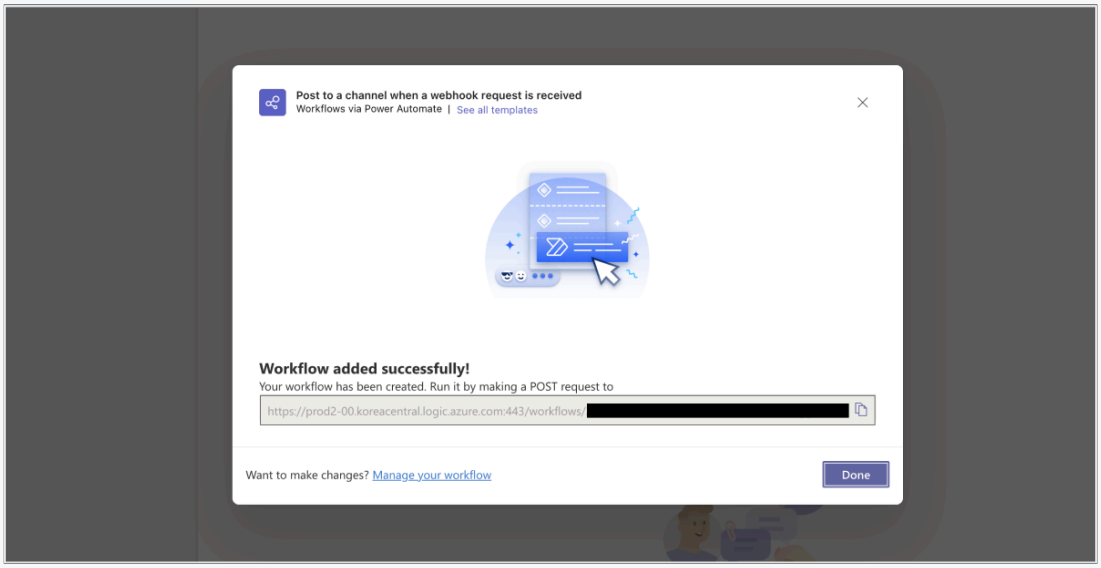
3. Click **More actions (...)** of the created channel, then select **Workflows**.



4. In the **Workflows** dialog, select **Post to a channel when a webhook request is received** from the workflow list, then click the [Add workflow] button.



5. Check the **Name** and **Connection** status, then click the [Next] button.
6. Review the **Details**, then click the [Add workflow] button.
7. The workflow has been added. Copy the generated URL.

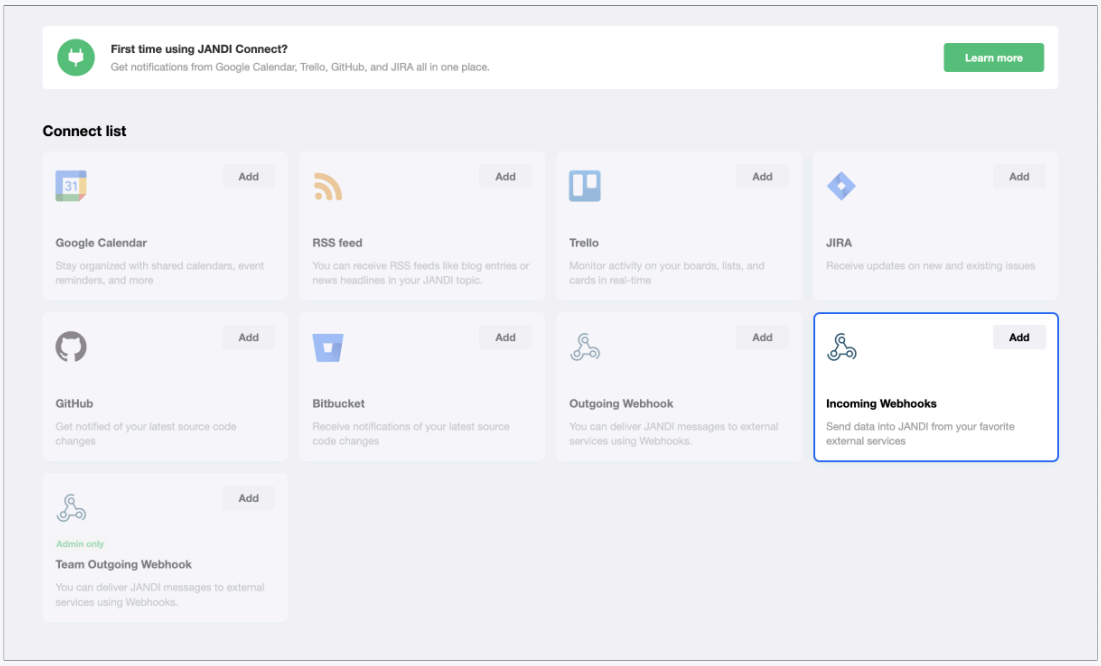


8. Return to the **Add a third-party plug-in** window in the WhaTap Monitoring service, and enter values for **Channel Name** and **Webhook URL** at the bottom of the **Teams Workflows** tab.

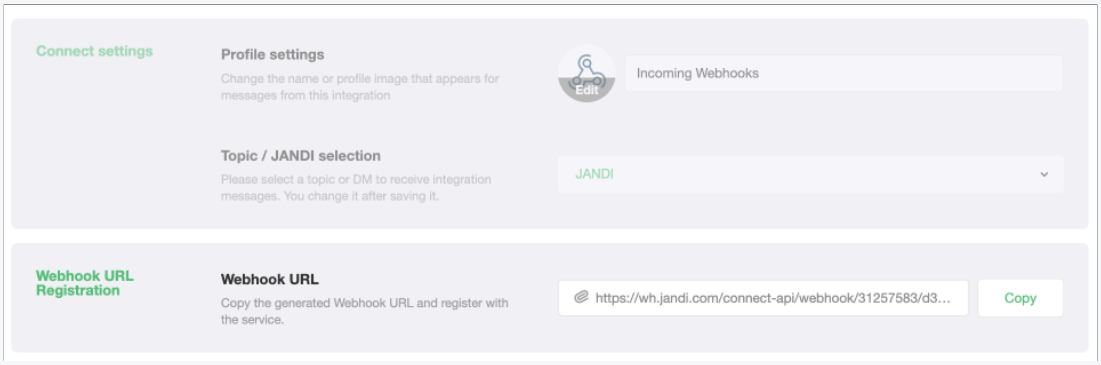
9. Click the [**Registration**] button.

Jandi

1. Go to **Jandi Connect settings** in the topic or chat where you want to receive notifications.
2. In the **Connect list**, find **Webhook Receiver (Incoming Webhook)** and click the [Connect] button.



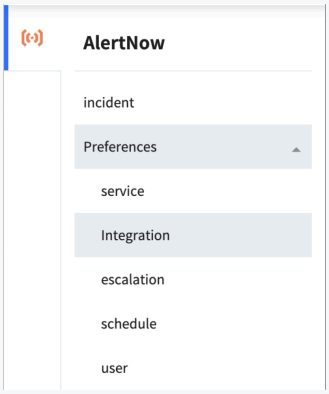
3. Click the [Add Webhook Receiver] button.
4. Copy the Webhook URL from the **Register Webhook URL** section.



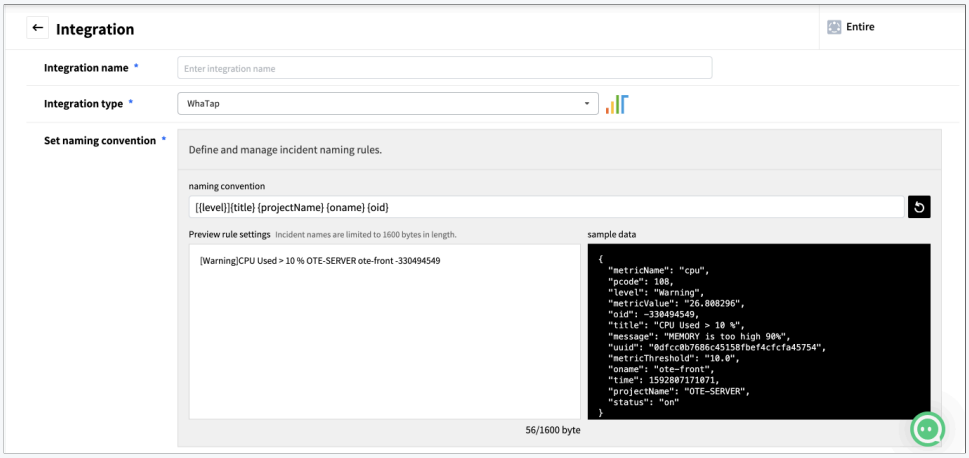
5. Return to the **Add a third-party plug-in** window in the WhaTap Monitoring service, and enter values for **Channel Name** and **Webhook URL** at the bottom of the **Jandi** tab.
6. Click the [**Registration**] button.

AlertNow

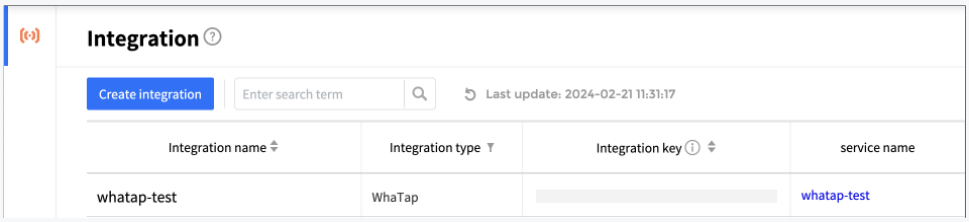
1. Access the **AlertNow** service, then go to **Settings** > **Integration**.



2. On the **Integration** screen, click the [Create Integration] button.
3. From the list of integration types, select **WhaTap**.
4. On the **Create Integration** screen, configure the **Name** and **Naming Rules**.



5. Click the [Confirm] button to create the integration.
6. Check the list of created integrations and select the newly created integration.



7. On the details screen, click the copy icon on the right side of the URL to copy it.



8. Return to the **Add a third-party plug-in** window in the WhaTap Monitoring service, and enter values for **Channel Name** and **Webhook URL** at the bottom of the **AlertNow** tab.
9. Click the [**Registration**] button.

Webhook

1. Register the Webhook that will receive notifications. The alert `title` and `text` are sent to the registered URL via the **POST** method.

```
{
  "title":"Alert Title",
  "text": "Sample warning message"
}
```

2. Enter values for **Webhook Name** and **Webhook URL**.
3. Click the [**Registration**] button.

Webhook JSON

1. Register the Webhook that will receive notifications. Alert data is sent to the registered URL in **JSON format** via the **POST** method. For substitution variables that can be included in the JSON payload, see **Available Variables**.
   - `${something}` represents a substitution value provided by WhaTap. Refer to **Available Variables** for the full list and descriptions.
   - The event status (`status`) is categorized as **event triggered** (`on`) or **event recovered** (`off`).
   - The `level` is categorized as `None` , `Info` , `Warning` , or `Critical` .

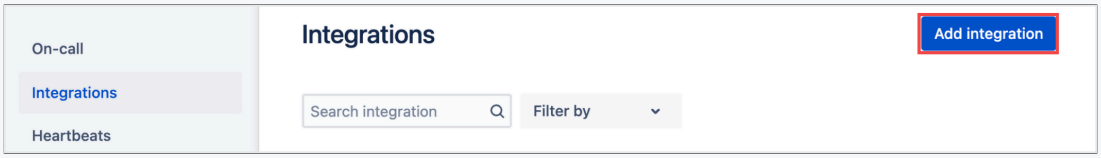
```
{
  "pcode": "${pcode}",
  "projectName": "${projectName}",
  "time": "${time}",
  "oid": "${oid}",
  "oname": "${oname}",
  "title": "${title}",
  "message": "${message}",
  "level": "${level}",
  "status": "${status}",
```

```
  "metricName": "${metricName}",
  "metricValue": "${metricValue}",
  "metricThreshold": "${metricThreshold}",
  "uuid": "${uuid}",
  "okind": "${okind}",
  "onode": "${onode}"
}
```

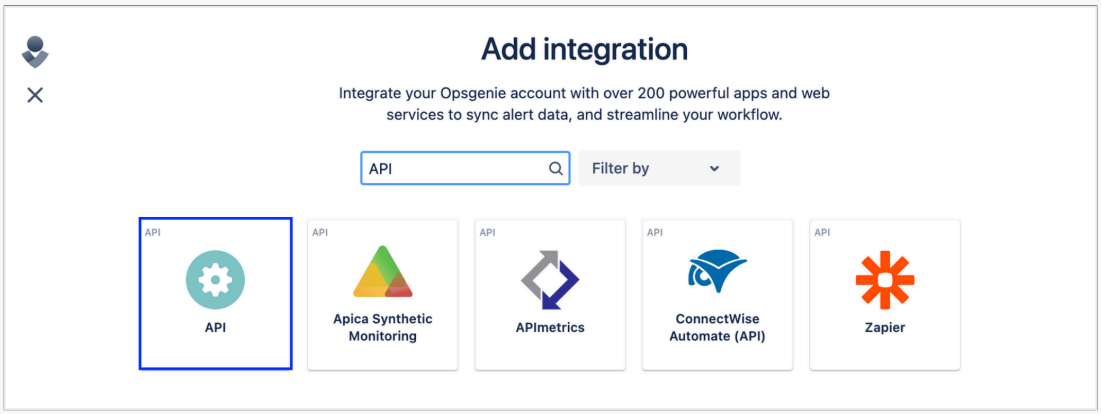
2. Enter values for the **Webhook Name** and **Webhook URL** fields.
3. To add a header value, click **+ Add** in **Add user header value**, then enter the key and value.
4. Click the [**Registration**] button.

Opsgenie

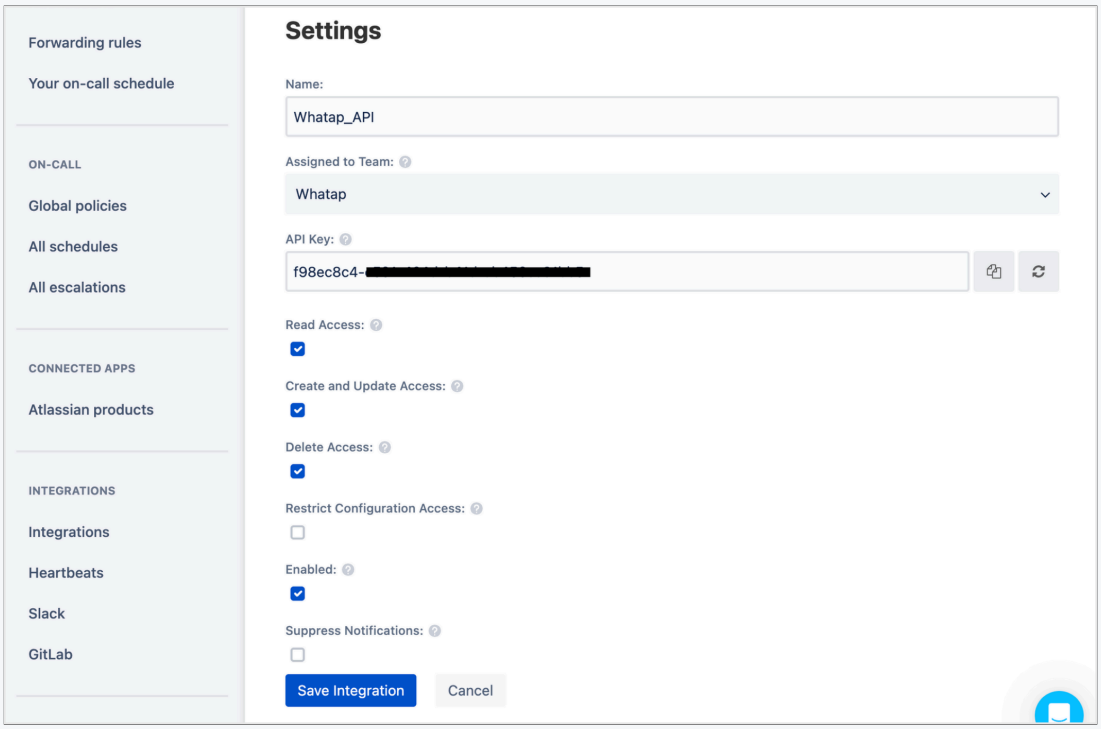
1. In the **Teams** menu, select the team that will receive notifications.
2. Go to the **Integration** menu and click the [Add Integration] button.



3. In the integration list, search for and select **API**.



4. Set the API name, assign a team, configure the required permissions, and then click the [Save Integration] button.



5. Return to the **Add a third-party plug-in** window in the WhaTap monitoring service, and enter values for **Name** and **API Key** at the bottom of the **Opsgenie** tab.
6. Click the [**Registration**] button.

PagerDuty

1. Sign up for a PagerDuty account and log in.
2. In PagerDuty, go to **Services** > **Service Directory**, then click the [+ New Service] button.

   PagerDuty
3. Enter the service information (**Name**) and click the [Next] button.

   PagerDuty
4. Configure the **Escalation Policy**, then click the [Next] button.

PagerDuty

5. Configure **Alert Grouping** and **Transient Alerts**, then click the [Next] button.

   PagerDuty

6. In **Integrations**, select **Event API V2**, then click the [Create Service] button.

   PagerDuty

7. On the created service page, copy the **Integration Name** and **Integration Key** values.

   PagerDuty

8. Return to the **Add a third-party plug-in** screen in the WhaTap monitoring service and enter the **Integration Name** and **Integration Key** values.
9. Click the [Register] button.

> If you do not receive messages or the plugin is not registered, check whether the **Integration Name** or **Integration Key** values are correct.

LINE

1. Install the LINE app.
2. Scan the QR code below to add the official **WhaTapLabs** channel as a friend.

   Add LINE 3rd-party plugin

3. Use the following commands to link your project with LINE and configure notifications.

   Commands

```
/help # Show available commands
/list # View the list of linked projects
/license [ACCESS_KEY] # Add a new project linkage
/remove [PROJECT_CODE] # Remove a linked project
/language [ko|ja|en] # Set notification language (Korean/Japanese/English)
```

- You can find the project access key under **Management > Project Management**.
- Using the `/language` command in LINE applies **only to the current LINE chat room**. It does not change the notification language for other 3rd-party plugins or projects.

Examples

```
/license XXXXXXXXXX # Link a project
/language ko # Set language to Korean
/list # Check linked projects
```

- **Project access key**: Available under **Management > Project Management** in the WhaTap project.
- **Language settings**: The `/language` command applies only within the LINE chat room and does not affect other notification channels.

ilert

1. In the ilert console, click [+ Create alert source] at the top right of the **Alert sources** page.
2. In the **Integration type** step, enter `WhaTap` in the **Filter integrations** search field at the bottom to search.
3. From the search results, select the **WhaTap** tile and click the [Next] button.
4. Enter the alert source name and assign a team, then click the [Next] button.
   - You can create a new escalation policy or select an existing one.
5. Select the alert grouping method and click the [Continue setup] button.
   - The initially selected **Do not group alerts** option can be changed later.
6. On the next screen, you can configure additional settings such as **alert templates and priorities**. For now, click the [Finish setup] button to complete the setup.
7. When the setup is complete, an **API key** or **Webhook URL** will be generated on the completion screen.

8. Copy the generated **Webhook URL** from the **Webhook JSON** tab and paste it into the **Webhook URL** input field below.
9. Click the [Register] button.

Preventing Alert Storms

If a large number of alerts occur within a short period, alert notifications are temporarily paused for a specified duration. For example, if 20 events occur within 5 minutes, alert notifications will be suspended for the next 5 minutes. Once the configured **Pause Duration** has elapsed, the alert-storm prevention feature is automatically released.

1. Go to **Alert** > **Notifications** > the **Avoids a large number of alerts** section.
2. Click the toggle button to **Enable** or **Disable** the feature.
   - **Enabled**: Temporarily pause alerts when an alert storm is detected
   - **Disabled**: Resume alert notifications
3. Configure the **Detection Time**, **Detection Count**, and **Pause Duration**.
   - If the number of events exceeds the **Detection Count** within the **Detection Time**, alert notifications are paused for the configured **Pause Duration**.

When alerts are blocked due to an alert storm, a notification banner is displayed. To immediately release the alert-storm block, click the **Stop Alert Storm Prevention** button in the banner.

Event Notification Format

Common Alert Template

The common template can be applied in most cases. It allows you to receive alerts in a consistent format across various environments.

- Product Type: Application, Database, Kubernetes
- Event Type: Application Alerts, Database Alerts, Metrics Alerts
- Event Channel: SMS, Mobile, 3rd-party plugins, Plugins

Event Title

Event title format

```
[Level][Platform][ProjectName][ApplicationName][EventTitle]
```

Event title example

```
[Info][JAVA][Application Project][TC-0-1-8081][CRITICAL_HIGH_MEMORY]
```

> If ApplicationName is not set, it is omitted from the event title.
>
> - Platform is displayed as one of the following.
>   - 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

Event Message

This section describes the information that can be included in the event message. If `optional` is set to `false`, the item is always included in the message. If `optional` is set to `true`, the item is displayed only when the data is available.

Event message example

```
Project Name : Application Project
Project Code : 3
Agent Name : TC-0-1-8081
Message : RECOVERED: Memory is too high. less than 10%
```

```
Event 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
```

Table | Event Message Components

En	Ko	Supported Alert Types	Description
Project Name	Project name	All	-
Project Code	Project code	All	-
Agent Name	Agent name	All (optional)	oname
Message	Event message	All	-
Alert Type	Alert type	All	See the AlertType table below
Event Time	Event occurrence time	All	In `2022-04-13 10:40:49 +0900`, `+0900` indicates the GMT offset
Event Off Time	Event recovery time	All (optional)	In `2022-04-13 10:40:49 +0900`, `+0900` indicates the GMT offset
Metric Name	Metric name	All (optional)	Name of the metric used to evaluate the event condition
Metric Value	Metric value	All (optional)	Included when the metric value exceeds the threshold
Metric Threshold	Metric threshold	All	Threshold value used to determine the event

En	Ko	Supported Alert Types	Description
		(optional)	condition
Stateful	Recovered-event notification	All (optional)	`true` if recovered-event notification is enabled; otherwise `false`
Event Rule	Event trigger condition	Metrics alerts	-
Target Filter	Event target filter	Metrics alerts	Evaluates event conditions only for metrics collected from specific targets
Repeat Count	Event repeat count	Metrics alerts	The event is triggered when the condition is met the specified number of times within the repeat duration
Repeat Duration	Event repeat duration	Metrics alerts	Time window used to evaluate the repeat count
Receiver	Receiver	Metrics alerts	-
Query	MXQL query	Composite metrics alerts	-
Rule	Event trigger condition	Composite metrics alerts	-
Query Period	Query period	Composite metrics alerts	-
Query Interval	Query interval	Composite	-

En	Ko	Supported Alert Types	Description
		metrics alerts	
Silent Time	Silent time	Composite metrics alerts	-
Query	URL	Exception alerts	URL of the request that caused the exception
TXID	Transaction ID	Exception alerts	-
Class	Error class name	Exception alerts	-
Log Message	Log message	Server – file log alerts	-
Log File	Log file path	Server – file log alerts	-
IP	IP address	All server alerts	-
CPU	CPU usage	All server alerts	Snapshot at the time the event occurred
CPU_load1	CPU load (1 min)	All server alerts	Snapshot at the time the event occurred
CPU_loadPerCore	CPU load per core	All server	Snapshot at the time the event occurred

En	Ko	Supported Alert Types	Description
		alerts	
Memory	Memory usage	All server alerts	Snapshot at the time the event occurred
Swap	Swap usage	All server alerts	Snapshot at the time the event occurred
Disk Name | Used Percent | Free Size | IO Percent	Disk performance	All server alerts	Snapshot at the time the event occurred
Name | Bps | Pps	Network traffic performance	All server alerts	Snapshot at the time the event occurred
Message | Time | Name	Acknowledgement information	All server alerts	-

This section displays as much information as possible that can be provided for the event.

- **AlertType** is displayed as one of the following values.

AlertType	Description
APPLICATION_CPU	Application CPU alert
APPLICATION_MEMORY	Application memory alert
APPLICATION_DISK	Application disk alert
APPLICATION_ACTIVE_TRANSACTION	Application active transaction alert
APPLICATION_ERROR_TRANSACTION	Application error transaction alert

AlertType	Description
APPLICATION_SLOW_TRANSACTION	Application slow transaction (response time) alert
METRICS	Metrics alert
COMPOSITE_METRICS	Composite metrics alert
ANOMALY	Anomaly detection alert
LOG_REALTIME	Real-time log alert
COMPOSITE_LOG	Composite log alert
SERVER_REBOOT	Server reboot alert
SERVER_NO_DATA	Server no-data alert
SERVER_PORT	Server port alert
SERVER_NETWORK_IOPS	Server network IOPS alert
SERVER_NETWORK_BPS	Server network BPS alert
SERVER_DISK_IO	Server disk I/O alert
SERVER_DISK_QUOTA	Server disk usage alert
SERVER_DISK_INODE	Server inode alert
SERVER_CPU	Server CPU alert
SERVER_MEMORY	Server memory alert
SERVER_CPU_STEAL	Server CPU steal alert

AlertType	Description
SERVER_MEMORY_SWAP	Server swap memory alert
SERVER_LOG_FILE	Server log file alert
SERVER_WINDOW_EVENT	Server Windows event alert
SERVER_OFF	Server alert OFF notification
SERVER_ACKNOWLEDGE	Server acknowledge notification
SERVER_PROCESS_COUNT	Server process count alert
SERVER_PROCESS_CPU	Server process CPU alert
SERVER_PROCESS_MEMORY	Server process memory alert
SERVER_PROCESS_OFF	Server process alert OFF notification
AGENT_ACTIVE	Agent activated alert
AGENT_INACTIVE	Agent deactivated alert
AGENT_REACTIVATED	Agent reactivated alert
URL	URL alert
TOO_MANY_EVENT	Too many events alert
CLOUD_WATCH	CloudWatch alert
EXCEPTION	Exception alert

Event History And Monitoring

Home > Select Project > **Alert** > **Event History**

Event History allows you to view events that have occurred in the project and analyze their detailed information. It supports Elasticsearch DSL queries, enabling you to find events that match specific conditions quickly and accurately.

Viewing Event History

Click the **event title** in the search results to move to a screen where you can analyze detailed data for the time period when the event occurred. This allows you to comprehensively understand the situation before and after the event.

1. Go to **Alert** > **Event History**.
2. In the **Event History** screen, use **Time Selection** to set the period for viewing event records.
3. Enter a query in the filter input field and click the [Magnifying Glass] button.
    - If no query is entered, all events are displayed.

Column Settings

You can choose which columns to display or hide.

1. In the **Event History** screen, click the [Column Settings] button at the top left.
2. In the **Column Settings** window, select the columns you want to show or hide.
    - You can change the column order in **Displayed Items**.
    - You can search for specific columns using the **Column Name Search** field.

Column Item	Description
Event Name	The title of the event

Column Item	Description
Message	The event message or snapshot information
Event Status	The current status of the event
Agent Name	The name (Oname) of the agent where the event occurred - Displayed as blank if the event is not associated with a specific agent
Agent Group	The agent type (OkindName) where the event occurred - Displayed as blank if the event is not associated with a specific agent
Agent Node	The server (OnodeName) of the agent where the event occurred - Displayed as blank if the event is not associated with a specific agent
Event Occurred Time	The time when the event occurred
Event Resolved Time	The time when the event was resolved - Displayed as `-` for events without a resolved state

3. After completing the settings, click the [Confirm] button to apply the column configuration.

CSV Download

You can download the query results as a CSV file.

1. Go to **Alert** > **Event History** and click the [CSV] button.
2. The list of events that match the current filter conditions is saved as a CSV file.

View Only Events in Progress

You can view only events that are currently in progress.

1. Go to **Alert** > **Event History** and click the [View Only Events in Progress] button.
2. The following condition is automatically added to the filter query and applied to the search.

```
Stateful: true and Status : "ON"
```

Event Search

- **Precise Search**: Specify exact conditions to retrieve only the events you need
- **Composite Conditions**: Search across multiple fields at once to handle complex conditions in a single query
- **Flexible Patterns**: Supports various search methods such as wildcards and partial matches
- **High Performance**: Fast searches powered by indexed data

Searchable Fields

Field names and values are not case sensitive.

- **Field Name Tag**: Event types that support the field
- **Field Type**: The data type of the field
  - #: Number
  - T: String
  - : Boolean
  - : Date

Event Information Fields

Field Name	Type	Description	Example Value
Title	String	Event title	"Database Connection Error"
OffTitle	String	Event recovery title	"RECOVERED: Database Connection Error"
Message	String	Event message	"Connection timeout occurred"
OffMessage	String	Event recovery message	"RECOVERED: Connection timeout occurred"

Field Name	Type	Description	Example Value
Level	String	Current event level	`Critical`, `Warning`, `Info`
OriginLevel	String	Original event level	`Critical`, `Warning`, `Info`
Status	String	Event status	`ON`, `OFF`, `CANCEL`, `ACKNOWLEDGE`, `MAINTENANCE`, `DISABLED`
ActCount	Number	Number of occurrences	2
MetricName	String	Metric name	"memory"
MetricValue	String	Metric value	"85.5"
OffValue	String	Recovery value	"72.8"
MetricThreshold	String	Threshold value	"80"
alertType	String	Event type	"BASIC", "METRICS", "TRANSACTION", etc.
alertId	String	Unique identifier of the event rule	"zf3uojer0fv4v7"

Type-specific Supported Fields

> ⓘ These fields are supported only for specific event types. If the event type does not match, the value may be empty.

Field Name	Type	Event Type	Description	Example Value
eventRule	String	Basic, Metrics	Event trigger rule	"memory ≥ 80"
field	String	Real-	Log search	"content"

Field Name	Type	Event Type	Description	Example Value
		time Log	key	
keyword	String	Real-time Log	Log search value	"Error"
logCategory	String	Real-time Log	Log category	"AppLog"
logContent	String	Real-time Log	Log content	"00:00000:00009:2025/04/06 23:03:52.55 server Error: 1601, Severity: 17, State: 3\n00:00000:00009:2025/04/06 23:03:52.55 server There are not enough 'user connections' available to start a new process."

Target Fields

Field Name	Type	Description	Example Value
Oid	Number	Agent unique identifier	-98765432
Oname	String	Agent name	"web-server-01"
Okind	Number	Agent type unique identifier	867318026
OkindName	String	Agent type name	"web-server"
Onode	Number	Agent node unique identifier	334634079
OnodeName	String	Agent node name	"production-node-1"

Unique Event Fields

Field Name	Type	Description	Example Value
Id	Number	Unique identifier of the event	5768121
UUID	String	Unique event identifier	"550e8400-e29b-41d4-a716"

State / Flag Fields

Field Name	Type	Description	Example Value
Stateful	Boolean	Whether the event is state based	true, false
Disabled	Boolean	Whether the event is disabled	true, false
Escalation	Boolean	Whether escalation is applied	true, false
SystemEvent	Boolean	Whether it is a system event	true, false

Query Syntax by Search Type

> ⓘ For the basic search syntax, refer to the WhaTap log search query syntax documentation.

1. Keyword Search

Events that contain the string "Database Connection"

```
title: "Database Connection"
```

2. Multiple Value Search

Retrieves events that include any one of the specified values.

```
title: Database Connection
```

Events whose current event level is Info or Warning

```
level: info warning
```

3. Pattern Search

You can search using patterns with wildcards.

Find events that occurred on specific agents using a name pattern

```
oname: web-*-prod
```

タイトルが Connection で終わるイベントを検索

```
title: *Connection
```

4. Complex Condition Search

You can combine multiple conditions to perform precise searches.

Events with a Critical level and a title containing Database

```
level: critical and title: Database
```

Warning or Critical events from a specific agent

```
oname: web-server-01 and level: Warning Critical
```

5. OR Condition Search

Retrieves events that satisfy **at least one** of the specified conditions.

Events where the title or message contains a specific keyword

```
title: Connection or message: Connection
```

6. Exclusion Conditions

You can exclude specific conditions from the search.

All events except those with an Info level

```
not level: info
```

Real-time Notifications

The **Alert status** feature displays pop-up notifications when events configured in **Event Configuration** occur. You can receive alerts immediately when an event is triggered, allowing you to quickly recognize issues and respond. You can also review notifications that have already occurred or were missed, helping you understand and resolve situations where real-time response was difficult.

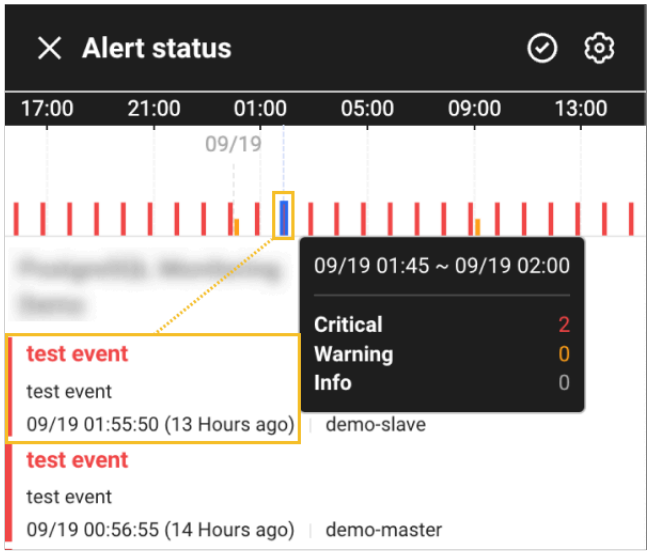
Viewing Real-time Notifications

Real-time notifications appear as pop-ups in the upper-right corner of the screen. You can check the time, title, and message of each notification.

- To close an individual notification, click the ✕ button in the upper-right corner of the message.
- To close all pop-up notifications displayed on the screen, click **Close All Notifications**.

Viewing Notification History

1. Click the 🔔 icon at the top of the screen after selecting a project.
2. The **Alert status** window appears.
    - You can view up to 100 notifications from the past 24 hours for the current project.
    - Selecting an individual event item navigates you to the related menu.
    - In the time-based chart within the **Alert status** window, selecting a specific time range takes you to the corresponding event items.



- To view the full notification history, click **View All Notification History**. This navigates to **Alert** > **Event History**.

> If the **Alert status** list is empty, the **View All Notification History** button is displayed. Disable the **Receive notifications without reception tags** option.

Resolving Events in Bulk

You can resolve all ongoing events at once from the **Alert status** list. The bulk resolution process may take approximately 30 seconds to 1 minute to complete.

- When using the **Add Acknowledge** feature, a large number of recovery (resolved) notifications may be generated for currently ongoing events.
- Even after using the **Add Acknowledge** feature, events may occur again if the event rules remain enabled and the data still exceeds the threshold.
- Recovery notifications can be disabled by editing the event in **Alert** > **Event Configuration**. Disable **Send additional notification when the event is resolved** or **Recovered notification**.

How to use:

1. Click the icon at the top of the screen after selecting a project.
2. When the **Real-time Notifications** window appears, click the icon in the upper-right corner.
3. When the message `Resolve all ongoing events in the real-time notification list.` appears, click the **Resolve** button.

Configuring Real-time Notifications

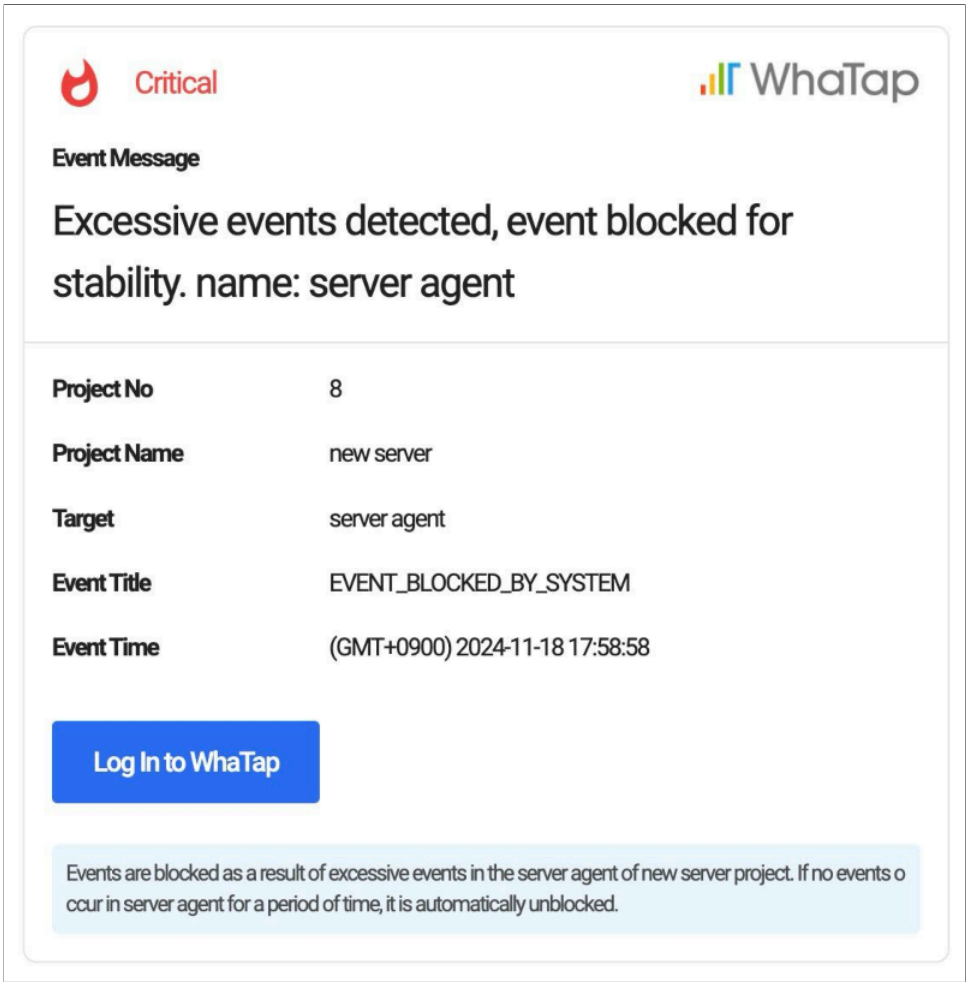
1. Click the icon at the top of the screen after selecting a project.
2. When the **Alert status** window appears, click the icon in the upper-right corner.
3. Configure each option in the real-time notification settings window.
   - **Popup notifications**: Displays notifications as pop-ups even when the **Alert status** window is closed.
   - **Show resolved events**: Allows resolved events to be displayed in the **Alert status** list.
   - **Preview notification sound**: Lets you preview browser notification sounds. Click the button next to Critical or Warning. If you cannot hear the sound, check your browser settings.

System Events

WhaTap Monitoring provides common system events that are automatically detected and generated across the platform. This document explains the types of events generated by the system, their trigger conditions, and recommended response actions.

EVENT_BLOCKED_BY_SYSTEM

If an excessive number of events occurs within a short period of time in a project, the WhaTap system detects this situation and blocks further events. When events are blocked, the system delivers the following message to the user. If you encounter this event, review and adjust your event configuration settings.



Blocking Criteria

Events are blocked based on the following criteria:

1. **Agent-level blocking**

   If **10 events** occur within **1 minute** from a single agent, all subsequent events from that agent are blocked.

2. **Project-wide blocking triggered by multiple agents**

   If **3 or more agents** are blocked within **1 minute** according to condition **1**, events from the entire project are blocked.

3. **Project-level blocking**

   If **100 events** occur within **1 minute** in a single project, all events from that project are blocked.

How to Release the Block

When events are blocked by the system, no events must occur from the affected agent or project for **1 minute**.

- Disable all configured events from **Alert > Event Configuration**. **Review your event settings and adjust thresholds to prevent excessive event generation.**
- Before configuring events, use the **Simulation** feature to estimate how many events may occur.

Java 실험실

사용자에게 새로운 기능 또는 실험적인 기능을 제공합니다. 사이드 메뉴의 **실험실** 또는 **사이트맵**을 통해 접근할 수 있습니다. 제품에 따라 제공하는 기능이 다를 수 있습니다.

사이트맵으로 실험실 메뉴 접근하기

사이드 메뉴에서 **실험실**가 보이지 않는다면 **사이트맵**으로 접근할 수 있습니다.

1. 전체 화면 왼쪽 아래에 **사이트맵** 아이콘을 클릭하세요.
2. **사이트맵** 창의 오른쪽 아래에 **실험실** 섹션에서 원하는 메뉴를 선택하세요.

주의

실험실의 기능은 현재 개발 중인 베타 버전으로 예기치 않은 오류가 발생할 수 있습니다.
해당 기능 사용 시 주의가 필요합니다. 중요한 데이터나 운영 환경에서 사용을 권장하지 않습니다. 피드백이나 문제점이 발생하면 지원팀(support@whatap.io)으로 문의하시기 바랍니다.

MXQL 데이터 조회

MXQL은 와탭의 성능 데이터(메트릭스)를 유연하게 조회하기 위한 쿼리 언어입니다. 하나의 프로젝트에 포함된 여러 에이전트에서 수집한 메트릭스들을 종합적으로 조회하고 활용하기 위해서 사용합니다. **MXQL**에 대한 자세한 내용은 다음 문서를 참고하세요.

비즈니스 대시보드

비즈니스 대시보드는 애플리케이션의 비즈니스 단위(App Context)별 성능을 모니터링할 수 있는 대시보드입니다. URL 패턴이나 경로를 기반으로 정의된 각 비즈니스 항목에 대해 실시간 및 과거의 성능 데이터를 조회하고 분석할 수 있습니다. TPS, 응답시간, 에러율 등의 주요 성능 지표를 비즈니스 단위로 확인함으로써, 특정 업무나 서비스의 성능 상태를 빠르게 파악할 수 있습니다.
자세한 내용은 다음 문서를 참고하세요.

에이전트 지원 버전

- **Java**: 2.2.40 이상

- **PHP**: 2.11.0 이상
- **Python**: 1.8.5 이상
- **.NET**: 2.4.3 이상

Collection Map

JVM에 지정된 클래스(Class)가 사용하는 Collection(예. `java.util.Map`) 정보를 제공합니다. 이 기능을 사용하려면 아래 에이전트 옵션을 추가하세요.

whatap.conf

```
collection_nonstatic_holders=io.home.test.baseapp.app.post.web.PostResource
```

에이전트 지원 버전

Java 에이전트 2.1.2 버전 이상에서 지원합니다.

외부 호출 상태

애플리케이션 에이전트의 설정을 통해 각 모니터링 대상별 외부(remote) 호출에 대한 성공/실패 여부를 기록한 로그를 시각화하여 제공합니다.

이 기능을 이용하려면 아래 옵션을 에이전트 설정(whatap.conf) 파일에 추가하거나 **관리** > **에이전트 설정** 메뉴에서 추가하세요.

whatap.conf

```
logsink_rt_enabled=true
logsink_enabled=true
```

알림 메시지

프로젝트 멤버에게 메일 또는 서드 파티 플러그인을 통해 전달하는 경고 알림 메시지를 수정할 수 있는 사용자 정의 기능을 제공합니다. 자세한 내용은 다음 문서를 참고하세요.