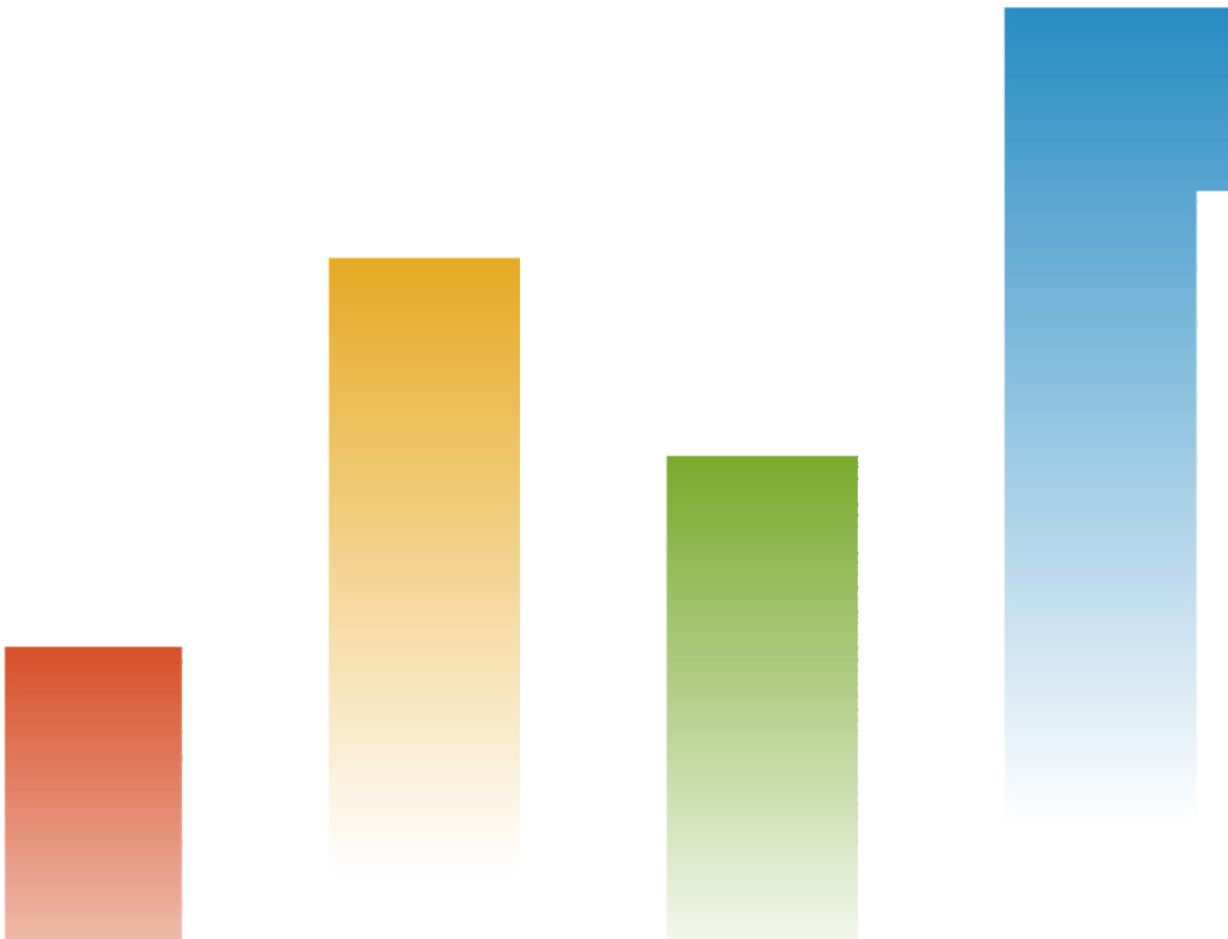
WhaTap

# Go Monitoring

release date. 2026. 09. 11.



# Table Of Contents


- Go Monitoring ..... 6
  - Getting started ..... 8
  - Support environment ..... 10
  - Installation
    - General installation ..... 17
    - Docker environment installation ..... 24
    - Installation checking items ..... 34
    - Data relay agent management ..... 36
    - Auto instrument guide
      - Basic usage ..... 38
      - Configuration guide ..... 47
      - Troubleshooting ..... 60
      - Custom instrumentation ..... 66
      - Multi-module projects ..... 80
    - Manual instrumentation guide ..... 90
  - Configuration ..... 116
    - Agent name identification ..... 124
    - Control of agent functions ..... 127
    - Agent communication settings ..... 129
    - Agent Log ..... 131
    - Transactions ..... 133
    - Tracing DB and SQL ..... 142
    - HTTPC, API Call ..... 143
    - User count ..... 144
    - Apdex ..... 146

- Statistics ..... 147
- Configuration for each library ..... 149

◦ Dashboard ..... 150

- Application Dashboard ..... 151
- Transaction Map ..... 179
- Active Transaction ..... 189
- HITMAP - Trace analysis ..... 193
- Flex Board ..... 218
  - Using Flex board ..... 221
  - Managing the widgets ..... 231

◦ Analysis ..... 236

- Daily Application Status ..... 238
- Performance Trend ..... 240
- CUBE ..... 244
- HITMAP ..... 248
- Multiple transaction trace ..... 274
- Transaction search ..... 280
- MSA Analysis ..... 290

◦ Metrics ..... 298

- Application metrics ..... 301
- Application performance counter ..... 320
- Metrics Chart ..... 328
- Metrics Chart ..... 351
- Metrics Search ..... 354
- Metric Anomaly Detection ..... 356

◦ Transactions ..... 359

- Analyzing the transactions....361
- Transaction Tracing....367
- Multi-transaction....393
- Transaction step collection method....397

◦ Application Topology....399

◦ Statistics....406

◦ Report....419

- Application report....422
- Integrated Report....430

◦ Alert....439

- Event Configuration....0
  - Metrics....446
  - Composite Metrics....468
  - Hitmap Patterns....475
  - Anomaly Detection....481
  - Real-time logs....485
  - Composite log....498
  - Transaction....512
- Application....524
- Event Notification Settings....528
  - Event Notification Format....550
- Event History and Monitoring....558
  - Real-time Notifications....566
  - System Events....569

◦ Instance Performance Management....572

◦ Correlated project management....577

◦ Log Monitoring ..... 581
◦ Laboratory ..... 0

# Go Monitoring

WhaTap Go Application Monitoring provides the monitoring service for the Go applications.

- It supports the web framework. It collects the data such as web transaction URL, response time, TPS information, error message, etc.
- It continuously collects data from the Go Runtime package. It also collects data including memory, goroutine, and GC related data.
- When multiple API/RPC services are called in the MSA environment, the call relationships are collected through link tracing.

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.

- **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.

- **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.

# Getting Started

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.

# Support Environment

## Go agent support environment

Before installing the WhaTap Go monitoring agent, check the support environment.

> ⓘ Last updated: 2026-01-26

### Supported versions

| Language | Supported version |
|---|---|
| Go | 1.18 or later |

### Operating system

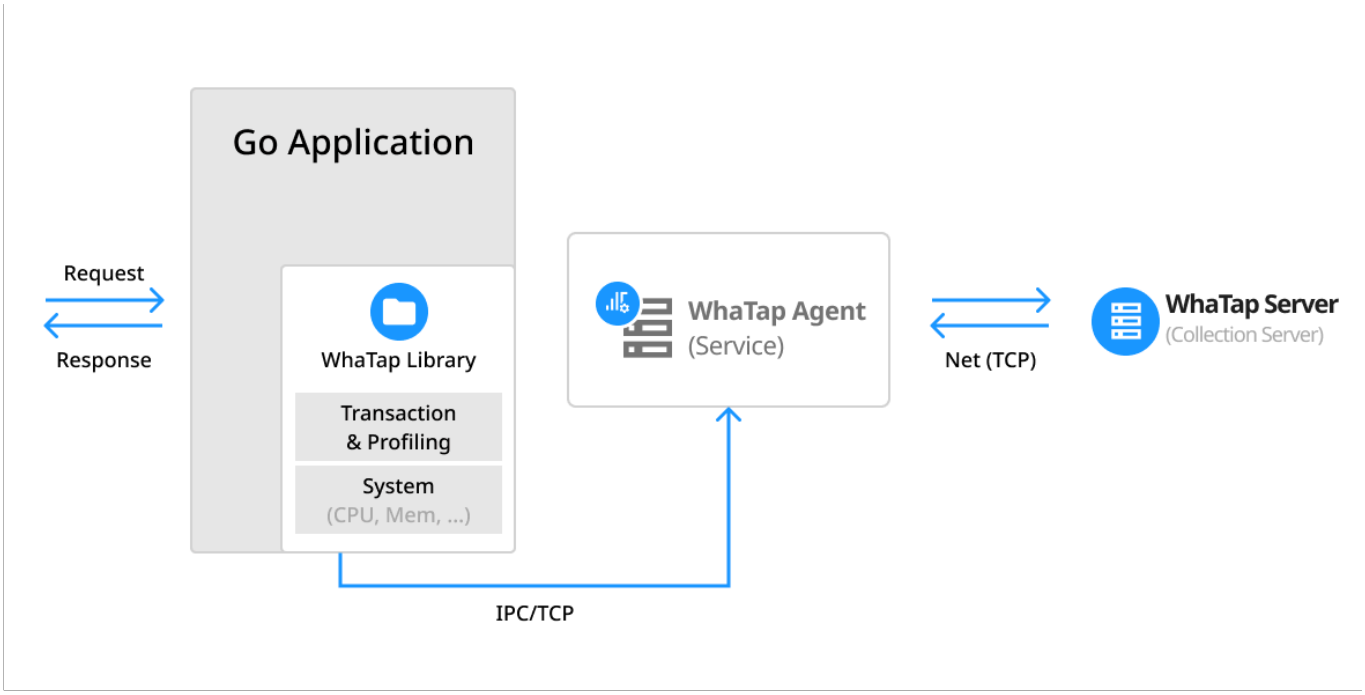
| OS | Minimum version | Architecture |
|---|---|---|
| CentOS / Red Hat | 6.x | 64bit |
| Debian / Ubuntu | 12.04 | 64bit |
| Amazon Linux | - | 64bit |
| Alpine Linux | - | 64bit |
| Linux | - | ARM64 (aarch64) |

## Library

| Category | Library |
|---|---|
| **Web framework** | net/http, gin, chi, fiber, gorilla, echo, fasthttp |
| **Database** | database/sql, gorm, redigo |
| **Others** | grpc, sarama, k8s.io/client-go |

ⓘ For framework or library application methods and examples, see the GitHub site.

# Go monitoring data flow diagram



### Communication structure

| Agent | Operation method |
|---|---|
| **Inbound** | Run as 127.0.0.1:6600 TCP server / Application → Agent (TCP 6600) |
| **Outbound** | Send received data to WhaTap collection server / Agent → WhaTap server (Outbound 6600) |

> ⓘ Go applications only communicate with the local agent, and the agent is responsible for external transmission.

# Common support 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.

## Browser support

WhaTap Monitoring is available in web browsers and mobile Apps.

| Browser | Recommended | Supported version |
|---|---|---|
| Google Chrome | O | 111 or later |
| 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

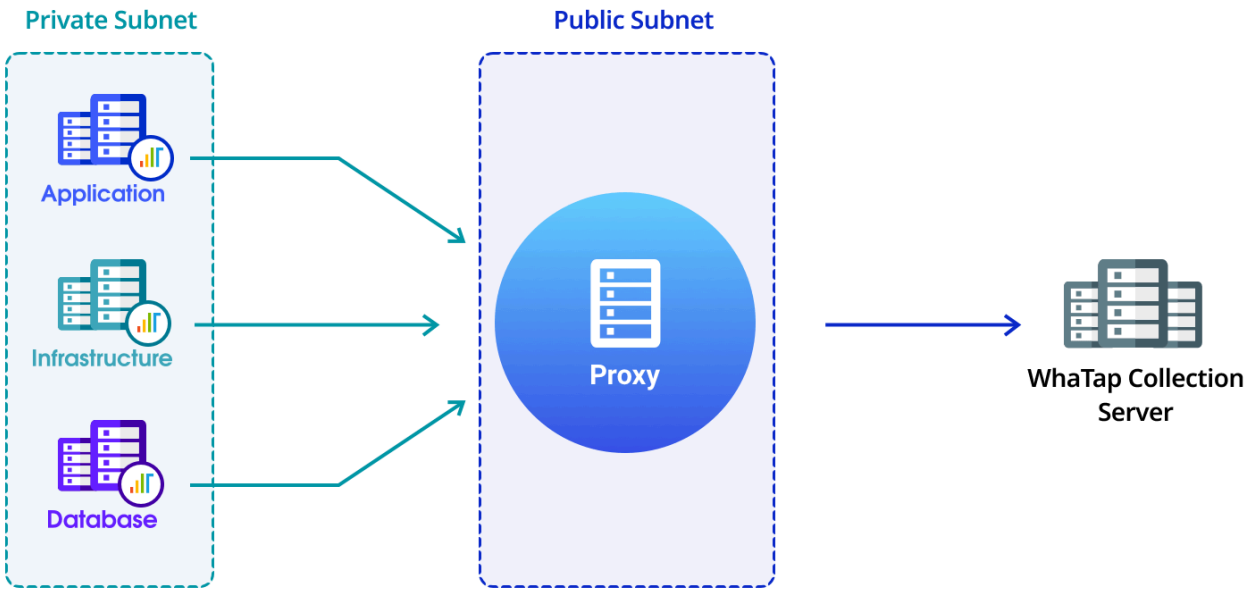
The WhaTap agent must be able to access the collection server through **TCP port 6600**. Allow access to the

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 |
| | 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 | Android |
| --- | --- |
| iOS 12 or later | Android 5.0 or later |

# General Installation

To install the WhaTap monitoring agent on a Go application in a Linux environment, follow these steps:


- Step 1. Apply Go library
- Step 2. Set monitoring environment variables
- Step 3. Install data relay agent


> Before starting the installation, prepare the following information:
>
> - WhaTap project **access key** (check from [Project menu] > [Management] > [Agent installation])
> - WhaTap **collection server IP address**

## Apply Go library

The following are methods to add the WhaTap Go Library to the application to be monitored. **Automatic instrumentation is recommended.**

**Library configuration methods**

| Category | Configuration method | Description | Note |
|---|---|---|---|
| Automatic instrumentation | **Using whatap-go-inst** | Automatic code insertion with `whatap-go-inst` tool | **Recommended** |
| Manual instrumentation | **Direct use of go-api** | Add API directly to code | When fine-grained control is needed |

### Automatic instrumentation (Recommended)

Automatically insert monitoring code into source code using the `whatap-go-inst` tool.

## 1. Install `whatap-go-inst`

| Go Version | Installation method |
| --- | --- |
| **Go 1.21+ (Recommended)** | Use go install |
| **Go 1.18~1.20** | Download binary |

- **Go 1.21+ (Recommended)**

```
go install github.com/whatap/go-api-inst/cmd/whatap-go-inst@latest
```

- **Go 1.18~1.20**

  Download pre-built binaries.

  Linux x86_64:

```
wget https://github.com/whatap/go-api-inst/releases/latest/download/whatap-go-inst_linux_amd64.tar.gz
tar -xvzf whatap-go-inst_linux_amd64.tar.gz
sudo mv whatap-go-inst /usr/local/bin/
```

Linux ARM64:

```
wget https://github.com/whatap/go-api-inst/releases/latest/download/whatap-go-inst_linux_arm64.tar.gz
tar -xvzf whatap-go-inst_linux_arm64.tar.gz
sudo mv whatap-go-inst /usr/local/bin/
```

## 2. Initialize project and build

Run the following commands in the project root directory where `go.mod` is located.

> If you are installing `whatap-go-inst` for the first time, you need to set the PATH environment variable first.

```
# Add environment variable (only needed for first installation)
```

```
export PATH=$PATH:$(go env GOPATH)/bin

# Download and install go-api package
go get github.com/whatap/go-api@latest
go mod tidy

# Build (automatically insert monitoring code)
whatap-go-inst go build -o myapp .
```

ⓘ For detailed usage, see the Automatic instrumentation guide.

## Manual instrumentation

Download `go-api` directly and manually insert monitoring code into source code.

### 1. Download `go-api`

```
go get github.com/whatap/go-api
```

### 2. Add monitoring code

```
import "github.com/whatap/go-api/trace"

func main(){
    trace.Init(nil)
    defer trace.Shutdown()

    ctx, _ := trace.Start(context.Background(), "Start Tx")
    ...
    trace.End(ctx, err)
}
```

ⓘ For detailed API usage, see the Manual instrumentation guide.

# Set monitoring environment variables

Monitoring environment variables are configured in the `whatap.conf` file.

`whatap.conf` must be created in the **working directory where the application runs**.

## Key settings

| Environment variable | Description |
|---|---|
| `license` | Project access key |
| `whatap.server.host` | WhaTap collection server address |
| `app_name` | Application name |

## `whatap.conf` location by execution method

| Execution method | Working Directory | `whatap.conf` location |
|---|---|---|
| `./myapp` | Current directory | `./whatap.conf` |
| `/home/user/app/myapp` | `/home/user/app` | `/home/user/app/whatap.conf` |
| Run with absolute path | `/` (root) | Requires `WHATAP_HOME` setting |

## Configuration methods

### Basic execution

```
echo "license={access key}" >> ./whatap.conf
echo "whatap.server.host={collection server IP address}" >> ./whatap.conf
```

```
echo "app_name={application name}" >> ./whatap.conf
./app
```

### WHATAP_HOME environment variable setting

```
mkdir ./whatap_home
echo "license={access key}" >> ./whatap_home/whatap.conf
echo "whatap.server.host={collection server IP address}" >> ./whatap_home/whatap.conf
echo "app_name={application name}" >> ./whatap_home/whatap.conf
WHATAP_HOME=./whatap_home ./app
```

### Agent name setting per process

For applications running with two or more processes, it is recommended to configure a separate `whatap.conf` file for each process.

```
# Application 1
mkdir ./whatap_home
echo "license={access key}" >> ./whatap_home/whatap.conf
echo "whatap.server.host={collection server IP address}" >> ./whatap_home/whatap.conf
echo "app_name={application name-1}" >> ./whatap_home/whatap.conf
WHATAP_HOME=./whatap_home ./app

# Application 2
mkdir ./whatap_home1
echo "license={access key}" >> ./whatap_home1/whatap.conf
echo "whatap.server.host={collection server IP address}" >> ./whatap_home1/whatap.conf
echo "app_name={application name-2}" >> ./whatap_home1/whatap.conf
WHATAP_HOME=./whatap_home1 ./app
```

> ⓘ `app_name` is a component for determining the agent name. For more information, see the following documentation.

# Install data relay agent

Once you have obtained the **access key**, download the agent using the following commands:

- **Red Hat/CentOS** **Debian/Ubuntu** **Amazon Linux** **Alpine Linux**

SH

```
## Register package repository
$ sudo rpm -Uvh https://repo.whatap.io/centos/5/noarch/whatap-repo-1.0-1.noarch.rpm

## Install package
$ sudo yum install whatap-agent
```

SH

```
## Register package repository
$ wget https://repo.whatap.io/debian/release.gpg -O -|sudo apt-key add -
$ wget https://repo.whatap.io/debian/whatap-repo_1.0_all.deb
$ sudo dpkg -i whatap-repo_1.0_all.deb
$ sudo apt-get update

## Install package
$ sudo apt-get install whatap-agent
```

```
sudo rpm --import https://repo.whatap.io/centos/release.gpg
echo "[whatap]" | sudo tee /etc/yum.repos.d/whatap.repo
echo "name=whatap packages for enterprise linux" | sudo tee -a /etc/yum.repos.d/whatap.repo
echo "baseurl=https://repo.whatap.io/centos/latest/\$basearch" | sudo tee -a /etc/yum.repos.d/whatap.repo
echo "enabled=1" | sudo tee -a /etc/yum.repos.d/whatap.repo
echo "gpgcheck=0" | sudo tee -a /etc/yum.repos.d/whatap.repo
sudo yum install whatap-agent
```

- **x64** **AArch64**

x64

```
wget https://s3.ap-northeast-2.amazonaws.com/repo.whatap.io/alpine/x86_64/whatap-agent.tar.gz
tar -xvzf whatap-agent.tar.gz -C /
```

AArch64

```
wget https://s3.ap-northeast-2.amazonaws.com/repo.whatap.io/alpine/aarch64/whatap-agent.tar.gz
tar -xvzf whatap-agent.tar.gz -C /
```

# Start monitoring

When you restart the application server, the agent starts collecting information.

# Docker Environment Installation

This guide provides instructions on how to install the WhaTap monitoring agent in Docker container-based Go applications. Choose one of the following two methods:

- **Quick installation method**: Complete installation process from writing Dockerfile to running containers and checking monitoring data
- **Configuration by deployment environment**: Additional configuration methods for each architecture (x64/ARM64), Kubernetes, and Docker Compose environment

> Before starting the installation, prepare the following information:
>
> - WhaTap project **access key** (check from [Project menu] > [Management] > [Agent installation])
> - WhaTap **collection server IP address**

## Quick installation method

You can build a Docker image with WhaTap monitoring applied using only 3 files.

### Step 1. Prepare required files

Create the following 3 files in the project root.

- **1. Dockerfile** **2. entrypoint.sh** **3. main.go (example)**

```
# ============================================
# Build stage - whatap-go-inst build
# ============================================
FROM golang:1.21-alpine AS builder

WORKDIR /app

# Install whatap-go-inst
#RUN go install github.com/whatap/go-api-inst/cmd/whatap-go-inst@latest
```

```
RUN wget -qO- https://github.com/whatap/go-api-inst/releases/latest/download/whatap-go-inst_linux_amd64.tar.gz | tar xz
-C /usr/local/bin/

# Copy source
COPY go.mod go.sum ./
RUN go mod download
COPY . .

# Build whatap-go-inst
RUN whatap-go-inst go build -o /app/main .

# ============================================
# Runtime stage
# ============================================
FROM alpine:latest

WORKDIR /app

# Install WhaTap data relay agent
RUN apk add --no-cache wget
RUN wget https://s3.ap-northeast-2.amazonaws.com/repo.whatap.io/alpine/x86_64/whatap-agent.tar.gz
RUN tar -xvzf whatap-agent.tar.gz -C /
RUN rm whatap-agent.tar.gz

# Copy application
COPY --from=builder /app/main .

# Startup script
COPY entrypoint.sh /app/
RUN chmod +x /app/entrypoint.sh

ENTRYPOINT ["/app/entrypoint.sh"]
```

```
#!/bin/sh

# Generate whatap.conf (from environment variables)
mkdir -p /app/whatap_home
cat > /app/whatap_home/whatap.conf << EOF
license=${WHATAP_LICENSE}
whatap.server.host=${WHATAP_SERVER_HOST}
```

```
app_name=${WHATAP_APP_NAME:-myapp}
EOF

export WHATAP_HOME=/app/whatap_home

# Start data relay agent and run application
/usr/whatap/agent/whatap-agent start
exec ./main
```

```
package main

import "github.com/gin-gonic/gin"

func main() {
    r := gin.Default()
    r.GET("/", func(c *gin.Context) {
        c.JSON(200, gin.H{"message": "Hello WhaTap"})
    })
    r.Run(":8080")
}
```

## Step 2. Build image

```
docker build -t my-go-app .
```

## Step 3. Run container

Pass WhaTap configuration through environment variables.

```
docker run -d \
  -e WHATAP_LICENSE={access key} \
  -e WHATAP_SERVER_HOST={collection server IP} \
  -e WHATAP_APP_NAME=myapp \
  -p 8080:8080 \
  my-go-app
```

### Step 4. Check monitoring

1. Generate traffic to the application: `curl http://localhost:8080`
2. Check data in WhaTap Monitoring Service

> Refer to the Troubleshooting section.

# Configuration by deployment environment

## Dockerfile configuration by architecture

The quick start example above is based on **x64 (AMD64)**. For ARM64 environments, only change the agent download URL.

### WhaTap agent URLs by architecture

| Architecture | Agent download URL |
|---|---|
| **x64 (AMD64)** | `https://s3.ap-northeast-2.amazonaws.com/repo.whatap.io/alpine/x86_64/whatap-agent.tar.gz` |
| **ARM64 (AArch64)** | `https://s3.ap-northeast-2.amazonaws.com/repo.whatap.io/alpine/aarch64/whatap-agent.tar.gz` |

### ARM64 Dockerfile modification example

Only change the agent installation part in the Dockerfile:

```
# Change in Runtime stage
RUN apk add --no-cache wget && \
    wget https://s3.ap-northeast-2.amazonaws.com/repo.whatap.io/alpine/aarch64/whatap-agent.tar.gz && \
```

```
tar -xvzf whatap-agent.tar.gz -C / && rm whatap-agent.tar.gz
```

View complete ARM64 Dockerfile

```
FROM golang:1.21-alpine AS builder

WORKDIR /app

# Install whatap-go-inst
#RUN go install github.com/whatap/go-api-inst/cmd/whatap-go-inst@latest
RUN wget -qO- https://github.com/whatap/go-api-inst/releases/latest/download/whatap-go-
inst_linux_amd64.tar.gz | tar xz -C /usr/local/bin/

COPY go.mod go.sum ./
RUN go mod download
COPY . .

RUN whatap-go-inst go build -o /app/main .

FROM alpine:latest

WORKDIR /app

RUN apk add --no-cache wget && \
    wget https://s3.ap-northeast-2.amazonaws.com/repo.whatap.io/alpine/aarch64/whatap-agent.tar.gz && \
    tar -xvzf whatap-agent.tar.gz -C / && rm whatap-agent.tar.gz

COPY --from=builder /app/main .
COPY entrypoint.sh /app/
RUN chmod +x /app/entrypoint.sh

ENTRYPOINT ["/app/entrypoint.sh"]
```

## Kubernetes deployment

After building the Docker image, you can deploy it to a Kubernetes cluster.

### Step 1: Push Docker image to registry

```
# Tag image
docker tag my-go-app:latest your-registry/my-go-app:v1.0

# Push to registry
docker push your-registry/my-go-app:v1.0
```

### Step 2: Create WhaTap configuration Secret

```
kubectl create secret generic whatap-secret \
  --from-literal=license={access key}
```

### Step 3: Deploy Deployment

- deployment.yaml service.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-go-app
spec:
  replicas: 3
  selector:
    matchLabels:
      app: my-go-app
  template:
    metadata:
      labels:
        app: my-go-app
    spec:
      containers:
      - name: my-go-app
        image: your-registry/my-go-app:v1.0
        ports:
        - containerPort: 8080
        env:
        - name: WHATAP_LICENSE
```

```
        valueFrom:
          secretKeyRef:
            name: whatap-secret
            key: license
      - name: WHATAP_SERVER_HOST
        value: "{collection server IP}"
      - name: WHATAP_APP_NAME
        value: "my-go-app"
      # Kubernetes metadata (optional)
      - name: NODE_IP
        valueFrom:
          fieldRef:
            fieldPath: status.hostIP
      - name: NODE_NAME
        valueFrom:
          fieldRef:
            fieldPath: spec.nodeName
      - name: POD_NAME
        valueFrom:
          fieldRef:
            fieldPath: metadata.name
```

```
apiVersion: v1
kind: Service
metadata:
  name: my-go-app
spec:
  selector:
    app: my-go-app
  ports:
  - port: 80
    targetPort: 8080
  type: LoadBalancer
```

```
kubectl apply -f deployment.yaml
kubectl apply -f service.yaml
```

### Environment variable description

#### Required environment variables

| Environment variable | Description | Example |
|---|---|---|
| WHATAP_LICENSE | WhaTap project access key | x4t2r... |
| WHATAP_SERVER_HOST | WhaTap collection server IP | 13.124.11.223 |
| WHATAP_APP_NAME | Application name (optional) | my-go-app |

#### Kubernetes metadata (optional)

Add these to pass pod execution location information to WhaTap:

| Environment variable | Description |
|---|---|
| NODE_IP | IP of the node where the pod is running |
| NODE_NAME | Name of the node where the pod is running |
| POD_NAME | Name of the pod |

## Docker Compose

docker-compose.yml

```
version: '3.8'

services:
  my-go-app:
    build: .
    ports:
      - "8080:8080"
    environment:
      - WHATAP_LICENSE={access key}
```

```
    - WHATAP_SERVER_HOST={collection server IP}
    - WHATAP_APP_NAME=my-go-app
  restart: unless-stopped
```

```
docker-compose up -d
```

# Troubleshooting

## When monitoring data is not visible

### Step 1: Check environment variables

Check if environment variables are correctly set inside the container:

```
docker exec -it <container_id> sh
echo $WHATAP_LICENSE
echo $WHATAP_SERVER_HOST
cat /app/whatap_home/whatap.conf
```

### Step 2: Check agent execution

```
# Check agent process
docker exec -it <container_id> ps -ef | grep whatap

# Check agent logs
docker exec -it <container_id> cat /app/whatap_home/logs/whatap-*.log
```

**Normal output example:**

```
root  12  1  0 Jan21 ?  00:00:03 /usr/whatap/agent/whatap-agent
```

### Step 3: Check network

Check if outbound connection to WhaTap collection server (port 6600) is possible:

```
docker exec -it <container_id> nc -zv {collection server IP} 6600
```

## When instrumentation is not applied

### Check detailed instrumentation logs

```
# Debug output during build
GO_API_AST_DEBUG=1 whatap-go-inst go build ./...
```

## Common issues and solutions

| Symptom | Cause | Solution |
|---|---|---|
| Data not received | Environment variables not set | Check `WHATAP_LICENSE`, `WHATAP_SERVER_HOST` |
| Agent fails to start | Network blocked | Allow port 6600 outbound |
| Instrumentation not applied | Initialization missing | Check that `whatap-go-inst init` is added to Dockerfile |
| ARM64 error | Incorrect binary | Check agent URL for architecture |

## Additional support

If the above methods do not resolve the issue:

1. Contact WhaTap Support Center
2. Provide the following information:
   - Complete Dockerfile content
   - Agent logs ( `/app/whatap_home/logs/` )
   - `docker logs <container_id>` output

# Installation Checking Items

## Checking the service status

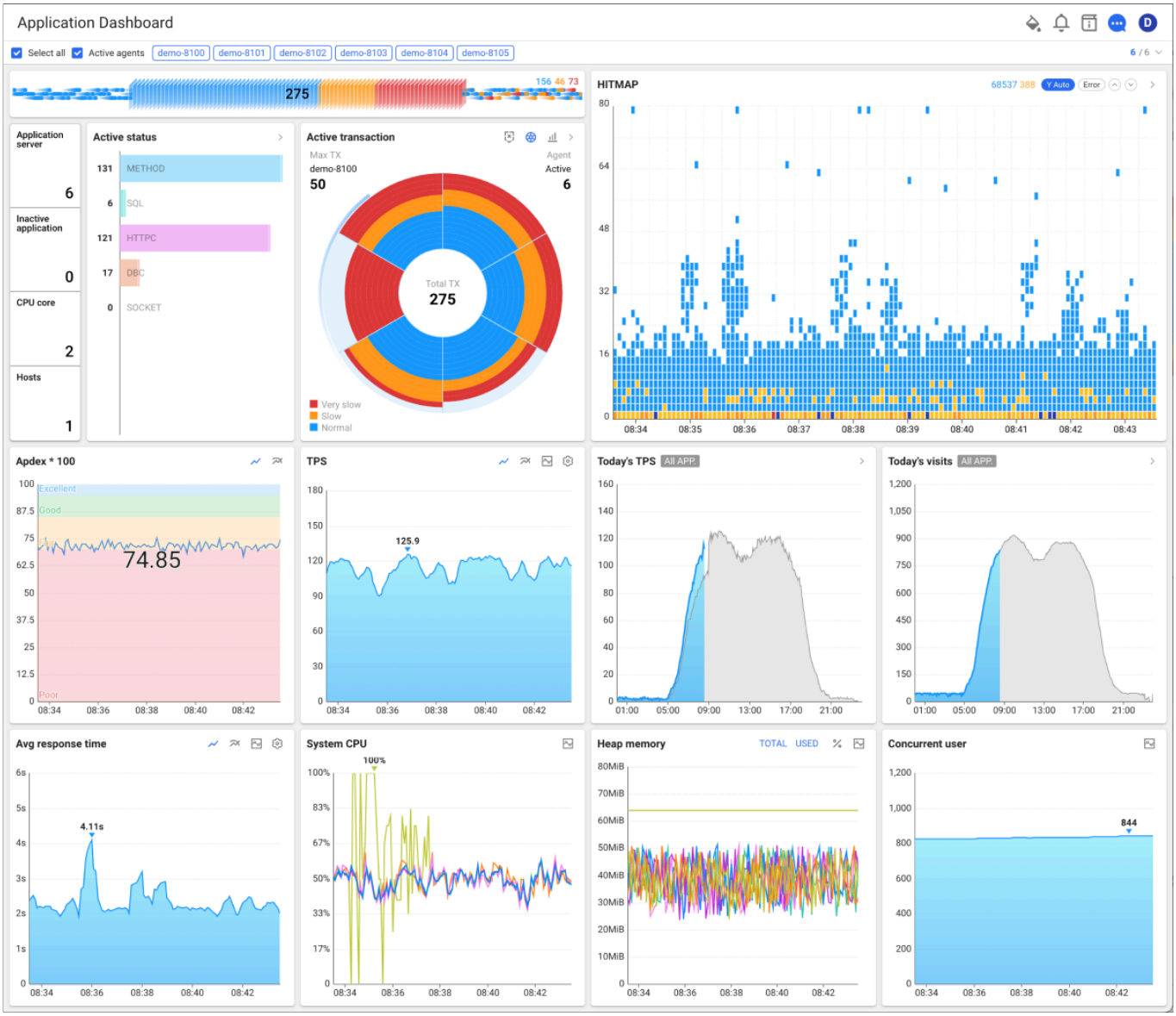
Check the status of the WhaTap agent service by using the following command:

SH

```
$ service whatap-agent status
```

## 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.


Application Dashboard
Select all
Active agents
demo-8100
demo-8101
demo-8102
demo-8103
demo-8104
demo-8105
275
Application server
6
Inactive application
0
CPU core
2
Hosts
1
Active status
131 METHOD
6 SQL
121 HTTPC
17 DBC
0 SOCKET
Active transaction
Max TX
demo-8100
50
Agent
Active
6
Total TX
275
Very slow
Slow
Normal
HITMAP
Apdex * 100
Excellent
Good
Poor
74.85
TPS
125.9
Today's TPS
All APP
Today's visits
All APP
Avg response time
4.11s
System CPU
100%
Heap memory
TOTAL
USED
Concurrent user
844

# Data Relay Agent Management

The following explains how to restart, pause, or delete the Go agent service.

## Restarting the agent service

If the WhaTap agent service is running or erroneous, restart it.

- Red Hat/CentOS Debian/Ubuntu Amazon Linux Alpine Linux

SH

```
$ service whatap-agent restart
```

SH

```
$ sudo service whatap-agent restart
```

SH

```
$ sudo service whatap-agent restart
```

SH

```
$ /usr/whatap/agent/whatap-agent stop
$ /usr/whatap/agent/whatap-agent start
```

## Stopping the whatap-agent service

- Red Hat/CentOS Debian/Ubuntu Amazon Linux Alpine Linux

SH

```
$ sudo service whatap-agent stop
```

SH

```
$ sudo service whatap-agent stop
```

SH

```
$ sudo service whatap-agent stop
```

SH

```
$ /usr/whatap/agent/whatap-agent stop
```

# Agent deletion

Delete the WhaTap library from the Go application's source code.

- Red Hat/CentOS Debian/Ubuntu Amazon Linux Alpine Linux

SH

```
$ sudo yum remove whatap-agent
```

SH

```
$ sudo apt-get purge whatap-agent
```

SH

```
$ sudo yum remove whatap-agent
```

SH

```
$ /usr/whatap/agent/whatap-agent stop

# /usr/whatap/agent Deleting the directory
$ sudo rm -r /usr/whatap/agent
```

# Basic Usage

whatap-go-inst is a CLI tool that automatically inserts monitoring code during build. You can add WhaTap monitoring to Go applications without directly modifying source code.

This document guides you through the installation and basic usage of whatap-go-inst.

> To add monitoring code by directly using the API, see the Manual instrumentation guide.

## Support environment

To use whatap-go-inst, the following environment is required.

| Item | Requirements | Note |
|---|---|---|
| **Go version** | 1.18 or later | Required when building instrumented code |
| **OS** | Linux | amd64, arm64 |

> The whatap-go-inst binary can run without Go installation. However, Go 1.18 or later is required to build instrumented code.

## Quick start

### Step 1: Install whatap-go-inst

There are 3 ways to install whatap-go-inst. Choose the method that suits your environment.

### Go install installation (Recommended)

This is the recommended installation method for Go 1.21 or later.

```
go install github.com/whatap/go-api-inst/cmd/whatap-go-inst@latest
```

### Direct binary installation

This is a method to install directly without Go installation. Users of Go 1.18-1.20 should use this method.

```
# Linux (amd64)
curl -sSL https://github.com/whatap/go-api-inst/releases/latest/download/whatap-go-inst_linux_amd64.tar.gz | tar xz
sudo mv whatap-go-inst /usr/local/bin/

# Linux (arm64)
curl -sSL https://github.com/whatap/go-api-inst/releases/latest/download/whatap-go-inst_linux_arm64.tar.gz | tar xz
sudo mv whatap-go-inst /usr/local/bin/
```

### Build from source

```
git clone https://github.com/whatap/go-api-inst.git
cd go-api-inst
go build -o whatap-go-inst .
```

## Step 2: Build and run

```
# Build with monitoring code automatically inserted
whatap-go-inst go build ./...

# Run
./myapp
```

You can now check data in WhaTap Monitoring Service.

Don't want to change the original code at all?

Using wrap mode, you can build without changes to `whatap_inst.tool.go` file or `go.mod`.

```
# Build directly without init
whatap-go-inst go --wrap build ./...
```

**Wrap mode features**

- No changes to original project files
- Useful for initial testing or temporary application
- Requires `--wrap` option every time

# whatap-go-inst detailed explanation

## How it works

`whatap-go-inst` analyzes source code during build and automatically inserts monitoring code.

### 1. main() function initialization

```
// Before change
func main() {
    // Application code
}

// After change
func main() {
    trace.Init(nil)        // Automatically added
    defer trace.Shutdown()  // Automatically added

    // Application code
}
```

### 2. Web framework middleware

```
// Before change
r := gin.Default()
r.GET("/api", handler)

// After change
r := gin.Default()
r.Use(whatapgin.Middleware()) // Automatically added
r.GET("/api", handler)
```

### 3. Database calls

```
// Before change
db, _ := sql.Open("mysql", dsn)

// After change
db, _ := whatapsql.Open("mysql", dsn) // Automatically changed
```

## Main supported libraries

- **Web frameworks**: Gin, Echo, Fiber, Chi, Gorilla Mux, net/http, FastHTTP
- **Databases**: database/sql, sqlx, GORM v1/v2
- **Cache/NoSQL**: Redis (go-redis, Redigo), MongoDB
- **External services**: gRPC, Kafka (Sarama), Kubernetes client-go
- **Logging**: log, logrus, zap

View complete list of supported libraries

| Category | Supported libraries |
|---|---|
| **Web frameworks** | Gin, Echo v4, Fiber v2, Chi v5, Gorilla Mux, net/http, FastHTTP |
| **Databases** | database/sql, sqlx, GORM v1/v2 |

| Category | Supported libraries |
|---|---|
| **Redis** | go-redis v8/v9, Redigo |
| **NoSQL** | MongoDB |
| **Message queue** | Kafka (Sarama - IBM/Shopify) |
| **RPC** | gRPC |
| **Kubernetes** | client-go |
| **Logging** | log, logrus, zap |

## Commands

| Category | Command | Description | Main usage |
|---|---|---|---|
| **Basic** | `whatap-go-inst go build` | Build with monitoring code inserted | Normal build |
| **Basic** | `whatap-go-inst go run` | Run with monitoring code inserted | Development/Test |
| **Basic** | `whatap-go-inst go test` | Test with monitoring code inserted | Unit test |
| **Advanced** | `whatap-go-inst inject` | Output instrumentation code to separate directory | Code review |
| **Advanced** | `whatap-go-inst remove` | Remove inserted monitoring code | Uninstrument |
| **Utility** | `whatap-go-inst version` | Check version | Version check |

After `whatap-go-inst go`, you can use it the same way as regular `go` commands.

```
# Example: Use like regular go commands
whatap-go-inst go build -o myapp -ldflags="-s -w" .
```

# Main options

## whatap-go-inst go options

| Option | Description |
|---|---|
| `--error-tracking` | Insert error tracking code (add `trace.Error` to `if err != nil` pattern) |

**Usage examples:**

```
# Wrap mode
whatap-go-inst go --wrap build ./...

# Include error tracking
whatap-go-inst go --error-tracking build ./...
```

## inject/remove options

| Option | Description |
|---|---|
| `-s, --src` | Source code path |
| `-o, --output` | Output directory |
| `--all` | (remove only) Also attempt to remove manual insertion patterns |

**Usage examples:**

```
# Save instrumentation code separately
whatap-go-inst inject -s ./src -o ./instrumented

# Remove instrumentation code
whatap-go-inst remove -s ./instrumented -o ./clean
```

## Global options

Common options that can be used with all commands.

| Option | Description |
|---|---|
| `--verbose, -v` | Verbose output (display conversion details per file) |
| `--quiet, -q` | Output summary only |
| `--config` | Configuration file path (default: `.whatap/config.yaml`) |
| `--report` | JSON report file path |

**Usage examples:**

```
# Verbose output
whatap-go-inst -v go build ./...

# Specify configuration file
whatap-go-inst --config ./my-config.yaml go build ./...
```

# Frequently asked questions

## Will the original code be changed?

The original code is not modified. whatap-go-inst copies source code to a tmp directory, downloads dependencies, and builds.

Build output is generated in the whatap-instrumented directory.

## What's different from existing go commands?

`whatap-go-inst go build` works the same as regular `go build`, but automatically inserts monitoring code

before building. All options and flags can be used as is.

```
# Use the same as regular build commands
whatap-go-inst go build -o myapp -ldflags="-s -w" .
```

## Will build time slow down?

Build time takes longer than a normal build due to copying to tmp directory, downloading dependencies, and code insertion.

## It seems instrumentation is not being applied?

Check the following:

```
# Delete build cache and rebuild
go clean -cache
whatap-go-inst go build ./...

# Check detailed logs in debug mode
GO_API_AST_DEBUG=1 whatap-go-inst go build ./...
```

For more details, see the Automatic instrumentation guide.

## Can it be used in Docker environment?

Yes, you can use it the same way in Dockerfile:

```
FROM golang:1.21 AS builder

WORKDIR /app

# Install whatap-go-inst
RUN go install github.com/whatap/go-api-inst/cmd/whatap-go-inst@latest

# Copy source
COPY . .
```

```
# Initialize and build
RUN whatap-go-inst go build -o /app/myapp .

FROM alpine:latest
COPY --from=builder /app/myapp /app/myapp
CMD ["/app/myapp"]
```

For more details, see the Docker installation guide.

# Configuration Guide

This guide explains how to control instrumentation behavior using whatap-go-inst configuration files and environment variables. You can easily configure using Presets or finely control by selecting individual packages.

For basic usage, see Basic usage first.

## Configuration file

### File location

Configuration files are searched in the following order.

| Priority | Location | Description |
|---|---|---|
| 1 | `--config` flag | `whatap-go-inst --config=/path/to/config.yaml` |
| 2 | `WHATAP_INST_CONFIG` environment variable | `WHATAP_INST_CONFIG=/path/to/config.yaml` |
| 3 | `.whatap/config.yaml` | `.whatap` directory in project root **(Recommended)** |
| 4 | `.whatap/whatap.yaml` | Alternative filename |

> It is recommended to create and use a `.whatap/config.yaml` file in the project root.

### Basic structure

```yaml
# .whatap/config.yaml
instrumentation:
  preset: "full"          # Preset selection (full/minimal/web/database/external/log/custom)
  error_tracking: false   # Whether to enable error tracking
  enabled_packages: []    # Additional packages to enable
```

```
  disabled_packages: []    # Packages to disable

exclude:                   # Instrumentation exclusion patterns (optional)
  - "**/*_test.go"
```

# Configuration items

## Preset options

You can easily select package groups to instrument using Presets.

| Preset | Included items | Description |
|---|---|---|
| `full` **(Default)** | Web + Database + External services + Log | Activate all packages |
| `minimal` | trace.Init/Shutdown | Minimal configuration (excluding framework middleware) |
| `web` | Gin, Echo, Fiber, Chi, Gorilla Mux, net/http, FastHTTP | Web frameworks only |
| `database` | database/sql, sqlx, GORM v1/v2 | Databases only |
| `external` | Redis, MongoDB, Kafka, gRPC, Kubernetes | External services only |
| `log` | log, logrus, zap | Log libraries only |
| `custom` | Directly specify with enabled_packages | User-defined |

### Preset combination

You can finely control by combining Presets and package options.

```
Final enabled packages = Preset packages + enabled_packages - disabled_packages
```

**Combination examples:**

- `preset: full` + `disabled_packages: ["grpc"]` → Full except gRPC
- `preset: web` + `enabled_packages: ["sql"]` → Web frameworks + SQL
- `preset: custom` + `enabled_packages: ["gin", "sql"]` → Only Gin and SQL

## Environment variables

You can also configure using environment variables instead of configuration files.

| Environment variable | Description | Value | Default |
|---|---|---|---|
| `GO_API_AST_DEBUG` | Enable debug output | `1` (enable), `0` (disable) | `0` |
| `GO_API_AST_OUTPUT_DIR` | Instrumented source output directory | Directory path | - |
| `WHATAP_INST_CONFIG` | Configuration file path | File path | `.whatap/config.yaml` |

### Configuration priority

Configuration values are applied in the following order.

```
CLI options > Environment variables > Configuration file > Default values
```

| Source | Example | Priority |
|---|---|---|
| CLI options | `--error-tracking` | 1 (Highest) |
| Environment variables | `GO_API_AST_DEBUG=1` | 2 |

| Source | Example | Priority |
|---|---|---|
| Configuration file | `error_tracking: true` | 3 |
| Default values | `false` | 4 (Lowest) |

## File exclusion patterns

Specify file patterns to exclude from instrumentation.

### Default exclusion patterns

If you don't specify `exclude` patterns in the configuration file, the following patterns are automatically applied:

| Pattern | Description |
|---|---|
| `**/*.pb.go` | protobuf generated files |
| `**/*.pb.gw.go` | grpc-gateway generated files |
| `**/*_grpc.pb.go` | grpc generated files |
| `**/*.connect.go` | connect-go generated files |
| `**/*_generated.go` | Auto-generated files |
| `**/*_gen.go` | Code generator output |
| `**/*_test.go` | Test files |
| `vendor/**` | vendor directory |
| `.git/**` | git directory |
| `node_modules/**` | node_modules directory |
| `whatap-instrumented/**` | Instrumented output directory |

**Exclusion reasons**

- **Generated code** (protobuf, grpc, etc.): May cause compilation errors
- **Test files**: Unnecessary for production monitoring
- **Dependency directories**: Third-party code

### Custom exclusion patterns

Specifying `exclude` patterns replaces the default values.

```yaml
# .whatap/config.yaml
exclude:
  - "**/*_test.go"
  - "vendor/**"
  - "internal/legacy/**"    # Exclude legacy code
  - "migrations/**"         # Exclude migration files
```

### Glob pattern syntax

| Pattern | Description | Match example |
|---|---|---|
| `*` | All characters in filename | `*.go` → `main.go` |
| `**` | Recursively all directories | `**/test/**` → `a/b/test/c/d.go` |
| `?` | Single character | `test?.go` → `test1.go` |
| `[abc]` | Character class | `test[12].go` → `test1.go` |

## Copy exclusion directories (copy_exclude)

In wrap mode ( `whatap-go-inst go build` ), specify directories to exclude when copying source files to a temporary directory.

### Default exclusion directories

The following directories are automatically excluded:

| Directory | Description |
|---|---|
| `.git` | Git repository |
| `.svn` | SVN repository |
| `.hg` | Mercurial repository |
| `node_modules` | Node.js dependencies |
| `vendor` | Go vendor directory |
| `.idea` | JetBrains IDE settings |
| `.vscode` | VS Code settings |
| `whatap-instrumented` | Instrumented source output |

> ⓘ `build` and `dist` directories are not excluded by default as they are often used as `go:embed` targets in Go projects.

### Custom exclusion directories

```yaml
# .whatap/config.yaml
copy_exclude:
  - "tmp"         # Temporary directory
  - "cache"       # Cache directory
  - "data"        # Large data directory
  - "testdata"    # Test data
```

> Custom `copy_exclude` items are **added** to the default list (not replaced).

### exclude vs copy_exclude differences

| Option | Purpose | Applied at |
|---|---|---|
| `exclude` | **File patterns** to exclude from instrumentation | During AST analysis |
| `copy_exclude` | **Directories** to exclude from copying | During wrap mode file copy |

- `exclude` : Exclude specific files using glob patterns like `_test.go` , `**/*.pb.go`
- `copy_exclude` : Exclude entire directories by directory name like `tmp` , `cache`

## Log collection

To enable log collection, configure as follows.

```yaml
# .whatap/config.yaml
instrumentation:
  preset: "full"  # Include log packages
```

```
# whatap.conf
logsink_enabled=true
```

> - When instrumented with whatap-go-inst, TraceLogWriter is automatically inserted
> - Transaction ID(@txid), multi-transaction ID(@mtid), etc. are automatically included in logs
> - TraceLogWriter method is recommended (transaction linkage possible)

# Configuration examples

## Enable all

```yaml
# .whatap/config.yaml
instrumentation:
  preset: "full"
```

All supported packages are enabled.

## Web and database only

```yaml
# .whatap/config.yaml
instrumentation:
  preset: "custom"
  enabled_packages:
    - "gin"
    - "echo"
    - "sql"
    - "gorm"
```

Only Gin, Echo, database/sql, and GORM are instrumented.

## Exclude specific packages

```yaml
# .whatap/config.yaml
instrumentation:
  preset: "full"
  disabled_packages:
    - "k8s"
    - "grpc"
```

All packages except Kubernetes and gRPC are instrumented.

## Enable error tracking

```
# .whatap/config.yaml
instrumentation:
  preset: "full"
  error_tracking: true
```

`trace.Error(ctx, err)` code is automatically inserted into `if err != nil` patterns.

```
// Before change
if err != nil {
    return err
}

// After change
if err != nil {
    trace.Error(ctx, err)  // Automatically added
    return err
}
```

## Environment-specific configuration files

```
# Development environment
WHATAP_INST_CONFIG=.whatap/dev-config.yaml whatap-go-inst go build ./...

# Production environment
WHATAP_INST_CONFIG=.whatap/prod-config.yaml whatap-go-inst go build ./...
```

```
# .whatap/dev-config.yaml
instrumentation:
  preset: "full"
  debug: true
  error_tracking: true
```

```
# .whatap/prod-config.yaml
instrumentation:
```

```
  preset: "full"
  error_tracking: true
  debug: false
```

## Debug mode

```
# .whatap/config.yaml
instrumentation:
  debug: true
  preset: "full"
```

Detailed debug information is output during build.

```
[whatap-go-inst] Config file: .whatap/config.yaml
[whatap-go-inst] Preset: full
[whatap-go-inst] Processing: main.go
[whatap-go-inst]   Added: trace.Init
...
```

## Instrumentation code output

```
# .whatap/config.yaml
instrumentation:
  output_dir: "./instrumented"
  preset: "full"
```

Instrumented source code is saved to the `./instrumented` directory.

**Use cases:**

- Review instrumentation results
- Analyze instrumentation code in CI/CD

# Troubleshooting

## When instrumentation is not applied

Check the following:

```
# Delete build cache and rebuild
go clean -cache
whatap-go-inst go build ./...

# Check detailed logs in debug mode
GO_API_AST_DEBUG=1 whatap-go-inst go build ./...
```

## When compilation errors occur

Auto-generated files like protobuf may be instrumented causing errors. Check the `exclude` patterns:

```
# .whatap/config.yaml
exclude:
  - "**/*.pb.go"
  - "**/*_generated.go"
```

## When wrap mode build is slow

Add large directories to `copy_exclude`:

```
# .whatap/config.yaml
copy_exclude:
  - "data"
  - "testdata"
  - "tmp"
```

# Complete list of supported packages

### Web frameworks

| Package name | Library | Inserted code |
|---|---|---|
| gin | github.com/gin-gonic/gin | whatapgin.Middleware() |
| echo | github.com/labstack/echo/v4 | whatapecho.Middleware() |
| fiber | github.com/gofiber/fiber/v2 | whatapfiber.Middleware() |
| chi | github.com/go-chi/chi/v5 | whatapchi.Middleware |
| gorilla | github.com/gorilla/mux | whatapmux.Middleware |
| nethttp | net/http | whataphttp.Func(), whataphttp.Handler() |
| fasthttp | github.com/valyala/fasthttp | whatapfasthttp.Middleware() |

### Databases

| Package name | Library | Inserted code |
|---|---|---|
| sql | database/sql | whatapsql.Open() |
| sqlx | github.com/jmoiron/sqlx | whatapsqlx.Open() |
| gorm | gorm.io/gorm | whatapgorm.Open() |
| jinzhugorm | github.com/jinzhu/gorm | whatapgorm.Open() |

### External services

| Package name | Library | Inserted code |
| --- | --- | --- |
| redigo | github.com/gomodule/redigo | whatapredigo.Dial() |
| goredis | github.com/redis/go-redis/v9 | whatapgoredis.NewClient() |
| mongo | go.mongodb.org/mongo-driver | whatapmongo.Connect() |
| sarama | github.com/IBM/sarama | Kafka Interceptor |
| grpc | google.golang.org/grpc | Server/Client Interceptor |
| k8s | k8s.io/client-go | config.Wrap() |

**Log libraries**

| Package name | Library | Inserted code |
| --- | --- | --- |
| log | log | log.SetOutput(logsink.GetTraceLogWriter()) |
| logrus | github.com/sirupsen/logrus | logrus.SetOutput(logsink.GetTraceLogWriter()) |
| zap | go.uber.org/zap | logsink.HookStderr() |

# Troubleshooting

This guide provides troubleshooting steps for issues that may occur when using whatap-go-inst.

## When instrumentation is not applied

### Step 1: Clear build cache

```
# Clear build cache
go clean -cache

# Rebuild
whatap-go-inst go build ./...
```

### Step 2: Check debug mode

```
# Check instrumentation process with debug output
GO_API_AST_DEBUG=1 whatap-go-inst go build ./...
```

## Common problem solving

### Specific files not instrumented

**Cause:**

- File is included in exclusion pattern
- Generated code files ( `*.pb.go` , `*_generated.go` , etc.)

**How to check:**

```
# Check skipped files
```

```
GO_API_AST_DEBUG=1 whatap-go-inst go build ./... 2>&1 | grep "SKIP"
```

**Solution:**

Adjust exclusion patterns or remove exclusion for specific files. For more details, see Configuration guide - Exclusion patterns.

## Build succeeds but no data visible

### 1. Check agent execution

```
ps -ef | grep whatap_agent
```

**If not present:**

```
# Start agent
/usr/whatap/agent/whatap_agent -d
```

### 2. Check main() initialization

```
# Check if trace.Init is in main() function
GO_API_AST_DEBUG=1 whatap-go-inst go build ./... 2>&1 | grep "trace.Init"
```

### 3. Check whatap.conf configuration

```
cat /usr/whatap/agent/whatap.conf
```

**Required configuration:**

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

## No instrumentation in Docker environment

**Problem:** whatap-go-inst not installed in multi-stage build

**Solution:**

```
FROM golang:1.21 AS builder

# Install whatap-go-inst
RUN go install github.com/whatap/go-api-inst/cmd/whatap-go-inst@latest

# Initialize and build
COPY . .
RUN whatap-go-inst go build -o /app/myapp .
```

For more details, see Docker installation guide.

# Execution mode comparison

whatap-go-inst provides multiple execution modes. Choose the mode that fits your situation.

## Mode comparison table

| Mode | Command | Original changes | Auto dependency | Recommended use |
|---|---|---|---|---|
| **default wrap mode** | `whatap-go-inst go build` | ❌ | ✅ | Recommended |
| **inject mode** | `whatap-go-inst inject` | Separate output | ❌ | Code review/CI |

## default wrap mode (recommended)

```
# Build directly without init
whatap-go-inst go --wrap build ./...
```

**Features:**

- ✅ No changes to original files at all
- ✅ Modified files are saved to ./whatap-instrumented. Easy error analysis.

**Use cases:**

- Selective instrumentation in CI/CD

### inject mode

```
# Output instrumented code to separate directory
whatap-go-inst inject -s ./src -o ./instrumented

# Build with output code
cd instrumented
go build ./...
```

**Features:**

- ✅ Save instrumented code separately
- ✅ Can compare original and instrumented code
- ⚠ Requires manual dependency installation
- ⚠ Requires inject execution before build

**Use cases:**

- Review instrumentation results
- Pre-instrumentation in CI/CD pipeline
- Check diff between original and transformed code

## Migrating from go-api

How to migrate a project that directly uses go-api to whatap-go-inst.

### Step 1: Remove existing whatap code

```
# Remove only automatic insertion patterns
whatap-go-inst remove -s . -o ./cleaned
```

```
# Also attempt to remove manual insertion patterns
whatap-go-inst remove --all -s . -o ./cleaned
```

## Step 2: Check changes

```
# Check differences with diff
diff -r . ./cleaned

# Or git diff
cp -r ./cleaned/* .
git diff
```

## Step 3: Restore custom code

Manually restore patterns not removed by `remove --all` or necessary custom code.

```
// Example: Keep custom transaction code
ctx, _ := trace.Start(context.Background(), "CustomTx")
defer trace.End(ctx, nil)
```

## Step 4: Build with whatap-go-inst

```
whatap-go-inst go build ./...
```

> **Patterns not automatically removed:**
>
> - Variable assignment: `ctx := trace.Start(...)`
> - Closure pattern: `whatapsql.Wrap(...)`
> - Complex custom instrumentation
>
> These patterns will show warning messages and must be removed manually.

# Additional support

If not resolved by the above methods:

1. **Collect debug logs**

```
GO_API_AST_DEBUG=1 whatap-go-inst go build ./... > debug.log 2>&1
```

2. **Collect environment information**

```
go version
whatap-go-inst version
cat .whatap/config.yaml
```

3. **Contact WhaTap Support Center**
    - Debug logs
    - Environment information
    - Reproducible minimal example

> If it cannot be handled with basic instrumentation, see Custom instrumentation.

# Custom Instrumentation

> This document is for **advanced users**. In most cases, Basic usage and Configuration guide are sufficient.

When the default instrumentation of whatap-go-inst is insufficient, you can add custom instrumentation rules.

## 5 instrumentation methods

| Method | Description | Main use cases |
|---|---|---|
| add | Create new file/function | Add helper functions |
| inject | Insert code inside function | Add trace to all functions |
| replace | Replace function call | `sql.Open` → `whatapsql.Open` |
| hook | Code before/after function call | Instrument legacy code |
| transform | Pattern transformation (advanced) | Complex transformations |

**Execution order:** `add → inject → replace → hook → transform`

**Basic configuration structure**

```yaml
# .whatap/config.yaml
custom:
  add: []       # Create new file/function
  inject: []    # Insert code inside function definition
  replace: []   # Replace function call
  hook: []      # Insert code before/after function call
  transform: [] # Code pattern -> template transformation
```

### When to use?

Custom instrumentation is needed in the following cases:

- Add monitoring to **internal common libraries**
- Instrument **unsupported frameworks**
- Add monitoring to **legacy code** without direct modification
- Selectively instrument **specific functions only**

# add - Create new file/function

Create helper functions or wrapper files. Can be used later in `replace`.

## Basic usage

```
custom:
  add:
    - package: "myapp/helper"          # Target package
      file: "whatap_helper.go"         # File to create
      content: |
        package helper

        import "github.com/whatap/go-api/trace"

        func WrapQuery(ctx context.Context, query string) string {
            ctx = trace.StartMethod(ctx, "WrapQuery")
            defer trace.EndMethod(ctx, nil)
            return query
        }
```

## Transformation result

```
// Created file: myapp/helper/whatap_helper.go
package helper
```

```
import "github.com/whatap/go-api/trace"

func WrapQuery(ctx context.Context, query string) string {
    ctx = trace.StartMethod(ctx, "WrapQuery")
    defer trace.EndMethod(ctx, nil)
    return query
}
```

More examples

## Using template file

```
custom:
  add:
    - package: "myapp/db"
      file: "whatap_db.go"
      content_file: "templates/db-helper.go.tmpl"  # External template file
```

## Add function to existing file

```
custom:
  add:
    - package: "myapp/service"
      file: "service.go"                # Existing file
      append: true                      # Append to end of file
      content: |
        func traceMethod(ctx context.Context, name string) (context.Context, func()) {
            return trace.StartMethod(ctx, name), func() { trace.EndMethod(ctx, nil) }
        }
```

## add + replace combination

```
custom:
  # Step 1: Create helper function
```

```
add:
  - package: "myapp/db"
    file: "whatap_wrapper.go"
    content: |
      package db

      func TracedQuery(ctx context.Context, sql string) (*Result, error) {
          ctx = trace.StartMethod(ctx, "db.query")
          defer trace.EndMethod(ctx, nil)
          return OriginalQuery(sql)
      }

# Step 2: Replace original function call with helper
replace:
  - package: "myapp/db"
    function: "Query"
    with: "TracedQuery"
```

# inject - Insert code inside function definition

Insert code at the start/end of function body.

- Can only target user-defined functions in the current module
- Cannot target Go standard library or external package functions

## Basic usage

```
custom:
  inject:
    - package: "myapp/service"      # Go import path
      function: "*"                 # Function name (* = all)
      start: |
        ctx = trace.StartMethod(ctx)
```

```
    defer trace.EndMethod(ctx, err)
  imports:
    - "github.com/whatap/go-api/trace"
```

## Transformation result

```go
// Before
func ProcessOrder(ctx context.Context) error {
    // Business logic
}

// After
func ProcessOrder(ctx context.Context) error {
    ctx = trace.StartMethod(ctx)          // <- start inserted
    defer trace.EndMethod(ctx, err)       // <- start inserted
    // Business logic
}
```

If there are returns in the middle of the function, use the defer pattern in `start` instead of `end`.

More examples

### Target specific function only

```
custom:
  inject:
    - package: "myapp/service"
      function: "ProcessOrder"       # Specific function name
      start: |
        ctx = trace.StartMethod(ctx, "ProcessOrder")
        defer trace.EndMethod(ctx, nil)
      imports:
        - "github.com/whatap/go-api/trace"
```

# replace - Replace function call

Replace function calls with another function.

## Basic usage

```yaml
custom:
  replace:
    - package: "database/sql"
      function: "Open"
      with: "whatapsql.Open"
      imports:
        - "github.com/whatap/go-api/instrumentation/database/sql/whatapsql"
```

## Transformation result

```go
// Before
db, err := sql.Open("mysql", dsn)

// After
db, err := whatapsql.Open("mysql", dsn)
```

More examples

### Replace multiple functions

```yaml
custom:
  replace:
    - package: "database/sql"
      function: "Open"
      with: "whatapsql.Open"
      imports:
        - "github.com/whatap/go-api/instrumentation/database/sql/whatapsql"
```

```yaml
- package: "github.com/jmoiron/sqlx"
  function: "Connect"
  with: "whatapsqlx.Connect"
  imports:
    - "github.com/whatap/go-api/instrumentation/github.com/jmoiron/sqlx/whatapsqlx"
```

# hook - Insert code before and after function call

Insert code before and after function calls.

> You can use only `before` or only `after`.

## Basic usage

```yaml
custom:
  hook:
    - package: "mycompany/mydb"
      function: "Query"
      before: "ctx, span := trace.Start(ctx, \"db.query\")"
      after: "span.End()"
      imports:
        - "github.com/whatap/go-api/trace"
```

## Transformation result

```go
// Before
result, err := mydb.Query(sql)

// After
ctx, span := trace.Start(ctx, "db.query")  // <- before inserted
result, err := mydb.Query(sql)
```

```
span.End()                    // <- after inserted
```

More examples

### Using before only

```yaml
custom:
  hook:
    - package: "mycompany/cache"
      function: "Get"
      before: "log.Printf(\"cache.Get called: %s\", key)"
      imports:
        - "log"
```

# transform - Transform code pattern

Transform complex code patterns with templates. The most flexible method.

## Basic usage: Add middleware

```yaml
custom:
  transform:
    - package: "github.com/gin-gonic/gin"
      function: "Default"
      template: |
        {{.Original}}
        {{.Var}}.Use(whatapgin.Middleware())
      imports:
        - "github.com/whatap/go-api/instrumentation/github.com/gin-gonic/gin/whatapgin"
```

**Transformation result:**

```
// Before
r := gin.Default()
```

```
// After
r := gin.Default()
r.Use(whatapgin.Middleware())
```

## Advanced usage: Wrap with closure

```
custom:
  transform:
    - package: "github.com/aerospike/aerospike-client-go"
      function: "NewClient"
      template: |
        func() (*as.Client, error) {
            ctx, done := whatapsql.Start(context.Background(), "aerospike")
            defer done()
            return {{.Original}}
        }()
      imports:
        - "context"
        - "github.com/whatap/go-api/instrumentation/database/sql/whatapsql"
```

**Transformation result:**

```
// Before
client, err := as.NewClient(policy, hosts...)

// After
client, err := func() (*as.Client, error) {
    ctx, done := whatapsql.Start(context.Background(), "aerospike")
    defer done()
    return as.NewClient(policy, hosts...)
}()
```

More examples

### HTTP client wrapping

```yaml
custom:
  transform:
    - package: "net/http"
      function: "Get"
      template: |
        func() (*http.Response, error) {
            ctx, done := httpc.Start(context.Background(), {{.Arg0}})
            defer done()
            return {{.Original}}
        }()
      imports:
        - "context"
        - "github.com/whatap/go-api/httpc"
```

# Template variables

Template variables available in `transform`.

| Variable | Description | Example |
|---|---|---|
| `{{.Original}}` | Matched original code | `gin.Default()` |
| `{{.Var}}` | Assigned variable name | `r` (from r := ...) |
| `{{.Args}}` | All function arguments | `policy, hosts...` |
| `{{.Arg0}}`, `{{.Arg1}}` | Individual arguments | First, second argument |
| `{{.FuncName}}` | Function name | `Default` |
| `{{.PkgName}}` | Package name | `gin` |

### Usage example

```
custom:
  transform:
    - package: "mycompany/db"
      function: "Query"
      template: |
        func() (*Result, error) {
            ctx := trace.StartMethod(context.Background(), "{{.PkgName}}.{{.FuncName}}")
            defer trace.EndMethod(ctx, nil)
            return {{.Original}}
        }()
```

**Transformation result:**

```
// Before
result, err := db.Query("SELECT * FROM users")

// After
result, err := func() (*Result, error) {
    ctx := trace.StartMethod(context.Background(), "db.Query")
    defer trace.EndMethod(ctx, nil)
    return db.Query("SELECT * FROM users")
}()
```

## Practical examples

### Example 1: Instrument internal library

Add monitoring to internal common libraries.

```
# .whatap/config.yaml
custom:
  # Add tracing to all service functions
  inject:
    - package: "mycompany/service"
```

```
    function: "*"
    start: |
      ctx, span := trace.Start(ctx, "service")
      defer span.End()
    imports:
      - "github.com/whatap/go-api/trace"

  # Replace internal DB library
  replace:
    - package: "mycompany/db"
      function: "Connect"
      with: "whatapsql.Open"
      imports:
        - "github.com/whatap/go-api/instrumentation/database/sql/whatapsql"
```

## Example 2: Instrument legacy code

Add monitoring to legacy code that's difficult to modify directly.

```
custom:
  # Add tracing to legacy HTTP client calls
  hook:
    - package: "legacy/httpclient"
      function: "Do"
      before: |
        ctx, span := httpc.Start(ctx, req.URL.String())
      after: |
        span.End()
      imports:
        - "github.com/whatap/go-api/httpc"

  # Log legacy cache calls
  hook:
    - package: "legacy/cache"
      function: "Get"
      before: |
        startTime := time.Now()
      after: |
        trace.Step(ctx, "cache.Get", time.Since(startTime).Milliseconds(), nil)
```

```
    imports:
      - "time"
      - "github.com/whatap/go-api/trace"
```

## Example 3: Custom middleware

Automatically add custom middleware.

```
custom:
  # Create helper middleware
  add:
    - package: "myapp/middleware"
      file: "whatap_middleware.go"
      content: |
        package middleware

        import (
            "github.com/gin-gonic/gin"
            "github.com/whatap/go-api/trace"
        )

        func CustomTracing() gin.HandlerFunc {
            return func(c *gin.Context) {
                ctx := trace.Start(c.Request.Context(), c.Request.URL.Path)
                c.Request = c.Request.WithContext(ctx)
                defer trace.End(ctx, nil)
                c.Next()
            }
        }

  # Add custom middleware to Gin router
  transform:
    - package: "github.com/gin-gonic/gin"
      function: "Default"
      template: |
        {{.Original}}
        {{.Var}}.Use(middleware.CustomTracing())
      imports:
        - "myapp/middleware"
```

## Example 4: Conditional instrumentation

Activate instrumentation only under specific conditions.

```
custom:
  transform:
    - package: "mycompany/db"
      function: "Query"
      template: |
        func() (*Result, error) {
            if os.Getenv("WHATAP_ENABLED") == "true" {
                ctx, done := whatapsql.Start(context.Background(), "db.query")
                defer done()
            }
            return {{.Original}}
        }()
      imports:
        - "os"
        - "context"
        - "github.com/whatap/go-api/instrumentation/database/sql/whatapsql"
```

# Multi-module Projects

This document explains how to use whatap-go-inst in projects composed of multiple Go modules.

## Terminology

| Term | Description | Example |
|---|---|---|
| **Module** | Unit with go.mod | `module mycompany/user-api` |
| **Package** | Directory-level code group | `mycompany/user-api/pkg/auth` |
| **External module** | Dependencies obtained via go get | `github.com/gin-gonic/gin` |
| **Local module** | Separate module within the same project | `replace ../shared-lib` |

## Processing by package type

whatap-go-inst determines whether to instrument based on package location.

### toolexec skip rules

```
// 1. Go standard library → Skip
if strings.HasPrefix(path, os.Getenv("GOROOT")) {
    return true  // No transformation
}

// 2. External packages (go get) → Skip
if strings.HasPrefix(path, os.Getenv("GOMODCACHE")) {
    return true  // No transformation
}
```

```
// 3. Others (my code) → Transform
return false
```

## Summary

| Package type | Path | Instrumentation |
|---|---|---|
| Go standard library | `$GOROOT/src/...` | Skip |
| External modules (go get) | `$GOMODCACHE/...` | Skip |
| My project code | Local path | **Instrumented** |
| replace local modules | Local path (../) | **Instrumented** (fast mode only) |

# Multi-module scenarios

## Scenario: 3 separate modules

```
C:/projects/
├── db-lib/            # Module A: DB logic library
│   ├── go.mod         # module mycompany/db-lib
│   ├── connection.go  # Uses sql.Open()
│   └── query.go
│
├── web-lib/           # Module B: Web server library
│   ├── go.mod         # module mycompany/web-lib
│   ├── server.go      # Uses gin.Default()
│   └── handler.go
│
└── main-app/          # Module C: Main app
    ├── go.mod         # module mycompany/main-app
    └── main.go        # Imports A, B
```

### Module C's go.mod

```
module mycompany/main-app

go 1.21

require (
    mycompany/db-lib v1.0.0
    mycompany/web-lib v1.0.0
)
```

# Recommended approaches

## Approach 1: Separate inject per module (for deployment)

Recommended for production environment deployment.

**Step 1: Inject each module**

```
# Instrument db-lib
cd db-lib
whatap-go-inst inject -s . -o ../db-lib-instrumented

# Instrument web-lib
cd web-lib
whatap-go-inst inject -s . -o ../web-lib-instrumented

# Instrument main-app
cd main-app
whatap-go-inst inject -s . -o ../main-app-instrumented
```

**Step 2: Replace with instrumented versions**

```
// main-app-instrumented/go.mod
replace mycompany/db-lib => ../db-lib-instrumented
replace mycompany/web-lib => ../web-lib-instrumented
```

**Step 3: Build**

```
cd main-app-instrumented
go build ./...
```

## Approach 2: Monorepo structure (new projects)

For new projects, a single module structure is simplest.

```
myproject/
├── go.mod            # Single module mycompany/myproject
├── cmd/
│   └── main/
│       └── main.go     # main package
├── internal/
│   ├── db/
│   │   └── connection.go
│   └── web/
│       └── server.go
└── pkg/
    └── shared/
        └── utils.go
```

```
# Build
whatap-go-inst go build ./cmd/main
```

# Behavior comparison by mode

## --wrap mode

| Item | --wrap mode |
|---|---|
| **main module** | Instrumented |
| **replace modules** | Not instrumented |

| Item | --wrap mode |
|---|---|
| **External modules (GOMODCACHE)** | Skip |
| **Prerequisites** | None |

### replace handling

```
// --wrap mode only adjusts replace paths
replace mycompany/db-lib => ../db-lib

// After copying to temporary directory:
replace mycompany/db-lib => /original/path/to/db-lib  // References original!
```

**Result:** replace target modules are not copied and reference the original (non-instrumented) code

**To instrument replace modules:** You need to inject each module separately.

```
# Inject each module separately
whatap-go-inst inject -s ../db-lib -o ../db-lib-instrumented
whatap-go-inst inject -s . -o ./instrumented

# Modify replace path in instrumented/go.mod
# replace mycompany/db-lib => ../db-lib-instrumented
```

## Considerations

### 1. Injecting libraries without main

Libraries without a main function can also be injected.

| Item | Processing |
|---|---|
| `trace.Init()` | Not added (normal) |

| Item | Processing |
| --- | --- |
| `sql.Open()` → `whatapsql.Open()` | Transformed |
| `gin.Default()` + middleware | Added |

```
# Inject library module
cd db-lib
whatap-go-inst inject -s . -o ./instrumented

# Result: Transformed to whatapsql.Open()
# trace.Init() not added (should only be called in main)
```

### 2. Dependency propagation

Modules using instrumented libraries also need whatap dependencies.

```
// db-lib-instrumented/connection.go
import "github.com/whatap/go-api/instrumentation/.../whatapsql"

// This dependency is needed to build in main-app
```

```
cd main-app
go get github.com/whatap/go-api@latest
```

## FAQ

### Q1: Are replace modules also instrumented?

**A:** No, wrap mode does **not instrument** replace modules. It only adjusts replace paths and references the original. To instrument replace modules, you need to inject each module separately.

**Inject each module separately**

```
whatap-go-inst inject -s ../db-lib -o ../db-lib-inst
whatap-go-inst inject -s ../web-lib -o ../web-lib-inst
```

## Q2: Can external libraries (gin, gorm, etc.) be instrumented?

**A:** No. External libraries are skipped because they are in `$GOMODCACHE`. This is intentional behavior.

- Modifying external libraries affects other projects
- Principle of immutability in Go module system

Instead, **calls to external libraries in user code** are transformed:

- `sql.Open()` → `whatapsql.Open()`
- `gin.Default()` → Middleware automatically added

## Q3: Monorepo vs multi-module, which is better?

| Structure | Advantages | Disadvantages |
|---|---|---|
| **Monorepo** | Simple instrumentation, easy dependency management | Difficult module reuse |
| **Multi-module** | Independent module deployment possible | Complex instrumentation setup |

**Recommendations:**

- New projects → Monorepo
- Existing multi-module → Separate inject per module

## Q4: How do I build multi-module in CI/CD?

See the CI/CD examples section below.

# CI/CD examples

## GitHub Actions

```yaml
# .github/workflows/build.yml
name: Build with WhaTap Instrumentation

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      # 1. Checkout all modules
      - uses: actions/checkout@v3
        with:
          path: main-app
      - uses: actions/checkout@v3
        with:
          repository: mycompany/db-lib
          path: db-lib
      - uses: actions/checkout@v3
        with:
          repository: mycompany/web-lib
          path: web-lib

      # 2. Install Go
      - uses: actions/setup-go@v4
        with:
          go-version: '1.21'

      # 3. Install whatap-go-inst
      - run: go install github.com/whatap/go-api-inst/cmd/whatap-go-inst@latest

      # 4. Build
      - run: |
          cd main-app
          whatap-go-inst go build -o myapp ./...
```

```
    # 5. Upload artifact
    - uses: actions/upload-artifact@v3
      with:
        name: myapp
        path: main-app/myapp
```

## GitLab CI

```
# .gitlab-ci.yml
stages:
  - build

build:
  stage: build
  image: golang:1.21
  before_script:
    # Install whatap-go-inst
    - go install github.com/whatap/go-api-inst/cmd/whatap-go-inst@latest
    # Clone other modules
    - git clone https://gitlab.com/mycompany/db-lib.git ../db-lib
    - git clone https://gitlab.com/mycompany/web-lib.git ../web-lib
  script:
    - whatap-go-inst go build -o myapp ./...
  artifacts:
    paths:
      - myapp
```

## Docker multi-stage build

```
# Dockerfile
FROM golang:1.21 AS builder

WORKDIR /workspace

# Install whatap-go-inst
RUN go install github.com/whatap/go-api-inst/cmd/whatap-go-inst@latest
```

```
# Copy source (all modules)
COPY db-lib/ ./db-lib/
COPY web-lib/ ./web-lib/
COPY main-app/ ./main-app/

# Initialize and add dependencies
WORKDIR /workspace/main-app

# Build
RUN whatap-go-inst go build -o /app/myapp ./...

# Runtime image
FROM alpine:latest
COPY --from=builder /app/myapp /app/myapp
CMD ["/app/myapp"]
```

# Manual Instrumentation Guide

You can manually send monitoring data to the WhaTap agent using the Go library. Example code is available at github.com/whatap/go-api-example.

> ⓘ **Automatic instrumentation vs Manual instrumentation**
>
> - **Automatic instrumentation (Recommended)**: Refer to Automatic instrumentation guide for automatic code insertion
> - **Manual instrumentation**: Add the APIs described in this document directly to your code

## Getting started

Install the Go API with the following command:

```
go get github.com/whatap/go-api@latest
```

### Init, Shutdown

Basic structure for initializing and shutting down the monitoring module:

```
import "github.com/whatap/go-api/trace"

func main(){
    trace.Init(nil)
    defer trace.Shutdown()
    ...
}
```

**Init function configuration options:**

- Can be configured in `map[string]string` format
- Can also be configured in `whatap.conf` file

- Default TCP connection: 127.0.0.1:6600

```
m := make(map[string]string)
m["net_ipc_host"] = "127.0.0.1"
m["net_ipc_port"] = "6601"
trace.Init(m)
```

**whatap.conf configuration:**

```
accesskey={access key}
whatap.server.host={collection server IP address}
net_ipc_host=127.0.0.1
net_ipc_port=6600
```

## Context management

The agent distinguishes transactions based on `whatap context(trace.TraceCtx)`. Performance information outside transactions is ignored or only statistics are collected.

### Creating transactions

```
var traceCtx *TraceCtx
traceCtx.Txid = keygen.Next()
ctx = context.WithValue(ctx, "whatap", traceCtx)
```

# Transaction tracing

## Web transaction tracing

```
http.HandleFunc("/index", func(w http.ResponseWriter, r *http.Request) {
    ctx, _ := trace.StartWithRequest(r)
    defer trace.End(ctx, nil)
})
```

**Wrapping functions:**

- `trace.Func()` : Sets RequestURI as transaction name
- `trace.HandlerFunc()` : Provides same functionality

### General transaction tracing

```
func main() {
    ctx := context.Background()
    ctx, _ := trace.Start(ctx, "Custom Transaction")
    ...
    trace.End(ctx, nil)
}
```

### Transaction API

```
func Start(ctx context.Context, name string) (context.Context, error)
func End(ctx context.Context, err error) error
func StartWithRequest(r *http.Request) (context.Context, error)
func Step(ctx context.Context, title, message string, elapsed, value int) error
func HandlerFunc(handler func(http.ResponseWriter, *http.Request)) http.HandlerFunc
func Func(handler func(http.ResponseWriter, *http.Request)) func(http.ResponseWriter, *http.Request)
```

## DB connection and SQL tracing

Limitations:

- SQL statement: Maximum 32KB
- Prepared parameters: Maximum 20, 256 bytes each

### DB Connection tracing

```
import whatapsql "github.com/whatap/go-api/sql"

ctx, _ := trace.Start(context.Background(), "Trace Open DB")
defer trace.End(ctx, nil)
```

```
sqlCtx, _ := whatapsql.StartOpen(ctx, "id@tcp(x.x.x.x:3306)/test")
db, err := sql.Open("mysql", "id:pwd@tcp(x.x.x.x:3306)/test")
whatapsql.End(sqlCtx, err)
defer db.Close()
```

## SQL Query tracing

```
query = "select id, subject from tbl_faq limit 10"
sqlCtx, _ = whatapsql.Start(ctx, "id:pwd@tcp(x.x.x.x:3306)/test", query)
rows, err := db.QueryContext(ctx, query)
whatapsql.End(sqlCtx, err)
```

## Prepared Statement tracing

```
query = "select id, subject from tbl_faq where id = ? limit ?"
stmt, err := db.Prepare(query)
defer stmt.Close()

params := make([]interface{}, 0)
params = append(params, 8)
params = append(params, 1)

sqlCtx, _ := whatapsql.StartWithParamArray(ctx, "id:pwd@tcp(x.x.x.x:3306)/test", query, params)
rows, err := stmt.QueryContext(ctx, params...)
whatapsql.End(sqlCtx, err)
```

## Using database/sql package

```
import (
   _ "github.com/go-sql-driver/mysql"
   "github.com/whatap/go-api/instrumentation/database/sql/whatapsql"
)

db, err := whatapsql.OpenContext(ctx, "mysql", dataSource)
defer db.Close()
```

```
if rows, err := db.QueryContext(ctx, query); err == nil {
    ...
}
```

### SQL API

```
func Start(ctx context.Context, dbhost, sql string) (*SqlCtx, error)
func StartOpen(ctx context.Context, dbhost string) (*SqlCtx, error)
func End(sqlCtx *SqlCtx, err error) error
func StartWithParam(ctx context.Context, dbhost, sql, param ...interface{}) (*SqlCtx, error)
func StartWithParamArray(ctx context.Context, dbhost, sql string, param []interface{}) (*SqlCtx, error)
func Trace(ctx context.Context, dbhost, sql, param string, elapsed int, err error) error
```

## HTTP request tracing

```
import "github.com/whatap/go-api/httc"

ctx, _ := trace.Start(context.Background(), "Trace Http Call")
defer trace.End(ctx, nil)

httpcCtx, _ := httpc.Start(ctx, callUrl)
resp, err := http.Get(callUrl)
if err == nil {
    httpc.End(httpcCtx, resp.StatusCode, "", nil)
} else {
    httpc.End(httpcCtx, 0, "", err)
}
```

### HTTP Transport RoundTrip

```
import "github.com/whatap/go-api/instrumentation/net/http/whataphttp"

ctx, _ := trace.Start(context.Background(), "Http call")
defer trace.End(ctx, nil)
```

```
client := http.DefaultClient
client.Transport = whataphttp.NewRoundTrip(ctx, http.DefaultTransport)
resp, err := client.Get(callUrl)
defer resp.Body.Close()
```

### HTTP API

```
func Start(ctx context.Context, url string) (*HttpcCtx, error)
func End(httpcCtx *HttpcCtx, status int, reason string, err error) error
func Trace(ctx context.Context, host string, port int, url string, elapsed int, status int, reason string, err error) error
```

## Multi-transaction tracing (Distributed tracing)

Trace transactions associated with other agents or projects.

**Header key values:**

- `x-wtap-po`
- `x-wtap-mst`
- `x-wtap-sp1`

Also supports OpenTrace's `traceparent` header.

### Agent configuration

```
mtrace_enabled=true
mtrace_rate=100
```

### Request Header processing

```
func UpdateMtrace(traceCtx *trace.TraceCtx, header http.Header)
```

`trace.StartWithRequest` internally calls this function.

```
ctx, traceCtx := trace.GetTraceContext(ctx)
if traceCtx != nil {
    trace.UpdateMtrace(traceCtx, header)
}
```

## Retrieving and adding Header information

```
func GetMTrace(ctx context.Context) http.Header
```

This function returns headers needed for distributed tracing:

```
headers := trace.GetMTrace(wCtx)
for key, _ := range headers {
    req.Header.Set(key, headers.Get(key))
}
```

## Automatic Header addition

Already included in WhaTap transport:

```
client := http.Client{Timeout: timeout}
client.Transport = whataphttp.NewRoundTrip(ctx, http.DefaultTransport)
resp, err := client.Get(callUrl)
```

**Using multiple Transports:**

```
client.Transport = NewAccessLogRoundTrip(whataphttp.NewRoundTrip(ctx, http.DefaultTransport))
```

# Function tracing

Measure execution time of user functions or specific sections:

```
import "github.com/whatap/go-api/method"

ctx, _ := trace.Start(context.Background(), "Trace Method")
```

```
defer trace.End(ctx, nil)

getUser(ctx)

func getUser(ctx context.Context) {
    methodCtx, _ := method.Start(ctx, "getUser")
    defer method.End(methodCtx, nil)
    time.Sleep(time.Duration(1) * time.Second)
}
```

### Function tracing API

```
func Start(ctx context.Context, name string) (*MethodCtx, error)
func End(methodCtx *MethodCtx, err error) error
func Trace(ctx context.Context, name string, elapsed int, err error) error
```

## Wrap functions (Go 1.18+ Generics)

`Wrap` functions allow you to write instrumentation code more concisely using closures. Utilizes Go 1.18+ Generics.

### Trace Wrap functions (Generic)

Used for general-purpose tracing not belonging to a specific domain.

```
import "github.com/whatap/go-api/trace"

// Wrap - Generic function returning (T, error)
result, err := trace.Wrap(ctx, "ThirdParty.Calculate", func() (int, error) {
    return thirdPartyLib.Calculate(input)
})

// WrapError - Returns only error
err := trace.WrapError(ctx, "FileProcessor.Process", func() error {
    return processFile(path)
})
```

```
// WrapVoid - No return value
trace.WrapVoid(ctx, "Cleanup.Execute", func() {
    cleanup()
})
```

## SQL Wrap functions

Useful for instrumenting DB libraries that don't support hooks (Aerospike, some NoSQL, etc.).

```
import "github.com/whatap/go-api/sql"

// Wrap - Query returning (T, error)
record, err := sql.Wrap(ctx, "aerospike://host:3000", "GET ns/set", func() (*aero.Record, error) {
    return client.Get(policy, key)
})

// WrapError - Query returning only error (INSERT, UPDATE, DELETE)
err := sql.WrapError(ctx, "aerospike://host:3000", "PUT ns/set", func() error {
    return client.Put(policy, key, bins)
})

// WrapP - Including parameter tracing
rows, err := sql.WrapP(ctx, "mysql://host:3306", "SELECT * FROM users WHERE id = ?",
    []interface{}{userId},
    func() (*sql.Rows, error) {
        return db.Query("SELECT * FROM users WHERE id = ?", userId)
    },
)

// WrapOpen - DB connection tracing
db, err := sql.WrapOpen(ctx, "mysql://host:3306", func() (*sql.DB, error) {
    return sql.Open("mysql", dsn)
})
```

## HTTP Wrap functions

Used for instrumenting external HTTP API calls.

```
import "github.com/whatap/go-api/httpc"

// Wrap - Returns (T, error)
resp, err := httpc.Wrap(ctx, "https://api.payment.com/charge", func() (*PaymentResp, error) {
    return paymentClient.Charge(amount)
})

// WrapError - Returns only error
err := httpc.WrapError(ctx, "https://api.example.com/webhook", func() error {
    return sendWebhook(payload)
})

// WrapWithStatus - Including HTTP status code
resp, status, err := httpc.WrapWithStatus(ctx, "https://api.example.com/data",
    func() (*Response, int, error) {
        resp, err := client.Get(url)
        if resp != nil {
            return resp, resp.StatusCode, err
        }
        return nil, 0, err
    },
)
```

## Method Wrap functions

Used for tracing custom methods/functions.

```
import "github.com/whatap/go-api/method"

// Wrap - Returns (T, error)
order, err := method.Wrap(ctx, "OrderService.ProcessOrder", func() (*Order, error) {
    return s.processOrderInternal(orderID)
})

// WrapError - Returns only error
err := method.WrapError(ctx, "OrderService.ValidateOrder", func() error {
    return s.validateOrder(order)
})
```

```
// WrapVoid - No return value
method.WrapVoid(ctx, "CacheService.Invalidate", func() {
    s.cache.Delete(key)
})
```

## Wrap function API summary

| Package | Function | Description |
|---|---|---|
| trace | Wrap[T] | Generic (T, error) return |
| trace | WrapError | Generic error only return |
| trace | WrapVoid | Generic no return value |
| sql | Wrap[T] | Query returning (T, error) |
| sql | WrapError | Query returning only error |
| sql | WrapP[T] | Including parameter tracing |
| sql | WrapErrorP | Parameters + error only |
| sql | WrapOpen[T] | DB connection tracing |
| httpc | Wrap[T] | HTTP call |
| httpc | WrapError | Returns only error |
| httpc | WrapWithStatus[T] | Including status code |
| method | Wrap[T] | Method tracing |
| method | WrapError | Returns only error |
| method | WrapVoid | No return value |

# Log collection

## Agent configuration

```
logsink_enabled=true            # Enable log collection (default: false)
logsink_trace_enabled=true      # Transaction linkage (default: true)
logsink_zip_enabled=true        # Compressed transmission (default: true)
```

## TraceLogWriter (Recommended)

Using `logsink.GetTraceLogWriter()` allows you to link logs with transactions. `@txid`, `@mtid`, `@gid` fields are automatically added to logs.

```
func GetTraceLogWriter(w io.Writer) io.Writer
func GetTraceLogWriterWithCategory(w io.Writer, category string) io.Writer
```

**Change notice**: It is recommended to use `GetTraceLogWriter(os.Stdout)`, `GetTraceLogWriter(os.Stderr)` instead of the existing `GetWriterHookStdout()`, `GetWriterHookStderr()`.

## Existing API (Compatibility maintained)

```
func GetWriterHookStdout() io.Writer  // → GetTraceLogWriter(os.Stdout) recommended
func GetWriterHookStderr() io.Writer  // → GetTraceLogWriter(os.Stderr) recommended
```

## log package example

```
import (
    "log"
    "os"
    "github.com/whatap/go-api/logsink"
    "github.com/whatap/go-api/trace"
)
```

```
func main() {
    trace.Init(nil)
    defer trace.Shutdown()

    // Recommended: Use TraceLogWriter
    log.SetOutput(logsink.GetTraceLogWriter(os.Stderr))

    // Existing method (compatibility maintained)
    // log.SetOutput(logsink.GetWriterHookStderr())

    log.Println("Application started")
}
```

## go.uber.org/zap example

```
import (
    "os"
    "github.com/whatap/go-api/logsink"
    "github.com/whatap/go-api/trace"
    "go.uber.org/zap"
    "go.uber.org/zap/zapcore"
)

func main() {
    trace.Init(nil)
    defer trace.Shutdown()

    // Wrap with TraceLogWriter
    writer := logsink.GetTraceLogWriter(os.Stdout)

    consoleCore := zapcore.NewCore(
        zapcore.NewConsoleEncoder(zap.NewDevelopmentEncoderConfig()),
        zapcore.AddSync(writer),
        zap.InfoLevel,
    )
    logger := zap.New(consoleCore)
    logger.Info("logger started")
}
```

### sirupsen/logrus example

```
import (
    "os"
    log "github.com/sirupsen/logrus"
    "github.com/whatap/go-api/logsink"
    "github.com/whatap/go-api/trace"
)

func main() {
    trace.Init(nil)
    defer trace.Shutdown()

    // Wrap with TraceLogWriter
    log.SetOutput(logsink.GetTraceLogWriter(os.Stderr))
    log.Info("Log message")
}
```

## Web framework instrumentation

Using the `instrumentation` package enables automatic tracing for major web frameworks.

### Gin

```
import (
    "github.com/gin-gonic/gin"
    "github.com/whatap/go-api/instrumentation/github.com/gin-gonic/gin/whatapgin"
)

func main() {
    trace.Init(nil)
    defer trace.Shutdown()

    r := gin.Default()
    r.Use(whatapgin.Middleware())
```

```
    r.GET("/hello", func(c *gin.Context) {
        c.JSON(200, gin.H{"message": "hello"})
    })
    r.Run(":8080")
}
```

## Echo v4

```
import (
    "github.com/labstack/echo/v4"
    "github.com/whatap/go-api/instrumentation/github.com/labstack/echo/v4/whatapecho"
)

func main() {
    trace.Init(nil)
    defer trace.Shutdown()

    e := echo.New()
    e.Use(whatapecho.Middleware())

    e.GET("/hello", func(c echo.Context) error {
        return c.JSON(200, map[string]string{"message": "hello"})
    })
    e.Start(":8080")
}
```

## Fiber v2

```
import (
    "github.com/gofiber/fiber/v2"
    "github.com/whatap/go-api/instrumentation/github.com/gofiber/fiber/v2/whatapfiber"
)

func main() {
    trace.Init(nil)
    defer trace.Shutdown()
```

```
    app := fiber.New()
    app.Use(whatapfiber.Middleware())

    app.Get("/hello", func(c *fiber.Ctx) error {
        return c.JSON(fiber.Map{"message": "hello"})
    })
    app.Listen(":8080")
}
```

## Chi v5

```
import (
    "net/http"
    "github.com/go-chi/chi/v5"
    "github.com/whatap/go-api/instrumentation/github.com/go-chi/chi/whatapchi"
)

func main() {
    trace.Init(nil)
    defer trace.Shutdown()

    r := chi.NewRouter()
    r.Use(whatapchi.Middleware)

    r.Get("/hello", func(w http.ResponseWriter, r *http.Request) {
        w.Write([]byte("hello"))
    })
    http.ListenAndServe(":8080", r)
}
```

## Gorilla Mux

```
import (
    "net/http"
    "github.com/gorilla/mux"
    "github.com/whatap/go-api/instrumentation/github.com/gorilla/mux/whatapmux"
)
```

```
func main() {
    trace.Init(nil)
    defer trace.Shutdown()

    r := mux.NewRouter()
    r.Use(whatapmux.Middleware)

    r.HandleFunc("/hello", func(w http.ResponseWriter, r *http.Request) {
        w.Write([]byte("hello"))
    })
    http.ListenAndServe(":8080", r)
}
```

## FastHTTP

```
import (
    "github.com/valyala/fasthttp"
    "github.com/whatap/go-api/instrumentation/github.com/valyala/fasthttp/whatapfasthttp"
)

func main() {
    trace.Init(nil)
    defer trace.Shutdown()

    handler := func(ctx *fasthttp.RequestCtx) {
        ctx.WriteString("hello")
    }

    fasthttp.ListenAndServe(":8080", whatapfasthttp.Middleware(handler))
}
```

## net/http handler wrapping

```
import (
    "net/http"
    "github.com/whatap/go-api/instrumentation/net/http/whataphttp"
```

```
)

func main() {
    trace.Init(nil)
    defer trace.Shutdown()

    // HandleFunc wrapping
    http.HandleFunc("/api", whataphttp.Func(func(w http.ResponseWriter, r *http.Request) {
        w.Write([]byte("hello"))
    }))

    // Handler wrapping
    http.Handle("/handler", whataphttp.Handler(myHandler))

    http.ListenAndServe(":8080", nil)
}
```

## database/sql instrumentation (whatapsql)

Wraps the `database/sql` package to automatically track SQL queries.

```
import (
    _ "github.com/go-sql-driver/mysql"
    "github.com/whatap/go-api/instrumentation/database/sql/whatapsql"
    "github.com/whatap/go-api/trace"
)

func main() {
    trace.Init(nil)
    defer trace.Shutdown()

    // Use whatapsql.Open instead of sql.Open
    db, err := whatapsql.Open("mysql", "user:password@tcp(localhost:3306)/dbname")
    if err != nil {
        panic(err)
    }
    defer db.Close()
```

```
    // db.Query, db.Exec, etc. are automatically tracked
    rows, err := db.QueryContext(ctx, "SELECT * FROM users WHERE id = ?", 1)
}
```

### OpenContext (Transaction context linkage)

```
// Link with transaction context
db, err := whatapsql.OpenContext(ctx, "mysql", dsn)
```

## sqlx instrumentation

```
import (
    "github.com/whatap/go-api/instrumentation/github.com/jmoiron/sqlx/whatapsqlx"
)

func main() {
    trace.Init(nil)
    defer trace.Shutdown()

    // Use whatapsqlx.Open instead of sqlx.Open
    db, err := whatapsqlx.Open("mysql", dsn)

    // Use whatapsqlx.Connect instead of sqlx.Connect
    db, err := whatapsqlx.Connect("postgres", dsn)
}
```

## GORM instrumentation

### gorm.io/gorm (GORM v2)

```
import (
    "gorm.io/driver/mysql"
    "github.com/whatap/go-api/instrumentation/github.com/go-gorm/gorm/whatapgorm"
```

```
)

func main() {
    trace.Init(nil)
    defer trace.Shutdown()

    // Use whatapgorm.Open instead of gorm.Open
    db, err := whatapgorm.Open(mysql.Open(dsn), &gorm.Config{})
}
```

### github.com/jinzhu/gorm (GORM v1)

```
import (
    "github.com/whatap/go-api/instrumentation/github.com/jinzhu/gorm/whatapgorm"
)

func main() {
    trace.Init(nil)
    defer trace.Shutdown()

    db, err := whatapgorm.Open("mysql", dsn)
}
```

## Redis instrumentation

### go-redis v9

```
import (
    "github.com/whatap/go-api/instrumentation/github.com/redis/go-redis/v9/whatapgoredis"
)

func main() {
    trace.Init(nil)
    defer trace.Shutdown()
```

```
    // Use whatapgoredis.NewClient instead of redis.NewClient
    rdb := whatapgoredis.NewClient(&redis.Options{
        Addr: "localhost:6379",
    })

    // redis.NewClusterClient
    cluster := whatapgoredis.NewClusterClient(&redis.ClusterOptions{
        Addrs: []string{"localhost:7000", "localhost:7001"},
    })
}
```

## go-redis v8

```
import (
    "github.com/whatap/go-api/instrumentation/github.com/go-redis/redis/v8/whatapgoredis"
)

// Same usage as v9
rdb := whatapgoredis.NewClient(&redis.Options{
    Addr: "localhost:6379",
})
```

## Redigo

```
import (
    "github.com/whatap/go-api/instrumentation/github.com/gomodule/redigo/whatapredigo"
)

func main() {
    trace.Init(nil)
    defer trace.Shutdown()

    // Use whatapredigo.Dial instead of redis.Dial
    conn, err := whatapredigo.Dial("tcp", "localhost:6379")
    defer conn.Close()

    // Use regular redis commands
```

```
    conn.Do("SET", "key", "value")
}
```

# MongoDB instrumentation

```
import (
    "go.mongodb.org/mongo-driver/mongo"
    "github.com/whatap/go-api/instrumentation/go.mongodb.org/mongo-driver/mongo/whatapmongo"
)

func main() {
    trace.Init(nil)
    defer trace.Shutdown()

    // Use whatapmongo.Connect instead of mongo.Connect
    client, err := whatapmongo.Connect(ctx, options.Client().ApplyURI("mongodb://localhost:27017"))
    defer client.Disconnect(ctx)

    collection := client.Database("test").Collection("users")
    collection.FindOne(ctx, bson.M{"name": "john"})
}
```

# gRPC instrumentation

## Server side

```
import (
    "google.golang.org/grpc"
    "github.com/whatap/go-api/instrumentation/google.golang.org/grpc/whatapgrpc"
)

func main() {
    trace.Init(nil)
    defer trace.Shutdown()
```

```
    // Add Interceptor
    server := grpc.NewServer(
        grpc.UnaryInterceptor(whatapgrpc.UnaryServerInterceptor()),
        grpc.StreamInterceptor(whatapgrpc.StreamServerInterceptor()),
    )

    // Register service and start server
    pb.RegisterMyServiceServer(server, &myService{})
    server.Serve(lis)
}
```

### Client side

```
func main() {
    trace.Init(nil)
    defer trace.Shutdown()

    conn, err := grpc.Dial("localhost:50051",
        grpc.WithInsecure(),
        grpc.WithUnaryInterceptor(whatapgrpc.UnaryClientInterceptor()),
        grpc.WithStreamInterceptor(whatapgrpc.StreamClientInterceptor()),
    )
    defer conn.Close()

    client := pb.NewMyServiceClient(conn)
    resp, err := client.MyMethod(ctx, &pb.Request{})
}
```

## Kafka instrumentation (Sarama)

```
import (
    "github.com/IBM/sarama"
    "github.com/whatap/go-api/instrumentation/github.com/IBM/sarama/whatapsarama"
)
```

```
func main() {
    trace.Init(nil)
    defer trace.Shutdown()

    config := sarama.NewConfig()

    // Producer wrapping
    producer, err := sarama.NewSyncProducer(brokers, config)
    wrappedProducer := whatapsarama.WrapSyncProducer(producer)

    // Consumer wrapping
    consumer, err := sarama.NewConsumer(brokers, config)
    wrappedConsumer := whatapsarama.WrapConsumer(consumer)
}
```

## Kubernetes client-go instrumentation

```
import (
    "k8s.io/client-go/kubernetes"
    "k8s.io/client-go/rest"
    "github.com/whatap/go-api/instrumentation/k8s.io/client-go/kubernetes/whatapkubernetes"
)

func main() {
    trace.Init(nil)
    defer trace.Shutdown()

    config, err := rest.InClusterConfig()
    if err != nil {
        panic(err)
    }

    // config wrapping
    config = whatapkubernetes.WrapConfig(config)

    clientset, err := kubernetes.NewForConfig(config)
    // Kubernetes API calls are automatically tracked
}
```

# Instrumentation package summary

| Category | Package | Import path |
|---|---|---|
| **Web frameworks** | Gin | `github.com/whatap/go-api/instrumentation/github.com/gin-gonic/gin/whatapgin` |
| | Echo v4 | `github.com/whatap/go-api/instrumentation/github.com/labstack/echo/v4/whatapecho` |
| | Fiber v2 | `github.com/whatap/go-api/instrumentation/github.com/gofiber/fiber/v2/whatapfiber` |
| | Chi | `github.com/whatap/go-api/instrumentation/github.com/go-chi/chi/whatapchi` |
| | Gorilla Mux | `github.com/whatap/go-api/instrumentation/github.com/gorilla/mux/whatapmux` |
| | FastHTTP | `github.com/whatap/go-api/instrumentation/github.com/valyala/fasthttp/whatapfasthttp` |
| | net/http | `github.com/whatap/go-api/instrumentation/net/http/whataphttp` |
| **Databases** | database/sql | `github.com/whatap/go-api/instrumentation/database/sql/whatapsql` |
| | sqlx | `github.com/whatap/go-api/instrumentation/github.com/jmoiron/sqlx/whatapsqlx` |
| | GORM v2 | `github.com/whatap/go-api/instrumentation/github.com/go-gorm/gorm/whatapgorm` |
| | GORM v1 | `github.com/whatap/go-api/instrumentation/github.com/jinzhu/gorm/whatapgorm` |
| **Redis** | go-redis v9 | `github.com/whatap/go-api/instrumentation/github.com/redis/go-redis/v9/whatapgoredis` |

| Category | Package | Import path |
|---|---|---|
| | go-redis v8 | `github.com/whatap/go-api/instrumentation/github.com/go-redis/redis/v8/whatapgoredis` |
| | Redigo | `github.com/whatap/go-api/instrumentation/github.com/gomodule/redigo/whatapredigo` |
| **NoSQL** | MongoDB | `github.com/whatap/go-api/instrumentation/go.mongodb.org/mongo-driver/mongo/whatapmongo` |
| **RPC** | gRPC | `github.com/whatap/go-api/instrumentation/google.golang.org/grpc/whatapgrpc` |
| **Message queue** | Sarama (IBM) | `github.com/whatap/go-api/instrumentation/github.com/IBM/sarama/whatapsarama` |
| | Sarama (Shopify) | `github.com/whatap/go-api/instrumentation/github.com/Shopify/sarama/whatapsarama` |
| **Cloud** | Kubernetes | `github.com/whatap/go-api/instrumentation/k8s.io/client-go/kubernetes/whatapkubernetes` |

# 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 has been implemented to find whatap.conf in the path where it is located. 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.

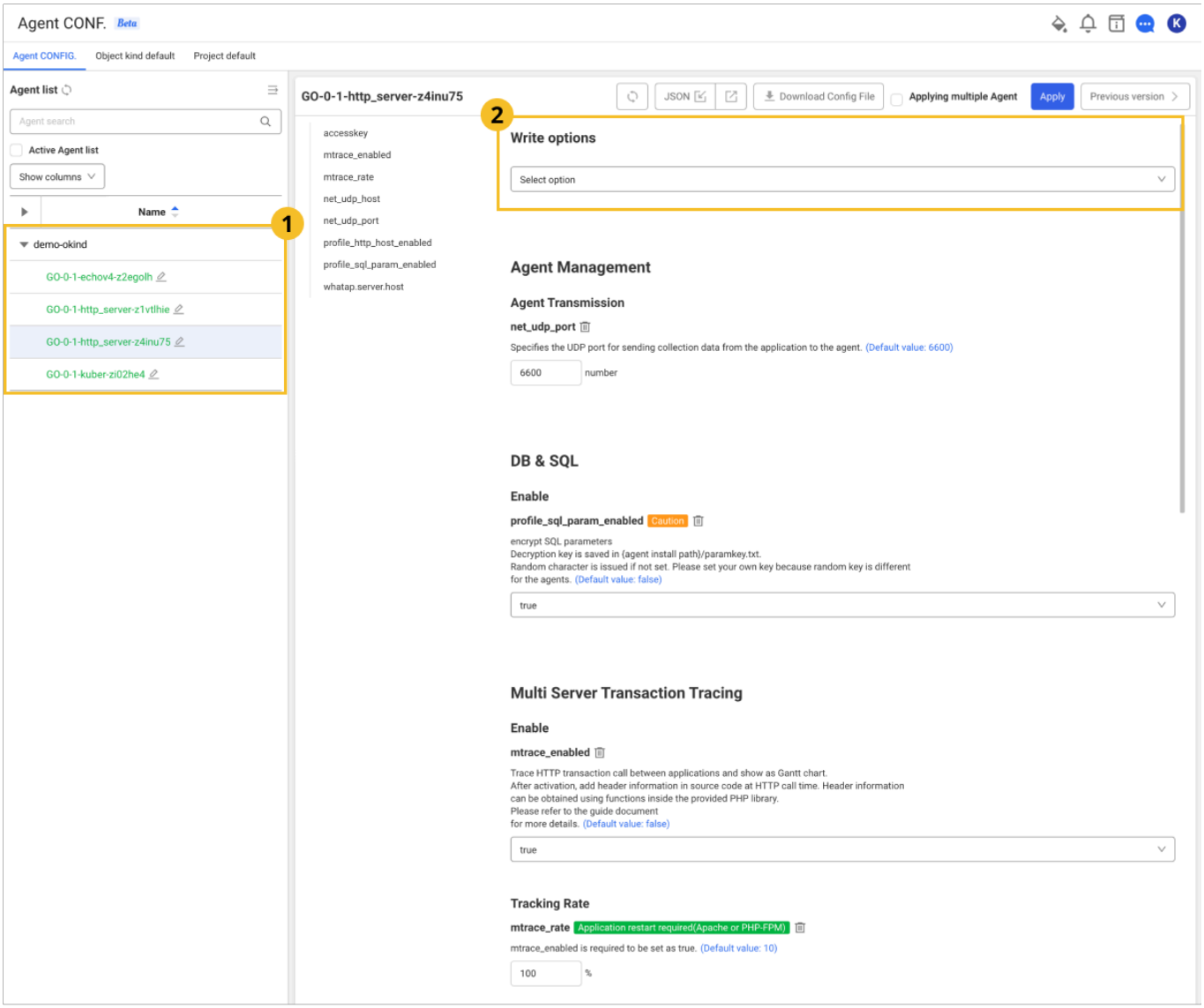
## Configuring the agent on the service screen

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.



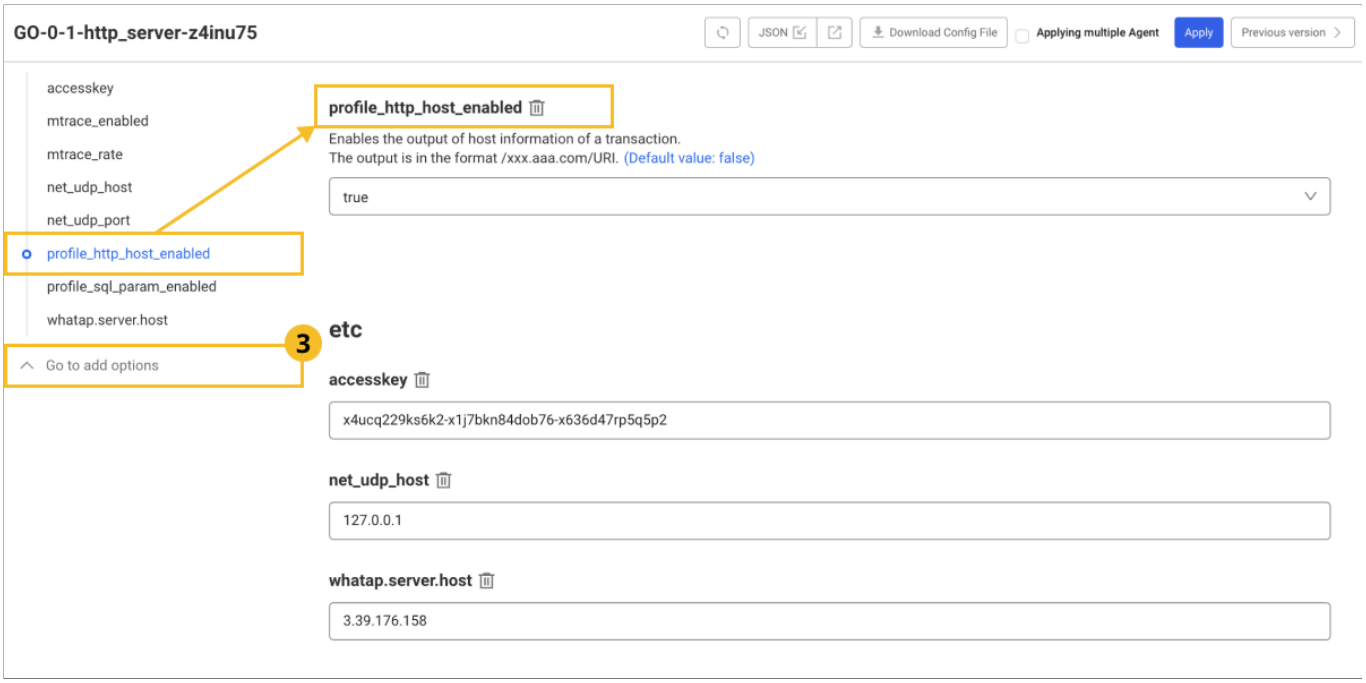

- To cancel the selected option, select .
- To add new options, select **Add** and then repeat Step 2.

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.

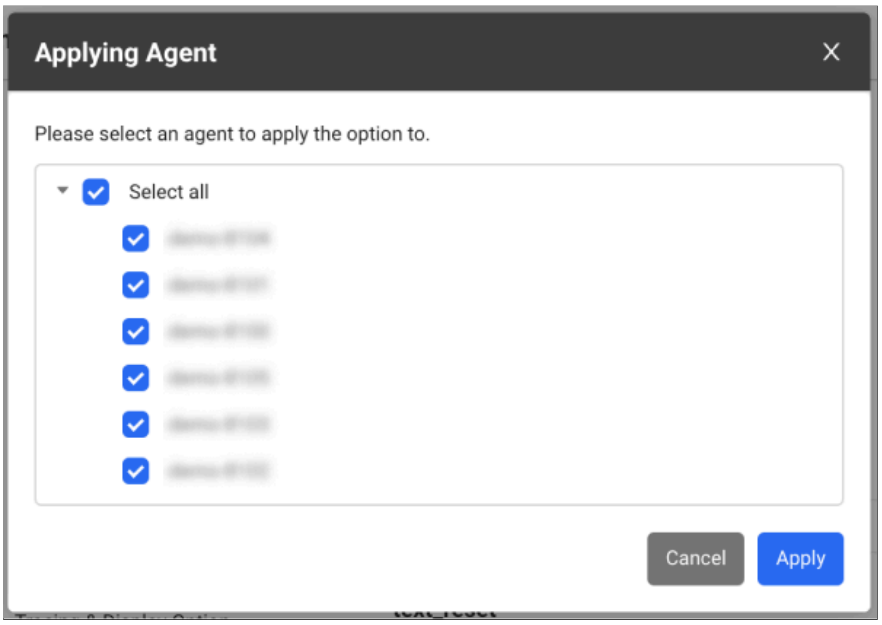
> ⓘ
> - If you want to move to the top of the screen, select 3 **Go to add options**.
> - If you select **Apply** without entering options, the options are deleted.

## 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 identification**

The following explains how to set the agent name in order to identify the monitoring target.

### Control of agent functions

The following provides the method how to control the functions of the application server.

### Agent communication settings

The following provides the method how to set the communication for the application server.

### Agent Log

The options of the agent log can be set.

### Transactions

The agent traces the transaction performance and provides the following configurable options.

### Tracing DB and SQL

It provides the options related to the Go agent's DB and SQL.

### HTTPC, API Call

It provides the options regarding the PHP agent's HTTPC and API call.

### User count

It aggregates the numbers of real-time users connected to WAS.

### Apdex

It describes the Go agent Apdex.

### Statistics

It describes the Go agent statistics.

### Configuration for each library

It guides you to the agent configuration for each Go library.

# 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.

## Agent naming method

The default patterns used to identify the application server are as follows:

Default pattern

```
{type}-{ip2}-{ip3}-{cmd}-{cmd_full}
```

To change the default pattern value, modify whatap.conf as follows:

whatap.conf

```
object_name={type}-{ip2}-{ip3}-{process}-{dockerid}-{ips}
```

## Variable of the agent name

| Variable | Description |
|---|---|
| {type} | Name of the application server |
| {ip0} | The first byte of the IPv4 address is used. (e.g. **10** in **10**.11.12.13) |

| Variable | Description |
|---|---|
| {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**) |
| {process} | Process name of the application server |
| {hostname} | Host name |
| {dockerid} | ID of the docker container |
| {ips} | It is the full IP (IPv4, IPv6). |
| {cmd} | Process name. It is the value of the executable args[0]. |
| {cmd_full} | All command lines are processed using hash value. They correspond to all executable args. |

# Setting the agent name

- **object_name** String

  Default {app_name}-{ip2}-{ip3}-{cmd}-{cmd_full}

  It is an agent naming (ONAME) scheme to identify applications. An OID is created based on the ONAME.

- **app_name** String

  It is a component used in the agent naming (ONAME) scheme to identify applications. It corresponds to the {type} among the object_name options.

- **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.

# Control Of Agent Functions

It can control the Go agent function.

- **enabled** Boolean

  Default `true`

  It enables all the functions.

- **accesskey / license** String

  Set the access key granted by the server when installing the agent. The access key includes the agent's project and the encryption key for secured communication.

- **stat_enabled** Boolean

  Default `true`

  It enables the function to trace the statistics. Statistical data such as transactions, SQL, HTTPCALL, UserAgent, and client IP collected in every 5 minutes are applicable.

- **realtime_user_thinktime_max** Milliseconds

  Default `300000`

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

- **time_sync_interval_ms** Milliseconds

  Default `300000`

  Set the synchronization interval between the agents and the server. For no synchronization, set it to 0.

- **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 you set a value that is different from the previous setting, it will be sent again.

  Transaction URLs and SQL strings correspond to text data.

- **use_env_first** Boolean

  Default `false`

The priority of agent configuration is that the whatap.conf file is applied first, followed by environment variables. To give priority to environment variables, set the option value to `true`. Alternatively, you can set the environment variable as `WHATAP_USE_ENV_FIRST=true`.

# Agent Communication Settings

The following provides the method how to set the communication for the Go application server.

- **whatap.server.host** ip_address

  Default `127.0.0.1,127.0.0.1`

  Set the server to deliver the agent's collected data to. If there are multiple IPs due to the collection server redundancy, they can be specified by separating with comma (,). The collection server's proxy daemon must be serviced in the listening state for the specified IP.

- **whatap.server.port** tcp_port

  Default `6600`

  Set the port of the collection server. Because only one port can be specified, the collection servers specified in whatap_server_host must use the same port.

- **tcp_so_timeout** Millisecond

  Default `60000`

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

- **tcp_connection_timeout** Millisecond

  Default `5000`

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

- **net_send_max_bytes** Byte

  Default `5242880`

  Set the maximum size of the data to send to the collection server at one time.

- **net_ipc_host** ip_address

  Default `127.0.0.1`

  It is the agent's TCP address for sending the data collected by the application via IPC/TCP to the agent. It is not recommended, but use this option when the agent is running on an external server.

> To apply the settings, restart the application.

- **net_ipc_port** tcp_port

  Default `6600`

  The WhaTap agent transfers the data collected by the tracer to the collection server via IPC/TCP. You can specify the port of the initial TCP server. If the provided port 6600 is in use, you can use this option. To change the TCP server port of the agent, modify in the /usr/whatap/agent/whatap.conf file path.

  > ◦ If the server port has been changed, update it also in the whatap.conf file of the application to establish the communication.
  >
  > ◦ To apply the settings, restart the application.

- **net_so_time** Int

  Default `120000` (Millisecond)

  You can set the maximum waiting time when reading data upon TCP connection. After the set time has elapsed, it is processed as a timeout.

- **net_write_buffer_size** Int

  Default `2*1024*1024` (2 Mb)

  This option sets the size of the buffer used when transmitting data collected over the network.

# Agent Log

When the application server runs, it starts collecting the monitoring data. The collected data is stored in the log file.

## Log setting

**log_keep_days** Day

Default `7`

Set the retention period of log files.

## Setting the LogSink

It is used to log data.

### Default settings

- **logsink_enabled** Boolean

  Default `false`

  If the value is `true`, the log data is collected.

- **logsink_stdout_enabled** Boolean

  Default `false`

  If the value is `true`, standard output logs are collected.

- **logsink_stderr_enabled** Boolean

  Default `false`

  If the value is `true`, standard error output logs are collected.

- **logsink_zip_enabled** Boolean

Default `true`

If the value is `true`, the zipped log data is transmitted. The default value is GZIP.

## Setting the log file

- **logsink.files** String

  You can set the full path and file name of the log file to be collected. To set multiple files, use comma (,) as the delimiter.

## Example

See the following example and then apply the agent settings for your environment.

whatap.conf

```
logsink_enabled=true
logsink_stdout_enabled=true
logsink_stderr_enabled=true
logsink_zip_enabled=true
logsink.files=/etc/httpd/logs/error_log, /etc/httpd/logs/access_log
```

You can use any variable in date format in log file names. Apply it based on the use of the Python `strftime()`.

e.g. 20240101: `%Y%m%d`

whatap.ini

```
whatap.logsink.files=/etc/httpd/logs/customlog-%Y%m%d
```

This function is supported in Go agent 0.3.2 or later.

For the API guide on log settings, see the following.

# Transactions

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.

> ⓘ For more information about transactions, see the following. For guidance on the transaction analysis using the hitmap, see the following.

## 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** Millisecond

  Default `500`

  Transaction 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.

- **profile_step_normal_count** Int

  Default `1000`

  Set the maximum step count for tracing transactions.

- **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 you increase the setting too high, the memory usage increases. We recommend that you do not change the default values.

- **profile_step_heavy_count** Int

  Default `1000`

  If the collection number of basic steps is exceeded, only the steps whose execution time exceeds profile_step_heavy_time are collected. Set the count of the step that can be collected. In case of default settings, if `profile_step_normal_count` is 800, up to 200 steps are collected.

- **profile_step_heavy_time** Millisecond

  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 within `profile_step_heavy_count`.

- **query_string_enabled** Boolean

  Default `false`

  It enables the function to collect query strings of the transaction URL together. It is applied to only the URLs registered in `query_string_urls`.

- **query_string_urls** String

  Register URLs to collect query strings from every transaction.

- **split_profile_enabled** Boolean

  Default `false`

  More profiles are collected by separating a certain amount of profiles into virtual transactions. They are divided into multiple transaction lists.

> If set at the same time, the priorities are as follows:
>
> 1. split

> 2. large
> 3. circular

- **large_profile_enabled** Boolean

  Default `false`

  By sending split and compressed profiles, you can collect more profiles without increasing memory usage.

  > ◦ We can collect up to 10,000 profiles (steps). For more data, it is recommended to use the `split_profile_enabled` option.
  > ◦ If set at the same time, the priorities are as follows:
  >
  > 1. split
  > 2. large
  > 3. circular

- **circular_profile_enabled** Boolean

  Default `false`

  The last 1024 profiling data is collected.

  > If set at the same time, the priorities are as follows:
  >
  > 1. split
  > 2. large
  > 3. circular

- **profile_zip_enabled** Boolean

  Default `true`

  Compressed profiles are transmitted. The default value is `true`.

- **trace_active_transaction_slow_time** Millisecond

Default `3000`

In the active transaction arc equalizer graph on the dashboard that checks collected data, specify the criteria for the transaction response time that can be recorded as a slow section. If the transaction response time exceeds the set time, the number of slow active transactions is included.

- **trace_active_transaction_very_slow_time** Millisecond

  Default `8000`

- **trace_active_transaction_lost_time** Millisecond

  Default `300000`

  If a transaction does not end within the timeout period (5 minutes), no further transaction data is collected. You can see `Lost Connection` in the transaction trace information.

- **trace_useragent_enabled** Boolean

  Default `false`

  If the value is `true`, the user agent data for transactions is collected.

- **trace_referer_enabled** Boolean

  Default `false`

  If the value is `true`, the referer data for transactions is collected.

# 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. Exception-handled items appear as normal transactions in the **Hitmap** or **Trace analysis** window.

- **biz_exceptions** String

  Set a specific error or exception as business exception. It is not recorded as an error in the hitmap but collected in the statistics. It is applied the same as the error class name that is collected. Multiple items can be registered using comma (,) as the delimiter.

  whatap.conf

```
biz_exceptions=*errors.errorString, *echo.HTTPError, *url.Error, *exceptions.CustomErrors
```

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

- **ignore_exceptions** String

  The error itself by the registered exception is ignored. The string to exclude is for the class related to the error type or error statistics in the profile.

  The type of error that occurs in Go is displayed as a string.

```
thr.ErrorClassName = fmt.Sprintf("%T", err)
thr.ErrorMessage = err.Error()

# Output the error type

*errors.errorString
*echo.HTTPError
*url.Error
```

  whatap.conf

```
ignore_exceptions=*errors.errorString, *echo.HTTPError, *url.Error, *exceptions.CustomErrors
```

> Ignore error class/Hitmap display level: Excludes the INFO (blue)/error statistics.

# 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 convert and normalize transaction URLs.

- **trace_normalize_urls** String

  Set the target URL to transform and normalize. It deletes the path parameter by transforming the call URL patterns.

  > **Example**
  >
  > If it is set as `/a/{v}/b` , the called transaction URLs are transformed to the `/a/{v}/b` format. To register multiple items, comma (,) is used as the delimiter.

- **trace_ignore_url_set** String

  If a specific URL is set, the corresponding transactions are excluded from tracing. The set URLs are not displayed in the transaction list because they do not collect transaction data. Set URLs to exclude from transaction tracing. To set multiple items, 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_url_prefix** String

  It sets URL prefixes to exclude from the transaction tracing.

  > The URLs registered through this option are excluded from the **Hitmap**, **Trace analysis**, or **Transaction information** window for collection.

- **profile_http_header_enabled** Boolean

  Default `false`

  It is used to record the HTTP headers in the trace history.

- **profile_http_header_ignore_keys** String

  Default `host,accept,user-agent,referer, accept-language, connection`

It is used to record HTTP headers in the profile details. The set HTTP header names are excluded from collection, and they are marked with "#."

- **profile_http_parameter_enabled** Boolean

  Default `false`

  It is used to record HTTP parameters in the trace history. The parameters can be viewed by entering a separate security key. The name and value of the text type are collected among Get and Post parameters. Up to 40 items (20 Gets, 20 Posts) are collected, and parameter names and values are collected up to 256 bytes.

  > ◦ Go agent v2.4.0 or later: The security key is set to 6 characters ( `paramkey` ) in the `WHATAP_HOME` /security.conf file where the whatap.conf file is located.
  >
  > ◦ Below Go agent v0.4.0: The security key is set to 6 characters within the `WHATAP_HOME` /paramkey.txt file where the whatap.conf file is located. If the paramkey.txt file does not exist, it is automatically created with a random value.

- **profile_http_parameter_url_prefix** String

  Default `/`

  It is used to define the prefix of URL to record HTTP parameters in the trace details.

- **profile_http_host_enabled** Boolean

  Default `false`

  The transation host is output. If the value is `false` , only the URI is recorded in the URL of a transaction. If the value is `true` , the output has the format of `/xxx.aaa.com/URL` .

- **ignore_http_method** String

  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.

# 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 `false`

  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`

  This option sets the issuance rate of the MTID (Multi Transaction ID) issued when the first transaction occurs.

- **mtrace_poid_key** String

  Default `x-wtap-po`

  Set the key name to deliver the caller project information to for MTID tracing.

- **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 callees. At this time, the URL length is restricted. This sets the length value.

# Tracing DB And SQL

- **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. The literal part is denoted with "#."

- **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.

  > ◦ Go agent v2.4.0 or later: The security key is set to 6 characters ( `paramkey` ) in the `WHATAP_HOME` /security.conf file where the whatap.conf file is located.
  >
  > ◦ Below Go agent v0.4.0: The security key is set to 6 characters within the `WHATAP_HOME` /paramkey.txt file where the whatap.conf file is located. If the paramkey.txt file does not exist, it is automatically created with a random value.

# HTTPC, API Call

- **trace_httpc_normalize_enabled** Boolean

  Default `true`

  It enables the function to parse and normalize HTTPC URLs for transactions.

- **trace_httpc_normalize_urls** String

  Define 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.

# User Count

It collects the numbers of users connected to the Go web application server.

- **trace_user_enabled** Boolean

  Default `true`

  Set whether or not to aggregate real-time users. If you set the value to `true`, it traces users based on the IP address or cookies.

- **trace_user_using_ip** Boolean

  Default `true`

  Set it when classifying real-time users by the IP address. This data can be found in Real Time User. If the value is `false`, users are identified based on the cookie name in the `trace_user_cookie_keys` option.

- **trace_user_cookie_keys** String

  Use the value that corresponds to the cookie key as the user identification value. With comma (,) as the delimiter, multiple values can be set.

- **trace_user_header_ticket** Boolean

  Default `false`

  To identify users based on a specific value in the HTTP header, set the HTTP key to use. If the HTTP key cannot be found, users are identified based on the cookie.

  > ⓘ This cannot be applied together with the `trace_user_using_ip` option.

- **trace_user_set_cookie** Boolean

  Default `false`

  To identify users, the user identification values are set as `WHATAP` key names in cookies. Set it when the `trace_user_header_ticket` option has been set and the header key cannot be found. If the HTTP key cannot be found, users are identified based on the cookie.

  > ⓘ This cannot be applied together with the `trace_user_using_ip` option.

- **trace_http_client_ip_header_key** String

  Default `X-Forwarded-For`

  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.

- **trace_http_client_ip_header_key_enabled** Boolean

  Default `true`

  If the value is `false`, the HTTP Header is ignored and the client IP is set as the remote IP value.

# Apdex

**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)

# Statistics

- **stat_domain_enabled** Boolean

  Default `false`

  It collects transaction statistics for each domain.

- **stat_domain_max_count** Int

  Default `7000`

  Maximum number of records of transaction statistics for each domain that collects 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. Any string can be specified. The 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_referer_enabled** Boolean

  Default `false`

  It collects transaction statistics by referer.

- **stat_referer_max_count** Int

  Default `7000`

  Maximum number of records of transaction statistics for each referer that collects by JVM for 5 minutes.

- **whatap.okind** String

  Arbitrarily assign it for the group topology. By setting multiple agents together, the topology can be checked

by the group name.

# Configuration For Each Library

It guides you to the agent configuration for each Go library.

goolge.golang.org/grpc

- **grpc_profile_enabled** Boolean

  Default `true`

  Set whether or not to collect the Google Remote Call (gRPC) data.

- **grpc_profile_stream_client_enabled** Boolean

  Default `true`

  Set whether or not to collect the client stream methods.

- **grpc_profile_stream_server_enabled** Boolean

  Default `true`

  Set whether or not to collect the server stream methods.

- **grpc_profile_ignore_method** String

  It does not collect the method for the set value. For multiple values, use comma (,) as the delimiter.

- **grpc_profile_stream_method** String

  It organizes the set stream method as a separate transaction. For multiple values, use comma (,) as the delimiter. For long keeping stream connections, each method call is treated as a single transaction. You can search by the full method in the hitmap and transaction search.

- **grpc_profile_stream_identify** Boolean

  Default `false`

  The stream full method is collected as a transaction name, and a prefix that can distinguish the client and server for the same full method is added.

  whatap.conf

```
grpc_profile_stream_identify=/StreamClient/[full method], /StreamServer/[full method]
```

# 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.

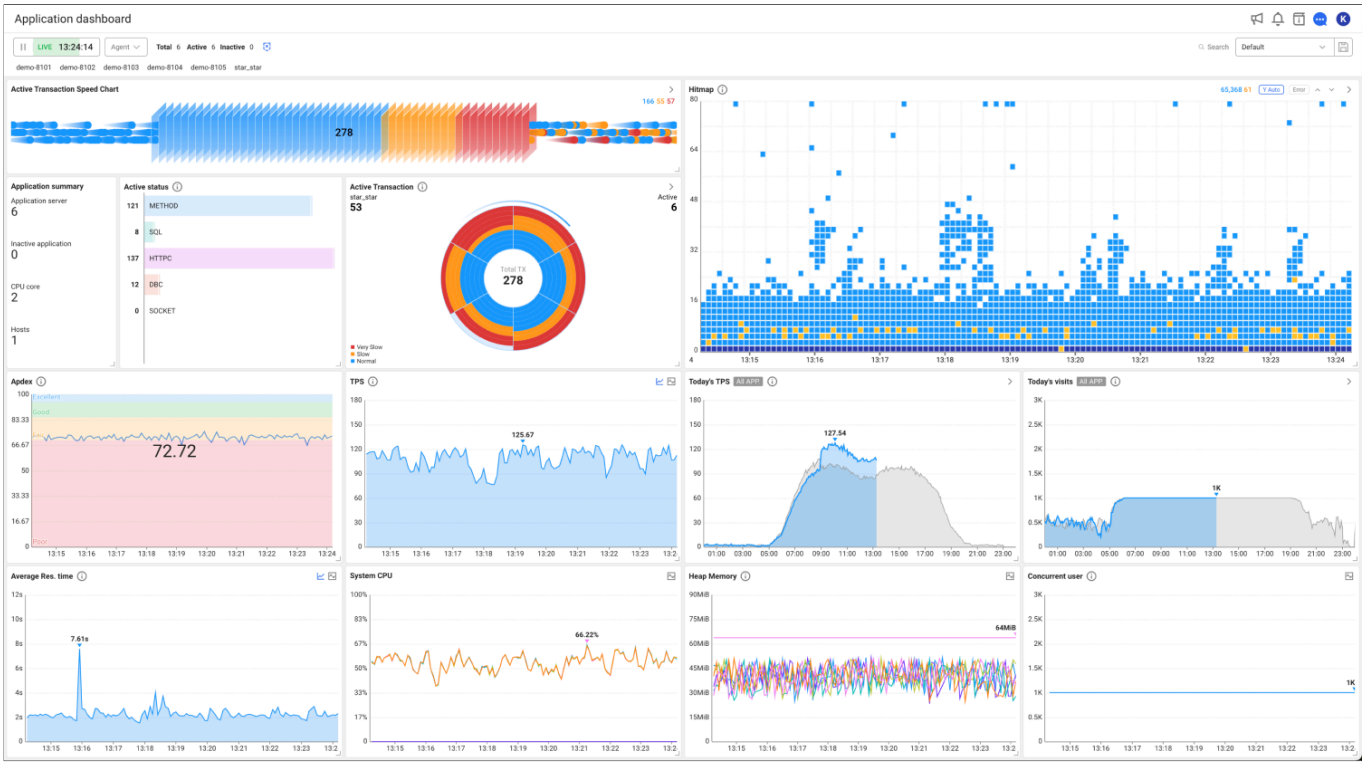
From the dashboard, you can see the following information:

- 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.

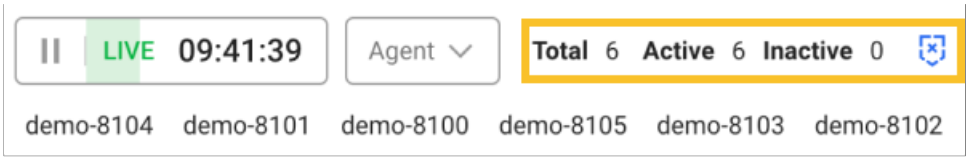
## Query Time Setting

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

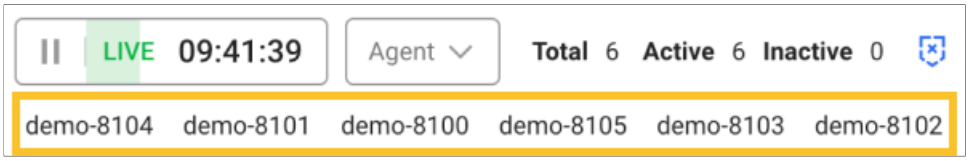
## Checking the agent connection status




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

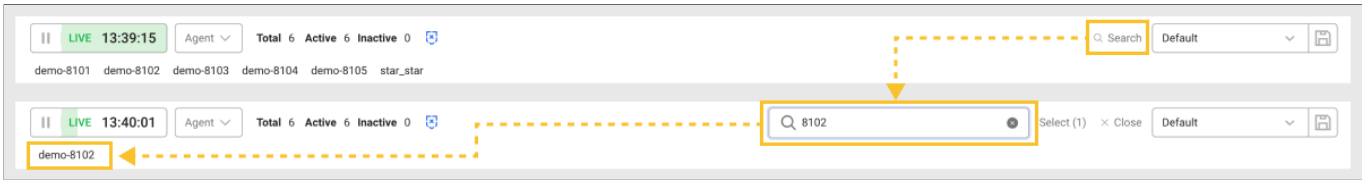
You can easily monitor multiple agents by categorizing them in Agent Settings.

- **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.



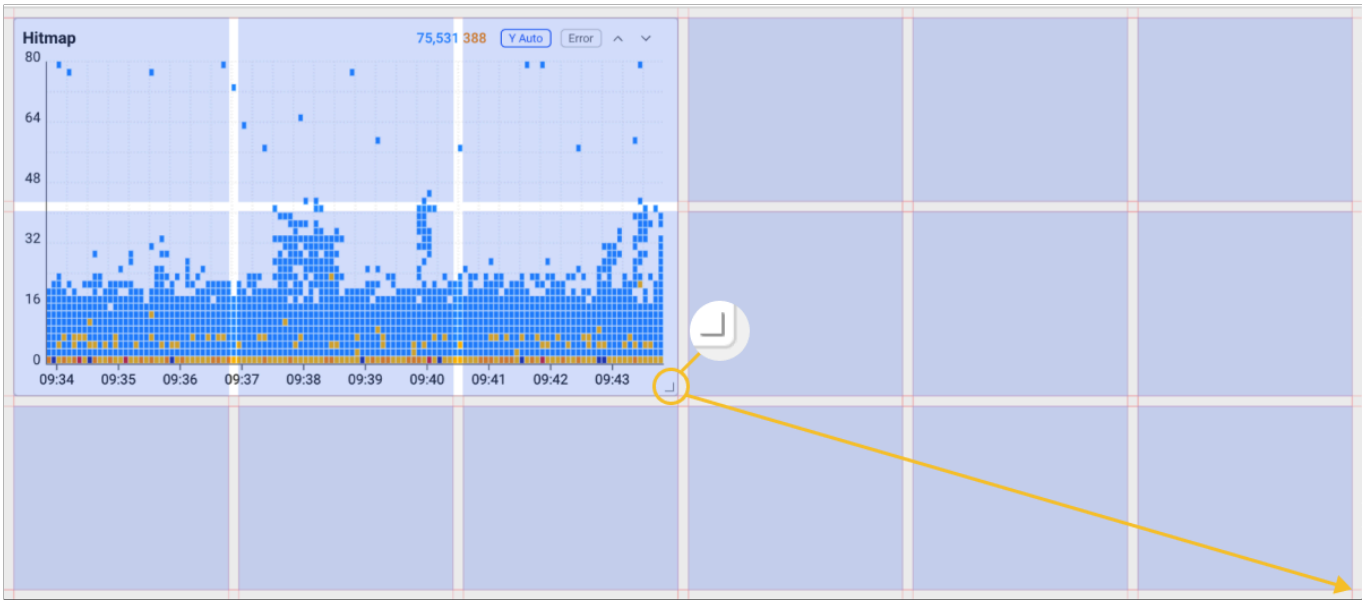

The agents that match the entered string appear in the list. To filter the data in the dashboard based on the searched agents, click **Select** on the right of the text entry box.

To reset the selected agent entries, click **Close** on the right of the input field.

# Widget Edit

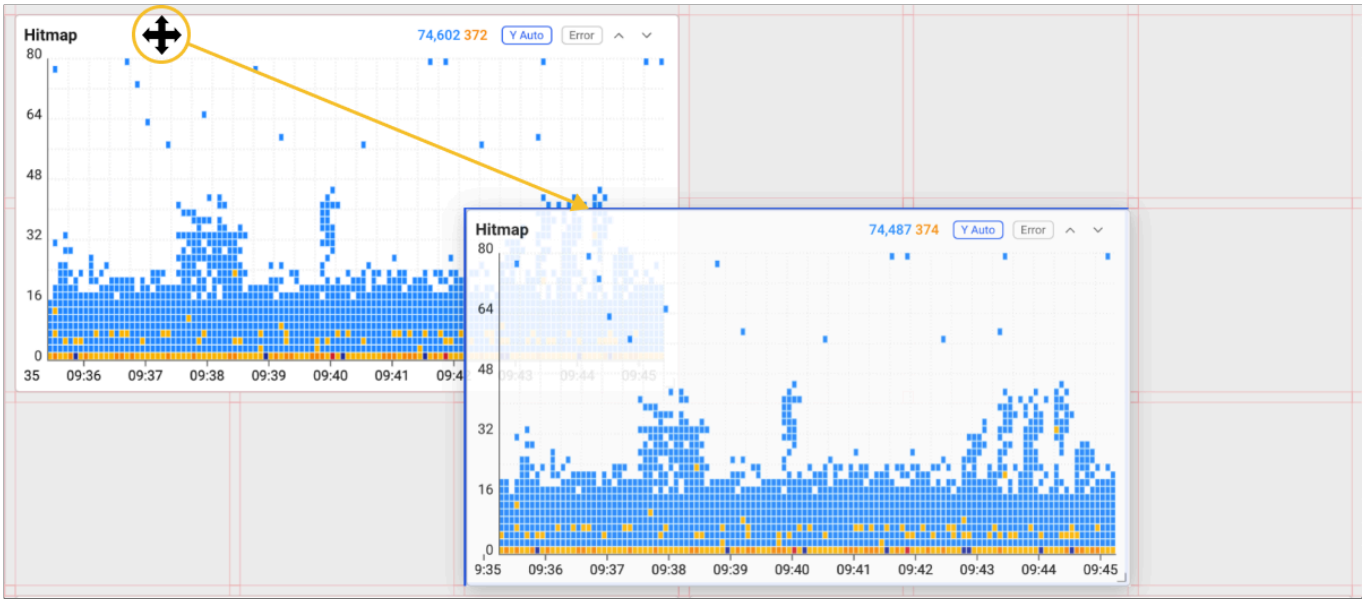
The widgets on the dashboard can be adjusted to the desired size and placed in the desired positions. You can also delete unnecessary widgets or add them again.

## Resizing a widget



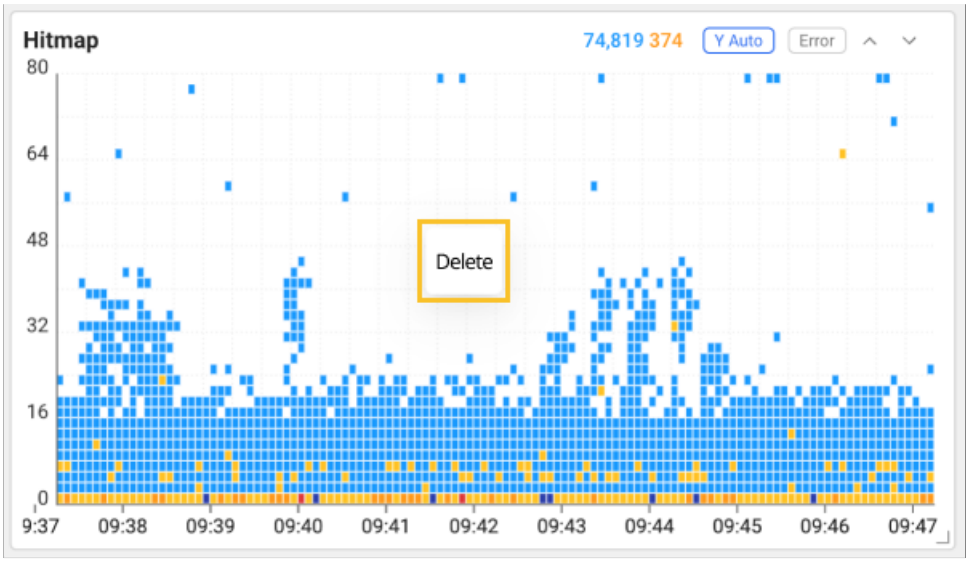

Click and hold the mouse on the ⌟ element at the lower right of the widget, and then drag it to the desired size. Grids with a uniform horizontal-to-vertical ratio appear and the size of the widget can be adjusted in each grid.

## Moving a widget



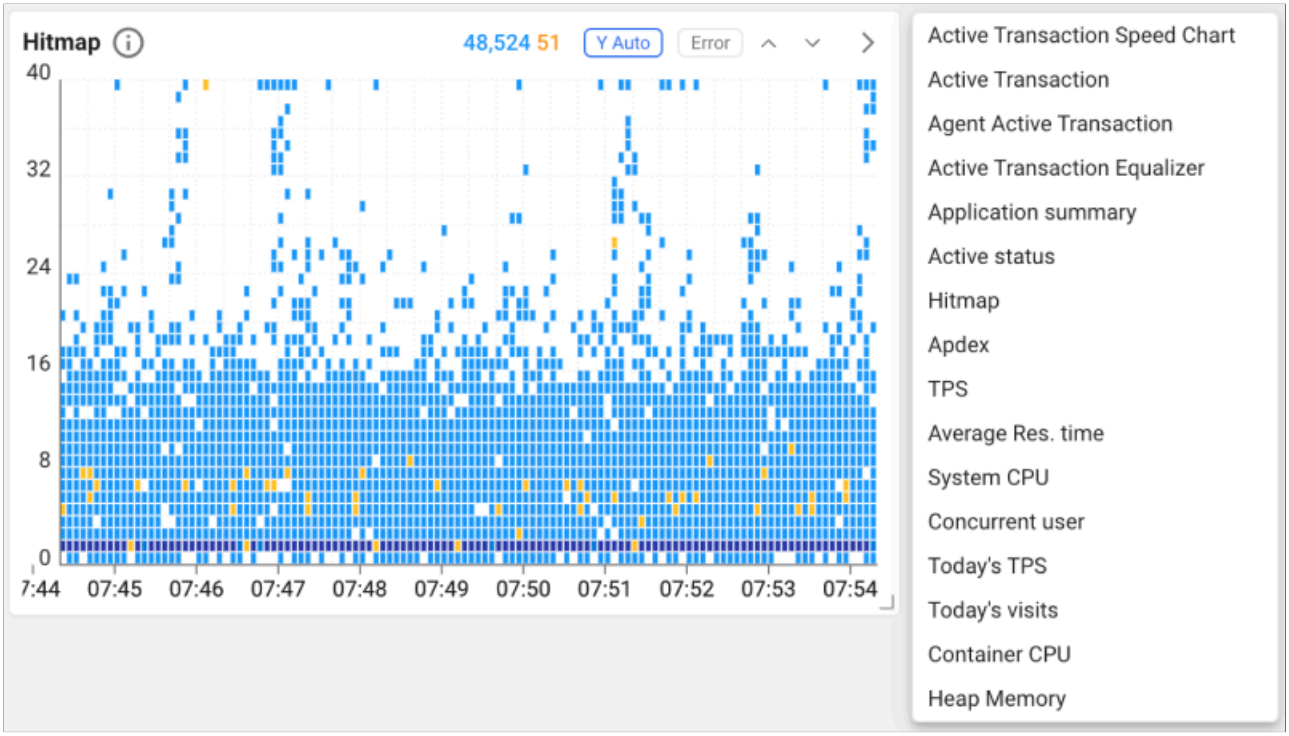

When you move the mouse cursor to the top of the widget, the cursor shape changes to ✢. At this time, you can move the widget by dragging with your left mouse button and move to the desired position.

## Deleting a widget



Right-click on a widget to delete. If you select **Delete**, the widget is deleted from the dashboard.

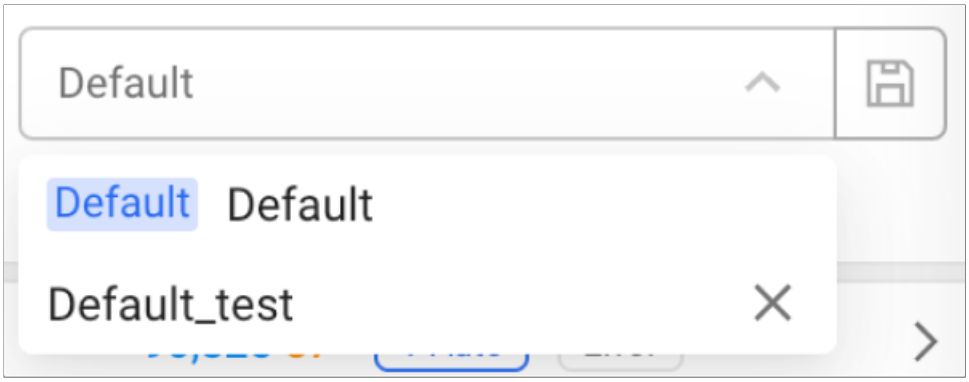
## Adding a widget




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

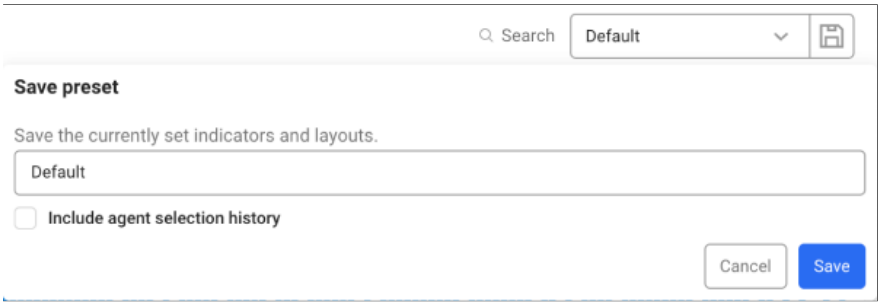



You can save and load the settings and layouts for widgets on the dashboard. You can create a new preset by adjusting its size and placing it on the desired position.

- 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

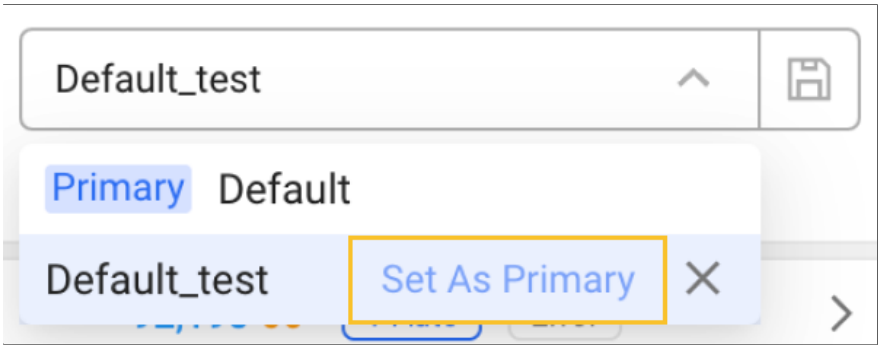
If you have unused presets, you can delete them from the preset list. Select ✕ on the right of the item to delete from the preset list.

The **Default** preset cannot be deleted.

## Changing the default preset

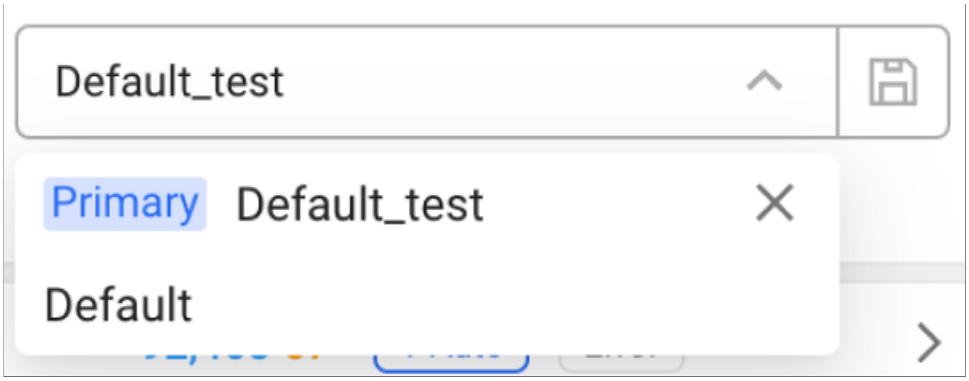
You can change the preset displayed when you first enter the **Application Dashboard** menu.

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.

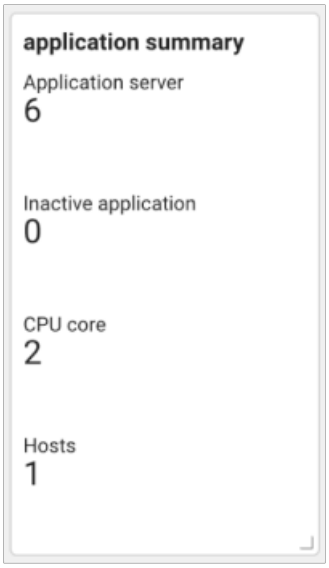
# Learning about the widgets

> ⓘ 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

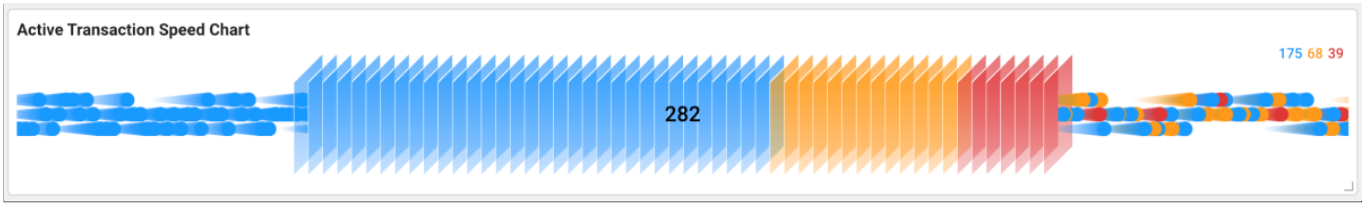



It displays the execution status and number of applications registered in the project, CPU core count, and host count.

## 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.

### 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

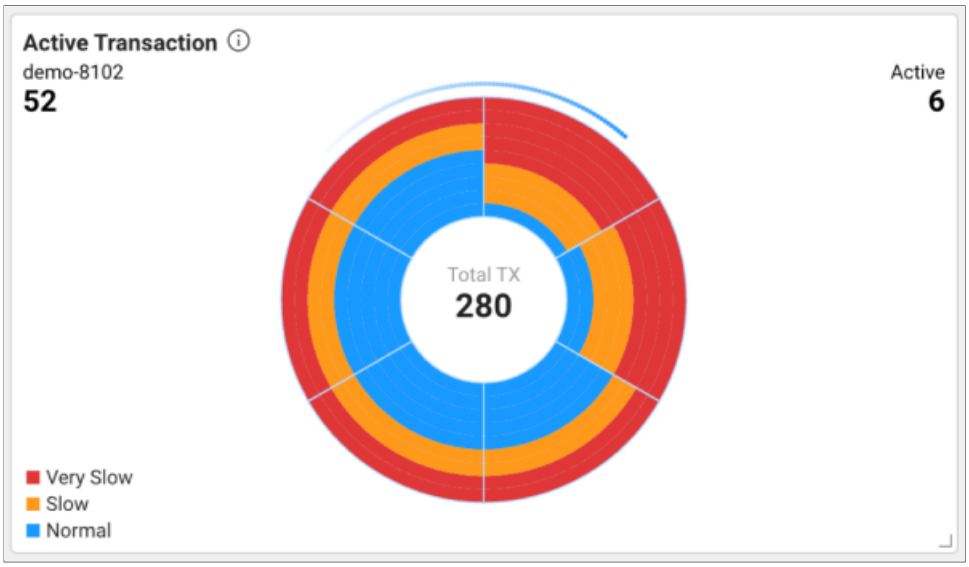
  This indicates the Application Performance Index (Apdex). This metric is measured for customer satisfaction

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



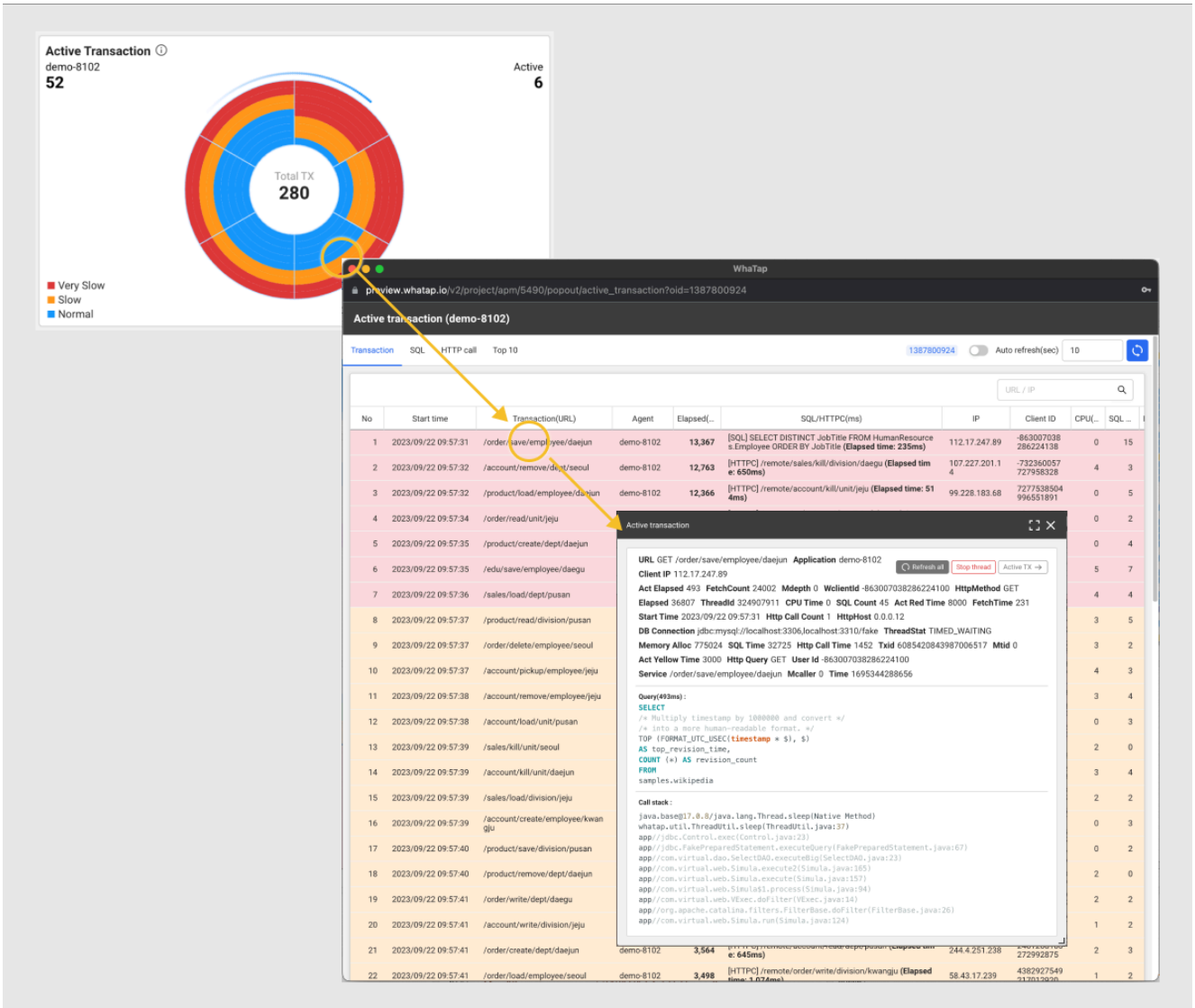

In the **Active Transaction** widget, the transactions in progress are displayed in the **arc equalizer** chart. You can see the status of delays by agent. The arc is divided by the number of agents. By representing the number of requests being processed by the server every 5 seconds, you can tell how 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

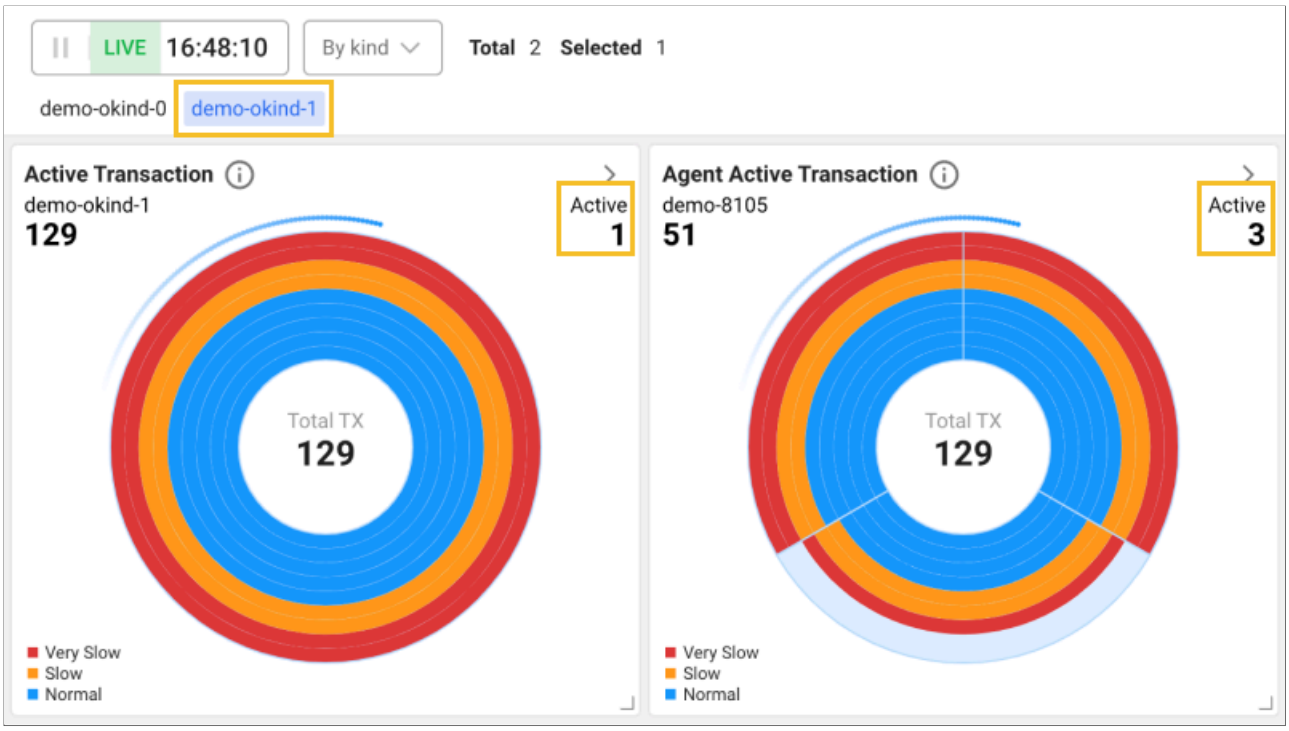
The situation where lots of blue colors are represented is not a problem. You must watch the trend to see whether some of them turn to orange or red. In case of a failure, the number of active transactions increases and the percentage of red color increases. On the other hand, a very responsive system 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.

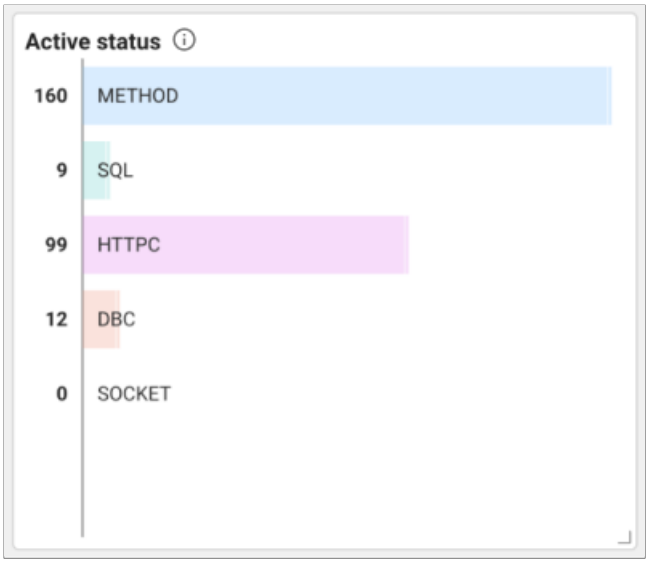
## Agent Active Transaction




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



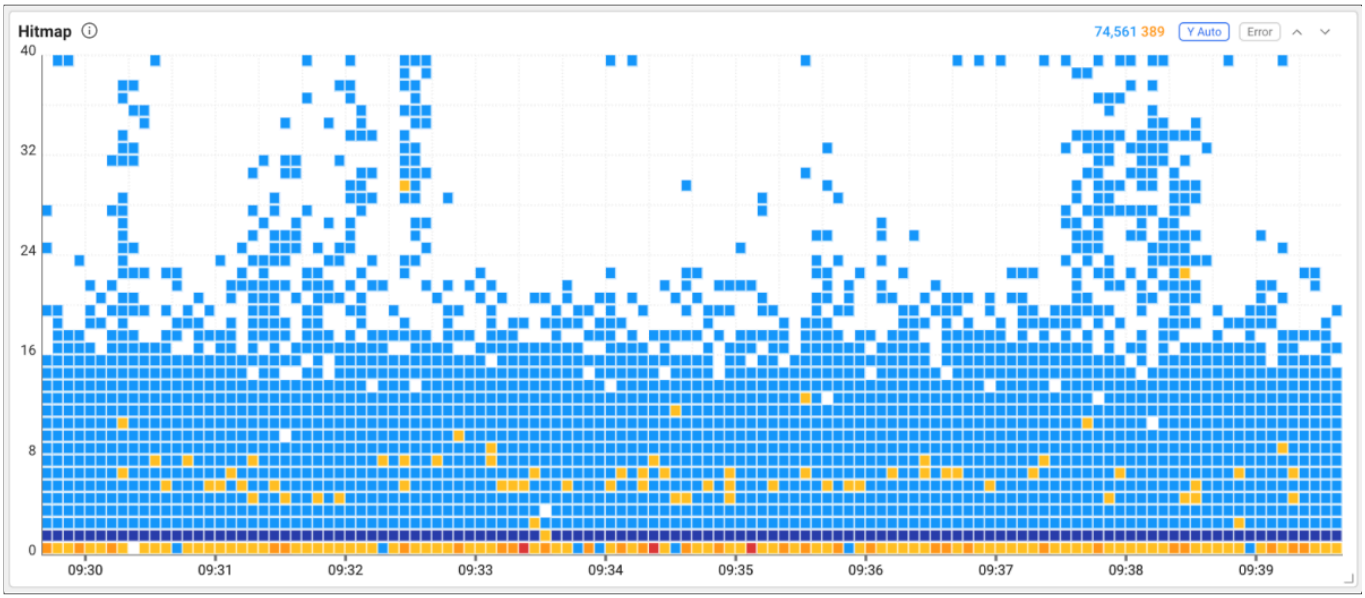

All agents in the project collect the active transaction count every 5 seconds, along with the progress status. The progress status is classified into **METHOD**, **SQL**, **HTTPC**, **DBC**, and **SOCKET**.

- **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

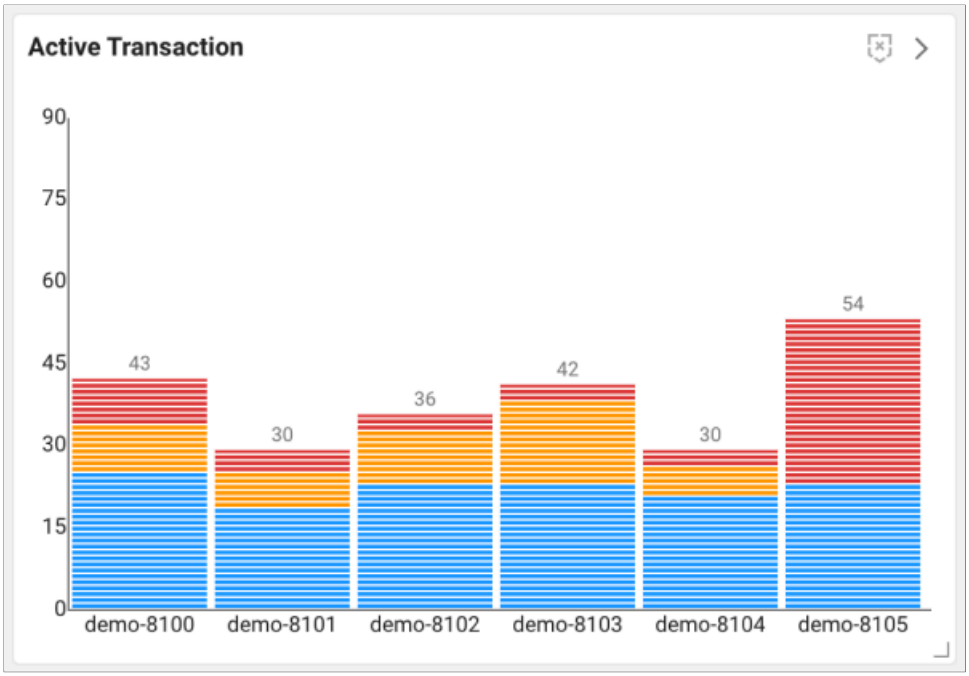



The **Hitmap** chart represents the distribution of response times of completed transactions. The horizontal axis is the transaction end time. The vertical axis is the execution time. Transactions with longer execution times are at the top of the distribution. If you drag the chart area in **Hitmap**, 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.

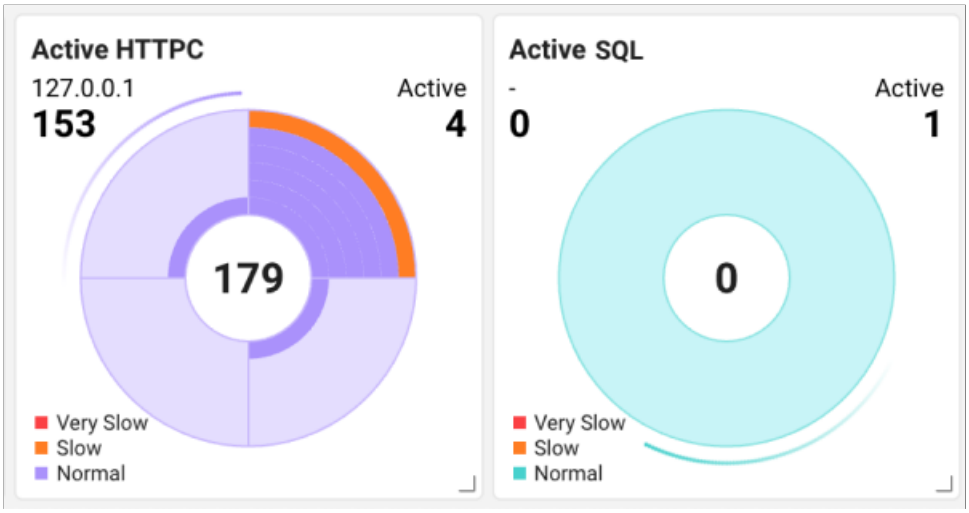
## Active Transaction Equalizer




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.

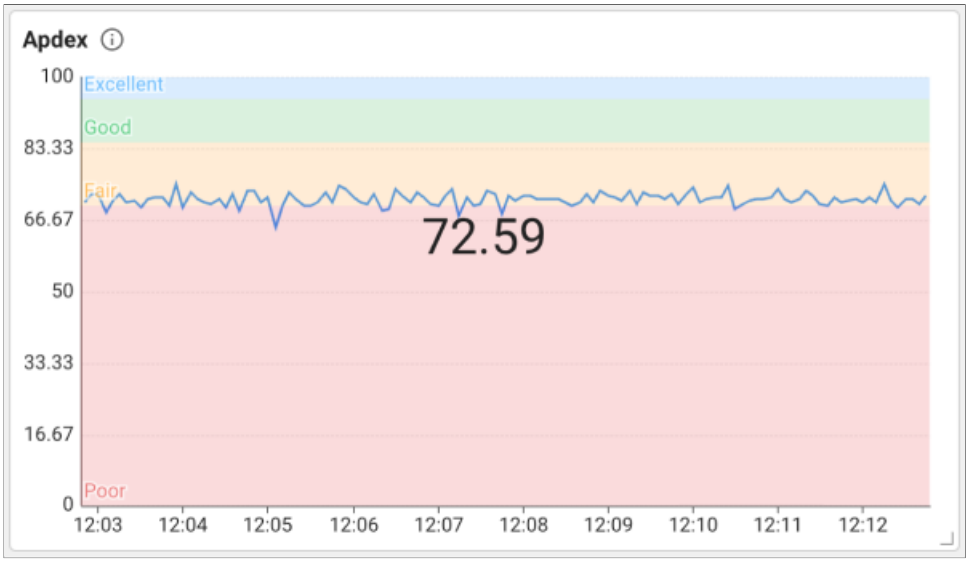
1. In the **Active HTTPC Top 100** popup, check the transaction list and trends.
2. Select a transaction that requires detailed analysis.

   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



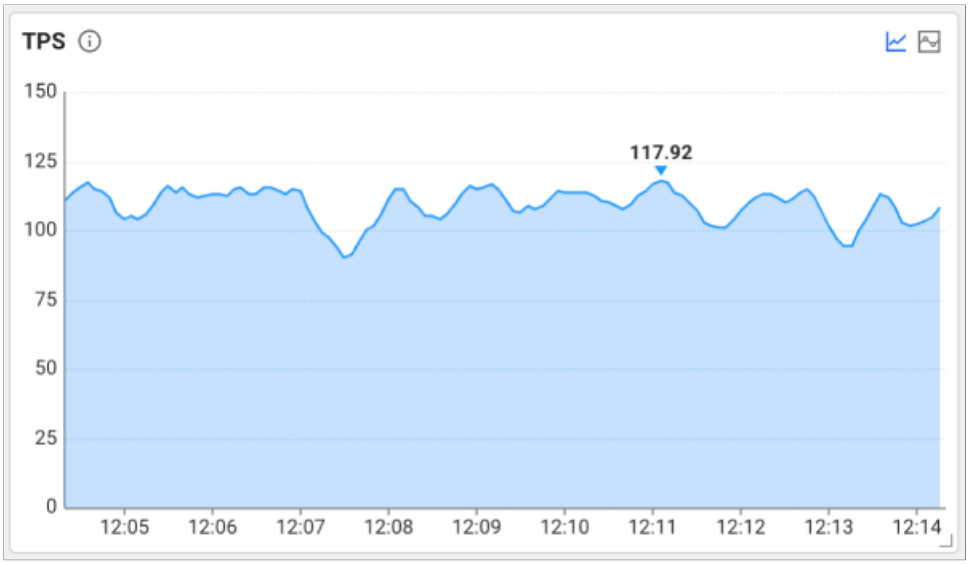

Application Performance Index (Apdex) is the indicator of the application performance. You can measure the customer satisfaction about your web applications. It can be used as an indicator of user satisfaction and the value is between 0 and 1.

(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.

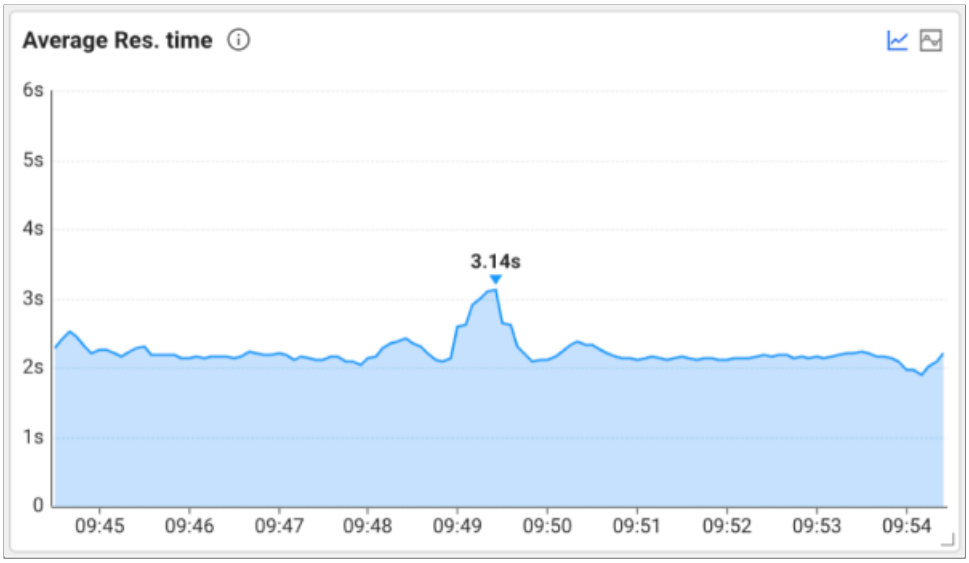
## TPS




**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

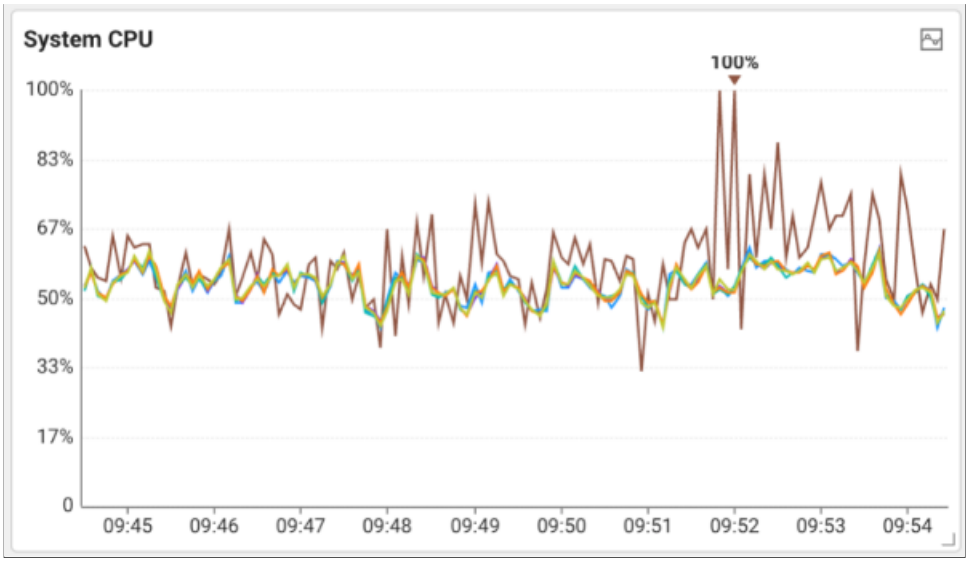



It monitors the times needed for the application server to return the request result. **Average Res. time** is meaningful as a tuning metric. The WhaTap's monitoring service calculates the average response time for transactions every 5 seconds.

> For more information about analyses for TPS, response time, and CPU utilization, see the following.

## Resource-related widget

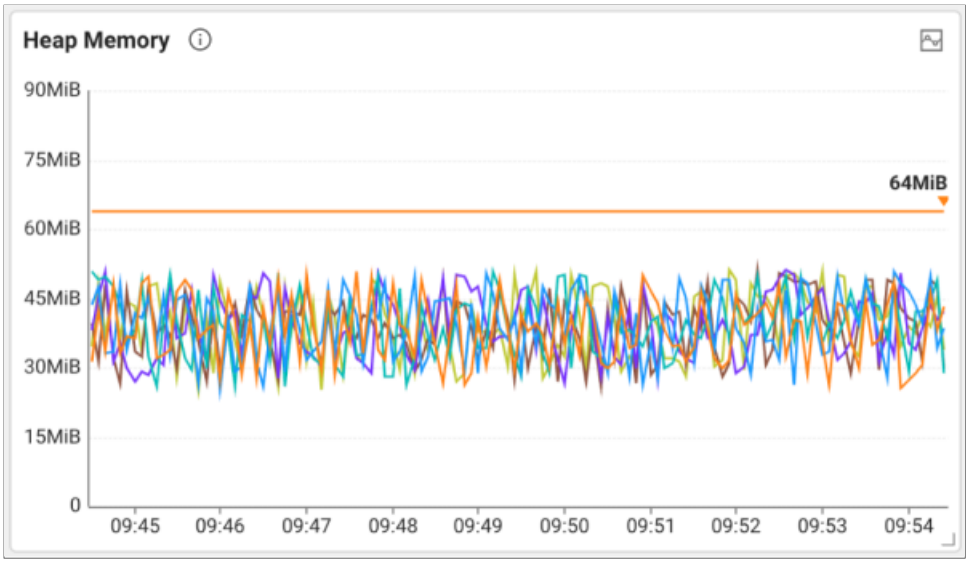
### System CPU




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

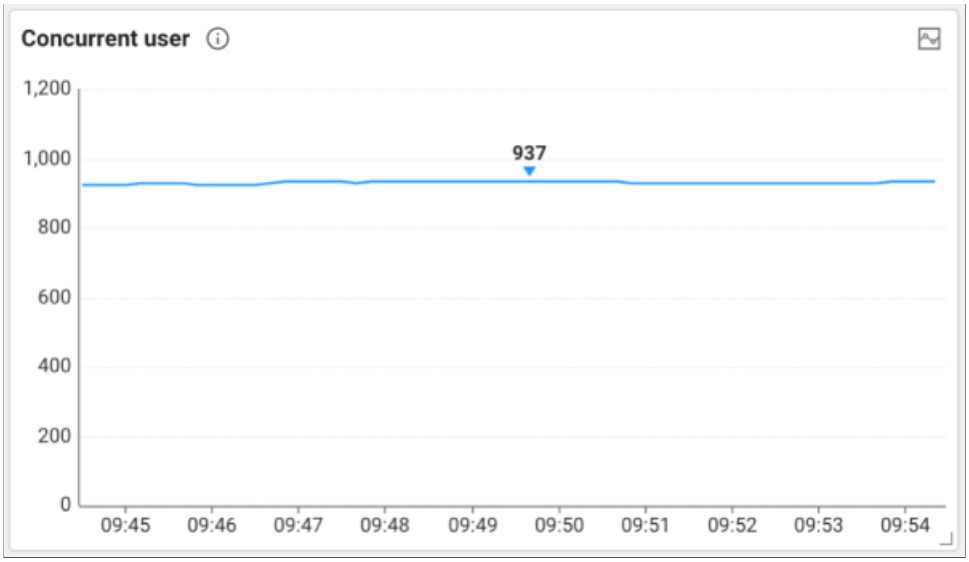



You can see which servers are at risk by displaying the maximum available memory and the current memory for each server. The changes in memory usage over time can be seen in real time.

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.

# User-related widget

## Concurrent user



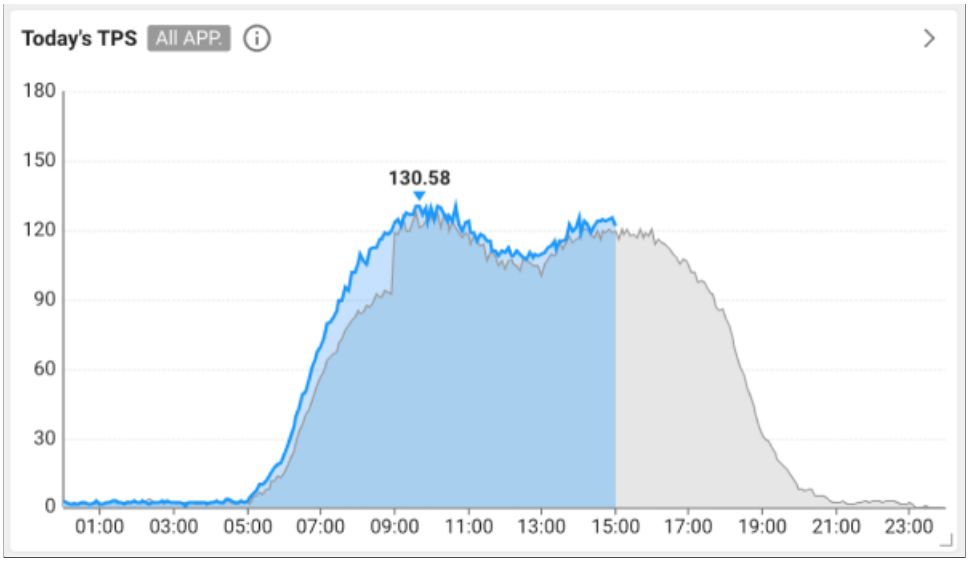

It displays the number of real-time browser users. Every 10 seconds, users who made transactions within the last 5 minutes are counted and displayed without duplicates. Users are counted based on their browser's IP.

- In 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

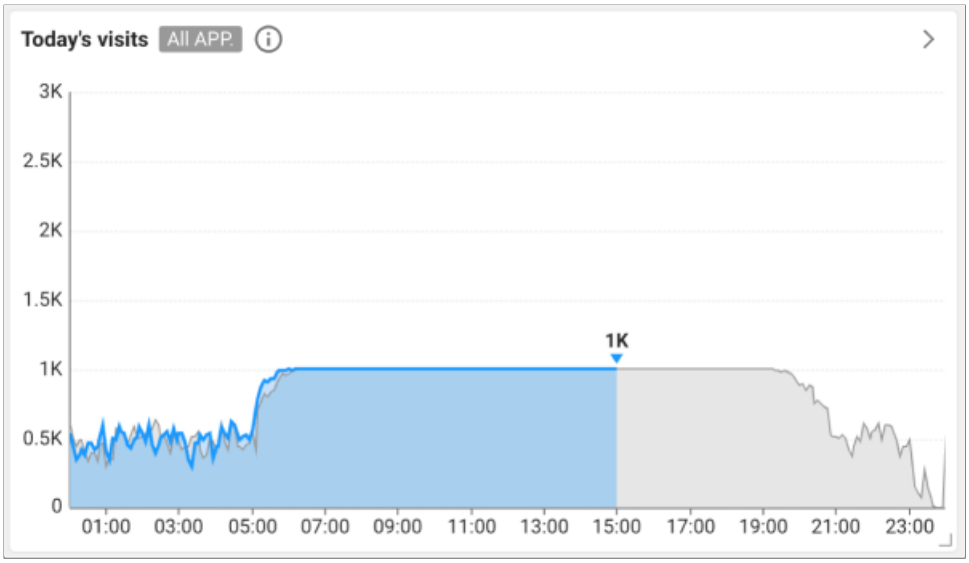
> The widgets marked with the `All APPS.` tag display the data for all agents.

### 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

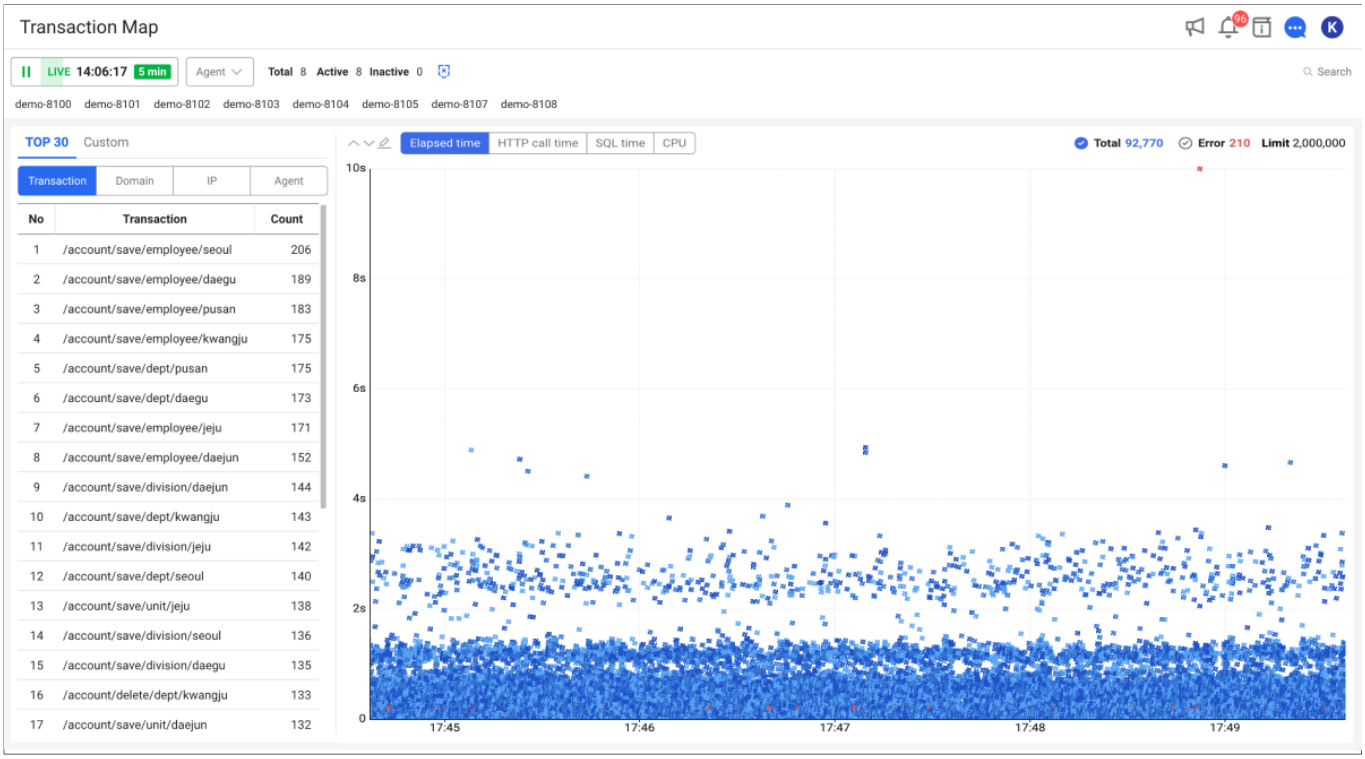
If you select **Alert Status** on the upper right of the screen, the recent events appear. You can turn on or off the browser notifications by clicking the toggle menu.

> 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.



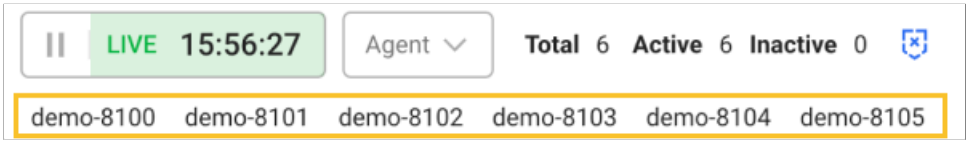
- You can perform filtering by setting multiple conditions. Added conditions are applied based on

&& . 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

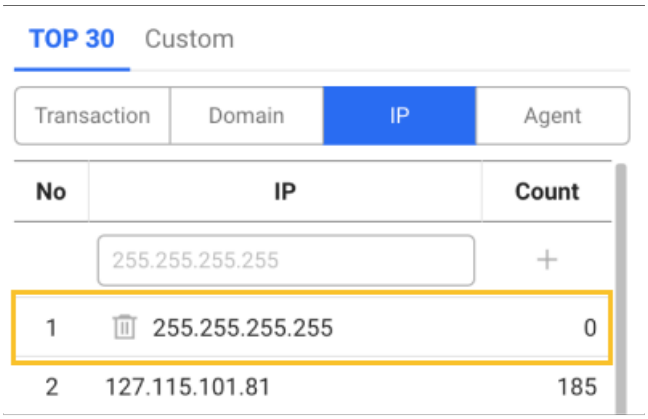
If you select each item from the list, the data is applied based on the item selected in the Transaction map chart. To select multiple items, hold down the `Ctrl` (Windows/Linux) or `CMD` (Mac) key and then click desired items.

> 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 ＋.



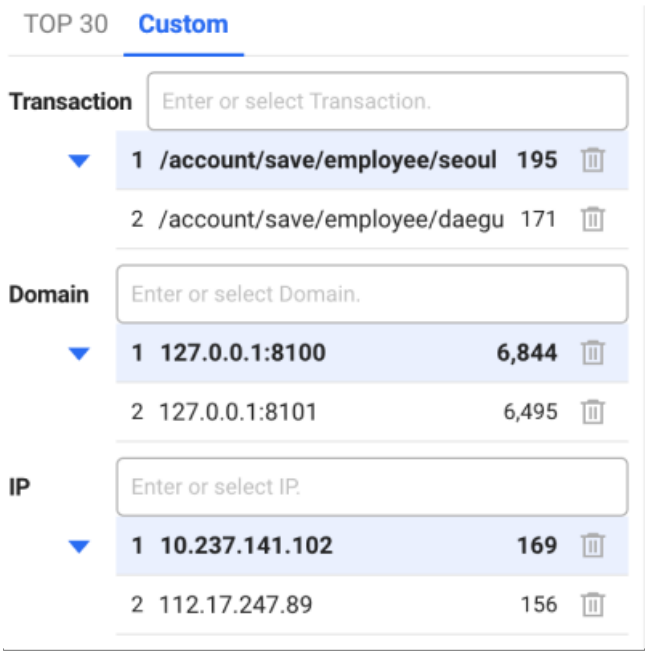

The specified IP address appears at the fixed top of the list.

> 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.

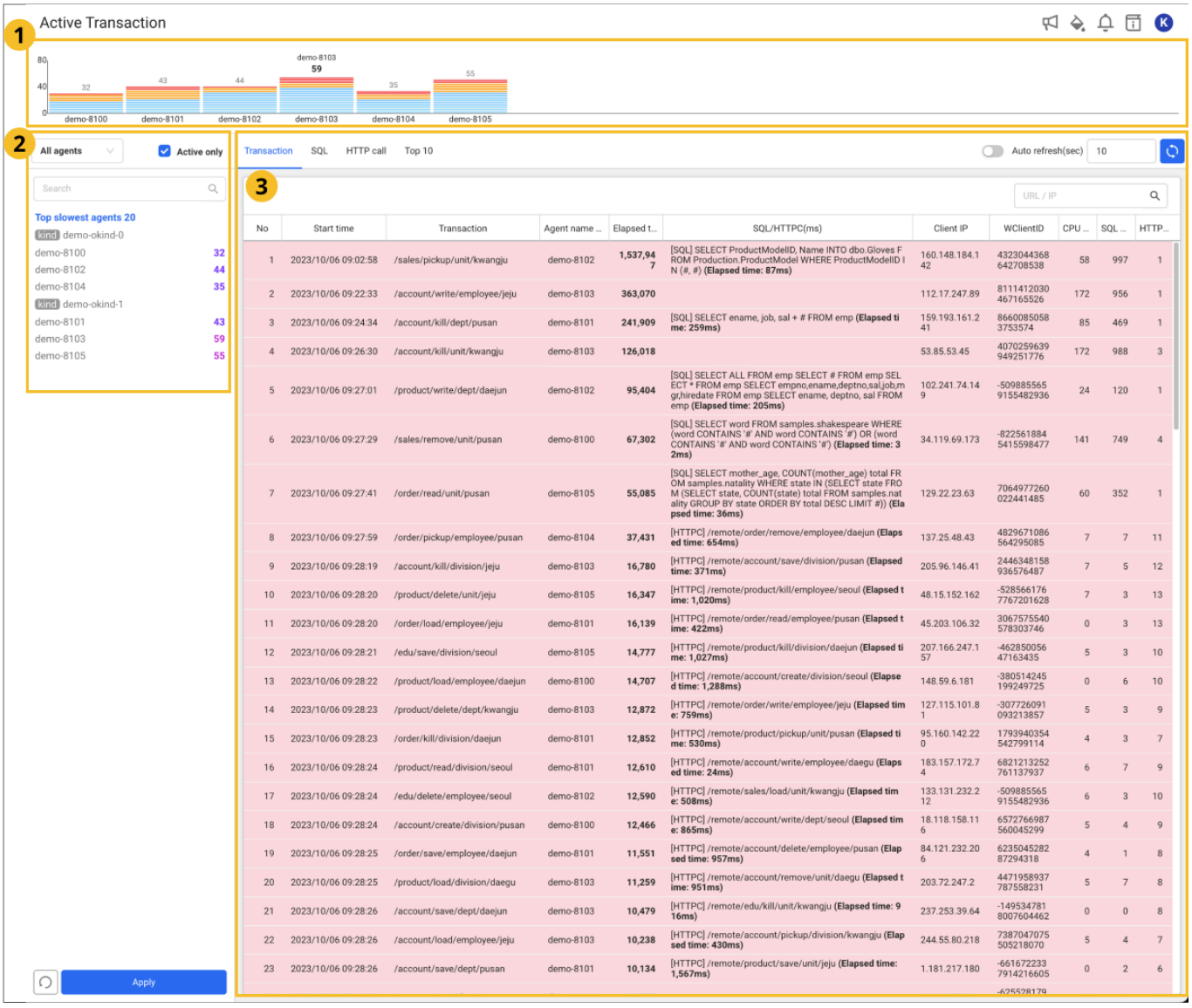
> To delete any added custom items, click 🗑.

# Active Transaction

In Home screen > Select Project > **Dashboard** > **Application Dashboard** > **Active Transaction** widget, select ›.

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.

# 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.

# 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

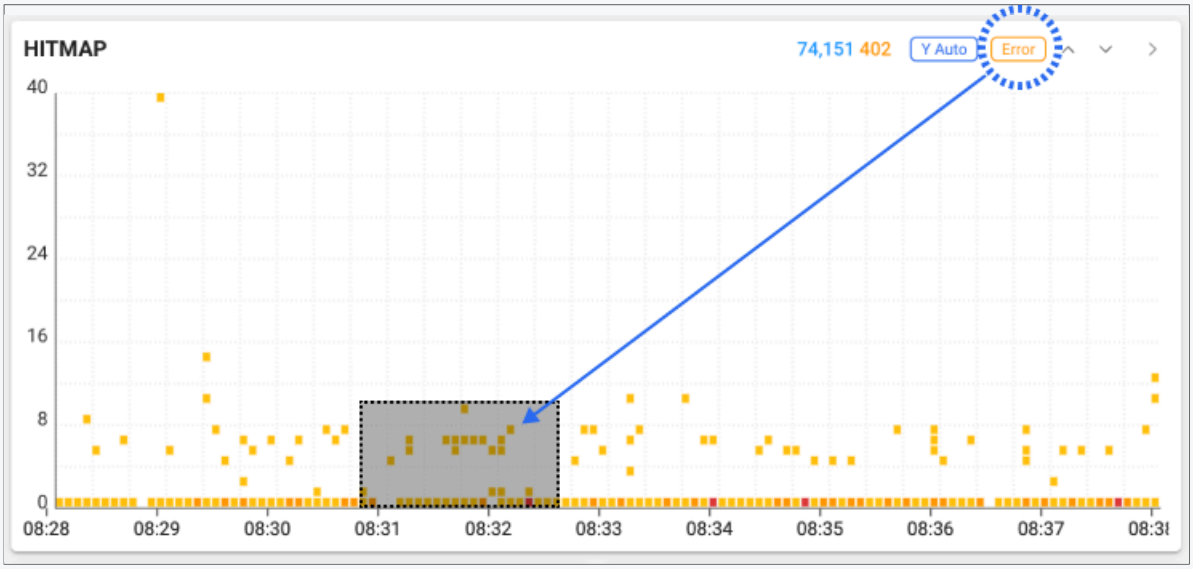
upper right and then press Enter or select 🔍. It filters with the transaction list that includes the input text.

To initialize the filtered list, delete text in the URL/IP field on the upper right, and then press Enter or select 🔍.

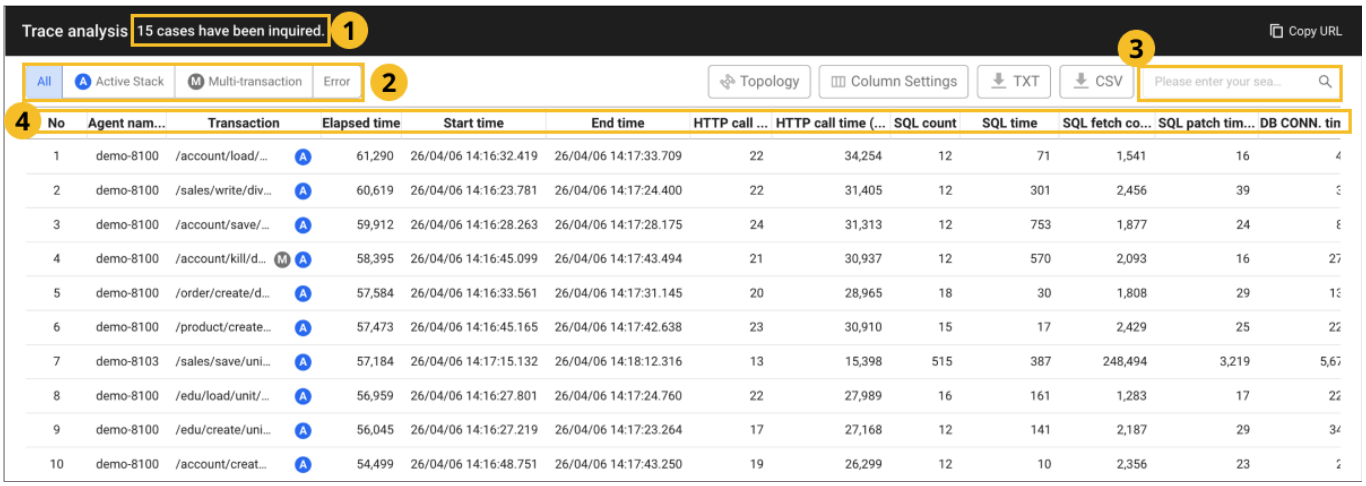
# HITMAP - Trace Analysis

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.

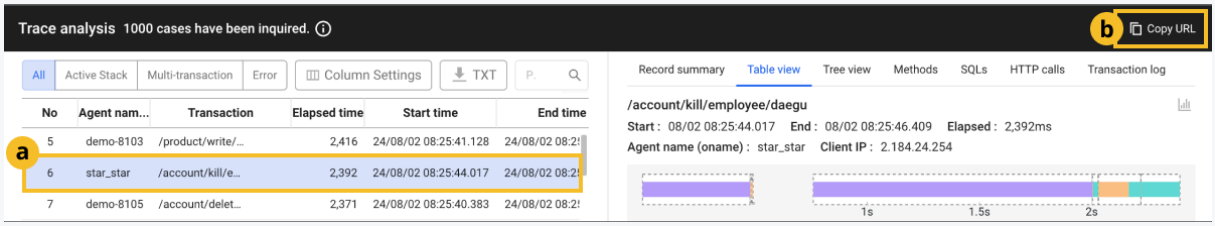


## Basic screen guide




- 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.

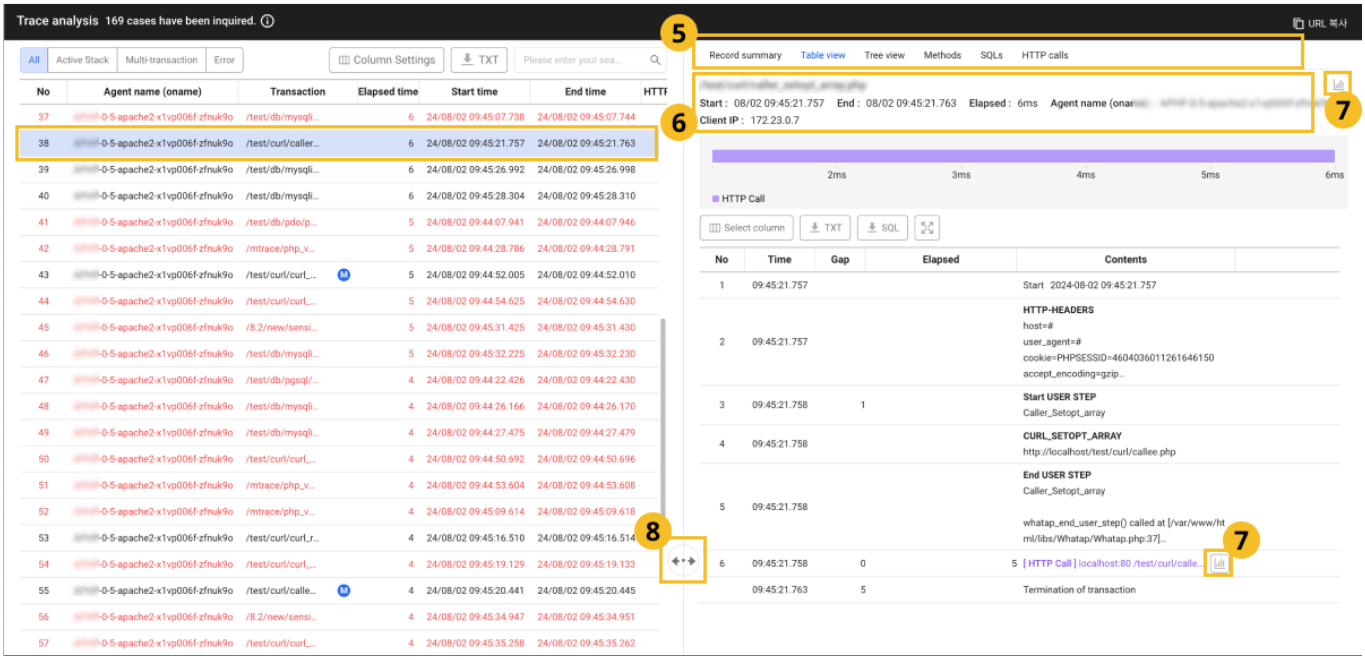
> - You can place, add, or hide desired columns to the front of the list. For more information, see the following.
> - If you drag another area in the **Hitmap** widget, a new trace analysis window appears. You can check it by comparing with the previous trace analysis window.
> - The transactions that 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.

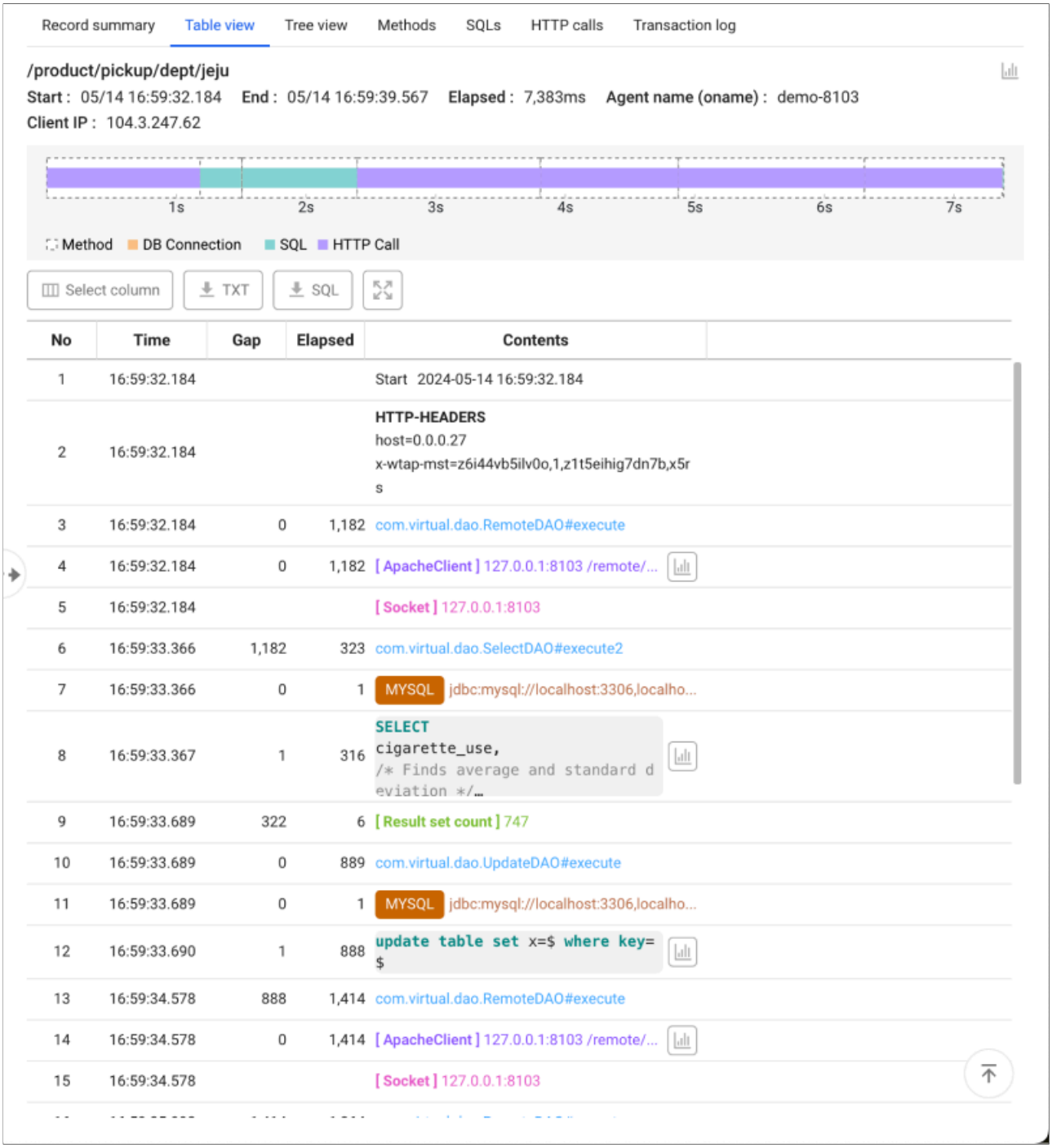
- Method step START-END

  The method response time is traced.

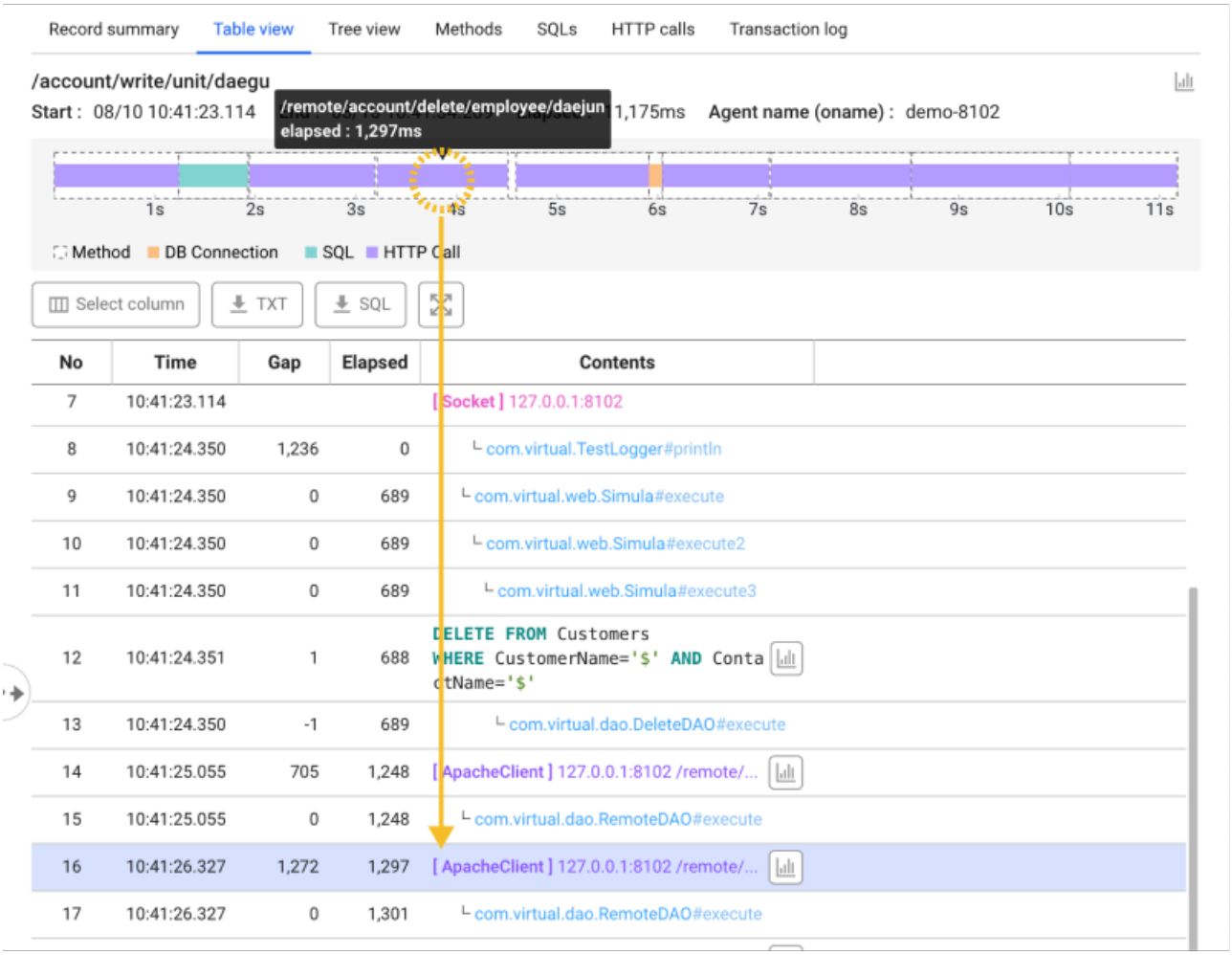
- In the WhaTap monitoring service, **Step** is used as the same meaning as **Span**.
- To see the updates in the **Trace analysis** function released on July 25, 2023, see the following.

## 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.



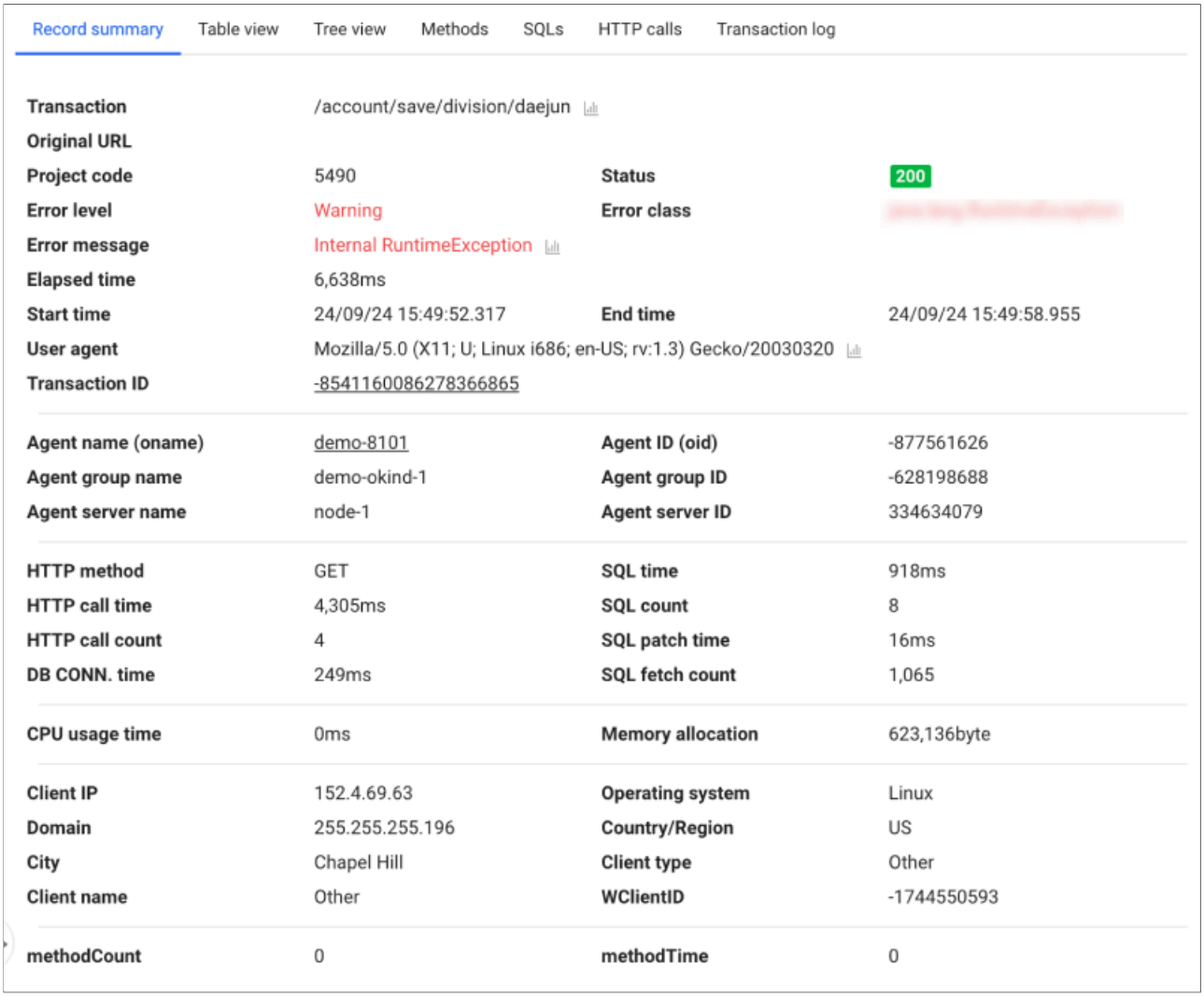

- **Select column**: You can add or hide the memory accumulation and CPU accumulation data as columns in the table list. The columns are as follows:
  - **No**: Step occurrence sequence
  - **Time**: Start time of each step
  - **Gap**: The waited time between the start time of the previous step and 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.

## 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. |
| | 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. |

| Classification | Properties | Description |
|---|---|---|
| | 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 |
| | Agent ID (oid) | It is a unique ID of the agent. |
| | Agent group name | Agent group name set with the `okind` option. |
| | Agent group | It is a unique ID of the agent group. |

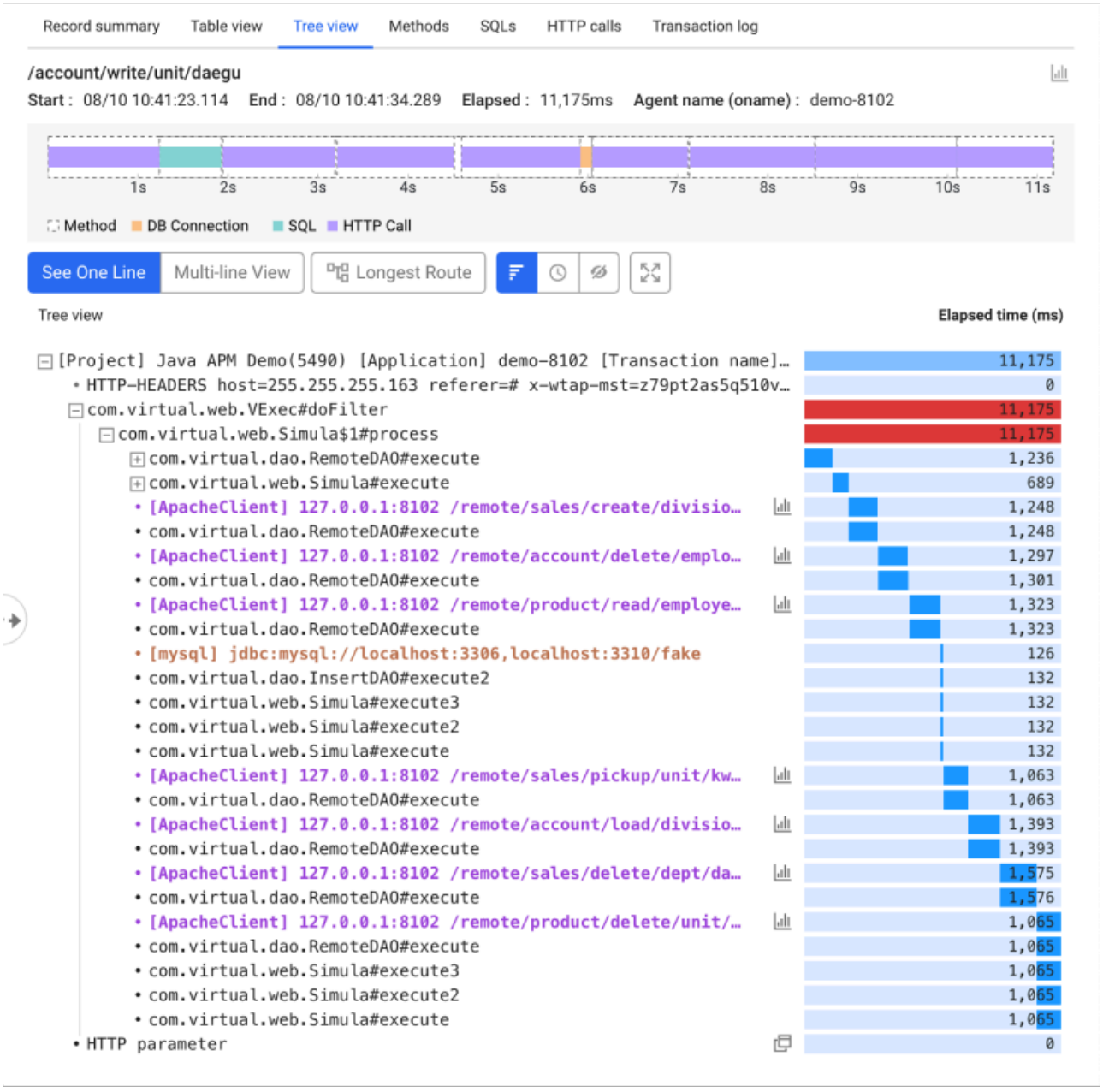
| Classification | Properties | Description |
|---|---|---|
| | ID | |
| | Agent server name | Agent server name set with the `ondoe` option. |
| | Agent server ID | It is a unique ID of the agent server. |
| **Step** | HTTP method | HTTP method. (GET, POST, PUT, HEAD, etc.) |
| | HTTP call time | Outbound 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 time | CPU time used for processing transactions. |
| | Memory allocation | Amount of memory used by the transaction. |
| **Containers** | Pod name | It is the name of the Kubernetes Pod that processed the transaction. |

| Classification | Properties | Description |
|---|---|---|
| | 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.



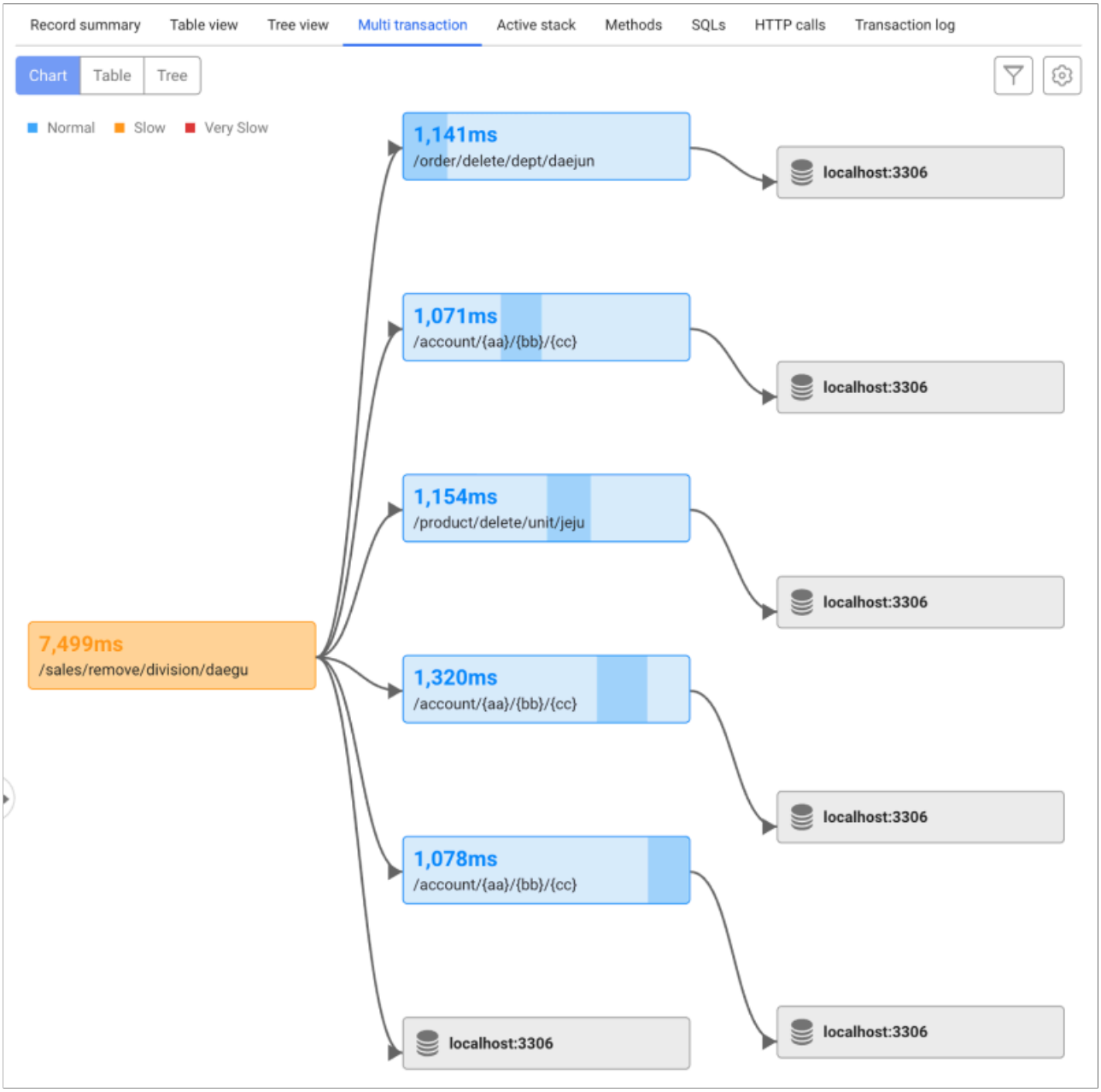
- **Single-line View**: You can arrange the tree format at intervals by displaying the texts displayed in the execution data for each section in a single line.
- **Multi-line View**: All the texts displayed in the execution data for each section, appear by wrapping words.
- **Longest Path**: The longest 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.

## 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.



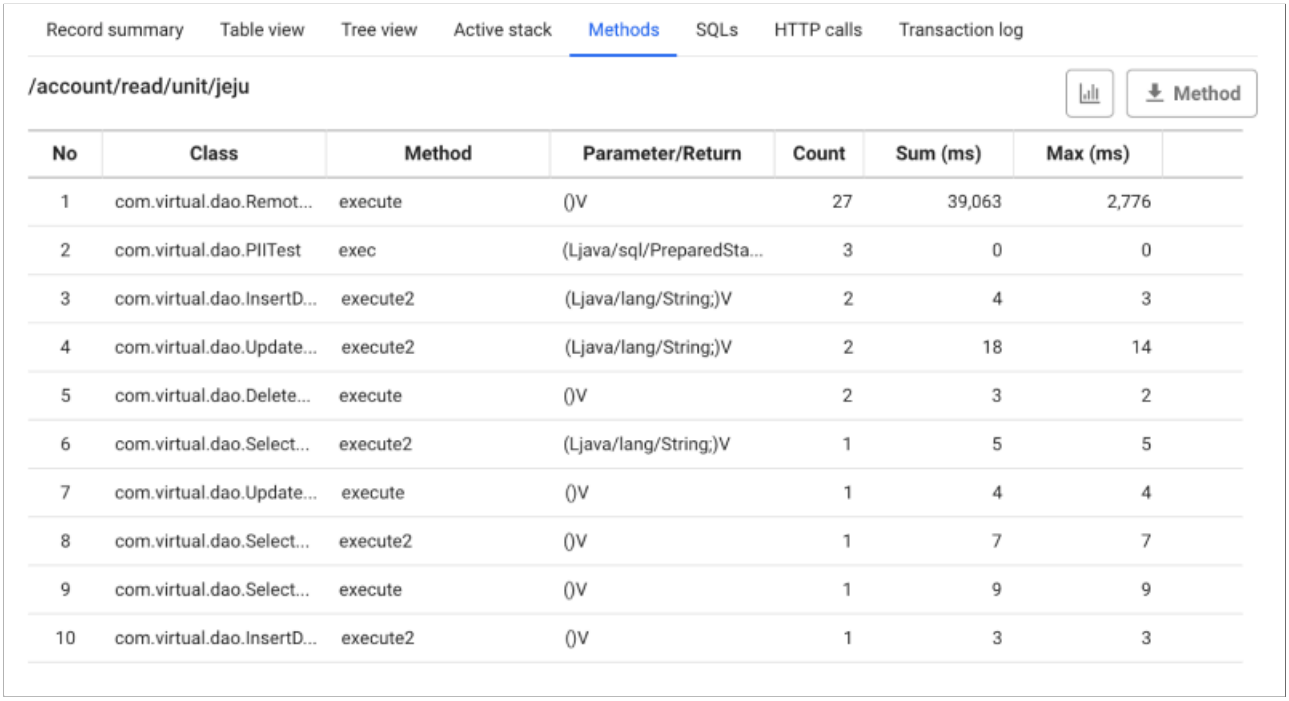

- **Chart**: Provides the call relationships for each transaction in a flowchart. Select a transaction node to see the trace analysis for the transaction node. In **Chart** mode, you can use the mouse to go to a desired position or zoom in or out by scrolling.
  - ⚙ **Chart view settings**: You can display 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 |

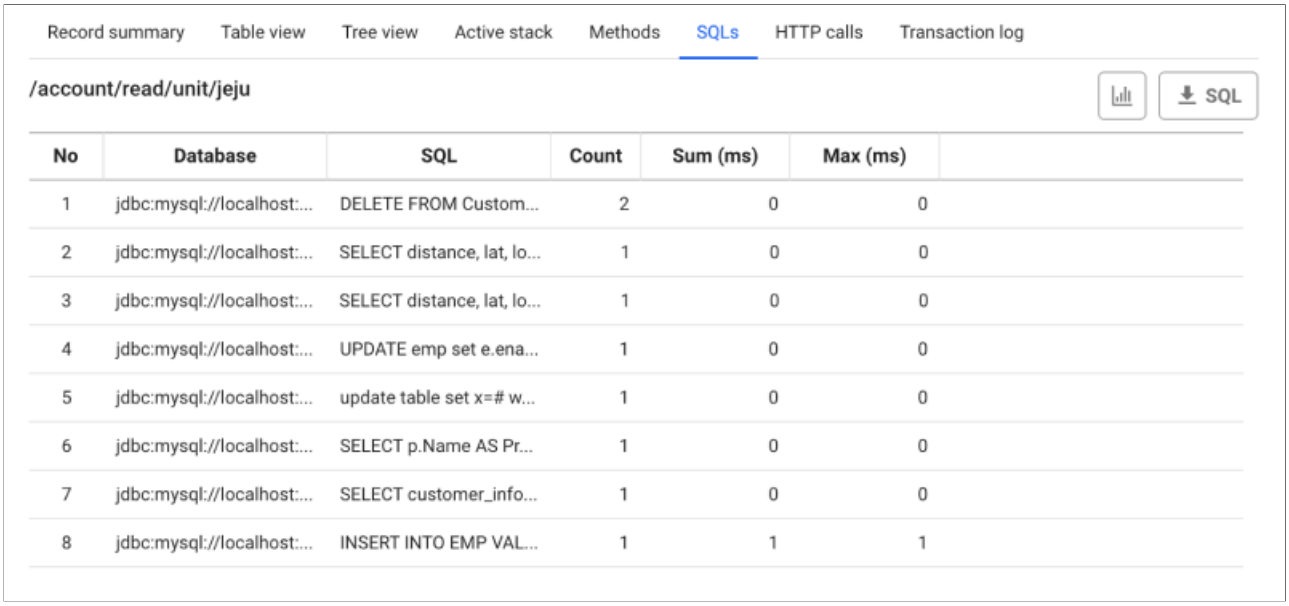
To see only the method data, select the **Methods** tab. It displays the method name and step time for which tracing has been set on the agent. When unnecessary logics are repeatedly executed, it can be used as analysis data to improve the method logic by finding the case where the execution takes longer 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

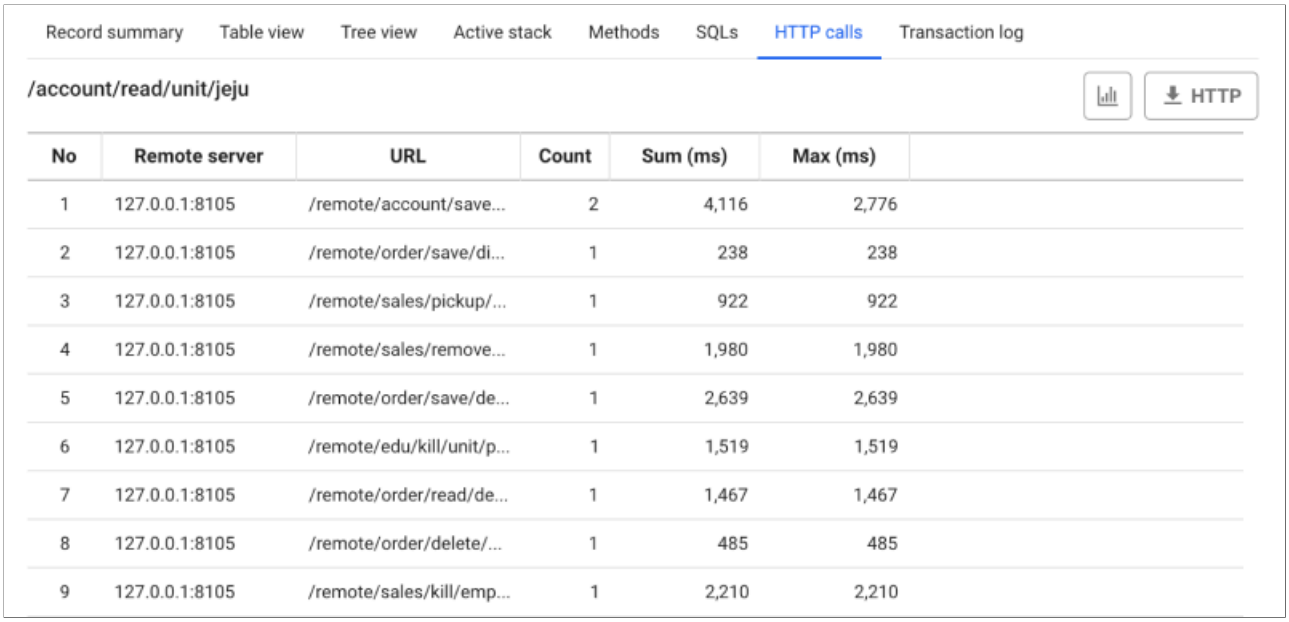



| No | Database | SQL | Count | Sum (ms) | Max (ms) |
|---|---|---|---|---|---|
| 1 | jdbc:mysql://localhost:... | DELETE FROM Custom... | 2 | 0 | 0 |
| 2 | jdbc:mysql://localhost:... | SELECT distance, lat, lo... | 1 | 0 | 0 |
| 3 | jdbc:mysql://localhost:... | SELECT distance, lat, lo... | 1 | 0 | 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



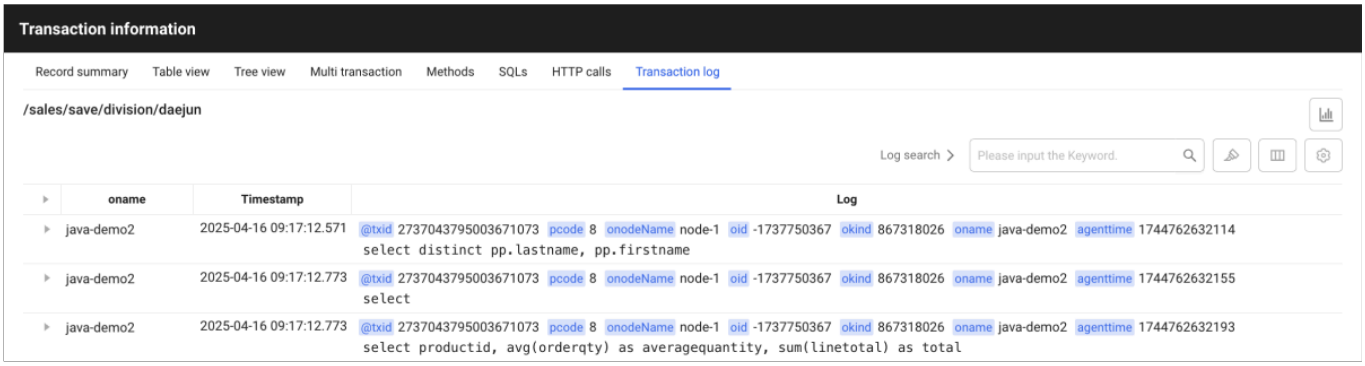

| No | Remote server | URL | Count | Sum (ms) | Max (ms) |
|---|---|---|---|---|---|
| 1 | 127.0.0.1:8105 | /remote/account/save... | 2 | 4,116 | 2,776 |
| 2 | 127.0.0.1:8105 | /remote/order/save/di... | 1 | 238 | 238 |
| 3 | 127.0.0.1:8105 | /remote/sales/pickup/... | 1 | 922 | 922 |
| 4 | 127.0.0.1:8105 | /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.

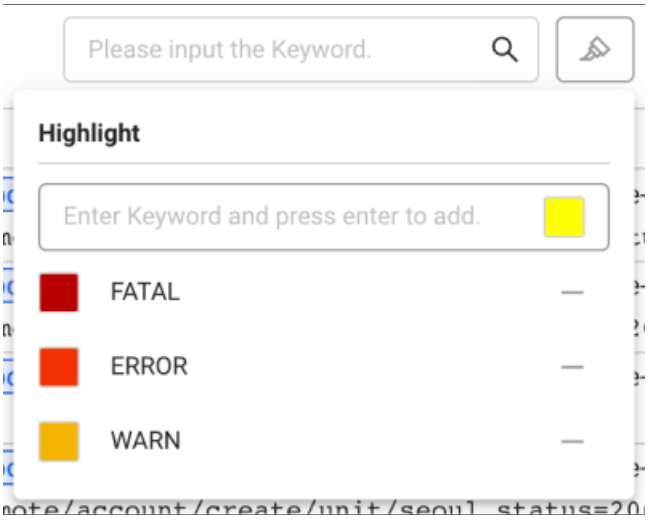



- : You can see the **Transaction statistics** information. If a graph on the chart is selected, the **Analysis** > **Transaction Search** menu appears. You can see the list of transactions for the selected time zone.
- **Log Search**: You can go to Log Search to easily find the desired log data.
- **Keyword 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.

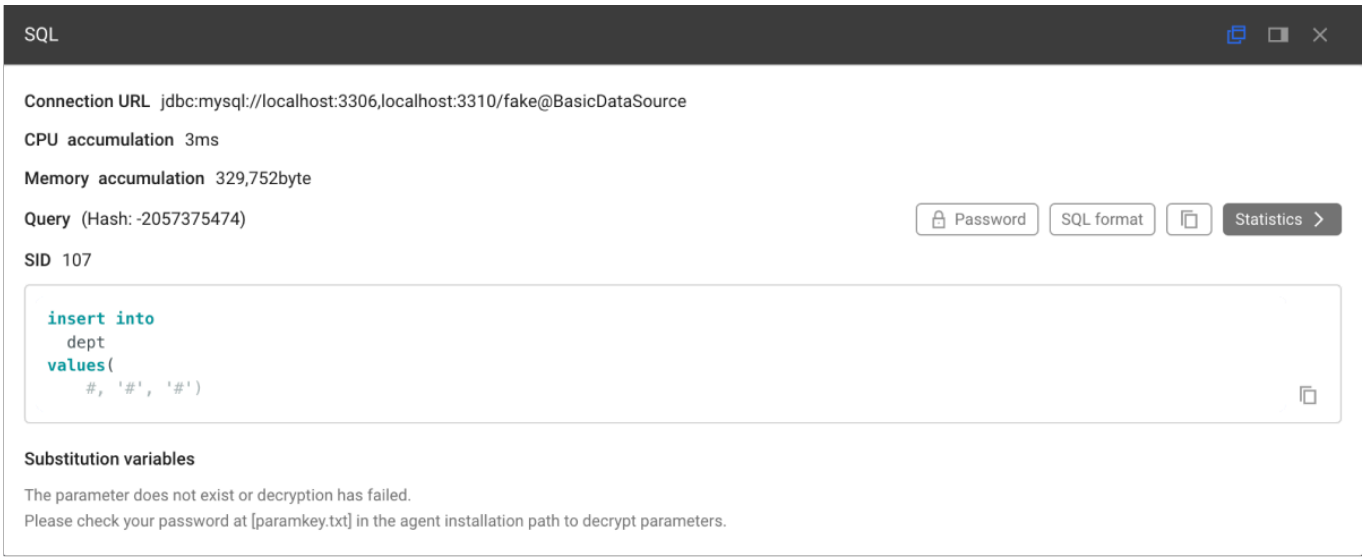
1. Scroll down the page to the final step of the transaction.
2. Select the 🔒 **HTTP parameters** item.
3. If the **HTTP-PARAMETERS** window appears, select **Password**.
4. Enter the specified Param Key. You can see the hidden parameters.

> ⓘ
> - 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 ( `paramkey` ) in the `WHATAP_HOME` /security.conf file. You can change it to another string. It is required to view the SQL variables and HTTP queries and to stop the threads.
> - Depending on the agent version, the password configuration file may be different. Change the

> string in the security.conf file or paramkey.txt file.

## 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.

  > Depending on the agent version, the password configuration file may be different. Change the string in the security.conf file or paramkey.txt file.

- **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.

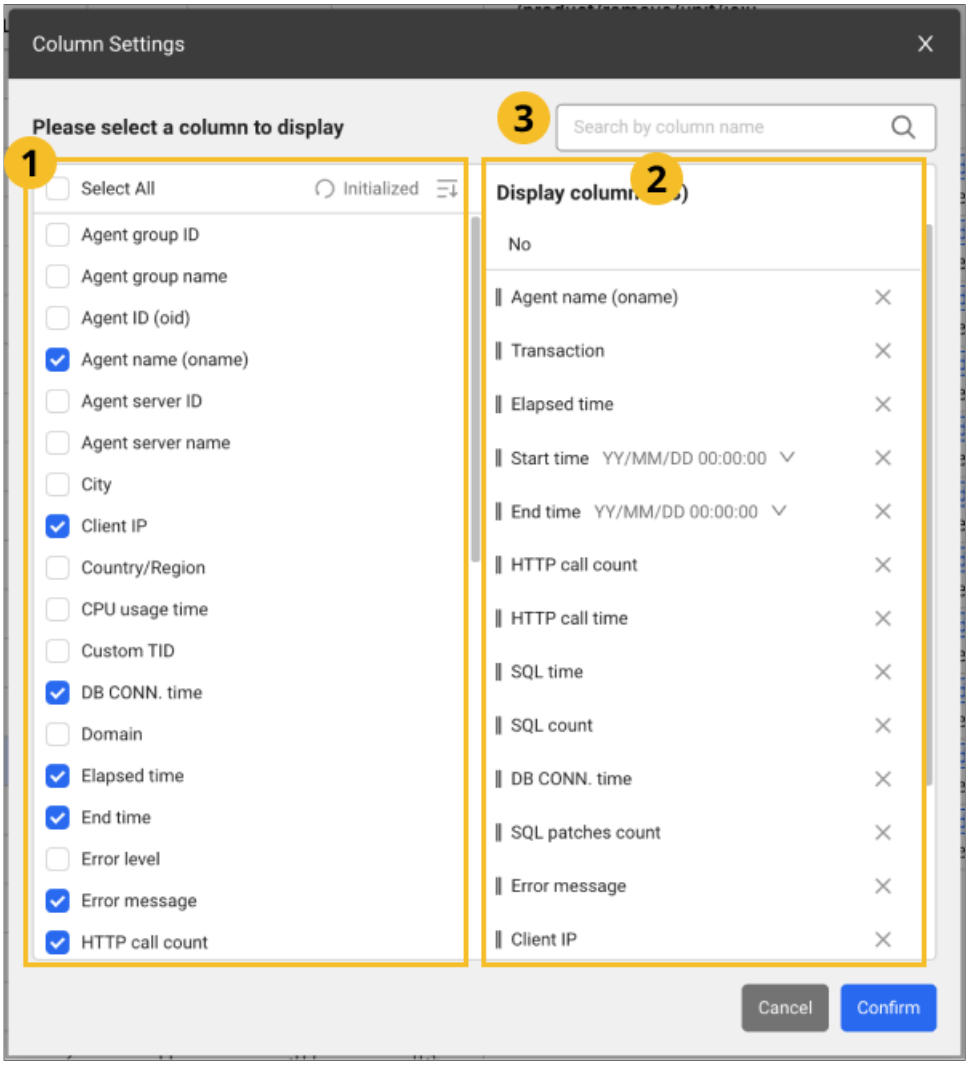
whatap.conf

```
# 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
```

## 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**.

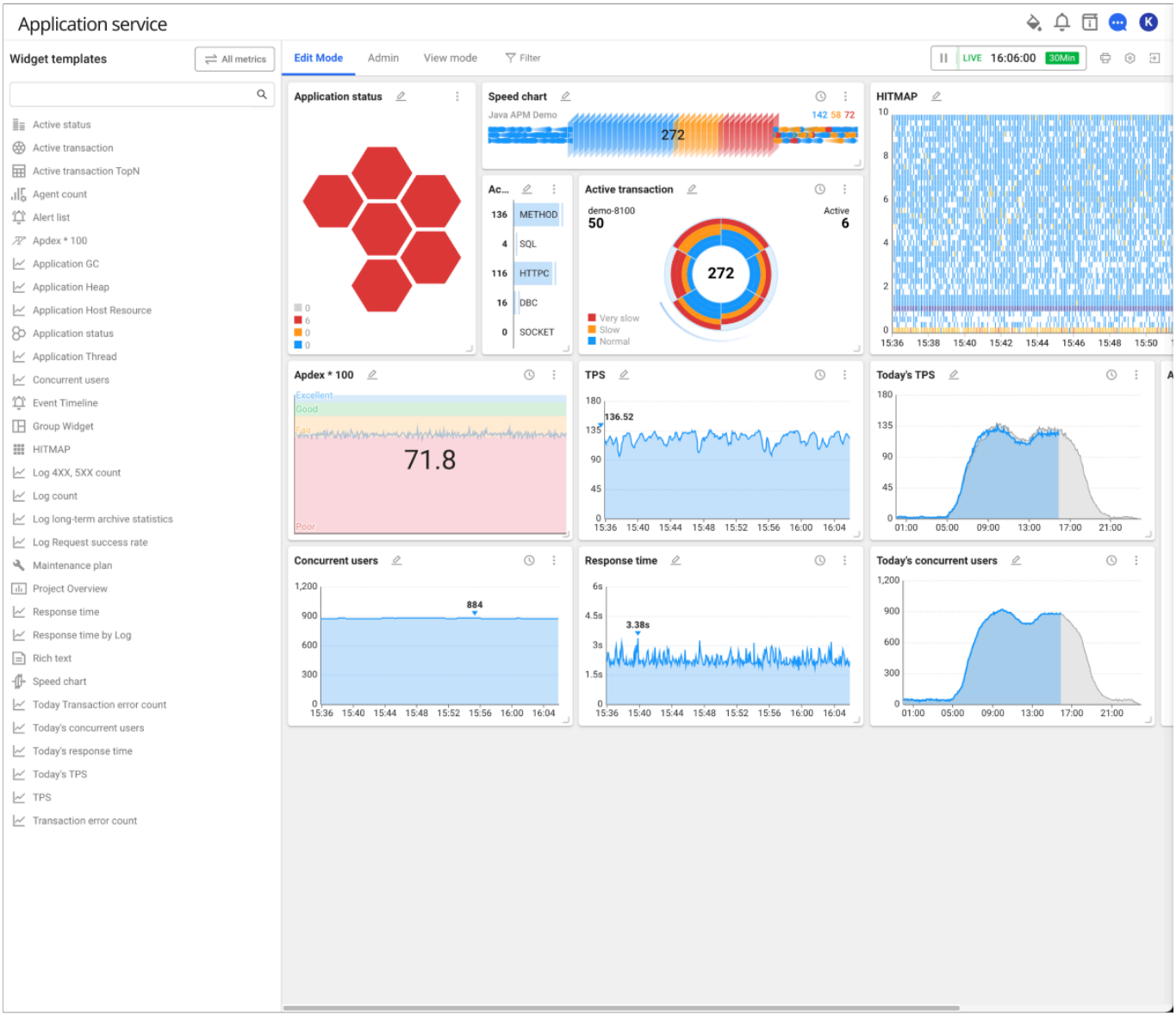
## Downloading the viewed data

To download the data output in the table in .txt file format, select TXT.

# 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.

# Using 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.

## 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.

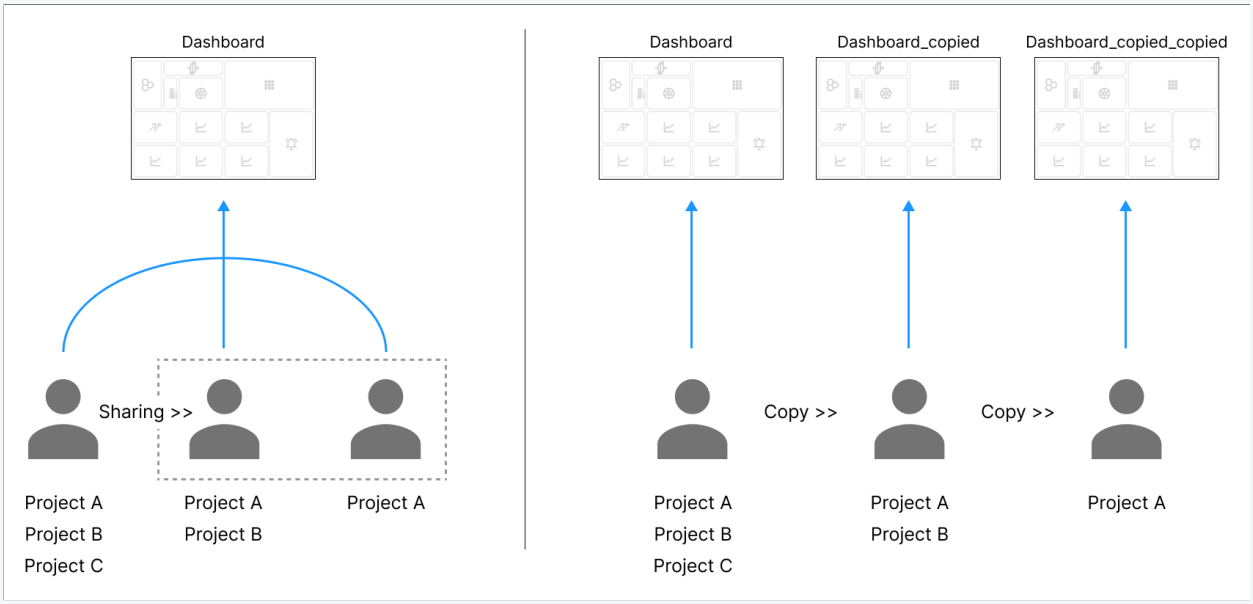
> Dashboard deletion behavior differs for owners and shared users.
>
> - If the dashboard owner deletes a dashboard, it is deleted for all user accounts with whom it was shared.

- 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**



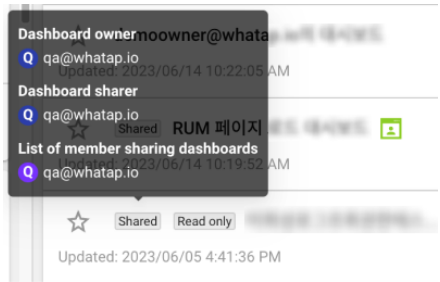
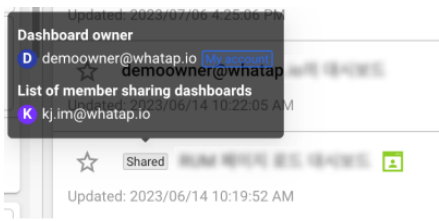

- Share: all members you share with view the same dashboard.
- Copy: if you copy a dashboard to another member or to yourself, you view a separate copied dashboard.
- Dashboards shared or copied as **Read only** cannot be edited. They can be edited only if shared as **Edit mode**.
- Members who receive 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 ★.

# Managing The 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**

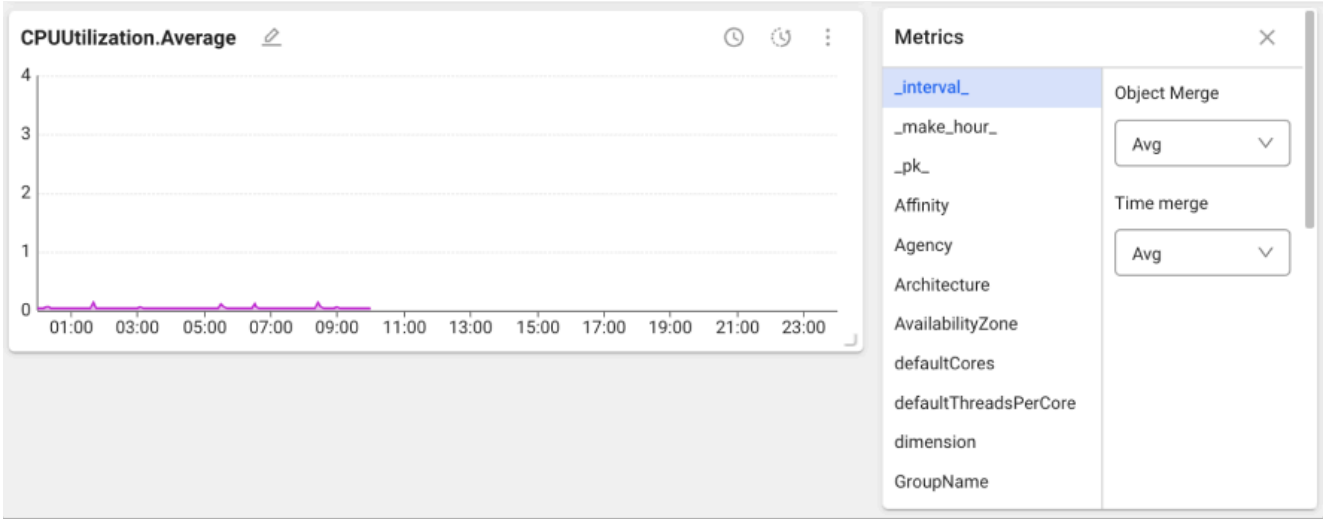
  It is used to group the chart data based on the tag (e.g. agent name) selected on the left. At this time, if the tags match among the data with different fields (e.g. CPU), the corresponding data is merged.

- **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

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.

## 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**

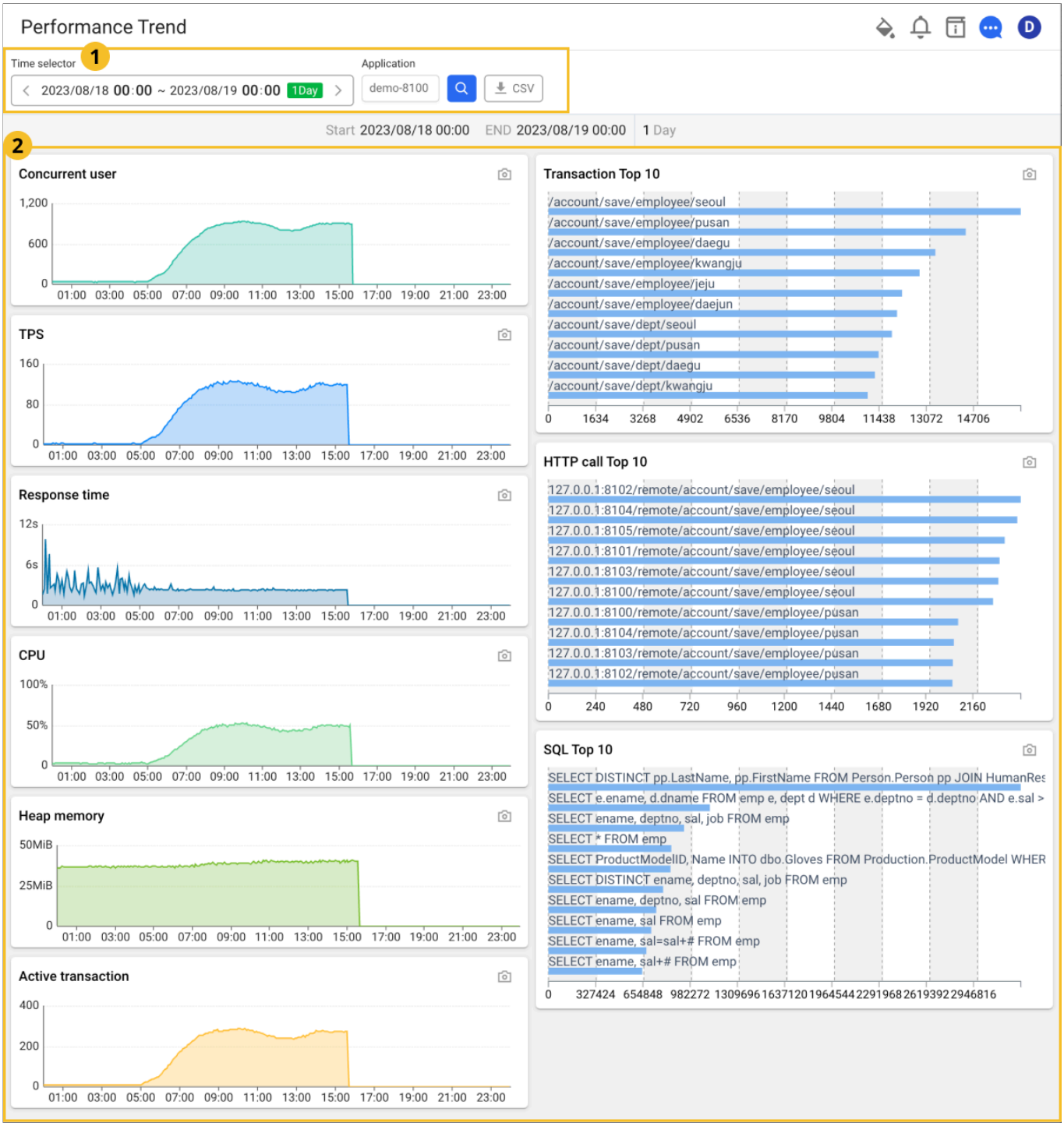
> ⓘ **Today's visits** is the sum of users accumulated for the day without duplication. Instead of simply adding up, it is calculated using **HyperLogLog**. **HyperLogLog** is a probabilistic data structure that estimates the number of elements in a set with very little memory.

- **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.

### Memory

It indicates the **memory** usage of the monitored process.

### 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

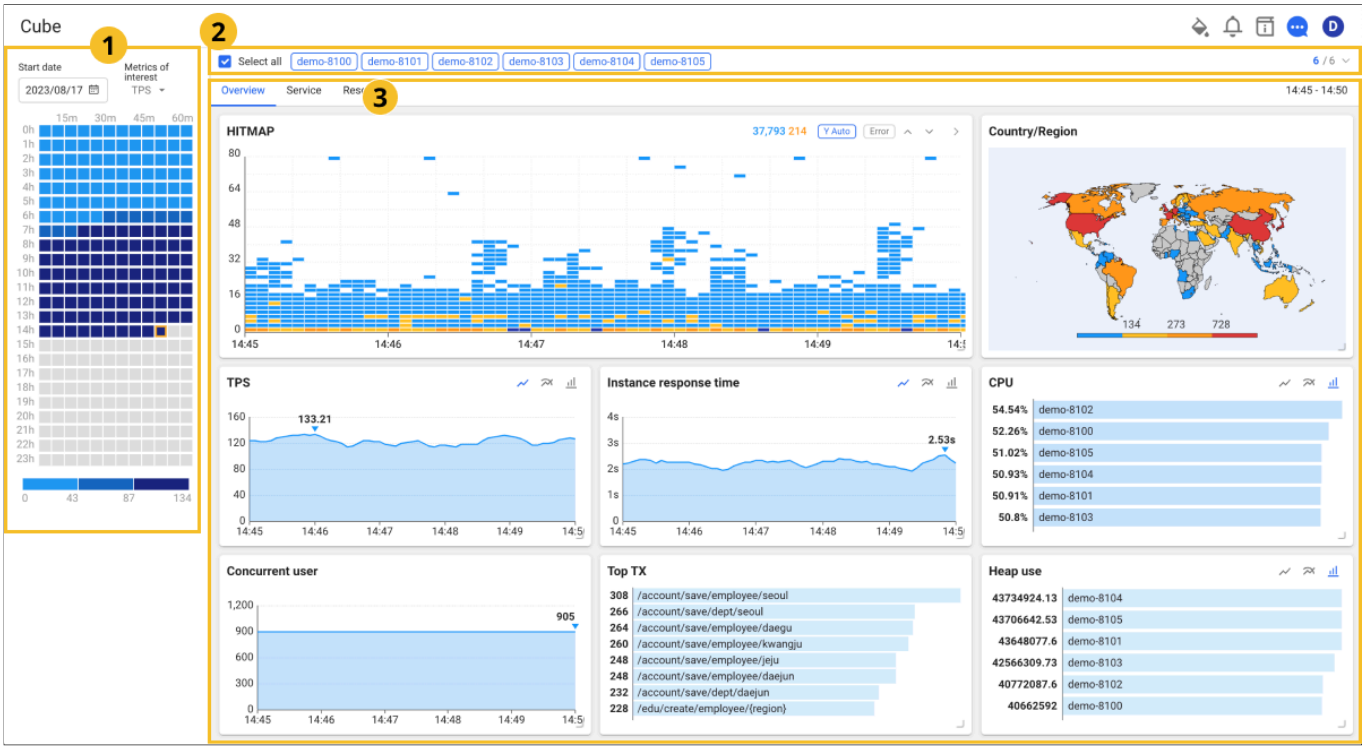
Home > Select Project > **Analysis** > **Cube**

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

## 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.

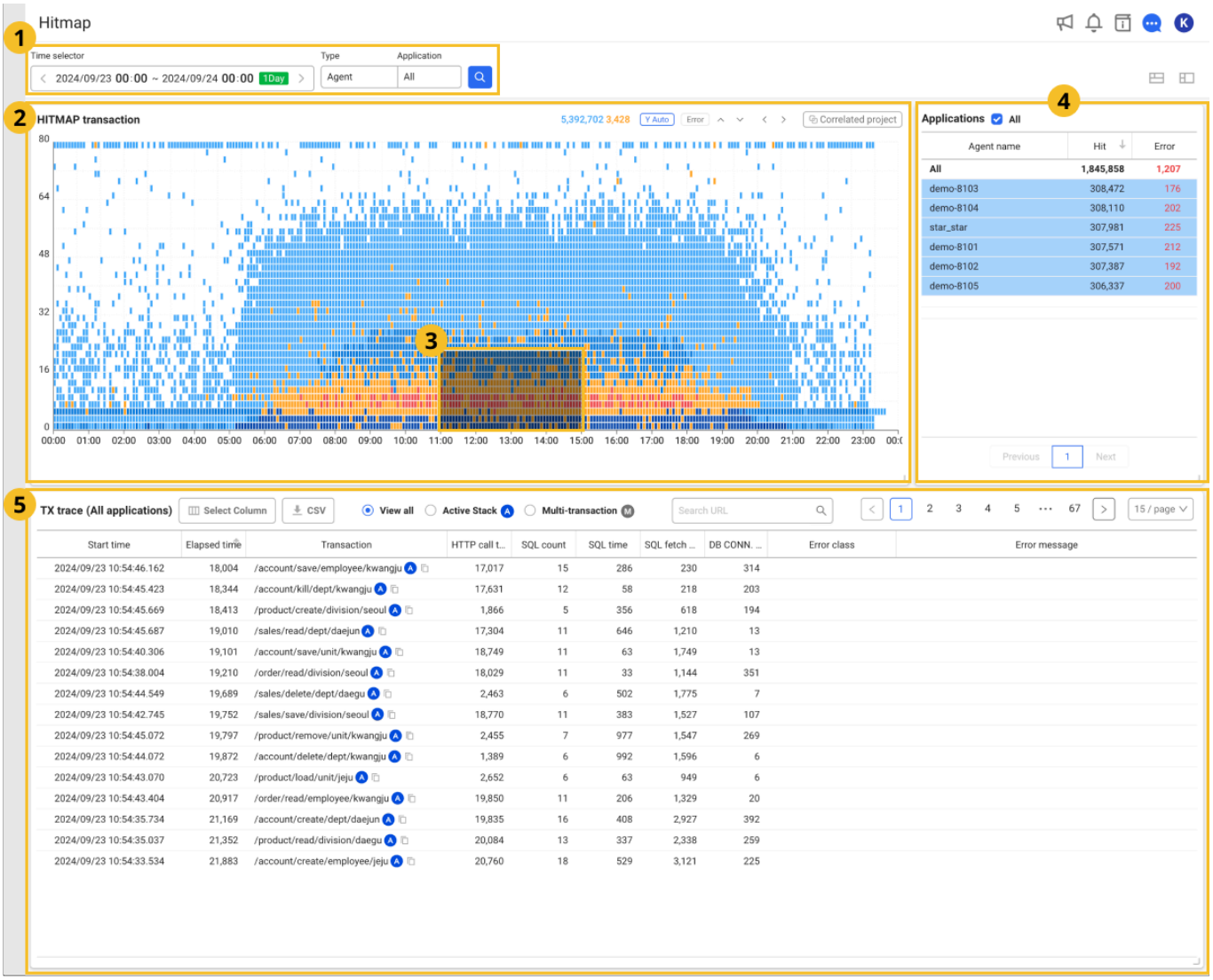
- **ACTIVE STACK Step** START-END

  It includes the active stack information. It is created and collected by a separate thread and it is added to the transaction trace.

> In the WhaTap monitoring service, **Step** is used as the same meaning as **Span**.

# Detailed analysis

Home > Select Project > **Analysis** > **Hitmap**



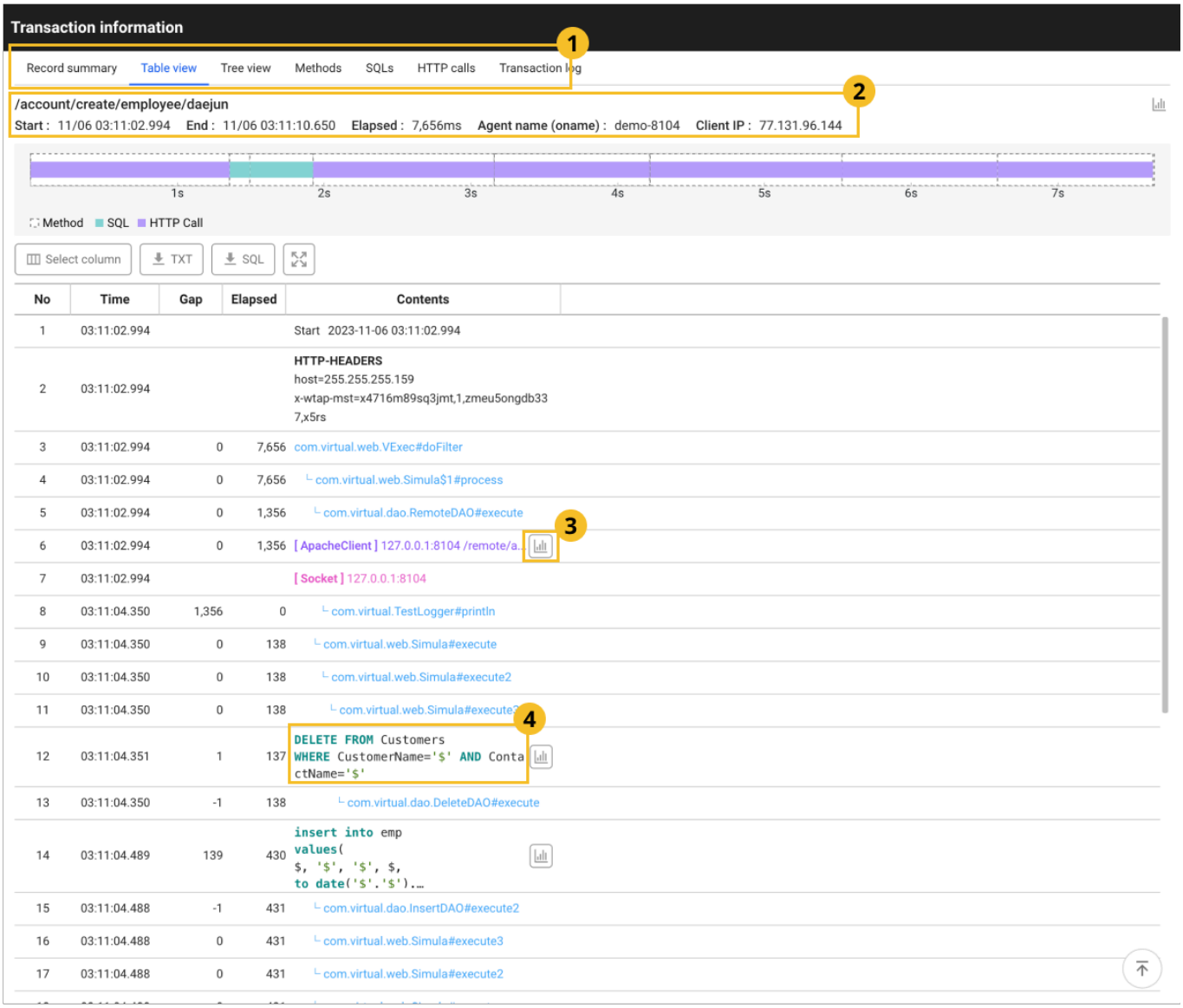

1. In 1 **Time selector**, select a time period to see the hitmap.
2. In 1 **Type** and **Application**, select the targets to view.

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



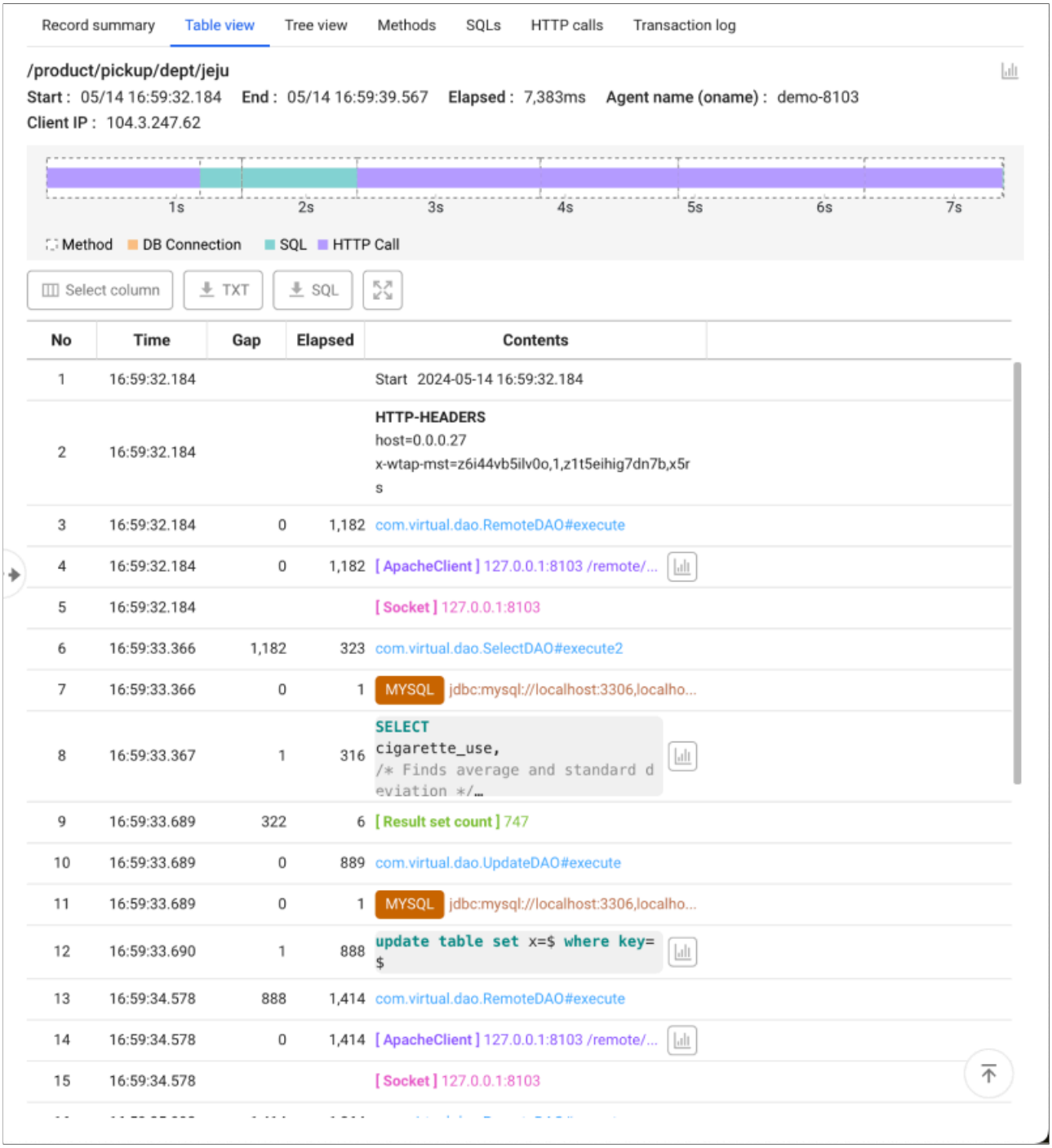

- In the 1 area, the step data can be seen in various formats. Select a desired type of tab.
- In the 2 area, you can see the basic information for transactions.
- 3 : The statistical data window for the corresponding URL or each execution section appears. Click a desired time on the graph chart in the 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.

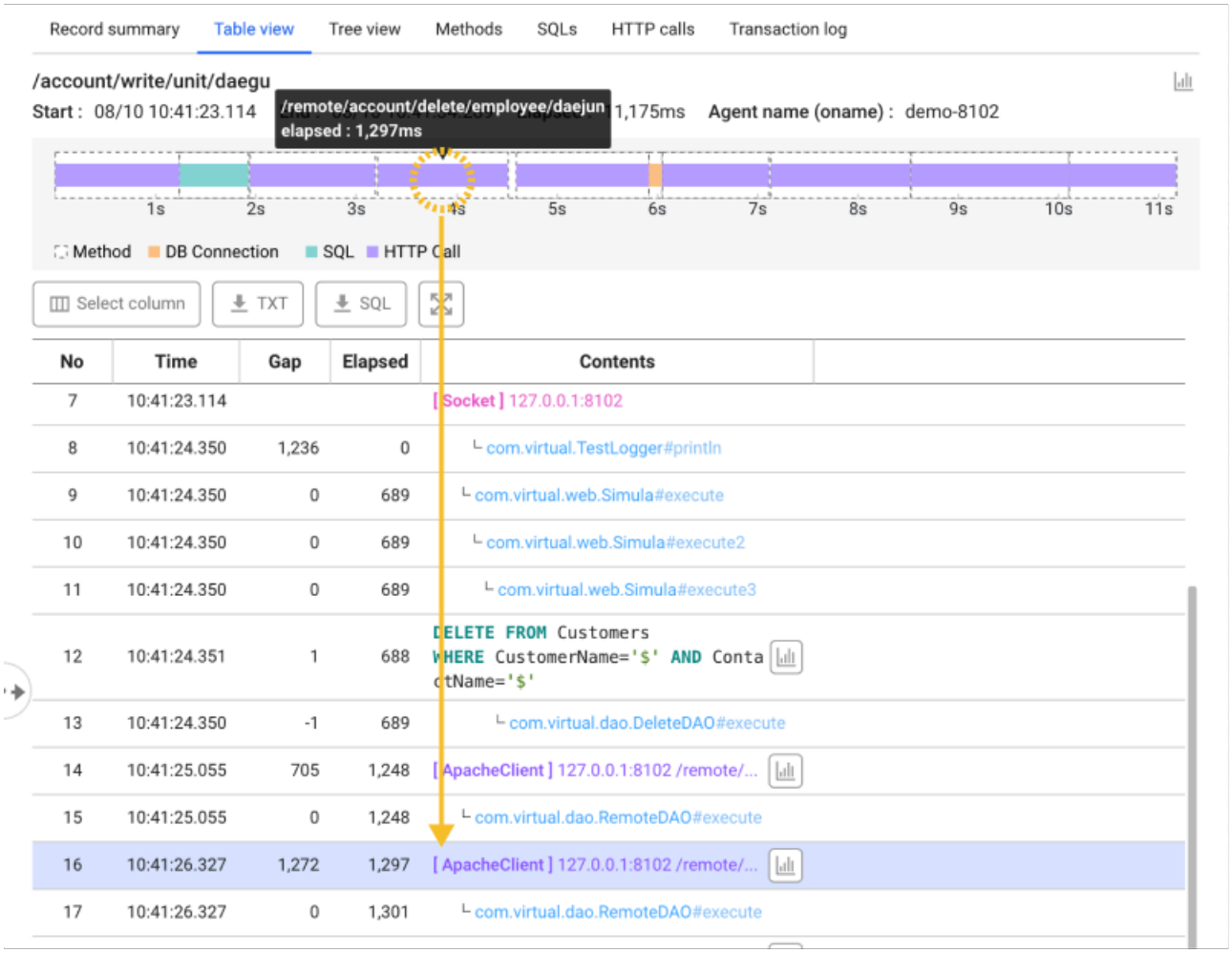
- If you select 4 SQL step, the SQL variables and HTTP queries can be viewed. For more information, see the following.

## 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.



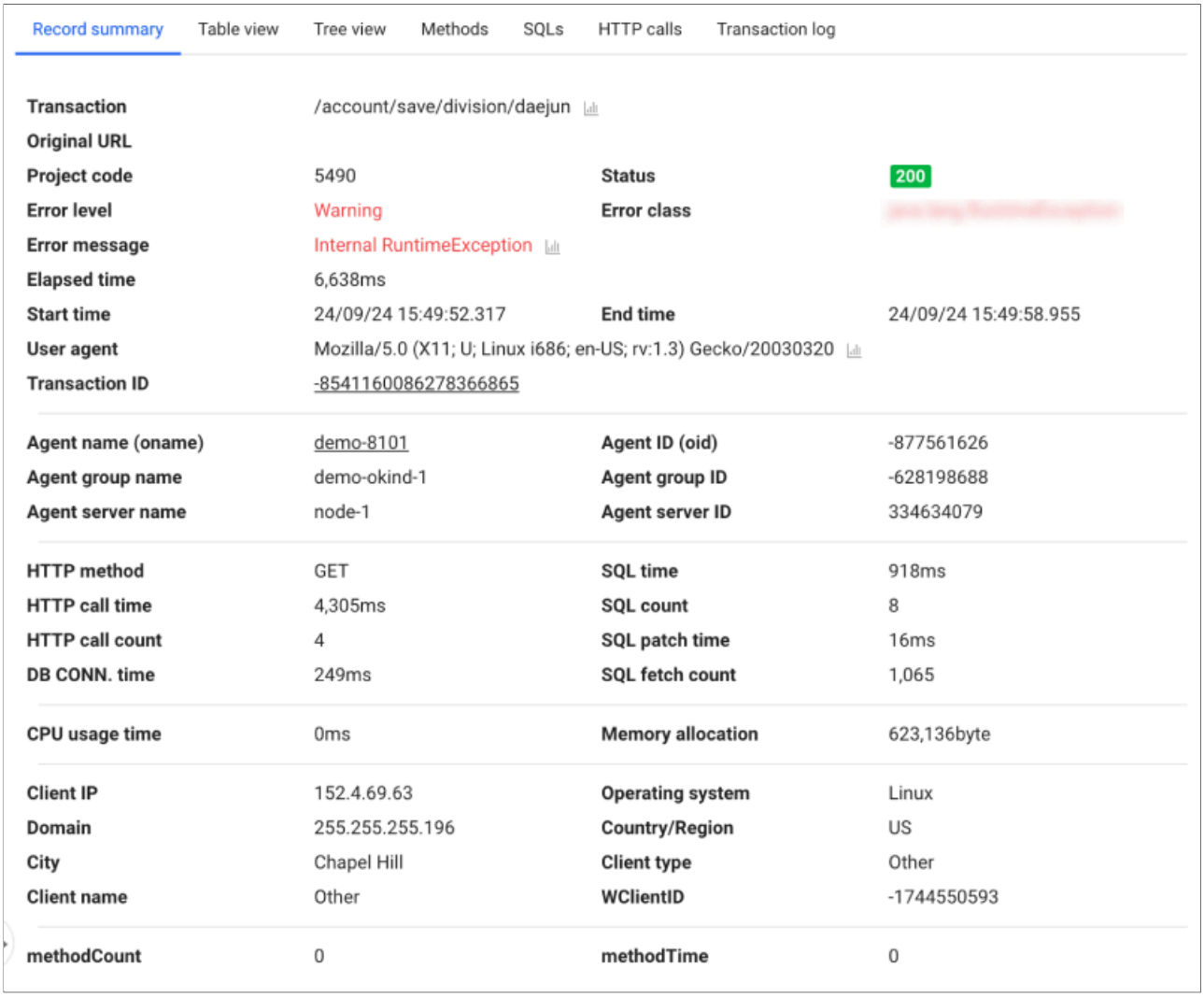

- **Select column**: You can add or hide the memory accumulation and CPU accumulation data as columns in the table list. The columns are as follows:
  - **No**: Step occurrence sequence
  - **Time**: Start time of each step
  - **Gap**: The waited time between the start time of the previous step and 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.

## 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. |
| | 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. |

| Classification | Properties | Description |
|---|---|---|
| | 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 |
| | Agent ID (oid) | It is a unique ID of the agent. |
| | Agent group name | Agent group name set with the `okind` option. |
| | Agent group | It is a unique ID of the agent group. |

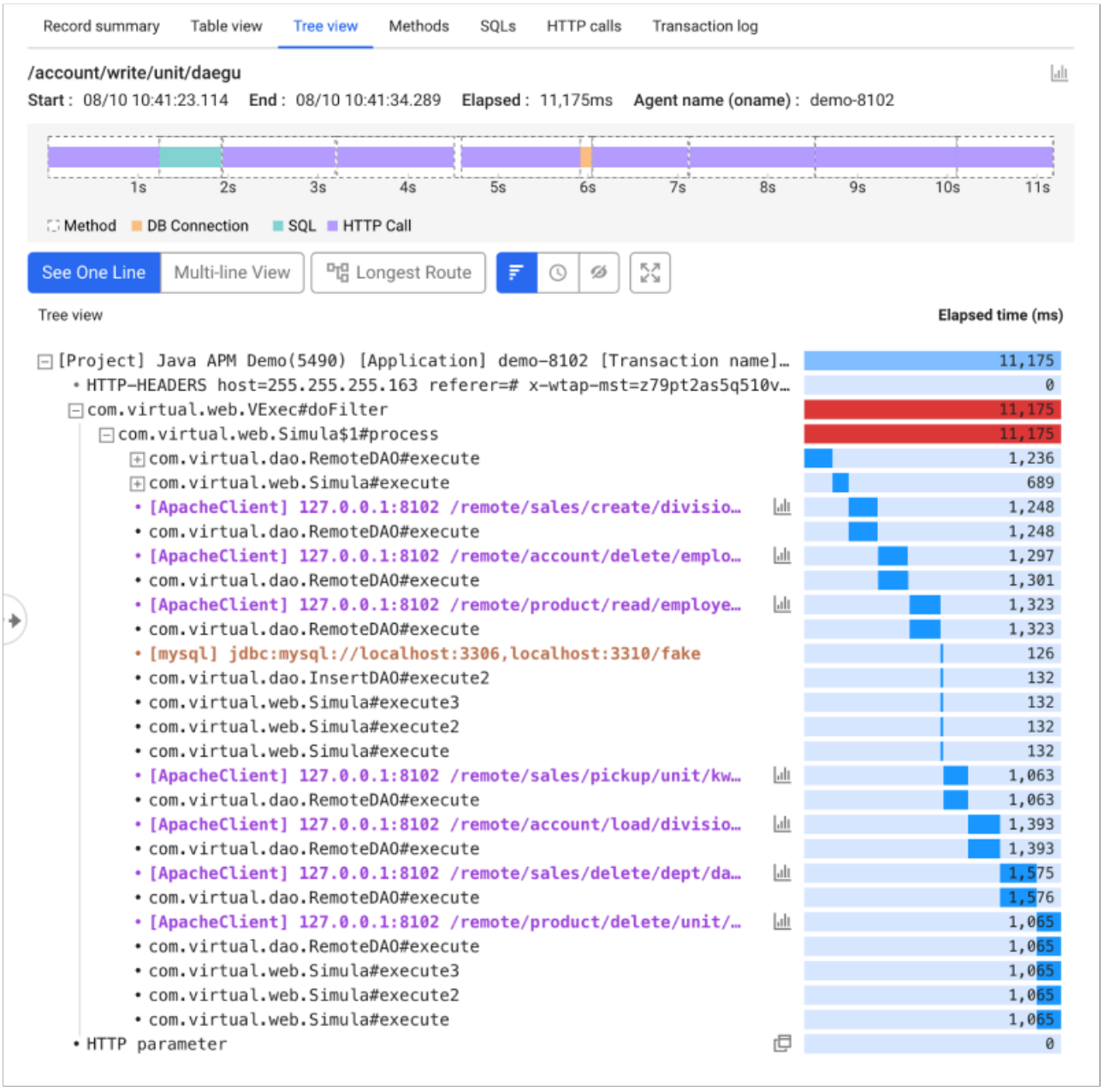
| Classification | Properties | Description |
|---|---|---|
| | ID | |
| | Agent server name | Agent server name set with the `ondoe` option. |
| | Agent server ID | It is a unique ID of the agent server. |
| **Step** | HTTP method | HTTP method. (GET, POST, PUT, HEAD, etc.) |
| | HTTP call time | Outbound 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 time | CPU time used for processing transactions. |
| | Memory allocation | Amount of memory used by the transaction. |
| **Containers** | Pod name | It is the name of the Kubernetes Pod that processed the transaction. |

| Classification | Properties | Description |
|---|---|---|
| | 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.



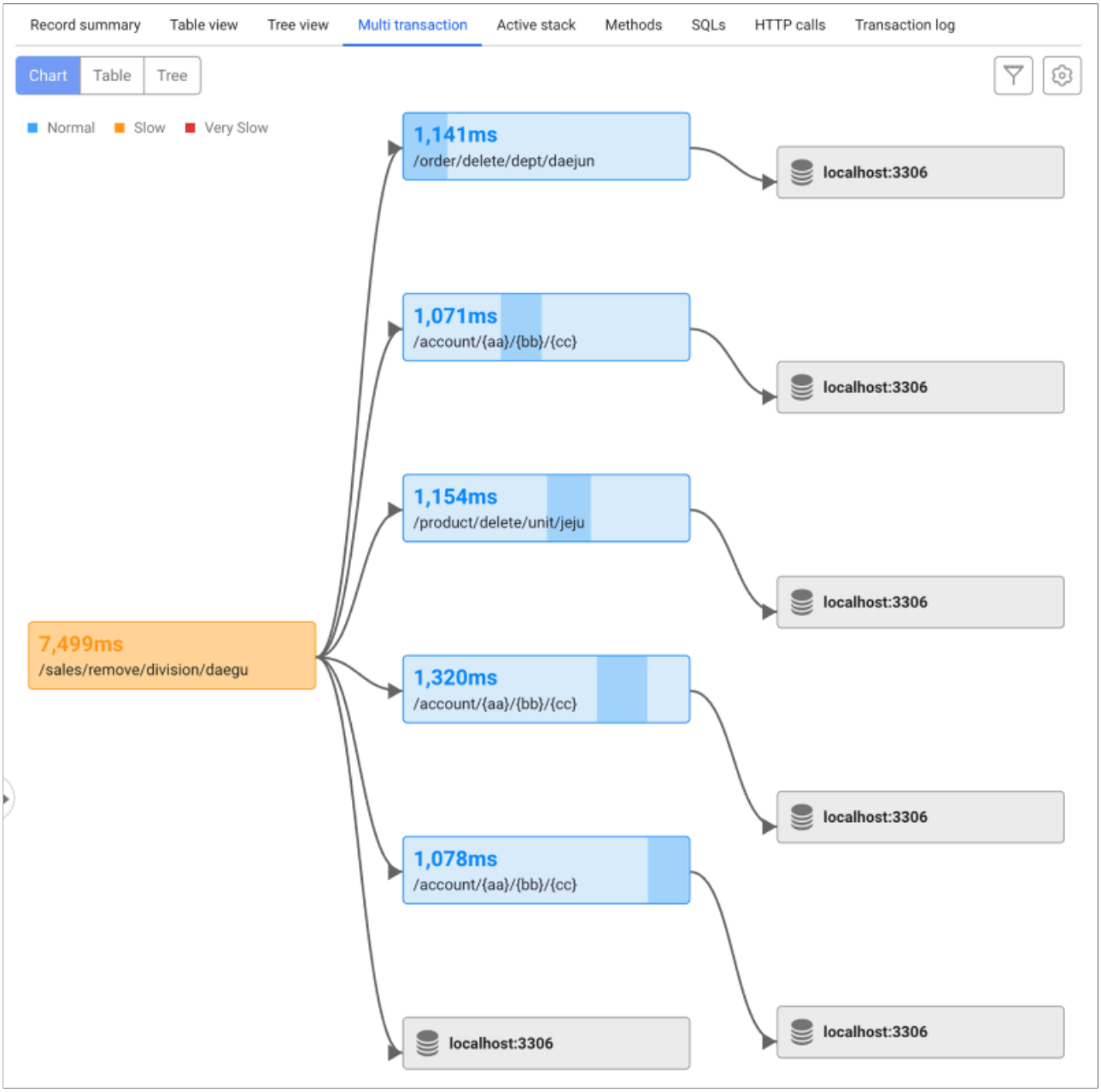
- **Single-line View**: You can arrange the tree format at intervals by displaying the texts displayed in the execution data for each section in a single line.
- **Multi-line View**: All the texts displayed in the execution data for each section, appear by wrapping words.
- **Longest Path**: The longest 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.

## 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.



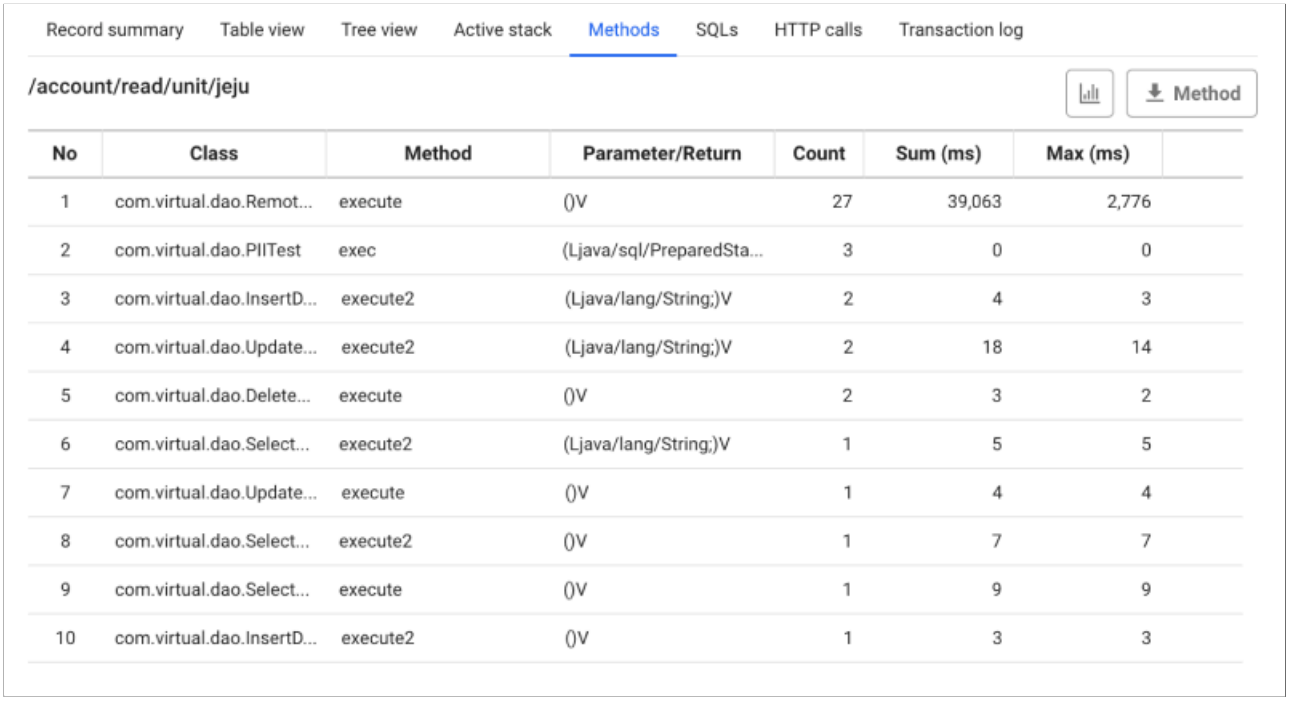

- **Chart**: Provides the call relationships for each transaction in a flowchart. Select a transaction node to see the trace analysis for the transaction node. In **Chart** mode, you can use the mouse to go to a desired position or zoom in or out by scrolling.
  - ⚙ **Chart view settings**: You can display 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 |

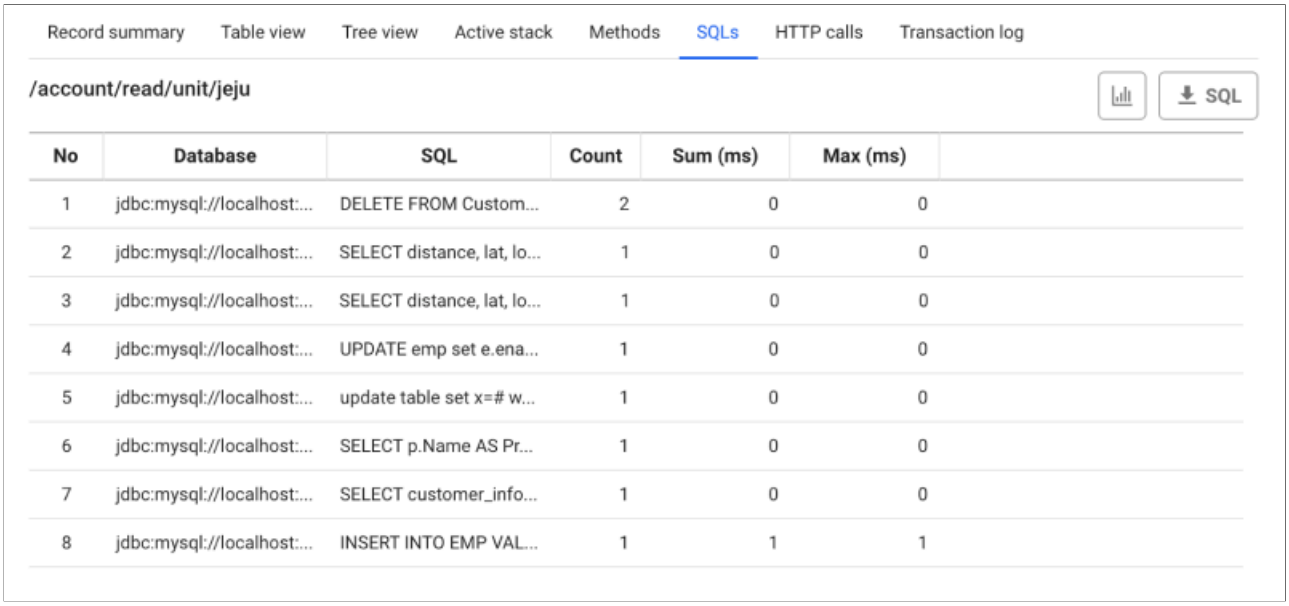
To see only the method data, select the **Methods** tab. It displays the method name and step time for which tracing has been set on the agent. When unnecessary logics are repeatedly executed, it can be used as analysis data to improve the method logic by finding the case where the execution takes longer 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

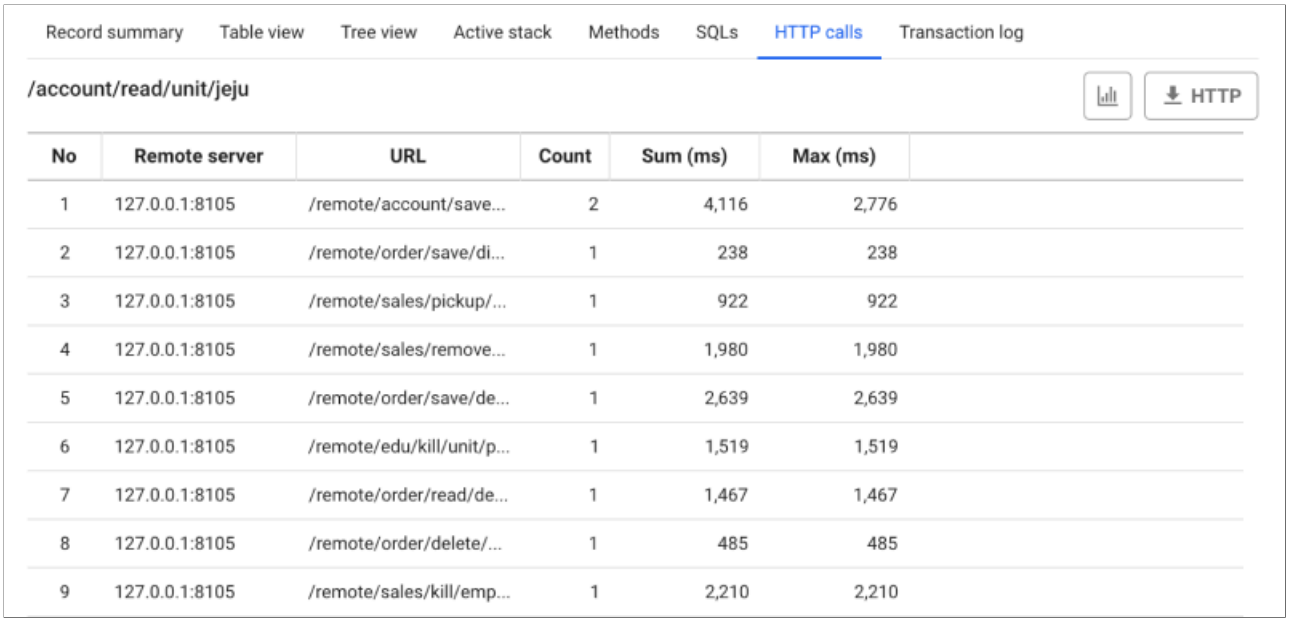



| No | Database | SQL | Count | Sum (ms) | Max (ms) |
|---|---|---|---|---|---|
| 1 | jdbc:mysql://localhost:... | DELETE FROM Custom... | 2 | 0 | 0 |
| 2 | jdbc:mysql://localhost:... | SELECT distance, lat, lo... | 1 | 0 | 0 |
| 3 | jdbc:mysql://localhost:... | SELECT distance, lat, lo... | 1 | 0 | 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



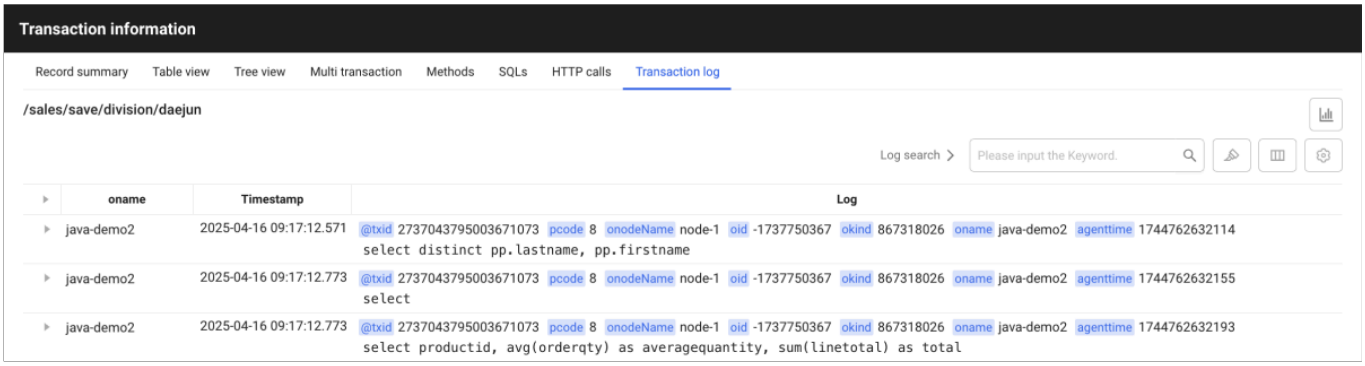

| No | Remote server | URL | Count | Sum (ms) | Max (ms) |
|---|---|---|---|---|---|
| 1 | 127.0.0.1:8105 | /remote/account/save... | 2 | 4,116 | 2,776 |
| 2 | 127.0.0.1:8105 | /remote/order/save/di... | 1 | 238 | 238 |
| 3 | 127.0.0.1:8105 | /remote/sales/pickup/... | 1 | 922 | 922 |
| 4 | 127.0.0.1:8105 | /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.

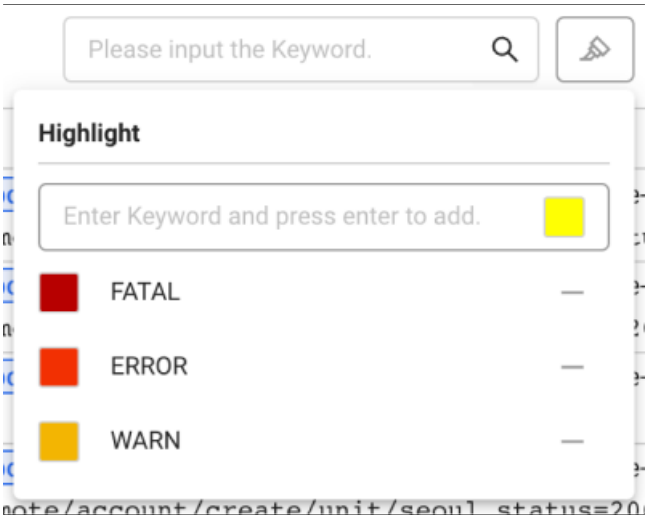



- : You can see the **Transaction statistics** information. If a graph on the chart is selected, the **Analysis** > **Transaction Search** menu appears. You can see the list of transactions for the selected time zone.
- **Log Search**: You can go to Log Search to easily find the desired log data.
- **Keyword 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.

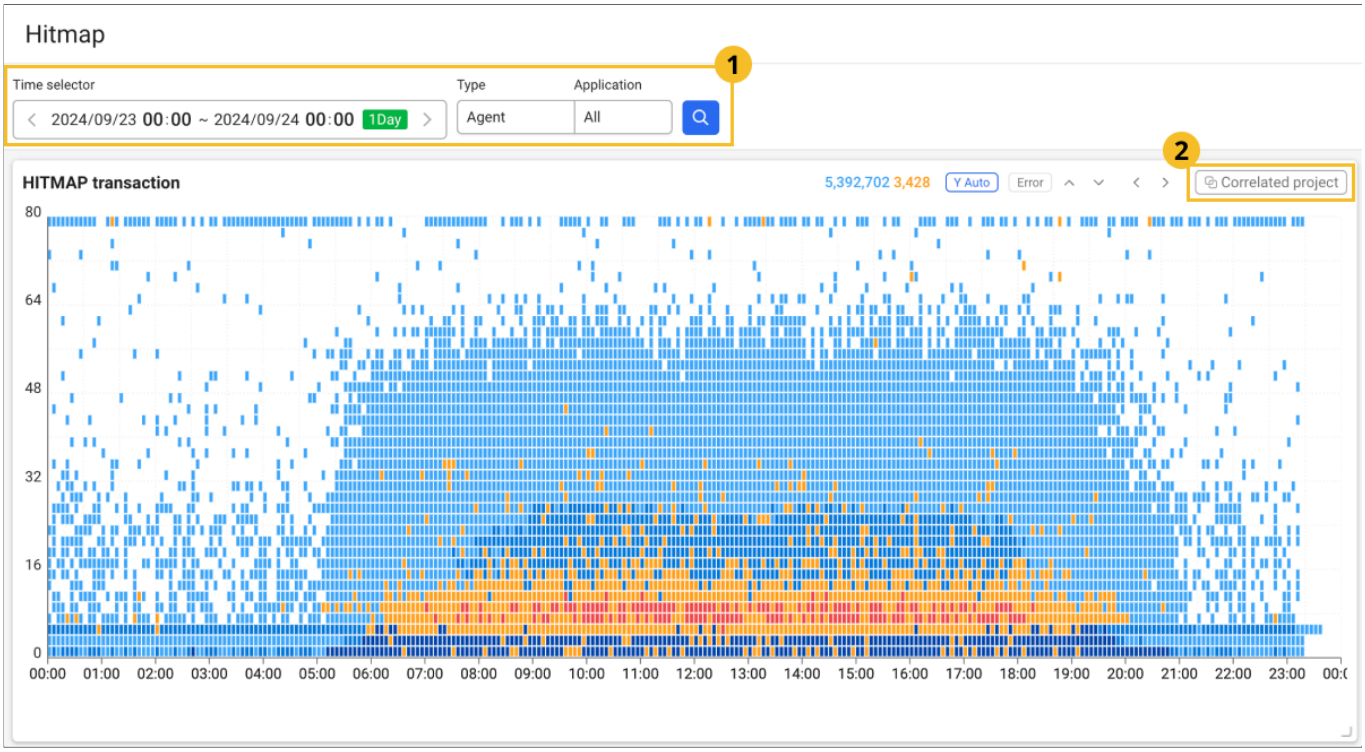
> ⓘ For more information about **Log view detail setting**, see the following.

# Additional options

## Checking the correlated project metrics

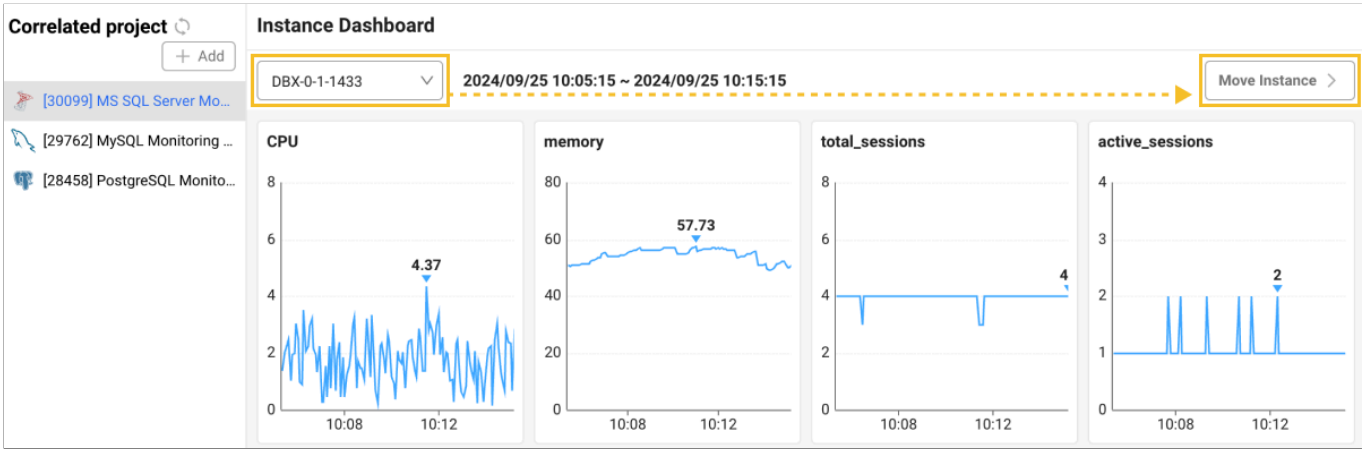
If you have created a database monitoring (DPM) project and an application monitoring (APM) project, you can relate the two projects for monitoring the data collected from DPM in the APM project. You can see performance metrics of the correlated projects while querying transactions via the **Correlated 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.

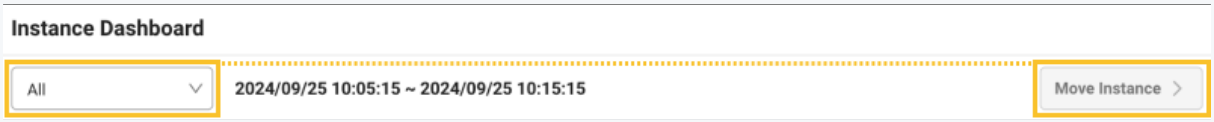
Select the correlated project from the left list and then check the metrics in the **Instance dashboard** section.




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.

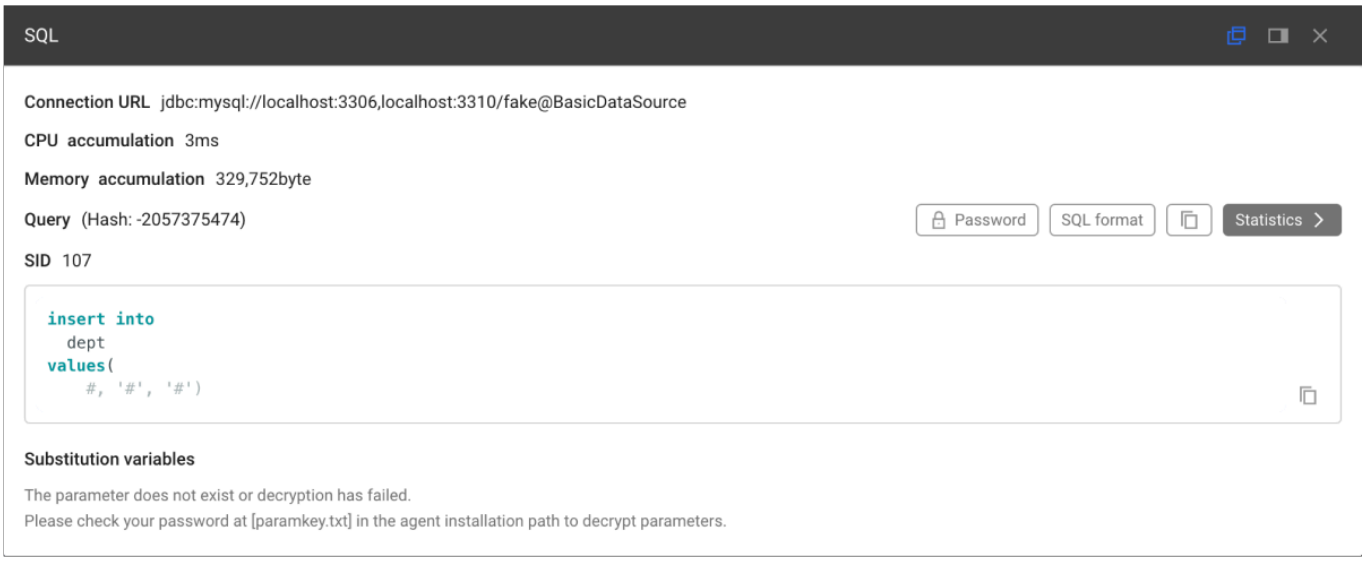
1. Scroll down the page to the final step of the transaction.
2. Select the **HTTP parameters** item.
3. If the **HTTP-PARAMETERS** window appears, select **Password**.
4. Enter the specified Param Key. You can see the hidden parameters.

- 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 ( `paramkey` ) in the `WHATAP_HOME` /security.conf file. You can change it to another string. It is required to view the SQL variables and HTTP queries and to stop the threads.
- Depending on the agent version, the password configuration file may be different. Change the string in the security.conf file or paramkey.txt file.

## 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.

  > Depending on the agent version, the password configuration file may be different. Change the string in the security.conf file or paramkey.txt file.

- **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**.

# Multiple Transaction Trace

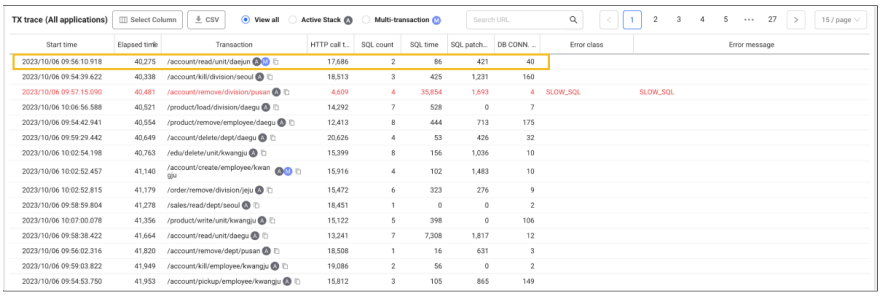
**Multi-transaction** means the transaction involving a different agent or project. **Multiple Transaction Trace** is to trace calls between the application services registered in the WhaTap project.

> **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.

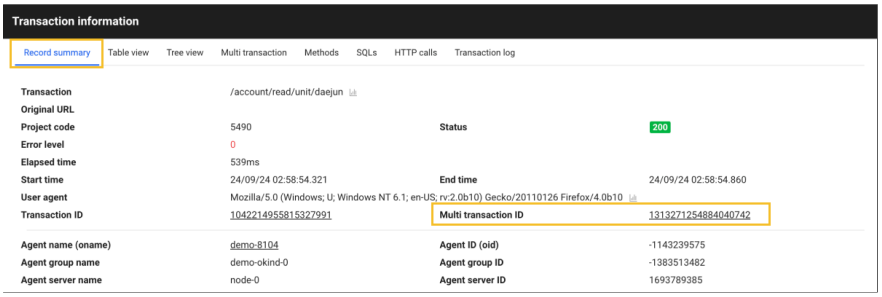
## Checking the multi-transaction ID

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 Ⓜ 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

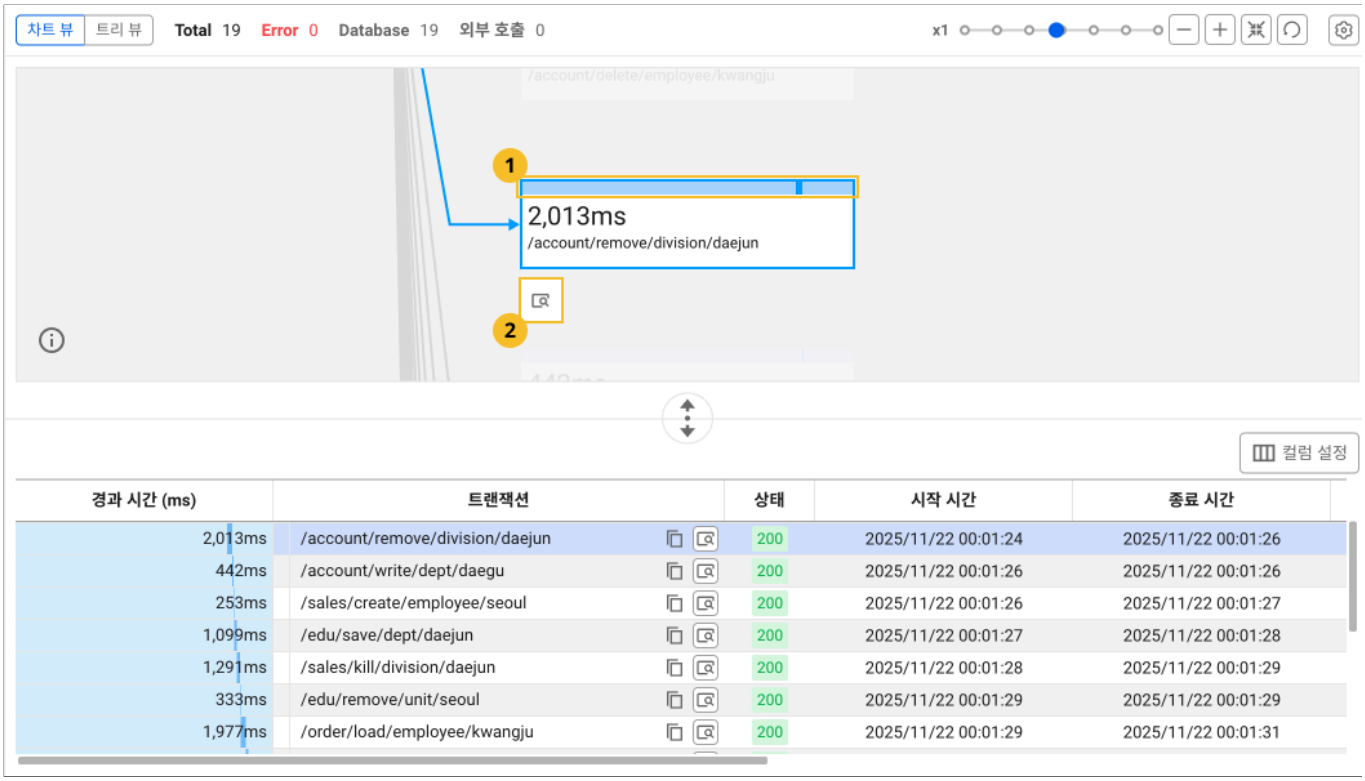
1. Go to **Analysis** > **Multiple Transaction Trace**.
2. Enter the **MTID** value found in the **Transaction information** window into the **Search by MTID / CUSTID** field.
3. Select the date and project you want to query.
4. Click the **Apply** button at the bottom of the screen.
   - The **Apply** 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.
> ```

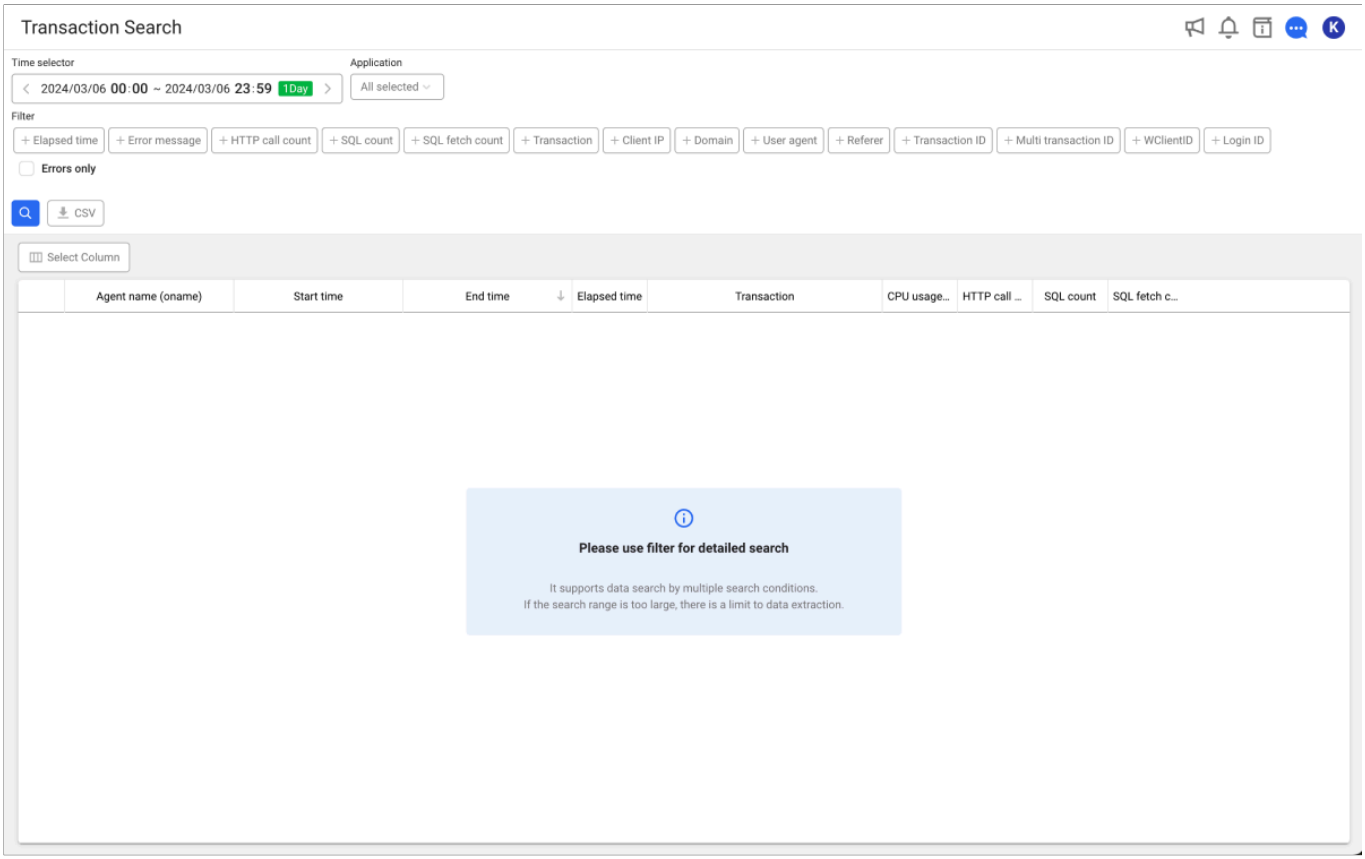
```
profile_sql_param_enabled=true

// HTTP parameter lookup option: If the option is applied, HTTP query parameters are collected after being encrypted.
profile_http_parameter_enabled=true
```

# Transaction Search

Home > Select Project > **Analysis** > **Transaction Search**

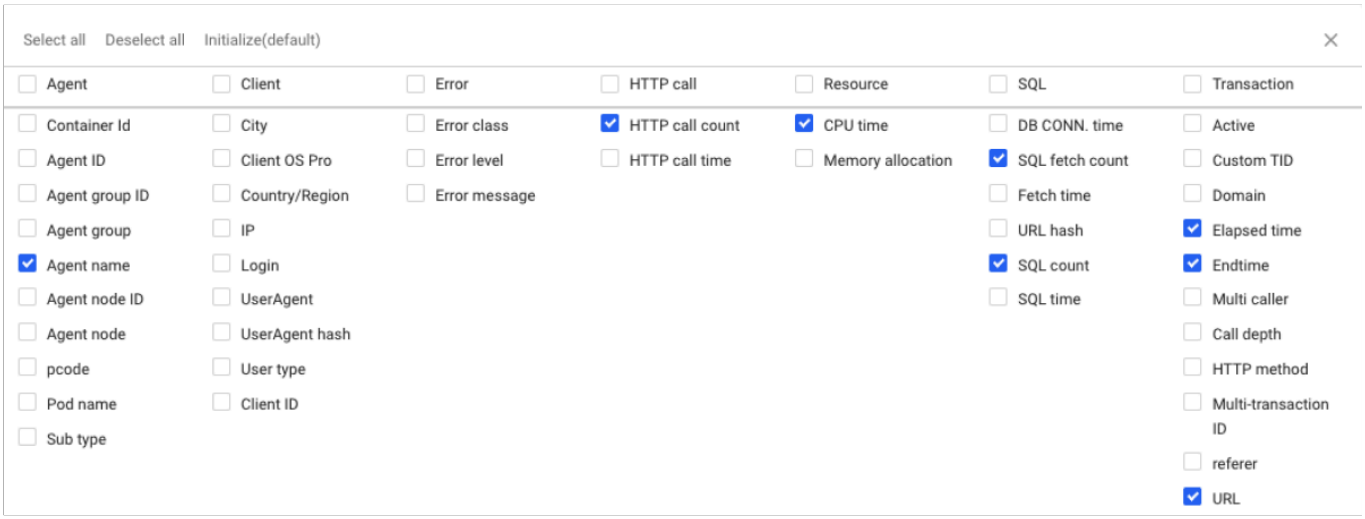
You can view the statistics on the performance properties and execution status for each transaction for a specific period of time. You can view all transactions and quickly check their status (normal or error). Through the provided information, you can find which transactions are busy and resource-consuming. 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



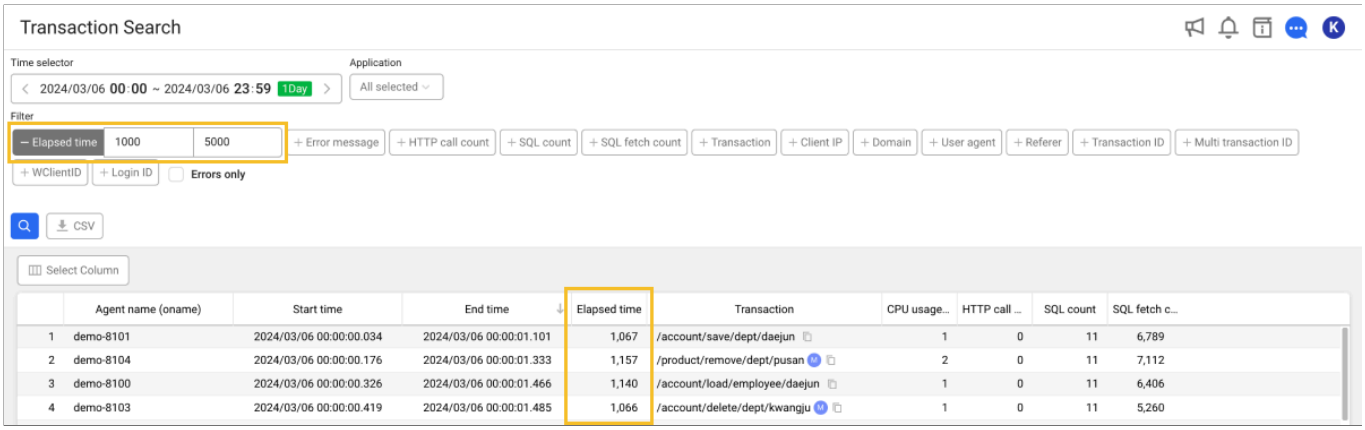

When viewing results, click **Select Column** and then select the columns to be checked.

# Applying filters

Use filters for more details. Data exploration is supported using multiple search conditions. After setting a filter(s), select .

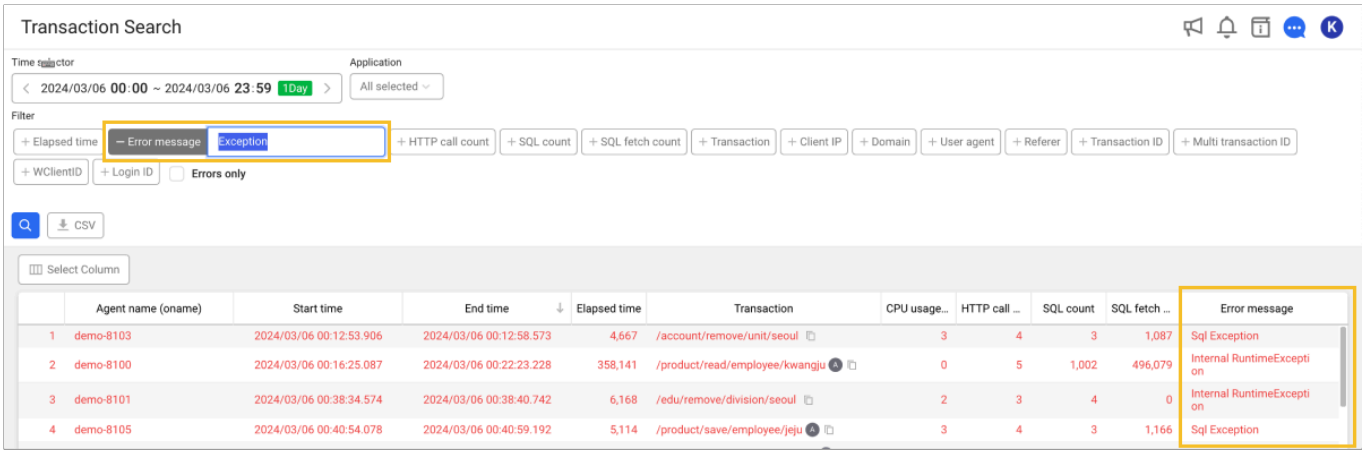
Some filters may not be supported depending on the product.

## Elapsed time




Search based on the transaction response time. Search by entering two values above and below the reference.

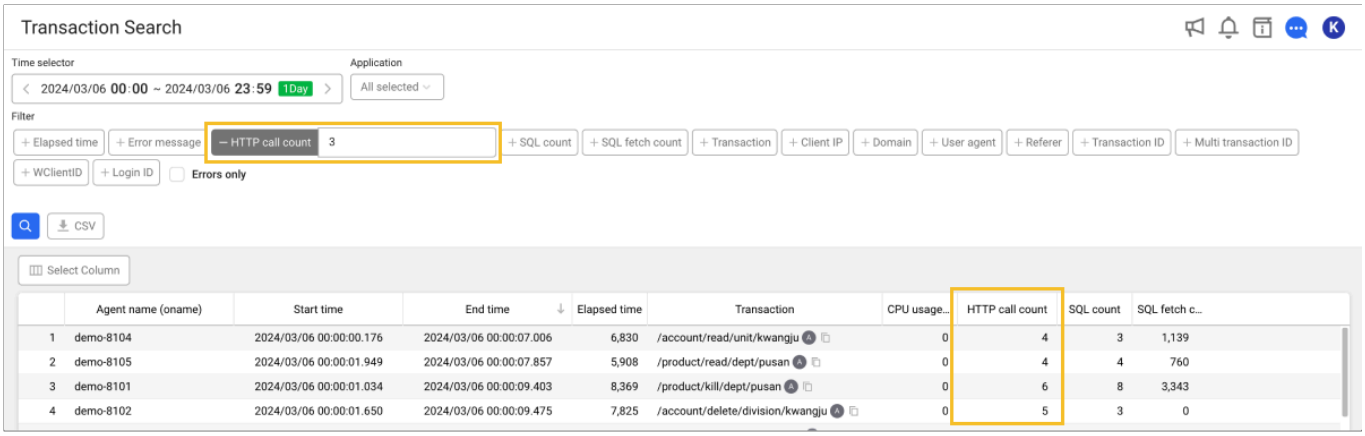
## Error message




Search based on the transaction error message. In the default state, error-related columns may not be seen. Select a column.

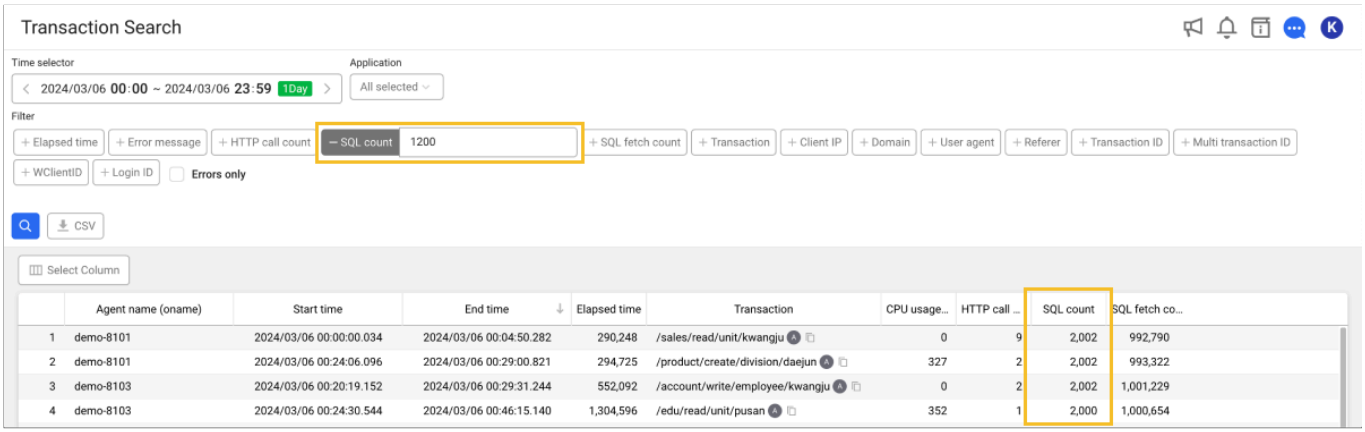
> Transactions in error are displayed in red.

## HTTP call count



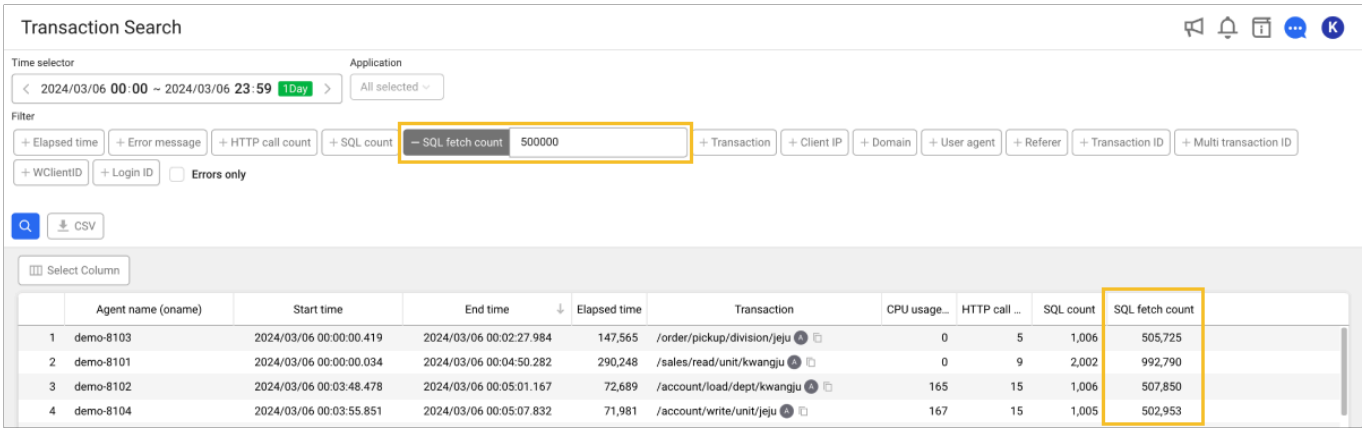

It searches for the transactions that exceed the specified number based on the number of calls for external HTTP services.

## SQL count



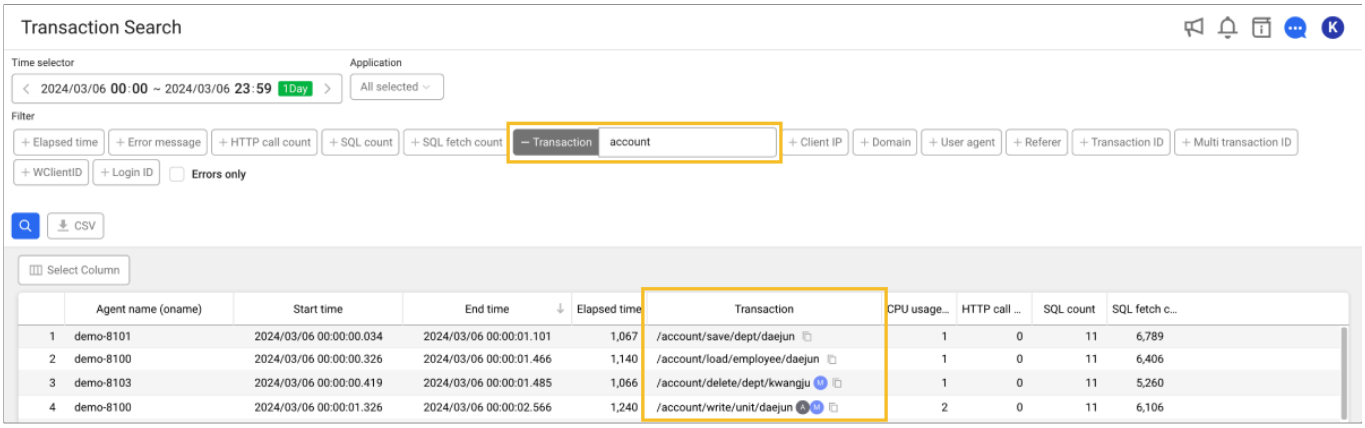

It searches for the transactions where the number of SQL executions per transaction is greater than or equal to the specified number. Exceptionally, this is used to search transactions that call SQLs excessively.

## SQL fetch count



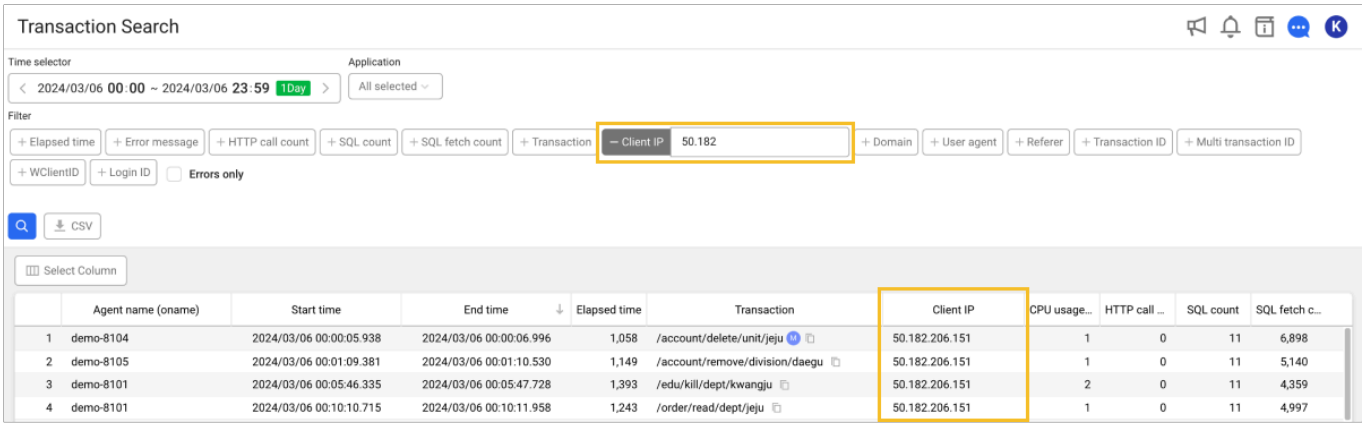

After executing an SQL, it searches for the transactions exceeding the specified number based on the number of DB patches. Use this to search transactions that perform excessive data fetches.

## Transaction



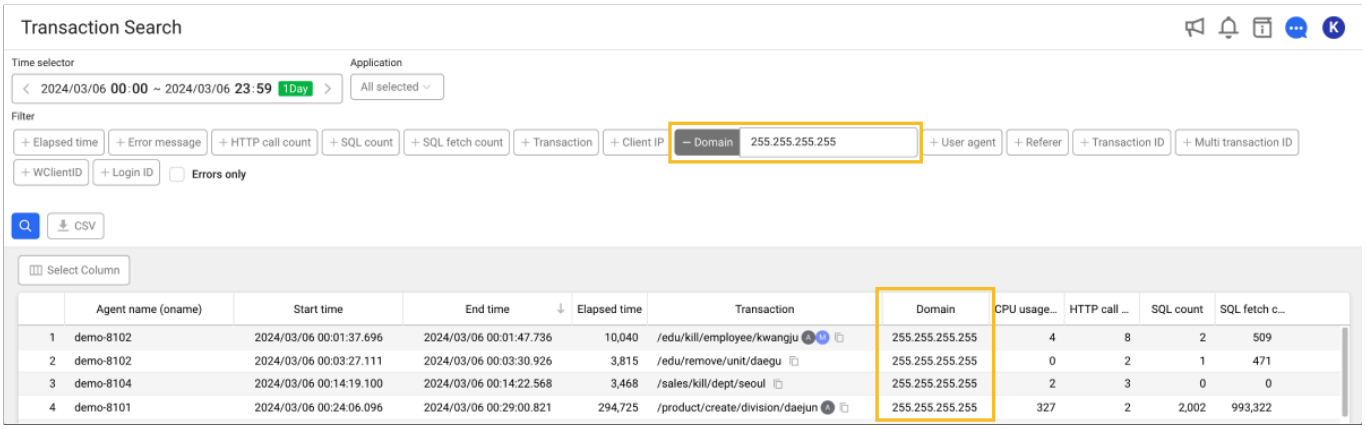

It searches based on the transaction. The entered search string is compared and searched by inclusion (partial matching).

## Client IP



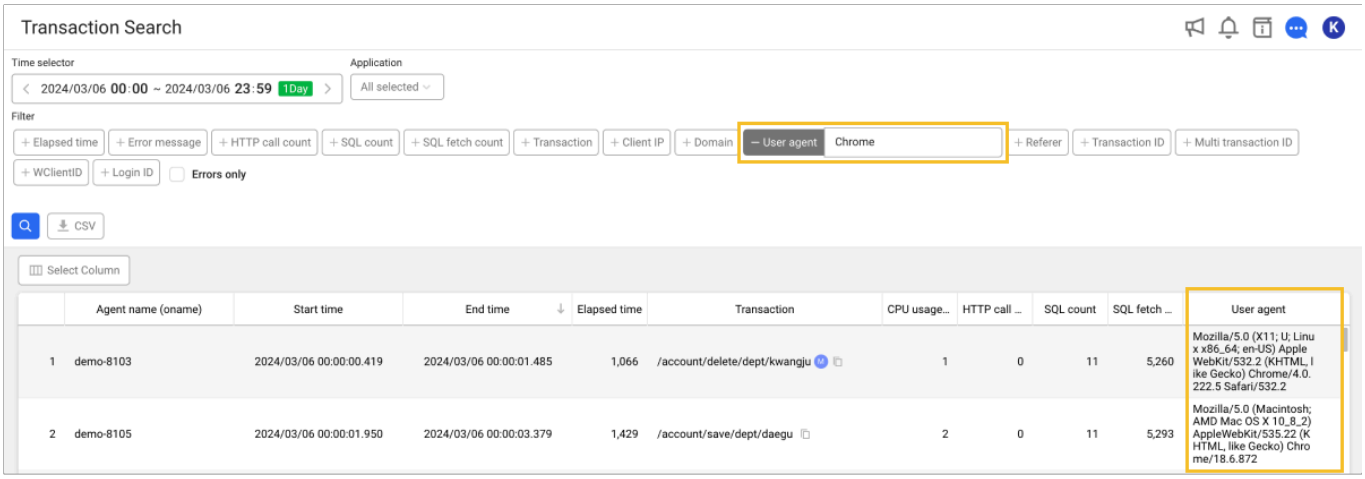

It searches transactions based on the client IP. However, IP strings are searched by prefix comparison (forward matching).

## Domain



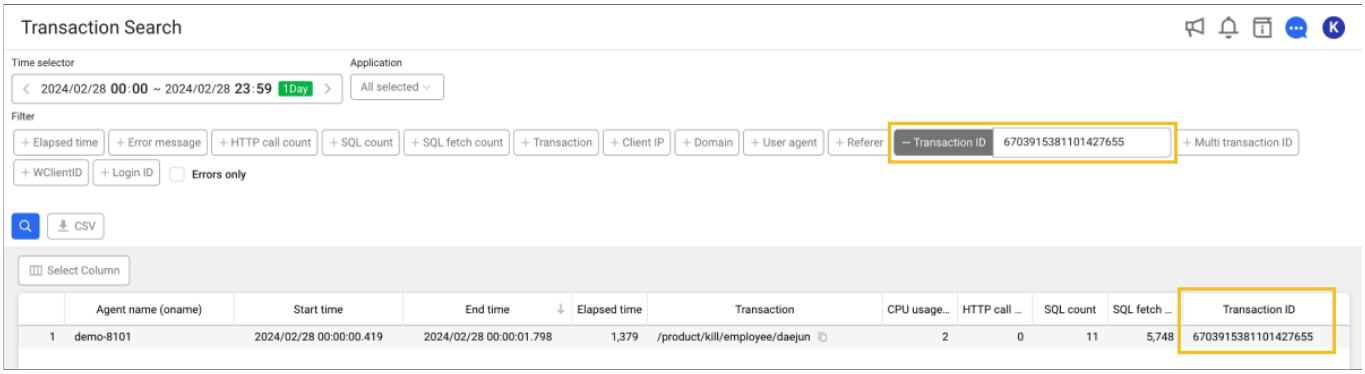

Retrieve transactions based on the HTTP service domain (host:port). The search conditions must match the domain completely (full match).

## User agent




Search transactions based on the user agent (partial match).

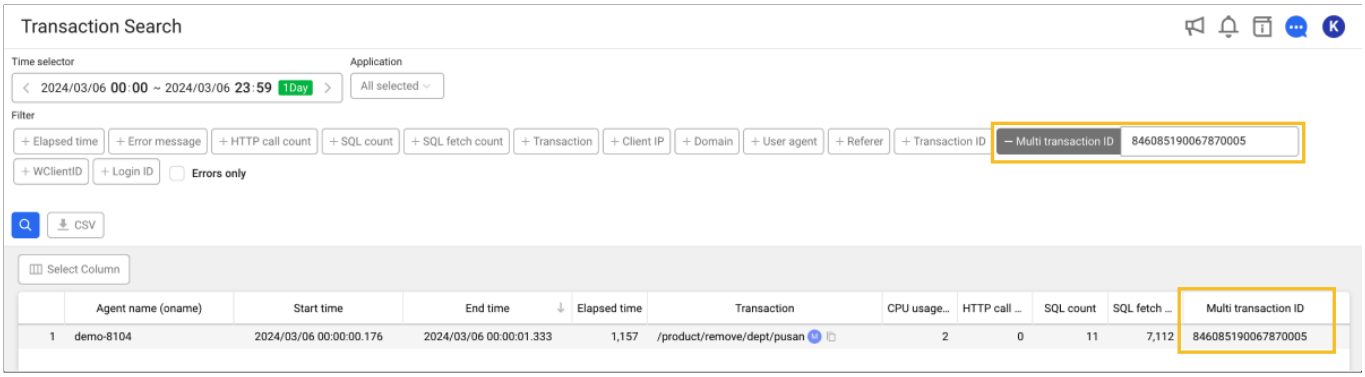
## Transaction ID




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.

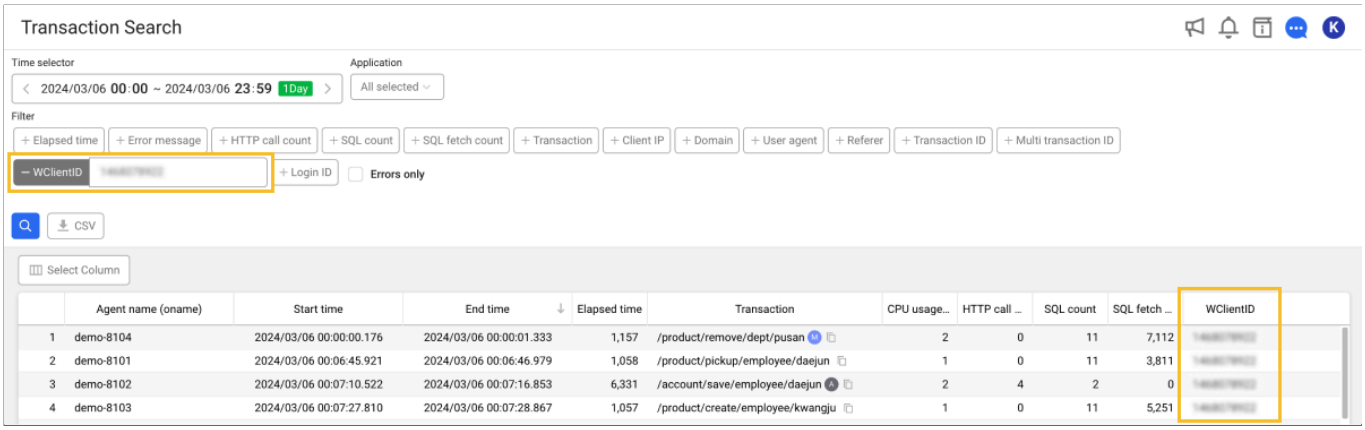
## Multi transaction ID




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



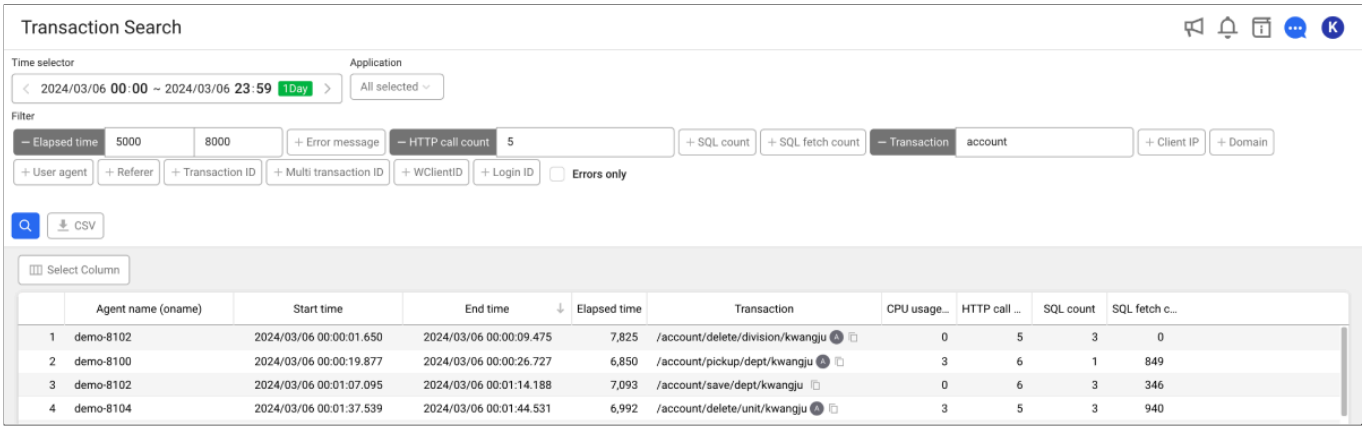

The client ID ( `wClientId` ) is an ID given by the WhaTap agent to identify users. Search transactions based on this

ID (full match). There is no separate index for searching.

> WhaTap basically stores the client-related information.

## 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.

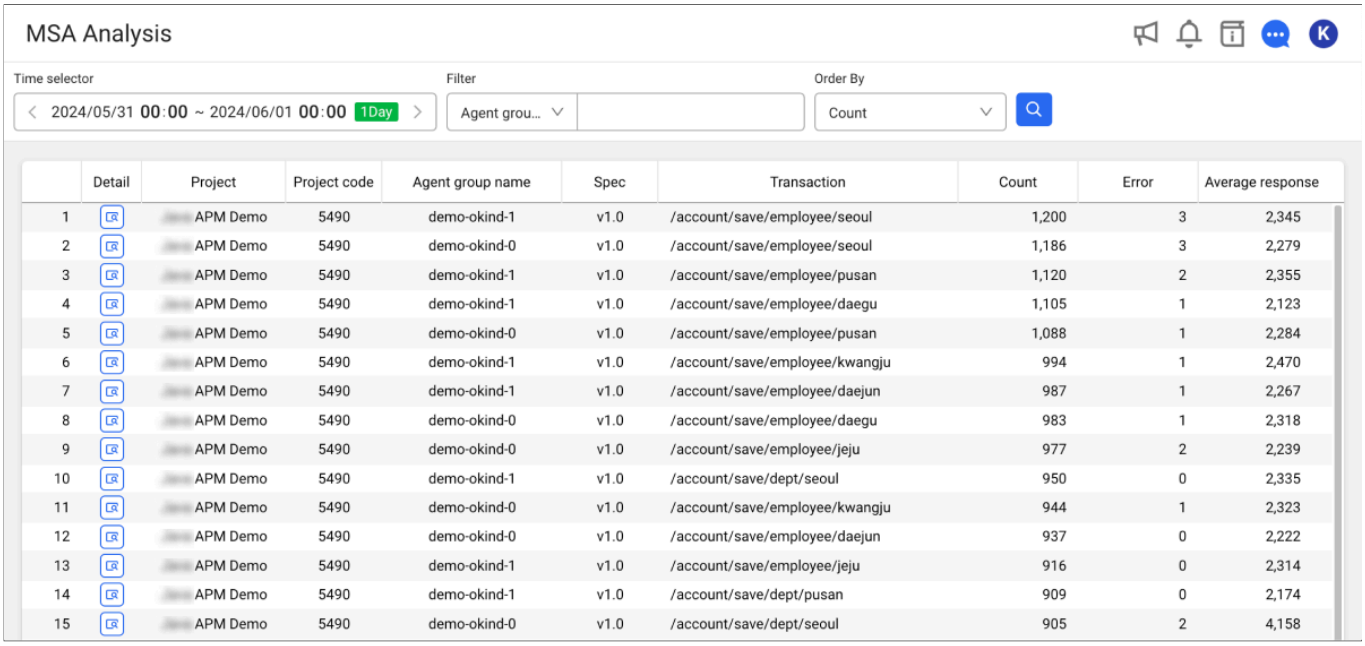
Microservice Architecture Caller & Callee

> 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.



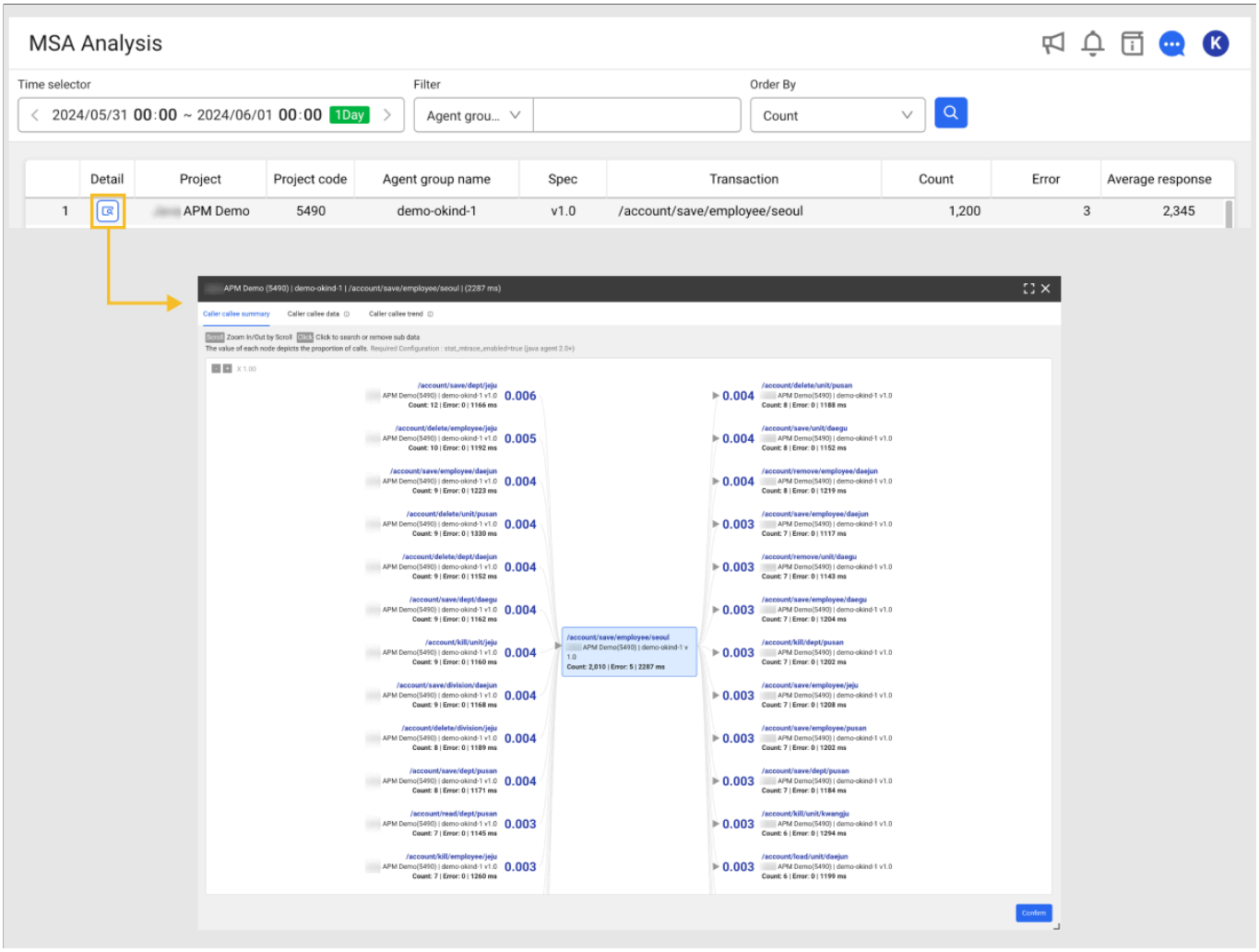

| | Detail | Project | Project code | Agent group name | Spec | Transaction | Count | Error | Average response |
|---|---|---|---|---|---|---|---|---|---|
| 1 | | APM Demo | 5490 | demo-okind-1 | v1.0 | /account/save/employee/seoul | 1,200 | 3 | 2,345 |
| 2 | | APM Demo | 5490 | demo-okind-0 | v1.0 | /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.

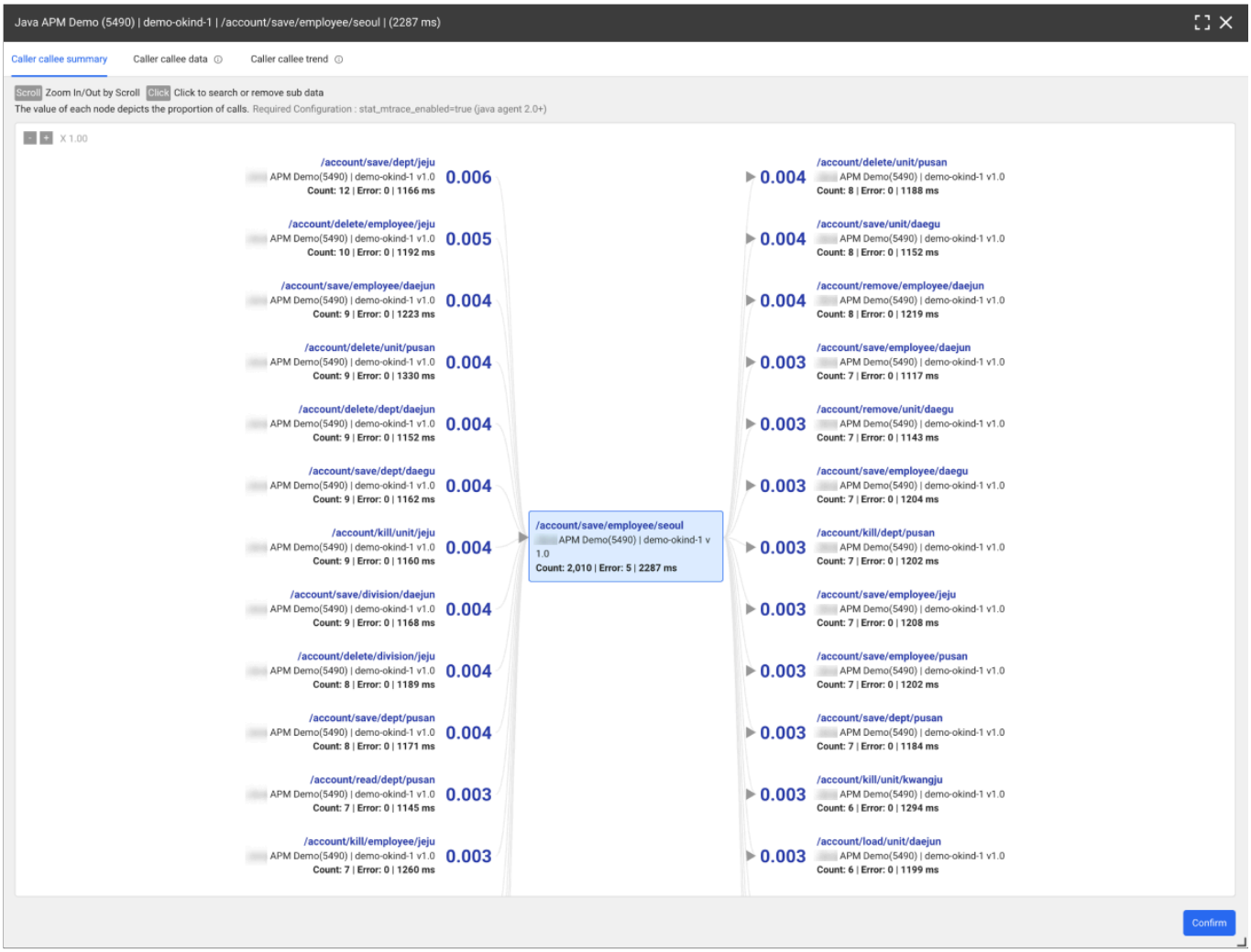
> ⓘ • You can view it in full screen when the ⛶ button is selected on the upper right of the Display Detail window.

- 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

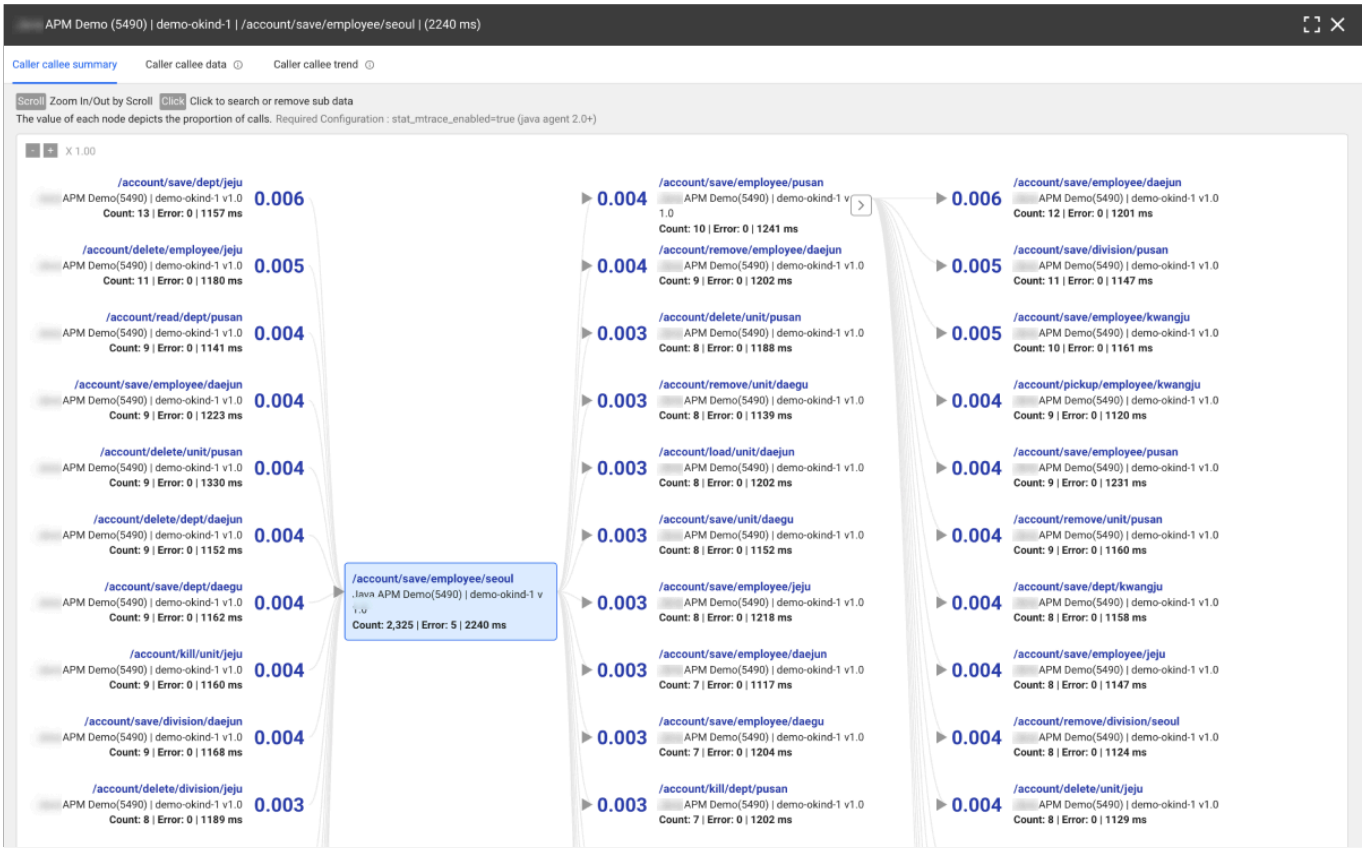



Each node consists of a transaction URL, project name, agent group name, and the following performance data:

- **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

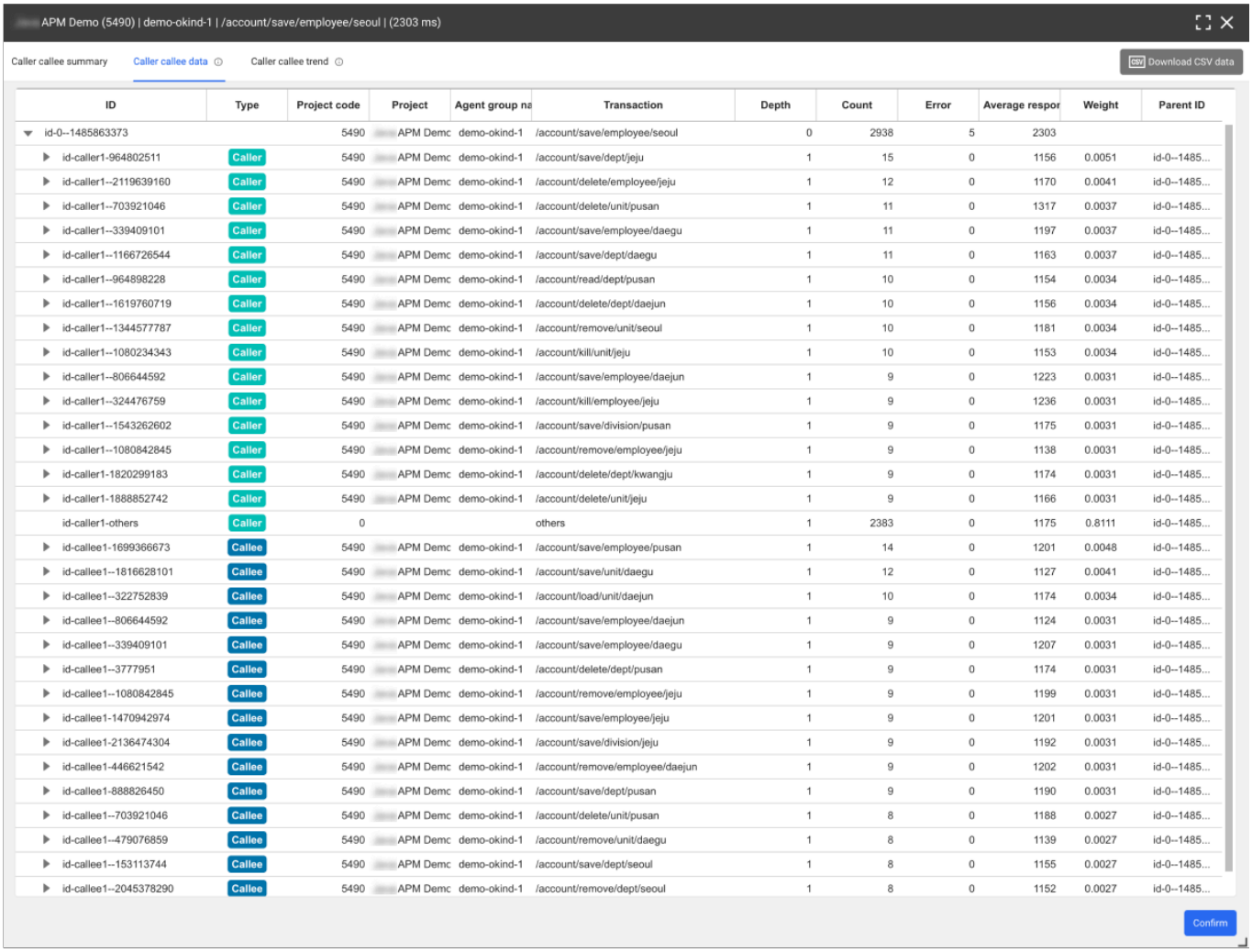
When you hover your mouse over a caller or callee node, the > button appears. If you select this button, it expandes into subnodes. By checking the entire flow of transactions, you can get answers in finding where problems occur.




- 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.



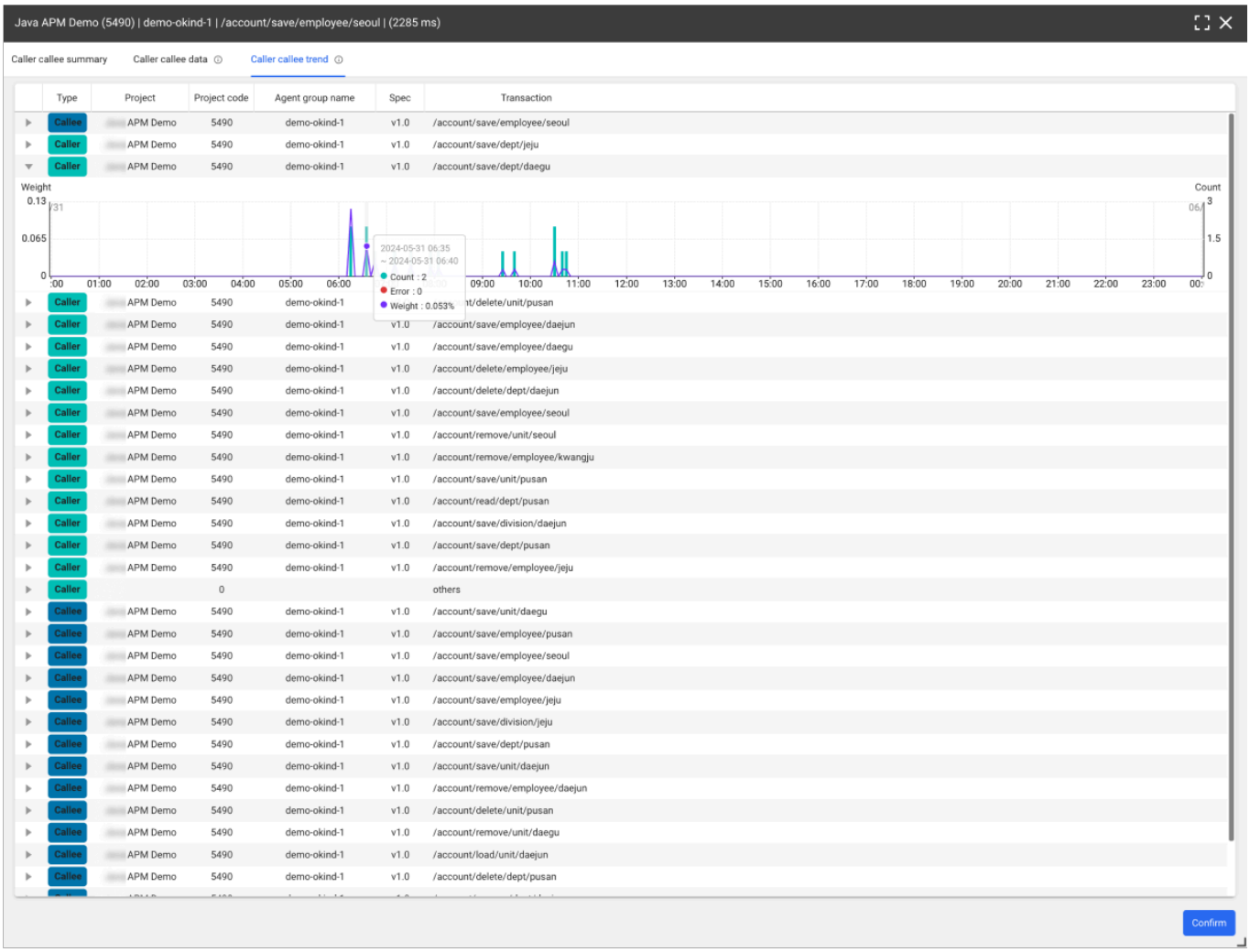

| ID | Type | Project code | Project | Agent group na | Transaction | Depth | Count | Error | Average respor | Weight | Parent ID |
|---|---|---|---|---|---|---|---|---|---|---|---|
| id-0--1485863373 | | 5490 | APM Demc | demo-okind-1 | /account/save/employee/seoul | 0 | 2938 | 5 | 2303 | | |
| id-caller1-964802511 | 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.

# Metrics

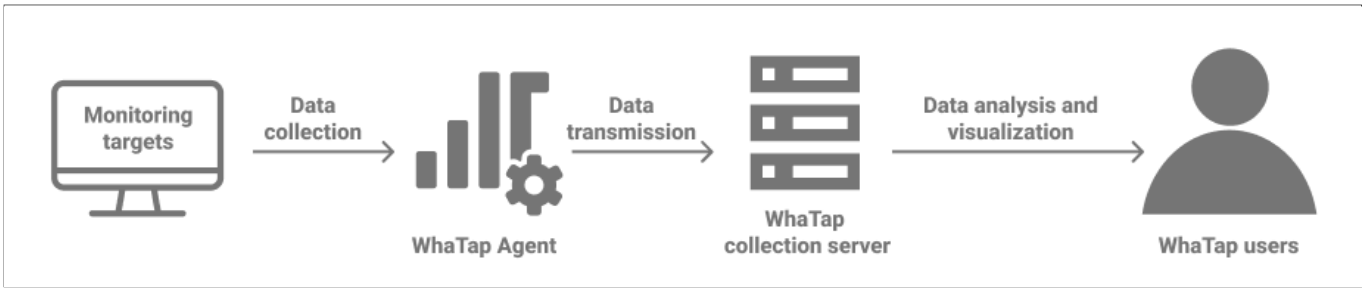
## What is a metric?

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 **Go applications**, first install the agent. See Application metrics for available metrics.

## Metric components

WhaTap **metrics** consist of the following components:

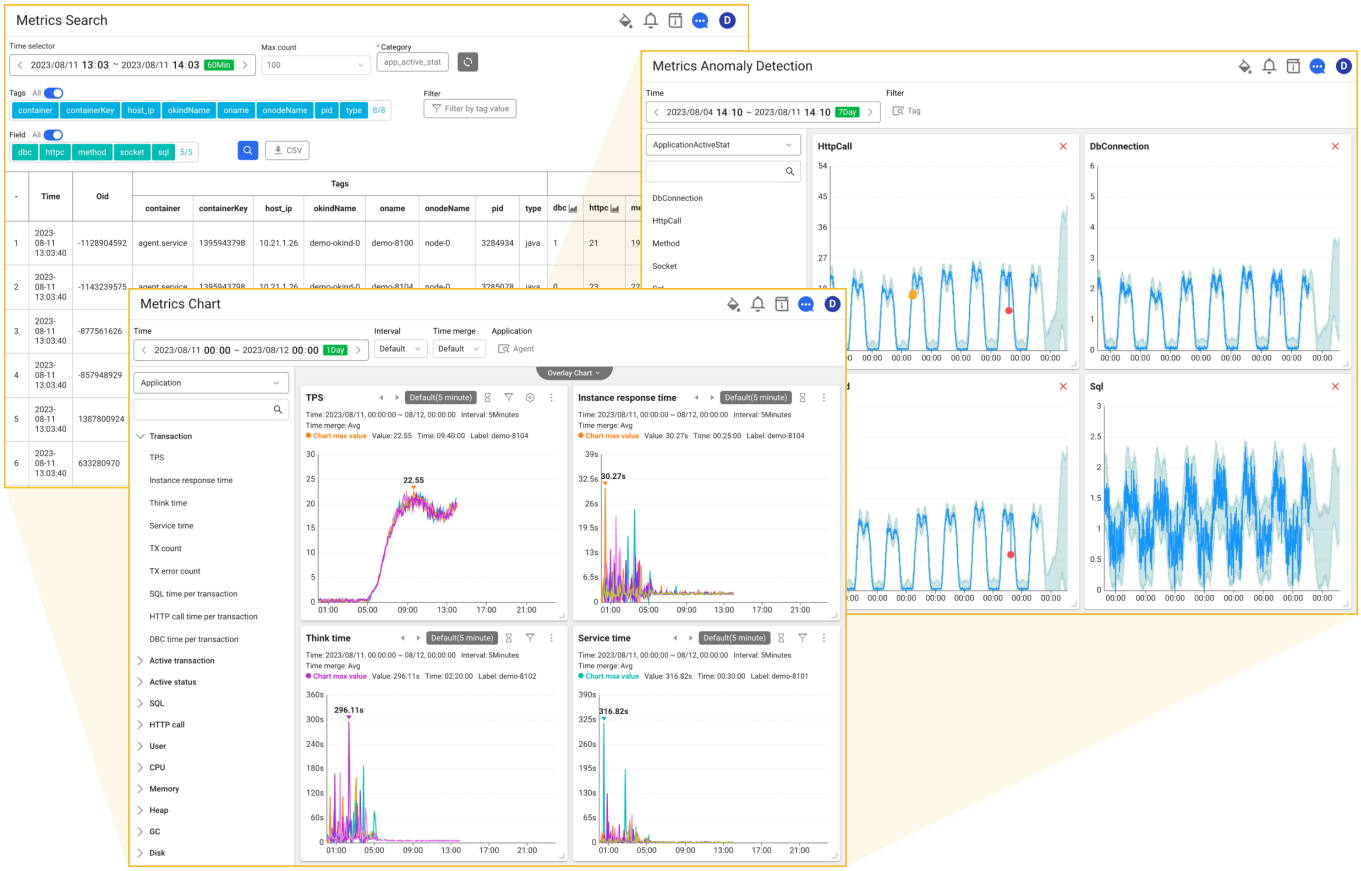
- **Category**: A name that groups related metrics.

- **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 |
| disk | - | percent | Root file system's disk utilization |

| Field | Type | Unit | Description |
|---|---|---|---|
| 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 |

| Field | Type | Unit | Description |
|---|---|---|---|
| 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 |

| Tag | Type | Unit | Description |
|---|---|---|---|
| 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 |
| $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) |

| Field | 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 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 |
| 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 |

| Field | Type | Unit | Description |
|---|---|---|---|
| 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 |

# Application Performance Counter

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

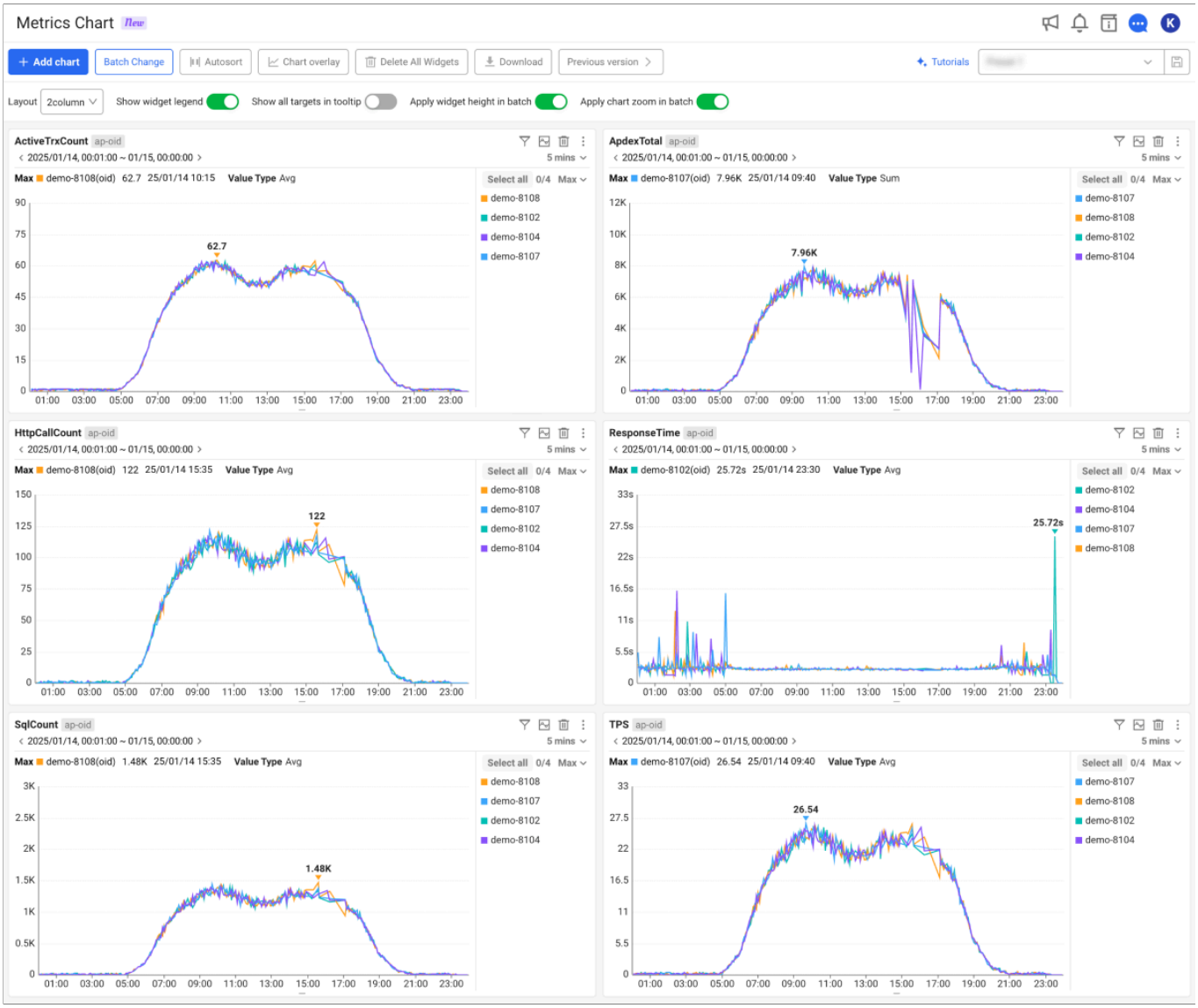
  Default `1200`

  The default value (**S**) can be modified in Agent CONFIG.

# Metrics Chart

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

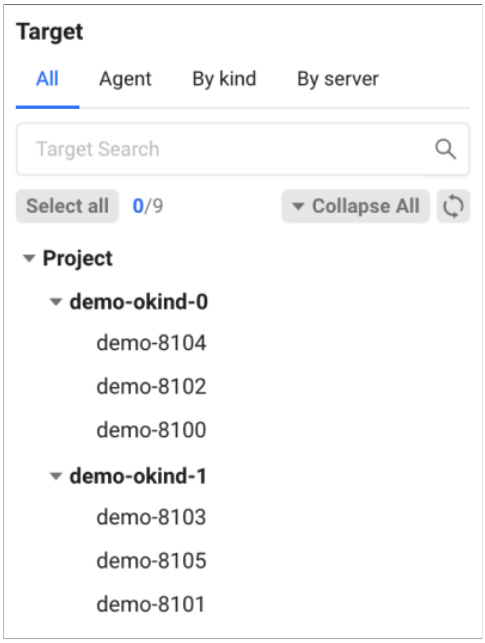
To add a metric as a widget to the dashboard, select **Add chart** on the upper left.

## 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.

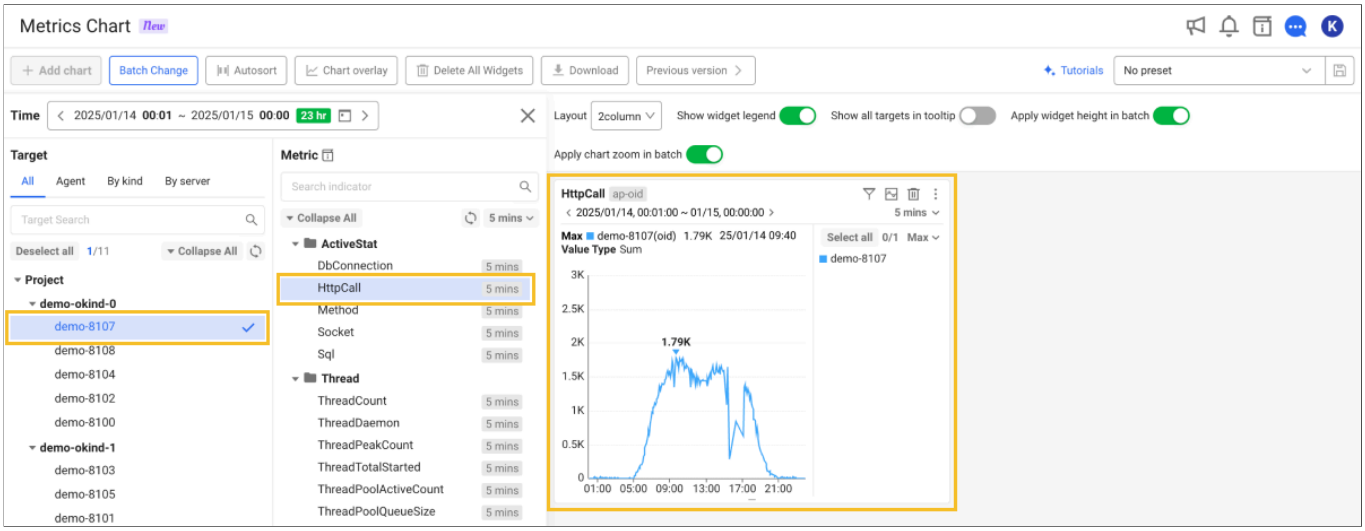



- **Agent**: You can select an individual agent or all agents.
- **By kind**: Agents can be monitored in each group classified by `whatap.okind` in the agent configuration.
- **By server**: Agents can be monitored in each group classified by `whatap.onode` in the agent configuration.

> ⓘ
> - 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

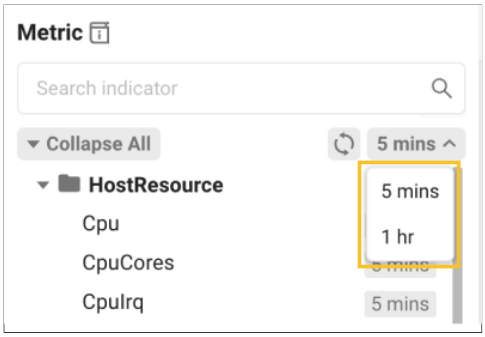
Select a target agent to monitor. Then the list that can be selected on the **Metric** panel appears. Select a desired metric in the list.




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**.

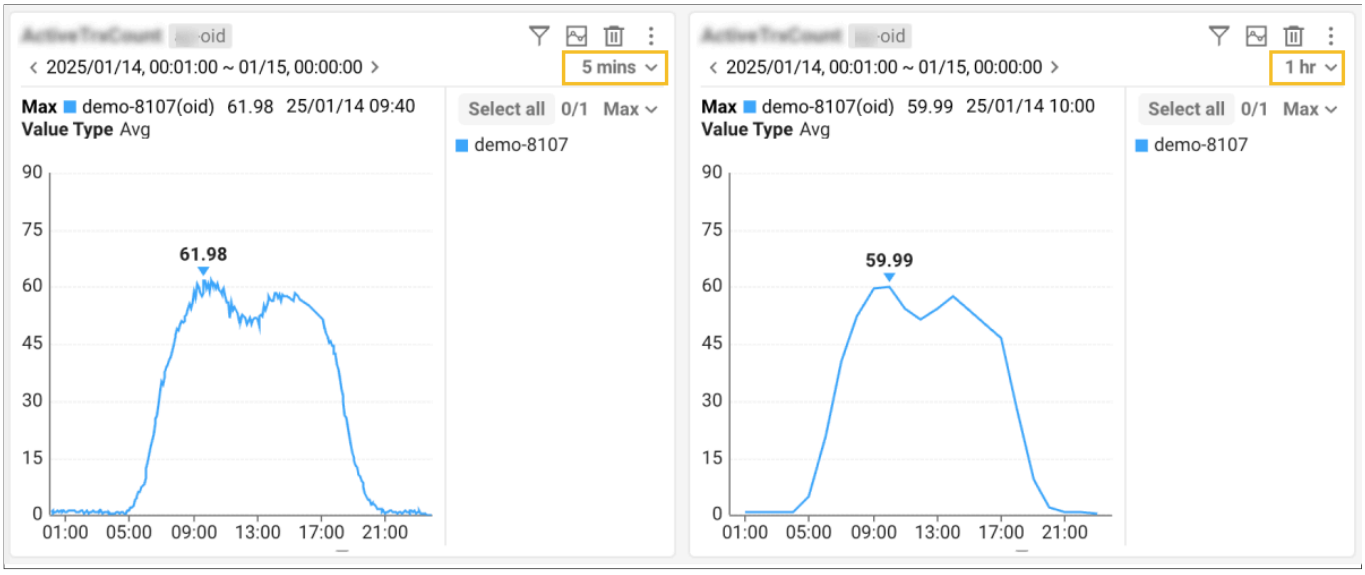
## Setting the interval




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.

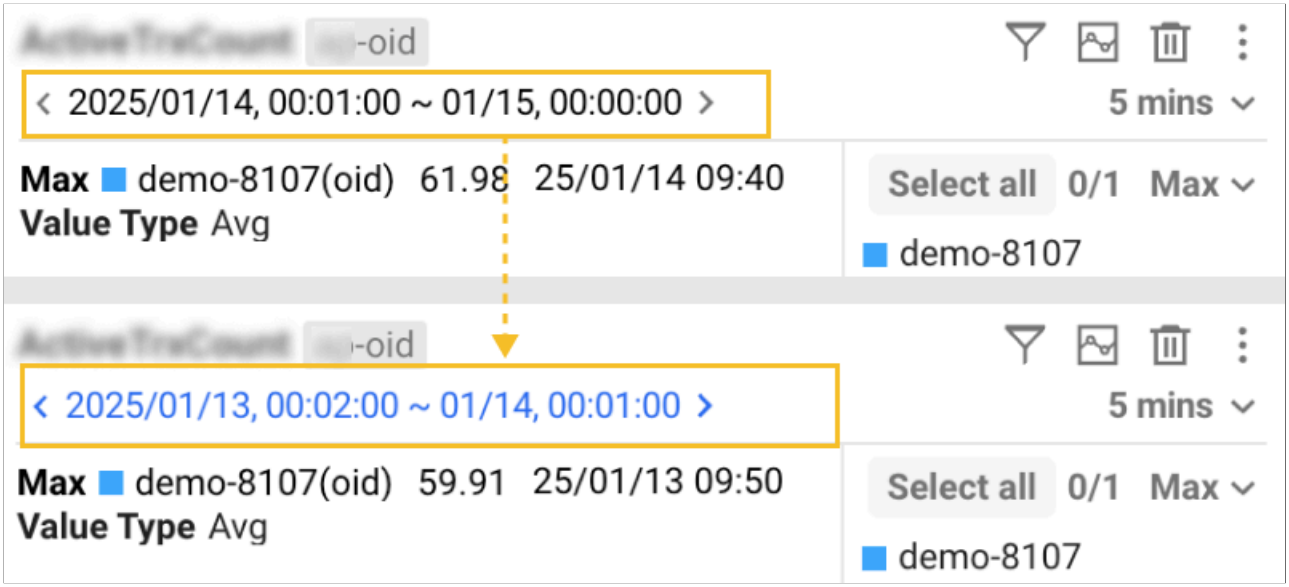
For example, the **5 min** interval displays average values for 5-minute intervals, and the **1 hour** interval displays average values for every one hour.



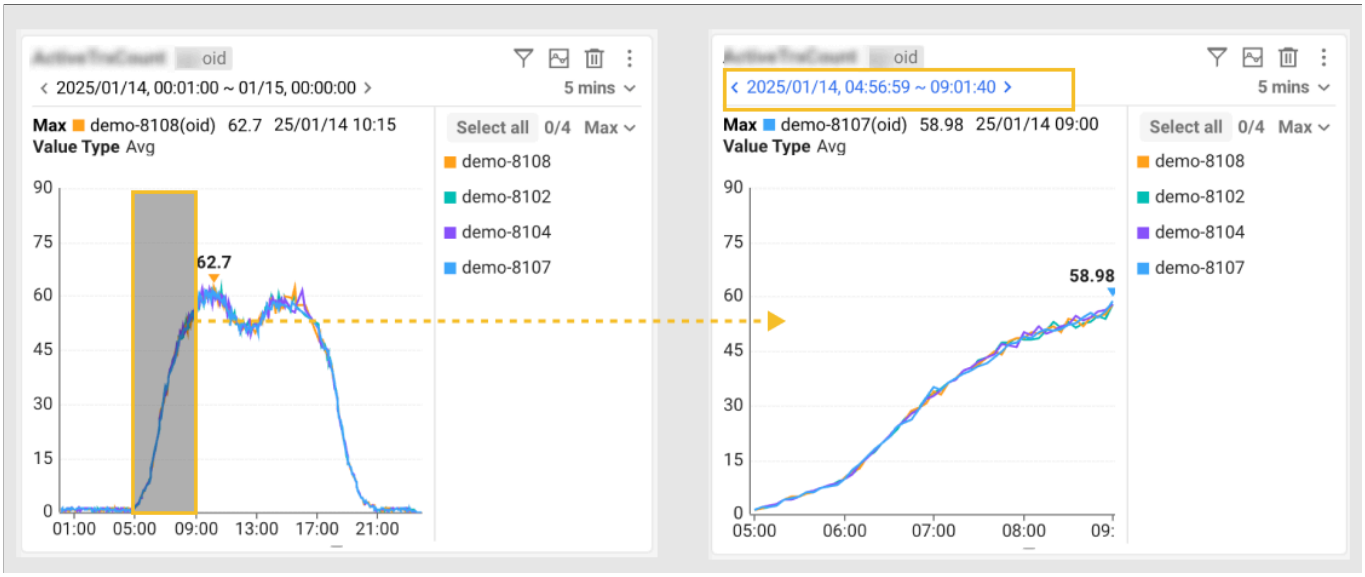

By adjusting the interval option, you can analyze the data from different perspectives, from detailed data over a short time to overall trend over a long time.

# Changing the query time

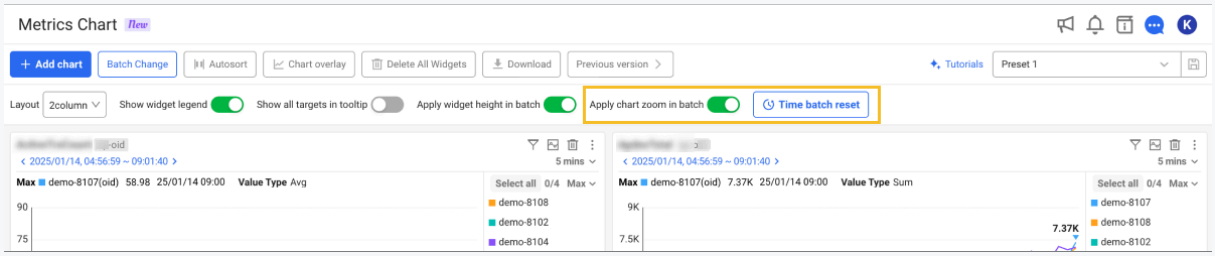
To change the chart query time range for an individual widget, select < or > at the top of the widget. The time change range is the same as the time reference set in the **Time** option. For example, if the **Time** option is set to **1 day** and you select < or >, you can change the time to a daily 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.



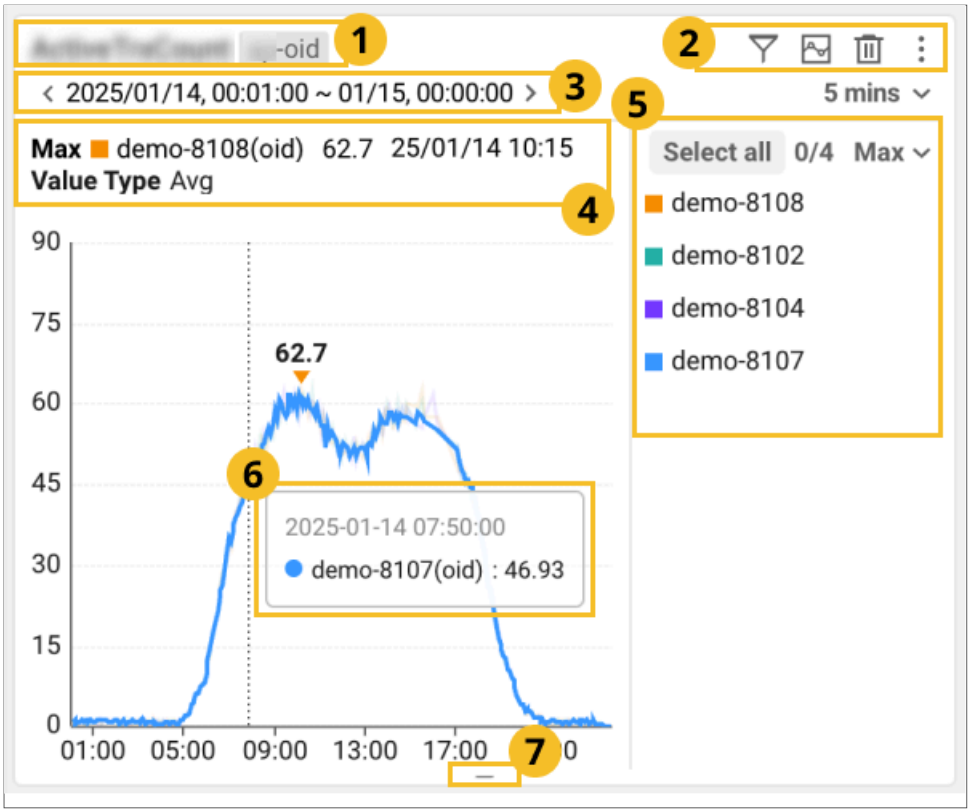

- To change the query time range and then change it back to a recent time, select .

  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.

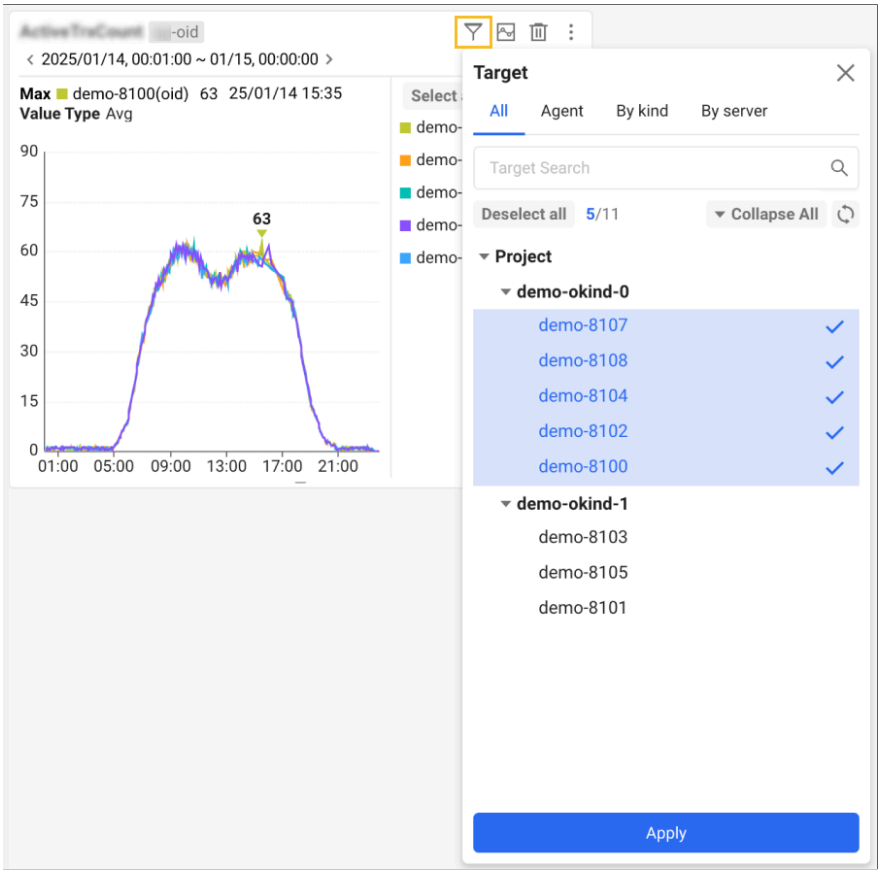
# Using the chart widget options

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.



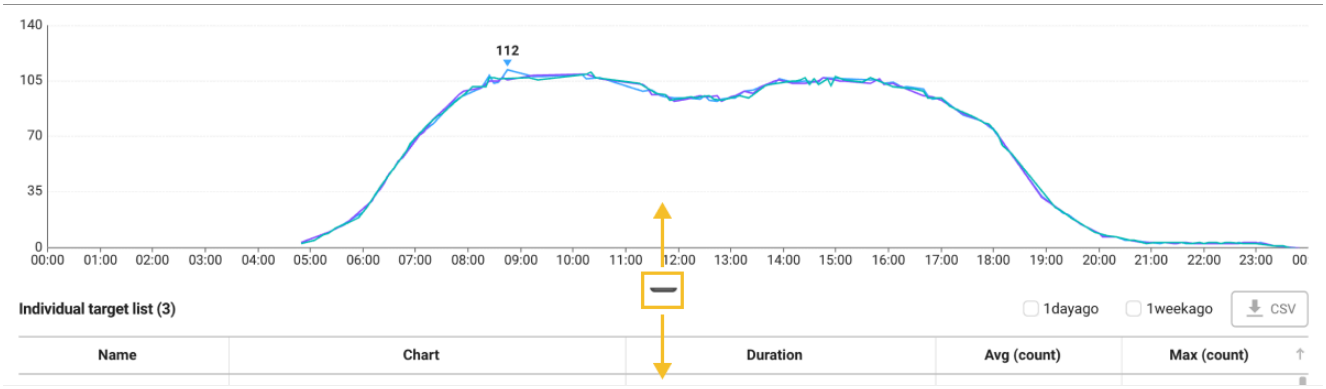

2. Select the desired targets according to the instructions on the screen.
3. Select **Apply**.

## 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.

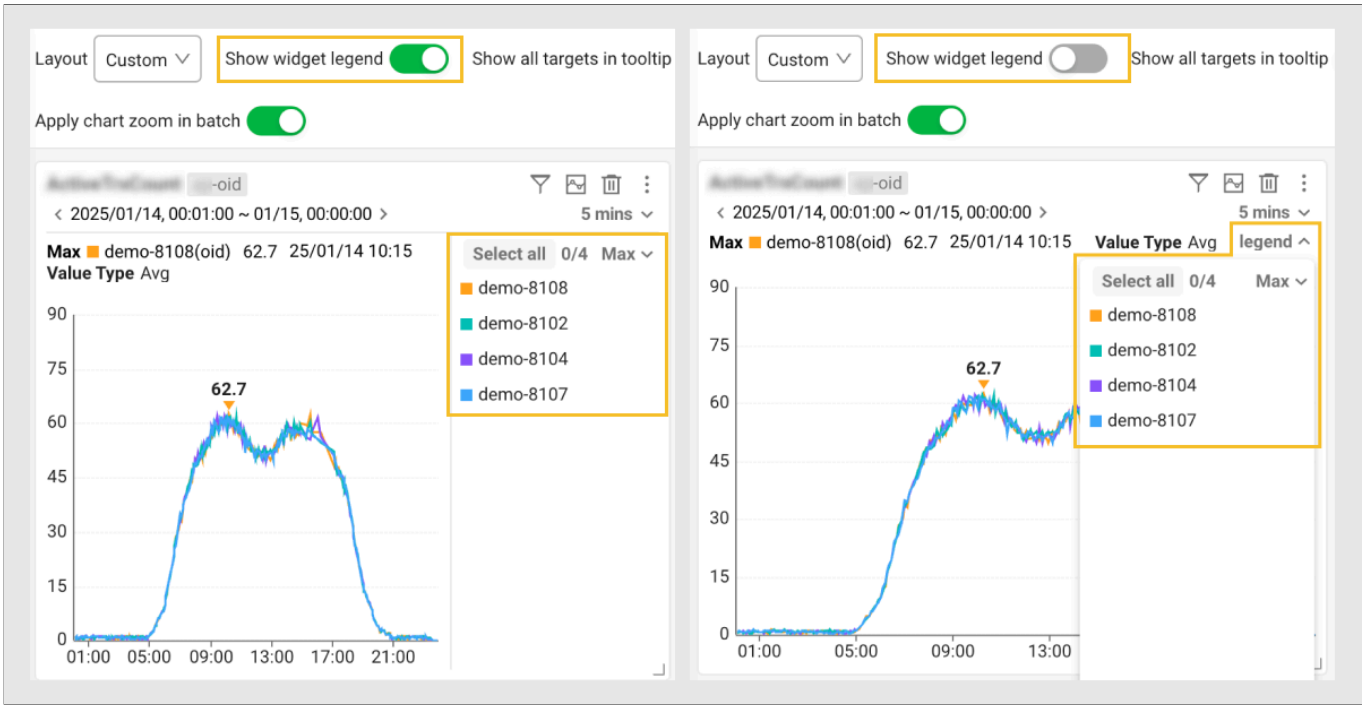



## Removing the chart widget

To remove any chart widgets placed on the screen, select .

## Using the legend

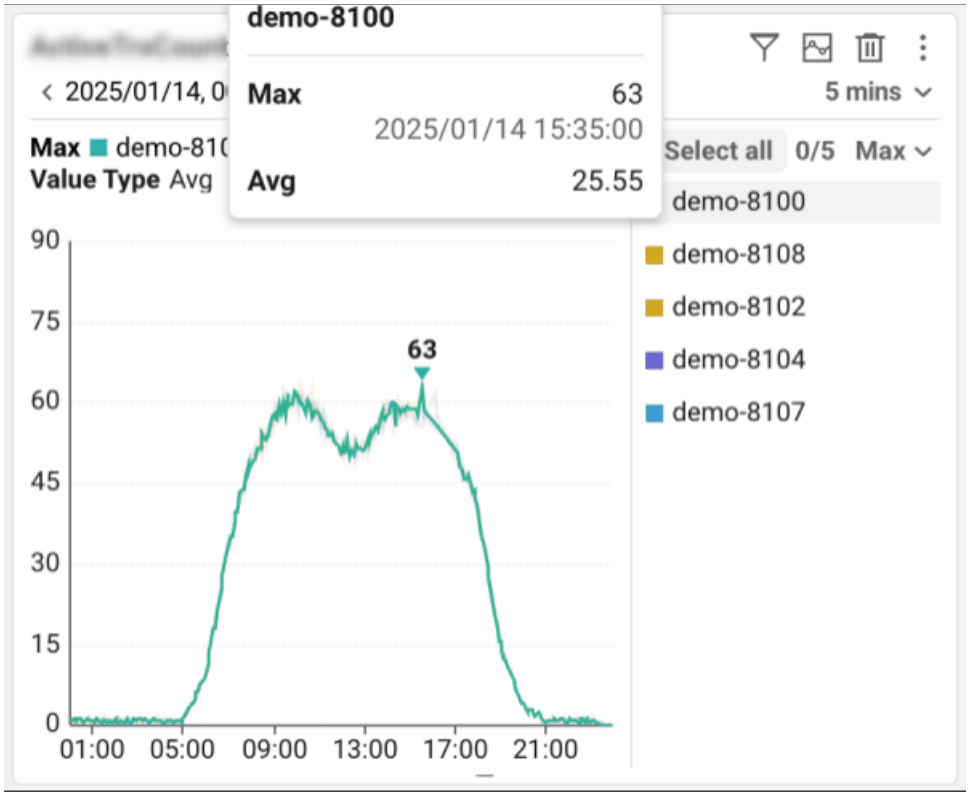
You can check the monitoring targets displayed on the chart through the legend and then sort or filter them by desired criteria. You can show or hide the legend on the right of the widget's chart by turning on or off the **Show widget legend** option at the top of the screen. With the option turned off, you can check and select the monitoring targets by clicking **legend**.



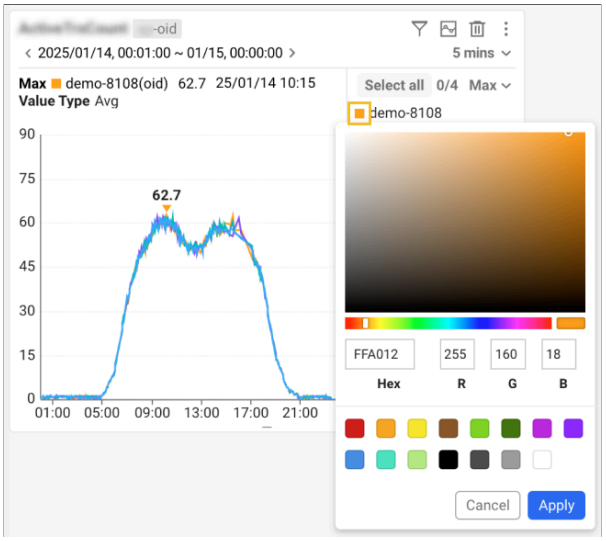

- You can select or unselect monitoring targets by selecting **Select all**/**Deselect all**.
- In the legend list, you can sort monitored objects by name, maximum value, or average value.




- 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



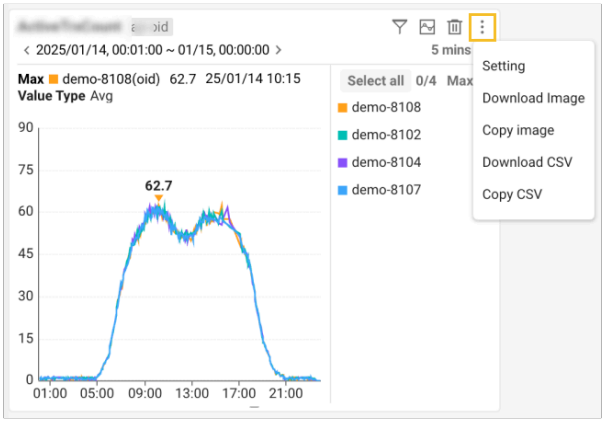

You can change the graph color corresponding to a monitoring target. In the legend list, select the square box

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.

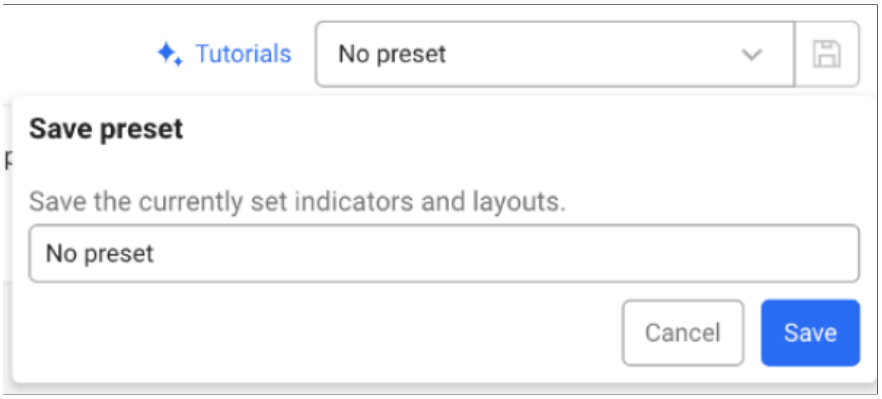
# Setting presets

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.

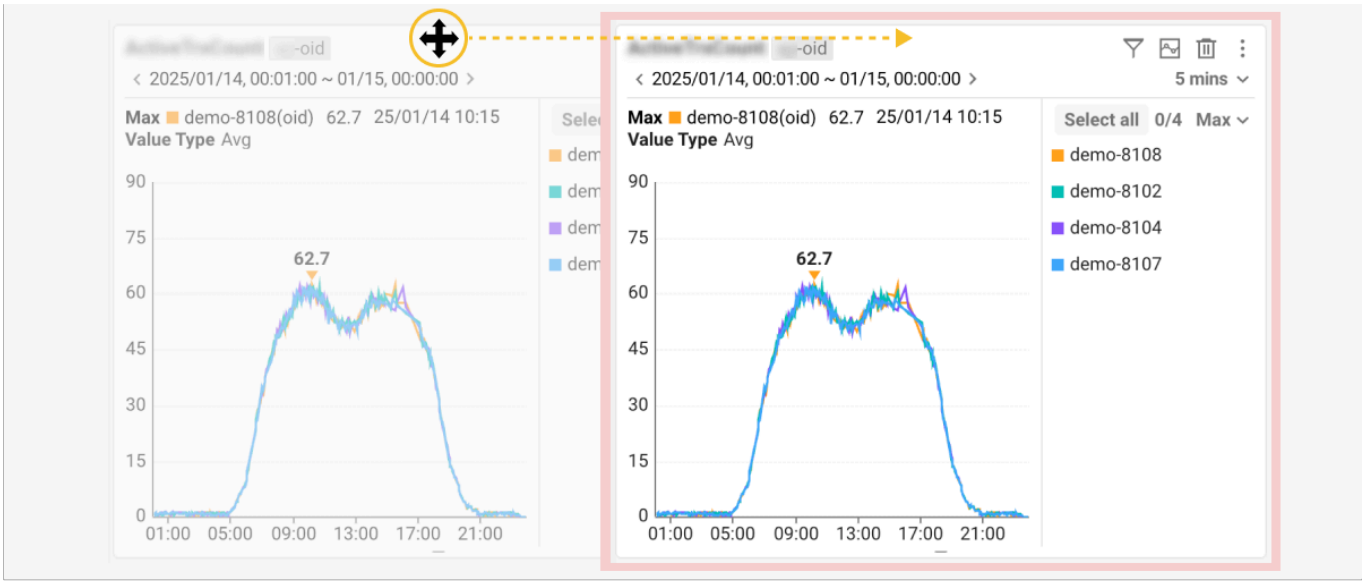
### Deleting presets

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.

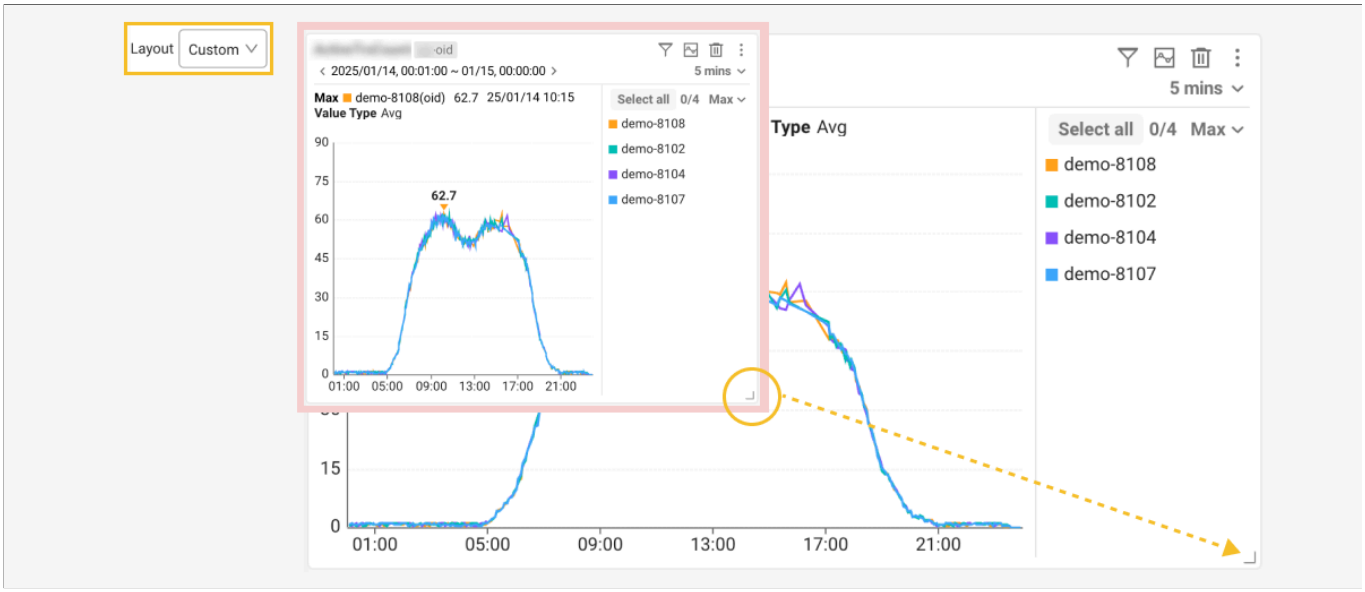
### Moving a widget




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.

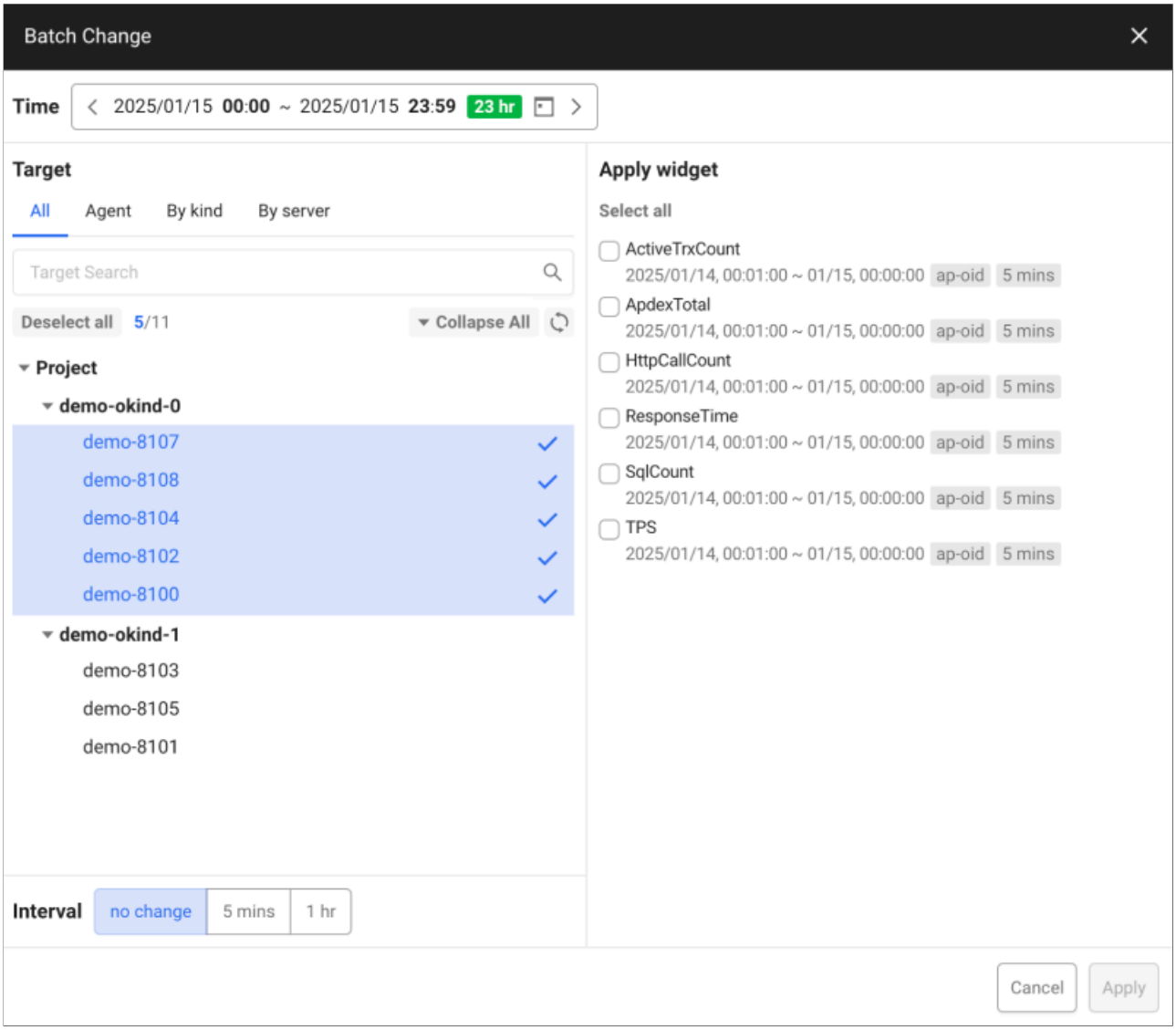
# Using the dashboard options

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.

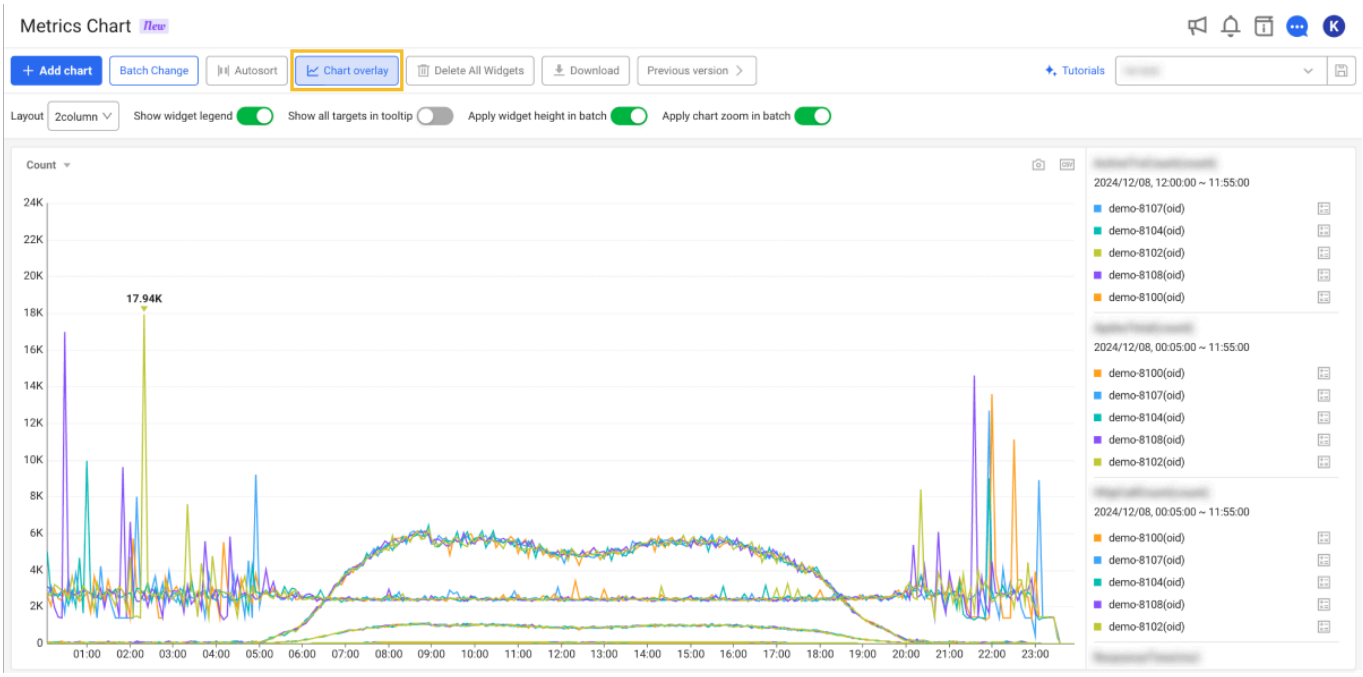
- **Apply widget**: Select a chart widget to apply the options to change. You can apply changed options to the selected widget at once.

## 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.

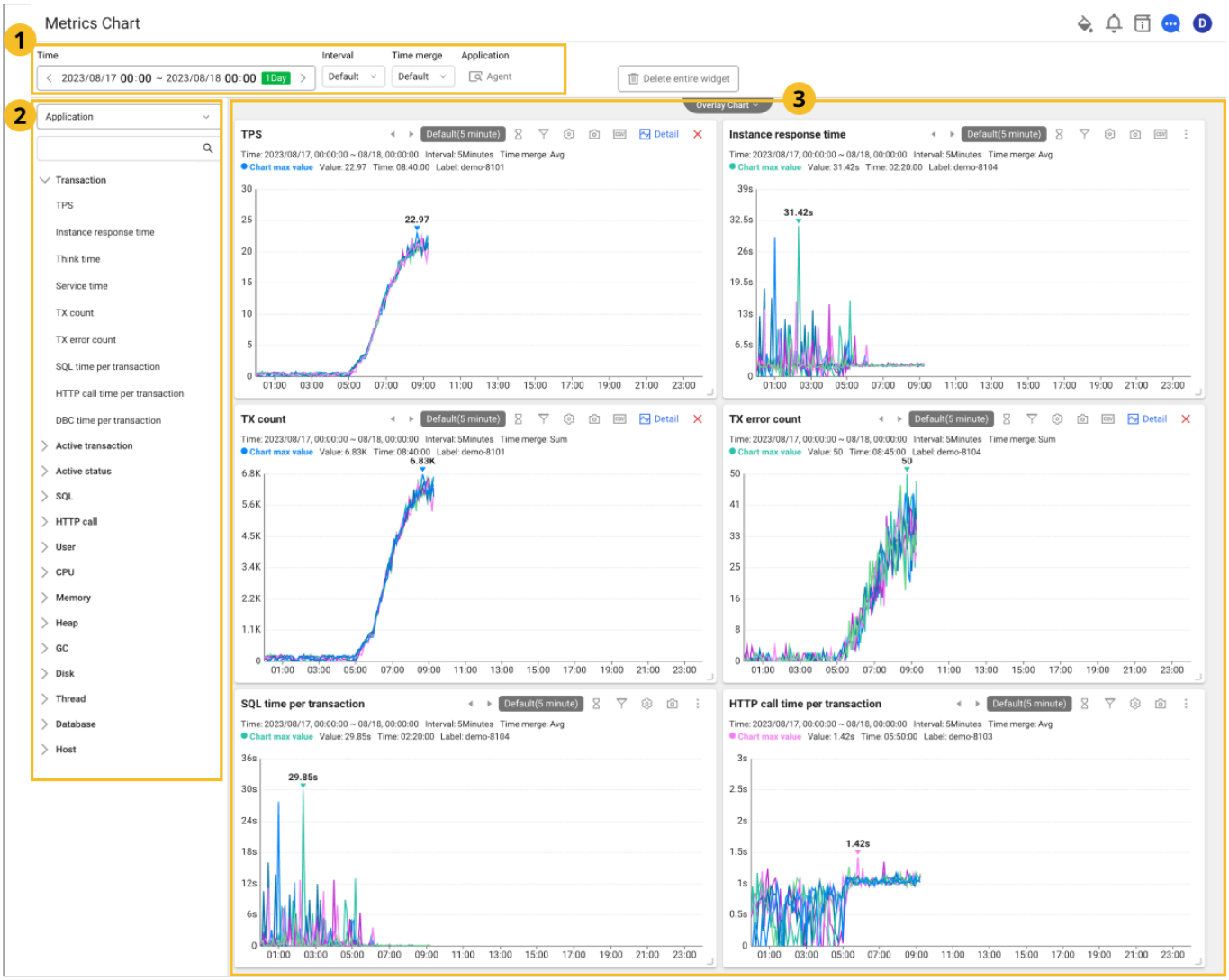
# Using the tutorial

When you select **Tutorials** on the upper right of the screen, you can check the changes after the Service 2.8.0 update.

# Metrics 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.

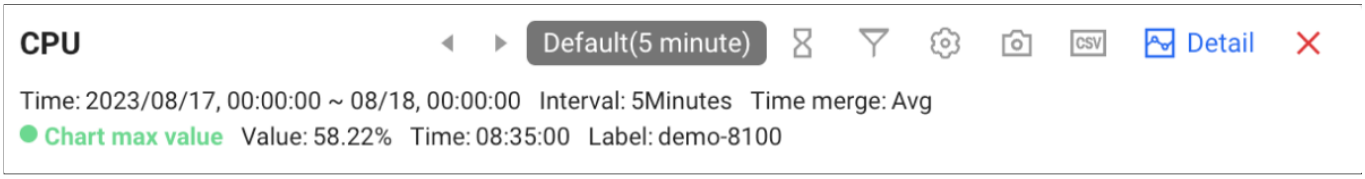
## Metrics list

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.

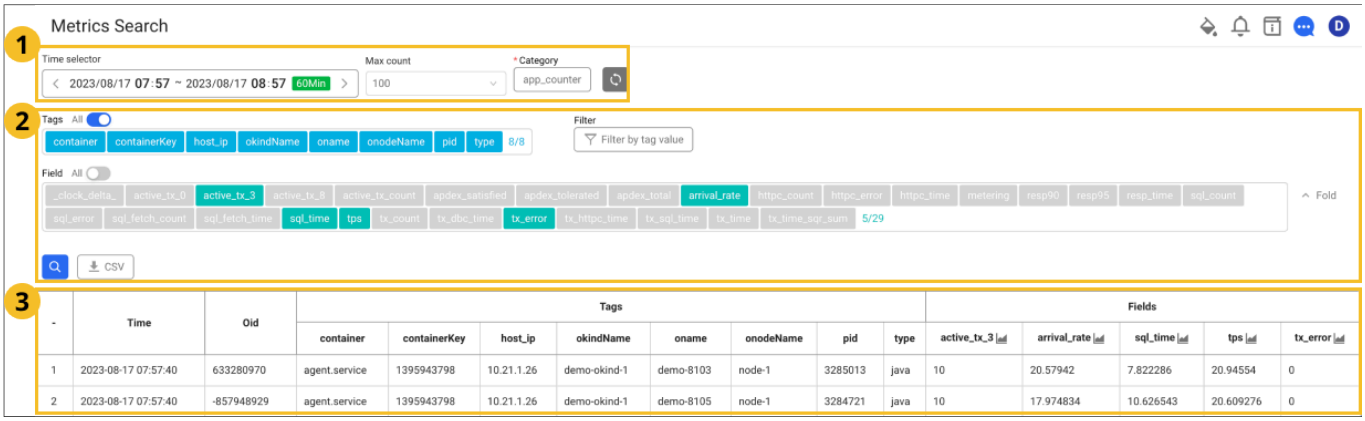
## Deleting a widget

To delete all widgets placed on the screen, select 🗑 **Delete All Widgets** on the screen.

# Metrics 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.

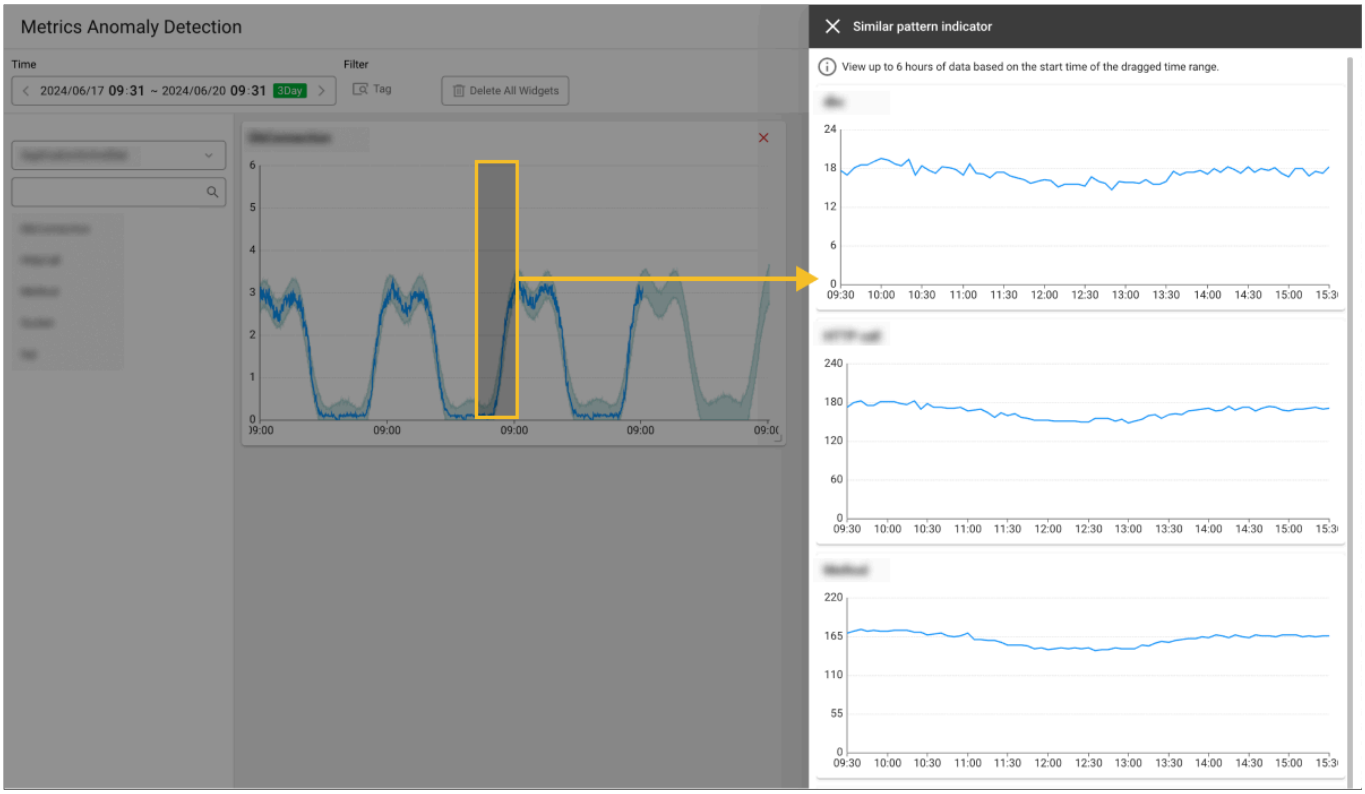
Place the selected metric to the right of the screen.

- 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.

# Transactions

## 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
```

# Analyzing The Transactions

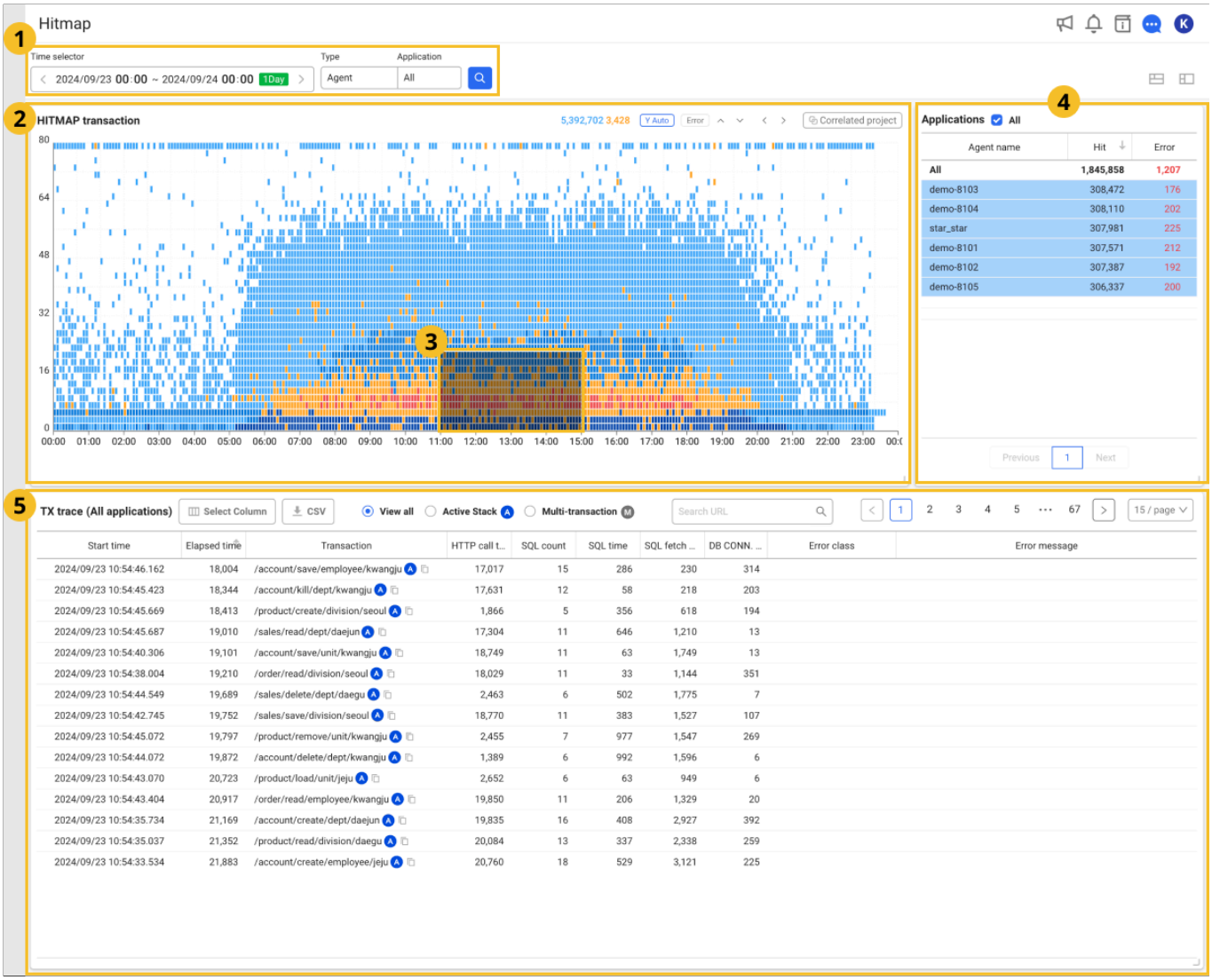
## Hitmap

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

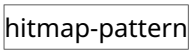
You can check the numbers of transactions and errors included in the selected area on the chart. If you select the desired application in the list, the information details of the application appear in 5 **TX trace**.

### 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.




## 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:

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

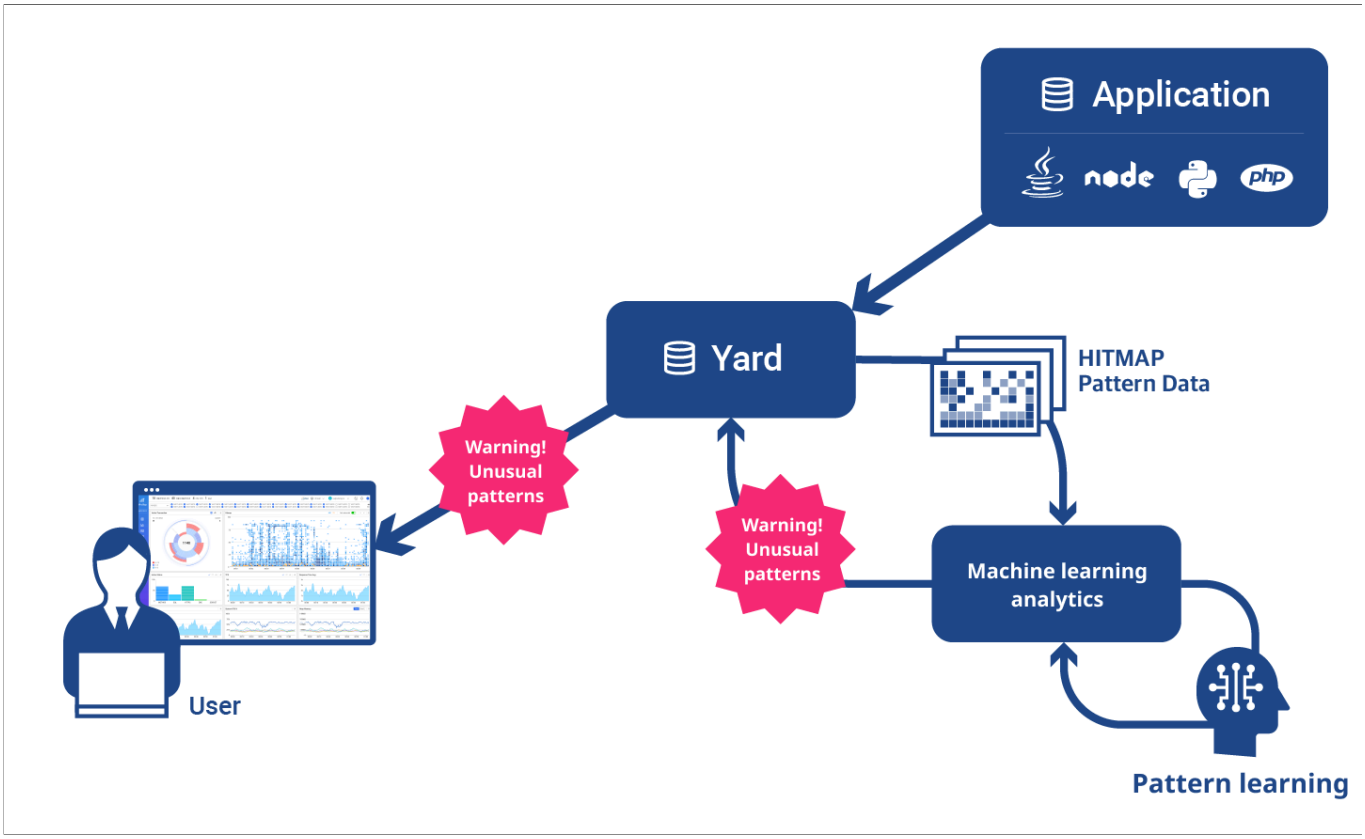
A line in the transaction response distribution indicates a bottleneck. If the lock is temporary, a vertical line is generated. If the bottleneck is release by timeout, a horizontal line is generated.

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

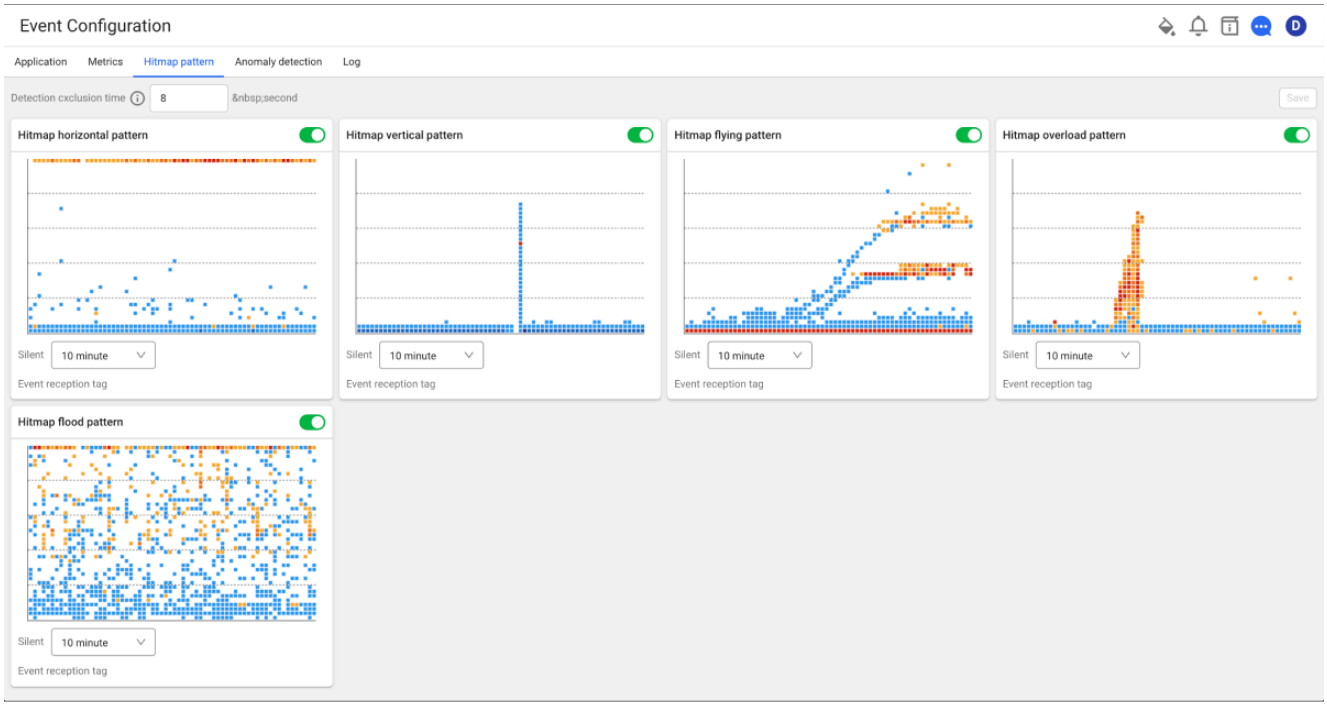
- **Horizontal line pattern**

  hitmap-abnormal2

- **Composite pattern**

  hitmap-abnormal3

- **Hitmap alarm**




For more information about the analysis method of the **HITMAP transaction** chart, see the following.

# Transaction Tracing

## 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.

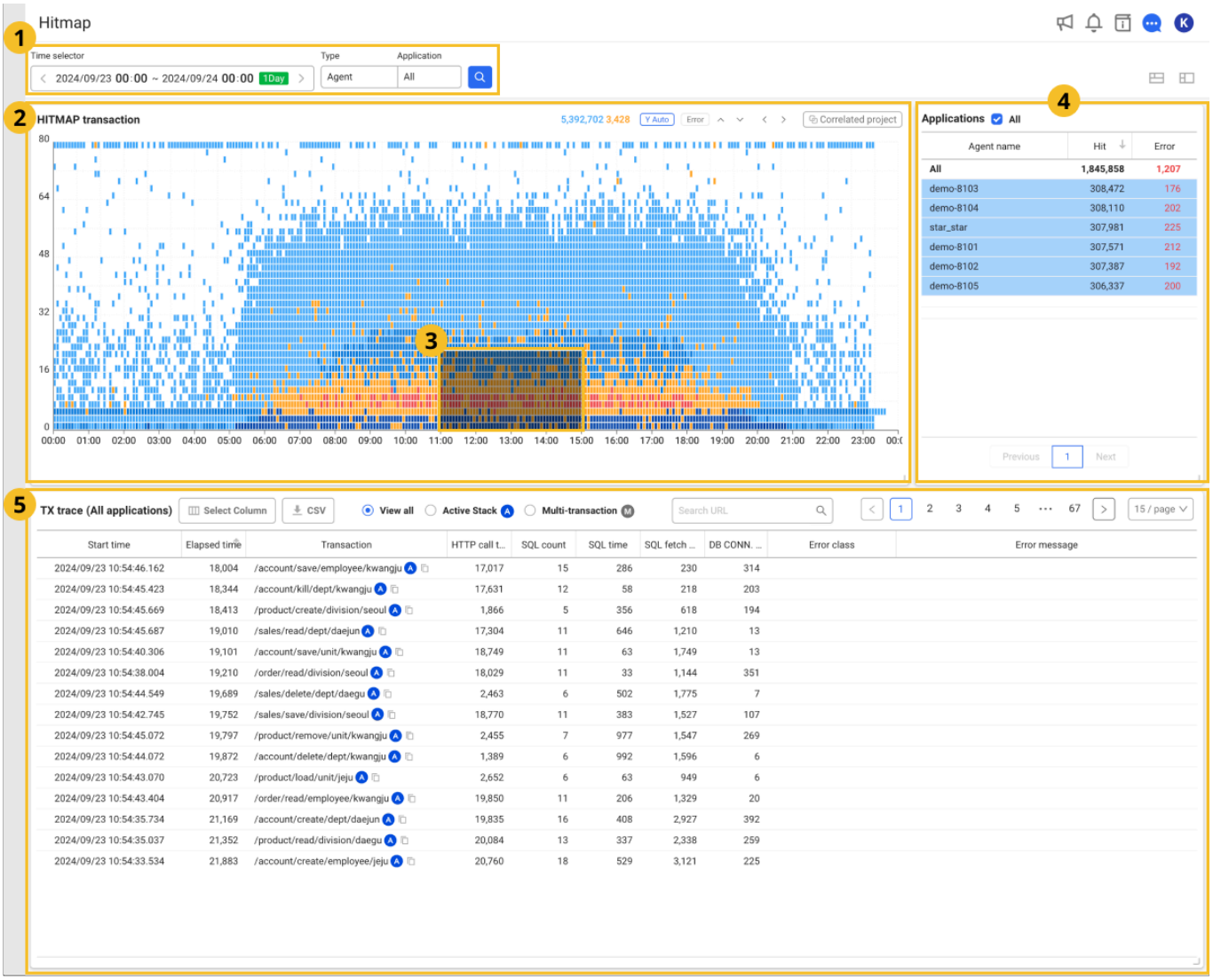
- **ACTIVE STACK Step** START-END

  It includes the active stack information. It is created and collected by a separate thread and it is added to the transaction trace.

> In the WhaTap monitoring service, **Step** is used as the same meaning as **Span**.

# Detailed analysis

Home > Select Project > **Analysis** > **Hitmap**



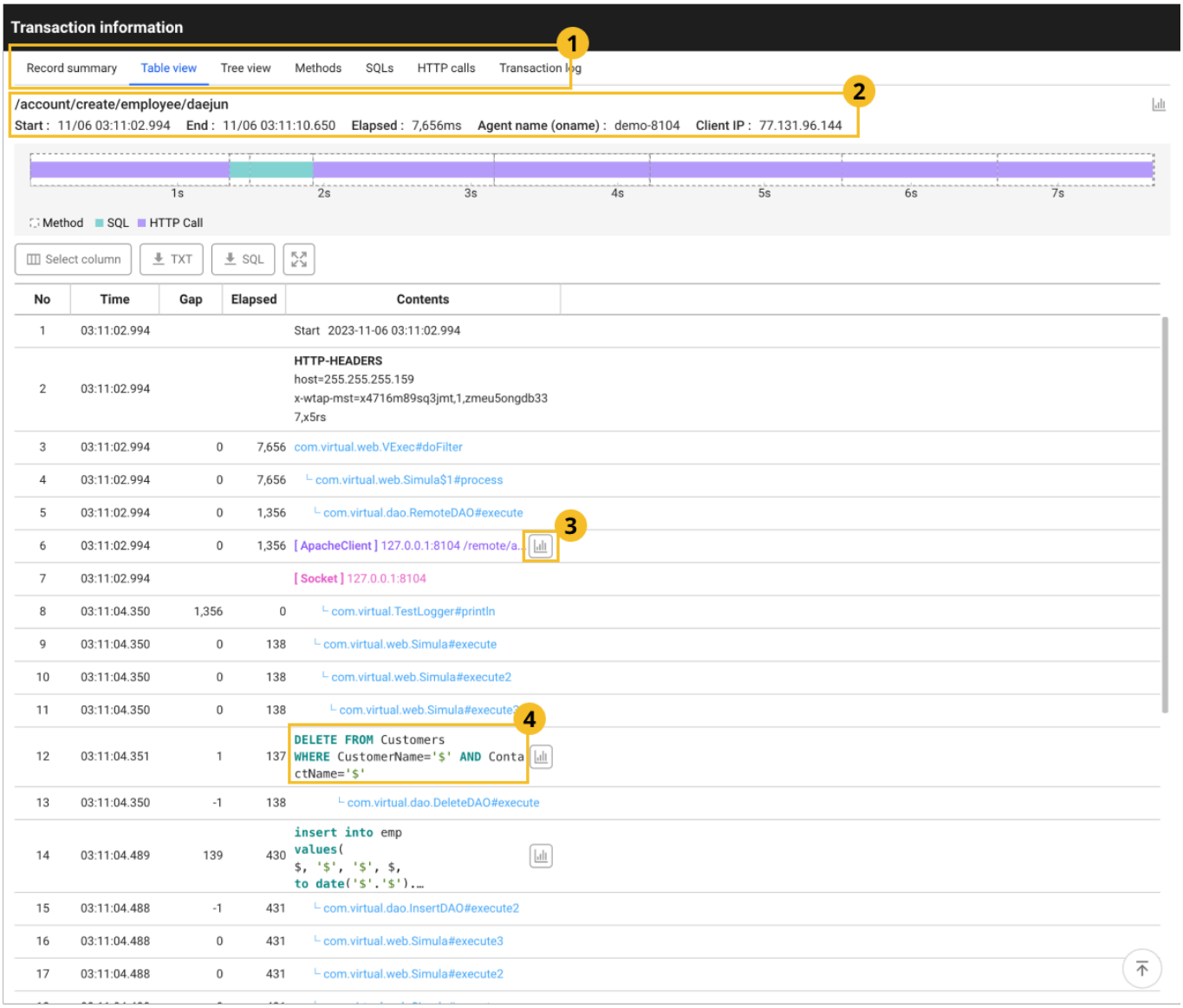

1. In 1 **Time selector**, select a time period to see the hitmap.
2. In 1 **Type** and **Application**, select the targets to view.

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



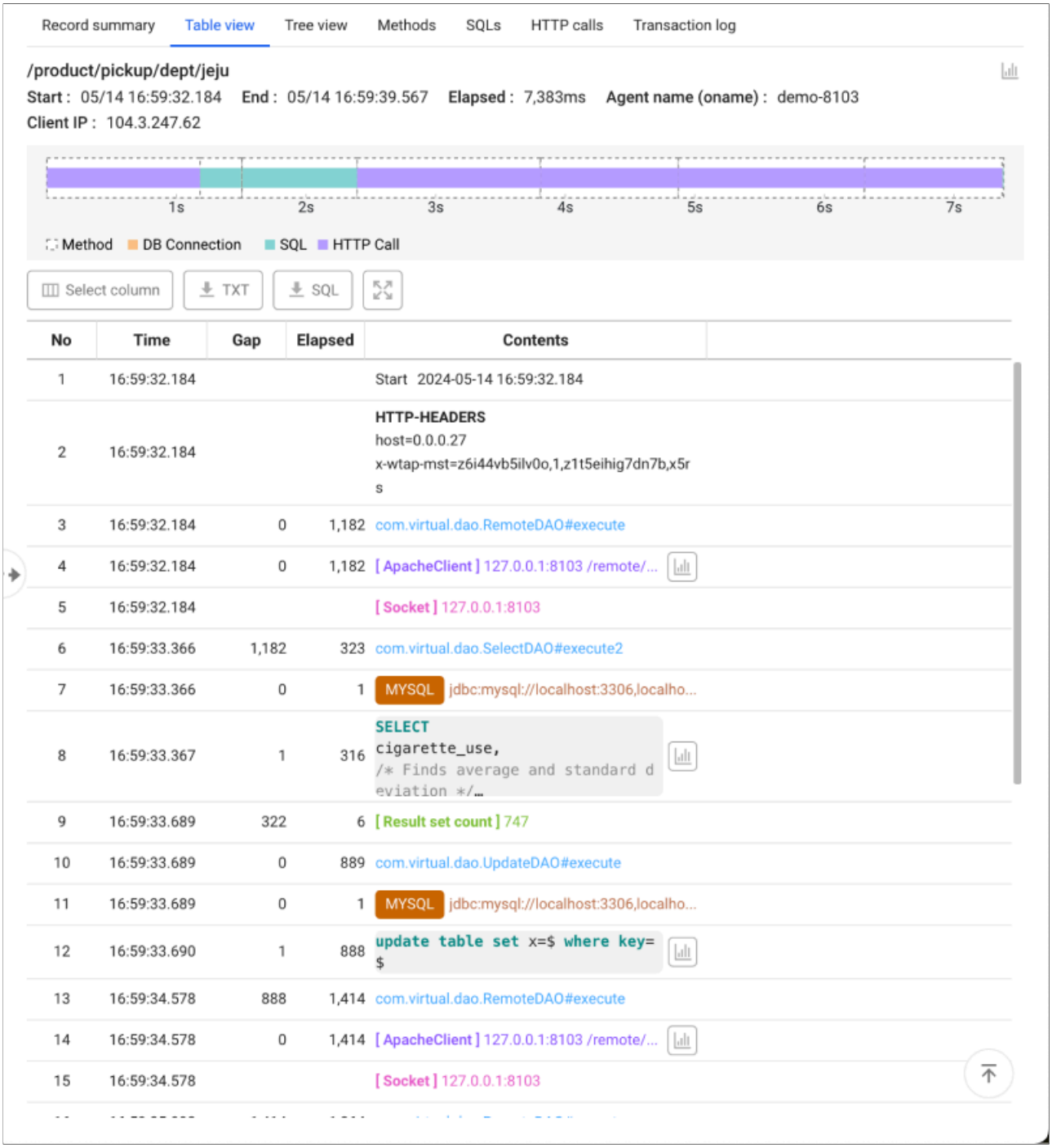

- In the 1 area, the step data can be seen in various formats. Select a desired type of tab.
- In the 2 area, you can see the basic information for transactions.
- 3 : The statistical data window for the corresponding URL or each execution section appears. Click a desired time on the graph chart in the 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.

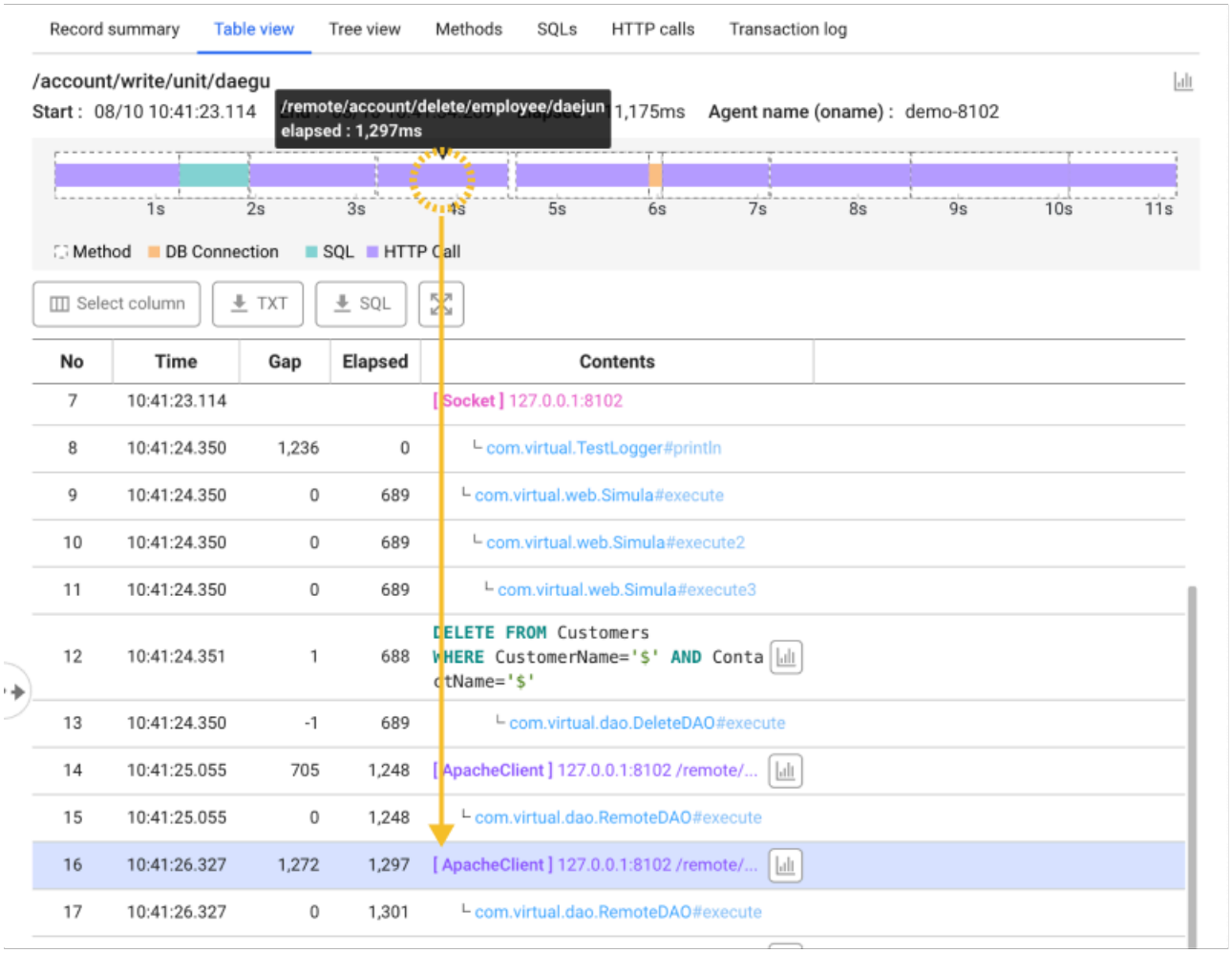
- If you select 4 SQL step, the SQL variables and HTTP queries can be viewed. For more information, see the following.

## 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.



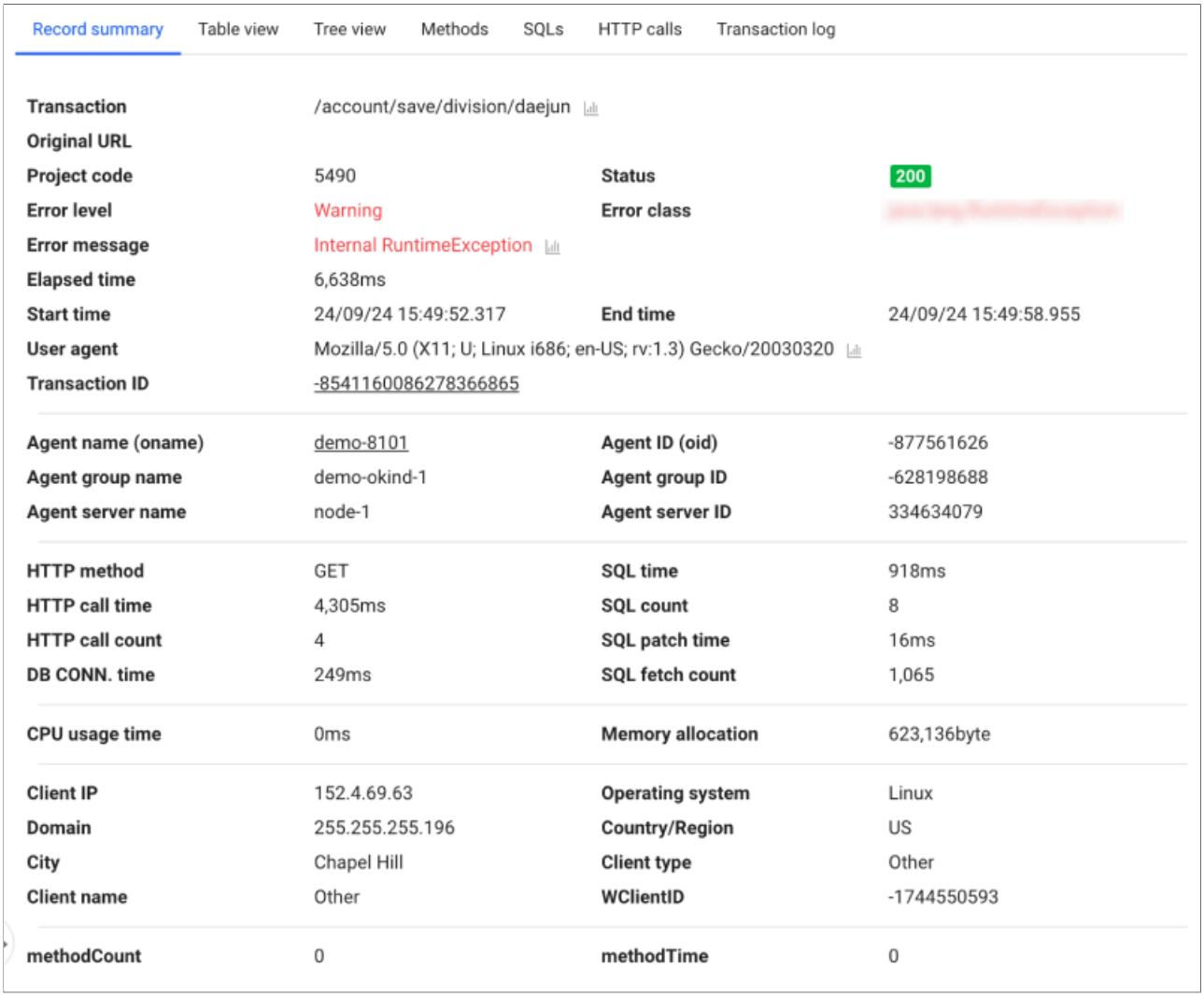

- **Select column**: You can add or hide the memory accumulation and CPU accumulation data as columns in the table list. The columns are as follows:
  - **No**: Step occurrence sequence
  - **Time**: Start time of each step
  - **Gap**: The waited time between the start time of the previous step and 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.

## 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. |
| | 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. |

| Classification | Properties | Description |
|---|---|---|
| | 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 |
| | Agent ID (oid) | It is a unique ID of the agent. |
| | Agent group name | Agent group name set with the `okind` option. |
| | Agent group | It is a unique ID of the agent group. |

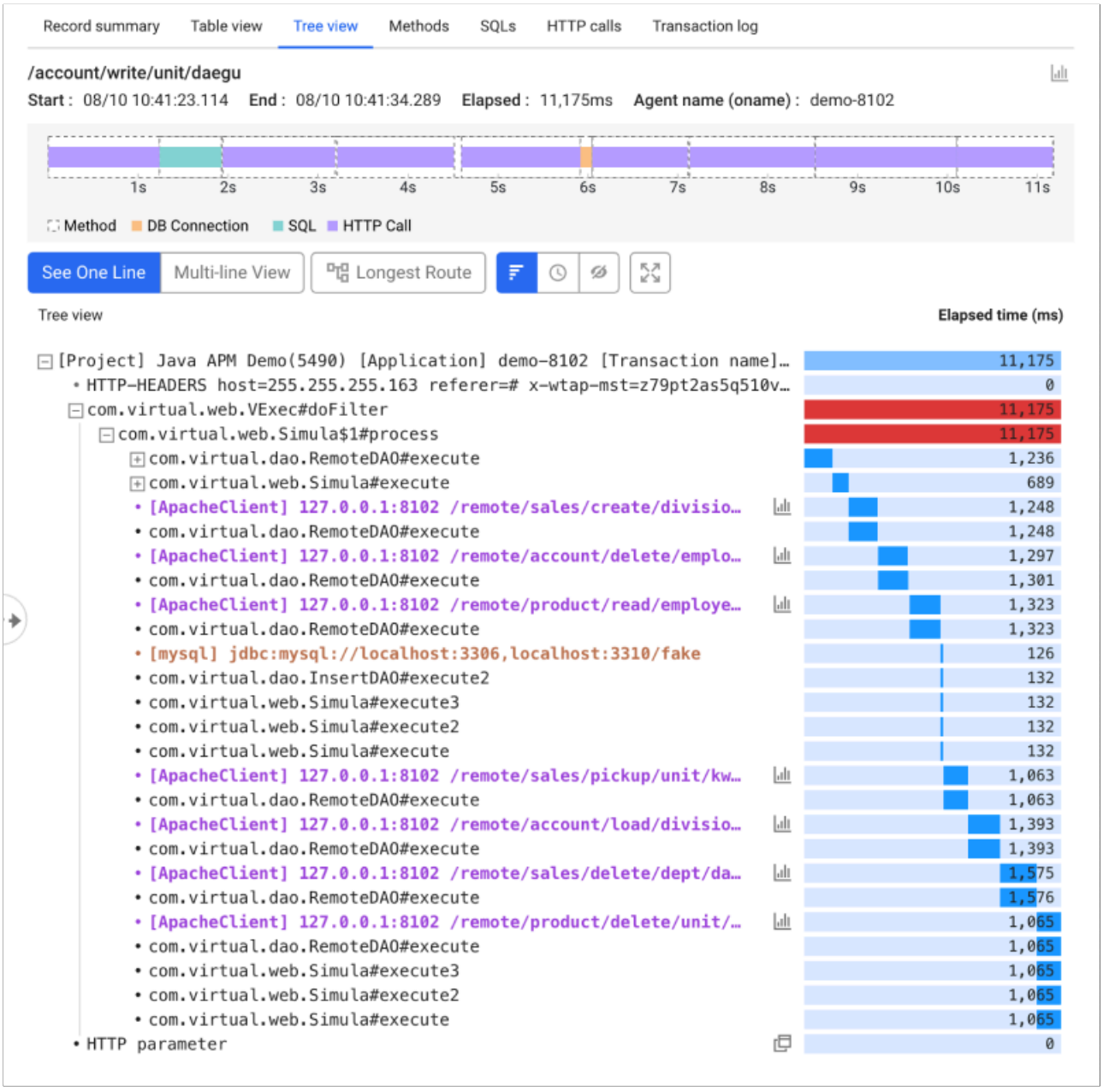
| Classification | Properties | Description |
|---|---|---|
| | ID | |
| | Agent server name | Agent server name set with the `ondoe` option. |
| | Agent server ID | It is a unique ID of the agent server. |
| **Step** | HTTP method | HTTP method. (GET, POST, PUT, HEAD, etc.) |
| | HTTP call time | Outbound 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 time | CPU time used for processing transactions. |
| | Memory allocation | Amount of memory used by the transaction. |
| **Containers** | Pod name | It is the name of the Kubernetes Pod that processed the transaction. |

| Classification | Properties | Description |
|---|---|---|
| | 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.



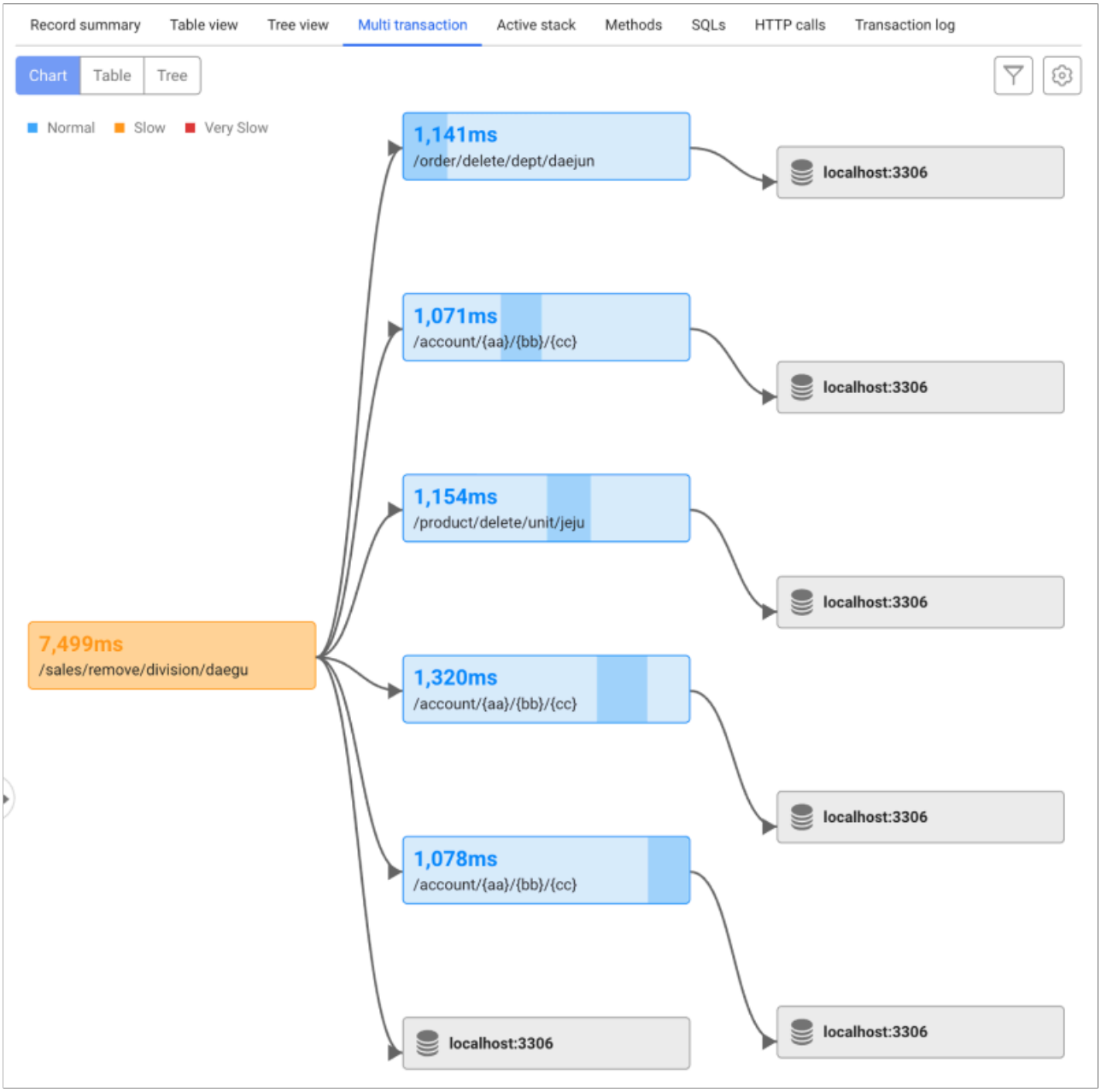
- **Single-line View**: You can arrange the tree format at intervals by displaying the texts displayed in the execution data for each section in a single line.
- **Multi-line View**: All the texts displayed in the execution data for each section, appear by wrapping words.
- **Longest Path**: The longest 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.

## 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.



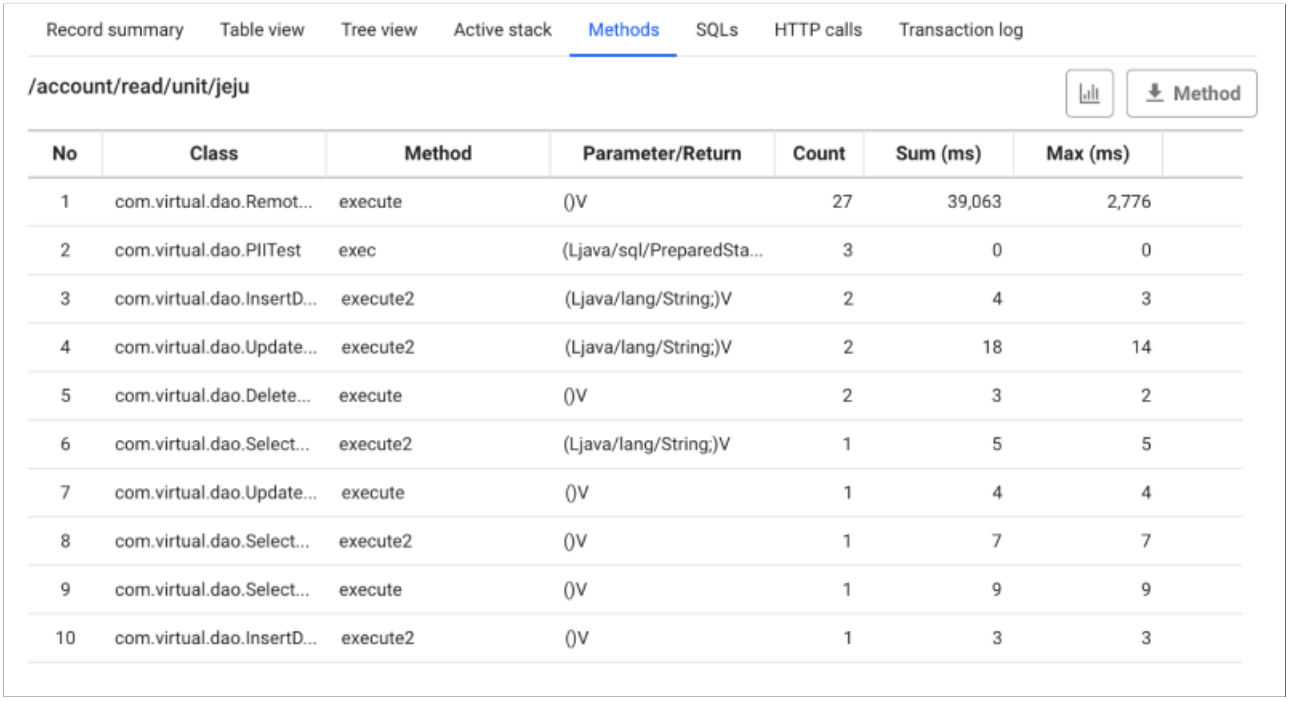

- **Chart**: Provides the call relationships for each transaction in a flowchart. Select a transaction node to see the trace analysis for the transaction node. In **Chart** mode, you can use the mouse to go to a desired position or zoom in or out by scrolling.
  - ⚙ **Chart view settings**: You can display 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 |

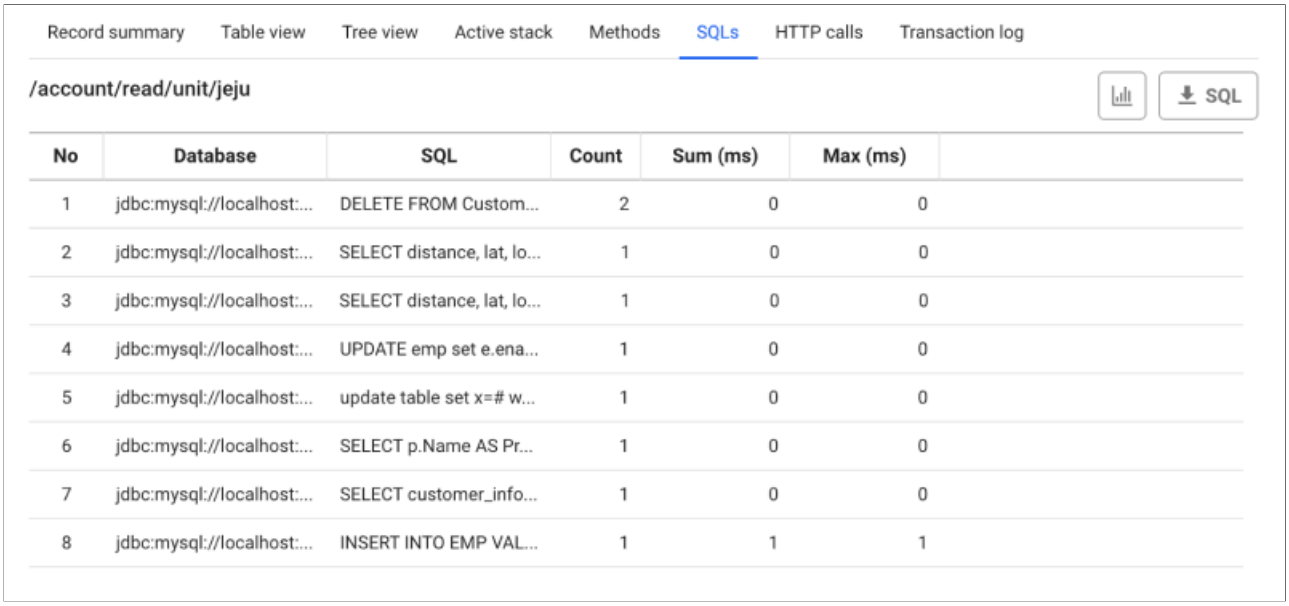
To see only the method data, select the **Methods** tab. It displays the method name and step time for which tracing has been set on the agent. When unnecessary logics are repeatedly executed, it can be used as analysis data to improve the method logic by finding the case where the execution takes longer 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

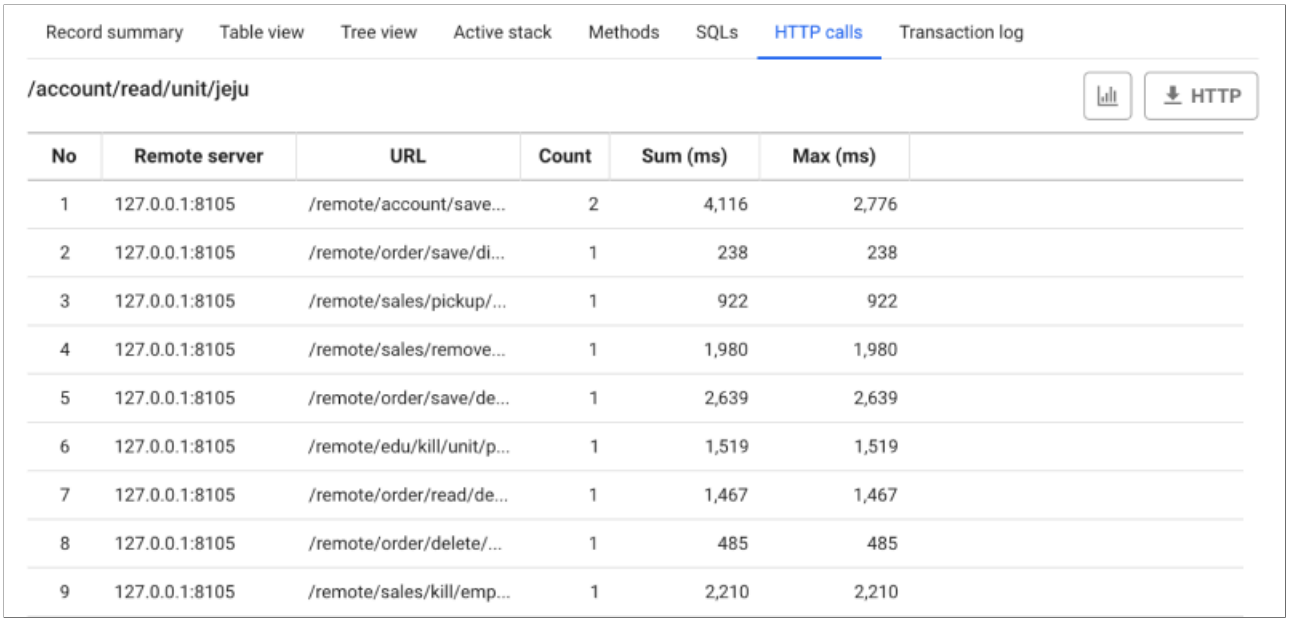



| No | Database | SQL | Count | Sum (ms) | Max (ms) |
|---|---|---|---|---|---|
| 1 | jdbc:mysql://localhost:... | DELETE FROM Custom... | 2 | 0 | 0 |
| 2 | jdbc:mysql://localhost:... | SELECT distance, lat, lo... | 1 | 0 | 0 |
| 3 | jdbc:mysql://localhost:... | SELECT distance, lat, lo... | 1 | 0 | 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



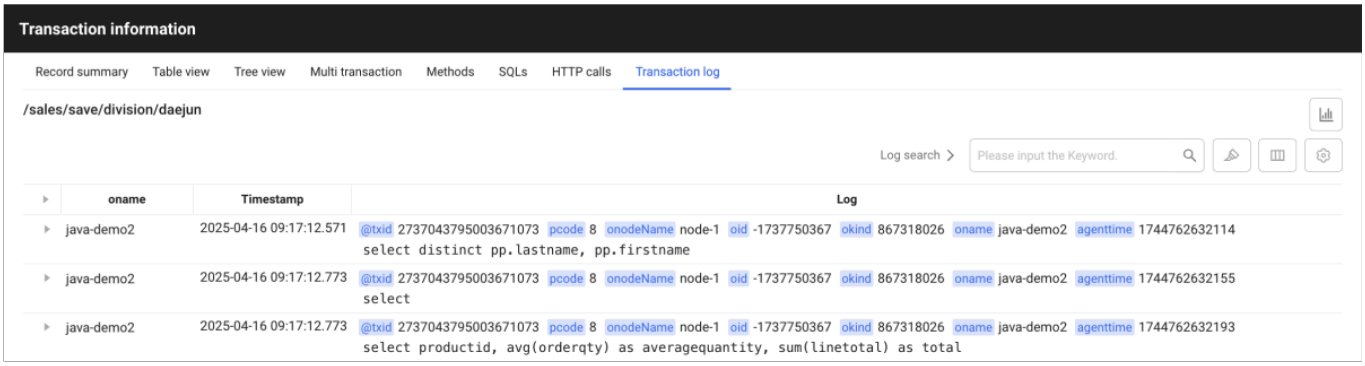

| No | Remote server | URL | Count | Sum (ms) | Max (ms) |
|---|---|---|---|---|---|
| 1 | 127.0.0.1:8105 | /remote/account/save... | 2 | 4,116 | 2,776 |
| 2 | 127.0.0.1:8105 | /remote/order/save/di... | 1 | 238 | 238 |
| 3 | 127.0.0.1:8105 | /remote/sales/pickup/... | 1 | 922 | 922 |
| 4 | 127.0.0.1:8105 | /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.

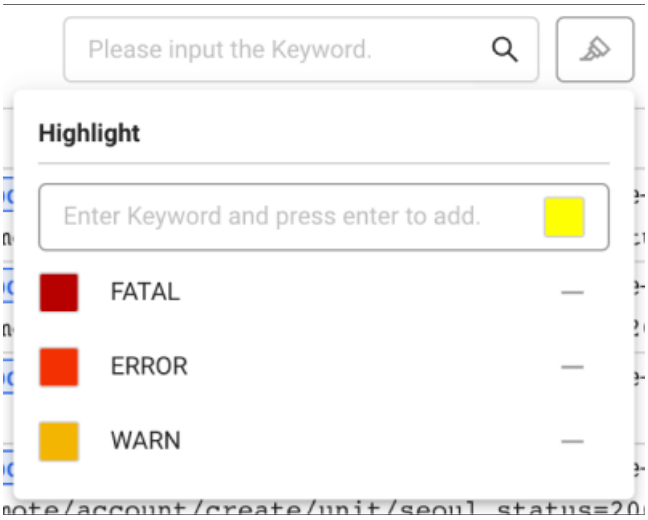



- : You can see the **Transaction statistics** information. If a graph on the chart is selected, the **Analysis** > **Transaction Search** menu appears. You can see the list of transactions for the selected time zone.
- **Log Search**: You can go to Log Search to easily find the desired log data.
- **Keyword 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.

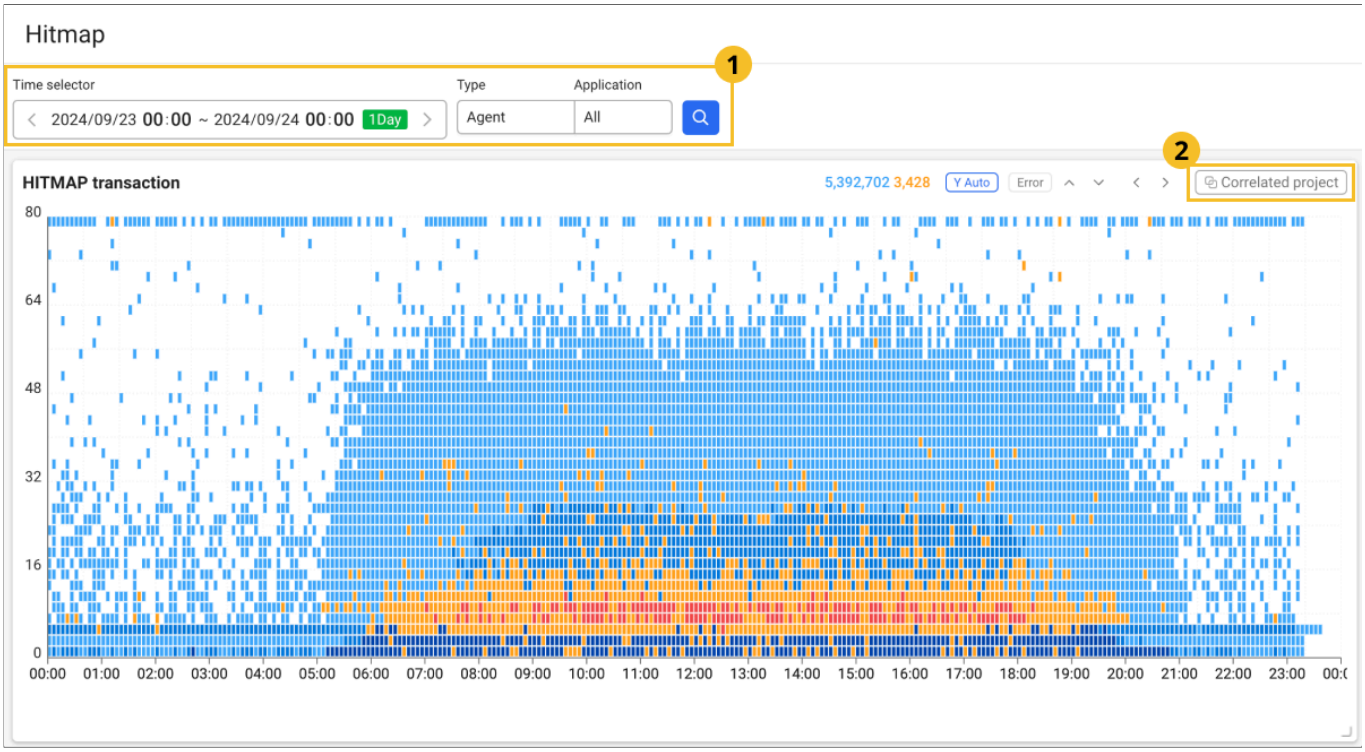
> ⓘ For more information about **Log view detail setting**, see the following.

# Additional options

## Checking the correlated project metrics

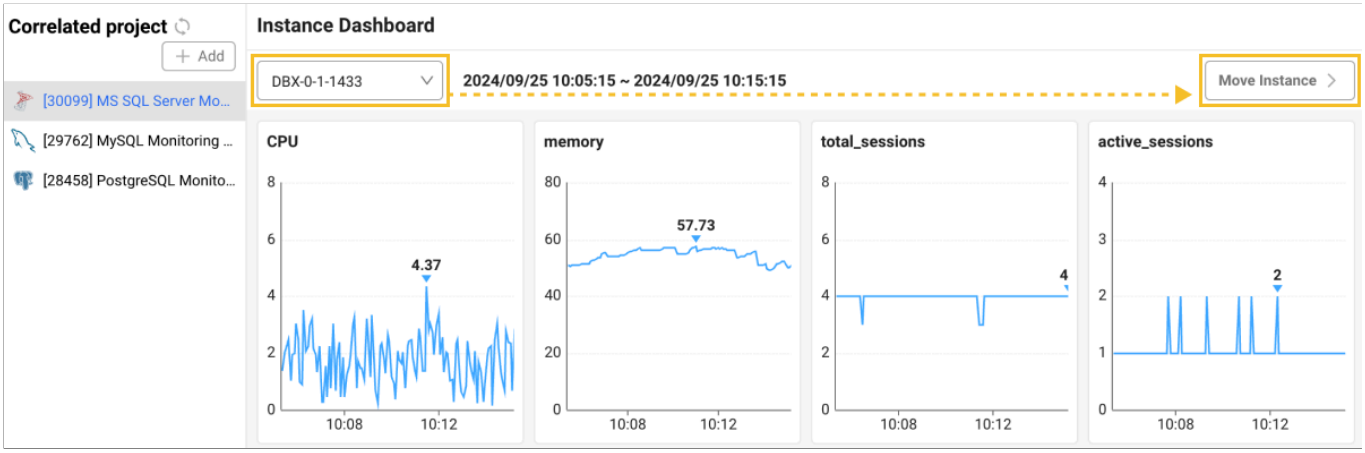
If you have created a database monitoring (DPM) project and an application monitoring (APM) project, you can relate the two projects for monitoring the data collected from DPM in the APM project. You can see performance metrics of the correlated projects while querying transactions via the **Correlated 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.

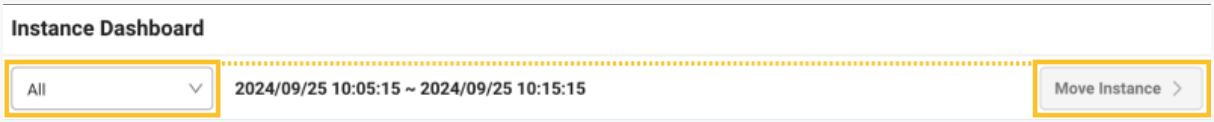
Select the correlated project from the left list and then check the metrics in the **Instance dashboard** section.




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.

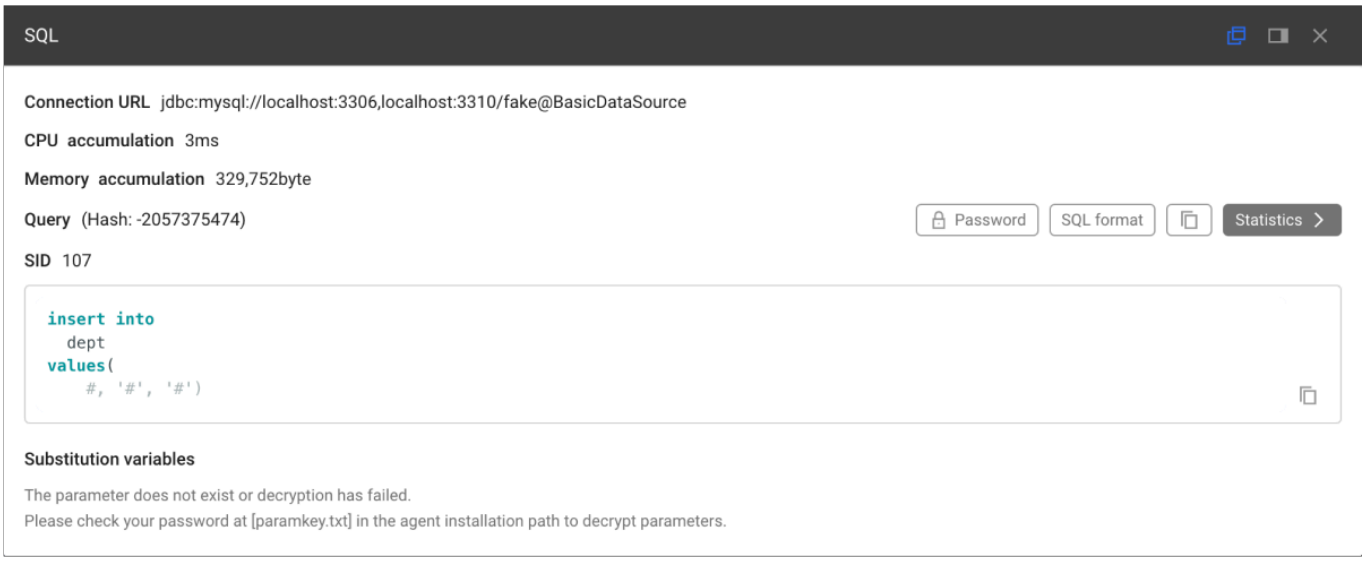
1. Scroll down the page to the final step of the transaction.
2. Select the **HTTP parameters** item.
3. If the **HTTP-PARAMETERS** window appears, select **Password**.
4. Enter the specified Param Key. You can see the hidden parameters.

- 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 ( `paramkey` ) in the `WHATAP_HOME` /security.conf file. You can change it to another string. It is required to view the SQL variables and HTTP queries and to stop the threads.
- Depending on the agent version, the password configuration file may be different. Change the string in the security.conf file or paramkey.txt file.

## 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.

  > Depending on the agent version, the password configuration file may be different. Change the string in the security.conf file or paramkey.txt file.

- **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

**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.

> ⓘ **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

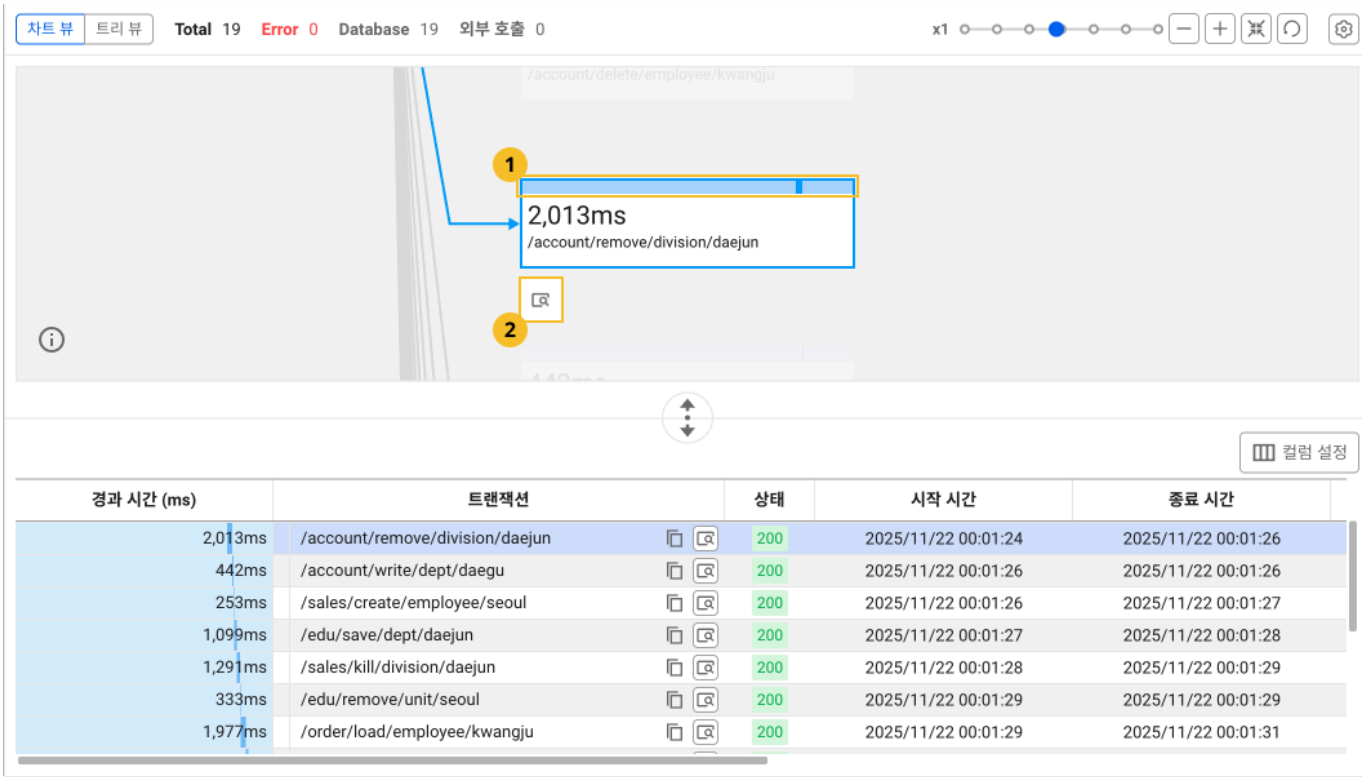
1. Go to **Analysis** > **Multiple Transaction Trace**.
2. Enter the **MTID** value checked in the **Transaction information** window into the **Search by MTID / CUSTID** field.
3. Select the date and project to query.
4. Click the **Apply** button at the bottom of the screen.
   - The **Apply** button 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.

# Transaction Step Collection Method

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

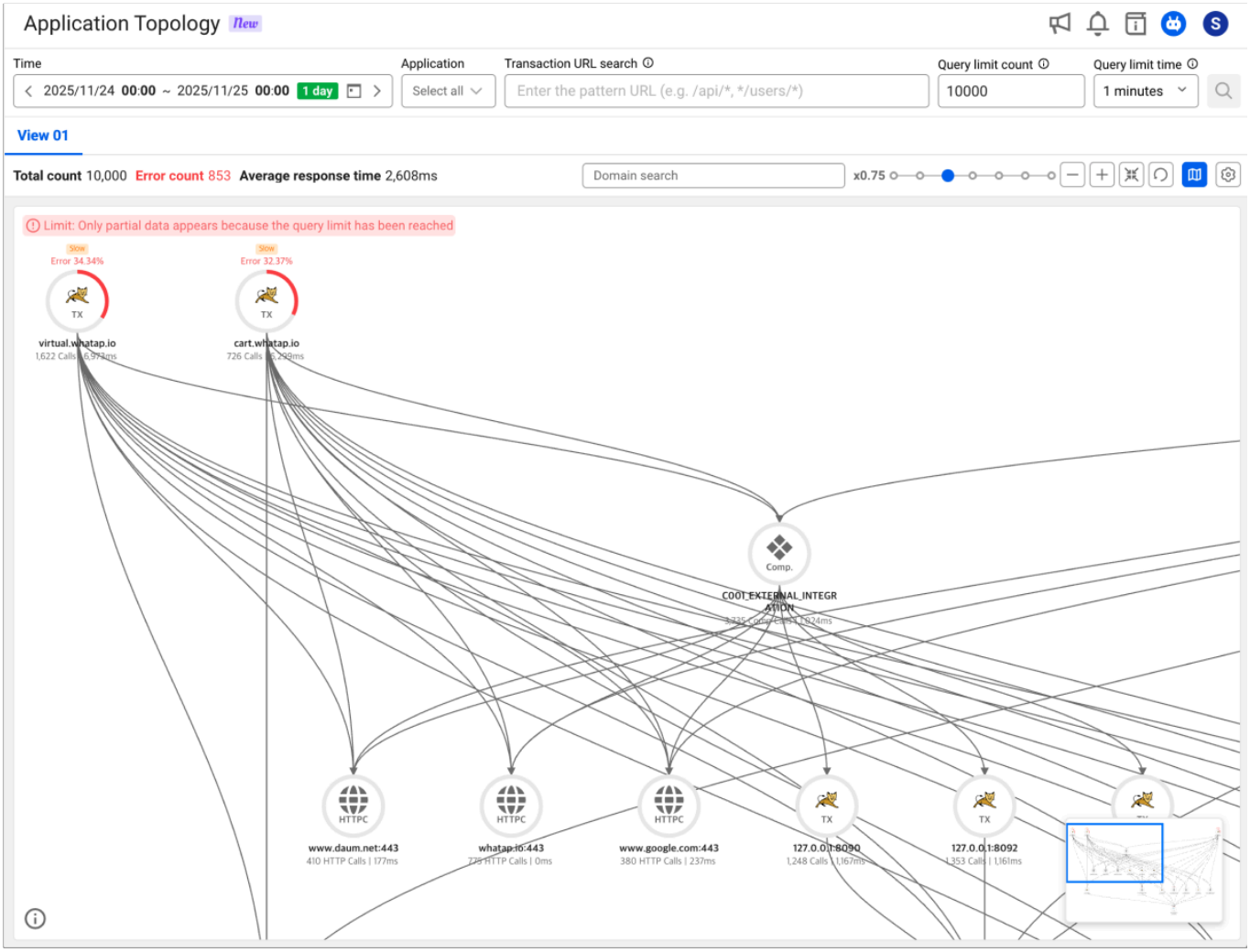
When the time limit is reached, only data up to that point is shown, and a warning appears.

- 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

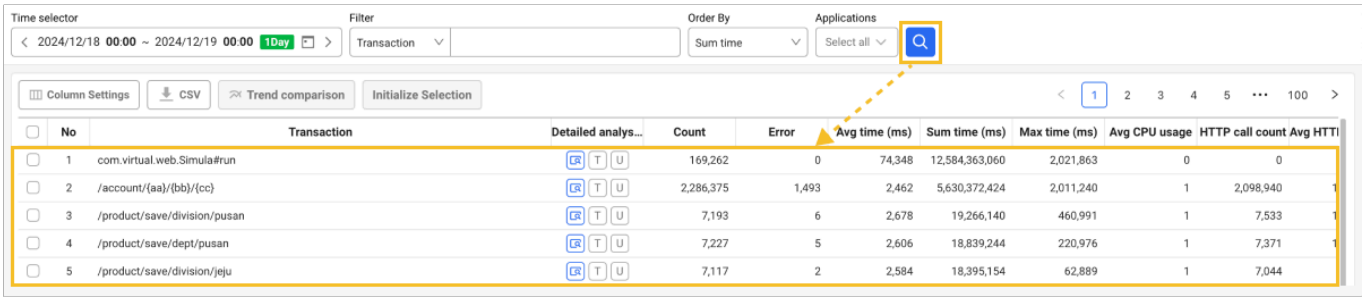
# Statistics

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.

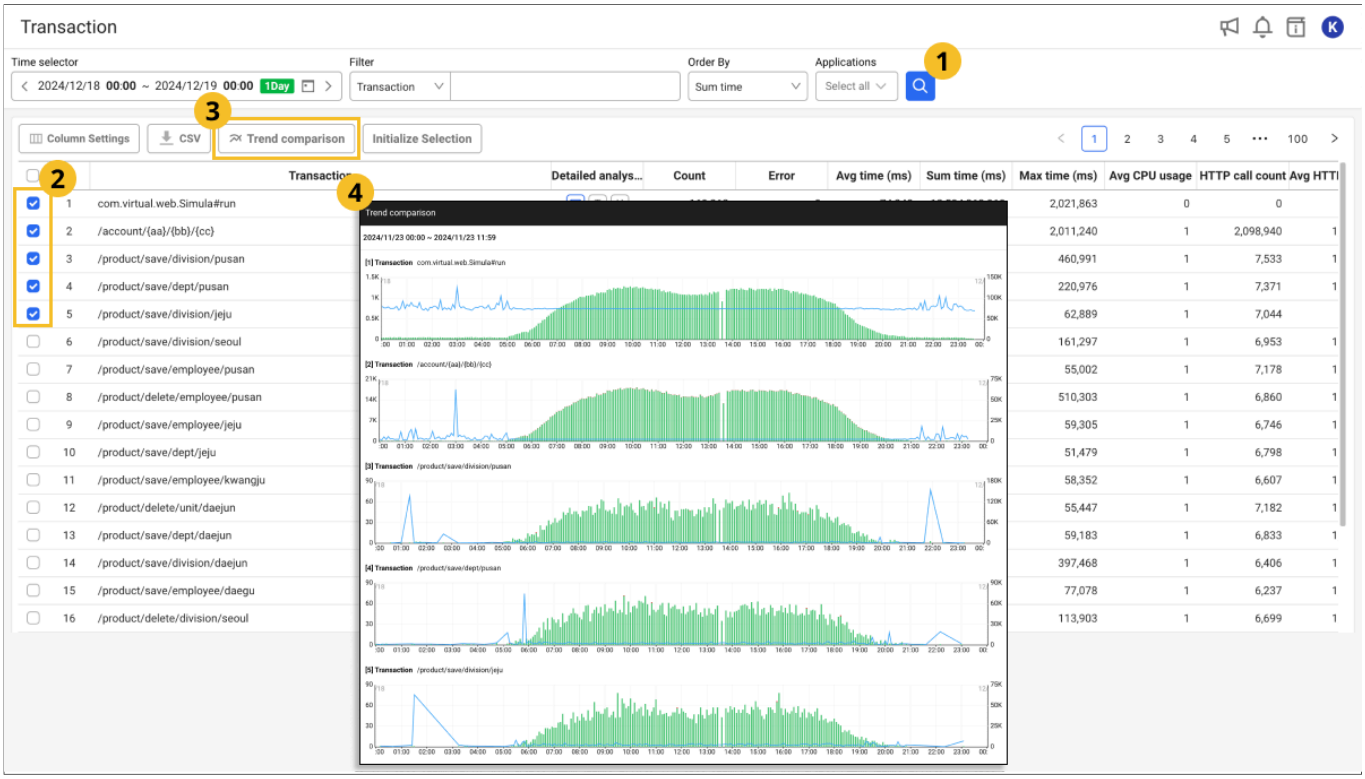
| Columns | Description |
|---|---|
| Transaction | Transaction address |
| Count | Number of execution cases |
| Error | Error occurrence count for the executed transactions |
| Avg time (ms) | Average of the total response time |
| Sum time (ms) | Sum of response times |
| Max time (ms) | Maximum response 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.

| 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. |
| 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 |
| 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 |

| Columns | Description |
| --- | --- |
| 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 |

| Columns | Description |
| --- | --- |
| 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 |
| 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 |

# Report

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.

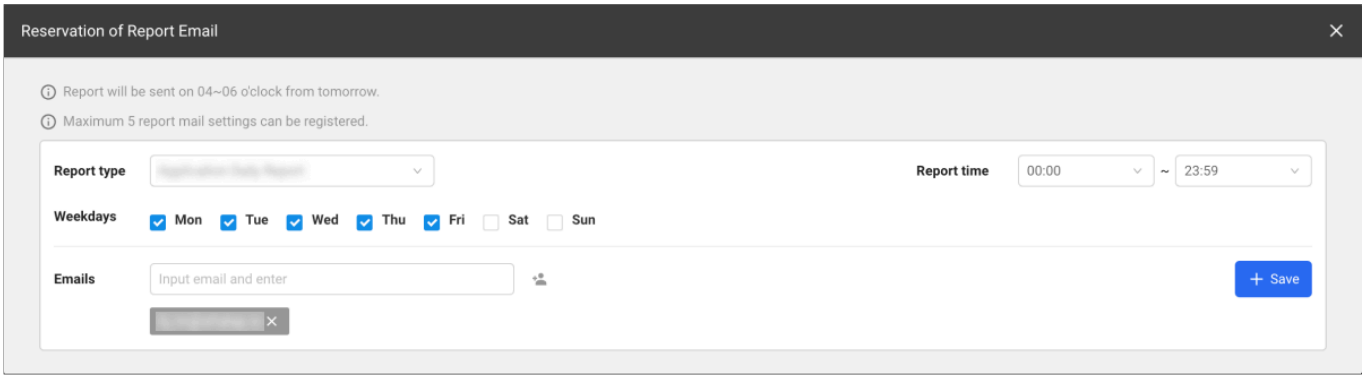
- Example file: After downloading, unzip the ZIP format to see the example file in the HTML format. Instructions to CSS and HTML areas for customization are described in the comments in the file.

**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

# Application 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.

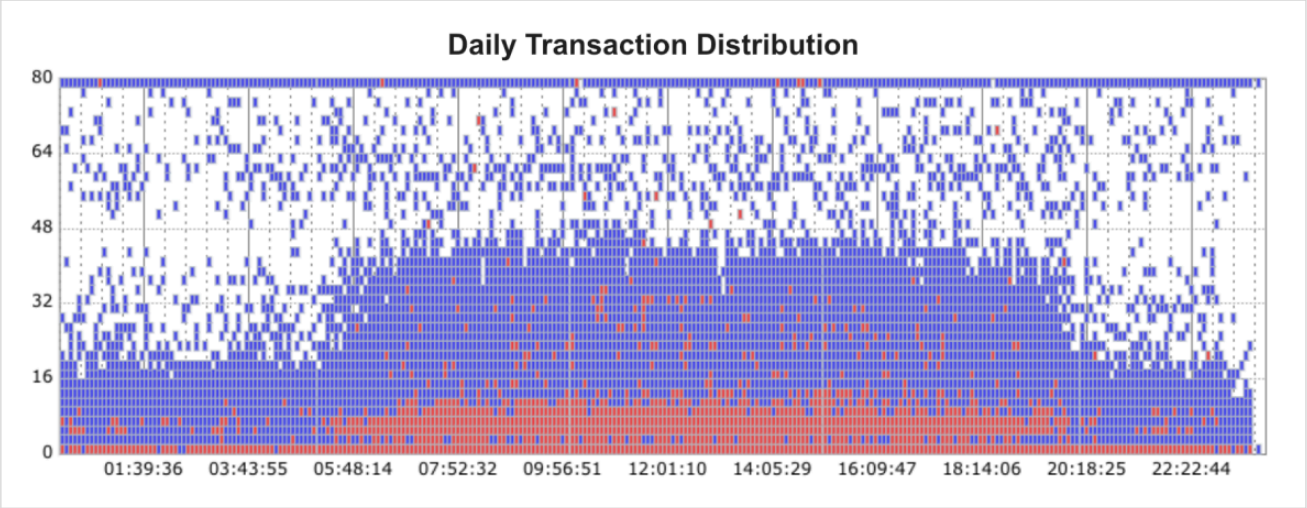
- **Intro**: You can see the key performance metrics of the application.
- **PeakTime Summary**: You can see the time of the day when the throughput (TPS) was the highest and the

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)

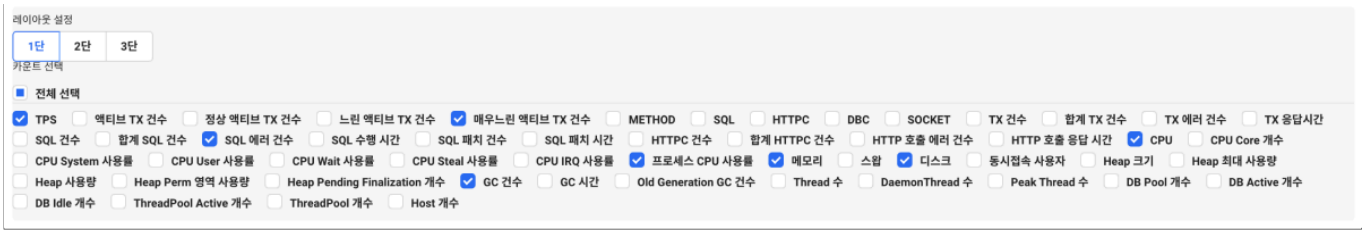
This report provides various performance metrics for applications in a time series chart. The average and maximum values for each metric are visually displayed. This allows you to monitor the system performance trends and use the data needed for performance optimization and troubleshooting.

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.

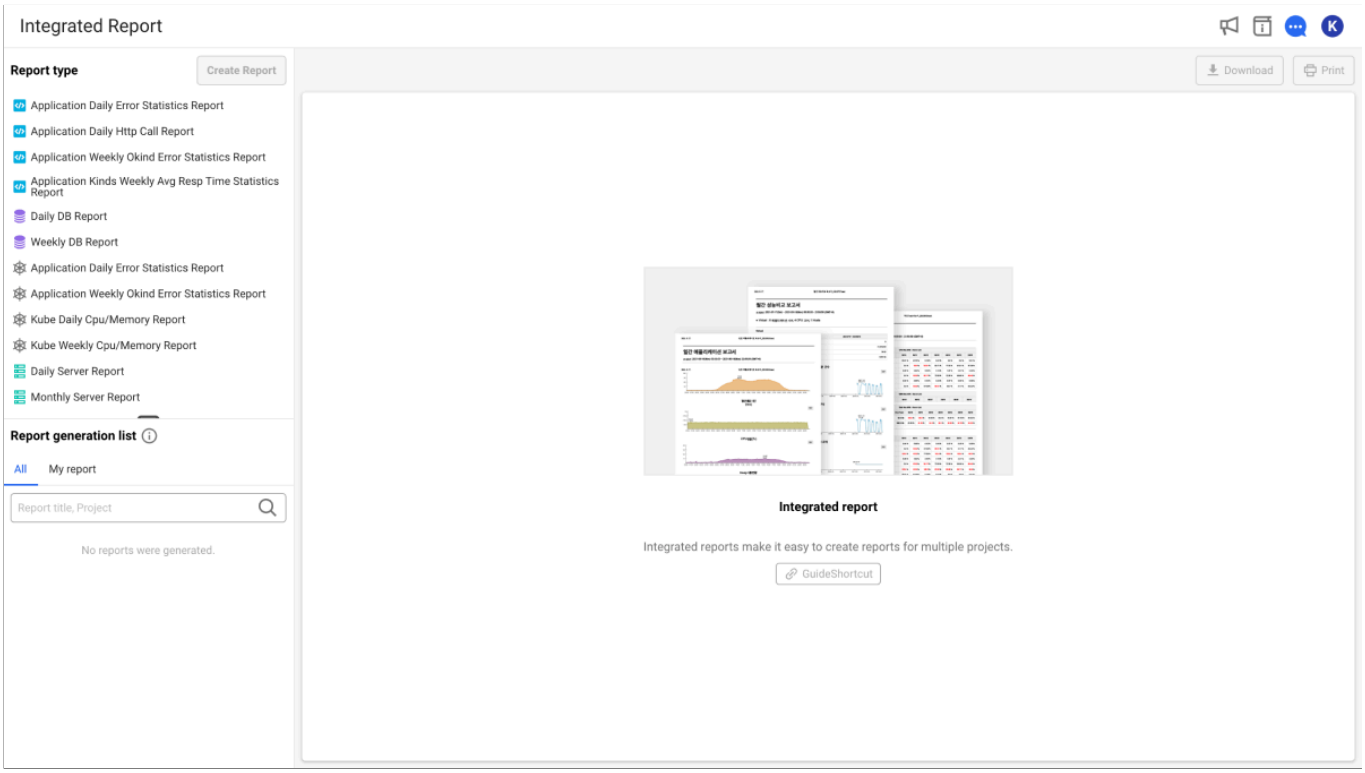
| Item | Description |
|---|---|
| **Visitor(DAU)** (**DAU**, Daily Active User) / **Total Users** | It is the number of daily active users or number of users who used the application for a day. |
| **Weekly User** | Total number of visitors who used the application for a week |
| **Monthly Active Users** (**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. |

# 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.

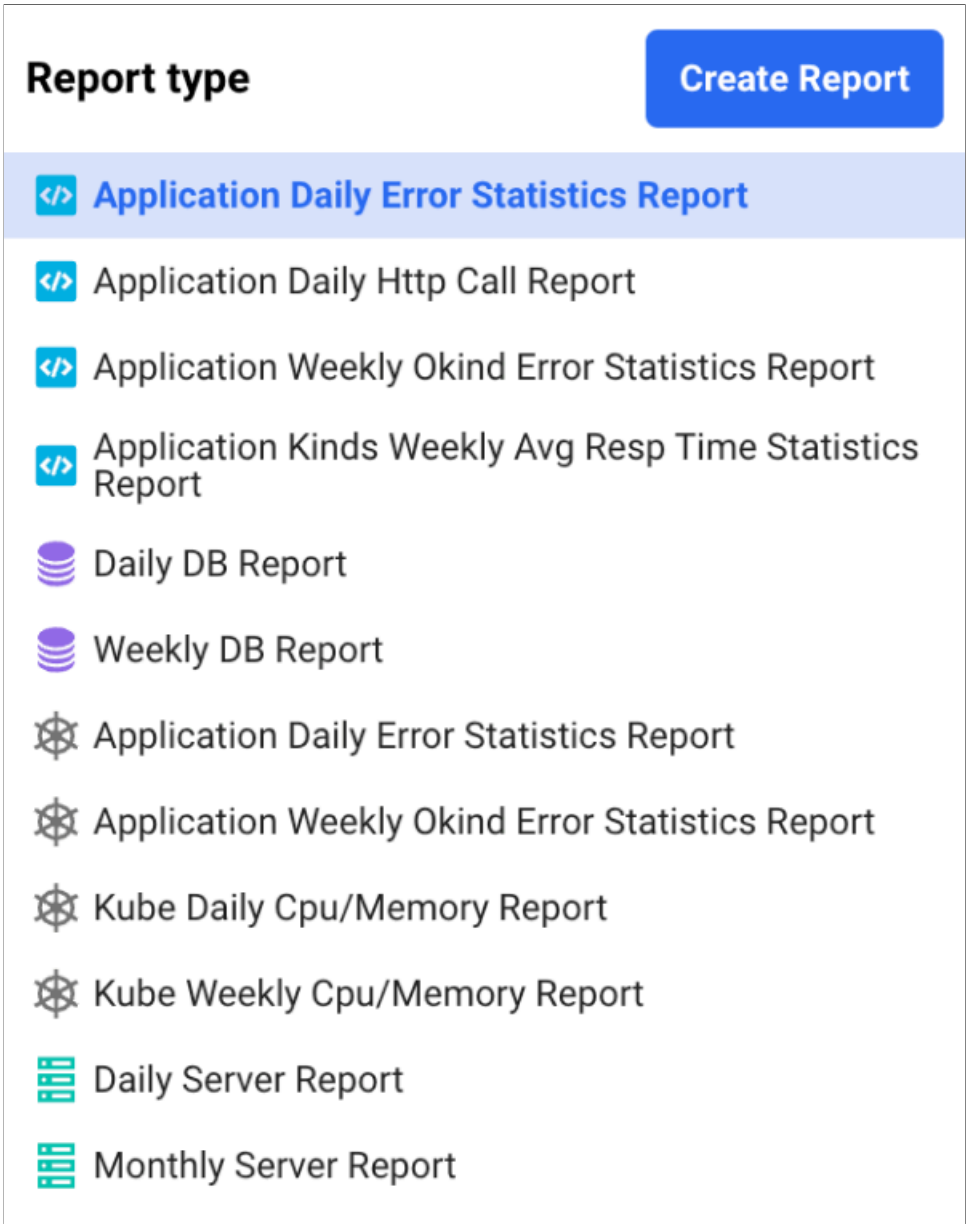
The **Integrated Report** menu can generate reports for multiple projects. It can be accessed from Home screen without selecting a project.

# 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

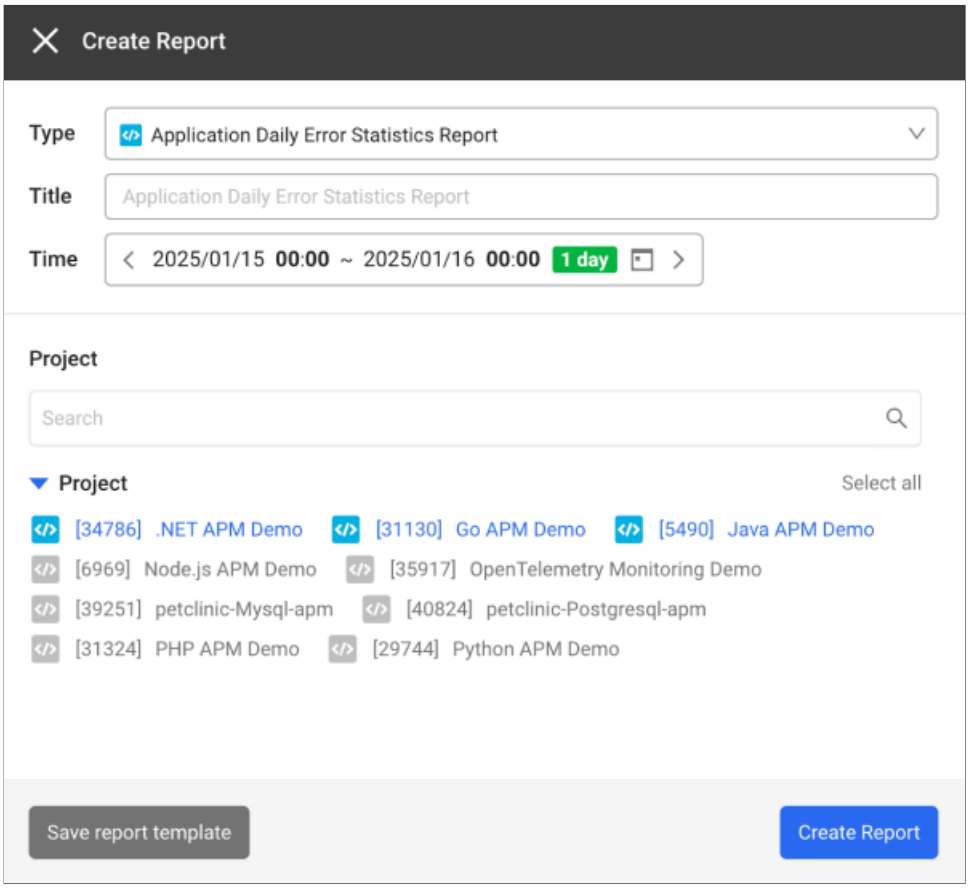
When the **Create Report** window appears, set each item.

- **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.

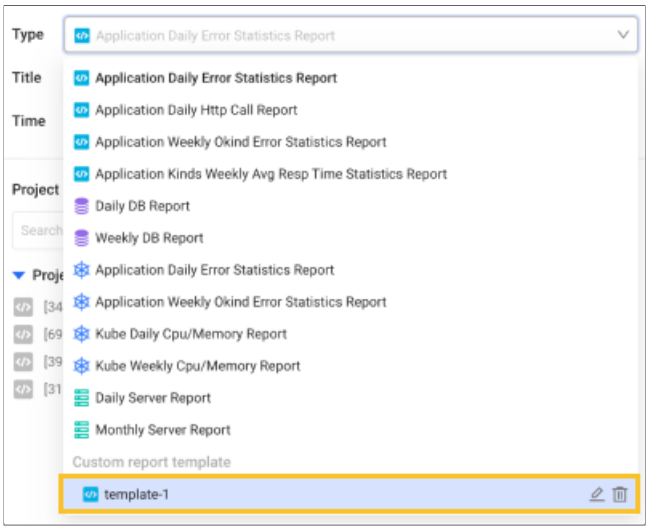
- For more information on how to use the **Time** option, see the following.
- For more information on how to load the saved report forms, see the following.

## 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.

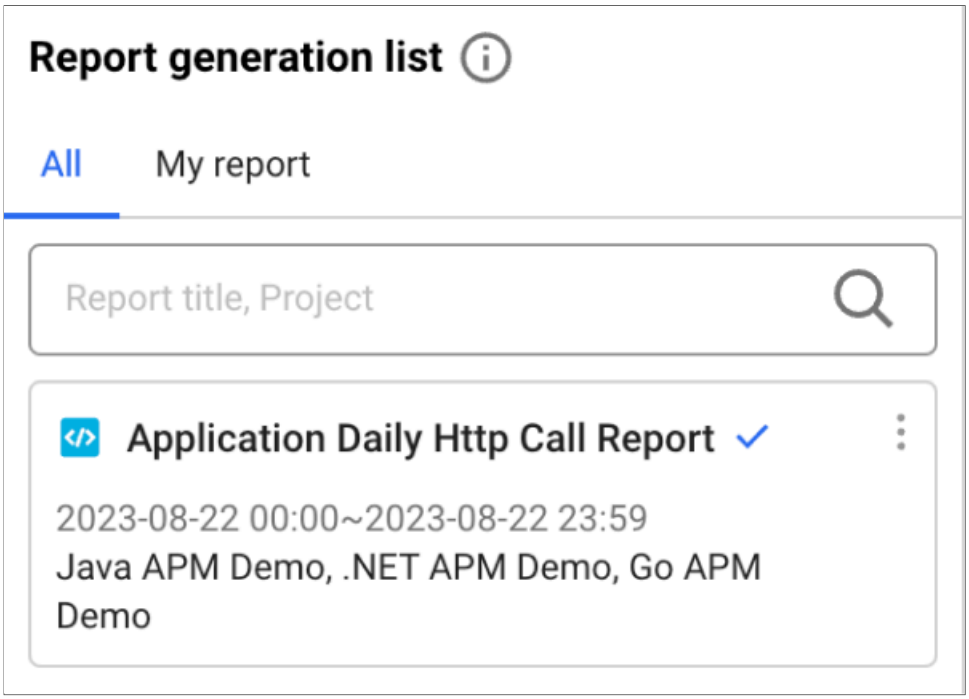
- : You can modify any saved report form.
- : You can delete any saved report form.




# 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**

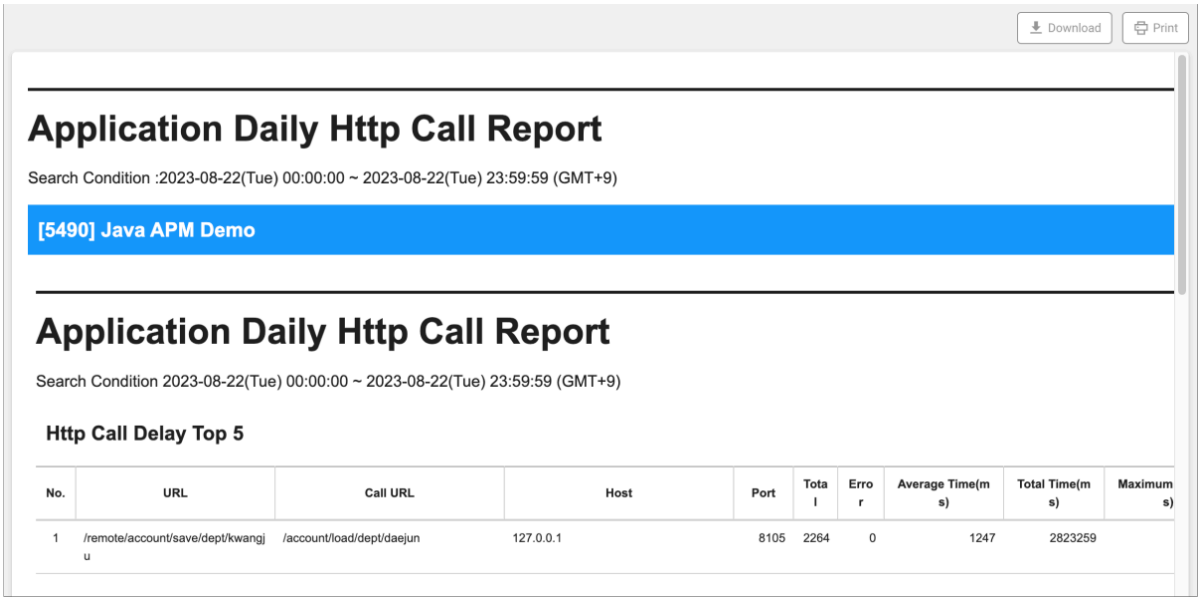
  In the **Report generation list** list, if you select a created report, you can see its details.

- **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.



# 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.

### Composite Metrics

The composite metrics feature allows you to create events using complex rules and send alert notifications.

### Hitmap Patterns

Hitmap patterns automatically detect anomalies based on changes in distributions such as response time and call density, and send alert notifications.

### Anomaly Detection

The anomaly detection feature allows you to configure alert notifications by detecting unexpected patterns.

**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.

**Transaction**

Transaction events allow you to closely monitor transactions based on specific URLs, errors, status codes, and other 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.

**Metrics**

Metrics events allow you to monitor system and application performance indicators based on real-time data.

**Composite Metrics**

The composite metrics feature allows you to create events using complex rules and send alert notifications.

### Hitmap Patterns

Hitmap patterns automatically detect anomalies based on changes in distributions such as response time and call density, and send alert notifications.

### Anomaly Detection

The anomaly detection feature allows you to configure alert notifications by detecting unexpected patterns.

### 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.

### Transaction

Transaction events allow you to closely monitor transactions based on specific URLs, errors, status codes, and other 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.

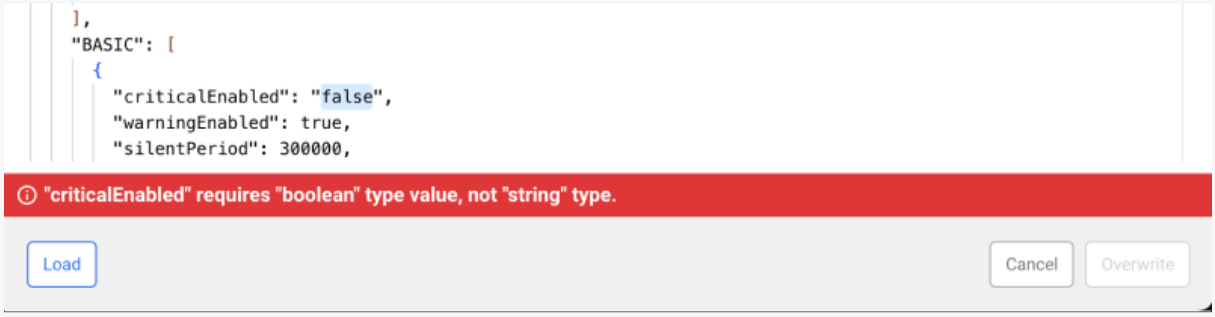
- **Batch JSON Edit**: Edit all event conditions in bulk using JSON format.
- **Batch JSON Download**: Save all currently registered event conditions as a JSON file.
- **Batch Excel Download**: Download the event list as an Excel file for external reporting or backup purposes.

## 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.

# 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<br>– Enabled: An event is triggered when conditions are met<br>– Disabled: The event does not run and conditions are not evaluated |
|  | Edit or delete event settings |
| Event Name | User-defined event name<br>– 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<br>– 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<br>– 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<br>– 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<br>– Enabled: Sends a RECOVERED notification when the status changes from in progress to normal and allows viewing active and resolved states in event history<br>– 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.

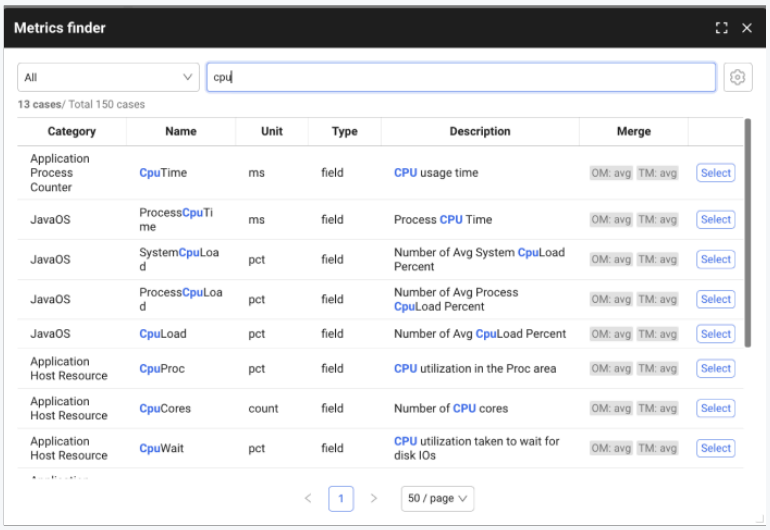
### Metric Settings

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

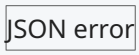
1. Click the [ ] button at the top right of the screen.
2. Select the JSON file downloaded using the **Export** function.
3. When the JSON editor opens, choose either [ Add to List ] or [ Overwrite ].

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 Metrics

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<br>- 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<br>– 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<br>– 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<br>– Enabled: Event is triggered when conditions are met<br>– 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 Patterns

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.

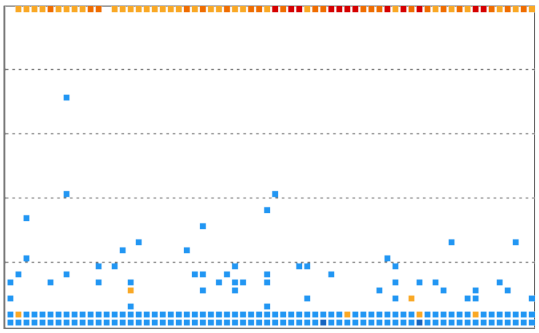
## Hitmap Pattern Types

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.

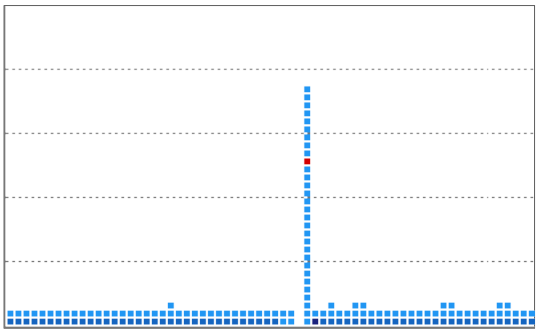
- Timeouts or delays occurring in external API, DB, or HTTPS calls
- Requests waiting for a certain duration before being processed all at once
- Repeated latency in shared I/O or network paths



## 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

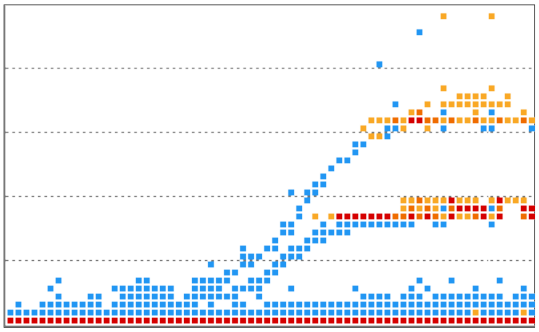


A Hitmap vertical line pattern is a signal that lock contention or bottleneck resources should be inspected.

## 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

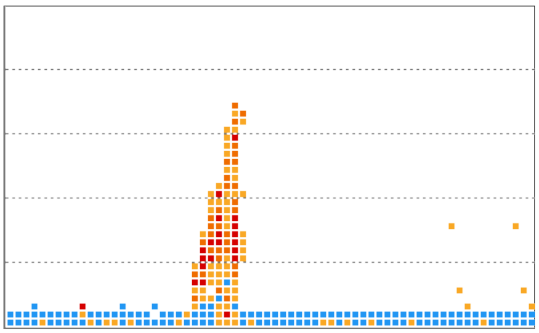


A Hitmap flying pattern strongly indicates periodic bottlenecks or resource shortages.

## 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

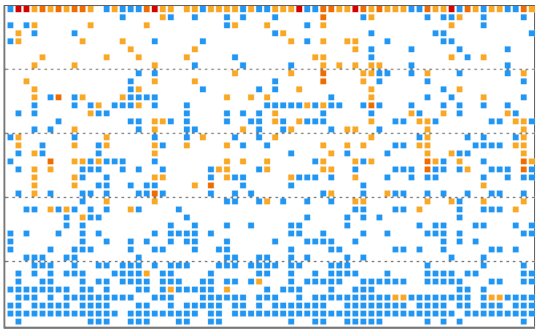


An overload pattern indicates temporary service quality degradation or load balancing failure.

## 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

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<br>- Enabled: Event is triggered when conditions are met<br>- 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<br>- Up to 255 bytes |
| Category | Log classification name (log folder name) |
| Search key | The field to search for in log data<br>- 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<br>- 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<br>- 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<br>- If no condition is set, alerts are evaluated for all collected data<br>- Select input: Build conditions by selecting tags, operators, and values<br>- 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<br>- Enabled: Event occurs when conditions are met<br>- 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<br>- Enabled: Sends a RECOVERED notification when transitioning from in-progress → normal status, and the in-progress/resolved status can be distinguished in event history<br>- 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<br>- 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<br>- 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<br>- Enabled: Event occurs when conditions are met<br>- 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

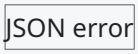
1. Click the [ ] button at the top right of the screen.
2. Select the JSON file downloaded using the **Export** function.
3. When the JSON editor opens, choose either [ Add to List ] or [ Overwrite ].

> 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

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<br>- Enabled: The event is triggered when conditions are met<br>- Disabled: The event does not operate and conditions are not evaluated |
|  | Edit or delete event settings |
| Event Name | User-defined event name<br>- 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<br>- Aggregation period: fixed at 1 minute (based on each minute from :00 to :59 seconds; the period cannot be changed)<br>- 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<br>- 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<br>- 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<br>- Example: `/api/*`, `*/users/*` |
| Error Class | Error class name pattern<br>- Example: `*Exception`, `*Error` |
| Error Message | Error message content pattern |
| Status Code | HTTP status code pattern<br>- 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/`<br>- `/users/*`: All URLs that contain `users`<br>- `/api/v1/users`: Only the URL that matches exactly |
| Error Class Patterns | - `Exception`: All errors that end with `Exception`<br>- `java.lang.*`: All classes that start with `java.lang`<br>- `Timeout*`: All errors that contain `Timeout` |
| Status Code Patterns | - `4**`: All 4xx status codes<br>- `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

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<br>- Applies only to Critical alerts |
| Smart Message (AI) | AI-powered notifications available when adding the Telegram plugin<br>- 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**.

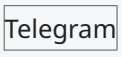
## How to Add 3rd-party Plugin Notifications

Slack

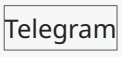
In the **Add 3rd-party Plugin** window, click the [ **Add To Slack** ] button on the **Slack** tab to register **WhaTapNotiBot** to your Slack team and channel.

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.

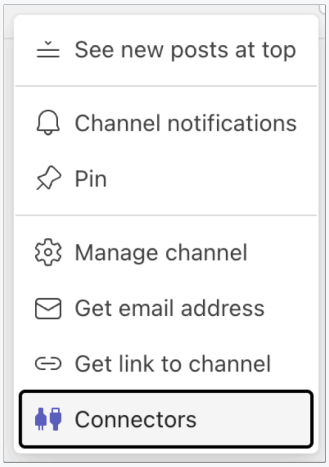
**Available commands**

- `/help` : View the list of commands
- `/license` : Enter the access key of the project to receive alerts

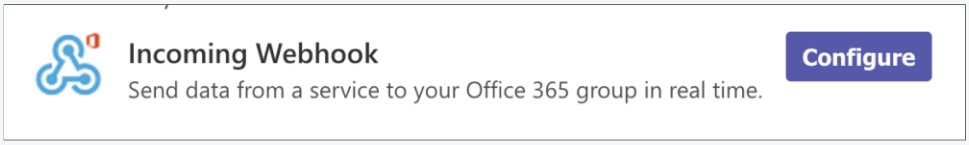
- `/remove` : Enter the project code for which you want to stop receiving alerts
- `/list` : View the list of project codes currently receiving alerts

Teams

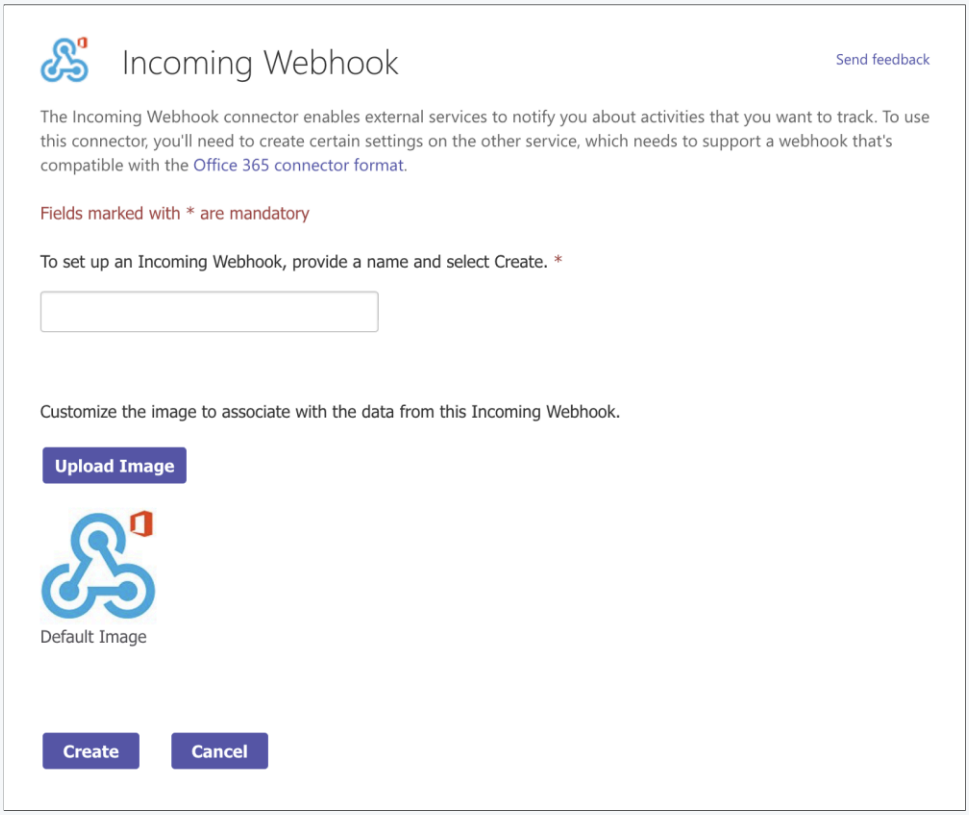
1. Go to the **connector** of the channel where you want to receive alerts.



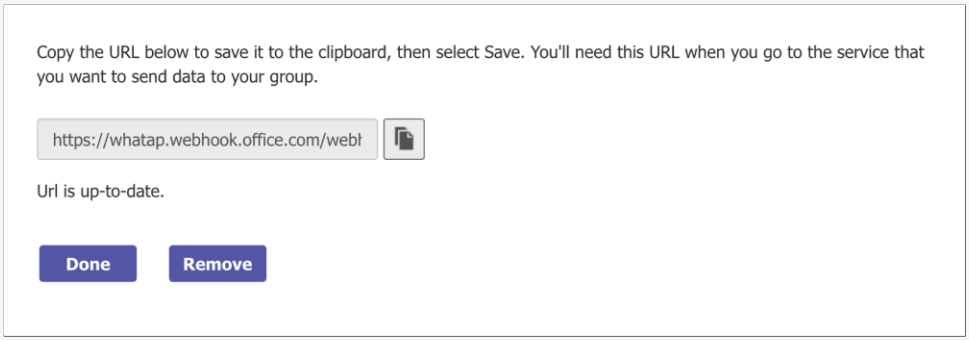

2. Add a **Webhook** and click the [ Configure ] button.




3. Enter a webhook name and click the [ Create ] button.




4. Copy the generated **Webhook URL**.

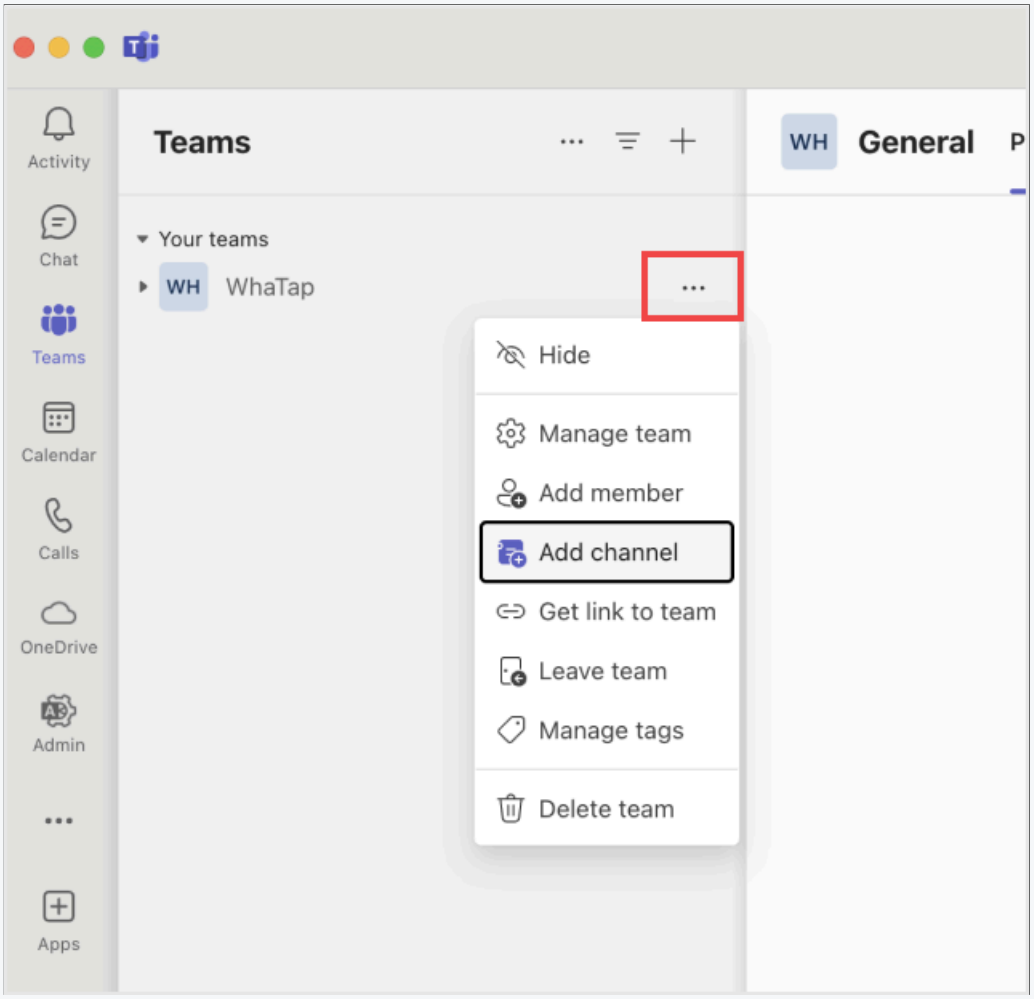



5. Return to the **Add a third-party plug-in** window in the WhaTap Monitoring service, and enter values for **Channel Name** and **Webhook URL** at the bottom of the **Teams** tab.
6. Click the [ **Registration** ] button.

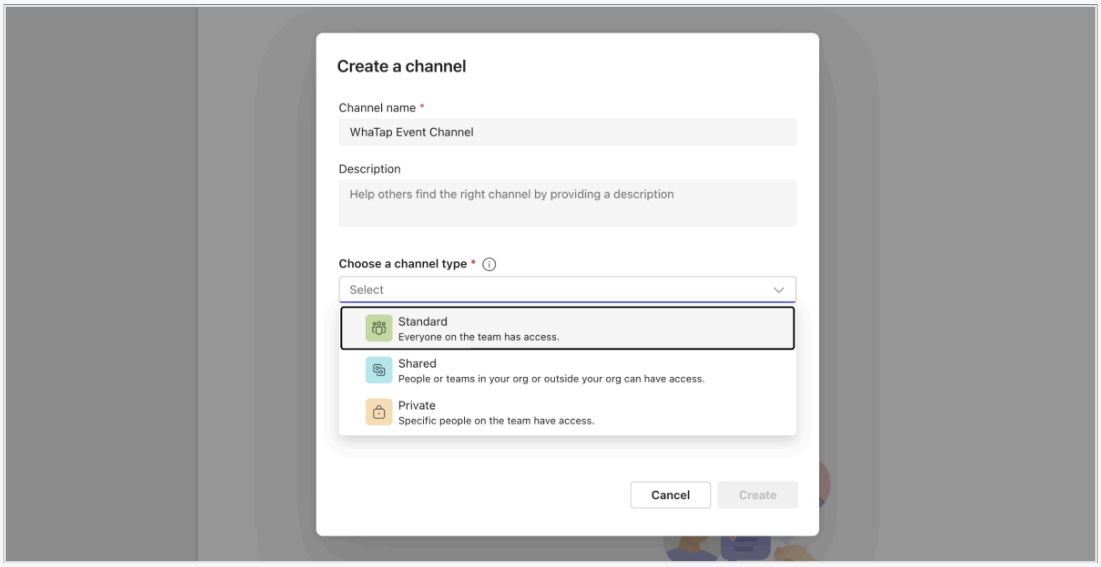
Teams Workflows

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**.

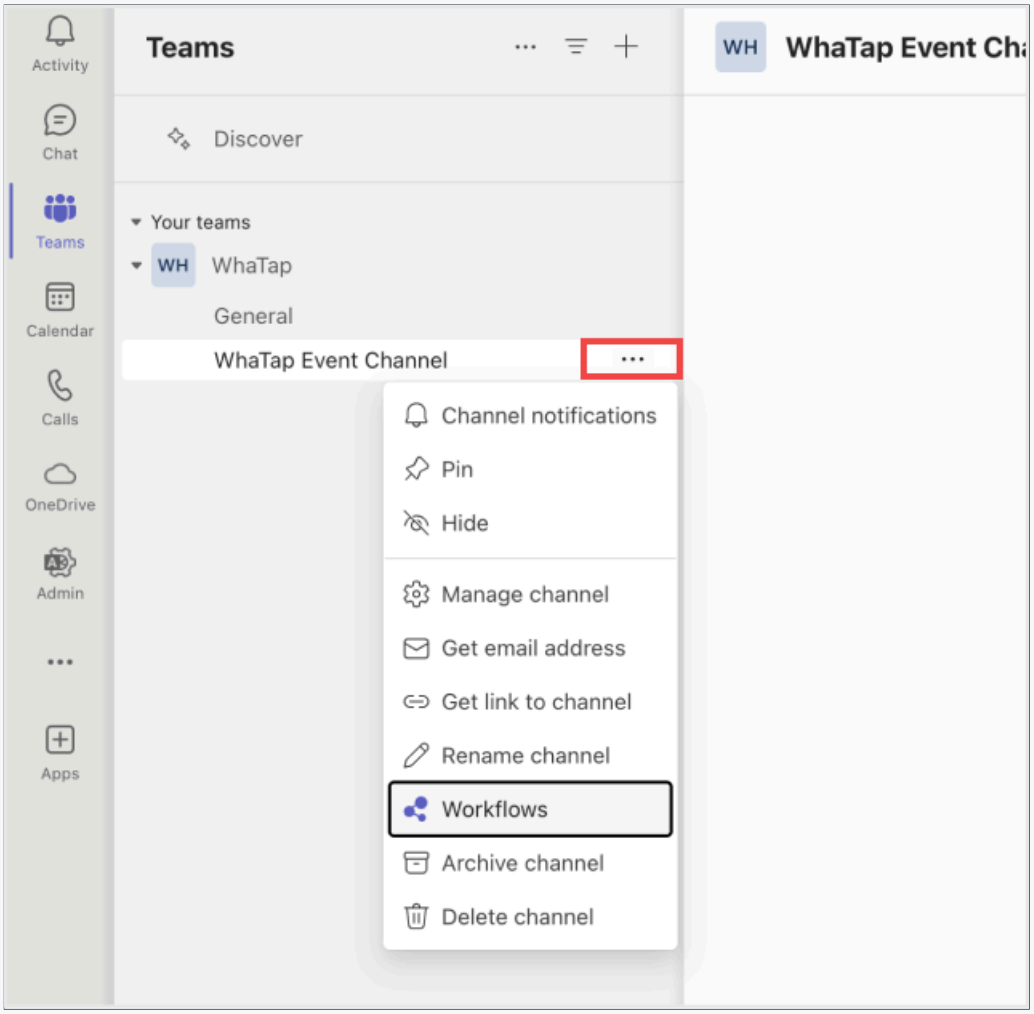



2. In the **Create a channel** window, enter the **Channel name** and **Description**. For **Channel type**, select **Standard**, then click the [ Create ] button.

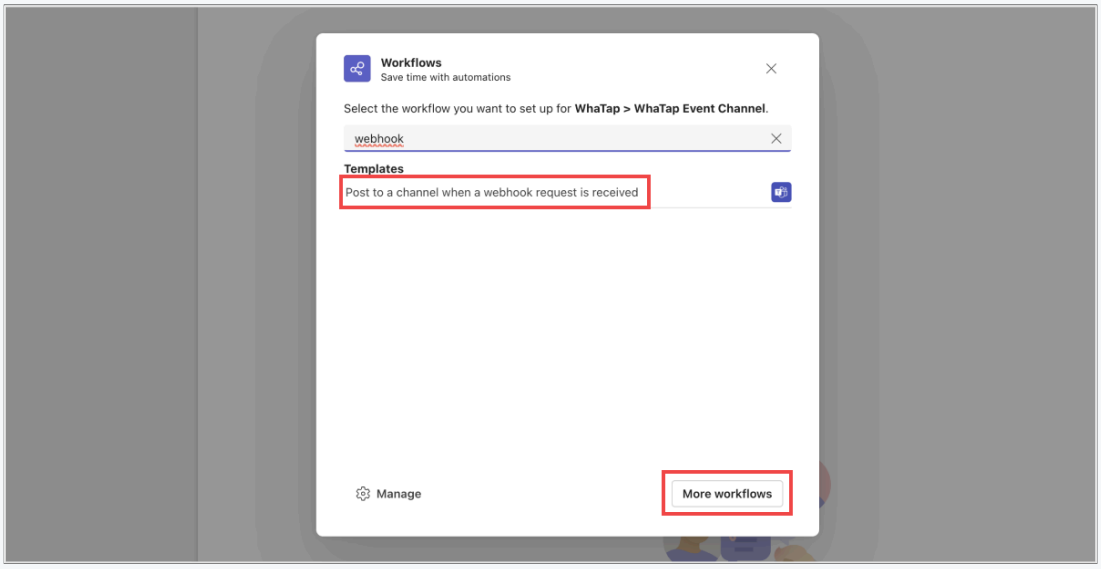



If you select **Shared** as the channel type, workflows cannot be used. If you select **Private**, workflows can be added but notifications cannot be received.

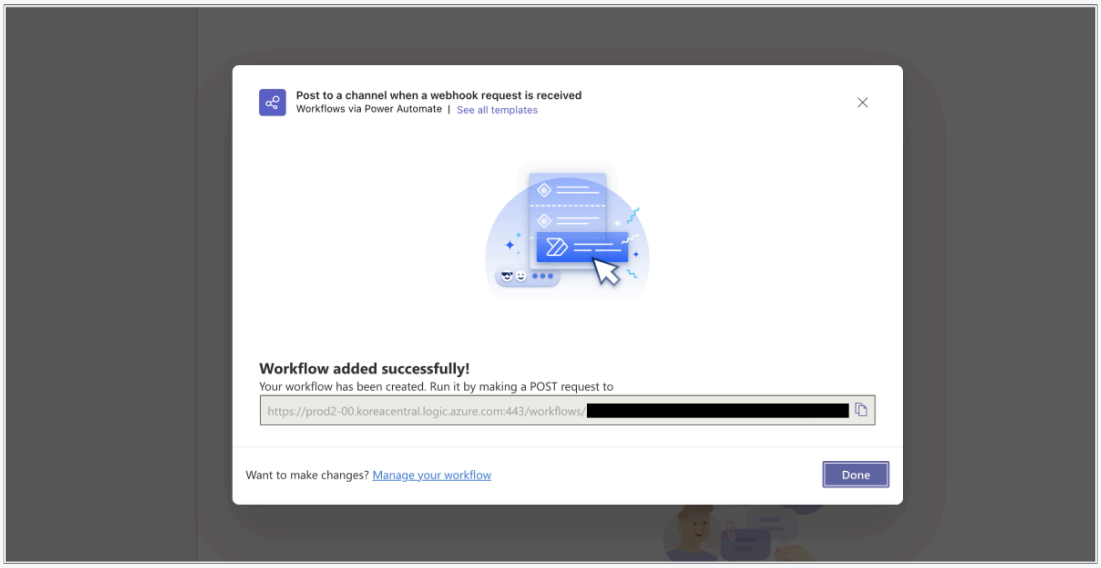
3. Click **More actions (...)** of the created channel, then select **Workflows**.




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.

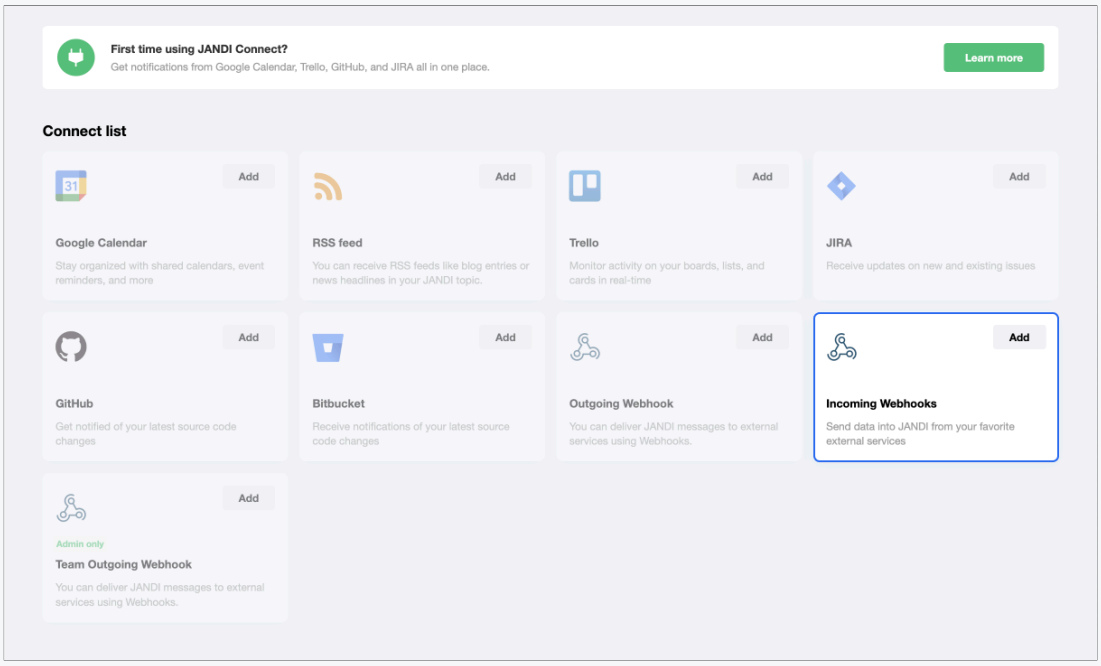



8. Return to the **Add a third-party plug-in** window in the WhaTap Monitoring service, and enter values for **Channel Name** and **Webhook URL** at the bottom of the **Teams Workflows** tab.

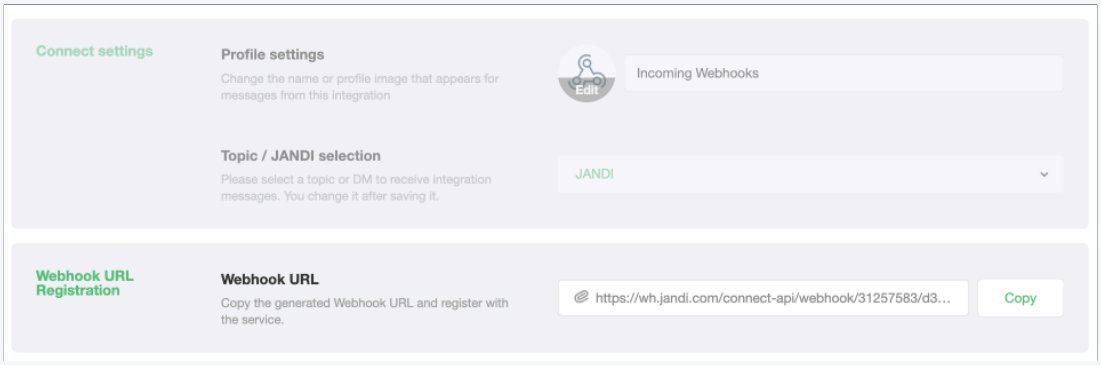
9. Click the [ **Registration** ] button.

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.



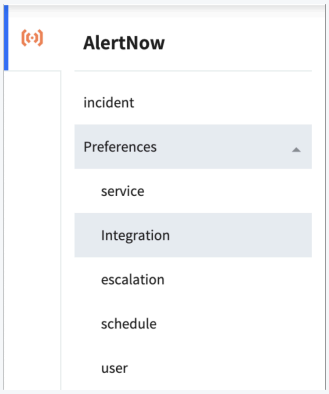

3. Click the [ Add Webhook Receiver ] button.
4. Copy the Webhook URL from the **Register Webhook URL** section.



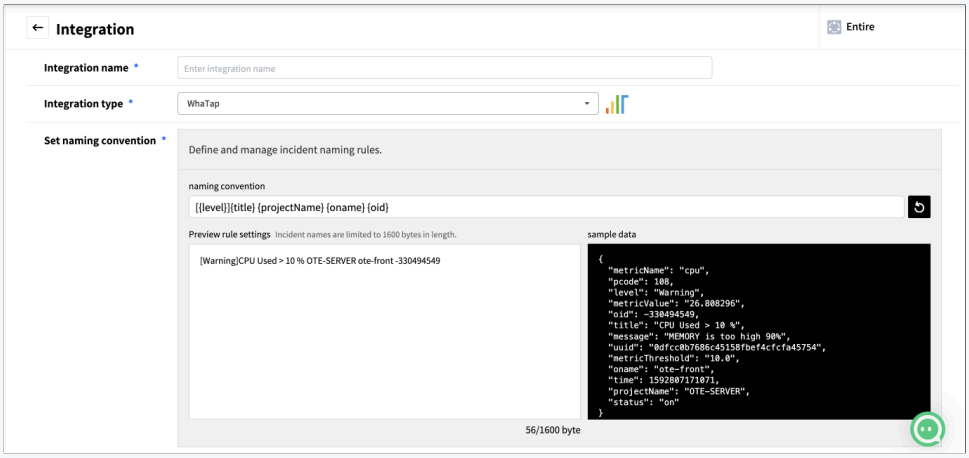

5. Return to the **Add a third-party plug-in** window in the WhaTap Monitoring service, and enter values for **Channel Name** and **Webhook URL** at the bottom of the **Jandi** tab.
6. Click the [ **Registration** ] button.

AlertNow

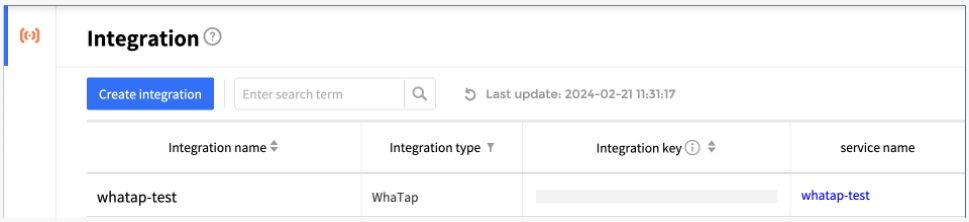
1. Access the **AlertNow** service, then go to **Settings** > **Integration**.




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` .

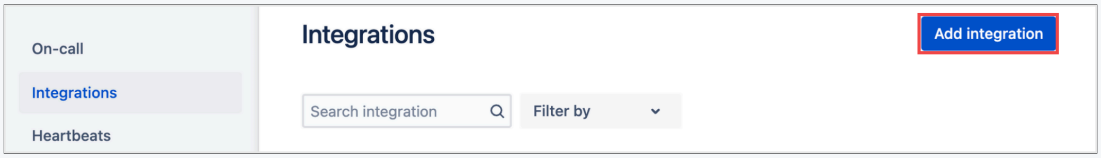
```
{
   "pcode": "${pcode}",
   "projectName": "${projectName}",
   "time": "${time}",
   "oid": "${oid}",
   "oname": "${oname}",
   "title": "${title}",
   "message": "${message}",
   "level": "${level}",
   "status": "${status}",
```

```
  "metricName": "${metricName}",
  "metricValue": "${metricValue}",
  "metricThreshold": "${metricThreshold}",
  "uuid": "${uuid}",
  "okind": "${okind}",
  "onode": "${onode}"
}
```

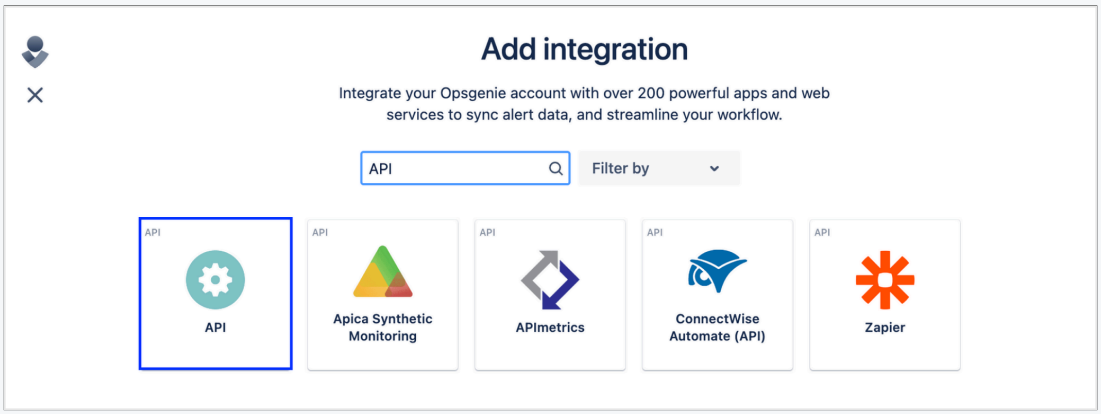
2. Enter values for the **Webhook Name** and **Webhook URL** fields.
3. To add a header value, click **+ Add** in **Add user header value**, then enter the key and value.
4. Click the [ **Registration** ] button.

Opsgenie

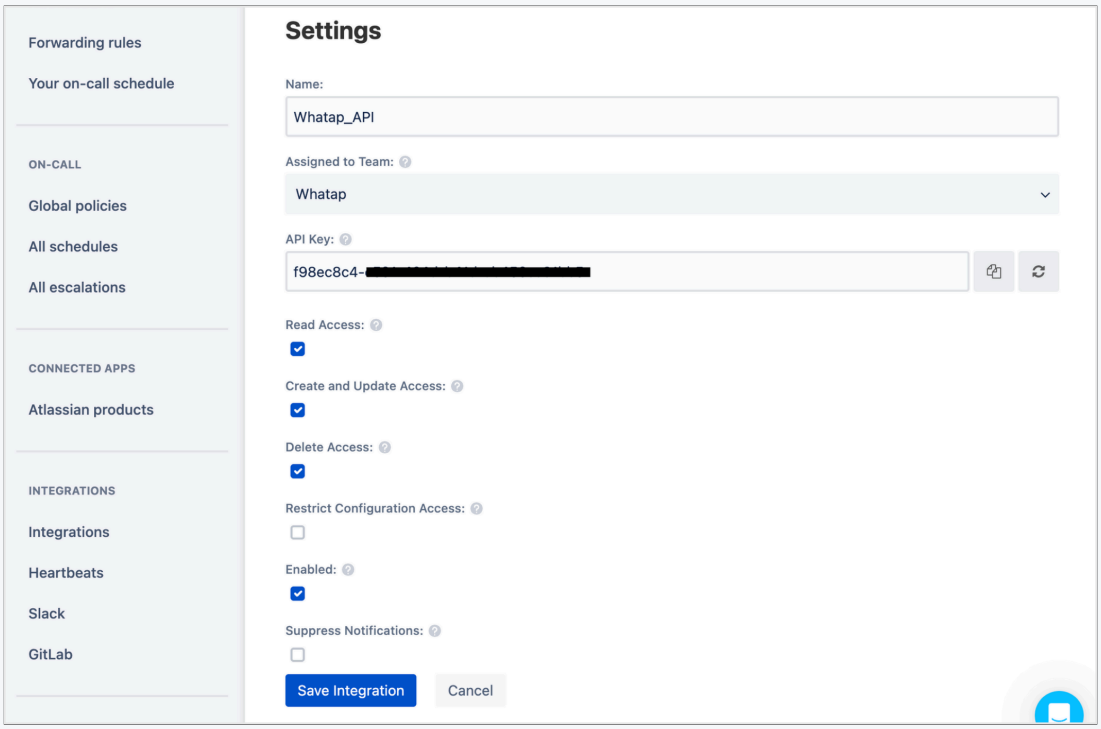
1. In the **Teams** menu, select the team that will receive notifications.
2. Go to the **Integration** menu and click the [ Add Integration ] button.




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 |

# 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<br>- Displayed as blank if the event is not associated with a specific agent |
| Agent Group | The agent type (OkindName) where the event occurred<br>- 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<br>- 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<br>- 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.

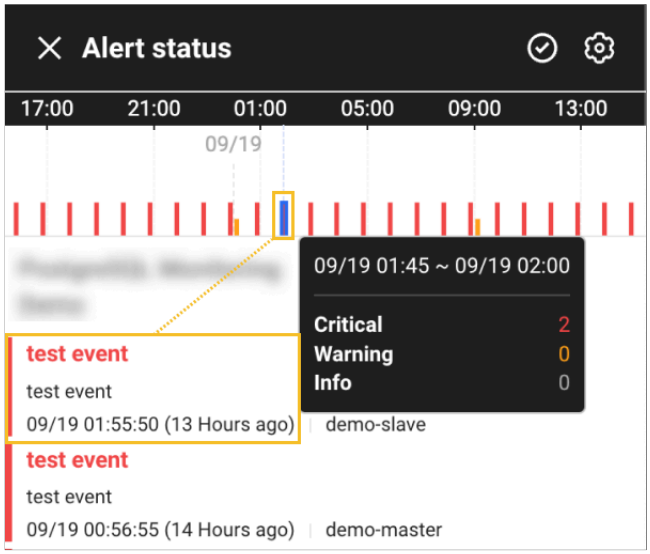
## Viewing Real-time Notifications

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

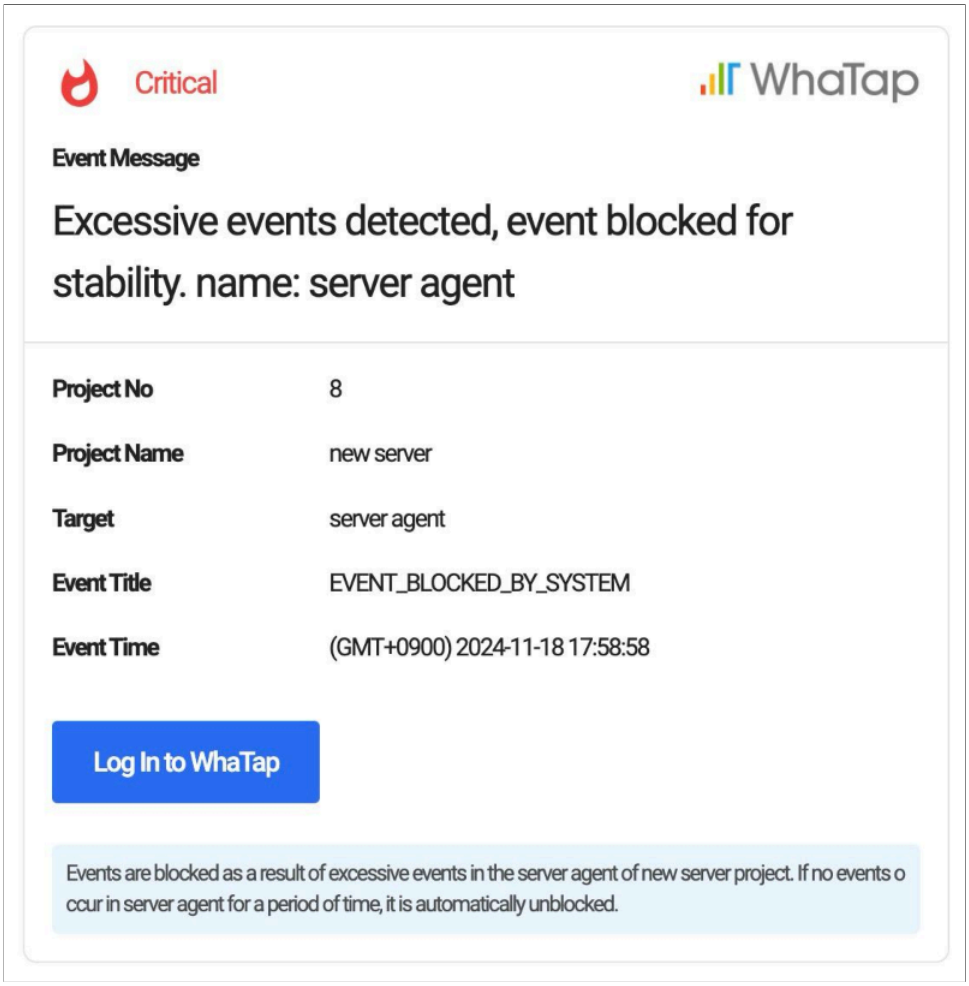
1. Click the icon at the top of the screen after selecting a project.
2. When the **Alert status** window appears, click the icon in the upper-right corner.
3. Configure each option in the real-time notification settings window.
   - **Popup notifications**: Displays notifications as pop-ups even when 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 Management

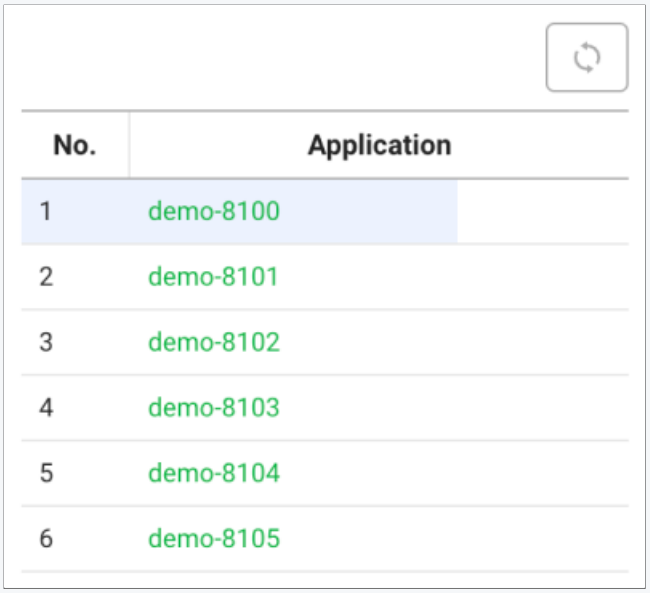
Home > Select Project > **Instance Performance Management**

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**

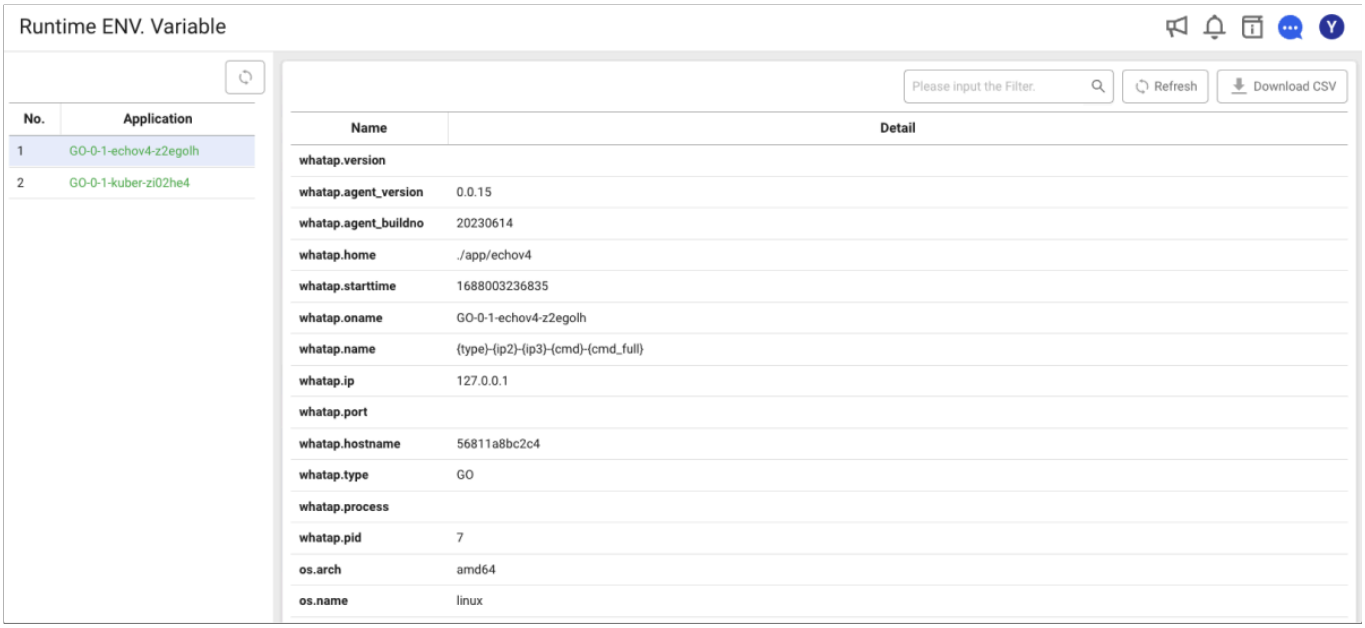
You can check the list of agents assigned to the project in the application list on the left of the screen. If you select an individual agent, you can view information for environment variables, agent settings, and performances set on the right of the screen.




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**

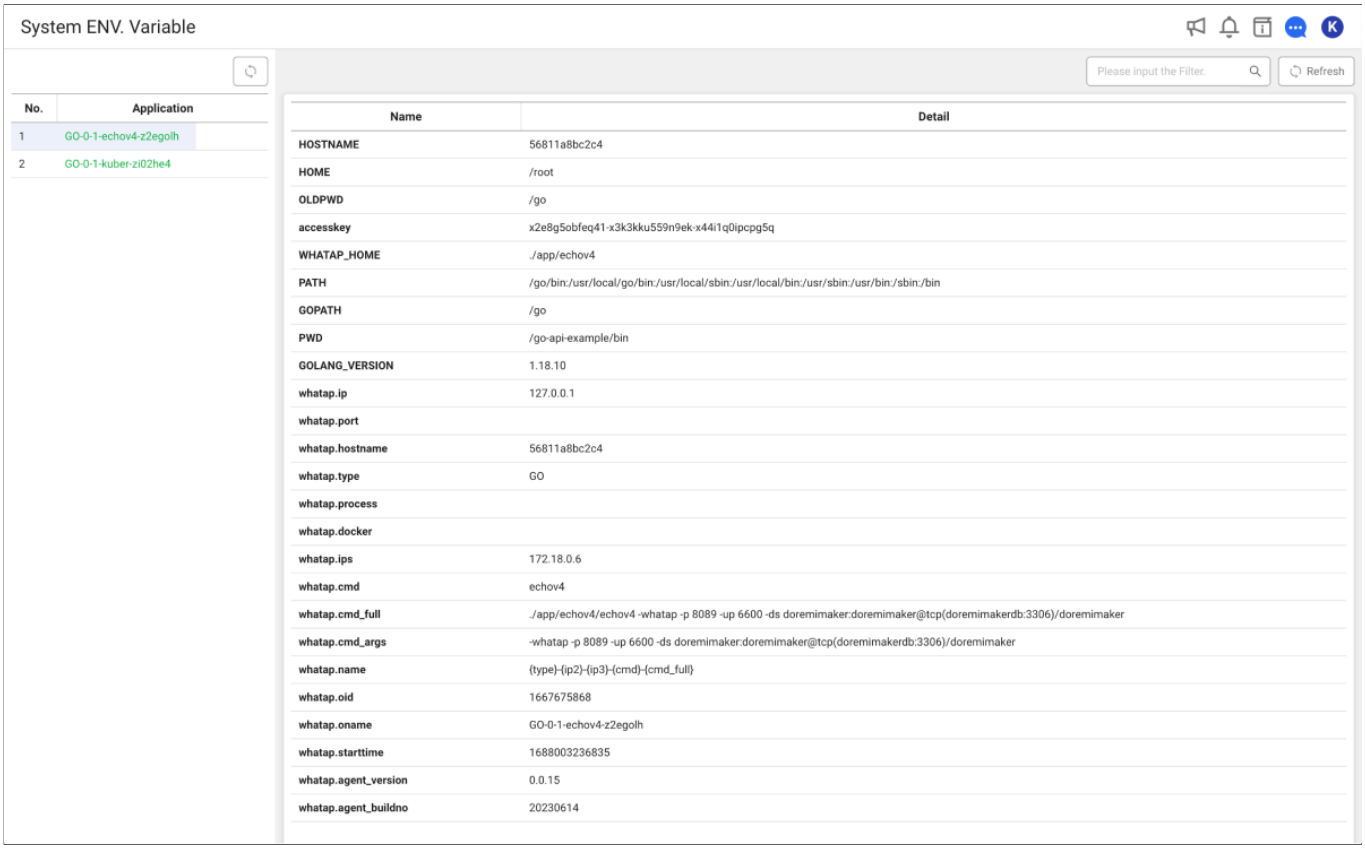



You can view environment variables for agent execution. Important information among environment variables or application status can be stored on the WaTap server so that they can be retrieved even if the agent is down. You can see the version, installation path, name, and IP address of the agent. You 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.

# System ENV. Variable

Home > Select Project > **Instance Performance Management** > **System ENV. Variable**

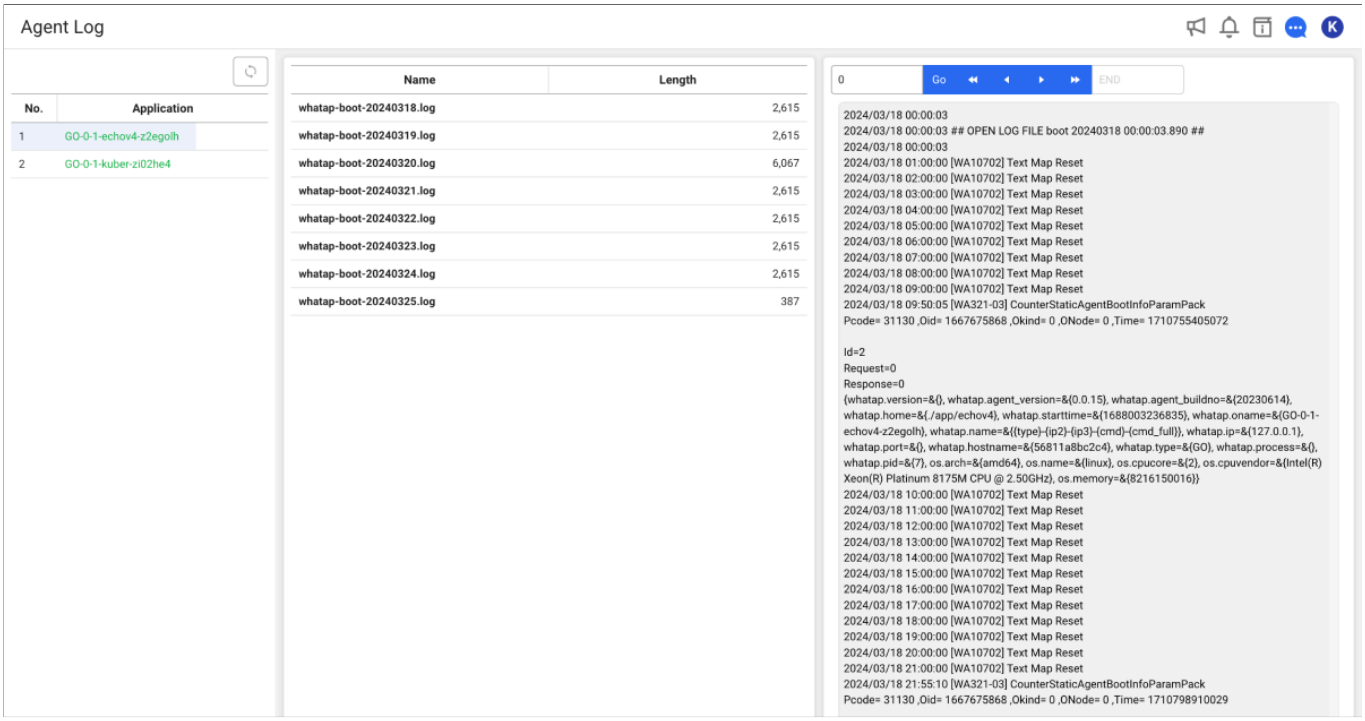



You can view environment variables for application execution. The **Boot environment** menu collects and provides environment variables at the time of query. It uses `System.getProperty()` to retrieve the value of the key passed as a parameter. The following are key points to check:

- `(Application).version` : You can check the application version.
- `file.encoding` : You can check the file encoding format such as `utf-8` .
- OS environment variables: For example, they are provided in the suffix `(env)` format, such as `(env) LANG` environment variable.

# Agent log

Home > Select Project > **Instance Performance Management** > **Agent Log**

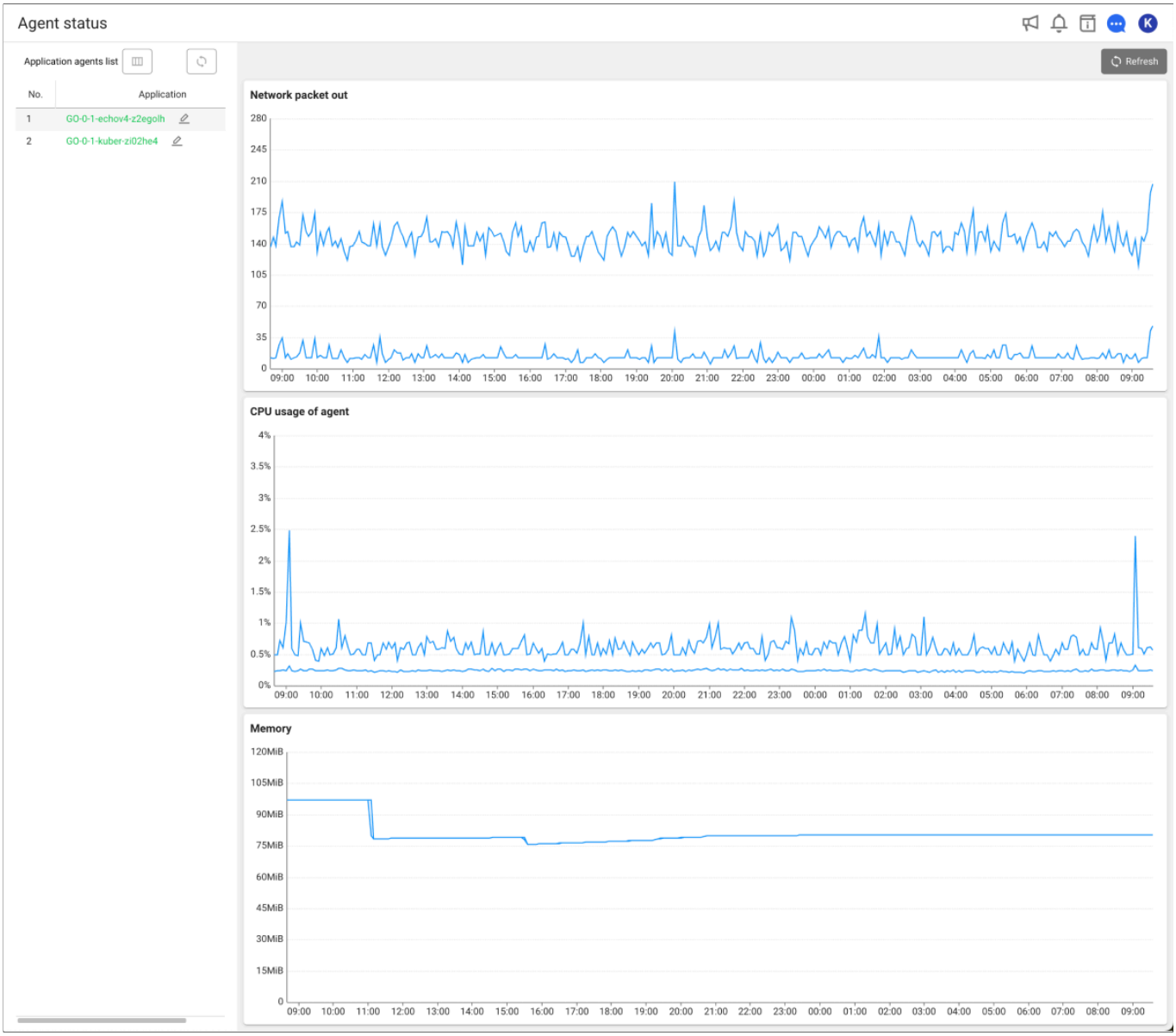



You can view the agent logs stored on the monitored server. The name of the log file is in the format of whatap-`YYYYMMDD`.log. You can select each log to access information about the errors and events captured in the logs.

> ⓘ For the agent settings for the log, see the following.

# Agent status

Home > Select Project > **Instance Performance Management** > **Agent Status**

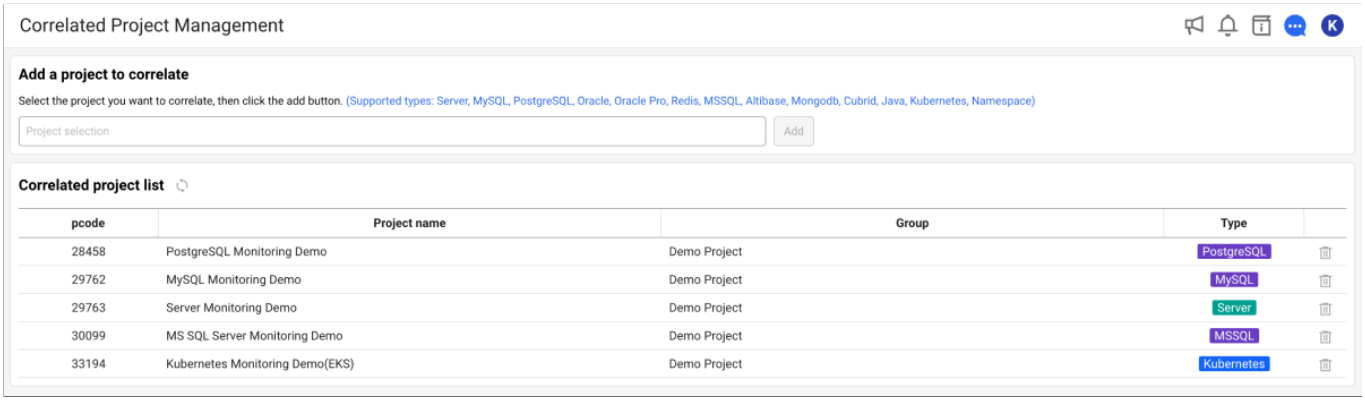



You can see the resource usages used by the agent over the past day.

# Correlated Project Management

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.

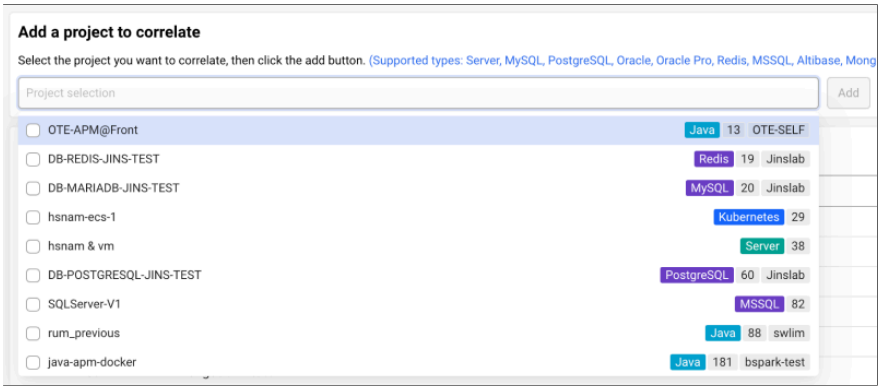



The **Correlated Project Management** feature allows you to go beyond product-/equipment-centric monitoring and view monitoring data across multiple projects. This is useful for analyzing how different system components such as applications and databases, interact together. In particular, when performance 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.

> **Notes before you begin**
>
> - Supportable databases: PostgreSQL, Oracle, MySQL, SQL Server, CUBRID, Altibase, Redis, MongoDB

## 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.

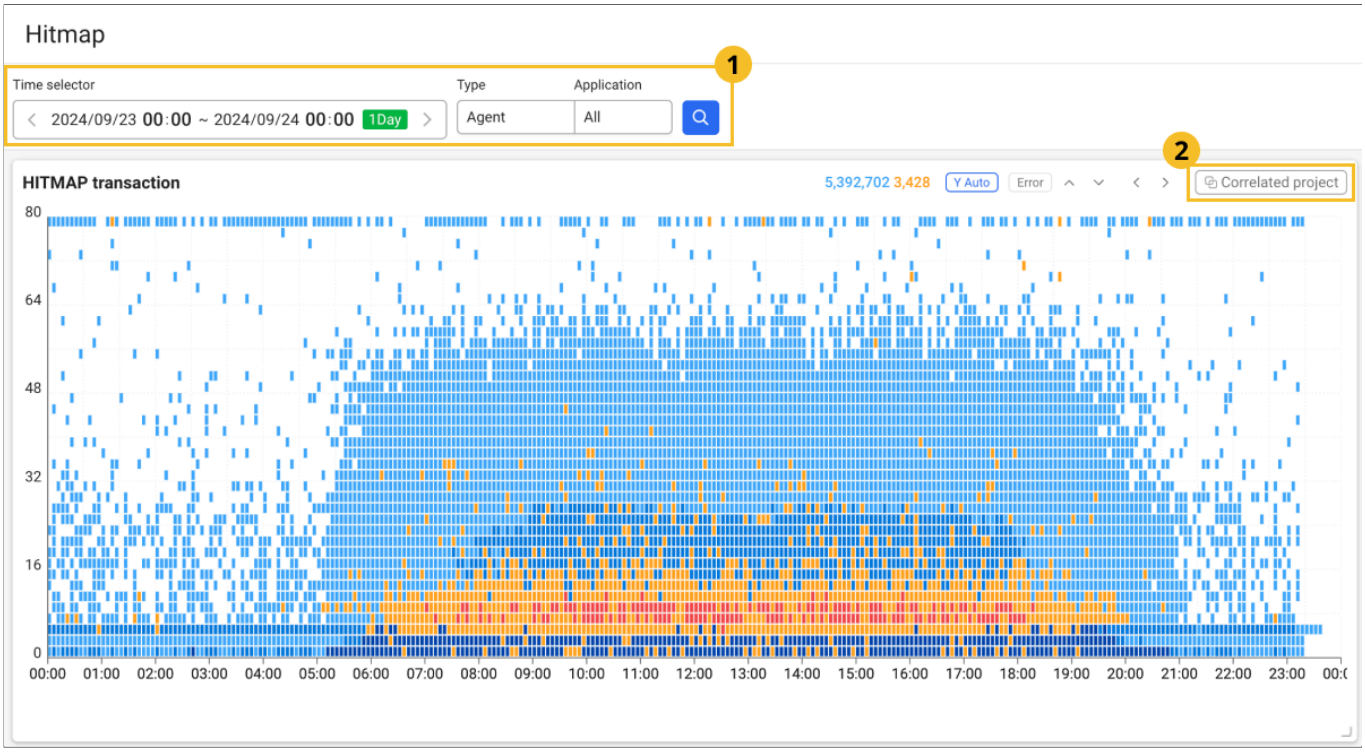
The selected project is added in **Correlated project list**.

> ⓘ 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.

# Checking the metrics of the linked project

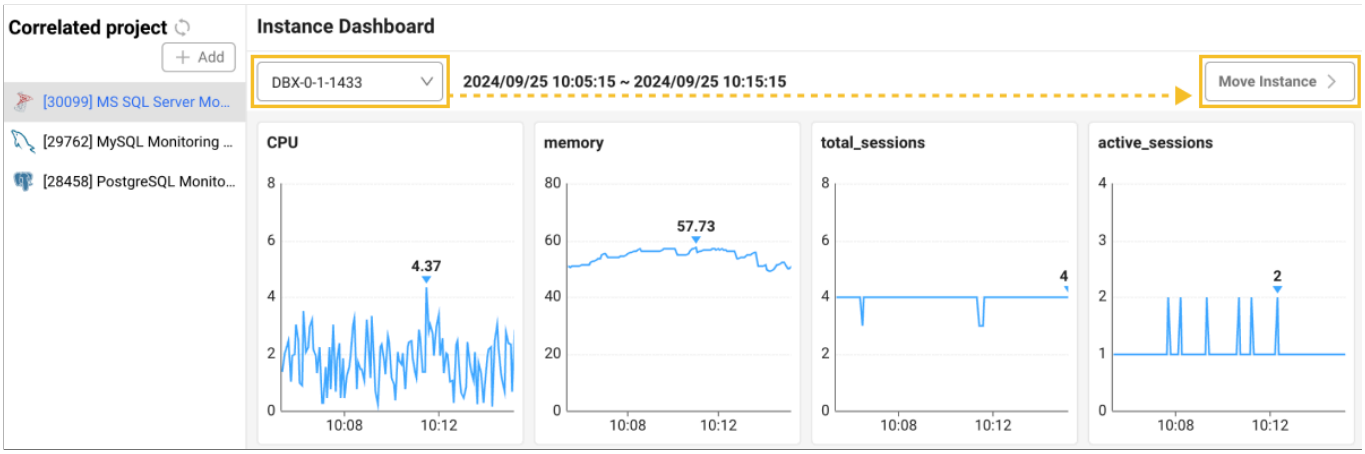
Metrics related to the added project can be checked in the **Analysis** > **Hitmap** menu. You can also see performance metrics for the linked project while viewing transactions.

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.



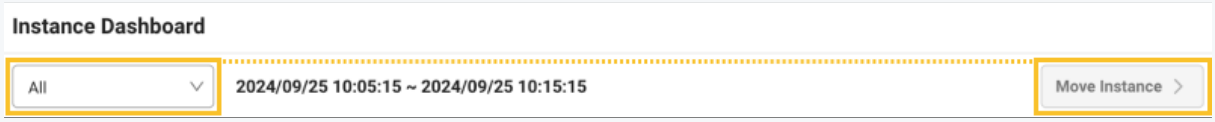

1. Set the time and **Application** for viewing the transactions, and then select .
2. In the **HITMAP transaction** section, select **Correlated project** on the upper right.
3. A new window appears where you can check the metrics for the correlated project.

Select the correlated project from the left list and then check the metrics in the **Instance dashboard** section. **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.

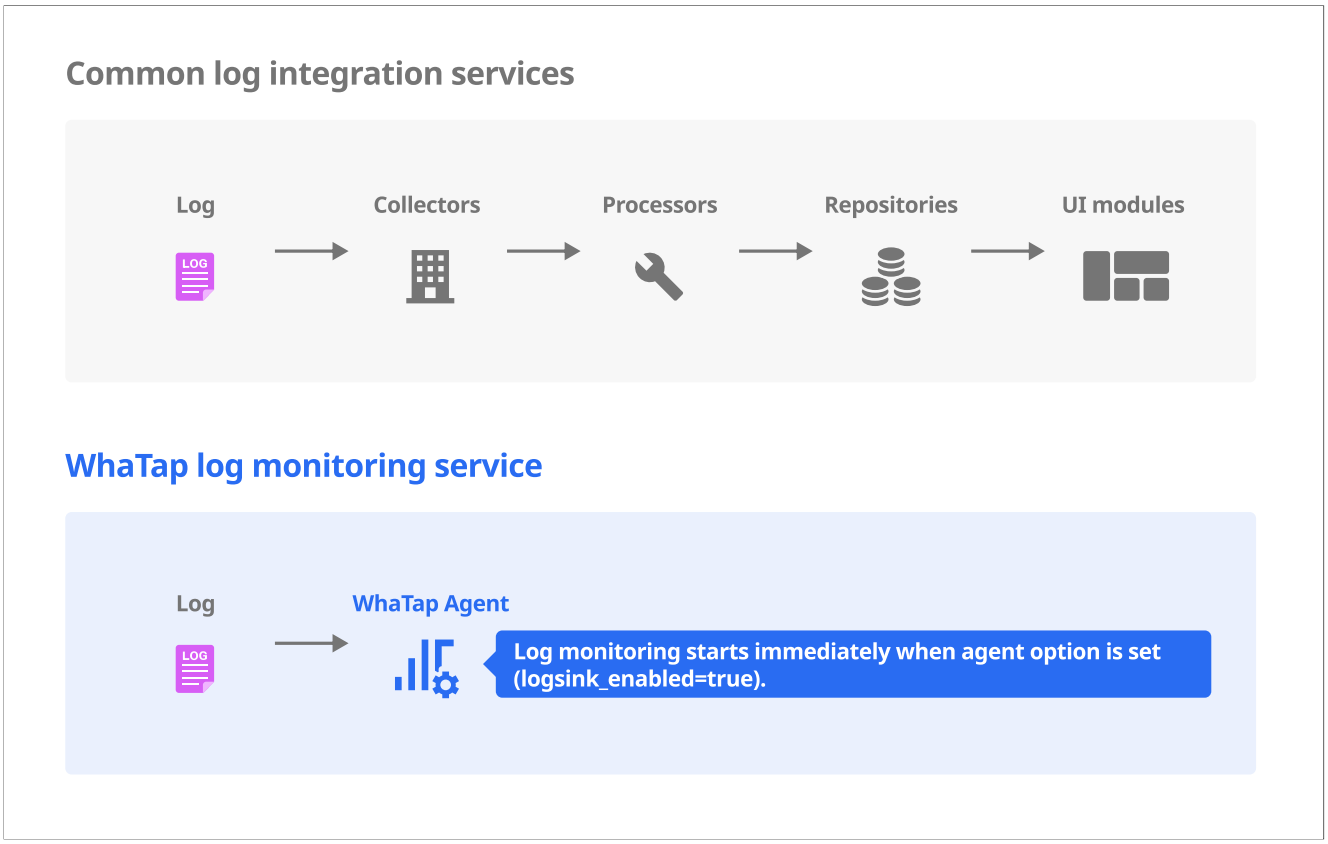
## Log Search

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`).

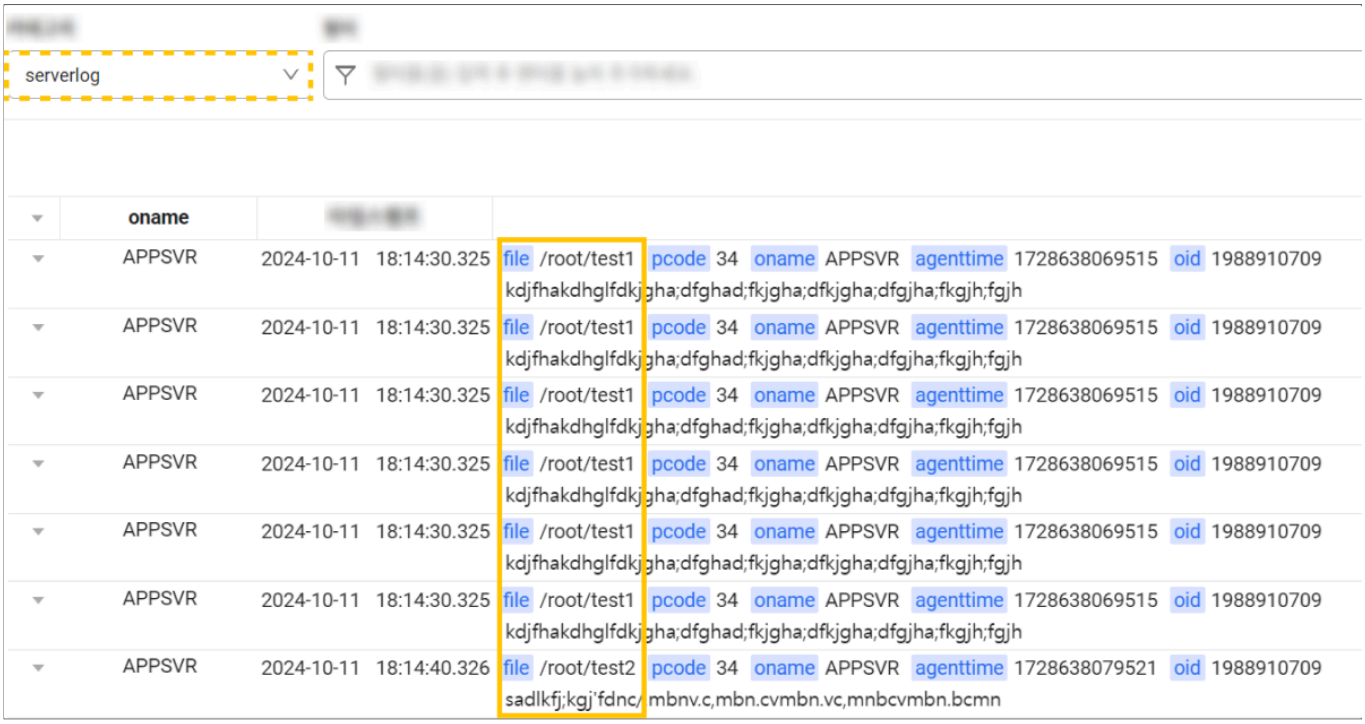
```
[[inputs.logsink]]
 category = "serverlog"
 stats_enabled = true
 stats_category = "logsink_stats"
 excludeNames = [ ".gz",".zip" ]

[[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"
```

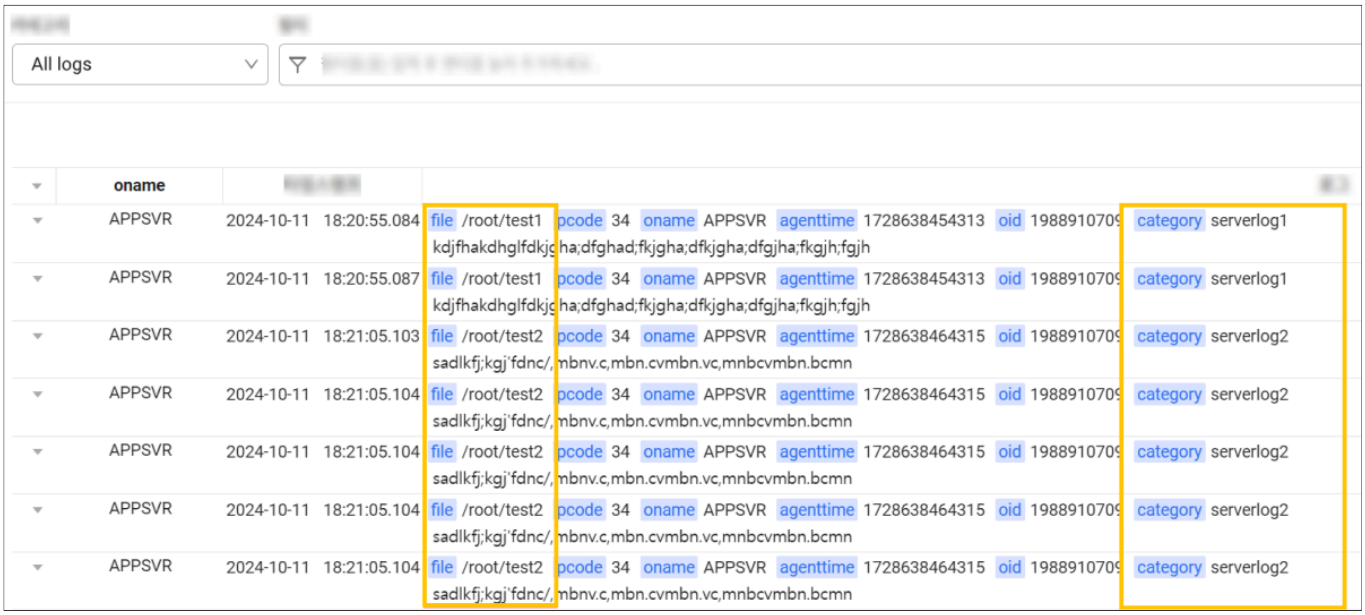
logsink1.conf

```
#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.

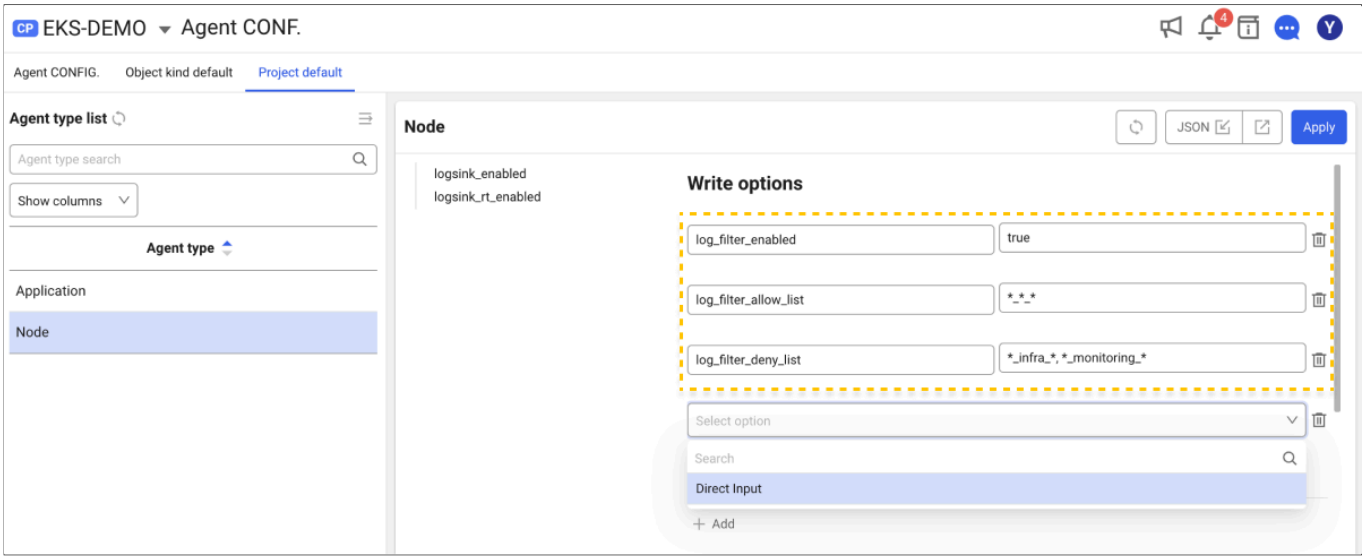
| Category | Description |
|---|---|
| #K8sEvent | • Logs that store the events that occur in Kubernetes<br>• Generation by default regardless of user settings |

| Category | Description |
| --- | --- |
| `#WhatapEvent` | • Logs that store the events that occur by WhaTap event settings<br>• Generation by default regardless of user settings |
| `containerStdout` | • Container Standard Out logs<br>• Generation by user settings<br>• When `logsink_enabled=true` setting is added to the node agent |
| `AppLog` | • Application logs in the container<br>• Generation by user settings<br>• 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.

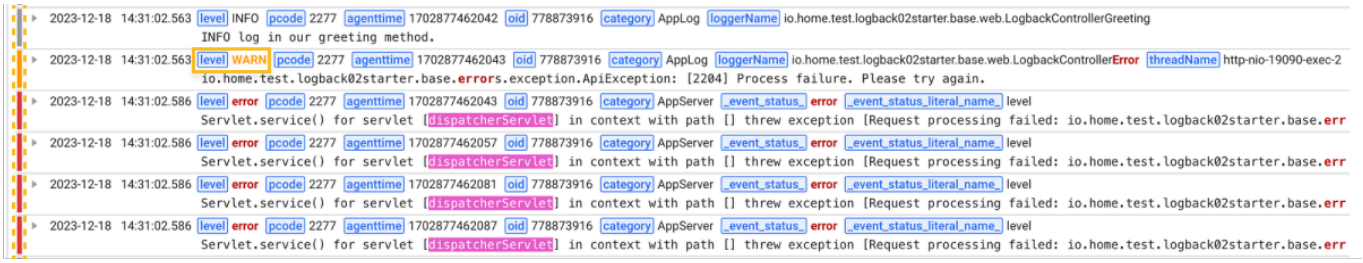
The major terms are as follows:

- **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

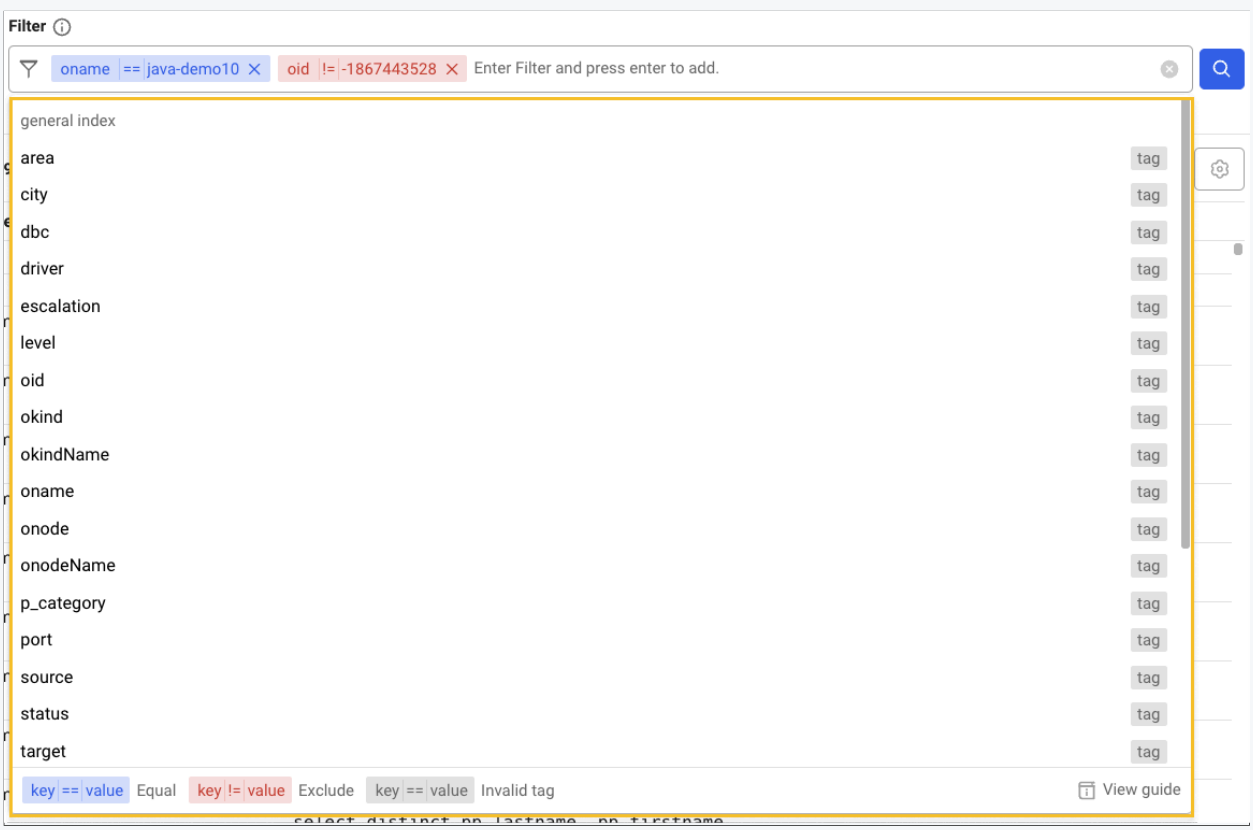
## Apply Filter

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

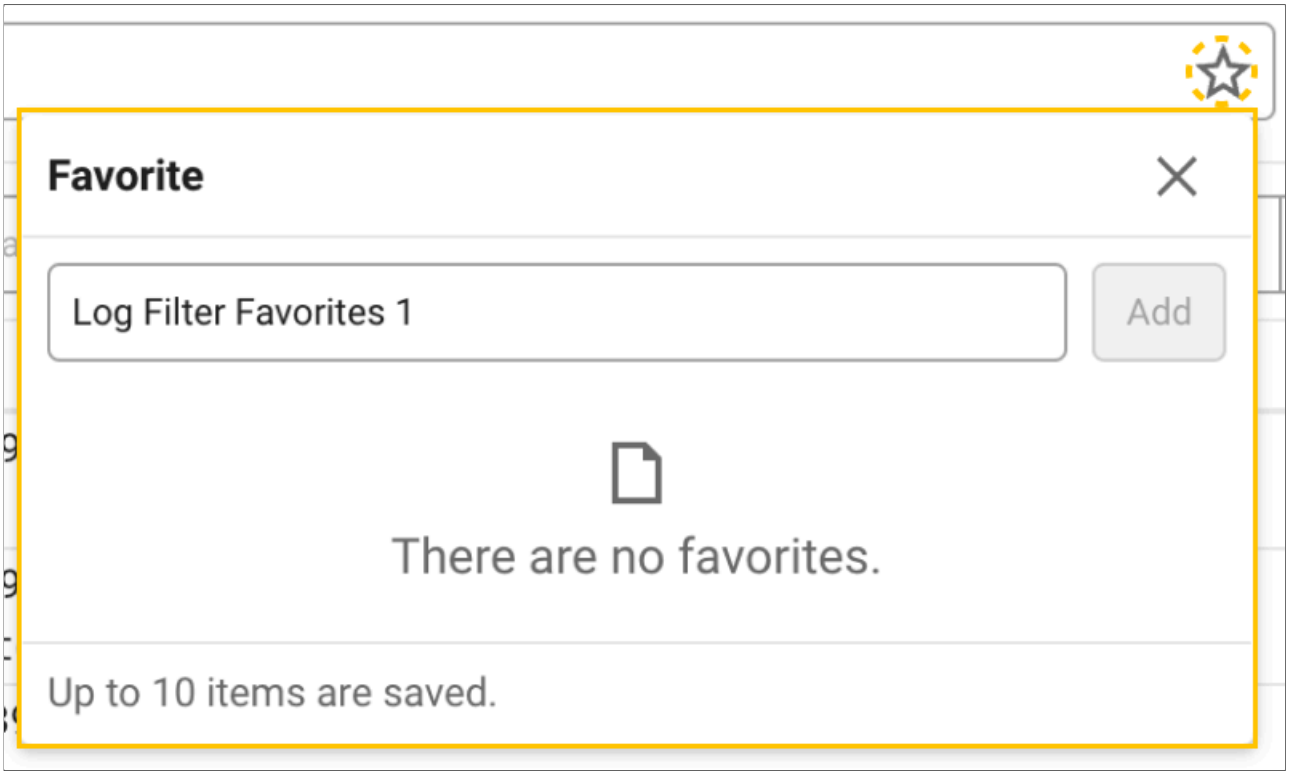
To add a filter tag, type the desired text in the input field and press **Enter** or **Tab**.
You can also select a recommended value from the **Guide UI** below the input field by clicking it or navigating with the arrow keys.

### 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

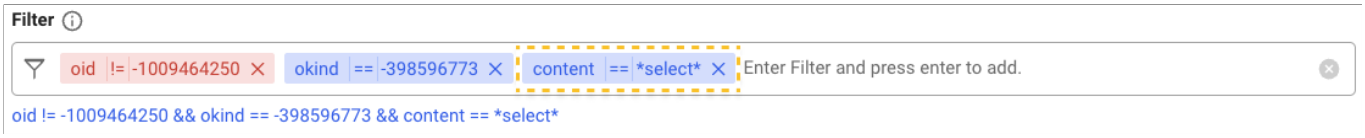
- In the tag with a numeric index ( `search key` that ends with `.n` ), only numbers can be input for the `search value` .
- Duplicate 'search key' and 'search value' cannot be entered.
- Searching is not possible if there is a tag whose `search key` or `search value` is missing. Invalid tags appear 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**.

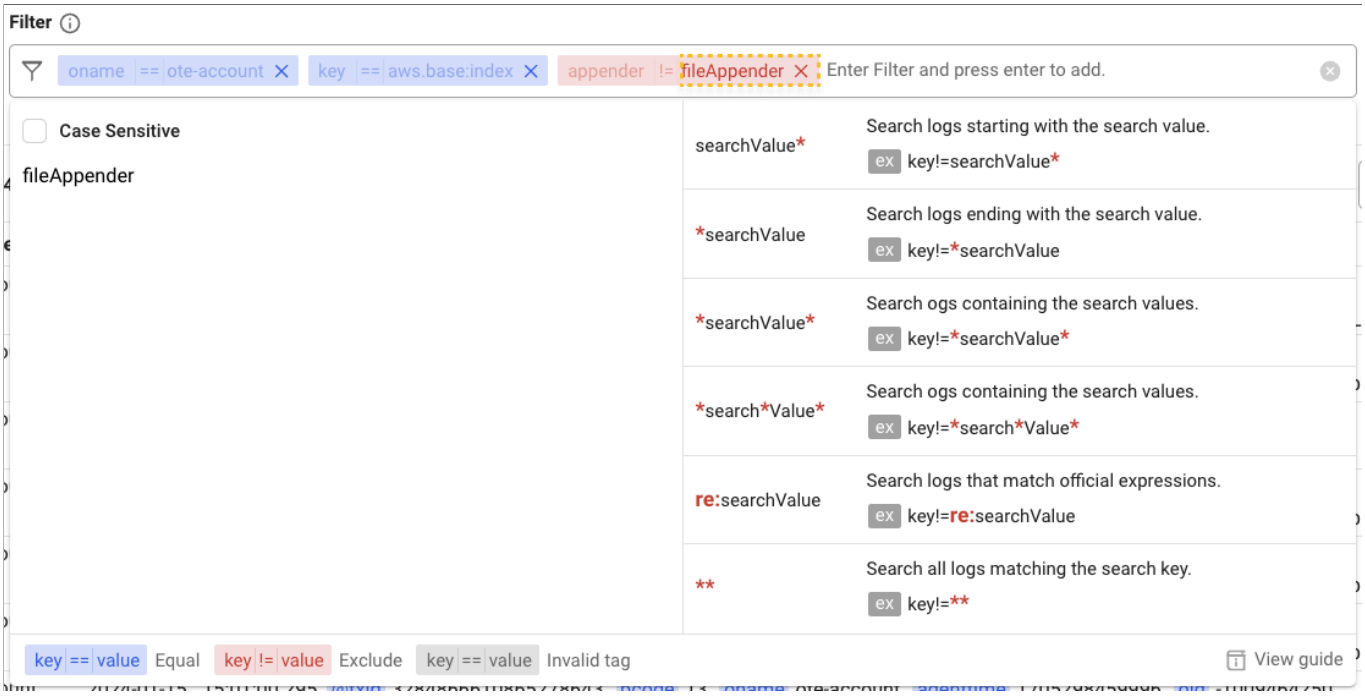



1. Select **Category**.
2. In the **Filter** field, enter a desired keyword for a search after leaving spaces on the `content` basis.
   e.g. `content *select*`
3. Click **Search** to view the logs.

> - 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<br>- Value: /data/whatap/logs/yard.log | fileName:/data/whatap/logs/yard.log |
| Numeric keyword | keyword.n | Response Time | - Key: response_time.n<br>- Value: 2945 | response_time.n>=2945 |
| Reserved keyword (predefined keyword) | @keyword | Transaction ID | - Key: @txid<br>- Value: 85459614215434144 | - |
| Log body's keyword | content | Log body | - Key: content<br>- 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 |

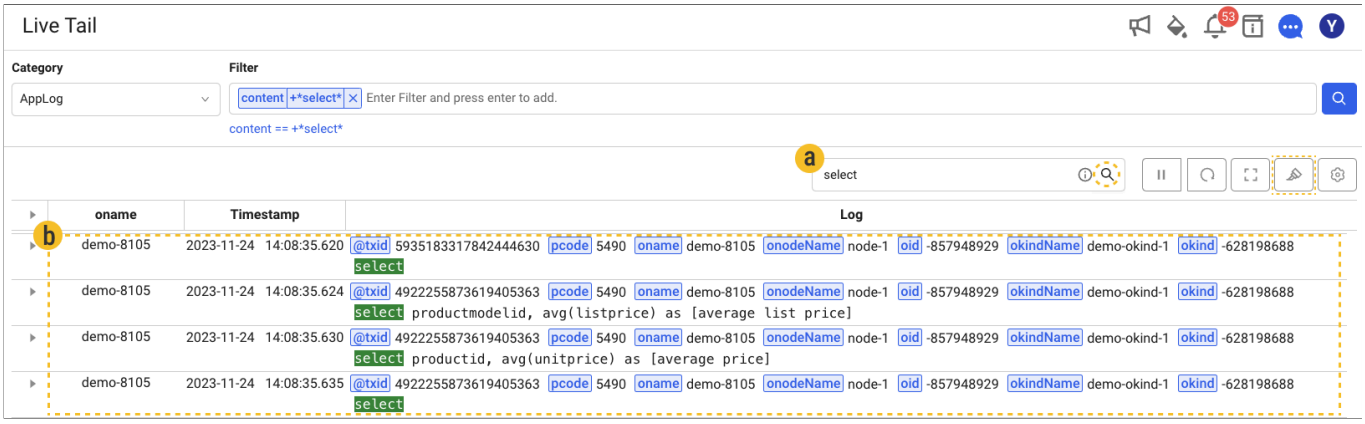
### Viewing personal information

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

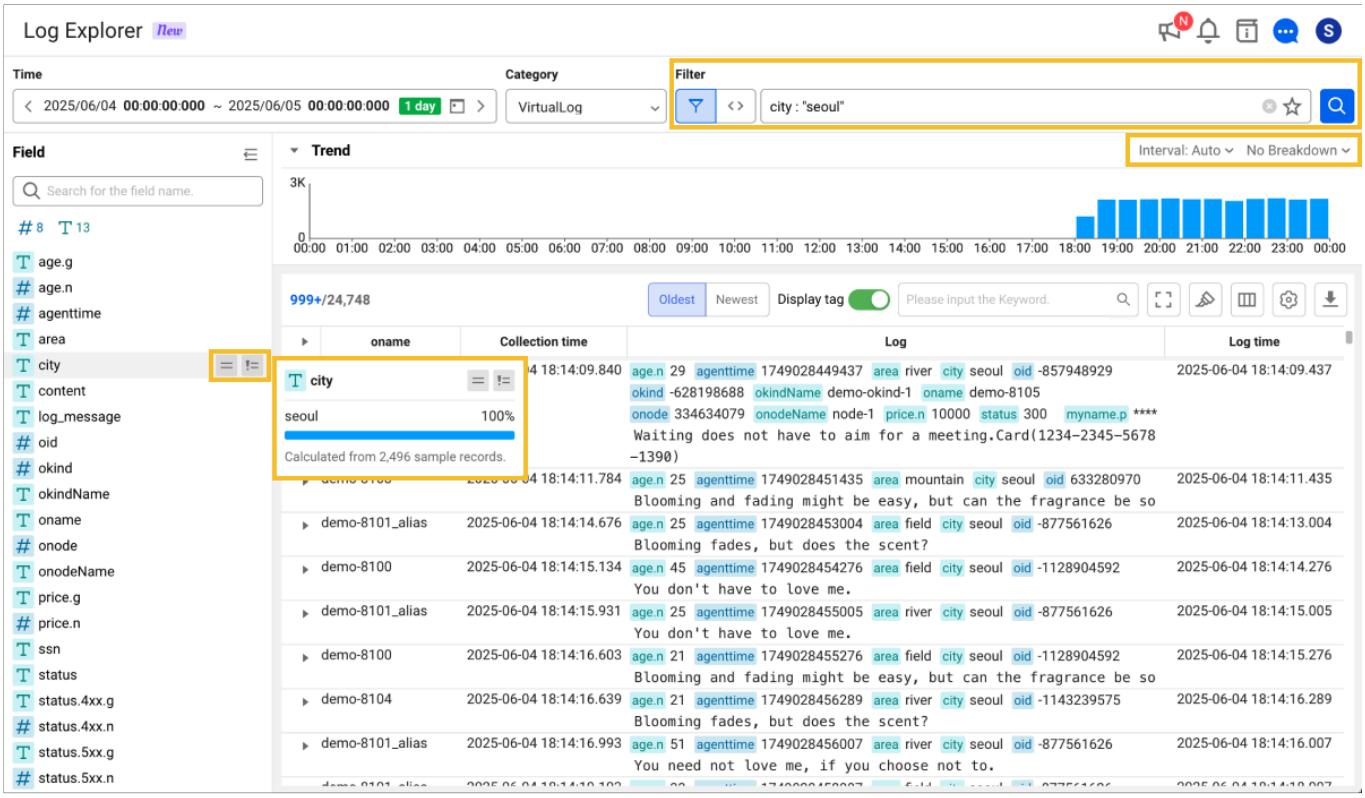
> ⓘ Without the **Log Read** role, you cannot enter the menu.

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

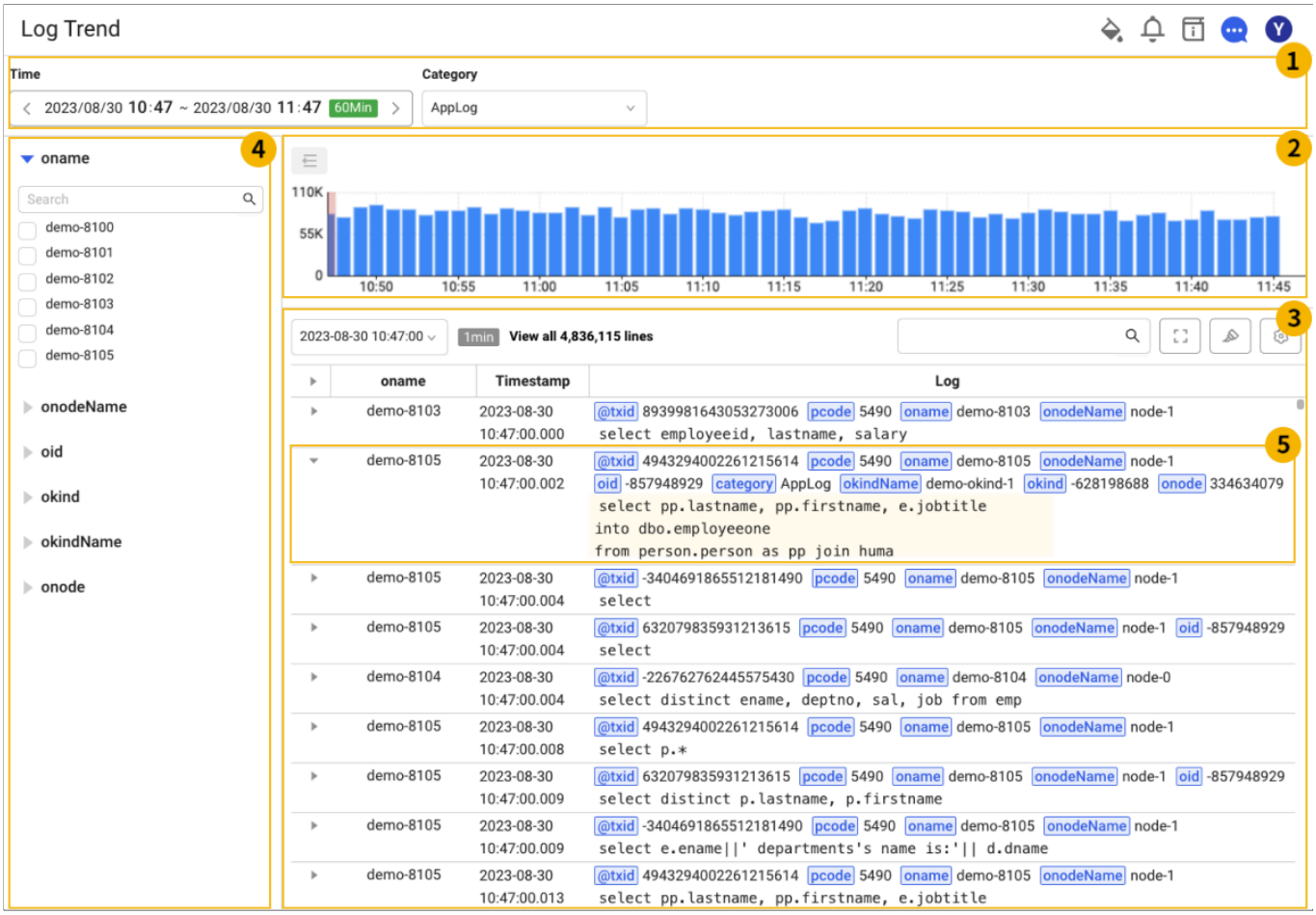
> ⓘ Without the **Log Read** role, you cannot enter the menu.

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:

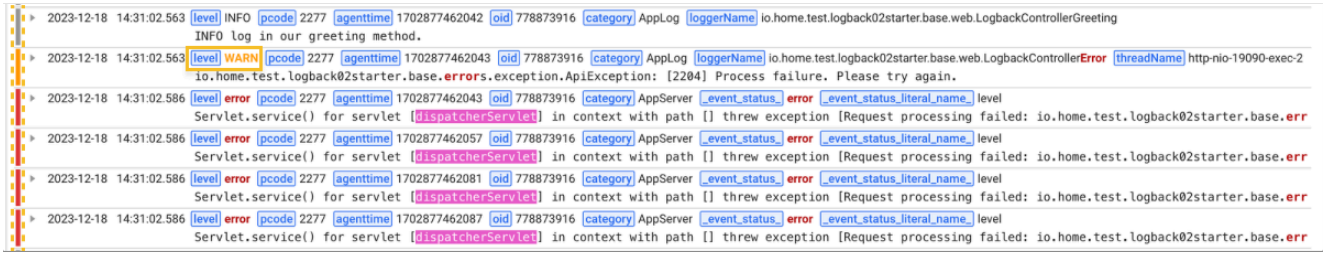
- **Category**: Unit for log collection and inquiry
- **Content**: Log messages
- **Search Key**: It is generated through the log parser settings.
- **Tag**: Key to search the collected logs




# 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:

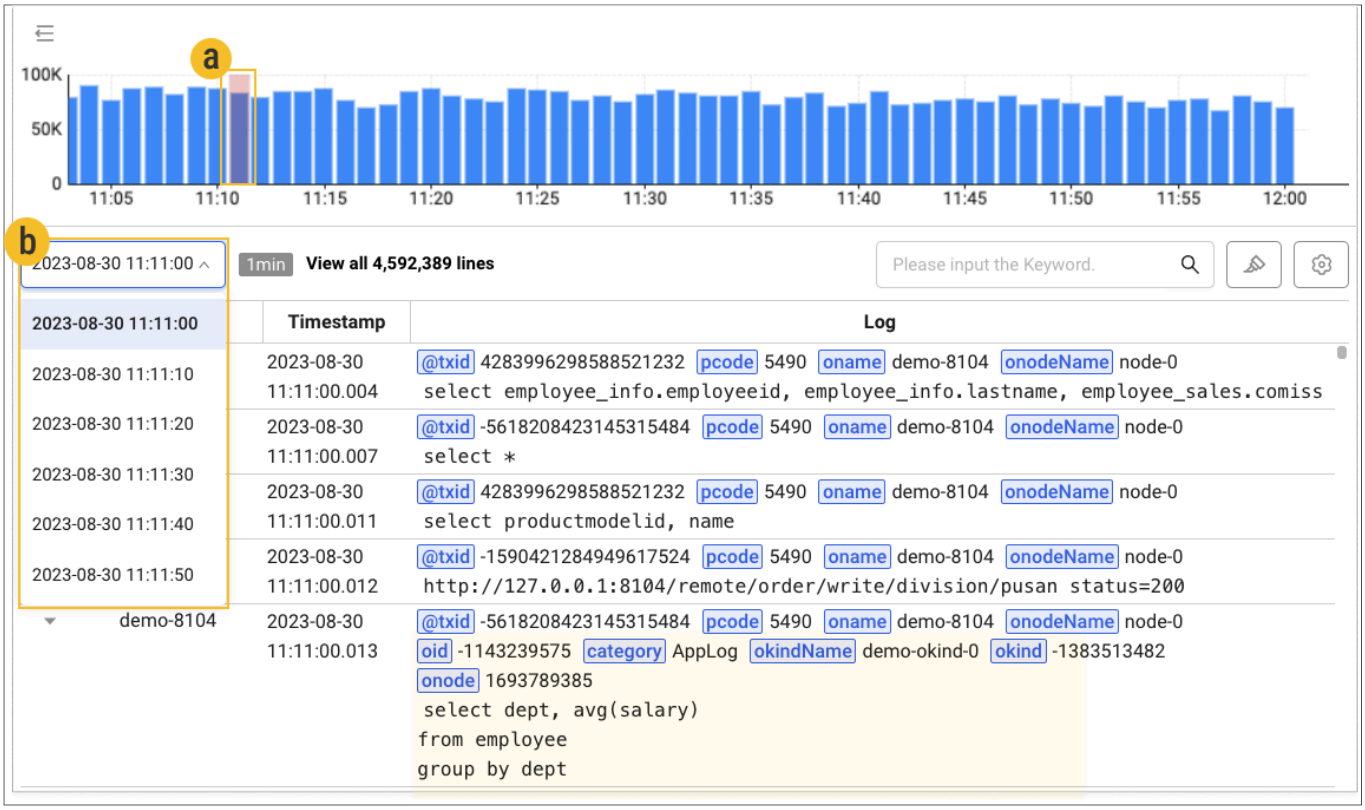
| String to input | Description | Result |
|---|---|---|
| `a b c` | Separate each keyword with a space. | a, b, c |
| `"Whatap is good."` | To include spaces in keywords, wrap them in `''` or `""`. | Whatap is good. |
| `"Whatap\\ is good."` | If `\` is included in the keyword surrounded by "", enter 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.

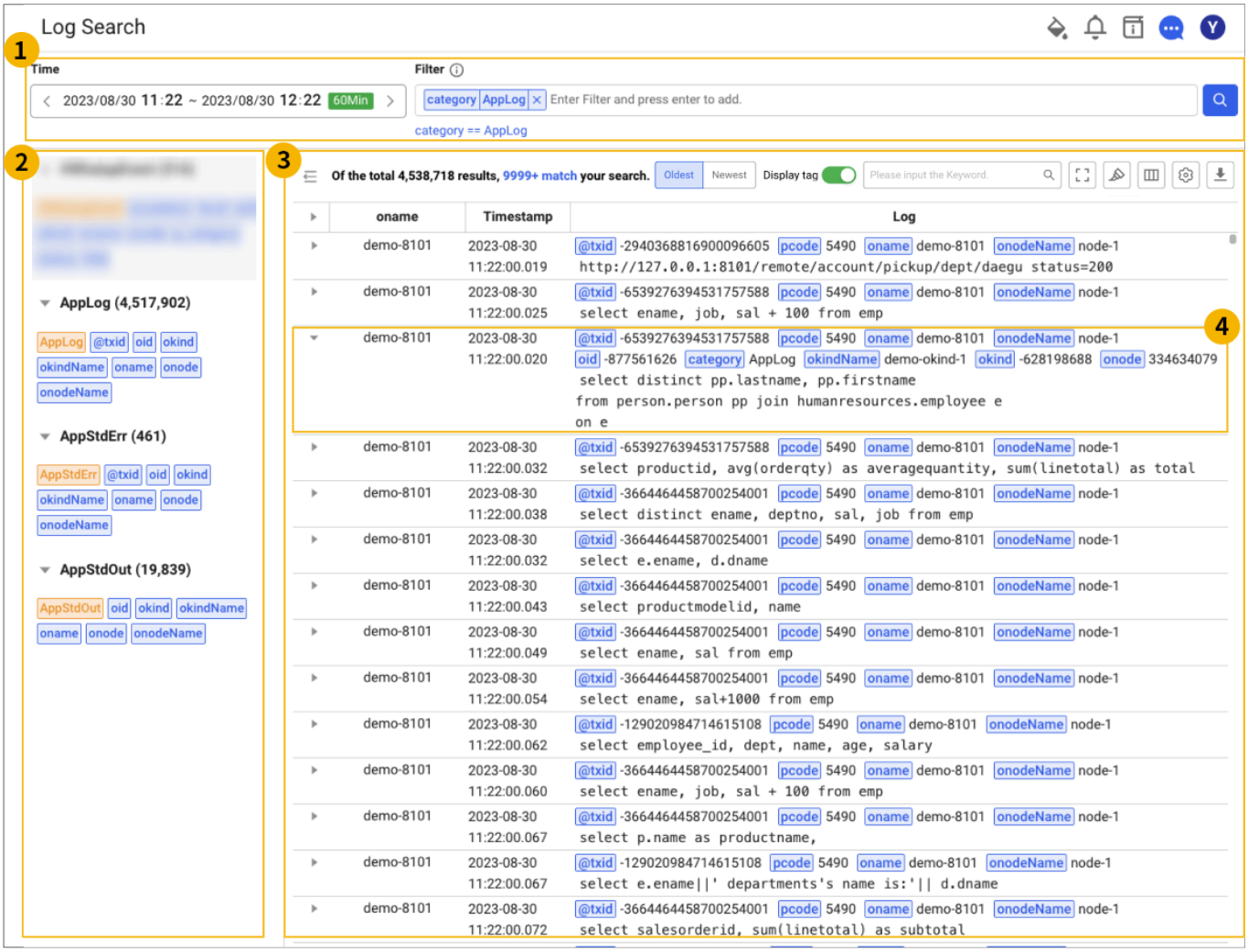
Home > Select Project > **Log** > **Log Search**

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

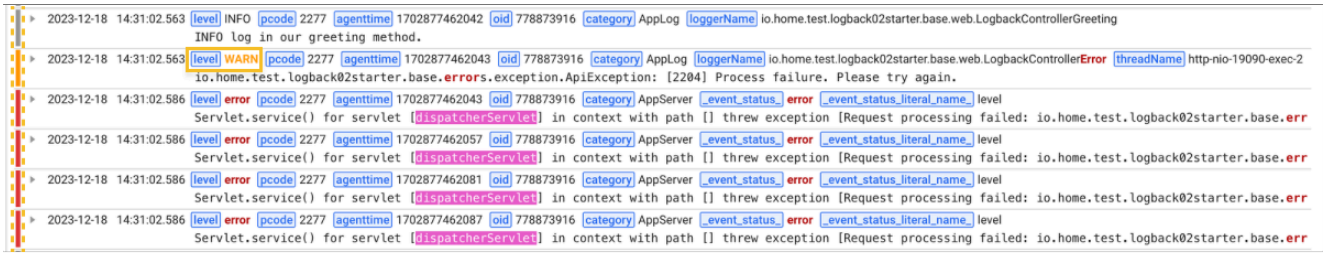



# Request Data

- 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**.

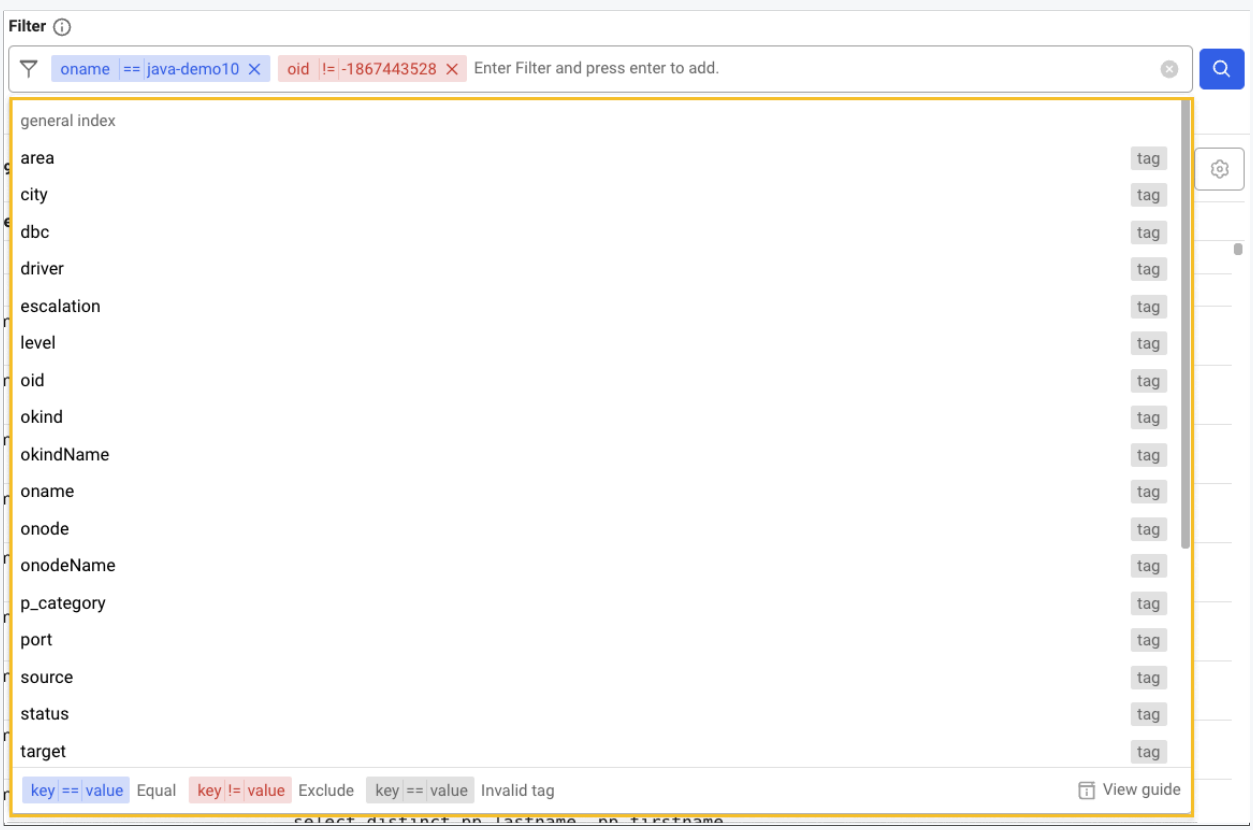
# Filter

## 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

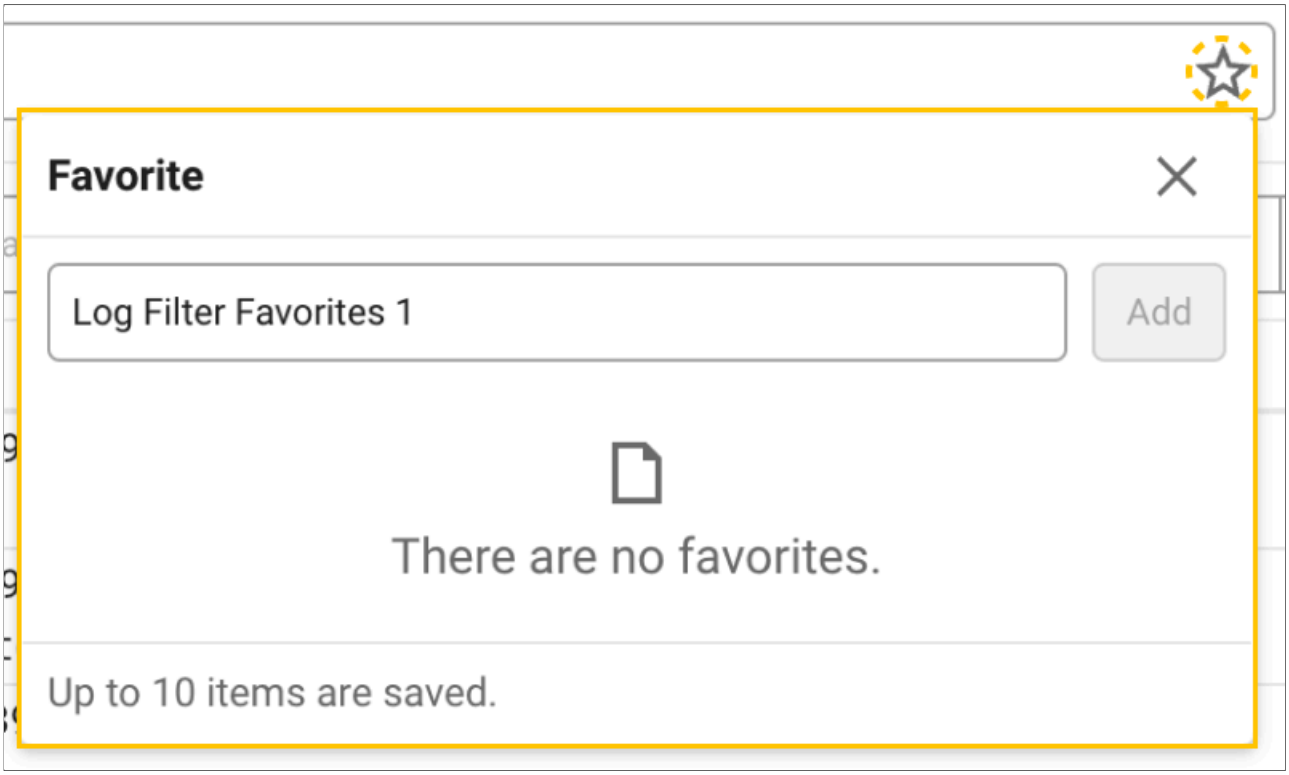
To add a filter tag, type the desired text in the input field and press **Enter** or **Tab**.
You can also select a recommended value from the **Guide UI** below the input field by clicking it or navigating with the arrow keys.

### 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.

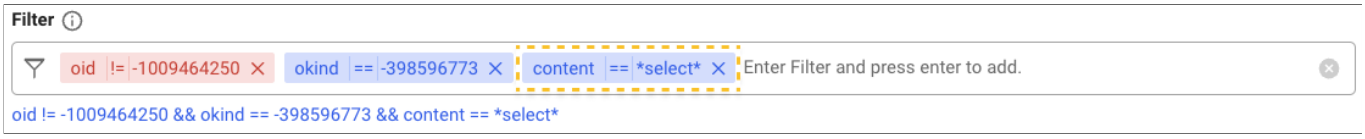



### Exceptional cases for application of filters

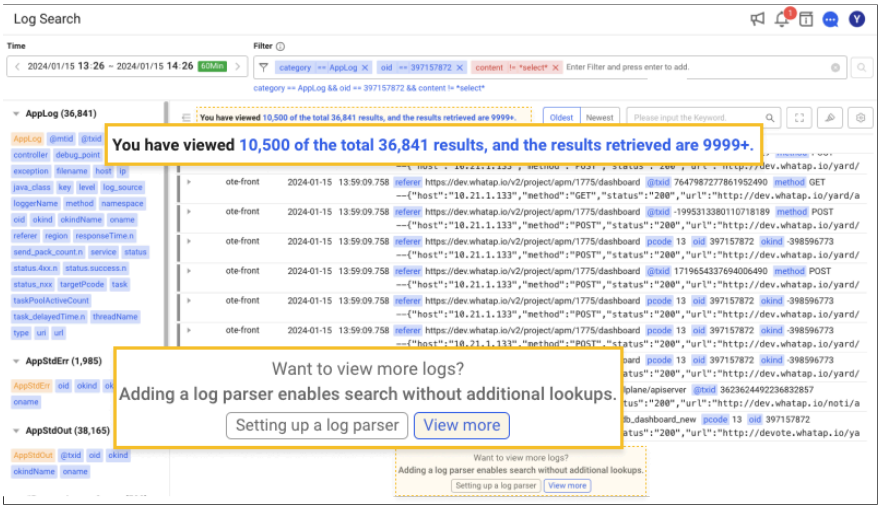
- In the tag with a numeric index ( `search key` that ends with `.n` ), only numbers can be input for the `search value` .
- Duplicate 'search key' and 'search value' cannot be entered.
- Searching is not possible if there is a tag whose `search key` or `search value` is missing. Invalid tags appear 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**.



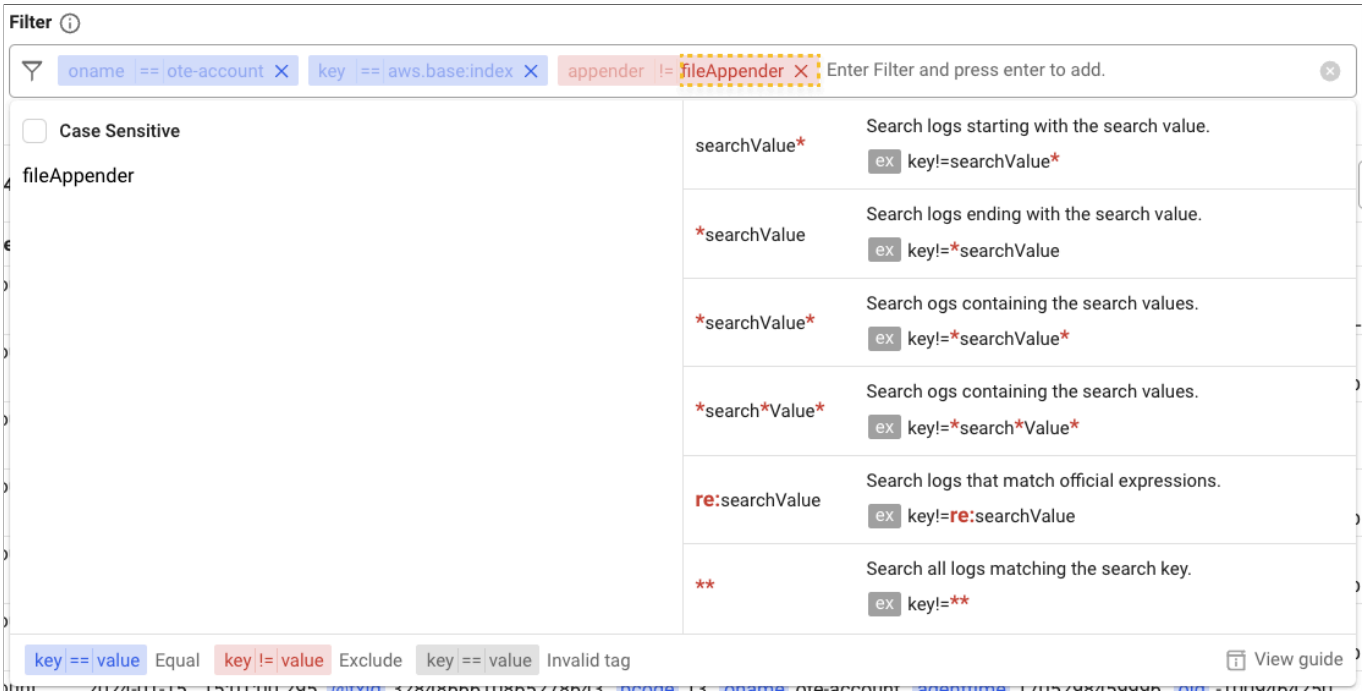

1. Select **Category**. It is required to set the category.
2. In the **Filter** field, enter a desired keyword for a search after leaving spaces on the `content` basis.
   e.g. `content *select*`
3. Click **Search** to view the logs. Several logs are viewed first. You can view a maximum of 10,000 logs 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.

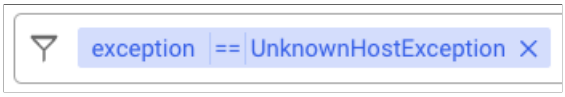
## Search Key

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<br>- Value: /data/whatap/logs/yard.log | fileName:/data/whatap/logs/yard.log |
| Numeric keyword | keyword.n | Response Time | - Key: response_time.n<br>- Value: 2945 | response_time.n>=2945 |
| Reserved keyword (predefined keyword) | @keyword | Transaction ID | - Key: @txid<br>- 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 |

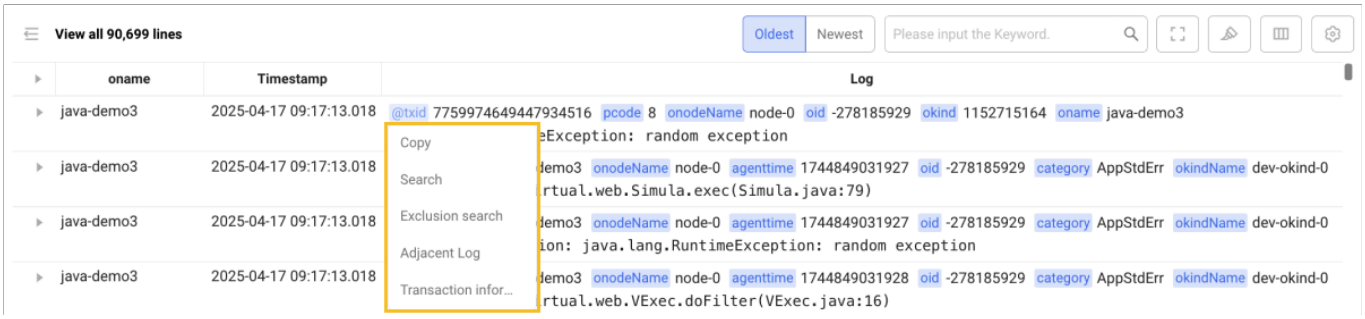
### Viewing personal information

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**.

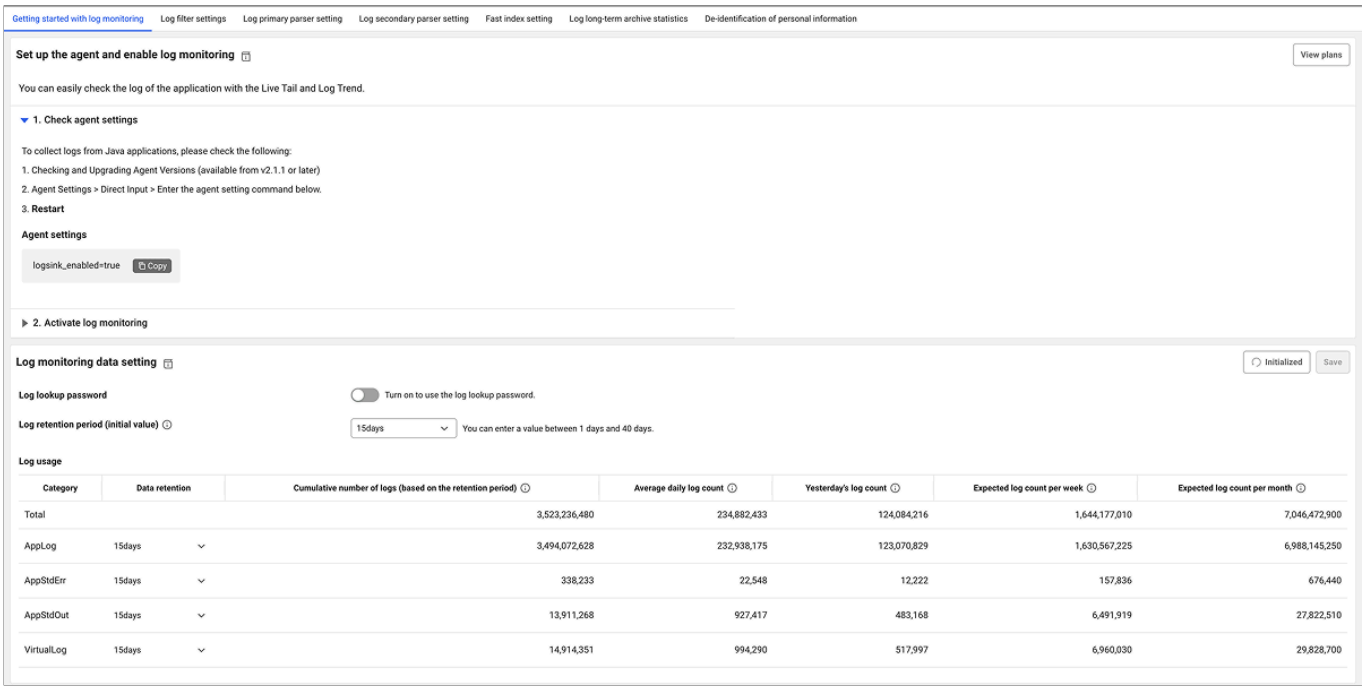
# Log Setting

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.

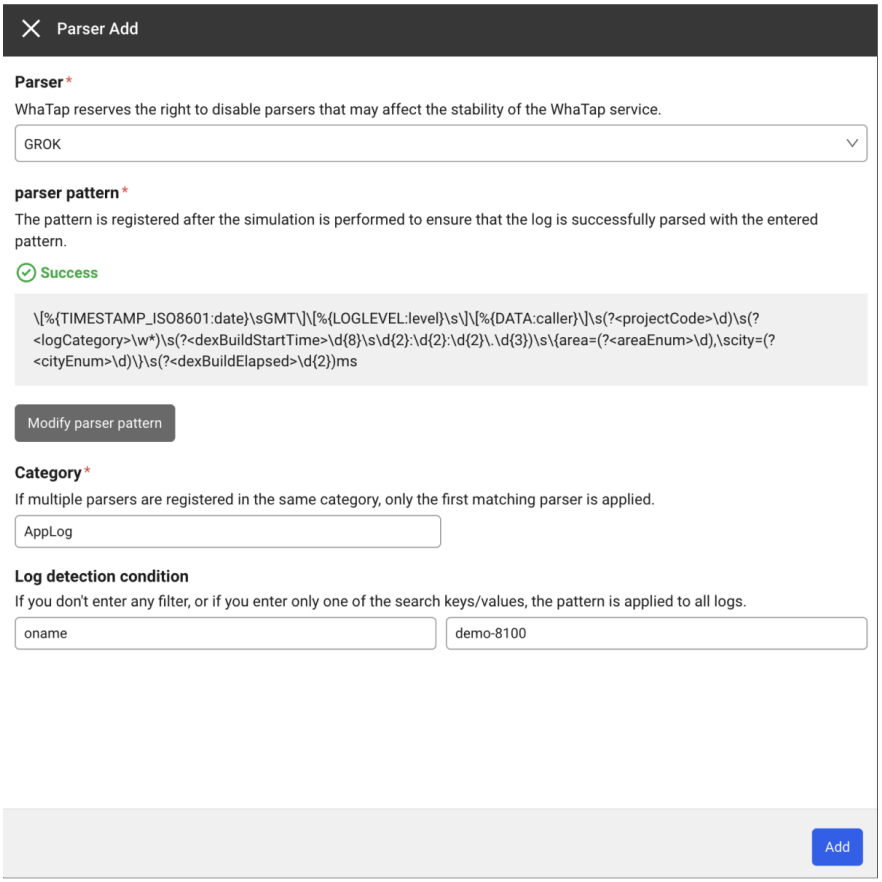
   **Pattern** e.g.

   ```
   [%{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.

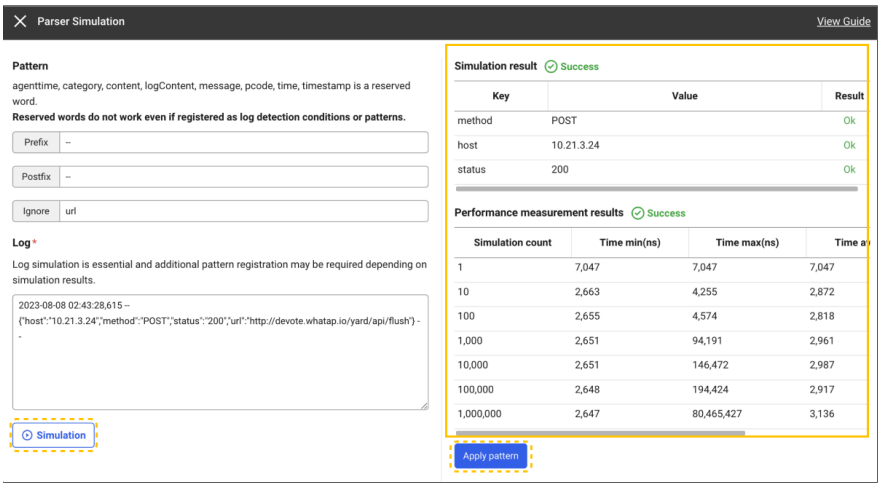
   Example

   ```
   2023-08-08 02:43:28,615 -- {"host":"10.21.3.24","method":"POST","status":"200","url":"http://dev.whatap.io/yard/api/flush"} --
   ```

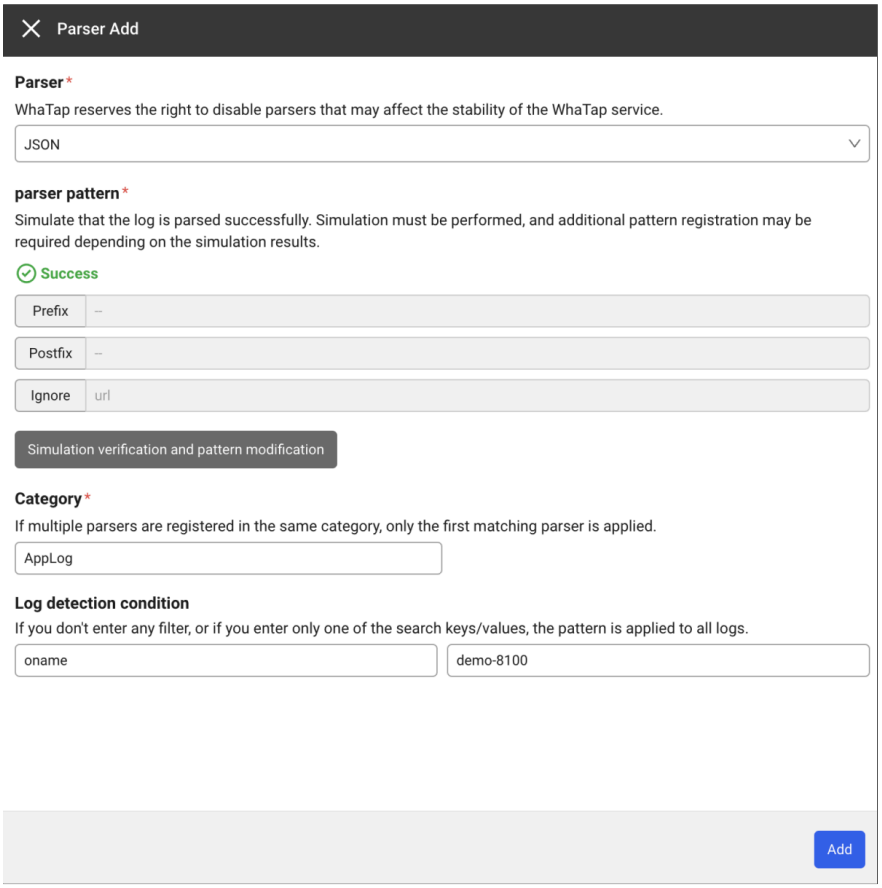
   In the example log, if **Prefix** and **Postfix** are specified as `--` and `url` is specified in **Ignore**, only three **Tag** ( `host` , `method` , `status` ) are created.

   - **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:

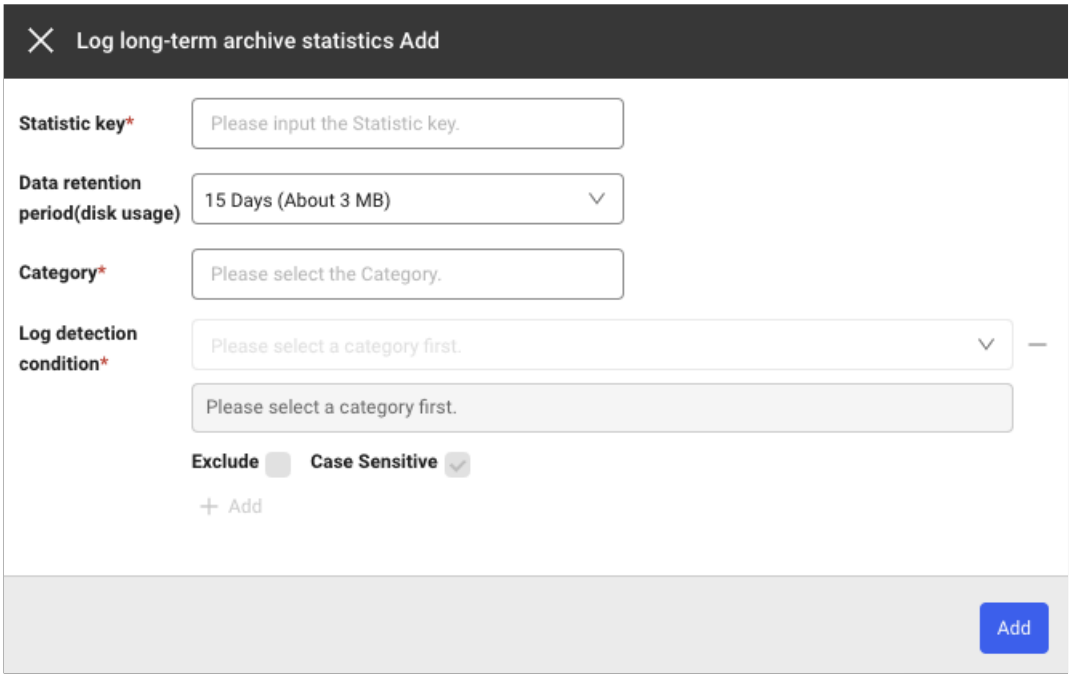
| Set value | Required | Description |
|---|---|---|
| Category | Mandatory | Category to be set as fast index |
| Search Key | Mandatory | Search key for fast index setting |
| Case insensitive | Option | Whether to be case sensitive |
| Rule | Mandatory | `*` must be included at least one. |
| Enabled | 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



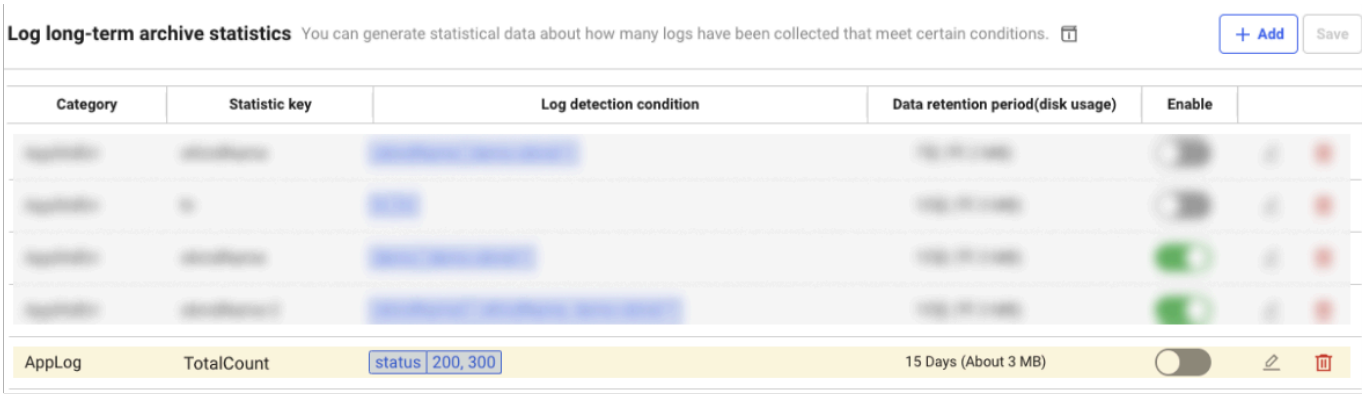

If you select **+ Add** under the **Log long-term archive statistics** tab, the **Log long-term archive statistics Add** window appears. You can add rules by using **+ Add** or delete the created rules by using the **-** icon.

## 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`) |

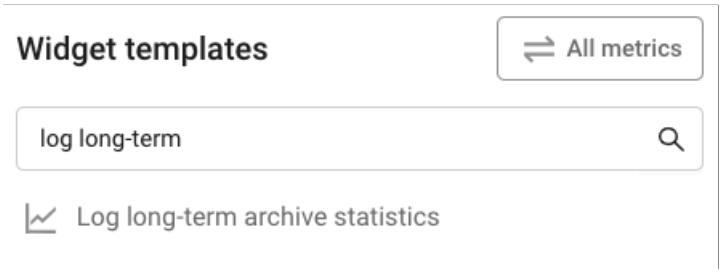
## Example

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`.

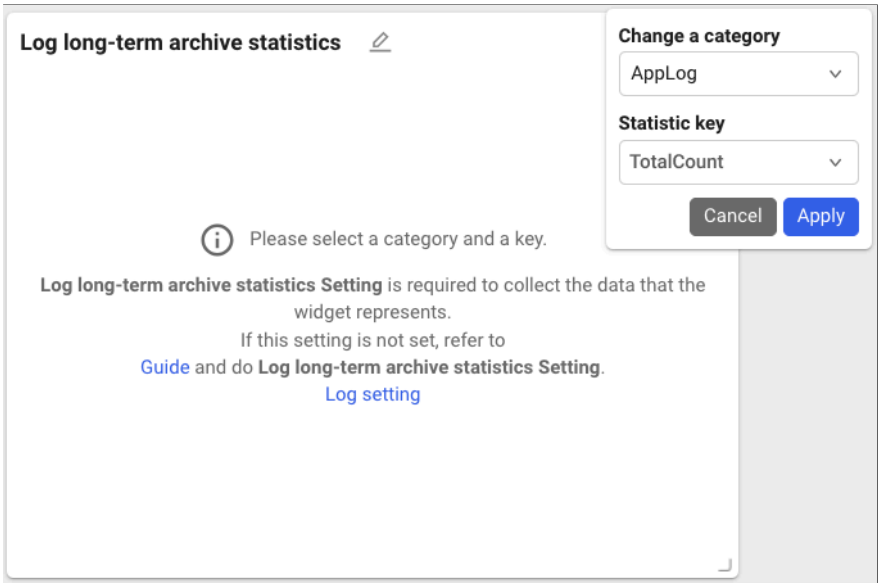


## Data Search

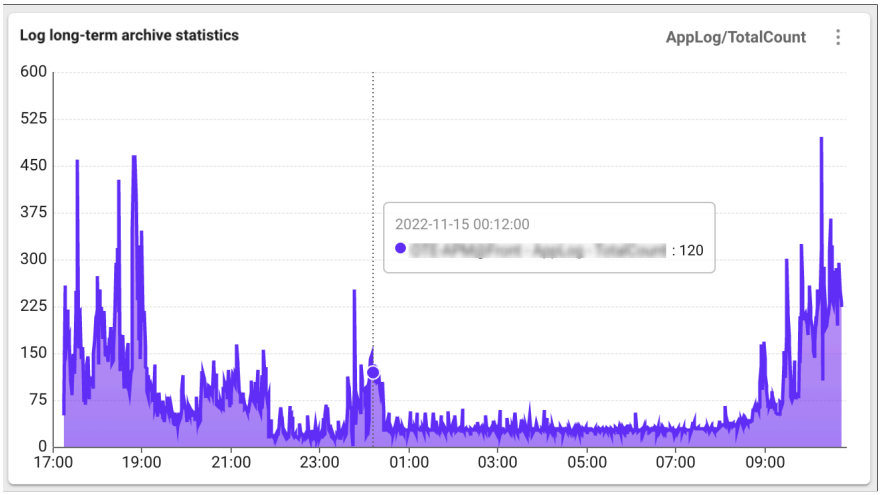
1. Create a widget by searching **Log long-term archive statistics** in **Widget templates** of **Integrated Flex Board**.




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

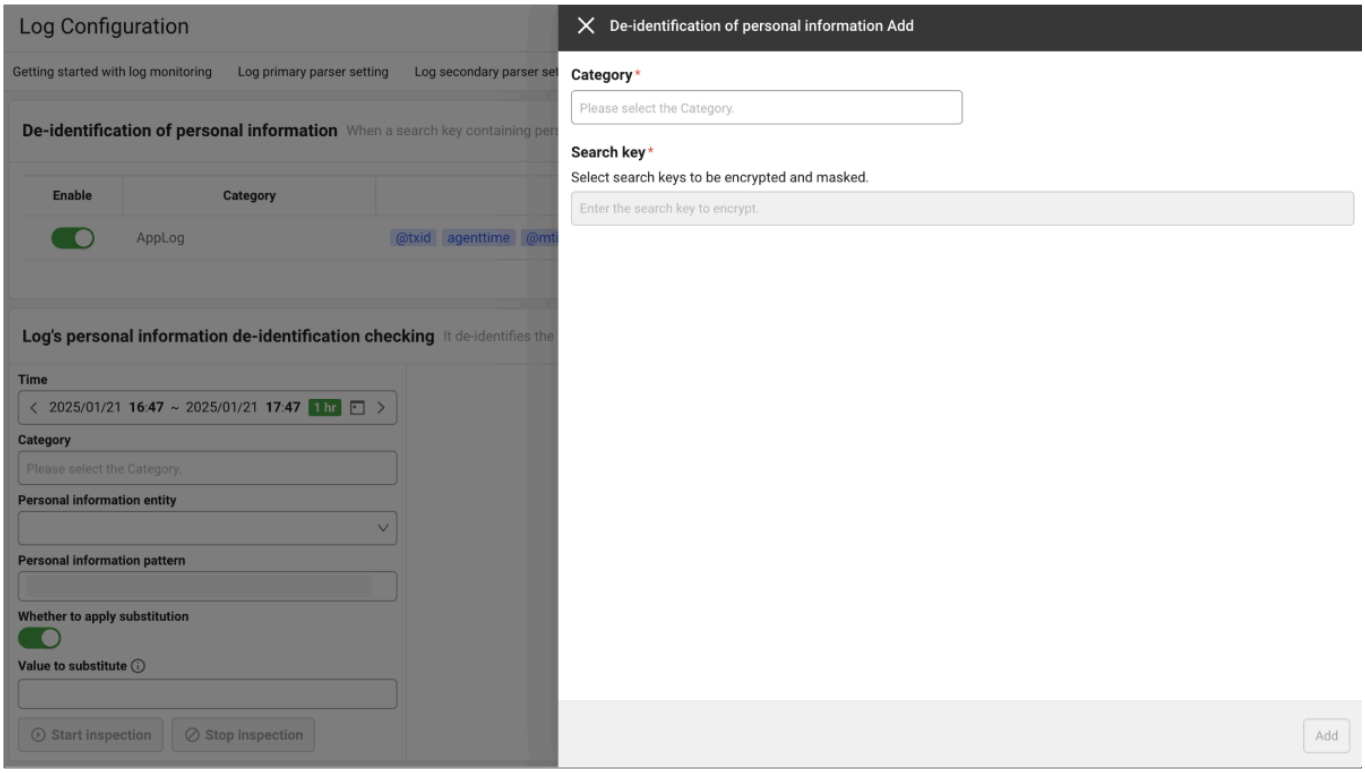
### Required roles

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.

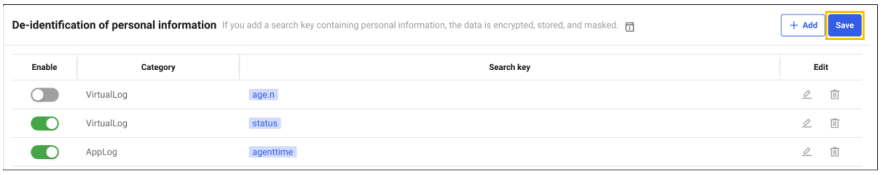
> For more information about the **log personal information query** and **log editing** roles, see the following.

## 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.

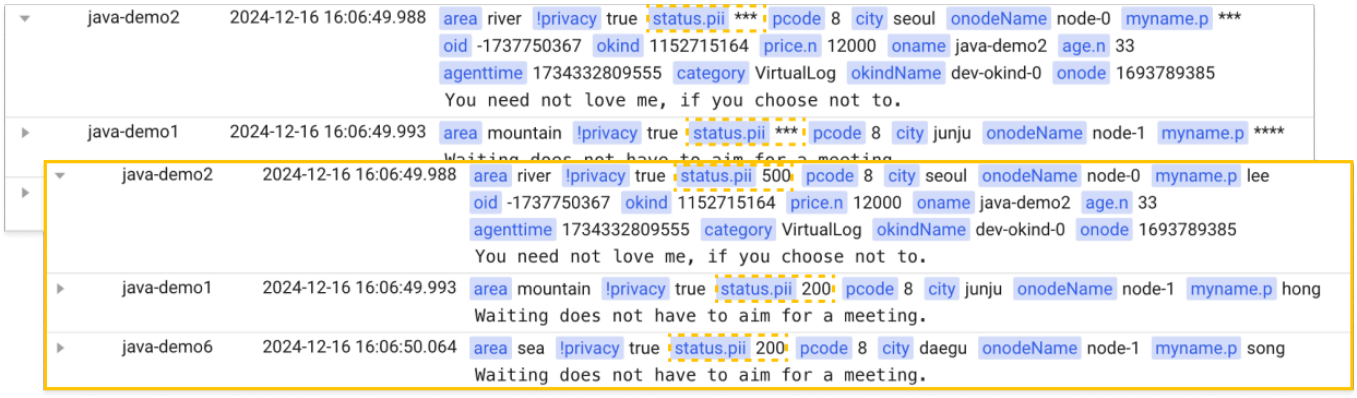



- The de-identified search keys are output in the format of `search key.pii` in the log content.
- You can perform **Enable** for de-identified items in the **Enable** column of the **De-identification of personal information** list.
- You can modify and remove in the **edit** column of the **De-identification 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.

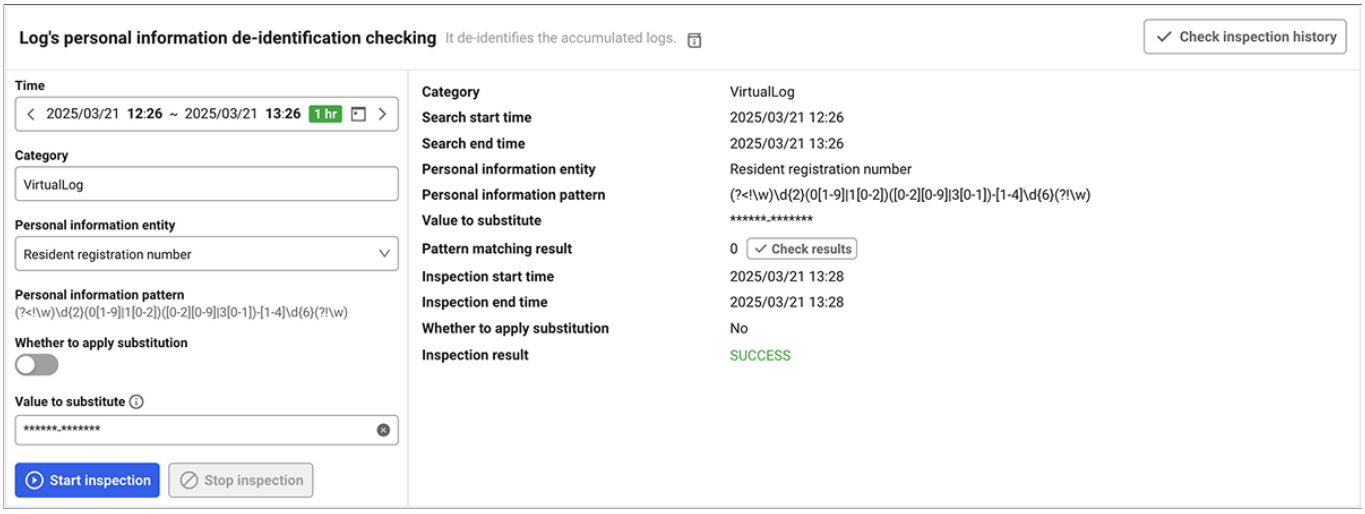
**Log's personal information de-identification checking** identifies personal information in existing log data and replaces it with the **Value to substitute** specified by the user.

### 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.

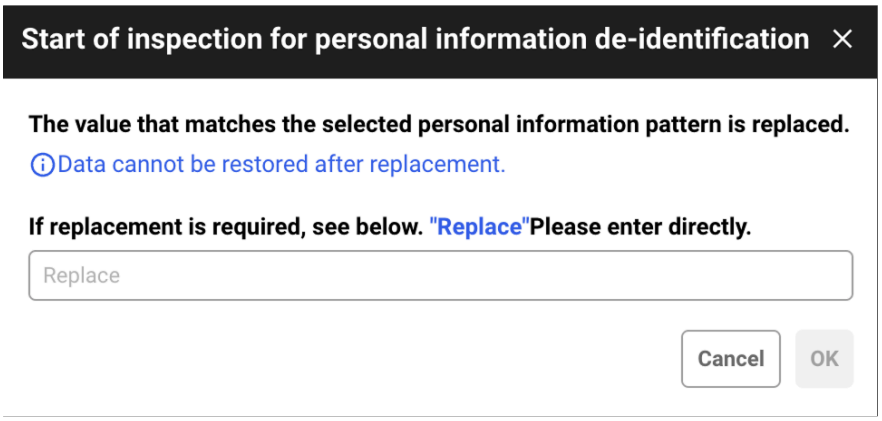
4. Select whether to replace personal information, if it is identified via the **Whether to apply substitution** toggle.
5. Enter **Value to substitute** to change the string identified by the regular expression pattern in the existing log.

   > - **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.



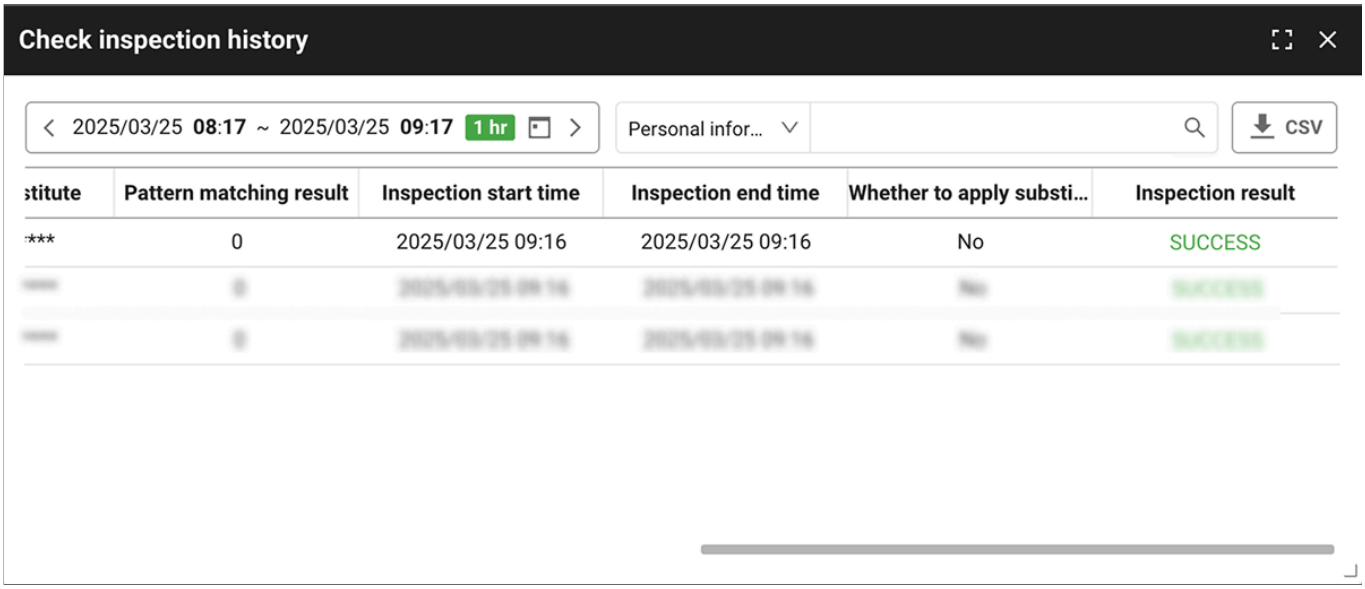

8. After the inspection is finished, you can see **Inspection result** on the right as in the example image.

   - **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.

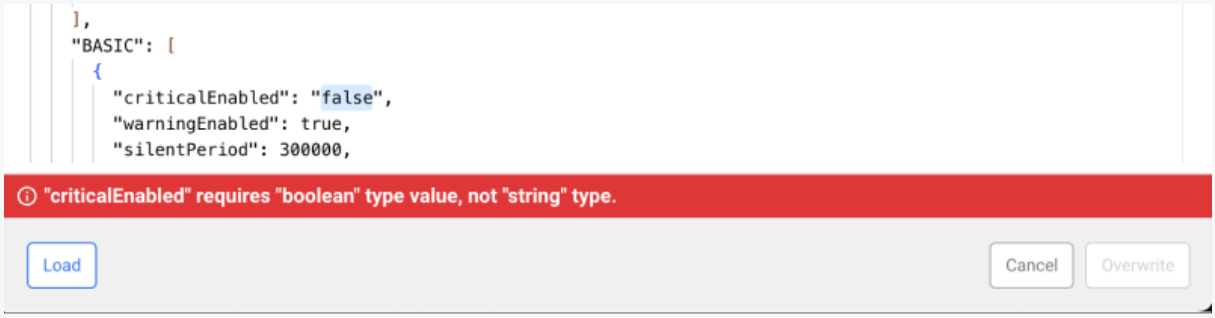
- **Batch JSON Edit**: Edit all event conditions in bulk using JSON format.
- **Batch JSON Download**: Save all currently registered event conditions as a JSON file.
- **Batch Excel Download**: Download the event list as an Excel file for external reporting or backup purposes.

## 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<br>– Enabled: An event is triggered when conditions are met<br>– Disabled: The event does not run and conditions are not evaluated |
|  | Edit or delete event settings |
| Event Name | User-defined event name<br>– 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<br>– 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<br>– 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<br>– 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<br>– Enabled: Sends a RECOVERED notification when the status changes from in progress to normal and allows viewing active and resolved states in event history<br>– 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.

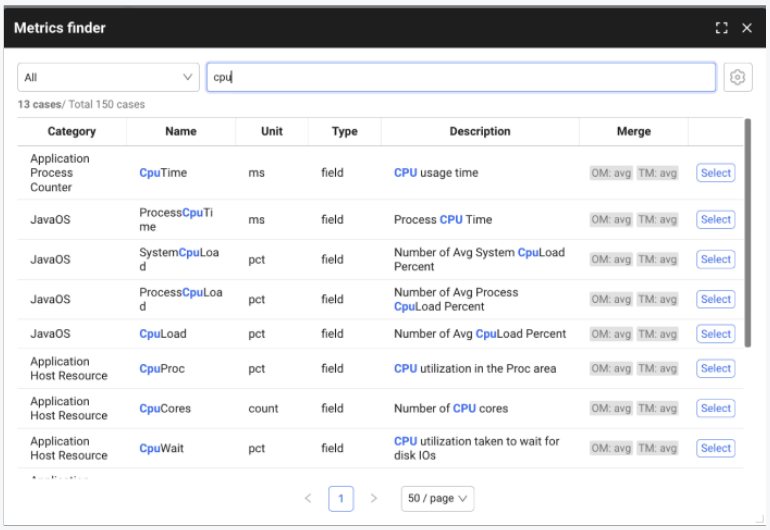
### Metric Settings

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

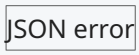
1. Click the [ ] button at the top right of the screen.
2. Select the JSON file downloaded using the **Export** function.
3. When the JSON editor opens, choose either [ Add to List ] or [ Overwrite ].

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<br>- Up to 255 bytes |
| Category | Log classification name (log folder name) |
| Search key | The field to search for in log data<br>- 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<br>- 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<br>- 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<br>- If no condition is set, alerts are evaluated for all collected data<br>- Select input: Build conditions by selecting tags, operators, and values<br>- 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<br>- Enabled: Event occurs when conditions are met<br>- 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<br>- Enabled: Sends a RECOVERED notification when transitioning from in-progress → normal status, and the in-progress/resolved status can be distinguished in event history<br>- 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<br>- 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<br>- 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<br>- Enabled: Event occurs when conditions are met<br>- 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

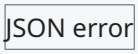
1. Click the [ ] button at the top right of the screen.
2. Select the JSON file downloaded using the **Export** function.
3. When the JSON editor opens, choose either [ Add to List ] or [ Overwrite ].

> 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<br>- Applies only to Critical alerts |
| Smart Message (AI) | AI-powered notifications available when adding the Telegram plugin<br>- 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**.

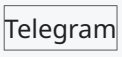
## How to Add 3rd-party Plugin Notifications

Slack

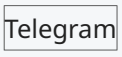
In the **Add 3rd-party Plugin** window, click the [ **Add To Slack** ] button on the **Slack** tab to register **WhaTapNotiBot** to your Slack team and channel.

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.

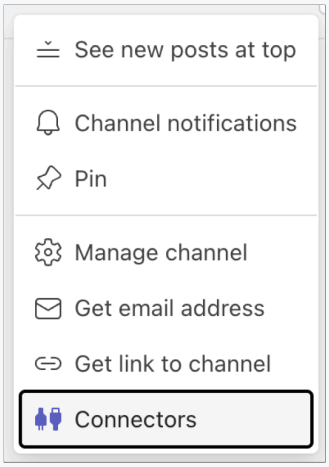
**Available commands**

- `/help` : View the list of commands
- `/license` : Enter the access key of the project to receive alerts

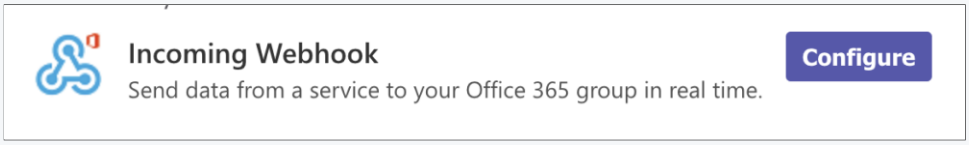
- `/remove` : Enter the project code for which you want to stop receiving alerts
- `/list` : View the list of project codes currently receiving alerts

Teams

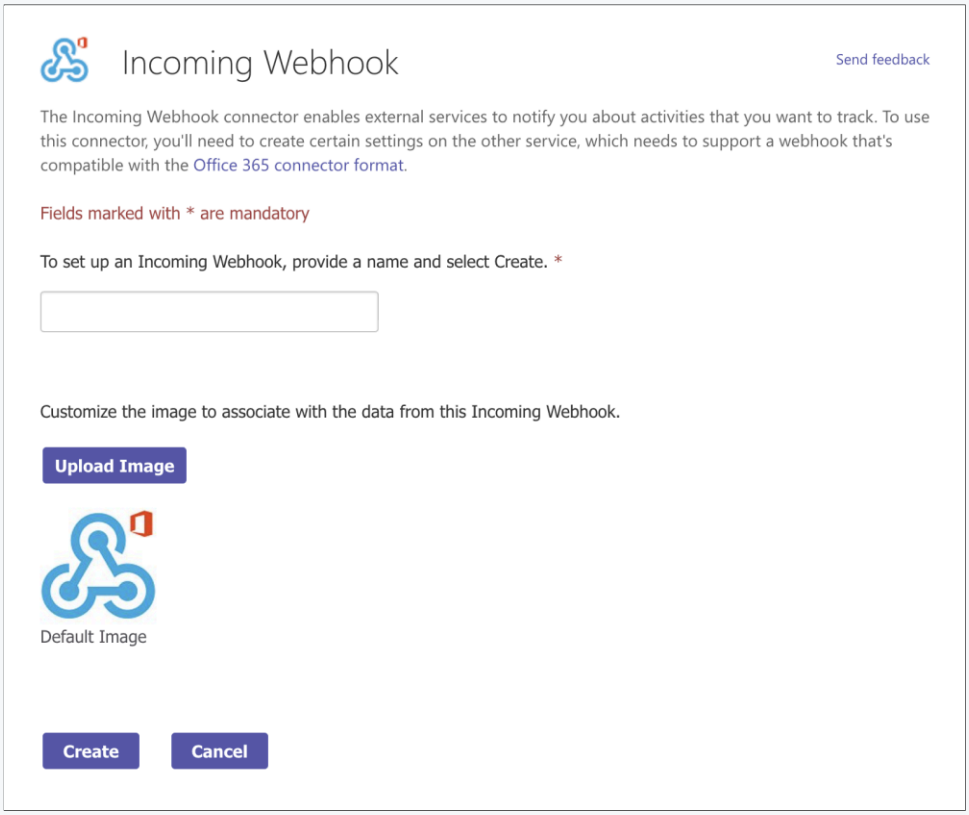
1. Go to the **connector** of the channel where you want to receive alerts.



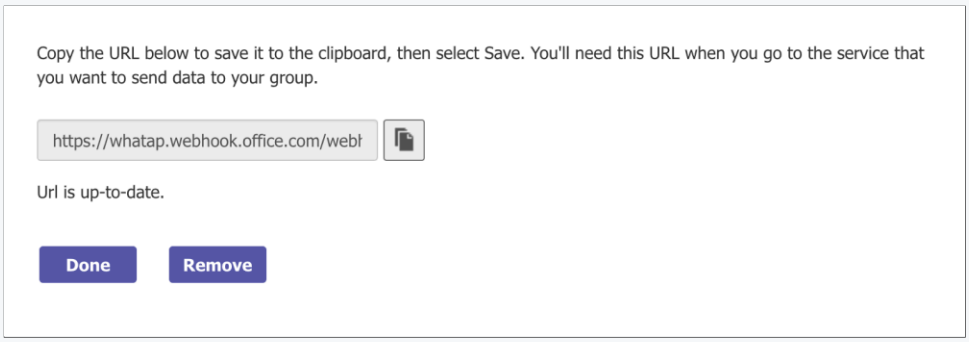

2. Add a **Webhook** and click the [ Configure ] button.




3. Enter a webhook name and click the [ Create ] button.




4. Copy the generated **Webhook URL**.

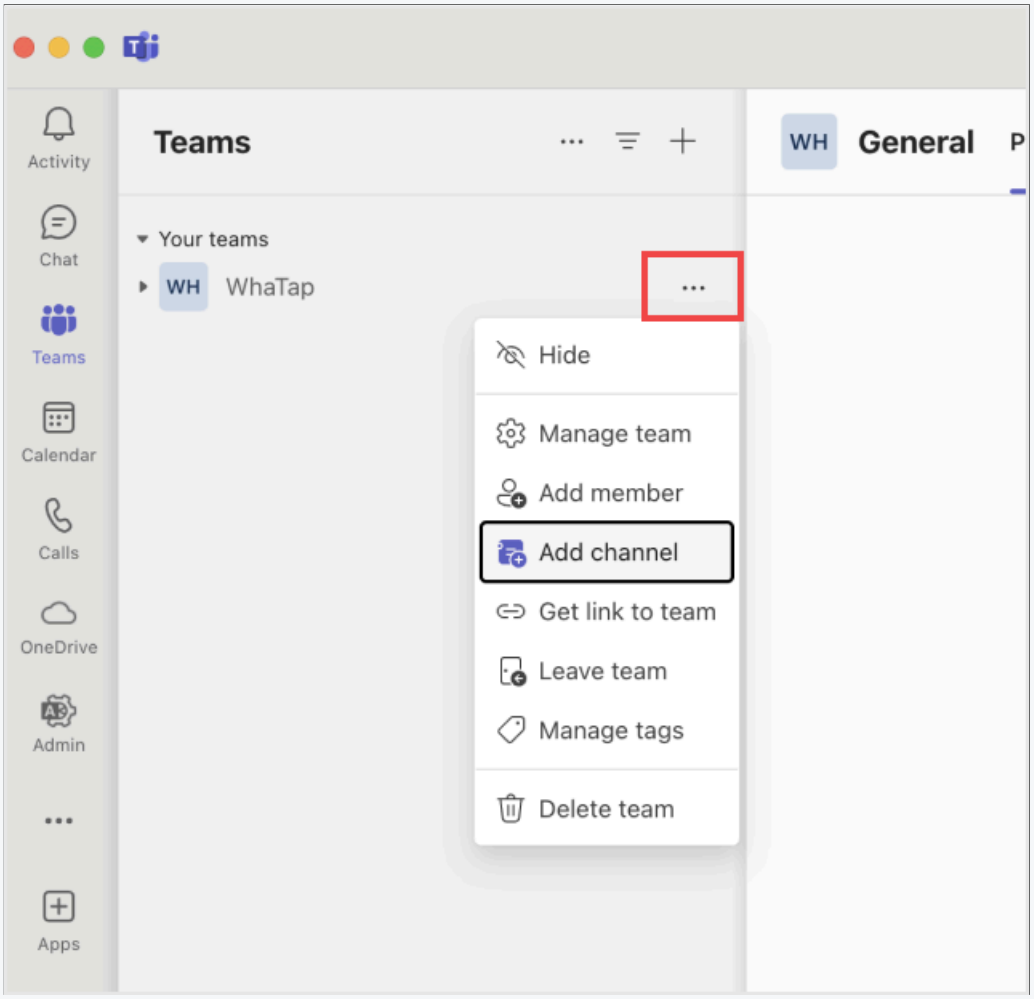



5. Return to the **Add a third-party plug-in** window in the WhaTap Monitoring service, and enter values for **Channel Name** and **Webhook URL** at the bottom of the **Teams** tab.
6. Click the [ **Registration** ] button.

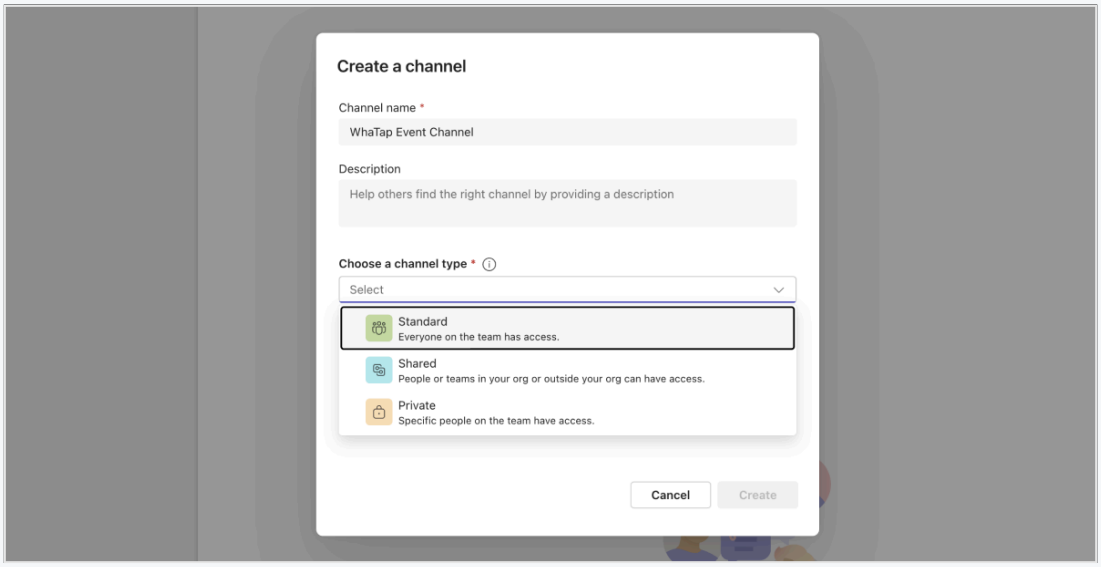
Teams Workflows

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**.

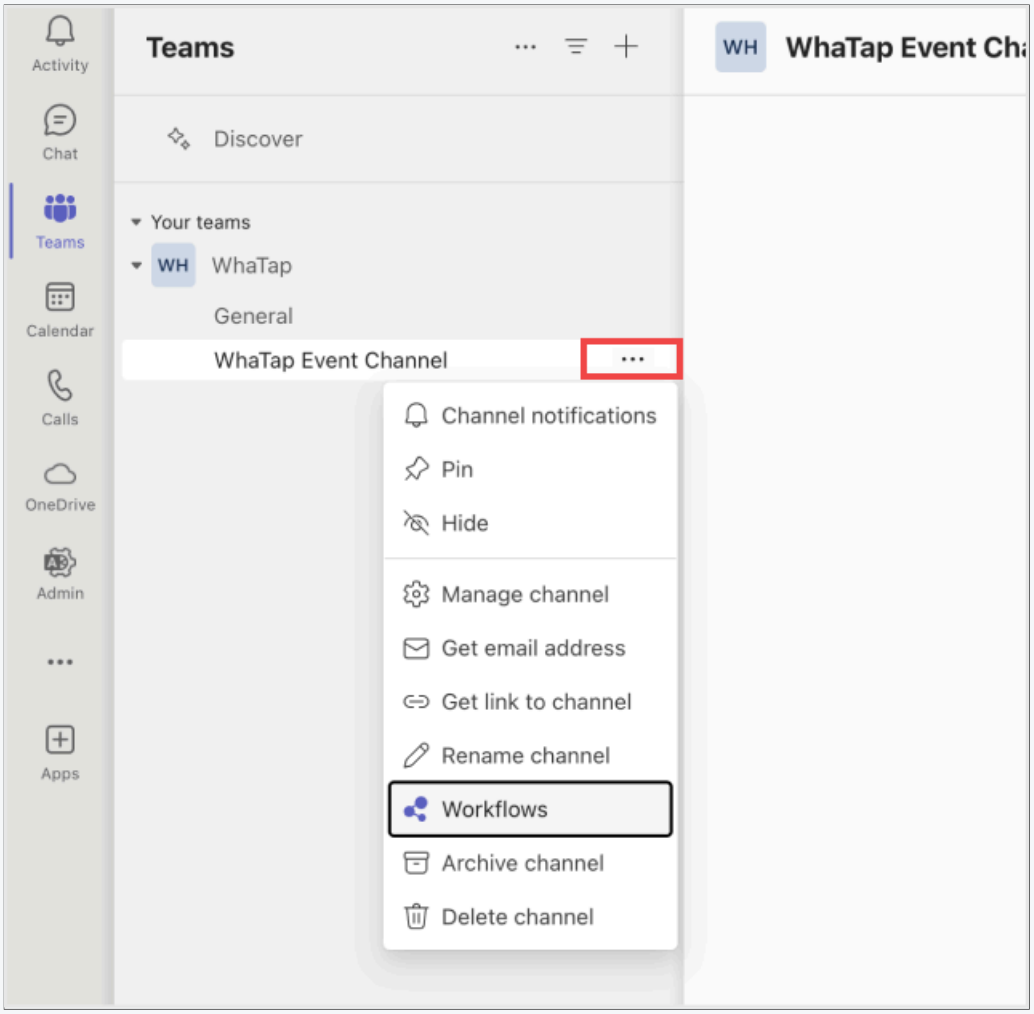



2. In the **Create a channel** window, enter the **Channel name** and **Description**. For **Channel type**, select **Standard**, then click the [ Create ] button.

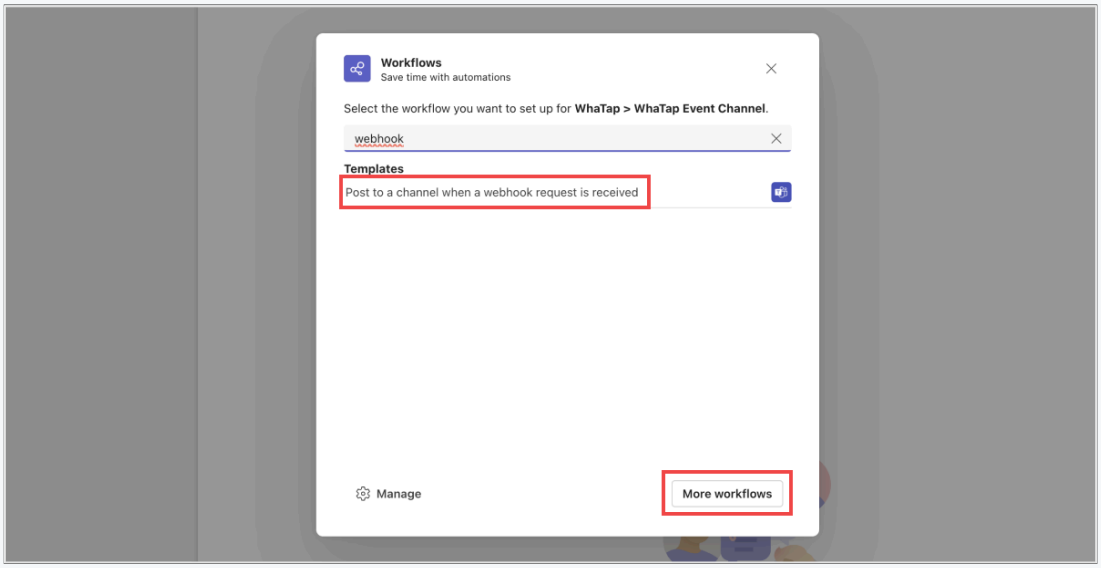



If you select **Shared** as the channel type, workflows cannot be used. If you select **Private**, workflows can be added but notifications cannot be received.

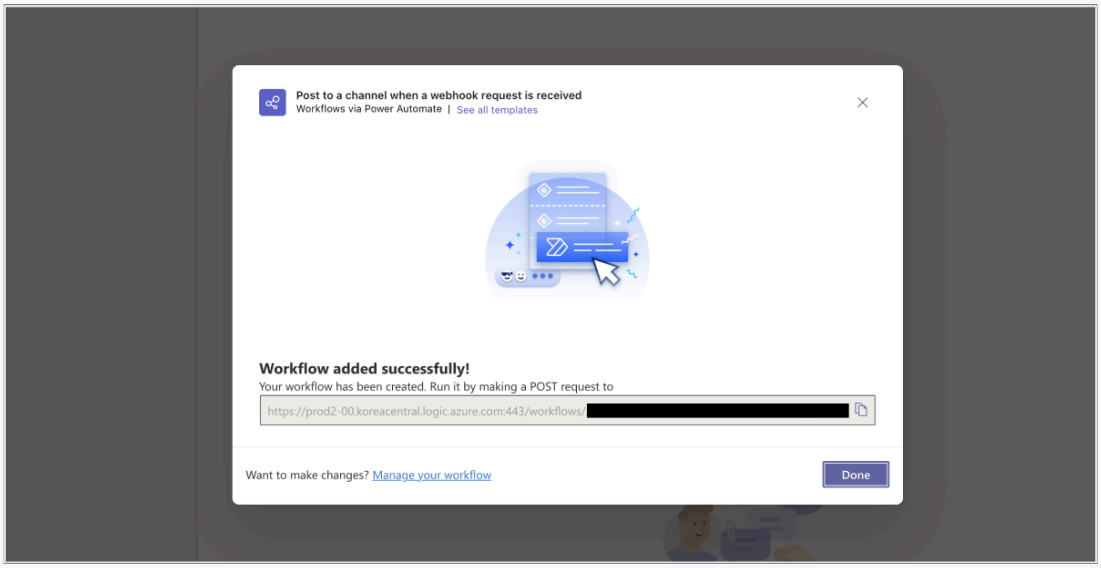
3. Click **More actions (...)** of the created channel, then select **Workflows**.




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.

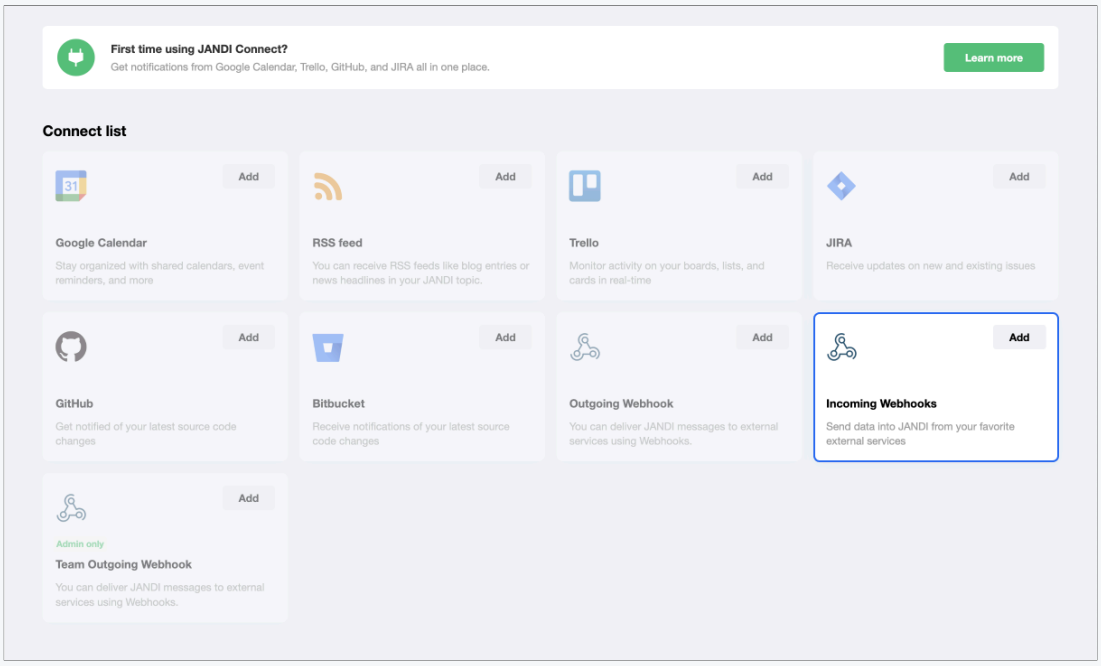



8. Return to the **Add a third-party plug-in** window in the WhaTap Monitoring service, and enter values for **Channel Name** and **Webhook URL** at the bottom of the **Teams Workflows** tab.

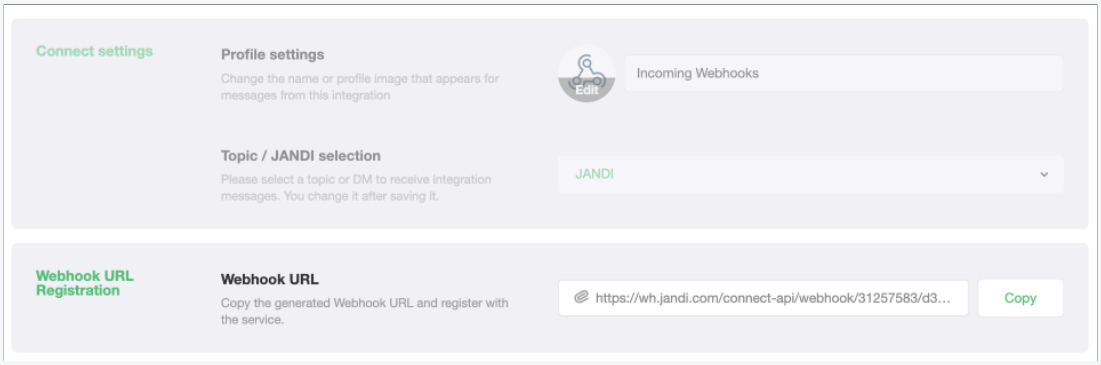
9. Click the [ **Registration** ] button.

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.



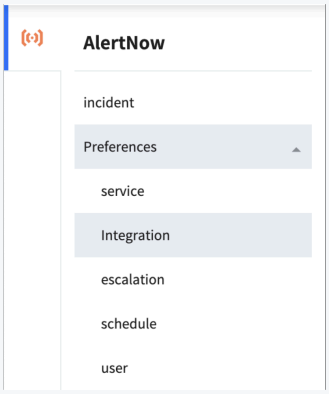

3. Click the [ Add Webhook Receiver ] button.
4. Copy the Webhook URL from the **Register Webhook URL** section.



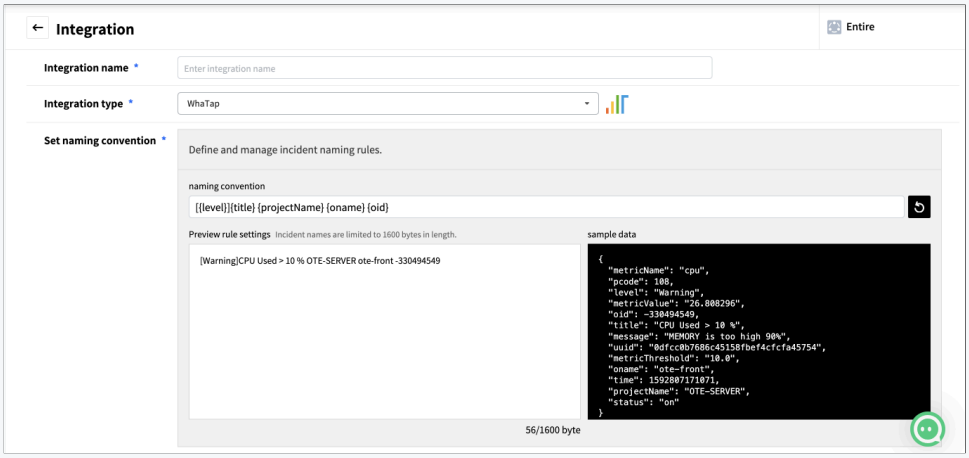

5. Return to the **Add a third-party plug-in** window in the WhaTap Monitoring service, and enter values for **Channel Name** and **Webhook URL** at the bottom of the **Jandi** tab.
6. Click the [ **Registration** ] button.

AlertNow

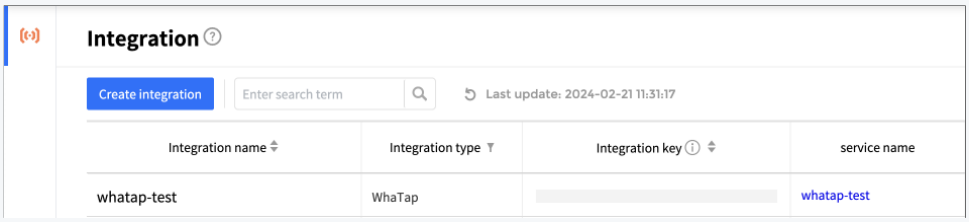
1. Access the **AlertNow** service, then go to **Settings** > **Integration**.




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` .

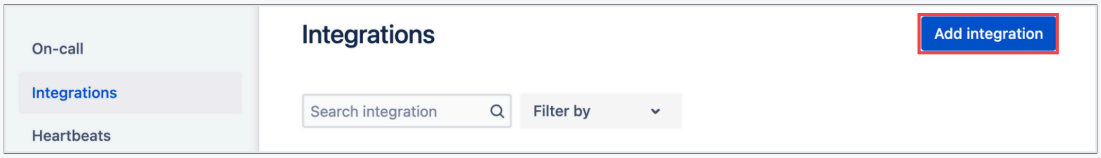
```
{
  "pcode": "${pcode}",
  "projectName": "${projectName}",
  "time": "${time}",
  "oid": "${oid}",
  "oname": "${oname}",
  "title": "${title}",
  "message": "${message}",
  "level": "${level}",
  "status": "${status}",
```

```
  "metricName": "${metricName}",
  "metricValue": "${metricValue}",
  "metricThreshold": "${metricThreshold}",
  "uuid": "${uuid}",
  "okind": "${okind}",
  "onode": "${onode}"
}
```

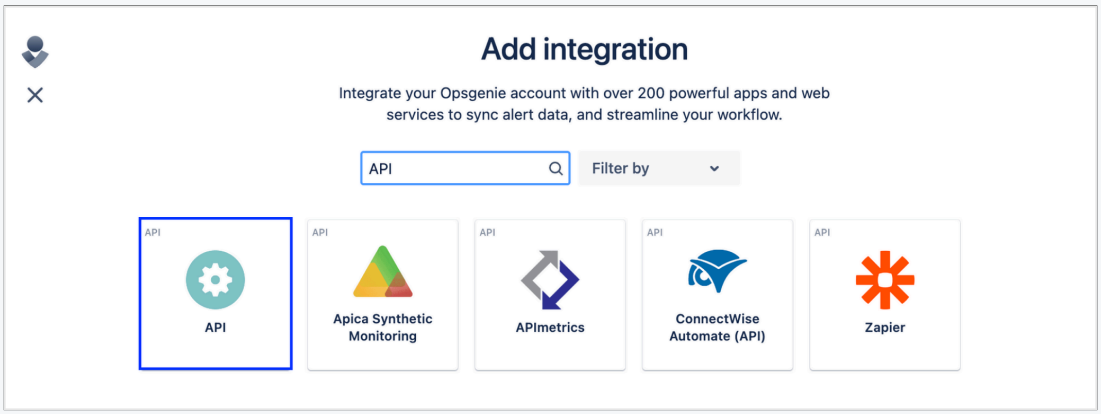
2. Enter values for the **Webhook Name** and **Webhook URL** fields.
3. To add a header value, click **+ Add** in **Add user header value**, then enter the key and value.
4. Click the [ **Registration** ] button.

Opsgenie

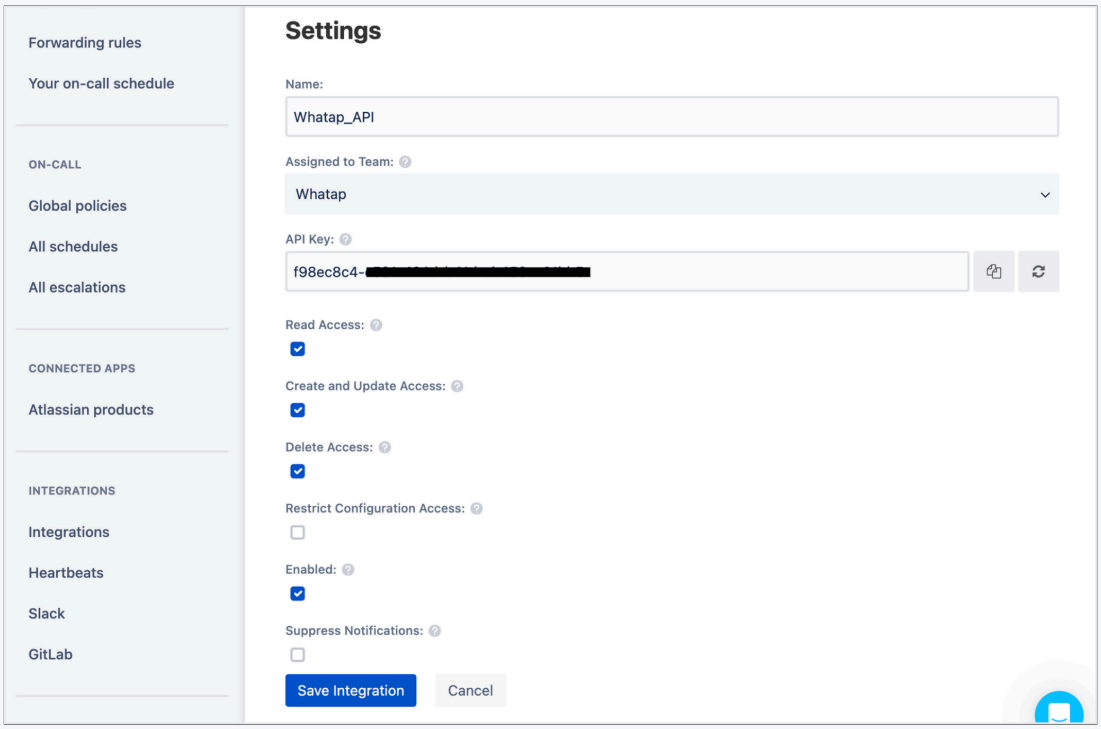
1. In the **Teams** menu, select the team that will receive notifications.
2. Go to the **Integration** menu and click the [ Add Integration ] button.




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 |

# 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<br>- Displayed as blank if the event is not associated with a specific agent |
| Agent Group | The agent type (OkindName) where the event occurred<br>- 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<br>- 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<br>- 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.

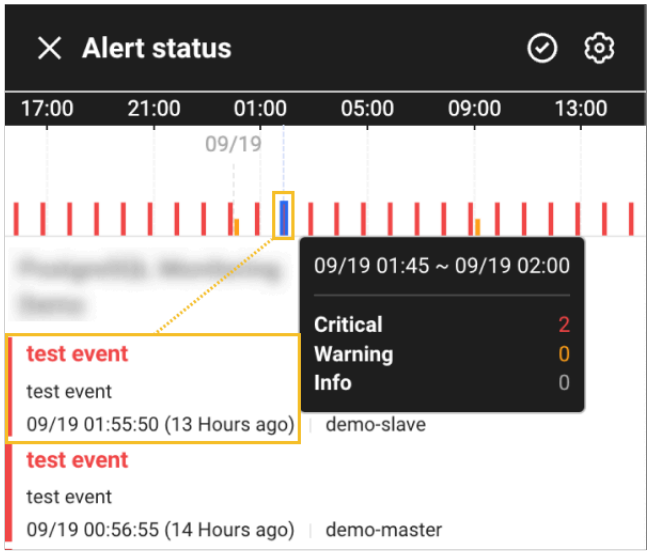
## Viewing Real-time Notifications

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

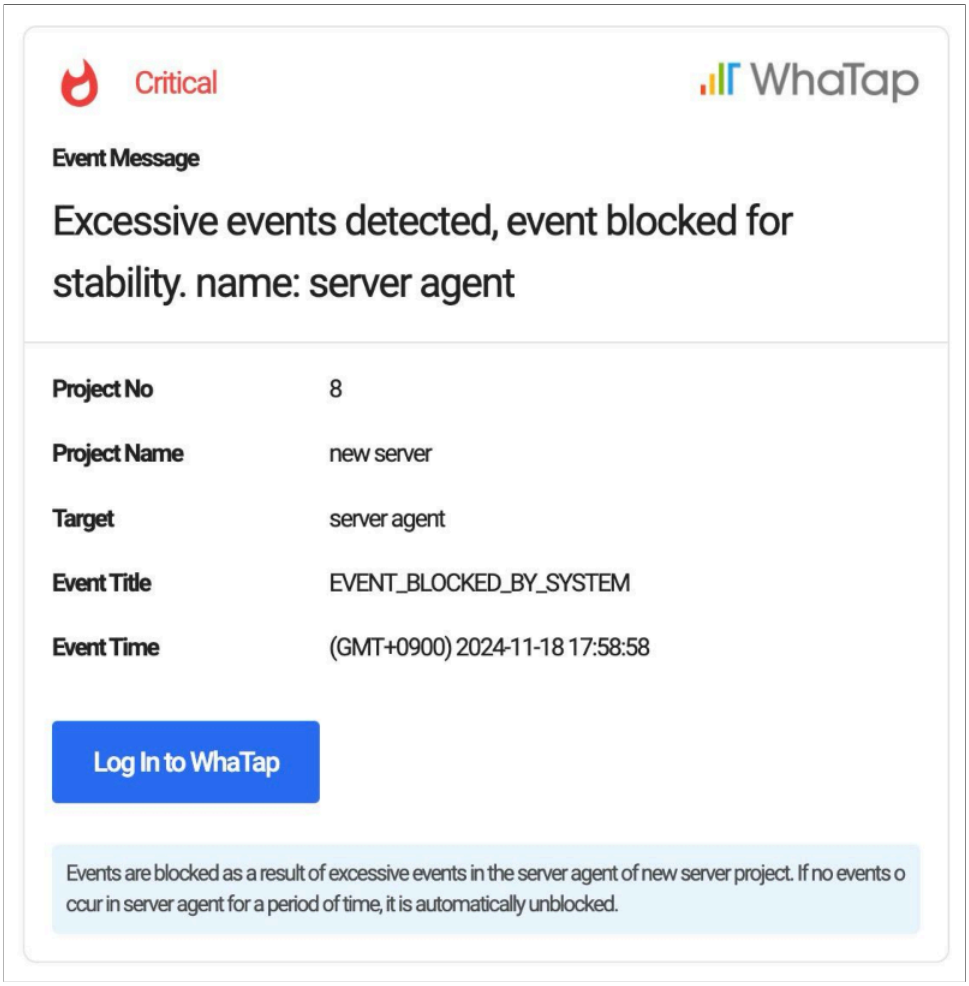
1. Click the icon at the top of the screen after selecting a project.
2. When the **Alert status** window appears, click the icon in the upper-right corner.
3. Configure each option in the real-time notification settings window.
   - **Popup notifications**: Displays notifications as pop-ups even when 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.

# 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**

**실험실**의 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

**MXQL**은 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.

## 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.