WhaTap

Kubernetes Monitoring

release date. 2026. 09. 10.

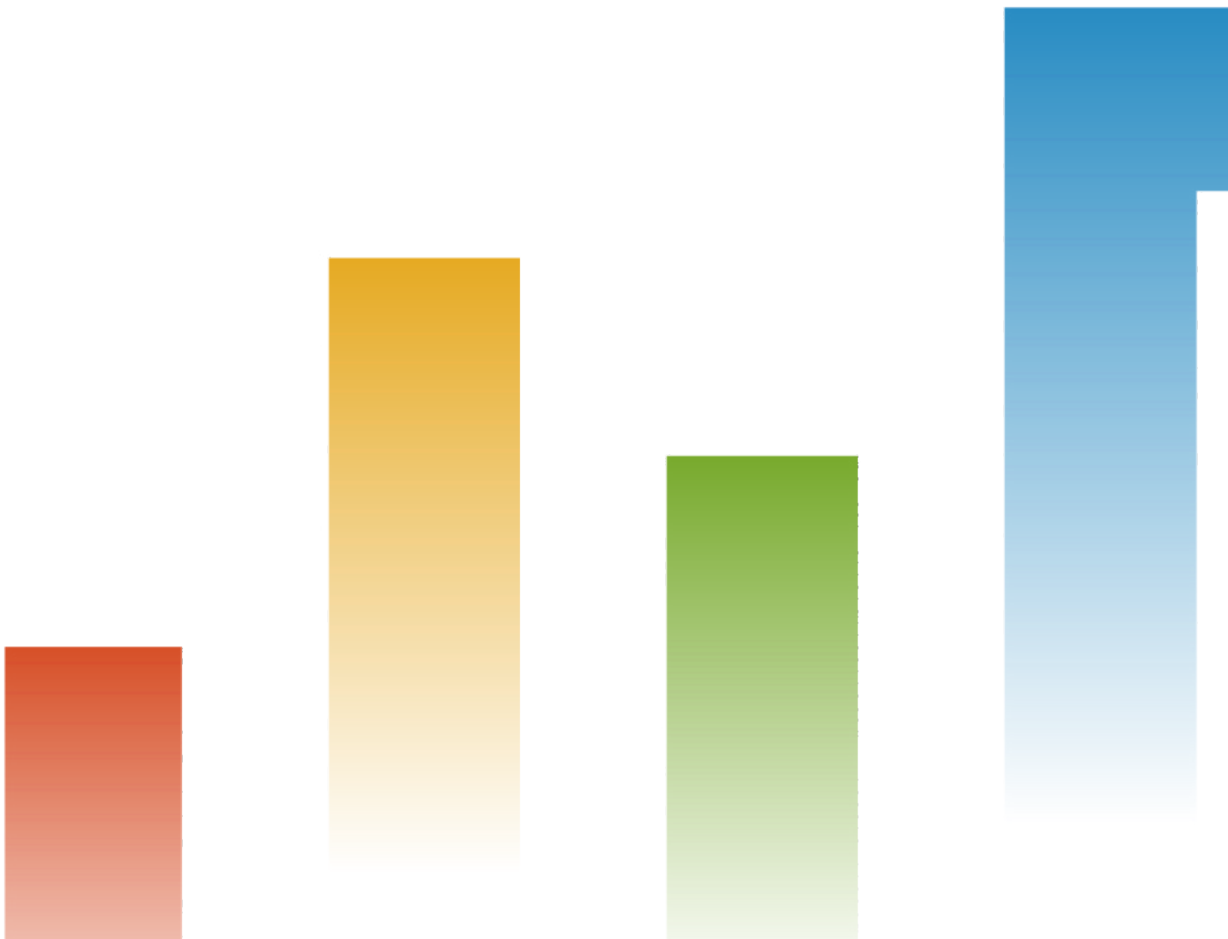


Table of Contents

- Kubernetes 모니터링 7
  - 시작하기 10
  - 지원 환경 13
  - 설치하기 19
    - 와탭 오퍼레이터 23
      - 와탭 오퍼레이터 설치 25
      - APM 자동 설치 구성 37
      - OpenAgent 구성 48
      - WhaTap CR 설정 옵션 72
    - 마스터 및 노드 에이전트 설치 80
    - 애플리케이션 에이전트 설치 96
      - Docker Java 설치 99
      - Docker Node.js 설치 112
      - Docker Python 설치 116
      - Docker PHP 설치 122
      - Docker Go 설치 126
    - 설치 문제 해결 136
  - 설정하기 142
  - 관리하기
    - 네임스페이스 관리 157
    - 연계 프로젝트 관리 160
    - 쿠버네티스 에이전트 163
    - 쿠버네티스 에이전트 목록 165
    - 에이전트 업데이트 168

- 에이전트 삭제 169

◦ 대시보드

- 컨테이너 맵 172
  - 컨테이너 뷰, Pod 뷰 186
  - 대상 그룹화 190
  - 컨테이너 맵 지표와 상태 197
  - 외부 프로젝트 연동과 설정 201
  - 컨테이너 맵 상세 분석 209
    - Target Information 213
    - Trace 216
    - Metrics 219
    - Log 224
    - Event 229
    - Call Information 236
- 노드 맵 242
- 퍼포먼스 요약 256
- Flex 보드 259
  - Flex 보드 사용하기 262
  - 위젯 관리하기 272

◦ OpenMetrics

- 템플릿 목록 277
- OpenMetrics 탐색기 280

◦ 클러스터

- 클러스터 요약 283
- 네임스페이스 현황 289

◦ 노드

- 노드 목록 291
- 노드 타임라인 303
- 노드 상세 312
- 노드 디스크 목록 314

◦ GPU 318

- GPU 대시보드 322
- GPU 트렌드 326
- GPU 서비스 그룹핑 328

◦ 컨테이너

- 컨테이너 목록 334
- 컨테이너 타임라인 339
- 컨테이너 볼륨 343
- 컨테이너 이미지 346
- Out Of Memory Killed 컨테이너 348

◦ 워크로드

- Pod 목록 351
- Pod 타임라인 357
- Pending Pod 현황 362
- 컨테이너 애플리케이션 목록 366
- Deployment 목록 372

◦ 서비스 네트워킹

- Service 목록 376
- Ingress 목록 381

◦ 스토리지

- PV/PVC 목록 385

◦ 컨트롤 플레인

- kube-apiserver 대시보드 392
- kube-apiserver 메트릭스 조회 400
- etcd 대시보드 409
- kube-scheduler 대시보드 416

◦ 분석

- 오브젝트 매니페스트 423
- 쿠버네티스 이벤트 430
- Pod 시작 분석 434
- Pod 시작 분석 438
- 애플리케이션 히트맵 443
- 멀티서비스 히트맵 445
- MSA 분석 447
- 애플리케이션 스택 455

◦ 메트릭스 456

- 쿠버네티스 메트릭 459
- GPU 메트릭 509
- 메트릭스 차트 512
- 메트릭스 차트 533
- 메트릭스 이상 탐지 536
- 메트릭스 조회 539

◦ 로그 541

◦ 보고서 546

◦ 통합 보고서 549

◦ 애플리케이션

- 애플리케이션 서비스 대시보드 557
- 히트맵 - 트레이스 분석 579

- 일자별 애플리케이션 현황 603
- 트랜잭션 검색 605
- 인스턴스 성능 모니터링
  - Java 0
  - Node.js 0
  - Python 0
  - PHP 0
  - Go 0

◦ 경고 알림 617

- 이벤트 설정 0
  - 메트릭스 623
  - 복합 메트릭스 644
  - 히트맵 패턴 651
  - 이상치 탐지 656
  - 실시간 로그 660
  - 복합 로그 673
  - 트랜잭션 687
- 이벤트 설정 698
- 이벤트 수신 설정 702
  - 이벤트 수신 포맷 725
- 이벤트 기록 및 모니터링 733
  - 실시간 알림 741
  - 시스템 이벤트 744

◦ 실험실 746

Kubernetes 모니터링

와탭 쿠버네티스 모니터링은 컨테이너, Pod, 노드, 클러스터 전체부터 컨테이너 안에서 실행되는 애플리케이션까지 한 화면에서 통합 분석할 수 있는 모니터링 솔루션입니다. 멀티 클라우드와 하이브리드 클라우드 환경을 모두 지원하며, 인프라 담당자와 애플리케이션 담당자가 동일한 데이터를 보며 협업할 수 있습니다.

쿠버네티스 모니터링 구조

쿠버네티스의 최소 배포 단위는 컨테이너가 아닌 Pod이며, 1개 Pod에는 1개 이상의 컨테이너가 포함됩니다. 따라서 쿠버네티스 모니터링은 컨테이너 관점과 Pod 관점이 모두 필요합니다. 와탭은 컨테이너와 Pod의 메트릭을 각각 별도 카테고리로 수집하고, **컨테이너 맵**에서 컨테이너 뷰와 Pod 뷰를 함께 제공합니다.

카테고리는 사용자 프로젝트로부터 수집한 메트릭을 구분하는 이름이며, 각 카테고리는 서로 다른 Tag와 Field로 구성됩니다.

주요 특장점

- **통합 모니터링** — 멀티 클라우드와 하이브리드 클라우드 환경의 쿠버네티스 시스템을 한 화면에서 모니터링합니다.
- **MSA 호출 관계 분석** — Pod 단위 분산 환경에서 트랜잭션의 API 호출 관계를 추적합니다.
- **Root Cause Analysis** — Host, 컨테이너, 애플리케이션을 함께 모니터링하여 장애 원인을 신속하게 분석할 수 있습니다.
- **컨테이너 내 애플리케이션 분석** — Java, Node.js, Python 애플리케이션의 히트맵, 트레이스, 멀티 트랜잭션 추적을 지원합니다.
- **컨테이너 맵으로 통합 가시성 확보** — 쿠버네티스 리소스, 메트릭, 이벤트, 로그를 한 화면에서 제공합니다. 인프라와 애플리케이션 담당자가 동일한 뷰를 보며 협업할 수 있습니다.
- **리소스 사용량 추이 분석** — 컨테이너와 Node의 사용량 추이 대시보드를 제공합니다. 와탭 프로젝트에 속한 컨테이너/Node 전체 목록부터 개별 대상의 세부 상태까지 확인할 수 있습니다.
- **메타 정보 즉시 조회** — 마스터 노드에 접속하지 않아도 컨테이너 맵에서 메타 정보를 바로 확인할 수 있습니다. 별도의 마스터 메타 대시보드에서 전체 구성 요소의 메타 정보를 한눈에 볼 수 있습니다.
- **로그 분석 집중 지원** — 컨테이너 로그 파일과 표준 출력을 실시간으로 저장하고, 분류·인덱싱을 자동화합니다. 컨테이너에 직접 접속하지 않아도 의미 있는 로그에 집중할 수 있습니다.

다음 튜토리얼 영상을 통해 와탭 쿠버네티스 모니터링을 경험해 보세요.

모니터링 대상

쿠버네티스 모니터링 대상은 쿠버네티스 클러스터와 그 위에서 실행되는 애플리케이션입니다.

- **쿠버네티스 클러스터**
  - 컨테이너의 CPU, Memory, Disk(Volume), Network
  - Namespace, Deployment, ReplicaSet, Pod 등 클러스터 구성 요소
  - Node(호스트)의 CPU, Memory, Disk, Network
- **컨테이너 내 애플리케이션** — Java, Node.js, Python, PHP, Go

이어서 설명하는 모니터링 절차는 클러스터 모니터링과 애플리케이션 모니터링으로 구분합니다.

모니터링 절차

클러스터 모니터링 절차

1. 와탭 모니터링 서비스에서 **클러스터 프로젝트**를 생성하세요. 처음 만든 쿠버네티스 프로젝트가 자동으로 클러스터 프로젝트가 됩니다. 자세한 내용은 시작하기를 참조하세요.
2. 쿠버네티스 환경에 마스터 에이전트(`whatap-master-agent`)와 노드 에이전트(`whatap-node-agent`)를 설치하세요. 두 에이전트는 사용자의 클러스터 내에서 Pod 형태로 실행됩니다. 자세한 내용은 에이전트 설치를 참조하세요.
3. (선택) 클러스터에 네임스페이스가 많다면 네임스페이스 단위로 프로젝트를 분리할 수 있습니다. 자세한 내용은 네임스페이스 프로젝트 생성을 참조하세요.

> 쿠버네티스 Namespace는 단일 클러스터 내에서 리소스 그룹을 격리하는 메커니즘입니다. 클러스터의 Namespace 목록은 `kubectl get namespace` 명령으로 확인할 수 있습니다.

애플리케이션 모니터링 절차

컨테이너 내에서 실행되는 애플리케이션을 모니터링하려면 애플리케이션 에이전트를 추가로 설치합니다. 지원 언어는 Java,

Node.js, Python, PHP, Go입니다. 자세한 내용은 애플리케이션 에이전트 설치를 참조하세요.

시작하기

와탭 모니터링은 다음 절차로 시작합니다.

1단계. 회원 가입하기
2단계. 제품 선택하기
3단계. 프로젝트 생성하기
4단계. 액세스 키 확인하기

이후 지원 환경을 확인한 후 에이전트 설치 단계로 이동합니다.

회원 가입하기

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

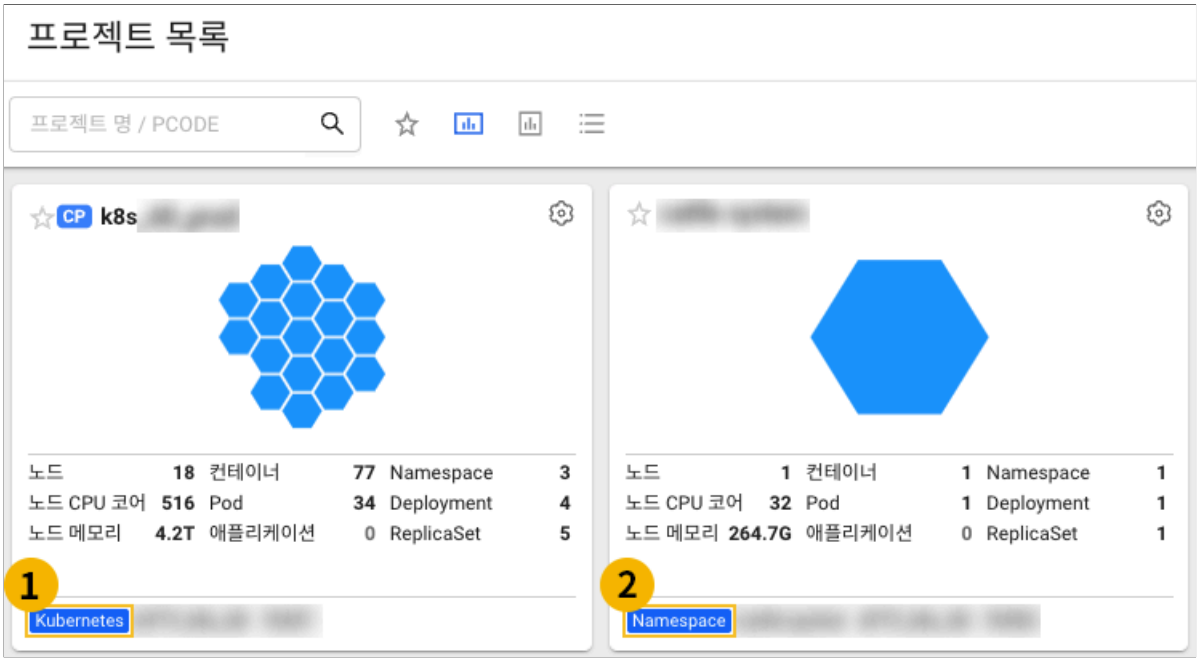
제품 선택하기

1. 와탭 모니터링 서비스에 로그인합니다.
2. **전체 프로젝트** > ＋ **프로젝트** 버튼을 클릭합니다.
3. ❶ **상품 선택** 화면에서 모니터링할 제품의 **생성하기** 버튼을 클릭하세요.

프로젝트 생성하기

쿠버네티스 프로젝트 구조 이해하기

쿠버네티스 모니터링을 시작하려면 먼저 **클러스터 프로젝트**를 만들어야 합니다. 클러스터에 네임스페이스가 많고 컨테이너/파드(Pod) 수가 많다면, **네임스페이스 프로젝트**를 추가로 만들어 모니터링 범위를 분리할 수 있습니다.



1 클러스터 프로젝트

- 쿠버네티스 클러스터 전체 자원을 모니터링하는 프로젝트입니다.
- 마스터 에이전트와 노드 에이전트는 **클러스터 프로젝트**에 설치합니다.
- 프로젝트 목록에서 **클러스터 프로젝트**는 이름 옆에 `CP` 태그, 아래쪽에 `Kubernetes` 태그를 함께 표시합니다.
- 자세한 생성 방법은 시작하기 문서를 참조하세요.

2 네임스페이스 프로젝트

- 쿠버네티스 클러스터 하위 네임스페이스(Namespace) 단위로 자원과 애플리케이션 컨테이너를 모니터링하는 프로젝트입니다.
- 프로젝트 목록에서 **네임스페이스 프로젝트**는 아래쪽에 `Namespace` 태그를 표시합니다.
- 자세한 관리 방법은 네임스페이스 프로젝트 관리하기 문서를 참조하세요.

쿠버네티스 프로젝트 생성하기

모니터링할 제품을 선택했다면, 에이전트를 설치하기 전에 먼저 프로젝트를 생성합니다.

1. ❷ **프로젝트 생성** 화면에서 프로젝트를 생성하려면 다음 항목을 입력하거나 선택하세요.
   - **프로젝트 이름**(필수): 프로젝트의 이름을 입력합니다.
   - **데이터 서버 지역**(필수): 데이터 서버가 위치한 리전(지역)을 선택합니다. 리전은 클라우드 서비스를 제공하는 데이터 센터의 묶음입니다. 선택한 리전에 사용자의 데이터가 저장됩니다.
   - **타임 존**(필수): 알림 및 보고서 생성 시 기준이 되는 시간을 설정합니다.
   - **알림 언어 설정**(필수): 프로젝트에서 발생하는 경고 알림 메시지의 언어를 설정합니다. (한글, 영어 지원)
   - **프로젝트 그룹**: 여러 프로젝트를 그룹으로 묶어 관리할 수 있습니다. 소속될 그룹이 있으면 선택하세요. 없다면 그룹 추가 버튼을 클릭해 그룹을 추가할 수 있습니다.
   - **프로젝트 설명**: 프로젝트에 대한 추가 설명을 입력합니다.
2. 모든 설정을 완료하면 **프로젝트 생성하기** 버튼을 클릭하세요.

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

액세스 키 확인

액세스 키는 와탭 서비스 활성화를 위한 고유 ID입니다.

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

> ⚠ **액세스 키**는 다른 프로젝트에 중복 사용할 수 없으며 **네임스페이스 프로젝트**에도 별도 **액세스 키**가 필요합니다.

> ⓘ 프로젝트를 생성한 후, 자동으로 **에이전트 설치** 페이지로 이동합니다. **에이전트 설치** 페이지로 이동하지 않는다면 화면 왼쪽 메뉴에서 **관리** > **에이전트 설치**를 선택하세요.

지원 환경

쿠버네티스 모니터링을 시작하기 전에 다음 지원 사항을 확인하세요. 와탭 쿠버네티스는 표준 쿠버네티스 1.16 이상 버전과 해당 버전에 대응하는 환경에서 설치할 수 있습니다.

지원 플랫폼

> ⓘ 최종 업데이트: 2025-11-18

플랫폼	지원 여부
Amazon EKS (Elastic Kubernetes Service)	✅
AKS (Azure Kubernetes Service)	✅
GKE (Google Kubernetes Engine)	✅
OKE (Oracle Container Engine for Kubernetes)	✅
Naver Cloud Kubernetes Service	✅
NHN Kubernetes Service (NKS)	✅
Red Hat OpenShift (3.x / 4.x)	✅
Cocktail	✅
Accordion	✅
PaaS-TA	✅
Kakao i Cloud	✅
Rancher	✅

> ⓘ 와탭은 CNCF(Cloud Native Computing Foundation) 인증을 받은 표준 쿠버네티스(Standard Kubernetes) 환경이라면 어디든 설치할 수 있습니다.

NVIDIA GPU

와탭 쿠버네티스 노드 에이전트는 NVIDIA의 데이터 센터 GPU 관리자(DCGM) Exporter를 통해 GPU 메트릭을 수집합니다. 수집 구성은 사이드카(Sidecar) 패턴을 사용합니다.

- **사이드카 패턴**

  DCGM Exporter는 주 애플리케이션 컨테이너와 함께 동일한 Pod 내에서 실행되는 보조 컨테이너로 설정됩니다. 이 사이드카 패턴은 DCGM Exporter가 GPU 상태 정보를 효율적으로 수집할 수 있도록 돕습니다.

- **DCGM Exporter**

  `dcgm-exporter` 컨테이너는 DCGM을 통해 GPU의 상태와 성능 관련 메트릭을 수집합니다.

- **메트릭 수집 및 전송**

  `whatap-node-agent` 컨테이너는 `dcgm-exporter`의 HTTP 엔드포인트를 통해 GPU 메트릭을 요청하고 이를 수집합니다.

  > ⓘ `dcgm-exporter`의 HTTP 엔드포인트는 보통 포트 9400을 사용합니다.

- **MIG 환경 지원**

  와탭 쿠버네티스는 NVIDIA GPU 클러스터 환경에서 **MIG(Multi-Instance GPU)** 기능을 지원하여, **물리 GPU 단위뿐만 아니라 MIG 인스턴스까지 세분화된 모니터링**을 제공합니다.

  MIG 모드를 사용하면 하나의 물리 GPU를 여러 논리 인스턴스로 분할해 워크로드를 격리하고 자원을 할당할 수 있습니다. 와탭은 이러한 환경에서도 정확한 메트릭 수집과 시각화를 지원합니다.

  > ⓘ GPU 모니터링을 위해서 해당 기능을 활성화해야 합니다. 자세한 내용은 GPU 모니터링 설치를 참고하세요.

수집되는 GPU 메트릭 목록은 GPU 메트릭 문서를 참조하세요.

브라우저

와탭 모니터링 서비스는 웹브라우저와 모바일 앱에서 이용할 수 있습니다.

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

지원 안내

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

제약 사항

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

서비스 제공 형태

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

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

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

제공 형태에 따른 기능 차이

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

> ⓘ 온프레미스에서 일부 기능을 사용하려면 별도 계약이 필요합니다.

방화벽

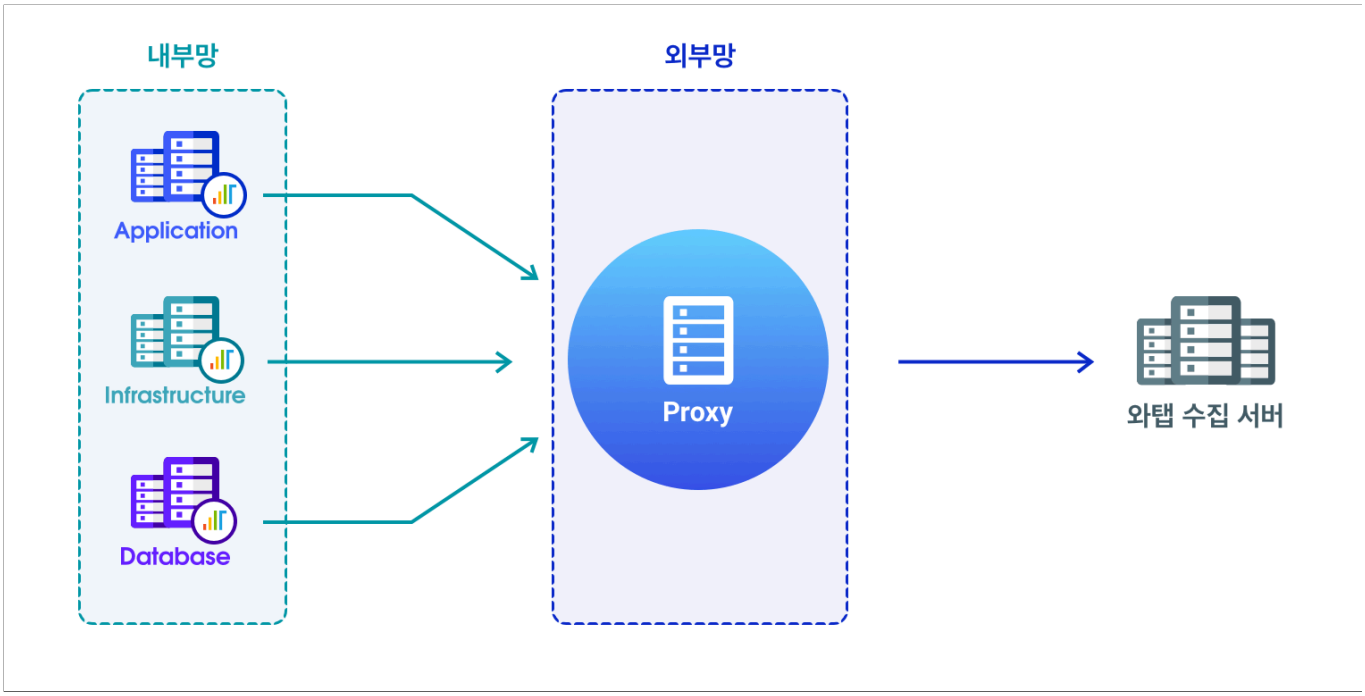
와탭 에이전트는 수집 서버 **TCP 6600** 포트로 접근할 수 있어야 합니다. 모니터링 대상과 가까운 수집 서버 주소를 허용하세요.

출발지: 와탭 에이전트

클라우드	목적지	목적지 IP	포트
AWS	와탭 서울 수집 서버	13.124.11.223 / 13.209.172.35	TCP 6600
	와탭 도쿄 수집 서버	52.68.36.166 / 52.193.60.176	TCP 6600
	와탭 싱가포르 수집 서버	18.138.0.93 / 18.139.67.236	TCP 6600
	와탭 자카르타 수집 서버	108.136.91.69 / 108.137.158.44	TCP 6600
	와탭 태국 수집 서버	43.209.109.219 / 43.209.48.86	TCP 6600
	와탭 뭄바이 수집 서버	13.127.125.69 / 13.235.15.118	TCP 6600

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

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



모바일 앱

와탭 모바일 앱은 안드로이드와 iOS 환경을 지원합니다. 링크로 이동하거나 QR 코드를 스캔해 와탭 앱을 설치할 수 있습니다.

모바일 앱에 대한 자세한 설명은 다음 문서를 참고하세요.

iOS	Android
iOS 12 버전 이상	Android 5.0 버전 이상

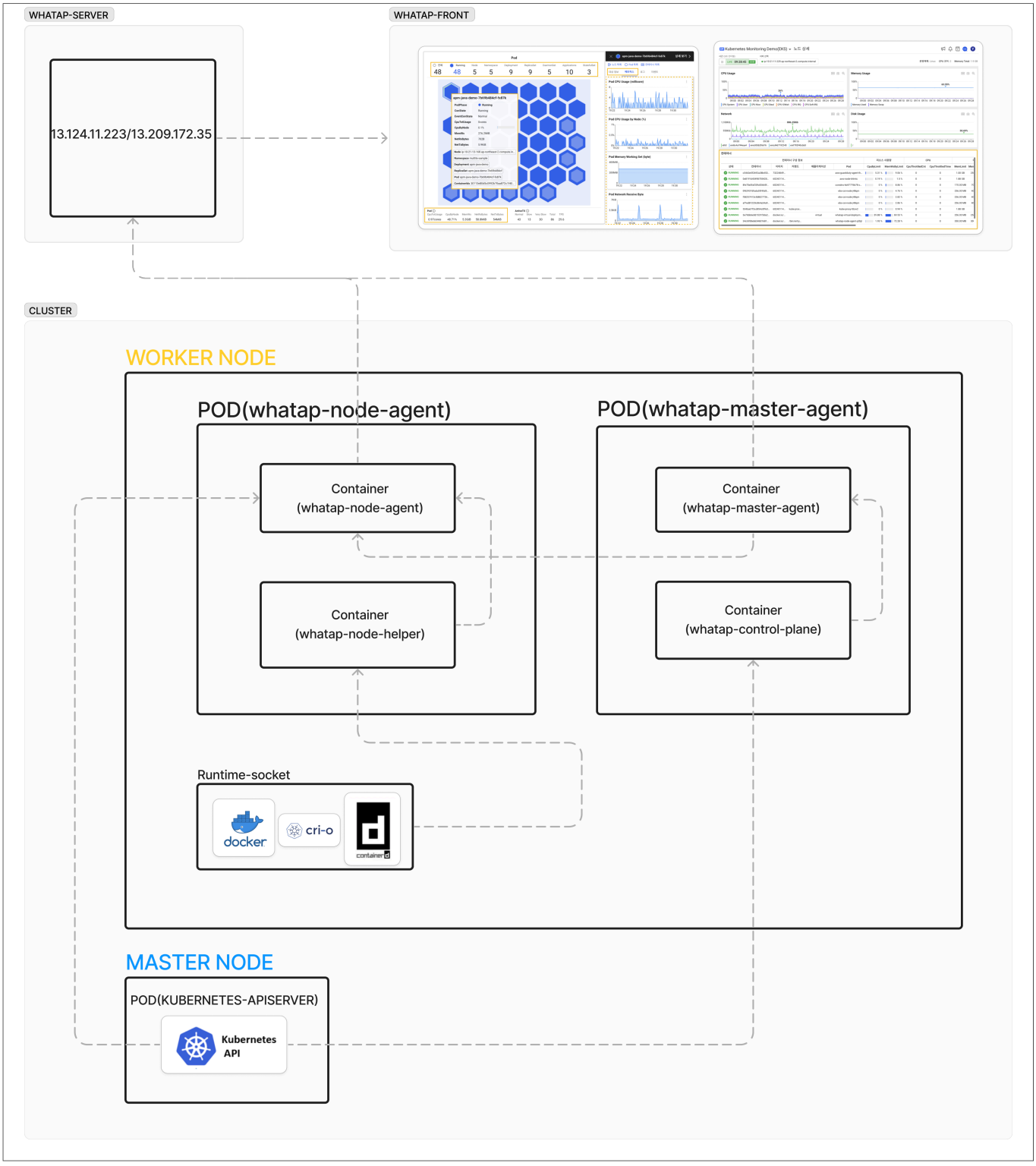
에이전트 설치

와탭 쿠버네티스 모니터링 에이전트의 구조와 두 가지 설치 방식을 소개합니다. 사전 준비와 아키텍처를 먼저 확인한 후, 운영 환경에 맞는 방식으로 설치를 진행하세요.

> ⓘ 에이전트를 설치하기 전에 시작하기 문서에서 다음 단계를 먼저 완료하세요.
>
> - 와탭 회원 가입
> - **클러스터 프로젝트** 생성
> - 액세스 키 발급

쿠버네티스 에이전트 아키텍처

와탭 쿠버네티스 에이전트는 마스터 에이전트 파드와 노드 에이전트 파드로 구성됩니다.


WHATAP-SERVER
13.124.11.223/13.209.172.35
WHATAP-FRONT
CLUSTER
WORKER NODE
POD(whatap-node-agent)
Container
(whatap-node-agent)
Container
(whatap-node-helper)
POD(whatap-master-agent)
Container
(whatap-master-agent)
Container
(whatap-control-plane)
Runtime-socket
docker
cri-o
MASTER NODE
POD(KUBERNETES-APISERVER)
Kubernetes
API

노드 에이전트 파드

`whatap-node-agent` 파드는 DaemonSet 형태로 배포되며, 노드당 1개씩 실행됩니다. 파드 내부에는 다음 두 컨테이너가 있습니다.

- **whatap-node-agent**

  kube-apiserver와 통신하여 노드 워크로드의 메트릭을 수집합니다. `kube_pod`, `container`, `kube_pod_volume` 등의 메트릭에 영향을 줍니다.

- **whatap-node-helper**

  Runtime-socket과 루트 볼륨을 마운트하여 노드와 컨테이너의 자원 사용량을 수집합니다.

마스터 에이전트 파드

`whatap-master-agent` 파드는 Deployment 형태로 배포되며, 클러스터당 1개가 실행됩니다. 파드에는 다음 컨테이너가 포함됩니다.

- **whatap-master-agent** (필수)

  클러스터 전반의 메트릭을 수집합니다. 네임스페이스 프로젝트 정보처럼 모든 노드 에이전트가 공유해야 하는 데이터를 각 노드 에이전트로 전달하고 동기화합니다.

- **whatap-control-plane** (선택)

  쿠버네티스 컨트롤 플레인 노드의 지표를 수집합니다. 현재는 apiserver의 상태와 요청 횟수를 수집합니다. 컨트롤 플레인 모니터링을 사용하려면 추가 기능 적용이 필요합니다. 자세한 내용은 컨트롤 플레인 대시보드 문서를 참조하세요.

> ⓘ 향후 여러 개의 마스터 에이전트 파드가 동시에 동작할 수 있도록 다중화 기능을 제공할 예정입니다.

설치 방식 선택하기

와탭 쿠버네티스 모니터링은 다음 두 가지 방식으로 에이전트를 설치할 수 있습니다. 운영 환경과 관리 방식에 맞는 방법을 선택하세요.

와탭 오퍼레이터로 설치하기

와탭 오퍼레이터는 단일 커스텀 리소스(CR)로 마스터·노드 에이전트, 애플리케이션 자동 계측, OpenAgent를 통합 관리하는 쿠버네티스 오퍼레이터입니다. CR 상태를 지속적으로 감시하여 변경 사항을 자동으로 조정합니다.

다음 페이지에서 설치와 구성 방법을 확인하세요.

- 와탭 오퍼레이터 설치 — Helm 차트로 오퍼레이터를 배포하고 `WhatapAgent` CR을 생성합니다.
- 애플리케이션 에이전트 자동 주입 — CR 설정만으로 컨테이너 내 APM 에이전트를 자동 주입합니다.
- Open Agent로 OpenMetrics 수집 — 외부 익스포터의 OpenMetrics 데이터를 수집합니다.
- WhatapAgent CR 구성 옵션 — CR에서 사용할 수 있는 모든 설정 옵션을 정리했습니다.

마스터·노드 에이전트 직접 설치하기

콘솔의 **설치 안내** 페이지를 따라 YAML 또는 Helm 차트로 마스터·노드 에이전트를 직접 배포하는 방식입니다. 컨테이너 런타임, 쿠버네티스 버전, GKE/OpenShift 등 환경별 옵션을 단계별로 선택할 수 있습니다.

- 마스터 및 노드 에이전트 설치

애플리케이션 에이전트 추가 설치

앞서 소개한 두 방식 중 어떤 것을 선택하더라도, 컨테이너 안에서 실행되는 Java/Node.js/Python/PHP/Go 애플리케이션을 모니터링하려면 애플리케이션 에이전트를 추가로 설치해야 합니다.

- 애플리케이션 에이전트 설치

설치 영상 보기

와탭 오퍼레이터

와탭 오퍼레이터는 쿠버네티스 환경에서 와탭 에이전트를 배포하고 구성하는 오픈소스 쿠버네티스 오퍼레이터입니다.

오퍼레이터를 활용하면 단일 커스텀 리소스(CRD)로 쿠버네티스 및 GPU(MIG) 모니터링, 자동 계측, **OpenMetrics** 수집까지 통합 관리합니다. 배포 리소스의 유효성을 자동 검증하여 복잡한 구성에서 발생할 수 있는 오류를 최소화하고 안정적인 운영을 지원합니다.

주요 특징

- **커스텀 리소스 기반 배포**: 와탭 에이전트와 관련 구성 요소를 쿠버네티스 커스텀 리소스로 배포하고 관리합니다.
- **간소화된 배포 구성**: 와탭 에이전트의 버전, 리소스 요청/제한, 모니터링 대상 등록 등 필수 요소를 쿠버네티스 네이티브 리소스로 간편하게 설정합니다.
- **자동 APM 계측**: 쿠버네티스 표준 라벨 선택자로 특정 파드에 APM 에이전트를 자동 주입합니다.
- **Open Agent 기반 OpenMetrics 수집**: Open Agent 설치로 **OpenMetrics** 데이터를 수집하고 활용합니다.
- **통합 모니터링 관리**: 단일 CR로 애플리케이션 성능 모니터링(APM)과 쿠버네티스 인프라 모니터링을 한 번에 구성하고 관리하여 운영 효율성을 높입니다.

와탭 오퍼레이터의 목표

와탭 오퍼레이터는 쿠버네티스 환경에서 모니터링 구성의 복잡성을 줄이고 관리를 단순화하는 데 중점을 둡니다.

- **에이전트 설치 및 구성 부담 감소**: 와탭 모니터링 에이전트의 설치와 관리 부담을 줄입니다.
- **쿠버네티스 네이티브 리소스를 통한 자동화**: 쿠버네티스 CRD를 활용하여 와탭 모니터링 대상에 대한 설정을 자동으로 관리합니다.

```
apiVersion: monitoring.whatap.com/v2alpha1
kind: WhatapAgent
metadata:
  name: whatap
spec:
  features:
```

```
apm:
  instrumentation:
    targets:
      - name: "java-app"
        enabled: true
        language: "java"
        whatapApmVersions:
          java: "2.2.58"
        # 선택자 구성...
```

- **구성 추상화 및 유효성 검증**: 복잡한 모니터링 구성을 단순화하며, 표준 쿠버네티스 선택자(`matchLabels` , `matchExpressions`)를 지원합니다. 구성 유효성 검증으로 오류를 최소화하여 안정적으로 운영할 수 있습니다.

왜 헬름 차트나 수동 데몬셋 설치 대신 와탭 오퍼레이터를 사용해야 할까요?

헬름 차트나 데몬셋으로도 와탭 에이전트를 설치할 수 있지만, 와탭 오퍼레이터는 다음 장점을 제공합니다.

1. 자동 상태 조정

헬름 차트나 데몬셋은 상태 변경 시 수동 개입이 필요하지만, 와탭 오퍼레이터는 쿠버네티스 **조정 루프(reconciliation loop)** 에 포함되어 CR의 상태를 지속적으로 감시하고 자동 조정합니다. 예를 들어 CR로 생성된 리소스가 실수로 삭제되거나 변경되어도 오퍼레이터가 이를 감지하여 CR에 정의된 상태로 복구합니다.

2. 구성 오류 최소화

수동 설치 방식은 오류 발생 가능성이 높습니다. 와탭 오퍼레이터는 에이전트 구성에 대한 철저한 유효성 검증을 수행하여 이러한 오류를 최소화합니다.

3. 통합 관리

K8s 클러스터 내의 와탭 모니터링 컴포넌트 구성을 단일 파일로 통합 관리할 수 있어 운영 효율성이 향상됩니다.

4. 쿠버네티스 표준 준수

오퍼레이터는 쿠버네티스 API의 일급 리소스로 취급되며, 쿠버네티스 표준 라벨 선택자를 완벽하게 지원합니다. 이를 통해 익숙한 쿠버네티스 패턴을 활용하여 모니터링 대상을 유연하게 선택할 수 있습니다.

와탭 오퍼레이터 설치

이 가이드는 쿠버네티스 환경에 와탭 오퍼레이터를 설치하고, `WhatapAgent` CR을 설정하여 와탭 모니터링을 시작하는 방법을 설명합니다.

설치 전 확인 사항

- 쿠버네티스 클러스터 (v1.19+)
- Helm 3.2 이상
- 와탭 계정 및 라이선스 키

> - 와탭 에이전트는 **whatap-monitoring** 네임스페이스에 설치됩니다. 에이전트가 설치되는 네임스페이스는 변경할 수 없습니다.
> - 수집 서버로 데이터를 전송하기 위해 수집 서버 주소와 포트가 열려 있어야 합니다(TCP Outbound). 수집 서버 주소와 포트는 와탭 모니터링 서비스의 **관리 > 에이전트 설치** 화면에서 확인할 수 있습니다.

설치 방법

다음 단계에 따라 와탭 오퍼레이터를 설치하고 `WhatapAgent` CR을 생성합니다.

> 와탭 모니터링 서비스의 **관리 > 에이전트 설치 > 쿠버네티스** 화면에서 설치 방식으로 **Operator**를 선택하면 아래 단계의 실행 명령어와 `WhatapAgent` CR 파일(whatap_operator.yaml)을 자동으로 생성해 제공합니다. 화면에서 이미지 버전, 컨테이너 런타임, 환경(GKE, OpenShift)과 추가 기능(GPU 모니터링, APM 자동 설치, 오픈 메트릭 수집)을 선택하면 해당 설정이 반영된 파일을 다운로드할 수 있습니다. 이 문서는 파일을 직접 작성하거나 다운로드한 파일을 수정할 때 참고하세요.

1. **와탭 operator를 설치합니다.**
    - 에이전트를 처음 설치하는 사용자는 다음 명령어를 실행합니다.

```
helm repo add whatap https://whatap.github.io/helm/
helm repo update
```

```
kubectl create ns whatap-monitoring
export WHATAP_HOST=<수집서버 IP>
export WHATAP_LICENSE=<와탭 라이센스>
export WHATAP_PORT=<와탭 포트>
kubectl create secret generic whatap-credentials --namespace whatap-monitoring --from-literal
WHATAP_LICENSE=$WHATAP_LICENSE --from-literal WHATAP_HOST=$WHATAP_HOST --from-literal
WHATAP_PORT=$WHATAP_PORT
helm install whatap-operator whatap/whatap-operator --namespace whatap-monitoring
```

고급 설정

이미지 레지스트리 및 Pull Secret

프라이빗 레지스트리를 사용하거나 이미지 태그/풀 정책을 조정해야 하는 경우 values.yaml에서 다음 값을 설정합니다.

```
image:
repository: public.ecr.aws/whatap/whatap-operator
tag: latest
pullPolicy: Always

# 프라이빗 레지스트리 사용 시 설정
imagePullSecret:
name: <your-registry-secret-name>
```

스케줄링/우선순위 옵션

오퍼레이터 파드의 스케줄링 제약과 우선순위를 제어할 수 있습니다. 값을 지정한 경우만 템플릿에 렌더링 되어 기존 사용자와의 호환성이 유지됩니다.

```
priorityClassName: ""   # 예: system-cluster-critical
nodeName: ""            # 특정 노드 고정 스케줄링이 필요한 경우
```

```
nodeSelector: {}       # 예: { kubernetes.io/os: linux }
affinity: {}           # 표준 K8s Affinity 스키마
tolerations: []        # 표준 K8s Toleration 리스트
```

- **예시**

```
priorityClassName: "high-priority"
nodeSelector:
 nodepool: system

affinity:
 nodeAffinity:
    requiredDuringSchedulingIgnoredDuringExecution:
    nodeSelectorTerms:
     - matchExpressions:
        - key: kubernetes.io/os
          operator: In
          values: ["linux"]

tolerations:
 - key: "node-role.kubernetes.io/control-plane"
   effect: NoSchedule
```

기본 values.yaml이 아닌 custom 설정을 통해 사용하는 경우 아래의 명령어를 통해 오퍼레이터를 설치합니다.

- 이미 네임스페이스가 있는 경우

```
helm install whatap-operator whatap/whatap-operator -n whatap-monitoring -f
custom-values.yaml
```

- 네임스페이스가 없는 경우, 자동 생성하여 설치

```
helm upgrade --install whatap-operator whatap/whatap-operator -n whatap-monitoring
--create-namespace -f custom-values.yaml
```

- 기존 와탭 쿠버네티스 에이전트 사용자는 다음 명령어를 실행해 에이전트를 설치합니다.

```
kubectl delete ns whatap-monitoring
kubectl delete clusterrole whatap
kubectl delete clusterrolebinding whatap
kubectl create ns whatap-monitoring
kubectl create secret generic whatap-credentials --namespace whatap-monitoring --from-literal
WHATAP_LICENSE=$WHATAP_LICENSE --from-literal WHATAP_HOST=$WHATAP_HOST --from-literal
WHATAP_PORT=$WHATAP_PORT
helm install whatap-operator whatap/whatap-operator --namespace whatap-monitoring
```

> ⓘ 기존 yaml 파일 혹은 helm 방식으로 와탭 쿠버네티스 에이전트를 설치한 경우 Clean Install이 필요합니다. 기존 쿠버네티스 에이전트 삭제 후 Operator 사용을 권장합니다.

- 오퍼레이터는 deployment로 배포됩니다. 아래의 명령어로 operator가 정상적으로 배포됐는지 확인합니다.

```
kubectl get pods -n whatap-monitoring | grep -i operator
```

- 명령어 실행 결과, `whatap-operator` 파드가 **Running** 상태여야 합니다.

```
➜  operator kubectl get pods -n whatap-monitoring | grep -i operator
whatap-operator-658b745bb7-6nbgc       1/1     Running   0          4m30s
```

2. **WhatapAgent CR를 생성합니다.**

- 설치 화면에서 다운로드한 whatap_operator.yaml 파일을 적용해 `WhatapAgent` 커스텀 리소스를 생성합니다. CR 파일을 직접 작성한 경우 해당 파일명을 사용합니다.

```
kubectl apply -f whatap_operator.yaml
```

- `WhatapAgent` CR은 와탭 에이전트의 배포와 구성을 정의합니다. K8s 에이전트, APM 자동 설치, 그리고 Open Agent를 통한 **OpenMetrics** 수집을 설정할 수 있습니다.

최소 구성

기본 Kubernetes 모니터링을 위해 Whatap 마스터 에이전트와 노드 에이전트를 활성화하는 최소 구성입니다.

whatap_operator.yaml

```yaml
apiVersion: monitoring.whatap.com/v2alpha1
kind: WhatapAgent
metadata:
  name: whatap
  namespace: whatap-monitoring
spec:
  features:
    k8sAgent:
      masterAgent:
        enabled: true
      nodeAgent:
        enabled: true
        runtime: containerd                          # 노드의 컨테이너 런타임
        runtimeSocketPath: /var/run/containerd/containerd.sock   # 컨테이너 런타임 소켓 경로
      gpuMonitoring:
        enabled: false
```

`runtime`과 `runtimeSocketPath`는 노드의 컨테이너 런타임에 맞게 설정합니다. 설치 화면에서 컨테이너 런타임을 선택하면 아래 값이 자동으로 반영됩니다.

Container runtime	`runtime`	`runtimeSocketPath`
containerd(기본값)	`containerd`	/var/run/containerd/containerd.sock
Docker Engine	`docker`	/var/run/docker.sock
CRI-O	`crio`	/var/run/crio/crio.sock

GKE 환경에서는 `k8sAgent` 하위에 다음 설정을 추가합니다.

```yaml
    k8sAgent:
      gke:
        enabled: true
```

최소 구성으로 설치하게 되면 아래와 같이 whatap-master-agent, whatap-node-agent가 추가로 설치됩니다.

```
→  ~ kubectl get pods -n whatap-monitoring -w
NAME                                    READY   STATUS    RESTARTS   AGE
whatap-master-agent-588c688954-6pj2c    1/1     Running   0          96m
whatap-node-agent-cw8k2                 2/2     Running   0          96m
whatap-operator-7f9d9bc7d-2qp44         1/1     Running   0          96m
```

ⓘ 오퍼레이터로 와탭 에이전트 설치 시, CR 설정(자원제약, tolerlation, imagePullSecret)에 대한 자세한 내용은 WhaTap CR 설정 옵션 문서를 참고하시기 바랍니다.

기본 구성

```
apiVersion: monitoring.whatap.com/v2alpha1
kind: WhatapAgent
metadata:
  name: whatap
spec:
  features:
    ### APM 자동 설치 사용시 주석 해제 - APM 에이전트를 애플리케이션 Pod에 자동으로 주입하여 애플리케이션 성능 모니터링을 활성화합니다.
    # apm:
    #   instrumentation:
    #     targets:
    #       - name: hello-world
    #         enabled: true
    #         language: "java"          # 지원 언어: java, python, nodejs
    #         whatapApmVersions:
    #           java: "2.2.58"          # (선택) 특정 APM 에이전트 버전으로 고정할 때 지정합니다.
    #         namespaceSelector:
    #           matchNames:
    #             - default             # 모니터링할 애플리케이션이 있는 네임스페이스
    #         podSelector:
    #           matchLabels:
    #             app: "hello-world"    # 모니터링할 애플리케이션 Pod의 라벨
    #         config:
    #           mode: default           # APM 에이전트 모드 설정
```

```
### K8s 모니터링시 주석 해제 - Kubernetes 클러스터, 노드, 컨테이너 모니터링을 활성화합니다.
# k8sAgent:
#   masterAgent:
#     enabled: true               # 마스터 에이전트 활성화 (클러스터 수준 메트릭 수집)
#   nodeAgent:
#     enabled: true               # 노드 에이전트 활성화 (노드 및 컨테이너 수준 메트릭 수집)
#     runtime: containerd         # 노드의 컨테이너 런타임 (containerd, docker, crio)
#     runtimeSocketPath: /var/run/containerd/containerd.sock   # 컨테이너 런타임 소켓 경로
### GPU 모니터링 사용 시 주석 해제 - NVIDIA GPU 메트릭 수집을 활성화합니다.
#   gpuMonitoring:
#     enabled: true               # GPU 모니터링 활성화 (NVIDIA DCGM-EXPORTER 가 whatap-node-agent 사이드카로 설치됩니다)

### 오픈메트릭(프로메테우스 형태의 지표수집) 사용 시 주석 해제 - Prometheus 형식의 메트릭을 수집합니다.
# openAgent:
#     enabled: true               # OpenAgent 활성화
#     targets:
#       - targetName: kube-apiserver
#         type: ServiceMonitor      # 대상 유형: ServiceMonitor, PodMonitor, StaticEndpoints
#         namespaceSelector:
#           matchNames:
#             - "default"         # 메트릭을 수집할 네임스페이스
#         selector:
#           matchLabels:
#             component: apiserver  # 메트릭을 수집할 서비스/Pod의 라벨
#             provider: kubernetes
#         endpoints:
#           - port: "https"         # 메트릭 엔드포인트 포트
#             path: "/metrics"      # 메트릭 경로
#             interval: "30s"       # 이 엔드포인트의 스크래핑 간격
#             scheme: "https"       # HTTP 스키마 (http 또는 https)
#             tlsConfig:
#               insecureSkipVerify: true  # TLS 인증서 검증 건너뛰기
#         metricRelabelConfigs:
#           - source_labels: ["__name__"]
#             regex: "apiserver_request_total"  # 수집할 메트릭 필터링
#             action: "keep"                  # 일치하는 메트릭만 유지
```

ⓘ Operator 설치 방식은 Istio 사이드카 주입 설정을 지원하지 않습니다. Istio 환경 설정이 필요한 경우 Helm 또는 YAML 설치 방식을 사용하세요. kube-apiserver 등 컨트롤 플레인 메트릭은 OpenAgent 구성으로 수집합니다.

> ⓘ 자세한 내용은 오픈에이전트 구성 문서를 참고하세요.

다양한 구성 예제

와탭 오퍼레이터는 지원하는 K8s 에이전트와 더불어, APM 자동 설치, 커스텀 메트릭 에이전트 설치를 지원합니다. 아래의 구성예제를 통해 설치할 수 있습니다.

GPU 모니터링

Kubernetes 모니터링 에이전트를 설치하지 않고 와탭 노드 에이전트 파드에 DCGM-EXPORTER 컨테이너를 사이드카 형태로 배포합니다. 사용자는 별다른 구성없이 아래 yaml을 적용하여 수집되는 메트릭스는 **GPU 대시보드**에서 확인할 수 있습니다.

> ⓘ 노드 에이전트에 내장된 사이드카가 아니라 직접 DCGM-EXPORTER를 구성한 경우, OpenAgent(오픈에이전트) 구성 문서를 참고하여 dcgm **OpenMetrics**을 수집할 수 있습니다.

GPU 모니터링 자동 설치

GPU 모니터링 자동 설치

```yaml
apiVersion: monitoring.whatap.com/v2alpha1
kind: WhatapAgent
metadata:
  name: whatap
spec:
  features:
    openAgent:
      enabled: true  # Open Agent 활성화
    k8sAgent:
      masterAgent:
        enabled: true
      nodeAgent:
        enabled: true
      gpuMonitoring:
        enabled: true  # GPU 모니터링 활성화
```

참고. 자동 설치 시 Open Agent가 자동으로 DCGM Exporter를 발견하고 스크래핑합니다.

GPU 노드 톨러레이션 설정

GPU 노드에 테인트가 있는 경우 톨러레이션을 추가해야 합니다.

GPU 노드 톨러레이션 설정

```
apiVersion: monitoring.whatap.com/v2alpha1
kind: WhatapAgent
metadata:
  name: whatap
spec:
  features:
    k8sAgent:
      nodeAgent:
        enabled: true
        tolerations:
          - key: "nvidia.com/gpu"
            operator: "Exists"
            effect: "NoSchedule"
          - key: "gpu"
            operator: "Exists"
            effect: "NoSchedule"
      gpuMonitoring:
        enabled: true
```

APM 자동 설치

Kubernetes 모니터링 에이전트를 활성화하지 않고 쿠버네티스 클러스터에 존재하는 APM을 자동 설치합니다.

> ⓘ 자세한 내용은 APM 자동 설치 구성 문서를 참고하세요.

APM 자동 설치

```
apiVersion: monitoring.whatap.com/v2alpha1
kind: WhatapAgent
metadata:
```

```
  name: whatap
spec:
  features:
    apm:
      instrumentation:
        targets:
          - name: hello-world
            enabled: true
            language: "java"
            whatapApmVersions:
              java: "2.2.58"
            namespaceSelector:
              matchNames:
                - default
            podSelector:
              matchLabels:
                app: "hello-world"
            config:
              mode: default
```

OpenAgent

Kubernetes 모니터링 에이전트나 APM 계측을 활성화하지 않고 Prometheus 스타일 메트릭을 수집하기 위한 OpenAgent 컴포넌트만 구성합니다.

> ⓘ 자세한 내용은 오픈에이전트 구성 문서를 참고하세요.

```
apiVersion: monitoring.whatap.com/v2alpha1
kind: WhatapAgent
metadata:
  name: whatap
spec:
  features:
    openAgent:
      enabled: true
      targets:
        - targetName: kube-apiserver
```

```
    type: ServiceMonitor
    namespaceSelector:
      matchNames:
        - "default"
    selector:
      matchLabels:
        component: apiserver
        provider: kubernetes
    endpoints:
    - port: "https"
      path: "/metrics"
      interval: "30s"
      scheme: "https"
      tlsConfig:
        insecureSkipVerify: true
      metricRelabelConfigs:
        - source_labels: ["__name__"]
          regex: "apiserver_request_total"
          action: "keep"
```

Kubernetes 모니터링과 APM 계측 함께 사용

```
apiVersion: monitoring.whatap.com/v2alpha1
kind: WhatapAgent
metadata:
  name: whatap
spec:
  features:
    apm:
      instrumentation:
        targets:
          - name: hello-world
            enabled: true
            language: "java"
            whatapApmVersions:
              java: "2.2.58"
            namespaceSelector:
              matchNames:
                - default
```

```
      podSelector:
        matchLabels:
          app: "hello-world"
      config:
        mode: default
k8sAgent:
  masterAgent:
    enabled: true
  nodeAgent:
    enabled: true
openAgent:
    enabled: true
    targets:
      - targetName: kube-apiserver
        type: ServiceMonitor
        namespaceSelector:
          matchNames:
            - "default"
        selector:
          matchLabels:
            component: apiserver
            provider: kubernetes
        endpoints:
        - port: "https"
          path: "/metrics"
          interval: "30s"
          scheme: "https"
          tlsConfig:
            insecureSkipVerify: true
          metricRelabelConfigs:
          - source_labels: ["__name__"]
            regex: "apiserver_request_total"
            action: "keep"
```

APM 자동 설치 구성

와탭 오퍼레이터 설치 후, `WhatapAgent` CR(Custom Resource)을 정의하면 쿠버네티스 파드나 네임스페이스에 APM 에이전트를 자동 주입합니다.

설치 전 요구사항

- Kubernetes 환경
- Helm 3.2 이상
- Whatap 라이선스 키(필수)
- Whatap 수집 서버 정보(IP 주소와 포트)
- whatap-operator(오퍼레이터 설치 가이드)

WhatapAgent CR 기본 구성

다음은 CR을 구성한 기본 샘플입니다. 추가 설정이 필요하면 APM 설정 예제를 참고하세요.

> ⓘ 이 문서의 CR 예제는 whatap-operator Helm 차트 **1.9.9** 기준으로 검증했습니다. 차트 버전에 따라 CR 구조가 달라질 수 있으므로, 설치한 버전과 다르면 `kubectl explain whatapagent.spec` 으로 확인하세요.

```yaml
apiVersion: monitoring.whatap.com/v2alpha1
kind: WhatapAgent
metadata:
  name: default                    # 리소스 이름
  namespace: whatap-monitoring     # 설치 대상 네임스페이스
spec:
  features:
    apm:
      instrumentation:
        targets:
          - name: "java-sample"    # APM 대상 이름 (식별용)
```

```
    enabled: "true"              # 자동 주입 활성화 여부
    language: "java"             # 언어: java, python, nodejs 등
    whatapApmVersions:
      java: "2.2.58"             # 해당 언어의 APM 에이전트 버전
    namespaceSelector:
      matchNames:
        - default                # APM 주입 대상 네임스페이스
    podSelector:
      matchLabels:
        app: hello-world         # APM 주입 대상 파드 라벨
    config:
      mode: default              # default 또는 custom. custom 은 configMapRef 필요
    envs:
      WHATAP_JAVA_AGENT_PATH: /whatap-agent/whatap.agent.java.jar
      OKIND: hello
```

APM 설정 예제

특정 네임스페이스 파드에 자동 주입

- `backend` 네임스페이스의 `app=shop` 라벨을 가진 파드에 Java APM을 자동 주입합니다.
- 현재 Java는 `latest` 버전만 지원하며, 다른 언어는 추후 지원 예정입니다.
- `ConfigMapRef`로 설정 정보를 전달합니다. ConfigMap은 대상 애플리케이션과 같은 네임스페이스에 있어야 합니다.

```
apiVersion: monitoring.whatap.com/v2alpha1
kind: WhatapAgent
metadata:
  name: whatap
  namespace: whatap-monitoring
spec:
  features:
    apm:
      instrumentation:
        targets:
          - name: shop-api
            enabled: "true"
```

```
    language: "java"
    whatapApmVersions:
      java: "latest"
    namespaceSelector:
      matchNames:
        - backend
    podSelector:
      matchLabels:
        app: shop
    config:
      mode: custom
      configMapRef:
        name: shop-api-config
```

shop-api-config.yaml

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: shop-api-config
  namespace: backend
data:
  whatap.conf: |
    mtrace_enabled=true
    mtrace_basetime=0
    trace_normalize_enabled=true
```

워크로드별 ConfigMap 분리 설정

`system` 네임스페이스의 `app=router` 파드와 `backend` 네임스페이스의 `app=shop` 파드에 각각 다른 ConfigMap(`request-router-config` , `shop-api-config`)을 주입합니다.

```
apiVersion: monitoring.whatap.com/v2alpha1
kind: WhatapAgent
metadata:
  name: whatap
  namespace: whatap-monitoring
spec:
  features:
```

```
  apm:
    instrumentation:
      targets:
        - name: request-router
          enabled: "true"
          language: "java"
          whatapApmVersions:
            java: "latest"
          namespaceSelector:
            matchNames:
              - system
          podSelector:
            matchLabels:
              app: router
          config:
            mode: custom
            configMapRef:
              name: request-router-config
        - name: shop-api
          enabled: "true"
          language: "java"
          whatapApmVersions:
            java: "latest"
          namespaceSelector:
            matchNames:
              - backend
          podSelector:
            matchLabels:
              app: shop
          config:
            mode: custom
            configMapRef:
              name: shop-api-config
```

shop-api-config.yaml

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: shop-api-config
```

```
  namespace: backend
data:
  whatap.conf: |
    mtrace_enabled=true
    mtrace_basetime=0
    trace_normalize_enabled=true
```

request-router-config.yaml

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: request-router-config
  namespace: backend
data:
  whatap.conf: |
    mtrace_enabled=false
    mtrace_basetime=100
    trace_normalize_enabled=true
```

CR 필드 레퍼런스

```
apiVersion: monitoring.whatap.com/v2alpha1
kind: WhatapAgent
metadata:
  name: whatap
spec:
  features:
    apm:
      instrumentation:
        targets:
          - name: <name>
            enabled: <true | false>
            language: <java | python | nodejs>
            whatapApmVersions:
              <language> : <version>
            namespaceSelector:
              matchNames:
```

```
    - <namespace>
  podSelector:
    matchLabels:
      <key>: <values>
  config:
    mode: default
```

기본 필드

Field	Description
`name`	타겟 이름. 모니터링에는 영향 없고 식별용입니다.
`enabled`	`"true"`일 경우 해당 타겟에 APM 에이전트를 자동 주입합니다.
`language`	타겟 애플리케이션의 언어. `java`, `python`, `nodejs` 등
`whatapApmVersions`	언어별 와탭 APM 에이전트 버전 지정
`namespaceSelector`	APM 자동 주입 대상 네임스페이스 목록
`podSelector`	APM 자동 주입 대상 파드를 라벨 기준으로 선택
`config.mode`	`"default"`(기본값) 또는 `"custom"`. `"custom"`으로 지정하면 `config.configMapRef.name`이 필수입니다.

네임스페이스 선택자 (namespaceSelector)

`WhatapAgent`는 쿠버네티스 표준 라벨 선택자를 지원합니다.

Field	Description
`matchNames`	대상 네임스페이스 이름 목록
`matchLabels`	키-값 쌍으로 네임스페이스 라벨 선택 (AND 조건)

Field	Description
matchExpressions	고급 라벨 선택 표현식 (AND 조건)

파드 선택자 (podSelector)

Field	Description
matchLabels	키-값 쌍으로 파드 라벨 선택 (AND 조건)
matchExpressions	고급 라벨 선택 표현식 (AND 조건)

matchExpressions 연산자

Operator	Description
In	라벨 값이 지정된 값 목록에 포함되어야 함
NotIn	라벨 값이 지정된 값 목록에 포함되지 않아야 함
Exists	지정된 키의 라벨이 존재해야 함 (values 필드는 비움)
DoesNotExist	지정된 키의 라벨이 존재하지 않아야 함 (values 필드는 비움)

K8sAgent 설정

Field	Description
agentImageVersion	에이전트 이미지 버전 (기본값: latest)
masterAgent.enabled	마스터 에이전트 활성화 여부
masterAgent.resources	마스터 에이전트의 리소스 요청 및 제한 설정

Field	Description
masterAgent.envs	마스터 에이전트에 추가할 환경 변수 목록
nodeAgent.enabled	노드 에이전트 활성화 여부
nodeAgent.resources	노드 에이전트의 리소스 요청 및 제한 설정
nodeAgent.envs	노드 에이전트에 추가할 환경 변수 목록
gpuMonitoring.enabled	GPU 모니터링 활성화 여부
apiserverMonitoring.enabled	API Server 모니터링 활성화 여부
etcdMonitoring.enabled	etcd 모니터링 활성화 여부
schedulerMonitoring.enabled	Scheduler 모니터링 활성화 여부

```yaml
# 여러 조건을 조합한 선택자 예제
namespaceSelector:
  matchNames:
    - production
    - staging
  matchLabels:
    environment: production
  matchExpressions:
    - {key: tier, operator: In, values: [frontend, backend]}
    - {key: restricted, operator: DoesNotExist}

podSelector:
  matchLabels:
    app: web-service
    version: v2
  matchExpressions:
    - {key: tier, operator: In, values: [backend, backend2, backend3]}
    - {key: environment, operator: NotIn, values: [dev, test]}
    - {key: ready, operator: Exists}
```

- namespaceSelector 상세 설명

네임스페이스 선택자는 세 가지 방식으로 대상 네임스페이스를 필터링합니다.

1. **matchNames**: 네임스페이스 이름으로 직접 선택
    1. `production` 또는 `staging` 이름을 가진 네임스페이스만 선택
    2. 이름 기반 필터링은 가장 기본적인 방법
2. **matchLabels**: 키-값 쌍으로 라벨 일치 여부 확인(AND 조건)
    1. `environment=production` 라벨을 가진 네임스페이스만 선택
    2. 모든 키-값 쌍이 일치해야 함(여러 라벨이 있을 경우 모두 만족해야 함)
3. **matchExpressions**: 고급 표현식으로 복잡한 조건 지정(AND 조건)
    1. `tier` 라벨이 `frontend` 또는 `backend` 값을 가져야 함(`In` 연산자)
    2. `restricted` 라벨이 존재하지 않아야 함(`DoesNotExist` 연산자)
    3. 모든 표현식이 참이어야 함(AND 조건)

- **podSelector 상세 설명**

  파드 선택자는 두 가지 방식으로 대상 파드를 필터링합니다.

  1. **matchLabels**: 키-값 쌍으로 라벨 일치 여부 확인(AND 조건)
      1. `app=web-service` 및 `version=v2` 라벨을 모두 가진 파드만 선택
      2. 지정된 모든 라벨이 정확히 일치해야 함
  2. **matchExpressions**: 고급 표현식으로 복잡한 조건 지정(AND 조건)
      1. `tier` 라벨이 `backend` , `backend2` , `backend3` 중 하나여야 함(`In` 연산자)
      2. `environment` 라벨이 `dev` 또는 `test` 가 아니어야 함(`NotIn` 연산자)
      3. `ready` 라벨이 존재해야 함(`Exists` 연산자)
      4. 모든 표현식이 참이어야 함(AND 조건)

테스트 샘플

APM 자동 주입을 테스트하려면 다음 YAML을 받아 설치하세요.

test-values.yaml

```yaml
### 테스트 목적 샘플 자바 애플리케이션
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-world
  labels:
    app: hello-world
spec:
  replicas: 1
  selector:
    matchLabels:
      app: hello-world
  template:
    metadata:
      labels:
        app: hello-world
    spec:
      containers:
      - name: hello-world
        image: whatap/base_java:0.0.13   # or 사용자 애플리케이션 이미지
        ports:
        - containerPort: 8080
---

apiVersion: monitoring.whatap.com/v2alpha1
kind: WhatapAgent
metadata:
  name: default                          # 리소스 이름
  namespace: whatap-monitoring               # 설치 대상 네임스페이스
spec:
  features:
    apm:
      instrumentation:
        targets:
          - name: "java-sample"              # APM 대상 이름 (식별용)
            enabled: "true"                # 자동 주입 활성화 여부
            language: "java"               # 언어: java, python, nodejs 등
            whatapApmVersions:
              java: "latest"               # 해당 언어의 APM 에이전트 버전
```

```
  namespaceSelector:
    matchNames:
      - default                  # APM 주입 대상 네임스페이스
  podSelector:
    matchLabels:
      app: hello-world           # APM 주입 대상 파드 라벨
  config:
    mode: default                # default 또는 custom. custom 은 configMapRef 필요
```

OpenAgent 구성

OpenAgent는 프로메테우스 엔드포인트에서 메트릭을 스크래핑하고, 이를 처리하여 와탭 서버로 전송하는 역할을 합니다.
프로메테우스 엔드포인트에서 메트릭을 수집하고 와탭 서버로 전송하는 Go 기반 에이전트입니다.

설정

단독 yaml로 설치하는 경우

기본적으로 오픈 에이전트는 whatap-monitoring 네임스페이스에 생성됩니다. 필요에 따라 조정할 수 있습니다.

```yaml
# Whatap Open Agent standalone manifests
# 네임스페이스는 필요에 따라 조정하세요. 예시로 whatap-monitoring을 사용합니다.
---
#apiVersion: v1
#kind: Namespace
#metadata:
#  name: whatap-monitoring
---
# Whatap 접속 정보 (라이센스/호스트/포트)
# 이미 동일한 이름의 Secret을 별도로 만들어 두었다면 이 Secret 블록을 삭제하고,
# 아래 Deployment의 secretKeyRef.name을 기존 Secret 이름과 맞추면 됩니다.
apiVersion: v1
kind: Secret
metadata:
  name: whatap-open-credentials
  namespace: whatap-monitoring
type: Opaque
stringData:
  WHATAP_LICENSE: "<YOUR_LICENSE>"
  WHATAP_HOST: "<WHATAP_HOST>"
  WHATAP_PORT: "6600"

---
apiVersion: v1
kind: ServiceAccount
metadata:
```

```
  name: whatap-open-agent-sa
  namespace: whatap-monitoring

---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: whatap-open-agent-role
rules:
  - apiGroups: ["*"]
    resources: ["pods", "services", "endpoints", "endpointslices", "namespaces", "secrets"]
    verbs: ["get", "list", "watch"]
  - nonResourceURLs: ["/metrics"]
    verbs: ["*"]
  - apiGroups: [""]
    resources: ["pods/exec"]
    verbs: ["create"]
  - apiGroups: [""]
    resources: ["configmaps"]
    verbs: ["*"]
  - nonResourceURLs: ["/metrics"]
    verbs: ["*"]

---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: whatap-open-agent-role-binding
subjects:
  - kind: ServiceAccount
    name: whatap-open-agent-sa
    namespace: whatap-monitoring
roleRef:
  kind: ClusterRole
  name: whatap-open-agent-role
  apiGroup: rbac.authorization.k8s.io

---
apiVersion: v1
```

```
kind: ConfigMap
metadata:
  name: whatap-open-agent-config
  namespace: whatap-monitoring
data:
  scrape_config.yaml: |
    features:
      openAgent:
        enabled: true
        globalInterval: "60s"
        globalPath: "/metrics"
        targets:
          - targetName: kube-apiserver
            type: ServiceMonitor
            namespaceSelector:
              matchNames:
                - "default"  # API 서버 서비스가 있는 네임스페이스로 변경
            selector:
              matchLabels:
                component: apiserver
                provider: kubernetes
            endpoints:
              - port: "https"
                path: "/metrics"
                interval: "30s"
                scheme: "https"
                timeout: "30s"
                tlsConfig:
                  insecureSkipVerify: true
                metricRelabelConfigs:
                  - source_labels: [__name__]
                    regex: "apiserver_request_total"
                    action: keep
                  - source_labels: [verb]
                    target_label: http_verb
                    replacement: "${1}"
                    action: replace
                  - target_label: metric_src
                    replacement: "whatap-open-agent"
```

```
                action: replace

---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: whatap-open-agent
  namespace: whatap-monitoring
  labels:
    name: whatap-open-agent
spec:
  replicas: 1
  selector:
    matchLabels:
      name: whatap-open-agent
  template:
    metadata:
      labels:
        name: whatap-open-agent
    spec:
      serviceAccountName: whatap-open-agent-sa
      volumes:
        - name: config-volume
          configMap:
            name: whatap-open-agent-config
        - name: logs-volume
          emptyDir: {}
      containers:
        - name: whatap-open-agent
          image: public.ecr.aws/whatap/open_agent:latest
          imagePullPolicy: Always
          resources:
            requests:
              memory: "300Mi"
              cpu: "100m"
            limits:
              memory: "1000Mi"
              cpu: "1000m"
          env:
```

```
    - name: WHATAP_LICENSE
      valueFrom:
        secretKeyRef:
          name: whatap-open-credentials
          key: WHATAP_LICENSE
    - name: WHATAP_HOST
      valueFrom:
        secretKeyRef:
          name: whatap-open-credentials
          key: WHATAP_HOST
    - name: WHATAP_PORT
      valueFrom:
        secretKeyRef:
          name: whatap-open-credentials
          key: WHATAP_PORT
    - name: debug
      value: "false"
  volumeMounts:
    - name: config-volume
      mountPath: /app/scrape_config.yaml
      subPath: scrape_config.yaml
    - name: logs-volume
      mountPath: /app/logs
  # 필요 시 컨테이너/노드 톨러레이션, affinity 등을 추가하십시오.
```

helm 설치

1. 다음 명령어를 실행해 와탭 Helm 리포지토리를 추가하세요.

```
helm repo add whatap https://whatap.github.io/helm/
helm repo update
```

2. 에이전트를 처음 설치하는 사용자는 다음 명령어를 실행해 에이전트 애플리케이션을 설치하세요. values.yaml은 아래 예시를 참고합니다.

- **처음 설치** **기존 사용자**

```
#에이전트를 처음 설치하는 사용자는 다음 명령어를 실행해 에이전트 애플리케이션을 설치하세요.
kubectl create ns whatap-monitoring
helm install whatap-kube-agent --namespace whatap-monitoring whatap/kube -f values.yaml
```

```
#기존 와탭 쿠버네티스 에이전트 사용자는 다음 명령어를 실행해 에이전트를 설치하세요.
#*기존 yaml 파일 방식으로 와탭 쿠버네티스 에이전트를 설치한 경우 Clean Install이 필요합니다. 기존 쿠버네티스 에이전트 삭제 후 Helm 차트를 통한 설치가 가능합니다.
kubectl delete ns whatap-monitoring
kubectl delete clusterrole whatap
kubectl delete clusterrolebinding whatap
kubectl create ns whatap-monitoring
helm install whatap-kube-agent --namespace whatap-monitoring whatap/kube -f values.yaml
```

values.yaml

```yaml
whatap:
  license: #라이센스 키 입력
  host: #수집서버 IP 입력
  port: #수집서버 Port 입력
  createSecret: true #default : true, if credentials already exist, set false
  existingSecretName: "whatap-credentials" #default : whatap-credentials
collect_custom_resources: true
containerRuntime: "containerd"
deployment:
  containers:
    controlPlaneHelper:
      enabled: false
      image: "whatap/kube_mon:1.8.8"
      name: "whatap-control-plane-helper"
      port: 9496
      resources:
        limits:
          cpu: "200m"
          memory: "350Mi"
        requests:
          cpu: "100m"
          memory: "300Mi"
```

```
  masterAgent:
    image: "whatap/kube_mon:1.8.8"
    name: "whatap-master-agent"
    port: 6600
    resources:
      limits:
        cpu: "200m"
        memory: "350Mi"
      requests:
        cpu: "100m"
        memory: "300Mi"
  label: "whatap-master-agent"
  name: "whatap-master-agent"
  replicas: 1

daemonSet:
  containers:
    nodeAgent:
      image: "whatap/kube_mon:1.8.8"
      limits:
        cpu: "200m"
        memory: "350Mi"
      name: "whatap-node-agent"
      requests:
        cpu: "100m"
        memory: "300Mi"
    nodeHelper:
      image: "whatap/kube_mon:1.8.8"
      limits:
        cpu: "200m"
        memory: "350Mi"
      name: "whatap-node-helper"
      requests:
        cpu: "100m"
        memory: "100Mi"
  initContainers:
    nodeDebugger:
      enabled: false
      image: "whatap/kube_mon:1.8.8"
```

```
    name: "whatap-node-debug"
  label: "whatap-node-agent"
  name: "whatap-node-agent"

clusterrole:
  extraResources:
  - "*"

deploymentOpen:
  enabled: true
  containers:
    openAgent:
      image: "public.ecr.aws/whatap/open_agent:latest"
  config:
    features:
      openAgent:
        enabled: true
        #globalInterval: "60s"
        #globalPath: "/metrics"
        targets:
          - targetName: "kube-apiserver"
            type: "ServiceMonitor"
            enabled: true
            namespaceSelector:
              matchNames:
                - "default"   # 클러스터 기본 레이블에 맞게 조정
            selector:
              matchLabels:
                component: "apiserver"
                provider: "kubernetes"
            endpoints:
              - port: "https"
                path: "/metrics"
                interval: "30s"
                scheme: "https"
                timeout: "30s"
                tlsConfig:
                  insecureSkipVerify: true   # 자체서명 인증서일 때 편의용, 보안 필요 시 false + CA 지정
                metricRelabelConfigs:
```

```
    - source_labels: [__name__]
      regex: "apiserver_request_total"
      action: keep
    - source_labels: [verb]
      target_label: http_verb
      replacement: "${1}"
      action: replace
    - target_label: metric_src
      replacement: "whatap-open-agent"
      action: replace
```

WhatapOperator 설치 후, 사용자는 `WhatapAgent` CR(Custom Resource)을 통해 **OpenMetrics** 대상을 설정합니다.

1. **PodMonitor**: Pod 레이블 셀렉터를 이용한 동적 디스커버리 (Prometheus Operator의 PodMonitor와 유사)
2. **ServiceMonitor**: Service 레이블 셀렉터를 이용한 동적 디스커버리 (Prometheus Operator의 ServiceMonitor와 유사)
3. **StaticEndpoints**: 고정된 IP 주소와 포트를 직접 입력 (Prometheus의 static_configs와 유사)

```
features:
  openAgent:
    enabled: true
    targets:
      # 1. PodMonitor: Pod 레이블 셀렉터를 이용한 동적 디스커버리
      - targetName: my-app-pod-metrics
        type: PodMonitor
        namespaceSelector:
          matchNames:
            - "production"
        selector:
          matchLabels:
            app: my-app
        endpoints:
        - port: "web-metrics"  # Pod Spec에 정의된 Port 이름 또는 실제 Port 번호
          path: "/metrics"     # 기본값은 /metrics, 필요시 재정의
          interval: "15s"      # 기본값은 전역 설정, 필요시 재정의
          scheme: "http"
          timeout: "10s"
```

```
    metricRelabelConfigs:  # 스크래핑 후 메트릭 재라벨링 설정
      - source_labels: [__name__]
        regex: "http_requests_total"
        action: keep
      - source_labels: [method]
        target_label: http_method
        replacement: "${1}"
        action: replace

  # 2. ServiceMonitor: Service 레이블 셀렉터를 이용한 동적 디스커버리
  - targetName: my-service-metrics
    type: ServiceMonitor
    namespaceSelector:
      matchNames:
        - "default"
    selector:
      matchLabels:
        service: my-backend-service
    endpoints:
    - port: "http-metrics"  # Service Spec에 정의된 Port 이름 또는 실제 Target Port 번호
      path: "/actuator/prometheus"
      interval: "30s"
      metricRelabelConfigs:
        - source_labels: [__name__]
          regex: "jvm_.*"
          action: keep
        - source_labels: [area]
          target_label: memory_area
          replacement: "${1}"
          action: replace

  # 3. StaticEndpoints: 고정된 IP 주소와 포트를 직접 입력
  - targetName: my-external-db-metrics
    type: StaticEndpoints
    # enabled: true  # 타겟 활성화 여부 (기본값: true, 생략 가능)
    endpoints:
      - address: "192.168.1.100:9100"
        scheme: "http"
        path: "/metrics"
        interval: "60s"
        metricRelabelConfigs:
```

```
    - source_labels: [__name__]
      regex: "node_(cpu|memory).*"
      action: keep
    - source_labels: [instance]
      target_label: server
      replacement: "${1}"
      action: replace
```

PodMetrics 및 ServiceMetrics 설정 요소

- **targetName**: 타겟의 이름(로깅 및 식별용)
- **type**: 타겟 유형("PodMetrics" 또는 "ServiceMetrics")
- **namespaceSelector**: 스크래핑할 네임스페이스
  - `matchNames` : 이름으로 네임스페이스 선택
  - `matchLabels` : 레이블로 네임스페이스 선택
  - `matchExpressions` : 표현식으로 네임스페이스 선택
- **selector**: 스크래핑할 파드 또는 서비스 선택
  - `matchLabels` : 레이블로 파드 또는 서비스 선택
  - `matchExpressions` : 표현식으로 파드 또는 서비스 선택
- **relabelConfigs**: 타겟 발견(Discovery) 단계에서 메타데이터(meta 라벨 등)를 기반으로 라벨을 추가하거나 변경합니다.(프로메테우스의 `relabel_configs` 와 유사)
- **endpoints**: 스크래핑할 엔드포인트 정의
  - `port` : 스크래핑할 포트 이름 또는 번호
  - `path` : 메트릭 경로(기본값은 globalPath, 필요시 재정의)
  - `interval` : 스크래핑 간격(기본값은 globalInterval, 필요시 재정의)
  - `scheme` : 스크래핑 프로토콜(http 또는 https, 기본값 http)
  - `timeout` : 스크래핑 타임아웃
  - `honorLabels` : 대상에서 제공하는 레이블을 우선시할지 여부
  - `metricRelabelConfigs` : 스크래핑 후 메트릭 재라벨링 설정(프로메테우스의 metric_relabel_configs와 유사)

StaticEndpoints 설정 요소

- **targetName**: 타겟의 이름(로깅 및 식별용)
- **type**: 타겟 유형("StaticEndpoints")
- **endpoints**: 스크래핑할 엔드포인트 정의
  - `address` : 스크래핑할 대상 주소 (IP:PORT 또는 HOSTNAME:PORT)
  - `path` : 메트릭 경로 (기본값: /metrics)
  - `scheme` : 스크래핑 프로토콜 (http 또는 https, 기본값 http)
  - `interval` : 스크래핑 간격 (기본값: 60s)
  - `tlsConfig` : TLS 설정
  - `metricRelabelConfigs` : 스크래핑 후 메트릭 재라벨링 설정

addNodeLabel 기능

스크랩 대상이 노드에 스케쥴링 되어 있는 경우 `addNodeLabel` 옵션을 사용하여 스케줄링된 노드의 이름을 메트릭에 라벨로 추가할 수 있습니다.

- **설정 위치**: 타겟 레벨 또는 엔드포인트 레벨에서 설정 가능
- **기본값**: `false`
- **동작**: `true` 로 설정하면 모든 메트릭에 `node` 라벨이 추가되며, 오브젝트가 실행 중인 노드의 이름이 값으로 추가됨

사용 예제

```yaml
# 타겟 레벨에서 addNodeLabel 설정
- targetName: node-exporter
  type: PodMonitor
  namespaceSelector:
    matchNames:
      - "monitoring"
  selector:
    matchLabels:
      app: node-exporter
  endpoints:
    - port: "metrics"
```

```
    path: "/metrics"
    interval: "30s"
    addNodeLabel: true  # 이 엔드포인트에 노드 라벨 추가
    metricRelabelConfigs:
     # 노드 라벨을 사용한 재라벨링 예제
     - source_labels: [node]
       target_label: kubernetes_node
       action: replace

# 엔드포인트 레벨에서 addNodeLabel 설정
- targetName: dcgm-exporter
  type: PodMonitor
  namespaceSelector:
    matchNames:
     - "kube-system"
  selector:
    matchLabels:
     app: dcgm-exporter
  endpoints:
  - port: "metrics"
    path: "/metrics"
    interval: "30s"
    addNodeLabel: true  # 이 엔드포인트에만 노드 라벨 추가
            metricRelabelConfigs:
     - source_labels: [node]
       target_label: gpu_node
       action: replace
```

Basic Auth 설정

OpenAgent는 인증이 필요한 엔드포인트를 스크래핑하기 위해 HTTP Basic Authentication을 지원합니다. 사용자 이름(username)과 비밀번호(password)는 Kubernetes Secret을 통해 안전하게 관리됩니다.

- 설정 위치: endpoints 하위의 basicAuth 항목에서 설정
- **username**: 사용자 이름을 담고 있는 Secret 정보
  - `name` : Secret의 이름
  - `key` : Secret 내의 Key
  - `namespace` : Secret이 위치한 네임스페이스 (생략 시 에이전트와 동일한 네임스페이스)

- **password**: 비밀번호를 담고 있는 Secret 정보
  - `name` : Secret의 이름
  - `key` : Secret 내의 Key
  - `namespace` : Secret이 위치한 네임스페이스

설정 예제

```
endpoints:
 - port: "metrics"
   path: "/metrics"
   basicAuth:
     username:
       name: my-auth-secret
       key: username
     password:
       name: my-auth-secret
       key: password
       namespace: monitoring # Secret이 다른 네임스페이스에 있는 경우 명시
```

TLS 설정

OpenAgent는 HTTPS 엔드포인트에 연결할 때 TLS(Transport Layer Security)를 지원합니다. 다음은 TLS 관련 설정 옵션입니다.

HTTP vs HTTPS 결정 방법

OpenAgent는 다음과 같은 규칙에 따라 HTTP 또는 HTTPS 프로토콜을 사용할지 결정합니다.

- **PodMonitor 및 ServiceMonitor 타겟**
  - 포트 이름이 "https"인 경우 기본적으로 HTTPS를 사용합니다.
  - 그 외의 경우 기본적으로 HTTP를 사용합니다.
- **StaticEndpoints 타겟**
  - TLS 설정이 존재하는 경우 기본적으로 HTTPS를 사용합니다.

    - 그 외의 경우 기본적으로 HTTP를 사용합니다.

- **모든 타겟 유형**
    - 엔드포인트나 타겟에 명시적으로 `scheme` 설정이 있는 경우, 이 설정이 기본값을 재정의합니다.

TLS 설정 옵션

TLS 설정은 `tlsConfig` 섹션에서 구성할 수 있습니다.

```
endpoints:
 - port: "https"
   path: "/metrics"
   scheme: "https"  # 명시적으로 HTTPS 사용 지정
   tlsConfig:
     insecureSkipVerify: true  # 인증서 검증 건너뛰기
```

insecureSkipVerify

`insecureSkipVerify` 옵션은 서버 인증서의 유효성 검사를 건너뛰도록 설정합니다. 이 옵션은 다음과 같은 경우에 유용합니다.

- 자체 서명된 인증서를 사용하는 서버에 연결할 때
- 개발 또는 테스트 환경에서 인증서 검증이 필요하지 않을 때
- 내부 네트워크에서 신뢰할 수 있는 서버에 연결할 때

> **주의**
>
> 프로덕션 환경에서는 보안상의 이유로 `insecureSkipVerify: false` 를 사용하는 것이 좋습니다. 자체 서명된 인증서를 사용하는 경우, 인증서를 신뢰할 수 있는 인증 기관(CA)으로 추가하는 것이 더 안전한 방법입니다.

설정 예제

ServiceMonitor에서 TLS 설정 예제

```
- targetName: kube-apiserver
```

```
type: ServiceMonitor
namespaceSelector:
  matchNames:
    - "default"
selector:
  matchLabels:
    component: apiserver
    provider: kubernetes
endpoints:
  - port: "https"  # 포트 이름이 "https"이므로 기본적으로 HTTPS 사용
    path: "/metrics"
    interval: "30s"
    scheme: "https"  # 명시적으로 HTTPS 지정 (선택사항)
    tlsConfig:
      insecureSkipVerify: true  # 인증서 검증 건너뛰기
    metricRelabelConfigs:
      - source_labels: [__name__]
        regex: "apiserver_request_total"
        action: keep
      - source_labels: [verb]
        target_label: http_verb
        replacement: "${1}"
        action: replace
```

타겟 재라벨링 설정(relabel Configs)

relabelConfigs는 스크래핑을 수행하기 전, 타겟(Target) 자체의 메타데이터를 기반으로 필터링하거나 라벨을 조작하는 기능입니다. (Prometheus의 `relabel_config`s와 유사)

1. **타겟 필터링 (Target Filtering)**
   - 특정 메타데이터 조건을 만족하지 않는 타겟을 수집 대상에서 제외합니다.
   - 주요 Action: `keep`, `drop`
2. **메타 라벨 승격 (Meta Label Promotion)**
   - `__meta_`로 시작하는 메타데이터는 스크래핑 후 사라집니다.
   - 영구적인 라벨로 저장하려면 이 단계에서 `replace` action을 사용하여 새로운 이름으로 변경해야 합니다.

Meta Labels 목록

OpenAgent에서 제공하는 주요 메타 라벨 목록입니다.

Common (공통)

- `__meta_kubernetes_namespace` : 오브젝트의 네임스페이스
- `__address__` : 타겟 주소 (IP:Port)

PodMonitor

- `__meta_kubernetes_pod_name` : Pod 이름
- `__meta_kubernetes_pod_ip` : Pod IP
- `__meta_kubernetes_pod_ready` : Pod 준비 상태 (`true` / `false`)
- `__meta_kubernetes_pod_phase` : Pod 상태 (`Running` , `Pending` 등)
- `__meta_kubernetes_pod_node_name` : Pod가 실행 중인 노드 이름
- `__meta_kubernetes_pod_label_<labelname>` : Pod의 라벨
- `__meta_kubernetes_pod_annotation_<annotationname>` : Pod의 어노테이션

ServiceMonitor

- `__meta_kubernetes_service_name` : Service 이름
- `__meta_kubernetes_service_cluster_ip` : Cluster IP
- `__meta_kubernetes_service_label_<labelname>` : Service의 라벨

```
targets:
  - targetName: my-pod-monitor
    type: PodMonitor
    selector:
      matchLabels:
        app: my-app
    # 타겟 발견 단계의 라벨링
    relabelConfigs:
      # 1. Pod 이름을 pod_name 라벨로 저장 (메트릭에 pod_name 라벨이 붙음)
```

```
  - source_labels: ["__meta_kubernetes_pod_name"]
    target_label: "pod_name"
    action: "replace"

  # 2. 특정 라벨(scrape_enabled=true)이 없는 Pod는 수집 제외
  - source_labels: ["__meta_kubernetes_pod_label_scrape_enabled"]
    regex: "true"
    action: "keep"
```

메트릭 재라벨링 설정(metricRelabelConfigs)

OpenAgent는 프로메테우스의 metric_relabel_configs와 유사한 메트릭 재라벨링 기능을 지원합니다. 이 기능을 사용하면 스크래핑 후 메트릭을 필터링하거나 레이블을 변경할 수 있습니다.

재라벨링 설정 요소

- **source_labels**: 소스 레이블 목록(배열)
- **separator**: 소스 레이블 값을 연결할 때 사용할 구분자(기본값: `;`)
- **target_label**: 대상 레이블(결과를 저장할 레이블)
- **regex**: 소스 레이블 값에 적용할 정규식
- **replacement**: 대체 값(정규식 캡처 그룹 참조 가능, 예: `${1}`)
- **action**: 수행할 작업(keep, drop, replace)

지원되는 작업(action)

- **keep**: 정규식과 일치하는 메트릭만 유지
- **drop**: 정규식과 일치하는 메트릭 제거
- **replace**: 대상 레이블의 값을 대체 값으로 변경

특수 레이블

- **name**: 메트릭 이름을 나타내는 특수 레이블

예제

예제 1. 특정 메트릭만 유지

이 설정은 `http_requests_total` 메트릭만 유지하고 나머지는 모두 제거합니다.

```
metricRelabelConfigs:
  - source_labels: [__name__]
    regex: "http_requests_total"
    action: keep
```

- **동작 예시**

  원래 스크랩된 메트릭이 다음과 같다고 가정합니다.

  ```
  http_requests_total{method="GET", status="200"} 100
  http_errors_total{method="GET", status="500"} 5
  node_cpu_seconds_total{cpu="0", mode="idle"} 1000
  ```

  위 metricRelabelConfigs를 적용하면 `http_requests_total` 메트릭만 유지되고 다른 메트릭들은 모두 제거됩니다.

  ```
  http_requests_total{method="GET", status="200"} 100
  ```

예제 2. 정규식을 사용한 메트릭 필터링

이 설정은 `node_cpu` 나 `node_memory` 로 시작하는 메트릭만 유지합니다.

```
metricRelabelConfigs:
  - source_labels: [__name__]
    regex: "node_(cpu|memory).*"
    action: keep
```

- **동작 예시**

  원래 스크랩된 메트릭이 다음과 같다고 가정합니다.

  ```
  node_cpu_seconds_total{cpu="0", mode="idle"} 1000
  ```

```
node_memory_MemTotal_bytes{} 16777216
node_disk_io_time_seconds_total{device="sda"} 100
http_requests_total{method="GET", status="200"} 100
```

위 metricRelabelConfigs를 적용하면 `node_cpu` 나 `node_memory` 로 시작하는 메트릭만 유지되고 다른 메트릭들은 모두 제거됩니다. 정규식을 사용하여 여러 메트릭 패턴을 한 번에 필터링할 수 있습니다.

```
node_cpu_seconds_total{cpu="0", mode="idle"} 1000
node_memory_MemTotal_bytes{} 16777216
```

예제 3. 레이블 이름 변경

이 설정은 `method` 레이블의 값을 `http_method` 레이블로 복사합니다.

```
metricRelabelConfigs:
  - source_labels: [method]
    target_label: http_method
    replacement: "${1}"
    action: replace
```

- **동작 예시**

  원래 스크랩된 메트릭이 다음과 같다고 가정합니다.

  ```
  http_requests_total{method="GET", path="/api", status="200"} 100
  http_requests_total{method="POST", path="/api/users", status="201"} 50
  ```

  위 metricRelabelConfigs를 적용하면 각 메트릭에 `method` 레이블의 값을 복사한 `http_method` 레이블이 추가됩니다. 원래 레이블은 유지되며, 새 레이블이 추가됩니다. `${1}` 은 소스 레이블의 값을 참조합니다.

  ```
  http_requests_total{method="GET", path="/api", status="200", http_method="GET"} 100
  http_requests_total{method="POST", path="/api/users", status="201", http_method="POST"} 50
  ```

예제 4. 여러 소스 레이블 조합

```
metricRelabelConfigs:
  - source_labels: [__name__, status]
    regex: "http_requests_total;(200|500)"
    action: keep
```

이 설정은 `http_requests_total` 메트릭 중 `status` 레이블이 `200` 또는 `500`인 메트릭만 유지합니다.

- **동작 예시**

  원래 스크랩된 메트릭이 다음과 같다고 가정합니다.

  ```
  http_requests_total{method="GET", path="/api", status="200"} 100
  http_requests_total{method="POST", path="/api/users", status="201"} 50
  http_requests_total{method="GET", path="/api/error", status="500"} 10
  http_requests_total{method="GET", path="/api/error", status="404"} 5
  ```

  위 metricRelabelConfigs를 적용하면 `http_requests_total` 메트릭 중에서 `status` 레이블이 `200` 또는 `500`인 메트릭만 유지됩니다. 여러 소스 레이블을 조합할 때는 기본적으로 `;` 구분자로 연결되며, 이를 `separator` 필드로 변경할 수 있습니다.

  ```
  http_requests_total{method="GET", path="/api", status="200"} 100
  http_requests_total{method="GET", path="/api/error", status="500"} 10
  ```

예제 5. 정적 레이블 추가

```
metricRelabelConfigs:
  - target_label: metric_src
    replacement: "whatap-open-agent"
    action: replace
```

이 설정은 모든 메트릭에 `metric_src="whatap-open-agent"` 레이블을 추가합니다. 소스 레이블을 지정하지 않으면 `replacement` 값이 직접 레이블 값으로 사용됩니다. 이 방법을 사용하여 모든 메트릭에 환경, 리전, 애플리케이션 이름 등의 정적 레이블을 추가할 수 있습니다.

- **동작 예시**

  원래 스크랩된 메트릭이 다음과 같다고 가정해 봅시다:

  ```
  http_requests_total{method="GET", path="/api", status="200"} 100
  node_cpu_seconds_total{cpu="0", mode="idle"} 1000
  ```

  위 metricRelabelConfigs를 적용하면, 모든 메트릭에 `metric_src="whatap-open-agent"` 레이블이 추가됩니다. 이 방법은 메트릭의 출처를 표시하거나, 환경(예: production, staging), 리전(예: us-east, eu-west), 또는 애플리케이션 이름 등을 표시하는 데 유용합니다.

  ```
  http_requests_total{method="GET", path="/api", status="200", metric_src="whatap-open-agent"} 100
  node_cpu_seconds_total{cpu="0", mode="idle", metric_src="whatap-open-agent"} 1000
  ```

예제. 종합적인 동작

원래 스크랩된 메트릭이 다음과 같다고 가정합니다.

```
apiserver_request_total{code="200", resource="pods", verb="GET"} 100
some_other_metric{label="value"} 50
```

위 metricRelabelConfigs를 적용하면

1. 첫 번째 룰(keep apiserver_request_total) 적용:
   - apiserver_request_total 메트릭은 유지됩니다.
   - some_other_metric 메트릭은 드롭됩니다.
2. 두 번째 룰(replace verb -> http_verb) 적용:
   - 유지된 apiserver_request_total 메트릭에 verb 레이블이 있으므로, 이 레이블의 값(GET)이 http_verb라는 새로운 레이블로 복사됩니다.

따라서 최종적으로 수집되는 메트릭은 다음과 같습니다.

```
apiserver_request_total{code="200", resource="pods", verb="GET", http_verb="GET"} 100
```

쿠버네티스 메트릭 수집 예제

다음은 쿠버네티스 API 서버에서 메트릭을 수집하는 예제입니다.

- 이 설정은 kube-system 네임스페이스에서 component=apiserver 및 provider=kubernetes 레이블을 가진 서비스를 찾아 해당 서비스의 엔드포인트에서 메트릭을 수집합니다.
- metricRelabelConfigs를 사용하여 apiserver_request_total 메트릭만 수집하고, verb 레이블을 http_verb 레이블로 변환하며, 모든 메트릭에 metric_src="whatap-open-agent" 정적 레이블을 추가하도록 지정할 수 있습니다.

쿠버네티스 메트릭 수집 예제

```yaml
# scrape_config.yaml
features:
  openAgent:
    enabled: true
    targets:
      - targetName: kube-apiserver
        type: ServiceMonitor
        namespaceSelector:
          matchNames:
            - "kube-system"
        selector:
          matchLabels:
            component: apiserver
            provider: kubernetes
        endpoints:
          - port: "https"
            path: "/metrics"
            interval: "30s"
            metricRelabelConfigs:
              - source_labels: [__name__]
                regex: "apiserver_request_total"
                action: keep
              - source_labels: [verb]
                target_label: http_verb
                replacement: "${1}"
                action: replace
            # 정적 레이블 추가
```

```
    - target_label: metric_src
      replacement: "whatap-open-agent"
      action: replace
```

WhaTap CR 설정 옵션

WhaTap CR 설정 옵션 문서는 이미지, 환경 변수, 리소스, 스케줄링, 인증, 우선순위 등을 세밀하게 지정해 에이전트 동작을 제어하는 방법을 제공합니다.

이미지 구성

```
apiVersion: monitoring.whatap.com/v2alpha1
kind: WhatapAgent
metadata:
  name: whatap
spec:
  features:
    k8sAgent:
      # 풀 이미지명
      customImageFullName: public.ecr.aws/whatap/kube_agent:1.8.8
      masterAgent:
        enabled: true
        masterAgentContainer:
          # 컨테이너별 오버라이드 가능
          customImageFullName: public.ecr.aws/whatap/kube_agent:1.8.8
      nodeAgent:
        enabled: true
        nodeAgentContainer:
          # 컨테이너별 오버라이드 가능
          customImageFullName: public.ecr.aws/whatap/kube_agent:1.8.8
        nodeHelperContainer:
          # 컨테이너별 오버라이드 가능
          customImageFullName: public.ecr.aws/whatap/kube_agent:1.8.8
    openAgent:
      enabled: true
      # 컨테이너별 오버라이드 가능
      customImageFullName: public.ecr.aws/whatap/open_agent:latest
```

1. 옵션을 컨테이너 env로 설정

쿠버네티스 에이전트는 컨테이너 env를 통해 옵션을 설정할 수 있습니다.

- 오퍼레이터는 기본 env(WHATAP_LICENSE/HOST/PORT, WHATAP_MEM_LIMIT 등)를 자동으로 추가합니다. `nodeAgent.nodeAgentContainer.envs` 에 넣은 값은 기본 env 뒤에 추가로 병합됩니다.

```
#와탭 노드 에이전트 데몬셋의 whatap-node-agent 컨테이너 옵션 설정 시
apiVersion: monitoring.whatap.com/v2alpha1
kind: WhatapAgent
metadata:
  name: whatap
spec:
  features:
    k8sAgent:
      masterAgent:
        enabled: true
      nodeAgent:
        enabled: true
        nodeAgentContainer:
          envs:
            - name: debug
              value: "true"
      gpuMonitoring:
        enabled: false
```

2. resources 설정 방법

오퍼레이터는 k8s 에이전트 설치 기본값을 포함합니다.

```
#와탭 노드 에이전트 데몬셋의 whatap-node-agent 컨테이너 옵션 설정 시
apiVersion: monitoring.whatap.com/v2alpha1
kind: WhatapAgent
metadata:
  name: whatap
spec:
  features:
    k8sAgent:
      masterAgent:
        enabled: true
        # 마스터 에이전트 컨테이너 리소스(컨테이너 단위 오버라이드)
        masterAgentContainer:
          resources:
```

```
      requests:
        cpu: "150m"
        memory: "256Mi"
      limits:
        cpu: "300m"
        memory: "512Mi"
  nodeAgent:
    enabled: true
    # 노드 에이전트 전체 기본 리소스(컴포넌트 레벨)
    resources:
      requests:
        cpu: "100m"
        memory: "300Mi"
      limits:
        cpu: "200m"
        memory: "350Mi"
    # 노드 에이전트 컨테이너 오버라이드
    nodeAgentContainer:
      resources:
        requests:
          cpu: "150m"
          memory: "350Mi"
        limits:
          cpu: "300m"
          memory: "512Mi"
    # 헬퍼 컨테이너 오버라이드
    nodeHelperContainer:
      resources:
        requests:
          cpu: "100m"
          memory: "100Mi"
        limits:
          cpu: "200m"
          memory: "350Mi"
  gpuMonitoring:
    enabled: false
```

- 컨테이너별 `resources`에 값이 있으면 해당 컨테이너에 대해 컴포넌트 기본값을 덮어씁니다.

컨테이너	requests	limits
whatap-master-agent	cpu: 100m memory: 300Mi	cpu: 200m memory: 350Mi
whatap-node-agent	request: 100m limit: 300Mi	request: 200Mi limit: 350Mi
whatap-node-helper	request: 100m limit: 100m	request:200Mi limit: 350Mi

3. tolerations, affinity, nodeSelector

tolerations

```yaml
spec:
  features:
    k8sAgent:
      masterAgent:
        enabled: true
        tolerations:
          - key: "node.kubernetes.io/not-ready"
            operator: "Exists"
            effect: "NoExecute"
            tolerationSeconds: 300
      nodeAgent:
        enabled: true
        tolerations:
          - key: "dedicated"
            operator: "Equal"
            value: "observability"
            effect: "NoSchedule"
```

affinity

마스터, 노드 에이전트는 다음 기본 tolerations를 내장하며, CR에서 설정한 값은 기본값 뒤에 병합됩니다.

- `node-role.kubernetes.io/master:NoSchedule`

- `node-role.kubernetes.io/control-plane:NoSchedule`

```
spec:
  features:
    k8sAgent:
      masterAgent:
        enabled: true
        affinity:
          nodeAffinity:
            requiredDuringSchedulingIgnoredDuringExecution:
              nodeSelectorTerms:
                - matchExpressions:
                    - key: kubernetes.io/os
                      operator: In
                      values: ["linux"]
      nodeAgent:
        enabled: true
        affinity:
          podAntiAffinity:
            preferredDuringSchedulingIgnoredDuringExecution:
              - weight: 100
                podAffinityTerm:
                  topologyKey: kubernetes.io/hostname
                  labelSelector:
                    matchLabels:
                      name: whatap-node-agent
```

nodeSelector

- 마스터 에이전트에 nodeSelector 적용

```
spec:
  features:
    k8sAgent:
      masterAgent:
        enabled: true
        nodeSelector:
          nodepool: system          # 예) system 노드풀로 고정
          kubernetes.io/os: linux   # 리눅스 노드만
```

- 노드 에이전트(DaemonSet) `nodeSelector` 적용

```
spec:
  features:
    k8sAgent:
      nodeAgent:
        enabled: true
        nodeSelector:
          nodepool: worker          # 예) 워커 노드만
          kubernetes.io/arch: amd64   # AMD64 아키텍처 노드만
```

> ⓘ 노드 에이전트는 모든 노드에 배포되는 DaemonSet입니다. `nodeSelector`를 설정하면 해당 조건의 노드만 파드가 생성됩니다.

  - GPU 사용 클러스터에서 GPU 노드에만 배포하고 싶다면 다음과 같이 사용할 수 있습니다.

```
spec:
  features:
    k8sAgent:
      nodeAgent:
        enabled: true
        nodeSelector:
          nvidia.com/gpu.present: "true"
      gpuMonitoring:
        enabled: true
```

- OpenAgent에 `nodeSelector` 적용

```
spec:
  features:
    openAgent:
      enabled: true
      nodeSelector:
        nodepool: system          # OpenAgent를 시스템 노드풀로 고정
        kubernetes.io/os: linux
      # 선택: 더 정교한 스케줄링은 affinity와 병행
      affinity:
```

```
      nodeAffinity:
        requiredDuringSchedulingIgnoredDuringExecution:
          nodeSelectorTerms:
            - matchExpressions:
                - key: kubernetes.io/hostname
                  operator: In
                  values: ["ip-10-0-0-12"]
```

4. imagePullSecret

```
spec:
  features:
    k8sAgent:
      # 전역 풀시크릿 (마스터/노드 공통 기본값)
      imagePullSecrets:
        - name: global-regcred
      masterAgent:
        enabled: true
        # 마스터 전용 오버라이드
        imagePullSecrets:
          - name: master-regcred
      nodeAgent:
        enabled: true
        # 노드 전용 오버라이드
        imagePullSecrets:
          - name: node-regcred
    openAgent:
      enabled: true
      imagePullSecrets:
        - name: openagent-regcred
```

5. PriorityClassName

```
spec:
  features:
    k8sAgent:
      masterAgent:
```

```
    enabled: true
    priorityClassName: system-cluster-critical
  nodeAgent:
    enabled: true
    priorityClassName: system-node-critical
openAgent:
  enabled: true
  priorityClassName: system-cluster-critical
```

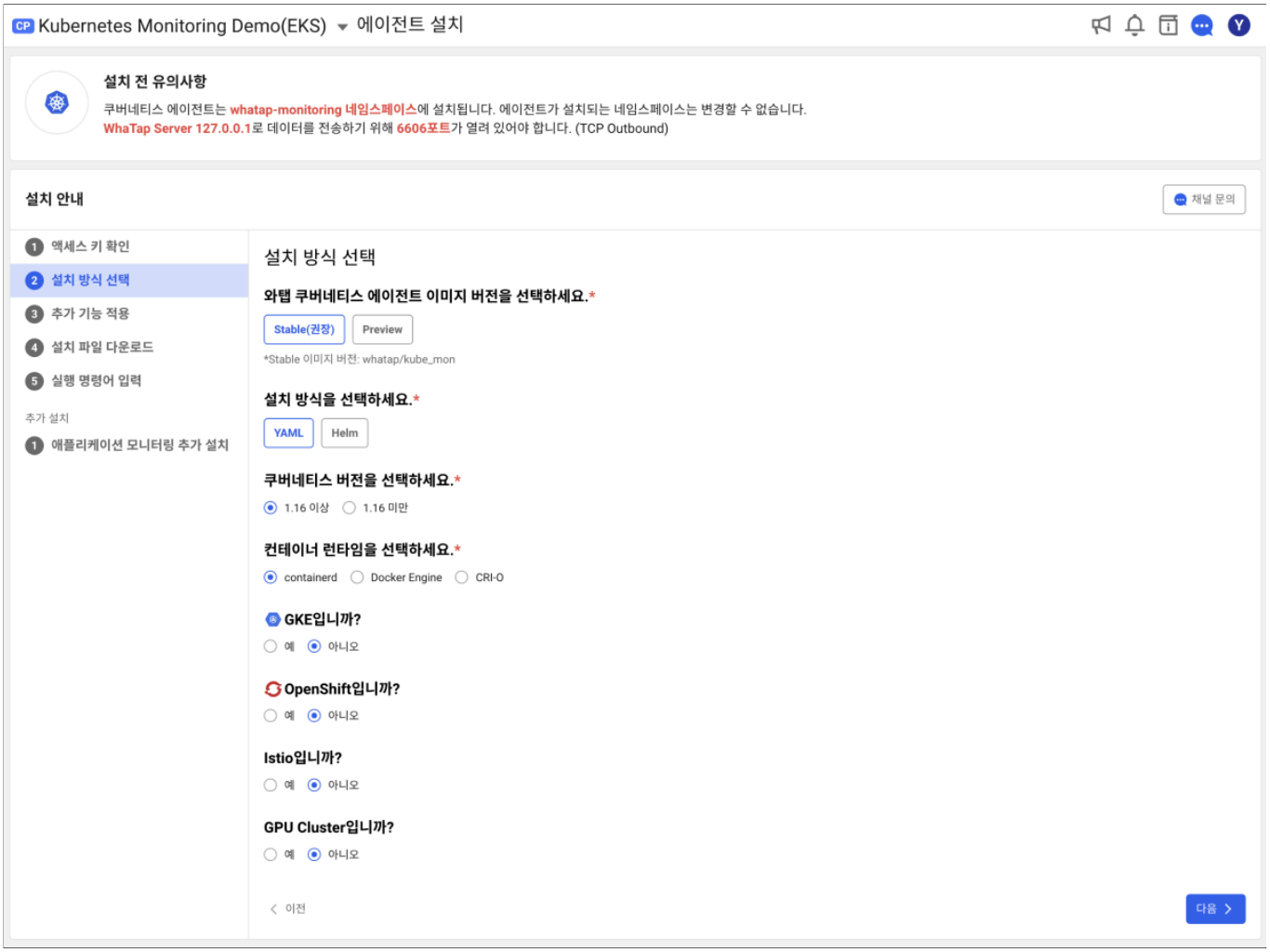
마스터 및 노드 에이전트 설치

다음과 같이 마스터 및 노드 에이전트 설치 과정을 안내합니다. **설치 안내** 화면과 동일한 과정입니다.

> EKS Fargate는 지원 예정입니다.

설치 방식 선택

액세스 키 확인 후 **설치 안내** 화면에서 **설치 방식 선택** 탭을 선택하세요. 와탭 쿠버네티스 에이전트가 지원하는 **Yaml** 파일과 **Helm** 차트 설치 방식 중 원하는 설치 방식을 선택 후 사용자의 쿠버네티스 환경에 따라 알맞은 항목을 선택하세요.



1. 와탭 쿠버네티스 에이전트 이미지 버전을 권장하는 **Stable** 버전과 **Preview** 버전 중 선택하세요.
2. **YAML** 파일과 **Helm** 차트 설치 방식 중 중 원하는 **설치 방식**을 선택하세요.
3. **1.16 이상** 버전 또는 **1.16 미만** 버전 중 해당하는 **쿠버네티스 버전**을 선택하세요.
4. **containerd**, **Docker Engine**, **CRI-O** 중 **컨테이너 런타임**을 선택하세요.
5. **사용자 쿠버네티스 환경**(GKE, OpenShift, Istio, GPU Cluster) 중 해당하는 항목이 있을 경우 선택하세요.
6. 설치 방식 관련 항목 선택을 마친 후 화면 오른쪽 아래의 **다음** 버튼을 클릭하거나 왼쪽 **추가 기능 적용** 탭을 선택해 다음 단계로 이동하세요.

와탭 쿠버네티스 에이전트 이미지 리포지토리 추가 지원

와탭 쿠버네티스 에이전트는 기존 DockerHub 외에 Public ECR을 추가로 지원합니다. 사용 환경에 따라 두 이미지 리포지토리 중 하나를 선택하여 설치할 수 있습니다.

Public ECR

- 리포지토리: `public.ecr.aws/whatap/kube_mon` 추천

ECR은 pull 제한 정책에 영향을 받지 않으므로, 빈번한 업데이트나 대규모 배포 환경에서 안정적으로 사용할 수 있어 권장합니다.

DockerHub

- 리포지토리: `whatap/kube_mon`

DockerHub는 인증되지 않은 클라이언트에 대해 pull 제한(policy)이 적용될 수 있습니다. 노드 수가 많거나 동시에 pull 요청이 발생하는 경우 rate limit에 걸려 설치가 실패할 가능성이 높습니다. 자세한 내용은 DockerHub 정책 안내 공식 문서를 참고해 주세요.

> pull 제한이 발생한 경우 다음 문서를 참고하세요.

추가 기능 적용

와탭 쿠버네티스 모니터링은 다양한 추가 기능을 제공합니다.

추가 적용한 기능이 많을 수록 수집되는 데이터의 양과 에이전트가 사용하는 리소스의 양이 증가할 수 있습니다. 설치 후 에이전트 설정을 통해 해당 기능을 활성화 또는 비활성화할 수 있습니다.

추가 기능 적용 여부를 선택 후 화면 오른쪽 아래의 **다음** 버튼을 클릭하거나 왼쪽 **설치 파일 다운로드** 탭을 선택해 다음 단계로 이동하세요.

컨트롤 플레인 모니터링 활성화

컨트롤 플레인 모니터링 활성화 및 각 기능에 대한 자세한 내용은 다음 문서를 참조하세요.

- **kube-apiserver 대시보드** 바로가기
- **kube-apiserver 메트릭스 조회** 바로가기
- **etcd 대시보드** 바로가기
- **kube-scheduler 대시보드** 바로가기

Custom Resource 모니터링 활성화

- Yaml Helm Chart

와탭 쿠버네티스 에이전트 설치 YAML 파일에서 `ClusterRole` 리소스의 `rules` 하위의 `resources` 항목에 사용하려는 `customResource` 를 추가하세요.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  creationTimestamp: "2025-01-15T01:48:26Z"
  name: whatap
  resourceVersion: "125200773"
  uid: 54b0bcdb-1f12-4779-b0cb-e8de2f6be64a
rules:
- apiGroups:
  - '*'
  resources:
  - namespaces
  - pods
  - pods/log
  - configmaps
  - services
  - endpoints
  - daemonsets
  - ingresses
  - deployments
  - nodes
  - persistentvolumes
  - persistentvolumeclaims
  - events
  - replicasets
  - roles
  - rolebindings
  - clusterroles
  - clusterrolebindings
  - jobs
  - cronjobs
  - statefulsets
  - serviceaccounts
```

```
  - configmaps
  - storageclasses
  - horizontalpodautoscalers
  - replicationcontrollers
  - customresourcedefinitions # CRD 수집 권한 추가
  - samplecrds # 모니터링할 Custom Resource의 이름 추가
  verbs:
  - get
  - list
  - watch
- apiGroups:
  - ""
  resources:
  - pods/exec
  verbs:
  - create
- apiGroups:
  - ""
  resources:
  - configmaps
  verbs:
  - '*'
- nonResourceURLs:
  - /metrics
  verbs:
  - '*'
```

와탭 쿠버네티스 에이전트를 Helm 차트 방식으로 설치할 경우 기본 values.yaml 파일에 모니터링할 Custom Resource 이름을 추가하세요. 다음과 같이 `resources` 항목에 해당 Custom Resource 이름을 추가하세요.

```
collect_custom_resource: true
clusterrole:
  extraResources:
    - "customresourcedefinitions" ##crd 수집 권한 추가
    - "samplecrds" ##수집하고자 하는 custom resource 추가
```

> ⓘ **오브젝트 매니페스트** 메뉴에서 **Custom Resource 모니터링**을 할 수 있습니다.

신규 수집 대상 ClusterRole 권한

다음 수집 기능은 `ClusterRole` 리소스의 `rules` 하위 `resources` 항목에 조회 권한이 추가되어야 동작합니다.

Resource	Feature	Description
`networkpolicies`	NetworkPolicy 이벤트 수집	`kube_network_policy` 카테고리 수집에 필요
`resourcequotas`	네임스페이스 ResourceQuota 수집	`kube_stat` 카테고리의 `quota_*` 필드 수집에 필요

- 권한이 없으면 해당 수집만 생략되고 나머지 수집은 계속됩니다.

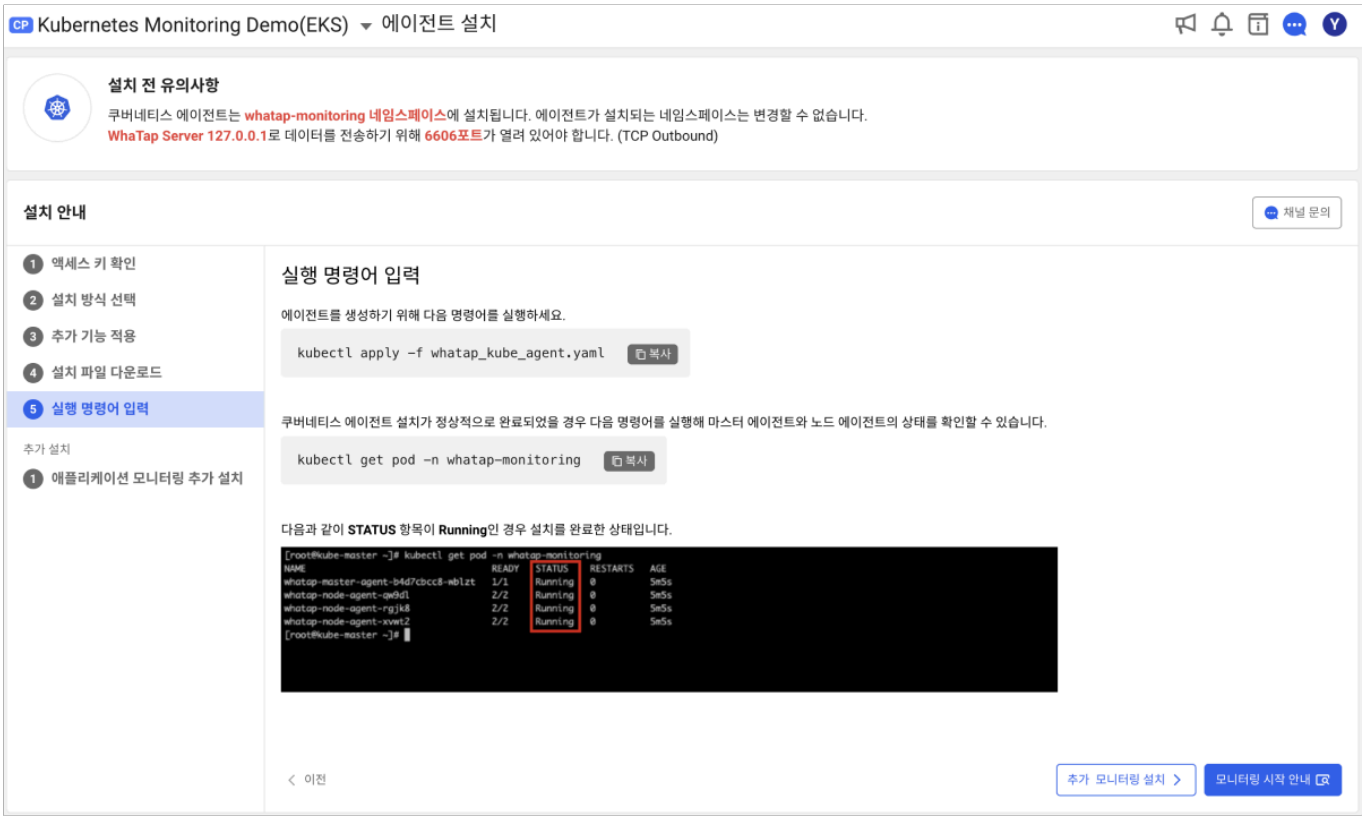
설치 파일 다운로드

설치 파일 다운로드 전 사용자 쿠버네티스 환경에 알맞은 항목이 선택되었는지 확인하세요. 선택 항목 확인 후 **설치 파일 다운로드** 버튼을 클릭해 설치에 필요한 기본 설정 파일(values.yaml)을 다운로드하세요.

다운로드 후 화면 오른쪽 아래의 **다음** 버튼을 클릭하거나 **실행 명령어 입력** 탭을 선택해 다음 단계로 이동하세요.

실행 명령어 입력

설치 방식 선택 탭에서 선택한 환경과 설치 방식 안내에 따라 실행 명령어를 입력해 설치를 진행하세요.



쿠버네티스 환경

설치 방식 선택 탭에서 **GKE** 또는 **OpenShift** 환경을 선택한 경우 다음 안내를 참조하세요. 해당하지 않을 경우 다음의 쿠버네티스 에이전트 생성 단계로 넘어가세요.

- **GKE 환경** **OpenShift 환경**

설치 방식 선택 탭에서 **Google Kubernetes Engine(GKE)** 환경을 선택한 경우 사용자 클러스터 생성 후 권한 설정을 위해 다음 명령을 실행하세요.

```
ACCOUNT=$(gcloud info --format='value(config.account)')
kubectl create clusterrolebinding owner-cluster-admin-binding \
```

```
--clusterrole cluster-admin \
--user $ACCOUNT
```

> gcloud 명령 설치에 대한 자세한 내용은 구글 클라우드 문서를 참조하세요.

설치 방식 선택 탭에서 **OpenShift** 환경을 선택한 경우 권한 설정을 위해 다음 명령을 실행하세요.

```
oc create -f whatap_openshift_scc.yaml
```

쿠버네티스 에이전트 생성

와탭 쿠버네티스 에이전트 설치 시 **Yaml** 파일 기준 설치 방식 안내와 **Helm** 차트 기준 설치 방식 안내를 다음과 같이 제공합니다.

- **Yaml** **Helm Chart**

Yaml 파일 기준 설치

설치 방식 선택 탭에서 **YAML** 파일 설치 방식을 선택한 경우 에이전트를 생성하기 위해 다음 명령어를 실행하세요.

```
kubectl apply -f whatap_kube_agent.yaml
```

> **마스터 에이전트가 사용하는 자원량**은 **사용자 환경의 노드 수**에 어느 정도 **비례**합니다. 즉 클러스터 내에 노드 수가 많을 경우 기본값으로 설정된 메모리 Limit(350MB) 이상의 자원을 사용해 OOM(Out Of Memory)이 발생할 수 있습니다. 마스터 에이전트에 할당된 메모리 Limit 수치는 에이전트 설치 YAML 파일을 통해 변경할 수 있습니다.

Helm 차트 기준 설치

> 와탭 쿠버네티스 에이전트는 기본적으로 `whatap-monitoring` 네임스페이스를 사용합니다. 해당 네임스페이스를 사용할 수 없거나 사용자가 특정 네임스페이스를 사용해야 하는 경우 다음 문서를 참조하세요.

설치 방식 선택 탭에서 **Helm** 차트 설치 방식을 선택한 경우 다음 안내를 참조하세요.

1. 다음 명령어를 통해 helm 버전을 확인하세요. 3.x 버전 이상을 지원합니다.

```
helm Version

# 출력 예시
# version.BuildInfo{Version:"v3.x", GitCommit:"xxx", GitTreeState:"clean", GoVersion:"go1.20.8"}
```

2. 와탭 Helm 리포지토리에 대한 네트워크 연결 상태를 확인하세요.

```
curl https://whatap.github.io/helm/ping
# 출력 예시
# pong
```

3. 다음 명령어를 실행해 와탭 Helm 리포지토리를 추가하세요.

```
helm repo add whatap https://whatap.github.io/helm/
helm repo update
```

4. 다음과 같이 에이전트 애플리케이션을 설치하세요.

- 에이전트를 **처음** 설치하는 사용자는 다음 명령어를 실행해 에이전트 애플리케이션을 설치하세요.

```
helm install whatap-kube-agent whatap/kube -f values.yaml
```

- **기존** 와탭 쿠버네티스 에이전트 사용자는 다음 명령어를 실행해 에이전트 애플리케이션을 설치하세요.

```
kubectl delete ns whatap-monitoring
kubectl delete clusterrole whatap
kubectl delete clusterrolebinding whatap
helm install whatap-kube-agent whatap/kube -f values.yaml
```

> ⓘ 기존 yaml 파일 방식으로 와탭 쿠버네티스 에이전트를 설치한 경우 Clean Install이 필요합니다. 기존 쿠버네티스 에이전트 삭제 후 helm 차트를 통한 설치가 가능합니다.

values.yaml 주요 설정 항목

와탭 쿠버네티스 에이전트 설정을 수정하기 위해서는 기본 설정 파일(values.yaml)을 수정해야 합니다. values.yaml 파일 내 주요 설정 항목을 다음과 같이 안내합니다.

키	타입	기본값	설명
whatap.license	String	# <license-key>	Whatap 라이센스 키
whatap.host	String	# <whatap-server-host>	Whatap 서버의 호스트 주소
whatap.port	Int	# <whatap-server-port>	Whatap 포트 번호
containerRuntime	String	"docker"	사용 중인 컨테이너 런타임. "docker", "containerd", "crio" 중 선택
daemonSet.name	String	whatap-node-agent	DaemonSet의 이름
daemonSet.label	String	whatap-node-agent	DaemonSet에 지정할 라벨
daemonSet.initContainers.nodeDebugger.enabled	Bool	true	Whatap 노드 디버거 컨테이너 활성화 여부
daemonSet.containers.nodeHelper.image	String	whatap/kube_mon	nodeHelper 컨테이너의 이미지
daemonSet.containers.nodeHelper.requests.memory	String	100Mi	nodeHelper MEMORY request
daemonSet.containers.nodeHelper.requests.cpu	String	100m	nodeHelper CPU request

키	타입	기본값	설명
daemonSet.containers.nodeHelper.limits.memory	String	350Mi	nodeHelper MEMORY limit
daemonSet.containers.nodeHelper.limits.cpu	String	200m	nodeHelper CPU limit
daemonSet.containers.nodeAgent.image	String	whatap/kube_mon	nodeAgent 컨테이너의 이미지
daemonSet.containers.nodeAgent.requests.memory	String	300Mi	nodeAgent MEMORY request
daemonSet.containers.nodeAgent.requests.cpu	String	100m	nodeAgent CPU request
daemonSet.containers.nodeAgent.limits.memory	String	350Mi	nodeAgent MEMORY limit
daemonSet.containers.nodeAgent.limits.cpu	String	200m	nodeAgent CPU limit
deployment.name	String	whatap-master-agent	Deployment의 이름
deployment.label	String	whatap-master-agent	Deployment에 지정할 라벨
deployment.replicas	Int	1	Deployment의 replica 수
deployment.containers.controlPlaneHelper.enabled	Boolean	true	whatap-control-plane-helper 컨테이너를 활성화 여부
deployment.containers.controlPlaneHelper.image	String	"whatap/kube_mon"	controlPlaneHelper 컨테이너의 이미지

키	타입	기본값	설명
deployment.containers.controlPlaneHelper.port	String	9496	controlPlaneHelper 컨테이너의 포트 번호입니다
deployment.containers.controlPlaneHelper.resources.requests.memory	String	"300Mi"	controlPlaneHelper 컨테이너 MEMORY request
deployment.containers.controlPlaneHelper.resources.requests.cpu	String	"100m"	controlPlaneHelper 컨테이너 CPU request
deployment.containers.controlPlaneHelper.resources.limits.memory	String	"350Mi"	controlPlaneHelper 컨테이너 MEMORY limit
deployment.containers.controlPlaneHelper.resources.limits.cpu	String	"200m"	controlPlaneHelper 컨테이너 CPU limit
deployment.containers.masterAgent.name	String	whatap-master-agent	masterAgent 컨테이너 이름
deployment.containers.masterAgent.image	String	"whatap/kube_mon"	masterAgent 컨테이너 이미지
deployment.containers.masterAgent.port	Int	6600	masterAgent 컨테이너 포트 번호
deployment.containers.masterAgent.resources.requests.memory	String	"300Mi"	masterAgent 컨테이너 MEMORY request
deployment.containers.masterAgent.resources.requests.cpu	String	"100m"	masterAgent 컨테이너 CPU request

키	타입	기본값	설명
`deployment.containers.masterAgent.resources.limits.memory`	String	`"350Mi"`	masterAgent 컨테이너 MEMORY limit
`deployment.containers.masterAgent.resources.limits.cpu`	String	`"200m"`	masterAgent 컨테이너 CPU request

ⓘ Helm 차트 기준 **설치 문제 해결**에 대한 안내는 다음 문서를 참조하세요.

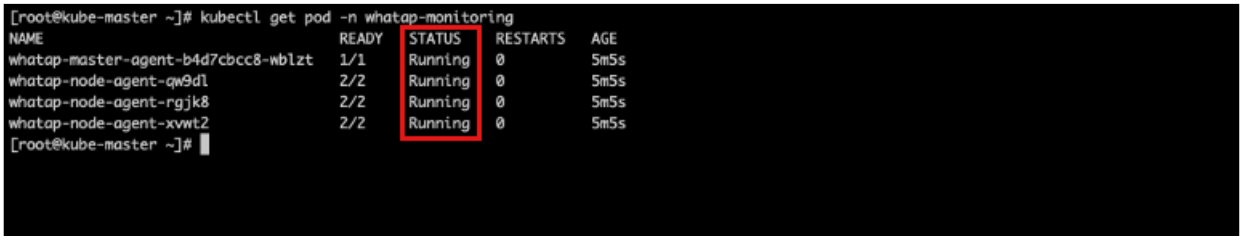
에이전트 설치 확인

쿠버네티스 에이전트 설치가 정상적으로 완료되었을 경우 다음 명령어를 실행해 마스터 에이전트와 노드 에이전트의 상태를 확인할 수 있습니다.

```
kubectl get pod -n whatap-monitoring
```

Running 상태인 경우

다음과 같이 **STATUS** 항목이 **Running**인 경우 **설치를 완료한 상태**입니다. 마스터 및 노드 에이전트의 설치를 완료하면 쿠버네티스 환경의 자원 모니터링을 시작합니다. 와탭 모니터링 서비스로 이동한 다음 **프로젝트 목록**과 **대시보드** > **컨테이너 맵** 메뉴에서 성능 정보를 수집하는 것을 확인하세요.



Running 상태가 아닌 경우

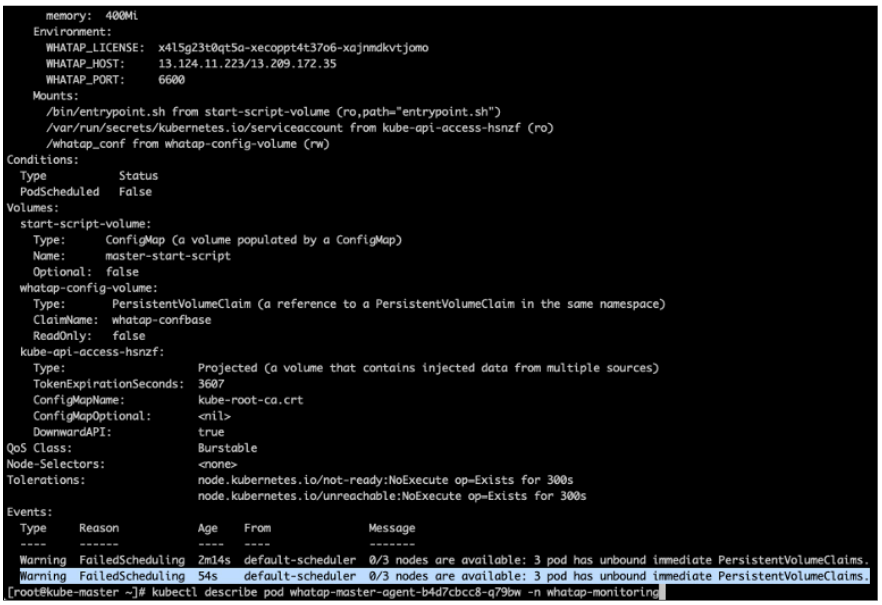
다음과 같이 **STATUS** 항목이 **Running**이 아닌 경우 설치가 제대로 이루어지지 않은 상태입니다.

```
[root@kube-master ~]# kubectl get pod -n whatap-monitoring
NAME                                    READY   STATUS    RESTARTS   AGE
whatap-master-agent-b4d7cbcc8-q79bw     0/1     Pending   0          31s
whatap-node-agent-6zcln                 2/2     Running   0          31s
whatap-node-agent-dsfxf                 2/2     Running   0          31s
whatap-node-agent-n2vnq                 2/2     Running   0          31s
[root@kube-master ~]#
```

1. 마스터 노드에서 다음 명령어를 실행해 Pod 생성의 실패 원인을 파악하세요.

   ```
   kubectl describe pod {POD_NAME} -n whatap-monitoring
   ```

2. 실행 결과 하단에 **Events:** 항목을 확인하세요.



> `kubectl` 명령어 사용 시 `-n {네임스페이스}` 옵션을 사용하면 특정 네임스페이스를 대상으로 작업을 수행합니다. 와탭 에이전트는 기본적으로 `whatap-monitoring` 이라는 네임스페이스에 설치되므로 `-n whatap-monitoring` 이라는 옵션을 추가합니다.

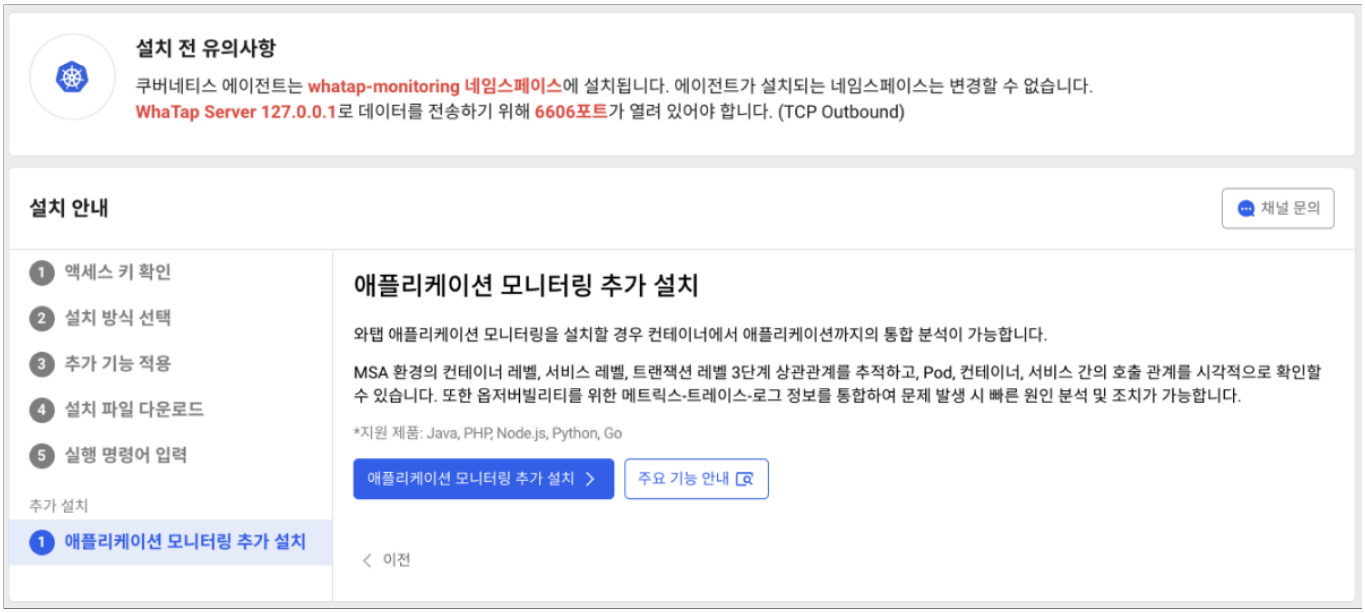
- 화면 오른쪽 아래에 **모니터링 시작 안내** 버튼 선택 시 와탭 쿠버네티스 모니터링의 주요 메뉴에 대한 안내를 확인할 수 있습니다. 각 주요 메뉴 안내 하단의 버튼 클릭 시 해당 메뉴로 이동합니다.
- 애플리케이션 에이전트를 설치를 진행하려면 화면 오른쪽 아래에 **추가 모니터링 설치** 버튼을 선택하세요. 자세한 내용은 다음 문서를 참조하세요.
- 에이전트를 최신 버전으로 업데이트하려면 다음 문서를 참조하세요.

애플리케이션 에이전트 설치

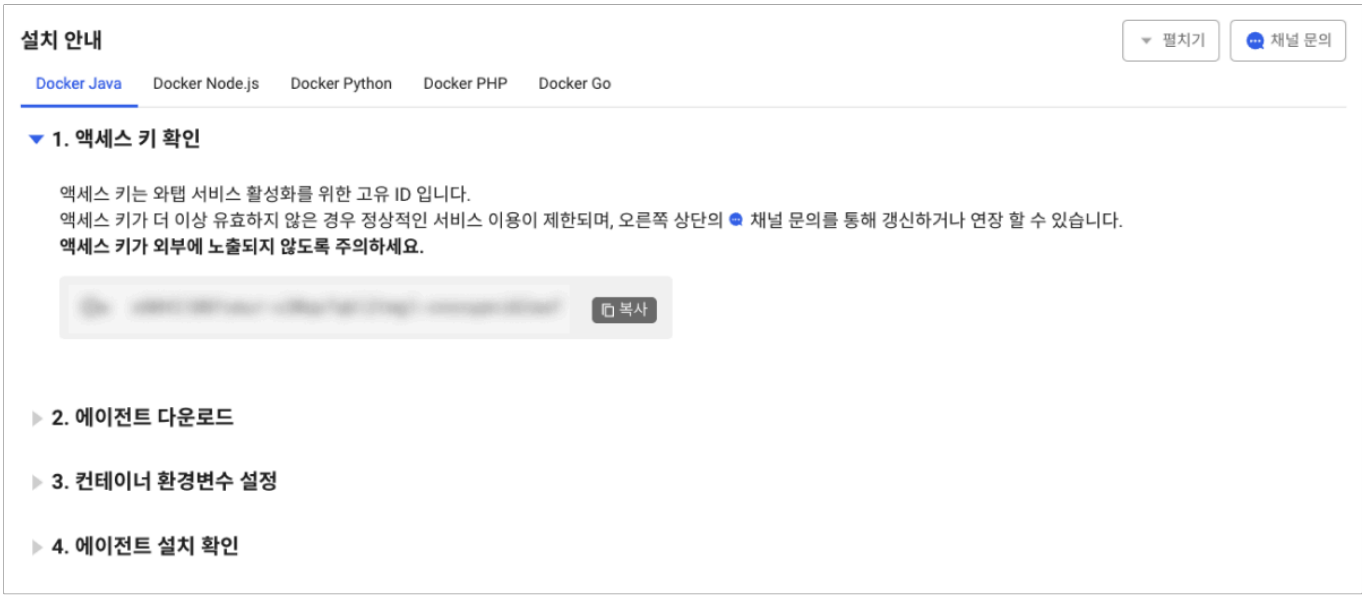
컨테이너 내 Java, Python, Node.js 등의 애플리케이션을 모니터링하기 위한 에이전트 설치 단계입니다.

마이크로 애플리케이션 에이전트의 설치는 와탭의 애플리케이션 모니터링 에이전트 설치 절차와 동일합니다. 다만 컨테이너화(Containerization) 작업이 추가되는 것이 다릅니다.

애플리케이션 에이전트를 설치를 진행하려면 **관리** > **에이전트 설치** 메뉴의 화면 왼쪽 목록 **추가 설치** 아래에 **애플리케이션 모니터링 추가 설치** 탭을 선택 후 하단의 **애플리케이션 모니터링 추가 설치** 버튼을 클릭하세요.



액세스 키 확인



설치 안내 페이지의 **액세스 키 확인** 섹션에서 **프로젝트 액세스 키 발급받기** 버튼을 선택하세요. 이미 발급 받은 경우 액세스 키가 표시됩니다.

> **액세스 키**는 다른 프로젝트와 중복 사용할 수 없으며 **클러스터 프로젝트** 인증키와 다릅니다.

애플리케이션 에이전트 컨테이너화

다음 문서를 참조하여 애플리케이션 에이전트 컨테이너화를 진행하세요.

Docker Java 설치

컨테이너 내 Java 애플리케이션을 모니터링하기 위한 에이전트 설치 단계입니다.

Docker Node.js 설치

컨테이너 내 Node.js 애플리케이션을 모니터링하기 위한 에이전트 설치 단계입니다.

Docker Python 설치

컨테이너 내 Python 애플리케이션을 모니터링하기 위한 에이전트 설치 단계입니다.

Docker PHP 설치

컨테이너 내 PHP 애플리케이션을 모니터링하기 위한 에이전트 설치 단계입니다.

Docker Go 설치

컨테이너 내 Go 애플리케이션을 모니터링하기 위한 에이전트 설치 단계입니다.

Docker Java 설치

관리 > **에이전트 설치** > 하단의 **애플리케이션 설치** 클릭 > 설치 안내 중 **Docker Java** 탭 선택

Docker 컨테이너 기반으로 실행하는 Java 애플리케이션의 JVM Option에 에이전트 적용을 위한 설정을 추가하고 컨테이너 이미지를 패키징하는 과정을 다음과 같이 안내합니다.

> ⓘ
> - EKS Fargate는 지원 예정입니다.
> - Java 애플리케이션 연동 과정에 대한 이해를 돕고자 Git 예시 코드를 제공합니다. 다음 문서를 참조하세요.

에이전트 다운로드

- **Docker Install** **직접 설치**

와탭 애플리케이션 에이전트 설치를 위해 whatap.conf 파일을 생성하세요. `{YOUR_PROJECT_ROOT}` 는 에이전트를 설치하려는 프로젝트의 최상위 경로로 변경하세요.

```
cat >{YOUR_PROJECT_ROOT}/whatap.conf <<EOL
whatap.server.host={proxyServer}
EOL
```

쿠버네티스 내부 애플리케이션 에이전트 옵션 설정

쿠버네티스 내부 애플리케이션 에이전트에 필요한 설정을 whatap.conf 파일에 작성하거나 또는 에이전트 배포 시 컨테이너 `env` 필드를 통해 설정할 수 있습니다.

- whatap.conf 파일 내 옵션이 컨테이너 `env` 필드에 설정된 옵션보다 높은 우선순위를 가집니다.
- whatap.conf 또는 컨테이너 `env` 에 설정된 값이 없는 경우 기본값이 사용됩니다.
- `license` 및 `whatap.server.host` 옵션의 경우 **컨테이너 환경변수** 설정을 활용하세요. 자세한 내용은 다음 문서를 확인하세요.

설정 파일의 내용이 변경되거나 중요한 설정 항목이 소실될 경우 정상적으로 동작하지 않을 가능성이 있기에 쿠버네티스 환경 내 **컨테이너 환경변수** 설정을 권장합니다. 그외 에이전트 기능 제어 옵션에 대한 자세한 내용은 다음 문서를 참조하세요.

쿠버네티스 환경 내 사용 불가 옵션

- whatap.name : 수집 서버가 에이전트를 식별하는 고유한 이름입니다. 해당 이름은 에이전트가 실행 중인 오브젝트 정보를 기반으로 생성됩니다. 사용자 임의 지정 시 에이전트 식별에 문제가 생길 수 있습니다.
- whatap.onode : 기본값으로 쿠버네티스 클러스터 노드 이름이 설정됩니다. 에이전트가 속한 노드 정보를 제공받기 위해 사용됩니다. 사용자 임의 지정 시 정확한 노드 식별이 어려워질 수 있습니다.

애플리케이션의 whatap.server.host 설정 확인

일부 컨테이너 이미지에서는 whatap.server.host 와 같이 . 이 포함된 환경변수를 직접 설정할 수 없습니다. 이로 인해 애플리케이션 에이전트가 정상적으로 기동되지 않는 경우, whatap.conf 파일을 직접 수정하여 whatap.server.host 값을 설정해야 합니다. 애플리케이션의 whatap.server.host 설정 확인 방법은 다음과 같습니다.

1. 와탭 애플리케이션 컨테이너에 진입하세요.
2. 다음 명령어를 실행하여 whatap.server.host 값이 포함되어 있는지 확인하세요.

```
$ env | grep whatap.server.host
```

값이 출력되지 않을 경우 해당 컨테이너 이미지가 . 이 포함된 환경변수 설정을 지원하지 않는 것입니다. 이 경우 환경변수가 아닌 설정 파일을 활용해야 합니다. whatap.conf 파일을 열고 다음과 같이 whatap.server.host 값을 수동으로 설정하세요.

```
whatap.server.host={와탭 수집 서버 IP}
```

- 최신 버전 에이전트를 이미지에 포함 **지정한 버전 에이전트를 이미지에 포함**

최신 버전 에이전트를 애플리케이션 이미지에 포함시킬 수 있도록 안내합니다.

1. 아래 표를 참고하여 각 kube_mon 버전에 포함된 Java APM 에이전트 버전을 확인하세요.

2025년 7월 2일 기준 Java APM 버전 정보

whatap/kube_mon 버전	Java APM 버전
latest	2.2.61
1.8.7	2.2.61
1.8.6	2.2.55
1.8.5	2.2.52
1.8.4	2.2.50
1.8.3	2.2.50
1.8.2	2.2.48
1.8.1	2.2.47
1.8.0	2.2.42
1.7.16	2.2.41
1.7.15	2.2.39
1.7.14	2.2.39
1.7.13	2.2.39
1.7.12	2.2.38
1.7.11	2.2.38
1.7.10	2.2.38

whatap/kube_mon 버전	Java APM 버전
1.7.9	2.2.38
1.7.8	2.2.36
1.7.7	2.2.35
1.7.6	2.2.35
1.7.5	2.2.34
1.7.4	2.2.33
1.7.3	2.2.33
1.7.2	2.2.33
1.7.1	2.2.32
1.7.0	2.2.32
1.6.1	2.2.32
1.6.0	2.2.31
1.5.9	2.2.31
1.5.8	2.2.30
1.5.7	2.2.28
1.5.6	2.2.27

whatap/kube_mon 버전	Java APM 버전
1.5.5	2.2.27
1.5.4	2.2.26
1.5.3	2.2.26
1.5.2	2.2.26
1.5.1	2.2.26
1.5.0	2.2.26
1.4.9	2.2.25

2. 애플리케이션 이미지에 와탭 에이전트를 포함시키려면 먼저 Dockrfile 최종 이미지 빌드 단계에서 다음 내용을 추가하세요. `{YOUR_PROJECT_ROOT}` 를 에이전트를 설치하려는 프로젝트의 최상위 경로로 변경하세요.

```
# 와탭 에이전트용 디렉터리를 사용자 컨테이너에 생성
RUN mkdir -p /whatap

# 와탭 Java 에이전트를 사용자 컨테이너에 복사
COPY --from=whatap/kube_mon /data/agent/micro/whatap.agent.kube.jar /whatap

# 생성한 whatap.conf 설정 파일을 사용자 컨테이너에 복사
COPY {YOUR_PROJECT_ROOT}/whatap.conf /whatap/
```

3. Dockerfile의 시작 명령어에 다음 내용을 추가하세요.

```
-javaagent:/whatap/whatap.agent.kube.jar -Dwhatap.micro.enabled=true
```

다음 Dockerfile의 정의 예시를 참고하세요.

Dockerfile

```
FROM openjdk:8-jdk-slim
RUN mkdir -p /app && mkdir /whatap
WORKDIR /app
COPY --from=whatap/kube_mon /data/agent/micro/whatap.agent.kube.jar /whatap
COPY ./whatap.conf /whatap
COPY ./target/myApp.jar /app/
CMD ["java","-javaagent:/whatap/whatap.agent.kube.jar","-Dwhatap.micro.enabled=true","-jar","/app/myApp.jar"]
EXPOSE 8080
```

4. Docker를 빌드하세요. `{YOUR_DOCKERFILE_DIR}` 를 Dockerfile이 위치한 경로로 변경하세요.

```
cd {YOUR_DOCKERFILE_DIR}
docker build -t {IMAGE_NAME} .
```

지정한 버전 에이전트를 애플리케이션 이미지에 포함시킬 수 있도록 안내합니다.

1. 애플리케이션 이미지에 와탭 에이전트를 포함시키려면 먼저 Dockrfile 최종 이미지 빌드 단계에서 다음 내용을 추가하세요. `{YOUR_PROJECT_ROOT}` 를 에이전트를 설치하려는 프로젝트의 최상위 경로로 변경하세요.

```
# 와탭 에이전트용 디렉터리를 사용자 컨테이너에 생성
RUN mkdir -p /whatap

# 와탭 Java 에이전트를 사용자 컨테이너에 복사
COPY --from=whatap/kube_mon /data/agent/micro/whatap.agent-*.jar /whatap

# 생성한 whatap.conf 설정 파일을 사용자 컨테이너에 복사
COPY {YOUR_PROJECT_ROOT}/whatap.conf /whatap/
```

2. 현재 와탭 에이전트 버전을 확인하세요.

```
docker run whatap/kube_mon ls /data/agent/micro | grep -E 'whatap\.agent-(.*?\.)(.*?\.)(.*?\.)jar' | sort | tail -1

# 출력 예시
## whatap.agent-X.Y.Z.jar
```

3. Dockerfile의 시작 명령어에 다음 내용을 추가하세요. **X.Y.Z** 부분에 앞서 확인한 에이전트 버전을 입력하세요.

```
-javaagent:/whatap/whatap.agent-X.Y.Z.jar -Dwhatap.micro.enabled=true
```

다음 Dockerfile의 정의 예시를 참조하세요.

Dockerfile

```
FROM openjdk:8-jdk-slim
RUN mkdir -p /app && mkdir /whatap
WORKDIR /app
COPY --from=whatap/kube_mon /data/agent/micro/whatap.agent-*.jar /whatap
COPY ./whatap.conf /whatap
COPY ./target/myApp.jar /app/
CMD ["java","-javaagent:/whatap/whatap.agent-X.Y.Z.jar","-Dwhatap.micro.enabled=true","-jar","/app/myApp.jar"]
EXPOSE 8080
```

4. Docker를 빌드하세요. `{YOUR_DOCKERFILE_DIR}` 를 Dockerfile이 위치한 경로로 변경하세요.

```
cd {YOUR_DOCKERFILE_DIR}
docker build -t {IMAGE_NAME} .
```

1. 설치 파일(whatap.agent.java.tar.gz)을 다운로드하려면 **다운로드** 버튼을 선택하세요.
2. Dockerfile을 작성할 수 있는 서버에 업로드한 다음 압축을 해제하세요.

> ⓘ 설치할 서버에 직접 다운로드하려면 다음 명령어를 실행하세요.
>
> ```
> wget https://api.whatap.io/agent/whatap.agent.java.tar.gz
> ```

3. 압축 해제 시 whatap 디렉터리가 생성됩니다. whatap 디렉터리 내부의 whatap.conf 파일에서 설정 내용을 확인 후 다음과 같이 수정하세요.

```
license={licenseKey}
whatap.server.host={proxyServer}
```

> **쿠버네티스 내부 애플리케이션 에이전트 옵션 설정**
>
> 쿠버네티스 내부 애플리케이션 에이전트에 필요한 설정을 whatap.conf 파일에 작성하거나 또는 에이전트 배포 시 컨테이너 `env` 필드를 통해 설정할 수 있습니다.
>
> - whatap.conf 파일 내 옵션이 컨테이너 `env` 필드에 설정된 옵션보다 높은 우선순위를 가집니다.
> - whatap.conf 또는 컨테이너 `env` 에 설정된 값이 없는 경우 기본값이 사용됩니다.
> - `license` 및 `whatap.server.host` 옵션의 경우 **컨테이너 환경변수** 설정을 활용하세요. 자세한 내용은 다음 문서를 확인하세요.
>
> 설정 파일의 내용이 변경되거나 중요한 설정 항목이 소실될 경우 정상적으로 동작하지 않을 가능성이 있기에 쿠버네티스 환경 내 **컨테이너 환경변수** 설정을 권장합니다. 그외 에이전트 기능 제어 옵션에 대한 자세한 내용은 다음 문서를 참조하세요.

> **쿠버네티스 환경 내 사용 불가 옵션**
>
> - `whatap.name` : 수집 서버가 에이전트를 식별하는 고유한 이름입니다. 해당 이름은 에이전트가 실행 중인 오브젝트 정보를 기반으로 생성됩니다. 사용자 임의 지정 시 에이전트 식별에 문제가 생길 수 있습니다.
> - `whatap.onode` : 기본값으로 쿠버네티스 클러스터 노드 이름이 설정됩니다. 에이전트가 속한 노드 정보를 제공받기 위해 사용됩니다. 사용자 임의 지정 시 정확한 노드 식별이 어려워질 수 있습니다.

4. Dockerfile의 시작 명령어에 다음 내용을 추가하세요. whatap 디렉터리 내부에서 에이전트 파일을 확인 후 **X.Y.Z** 부분에 해당 버전을 입력하세요.

```
-javaagent:/whatap/whatap.agent-X.Y.Z.jar -Dwhatap.micro.enabled=true
```

다음 Dockerfile의 정의 예시를 참조하세요.

Dockerfile

```
FROM openjdk:8-jdk-alpine
RUN mkdir -p /app && mkdir /whatap
WORKDIR /app
COPY ./whatap.agent-X.Y.Z.jar /whatap/
COPY ./whatap.conf /whatap/
```

```
COPY ./target/myApp.jar myApp.jar
COPY ./paramkey.txt /whatap/
CMD ["java","-javaagent:/whatap/whatap.agent-X.Y.Z.jar","-Dwhatap.micro.enabled=true","-jar","/app/myApp.jar"]
EXPOSE 8080
```

- whatap-virtual-X.Y.Z.jar: 모니터링 대상(샘플 애플리케이션)
- whatap.agent-X.Y.Z.jar: 와탭 에이전트
- 에이전트의 파일명이 whatap.agent-1.2.3.jar이라면 **X.Y.Z** 부분을 **1.2.3**으로 변경하세요.
- JVM 옵션 추가에 대한 자세한 내용은 다음 문서를 참조하세요.

애플리케이션의 `whatap.server.host` 설정 확인

일부 컨테이너 이미지에서는 `whatap.server.host`와 같이 `.`이 포함된 환경변수를 직접 설정할 수 없습니다. 이로 인해 애플리케이션 에이전트가 정상적으로 기동되지 않는 경우, whatap.conf 파일을 직접 수정하여 `whatap.server.host` 값을 설정해야 합니다. 애플리케이션의 `whatap.server.host` 설정 확인 방법은 다음과 같습니다.

1. 와탭 애플리케이션 컨테이너에 진입하세요.
2. 다음 명령어를 실행하여 `whatap.server.host` 값이 포함되어 있는지 확인하세요.

```
$ env | grep whatap.server.host
```

값이 출력되지 않을 경우 해당 컨테이너 이미지가 `.`이 포함된 환경변수 설정을 지원하지 않는 것입니다. 이 경우 환경변수가 아닌 설정 파일을 활용해야 합니다. whatap.conf 파일을 열고 다음과 같이 `whatap.server.host` 값을 수동으로 설정하세요.

```
whatap.server.host={와탭 수집 서버 IP}
```

Java 17 버전 이상의 경우 reflection 관련 `--add-opens=java.base/java.lang=ALL-UNNAMED` 옵션을 추가하세요.

보안키 설정하기

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

- Java 에이전트 2.2.2 버전 이상

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

  security.conf

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

- Java 에이전트 2.2.2 버전 미만

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

  paramkey.txt

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

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

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

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

- SQL 파라미터 정보 기록과 관련한 에이전트 설정은 다음 문서를 참조하세요.
- HTTP 파라미터 정보 기록과 관련한 에이전트 설정은 다음 문서를 참조하세요.

컨테이너 환경변수 설정

Docker 빌드 후 쿠버네티스 환경 내 **컨테이너 환경변수**를 설정하세요. 애플리케이션 배포 yaml 파일에 다음 내용을 추가하세요.

```
env:
- name: NODE_IP
  valueFrom: {fieldRef: {fieldPath: status.hostIP}}
- name: NODE_NAME
  valueFrom: {fieldRef: {fieldPath: spec.nodeName}}
- name: POD_NAME
  valueFrom: {fieldRef: {fieldPath: metadata.name}}
- name: OKIND
  value: {YOUR_OKIND_NAME}
- name: license
  value: {licenseKey}
- name: whatap.server.host
  value: {proxyServer}
- name: whatap.micro.enabled
  value: "true"
```

다음 예시를 참조하세요.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: #DeploymentName
  labels:
    app: #AppLabel
spec:
  replicas: 3
  selector:
    matchLabels:
      app: #AppLabel
  template:
    metadata:
      labels:
        app: #AppLabel
    spec:
```

```
containers:
- name: #ContainerName
  image: nginx
  ports:
  - containerPort: 80
  env:
  - name: NODE_IP
    valueFrom: {fieldRef: {fieldPath: "status.hostIP"}}
  - name: NODE_NAME
    valueFrom: {fieldRef: {fieldPath: "spec.nodeName"}}
  - name: POD_NAME
    valueFrom: {fieldRef: {fieldPath: "metadata.name "}}
  - name: OKIND
    value: #DeploymentName
  - name: license
    value: #licenseKey
  - name: whatap.server.host
    value: #proxyServer
  - name: whatap.micro.enabled
    value: "true"
```

환경변수 역할

- `NODE_IP` : 현재 Pod가 호스팅된 노드(Node)의 IP 주소를 수집합니다.
- `NODE_NAME` : 현재 Pod가 실행 중인 노드의 이름을 수집합니다.
- `POD_NAME` : 현재 Pod의 이름을 수집합니다.
- `OKIND` (선택사항): Pod에 해당하는 애플리케이션들을 그룹화합니다. 디플로이먼트(Deployment) 이름으로 설정하면 해당하는 Pod들을 하나로 그룹화합니다.
- `license` : 에이전트 인증 확인을 위한 키입니다.
- `whatap.server.host` : 와탭 수집 서버 호스트 IP입니다.
- `whatap.micro.enabled` : 컨테이너와의 연동을 활성화합니다.

에이전트 설치 확인

- 에이전트를 제대로 설치했는지 확인하려면 **대시보드** > **애플리케이션 서비스 대시보드** 메뉴로 이동하세요.
- 다운로드한 파일을 직접 설치한 다음 **대시보드** 메뉴에서 에이전트를 확인할 수 없다면 다음 사항을 확인하세요.
  - 컨테이너에서 `ps -ef | grep whatap` 명령어를 실행해 에이전트 옵션을 적용했는지 확인하세요.
  - 컨테이너의 /whatap/logs 경로의 내용을 확인하세요. 에이전트 로그는 logs/ `{whatap 설정파일명}` -yyyymmdd.log 형식의 파일명으로 출력됩니다.

Docker Node.js 설치

관리 > **에이전트 설치** > 하단의 **애플리케이션 설치** 클릭 > 설치 안내 중 **Docker Node.js** 탭 선택

Docker 컨테이너 기반으로 실행하는 Node.js 애플리케이션에 와탭 모니터링 에이전트를 적용하고 컨테이너 이미지를 패키징하는 과정을 다음과 같이 안내합니다.

- EKS Fargate는 지원 예정입니다.
- Node.js 애플리케이션 연동 과정에 대한 이해를 돕고자 Git 예시 코드를 제공합니다. 다음 문서를 참조하세요.

에이전트 다운로드

다음 명령어를 통해 whatap을 설치하세요.

```
npm install --save whatap
```

업데이트를 진행하려면 다음 명령어를 실행하세요.

```
npm update whatap
```

에이전트 설정

애플리케이션 메인 모듈의 첫 줄에 다음 코드를 추가하세요.

Example

```
const WhatapAgent = require('whatap').NodeAgent;
WhatapAgent.NodeAgent;
```

ECMAScript(ES) 활용 시 다음 코드를 추가하세요.

```
import WhatapAgent from 'whatap';
```

```
WhatapAgent.NodeAgent;
```

컨테이너 환경변수 설정

Docker 빌드 후 쿠버네티스 환경 내 **컨테이너 환경변수**를 설정하세요. 애플리케이션 배포 yaml 파일에 다음 내용을 추가하세요.

```
env:
- name: NODE_IP
  valueFrom: {fieldRef: {fieldPath: status.hostIP}}
- name: NODE_NAME
  valueFrom: {fieldRef: {fieldPath: spec.nodeName}}
- name: POD_NAME
  valueFrom: {fieldRef: {fieldPath: metadata.name}}
- name: WHATAP_OKIND
  value: {YOUR_OKIND_NAME}
- name: WHATAP_LICENSE
  value: {licenseKey}
- name: WHATAP_SERVER_HOST
  value: {proxyServer}
- name: WHATAP_MICRO_ENABLED
  value: "true"
```

다음 예시를 참조하세요.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: #DeploymentName
  labels:
    app: #AppLabel
spec:
  selector:
    matchLabels:
      app: #AppLabel
  replicas: 3
  template:
    metadata:
      labels:
```

```
    app: #AppLabel
spec:
  containers:
  - name: #ContainerName
    image: nginx
    ports:
    - containerPort: 80
    env:
    - name: NODE_IP
      valueFrom: {fieldRef: {fieldPath: status.hostIP}}
    - name: NODE_NAME
      valueFrom: {fieldRef: {fieldPath: spec.nodeName}}
    - name: POD_NAME
      valueFrom: {fieldRef: {fieldPath: metadata.name}}
    - name: WHATAP_OKIND
      value: #DeploymentName
    - name: WHATAP_LICENSE
      value: {licenseKey}
    - name: WHATAP_SERVER_HOST
      value: {proxyServer}
    - name: WHATAP_MICRO_ENABLED
      value: "true"
```

환경변수 역할

- `NODE_IP` : 현재 Pod가 호스팅된 노드(Node)의 IP 주소를 수집합니다.
- `NODE_NAME` : 현재 Pod가 실행 중인 노드의 이름을 수집합니다.
- `POD_NAME` : 현재 Pod의 이름을 수집합니다.
- `WHATAP_OKIND` (선택사항): Pod에 해당하는 애플리케이션들을 그룹화합니다. 디플로이먼트(Deployment) 이름으로 설정하면 해당하는 Pod들을 하나로 그룹화합니다.
- `license` : 에이전트 인증 확인을 위한 키입니다.
- `whatap_server_host` : 와탭 수집 서버 호스트 IP입니다.
- `whatap_micro_enabled` : 컨테이너와의 연동을 활성화합니다.

에이전트 설치 확인

- 에이전트를 제대로 설치했는지 확인하려면 **대시보드** > **애플리케이션 서비스 대시보드** 메뉴로 이동하세요.
- 다운로드한 파일을 직접 설치한 다음 **대시보드** 메뉴에서 에이전트를 확인할 수 없다면 다음 사항을 확인하세요.
  - 컨테이너에서 `ps -ef | grep whatap` 명령어를 실행해 에이전트 옵션을 적용했는지 확인하세요.
  - 컨테이너의 /whatap/logs 경로의 내용을 확인하세요.

Docker Python 설치

관리 > **에이전트 설치** > 하단의 **애플리케이션 설치** 클릭 > 설치 안내 중 **Docker Python** 탭 선택

Docker 컨테이너 기반으로 실행하는 Python 애플리케이션에 와탭 모니터링 에이전트를 적용하고 컨테이너 이미지를 패키징하는 과정을 다음과 같이 안내합니다. 쿠버네티스 애플리케이션을 배포하기 위해 도커 이미지가 필요합니다. 다음 과정을 통해 whatap-python 패키지가 설치된 도커 이미지를 생성합니다.

- EKS Fargate는 지원 예정입니다.
- Python 애플리케이션 연동 과정에 대한 이해를 돕고자 Git 예시 코드를 제공합니다. 다음 문서를 참조하세요.

에이전트 다운로드

Python 애플리케이션 도커 이미지 빌드 시 whatap-python 패키지를 설치하세요.

```
RUN pip3 install --upgrade whatap-python
```

다음 Dockerfile의 정의 예시를 참조하세요.

python ver 3.10

```
# python3.10을 도커환경에 설치합니다.
FROM python:3.10

# 작업 디렉터리를 /app 으로 설정합니다.
WORKDIR /app

# 현재 디렉터리의 모든 파일과 폴더를 컨테이너 내의 /app 디렉터리에 복사합니다.
ADD . /app/

# Python에서는 pip를 이용하여 외부 라이브러리를 설치할 수 있습니다.
# 와탭 Python 에이전트를 도커 이미지 빌드시 설치합니다.
RUN pip3 install --upgrade whatap-python
```

에이전트 설정 및 실행

애플리케이션 실행 전 Python 에이전트 작업 디렉터리를 정하고 기본 설정 파일을 생성하세요. 설정 파일(entrypoint.sh)을 통해 에이전트에 인증 정보를 전달하고 로그 추적 여부 등을 설정합니다.

설정 파일(entrypoint.sh) 완성 예시

entrypoint.sh

```
#!/bin/bash

# 컨테이너의 작업 디렉터리를 와탭 경로로 설정하세요. 해당 경로에 에이전트 로그 및 설정 파일을 생성합니다.
export WHATAP_HOME=${PWD}

# 권한 오류 발생 시 다음 주석을 제거 후 진행하세요.
#chmod -R 777 $WHATAP_HOME

# 다음은 에이전트 구성에 필수적인 설정 값입니다. 애플리케이션 배포 yaml 파일을 통해 설정합니다.
whatap-setting-config \
--host $whatap_server_host \
--license $license \
--app_name $app_name \
--app_process_name $app_process_name

# 다음 주석은 에이전트 그룹화 및 로그 수집 활성화 등의 추가 설정입니다. 필요한 경우에만 사용하세요.

# 에이전트 그룹화
#echo "whatap.okind=$whatap.okind" >> whatap.conf

# 로그 수집 활성화
#echo "logsink_enabled=true" >> whatap.conf
#echo "logsink_trace_enabled=true" >> whatap.conf
#echo "trace_logging_enabled=true" >> whatap.conf


# 다음과 같이 whatap-start-agent를 애플리케이션 시작 명령어 앞에 추가해 에이전트를 실행하세요.
whatap-start-agent uvicorn server:app --host 0.0.0.0 --port 8000
```

1. 컨테이너 작업 디렉터리를 `WHATAP_HOME` 환경변수로 설정하세요. 해당 경로에 에이전트 로그 및 설정 파일을 생성합니다.

```
export WHATAP_HOME=${PWD}
```

2. `WHATAP_HOME` 으로 설정한 경로에 whatap.conf 파일을 생성하고 설정하도록 다음 명령어를 실행하세요. `$` 로 표시된 변수는 에이전트 구성에 필수적인 설정 값으로 애플리케이션 배포 yaml 파일을 통해 설정합니다.

```
whatap-setting-config \
--host $whatap_server_host \
--license $license \
--app_name $app_name \
--app_process_name $app_process_name
```

3. 애플리케이션 시작 명령어 앞에 다음과 같이 `whatap-start-agent` 를 추가해 에이전트를 실행하세요.

```
whatap-start-agent uvicorn server:app --host 0.0.0.0 --port 8000
```

> 권한 문제가 발생하는 경우 다음과 같이 `$WHATAP_HOME` 에 권한을 부여하세요.
>
> ```
> echo `sudo chmod -R 777 $WHATAP_HOME`
> ```

> - yaml 작성 예제는 다음 컨테이너 환경변수 설정을 참고하세요.
> - 라이센스(`license`)는 와탭의 사용자 인증 정보입니다. **외부에 노출해서는 안됩니다**.

스크립트 실행 예시

다음 Dockerfile 정의 예시를 참조하세요. entrypoint.sh 스크립트를 실행하는 완성된 예시입니다.

```
FROM python:3.10
WORKDIR /app
ADD . /app/
RUN pip3 install --upgrade whatap-python
```

```
# entrypoint.sh 스크립트를 컨테이너에서 실행할 수 있도록 권한을 부여합니다.
RUN chmod +x ./entrypoint.sh

# 컨테이너 생성시 entrypoint.sh 스크립트를 실행합니다.
CMD ["./entrypoint.sh"]
```

추가 설정

다음 설정은 선택사항으로 필요한 경우에만 사용하세요. 설정 파일(entrypoint.sh)에 다음 추가 설정 외에도 로그 및 트랜잭션 관련 설정이 가능합니다. 기타 추가 설정의 경우 다음 문서를 참조하세요.

- **에이전트 그룹화**

```
echo "whatap.okind=$whatap.okind" >> whatap.conf
```

- **로그 수집 활성화**

```
echo "logsink_enabled=true" >> whatap.conf
echo "logsink_trace_enabled=true" >> whatap.conf
echo "trace_logging_enabled=true" >> whatap.conf
```

컨테이너 환경변수 설정

Docker 빌드 후 쿠버네티스 환경 내 **컨테이너 환경변수**를 설정하세요. 애플리케이션 배포 yaml 파일에 다음 내용을 추가하세요.

```
env:
- name: license
  value: XXXXXXXXXXXXXX-XXXXXXXXXXXXX-XXXXXXXXXXXXX
- name: whatap_server_host
  value: XXX.XXX.XXX.XXX
- name: app_name
  value: {YOUR_APP_NAME}
- name: app_process_name
  value: {YOUR_PROCESS_NAME}
- name: whatap.okind
  value: {YOUR_GROUP_NAME}
```

```
  - name: NODE_IP
    valueFrom: {fieldRef: {fieldPath: status.hostIP}}
  - name: NODE_NAME
    valueFrom: {fieldRef: {fieldPath: spec.nodeName}}
  - name: POD_NAME
    valueFrom: {fieldRef: {fieldPath: metadata.name}}
```

다음 예시를 참조하세요.

```
apiVersion: apps/v1
kind: Deployment
metadata:
 name: python-fastapi-deployment
spec:
 replicas: 3
 selector:
   matchLabels:
     app: python-fastapi-pod
 template:
   metadata:
     labels:
       app: python-fastapi-pod
 containers:
 - name: agent-python-fastapi
   image: whatap/agent-python-fastapi
   env:
   - name: license
     value: XXXXXXXXXXXXX-XXXXXXXXXXXXX-XXXXXXXXXXXXX
   - name: whatap_server_host
     value: XXX.XXX.XXX.XXX
   - name: app_name
     value: "myapp-python-fastapi"
   - name: app_process_name
     value: "uvicorn"
   - name: NODE_IP
     valueFrom: {fieldRef: {fieldPath: status.hostIP}}
   - name: NODE_NAME
     valueFrom: {fieldRef: {fieldPath: spec.nodeName}}
   - name: POD_NAME
     valueFrom: {fieldRef: {fieldPath: metadata.name}}
```

환경변수 역할

- `license` : 에이전트 인증 확인을 위한 키입니다.
- `whatap_server_host` : 와탭 수집 서버 호스트 IP입니다.
- `app_name` : 애플리케이션의 에이전트를 식별하는 이름입니다. 에이전트 식별에 대한 자세한 내용은 다음 문서를 참조하세요.
- `app_process_name` : CPU, Heap Memory 등 수집할 대상 프로세스를 설정합니다. 예로 uwsgi, gunicorn 등이 있습니다.
- `whatap.okind` (선택사항) : Pod에 해당하는 애플리케이션들을 그룹화합니다. 디플로이먼트(Deployment) 이름으로 설정 시 해당하는 Pod들을 하나로 그룹화합니다.
- `NODE_IP` : 현재 Pod가 호스팅된 노드(Node)의 IP 주소를 수집합니다.
- `NODE_NAME` : 현재 Pod가 실행 중인 노드의 이름을 수집합니다.
- `POD_NAME` : 현재 Pod의 이름을 수집합니다.

에이전트 설치 확인

- 에이전트를 제대로 설치했는지 확인하려면 **대시보드** > **애플리케이션 서비스 대시보드** 메뉴로 이동하세요.
- 컨테이너에서 `ps -ef | grep whatap_python` 명령어를 실행해 와탭 Python 서비스가 정상적으로 실행되는지 확인하세요.

Docker PHP 설치

관리 > **에이전트 설치** > 하단의 **애플리케이션 설치** 클릭 > 설치 안내 중 **Docker PHP** 탭 선택

Docker 컨테이너 기반으로 실행하는 PHP 애플리케이션에 와탭 모니터링 에이전트를 적용하고 컨테이너 이미지를 패키징하는 과정을 다음과 같이 안내합니다.

> ⓘ EKS Fargate는 지원 예정입니다.

에이전트 다운로드

PHP 애플리케이션 도커 이미지 빌드 시 whatap-php 패키지를 설치하세요.

- **x64** **AArch64**

x64

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

AArch64

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

에이전트 설정

- 다음 명령어를 이용해 설치 스크립트를 설정하세요. 와탭 액세스 키, 서버 IP 정보를 입력하면 PHP 환경 정보를 자동으로 수집합니다.

  ```
  RUN (echo "[액세스 키]"; echo "[서버 IP]")|/usr/whatap/php/install.sh
  ```

- 다음과 같이 PHP 환경 정보를 직접 입력할 수 있습니다.

```
# RUN /usr/whatap/php/install.sh -l [액세스 키] -s [서버 IP] -v [PHP 버전 x.x] -i [php.ini의 전체 경로] -p [프로세스명] -z [thread safety]
RUN /usr/whatap/php/install.sh -l xxxxx -s 1.1.1.1/2.2.2.2 -v 7.0 -i /etc/php.ini -p httpd [-z]
```

ⓘ **액세스 키**가 Docker Public 상태의 이미지에 노출되지 않도록 주의하세요.

컨테이너 환경변수 설정

Docker 빌드 후 쿠버네티스 환경 내 **컨테이너 환경변수**를 설정하세요. 애플리케이션 배포 yaml 파일에 다음 내용을 추가하세요.

```
env:
- name: NODE_IP
  valueFrom: {fieldRef: {fieldPath: status.hostIP}}
- name: NODE_NAME
  valueFrom: {fieldRef: {fieldPath: spec.nodeName}}
- name: POD_NAME
  valueFrom: {fieldRef: {fieldPath: metadata.name}}
- name: WHATAP_PHP_AGENT_PATH
  value: /usr/whatap/php/whatap_php_static
- name: WHATAP_CONTAINER_CONF_PATH
  value: /usr/whatap/php
```

다음 예시를 참조하세요.

```
apiVersion: v1
kind: Pod
metadata:
  name: # Pod Name
  labels: # Pod Labels
spec:
  containers:
    env:
    - name: NODE_IP
```

```
    valueFrom: {fieldRef: {fieldPath: status.hostIP}}
  - name: NODE_NAME
    valueFrom: {fieldRef: {fieldPath: spec.nodeName}}
  - name: POD_NAME
    valueFrom: {fieldRef: {fieldPath: metadata.name}}
  - name: WHATAP_PHP_AGENT_PATH
    value: /usr/whatap/php/whatap_php_static
  - name: WHATAP_CONTAINER_CONF_PATH
    value: /usr/whatap/php
```

환경변수 역할

- `NODE_IP` : 현재 Pod가 호스팅된 노드(Node)의 IP 주소를 수집합니다.
- `NODE_NAME` : 현재 Pod가 실행 중인 노드의 이름을 수집합니다.
- `POD_NAME` : 현재 Pod의 이름을 수집합니다.
- `WHATAP_PHP_AGENT_PATH` : PHP 에이전트 실행 파일 경로를 지정합니다.
- `WHATAP_CONTAINER_CONF_PATH` : 컨테이너 ID가 저장될 경로를 지정합니다.

에이전트 실행

- **Command** **Supervisor**

애플리케이션 시작 명령어 앞에 추가로 whatap-php 시작 명령어를 추가하세요.

```
sh -c "/usr/whatap/php/whatap-php start && [애플리케이션 시작 명령어]"
```

Supervisor로 애플리케이션을 시작하는 경우 다음 코드를 참조해 whatap-php 서비스를 추가하세요.

```
RUN echo "[program:whatap-php]" >> /etc/supervisor/conf.d/whatap.conf
RUN echo "command = sh -c '/usr/whatap/php/whatap-php start'" >> /etc/supervisor/conf.d/whatap.conf
RUN echo "user = root" >> /etc/supervisor/conf.d/whatap.conf
RUN echo "startsecs = 0" >> /etc/supervisor/conf.d/whatap.conf
RUN echo "autostart = true" >> /etc/supervisor/conf.d/whatap.conf
RUN echo "autorestart = false" >> /etc/supervisor/conf.d/whatap.conf
```

```
RUN echo "stdout_logfile = /dev/stdout" >> /etc/supervisor/conf.d/whatap.conf
RUN echo "stdout_logfile_maxbytes=0" >> /etc/supervisor/conf.d/whatap.conf
RUN echo "stderr_logfile = /dev/stderr" >> /etc/supervisor/conf.d/whatap.conf
RUN echo "stderr_logfile_maxbytes=0" >> /etc/supervisor/conf.d/whatap.conf
```

다음 명령어를 실행해 와탭 서비스가 정상 실행되었는지 확인하세요. 애플리케이션 서버가 실행되면 에이전트가 모니터링 정보를 수집하기 시작합니다.

```
ps -ef | grep whatap_php
```

에이전트 설치 확인

에이전트를 제대로 설치했는지 확인하려면 **대시보드** > **애플리케이션 서비스 대시보드** 메뉴로 이동하세요.

다운로드한 파일을 직접 설치한 다음, **대시보드** 메뉴에서 에이전트를 확인할 수 없다면 다음 사항을 확인하세요.

- 컨테이너에서 `ps -ef | grep whatap` 명령어를 실행해 에이전트 옵션을 적용했는지 확인하세요.
- 컨테이너의 /usr/whatap/php/logs 경로의 내용을 확인하세요. 에이전트 로그는 logs/whatap-`{boot or install}`-yyyymmdd.log 형식의 파일명으로 출력됩니다.

Docker Go 설치

관리 > **에이전트 설치** > 하단의 **애플리케이션 설치** 클릭 > 설치 안내 중 **Docker Go** 탭 선택

Docker 컨테이너 기반의 Go 애플리케이션에 WhaTap 모니터링 에이전트를 설치하는 방법을 안내합니다. 다음 2가지 방법 중 선택하세요.

- **빠른 설치 방법**: Dockerfile 작성부터 컨테이너 실행, 모니터링 데이터 확인까지 전체 설치 과정
- **배포 환경별 설정**: 아키텍처(x64/ARM64), Kubernetes, Docker Compose 환경별 추가 설정 방법

> EKS Fargate는 지원 예정입니다.

> 설치를 시작하기 전에 다음 정보를 준비하세요.
>
> - WhaTap 프로젝트 **액세스 키** ([프로젝트 메뉴] > [관리] > [에이전트 설치]에서 확인)
> - WhaTap **수집 서버 IP 주소**

빠른 설치 방법

3개 파일만으로 WhaTap 모니터링이 적용된 Docker 이미지를 빌드할 수 있습니다.

1단계. 필수 파일 준비

프로젝트 루트에 다음 3개 파일을 생성하세요.

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

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

WORKDIR /app
```

```
# whatap-go-inst 설치
#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/

# 소스 복사
COPY go.mod go.sum ./
RUN go mod download
COPY . .

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

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

WORKDIR /app

# WhaTap 데이터 중계 에이전트 설치
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

# 애플리케이션 복사
COPY --from=builder /app/main .

# 시작 스크립트
COPY entrypoint.sh /app/
RUN chmod +x /app/entrypoint.sh

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

```
#!/bin/sh

# whatap.conf 생성 (환경변수에서)
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

# 데이터 중계 에이전트 시작 후 애플리케이션 실행
/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")
}
```

2단계. 이미지 빌드

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

3단계. 컨테이너 실행

환경변수로 WhaTap 설정을 전달합니다.

```
docker run -d \
  -e WHATAP_LICENSE={액세스 키} \
  -e WHATAP_SERVER_HOST={수집 서버 IP} \
  -e WHATAP_APP_NAME=myapp \
  -p 8080:8080 \
```

```
my-go-app
```

4단계. 모니터링 확인

1. 애플리케이션에 트래픽 발생: `curl http://localhost:8080`
2. WhaTap 모니터링 서비스에서 데이터 확인

> ✅ 문제 해결 섹션을 참고하세요.

배포 환경별 설정

아키텍처별 Dockerfile 설정

앞의 빠른 시작 예제는 **x64 (AMD64)** 기준입니다. ARM64 환경에서는 에이전트 다운로드 URL만 변경하세요.

WhaTap 에이전트 아키텍처별 URL

아키텍처	에이전트 다운로드 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 수정 예제

Dockerfile의 에이전트 설치 부분만 변경:

```
# Runtime stage에서 변경
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
```

ARM64 전체 Dockerfile 보기

```
FROM golang:1.21-alpine AS builder

WORKDIR /app

# whatap-go-inst 설치
#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 배포

Docker 이미지를 빌드한 후 Kubernetes 클러스터에 배포할 수 있습니다.

1단계: Docker 이미지 레지스트리 푸시

```
# 이미지 태그 지정
docker tag my-go-app:latest your-registry/my-go-app:v1.0

# 레지스트리 푸시
docker push your-registry/my-go-app:v1.0
```

2단계: WhaTap 설정 Secret 생성

```
kubectl create secret generic whatap-secret \
  --from-literal=license={액세스 키}
```

3단계: Deployment 배포

- 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: "{수집 서버 IP}"
      - name: WHATAP_APP_NAME
        value: "my-go-app"
      # Kubernetes 메타데이터 (선택사항)
      - 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
```

환경변수 설명

필수 환경변수

환경변수	설명	예시
WHATAP_LICENSE	WhaTap 프로젝트 액세스 키	x4t2r...
WHATAP_SERVER_HOST	WhaTap 수집 서버 IP	13.124.11.223
WHATAP_APP_NAME	애플리케이션 이름 (선택)	my-go-app

Kubernetes 메타데이터 (선택사항)

파드의 실행 위치 정보를 WhaTap에 전달하려면 추가하세요:

환경변수	설명
NODE_IP	파드가 실행 중인 노드의 IP
NODE_NAME	파드가 실행 중인 노드의 이름
POD_NAME	파드의 이름

Docker Compose

docker-compose.yml

```
version: '3.8'

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

```
    - WHATAP_SERVER_HOST={수집 서버 IP}
    - WHATAP_APP_NAME=my-go-app
  restart: unless-stopped
```

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

문제 해결

모니터링 데이터가 보이지 않는 경우

1단계: 환경변수 확인

컨테이너 내부에서 환경변수가 올바르게 설정되었는지 확인:

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

2단계: 에이전트 실행 확인

```
# 에이전트 프로세스 확인
docker exec -it <container_id> ps -ef | grep whatap

# 에이전트 로그 확인
docker exec -it <container_id> cat /app/whatap_home/logs/whatap-*.log
```

정상 출력 예시:

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

3단계: 네트워크 확인

WhaTap 수집 서버(6600 포트)로 아웃바운드 연결이 가능한지 확인:

```
docker exec -it <container_id> nc -zv {수집 서버 IP} 6600
```

계측이 적용되지 않는 경우

계측 상세 로그 확인

```
# 빌드 시 디버그 출력
GO_API_AST_DEBUG=1 whatap-go-inst go build ./...
```

일반적인 문제와 해결 방법

증상	원인	해결 방법
데이터 미수신	환경변수 미설정	`WHATAP_LICENSE`, `WHATAP_SERVER_HOST` 확인
에이전트 시작 실패	네트워크 차단	6600 포트 아웃바운드 허용
계측 미적용	초기화 누락	Dockerfile에 `whatap-go-inst init` 추가 확인
ARM64 에러	잘못된 바이너리	아키텍처별 에이전트 URL 확인

추가 지원

위 방법으로 해결되지 않는 경우:

1. WhaTap 지원 센터 문의
2. 다음 정보 제공:
   - Dockerfile 전체 내용
   - 에이전트 로그 (`/app/whatap_home/logs/`)
   - `docker logs <container_id>` 출력

설치 문제 해결

Helm 차트 기준 설치 문제 해결

이름이 `whatap` 다른 Helm 리포지토리가 있을 경우

"whatap" already exists with the same configuration, skipping

이미 동일한 이름의 다른 `whatap` 리포지토리가 존재할 경우 다음 명령어를 통해 기존 `whatap` 리포지토리 삭제 후 설치를 진행하세요.

```
helm repo remove whatap
```

이름이 `whatap-monitoring`인 다른 네임스페이스가 클러스터에 있을 경우

Error: INSTALLATION FAILED: Unable to continue with install: Namespace "whatap-monitoring" in namespace "" exists and cannot be imported into the current release

클러스터에 `whatap-monitoring` 네임스페이스가 존재한다면 사용자가 yaml 파일 기준으로 이미 와탭 쿠버네티스 에이전트를 설치했기 때문입니다. 이 경우 다음 명령어를 통해 기존 `whatap-monitoring` 네임스페이스 및 관련 리소스를 삭제하세요.

```
kubectl delete namespace whatap-monitoring
kubectl delete clusterrolebinding whatap
kubectl delete clusterrole whatap
```

`clusterrole`, `clusterrolebinding` 리소스가 이미 클러스터에 있을 경우

Error: INSTALLATION FAILED: Unable to continue with install: ClusterRole "whatap" in namespace "" exists and cannot be imported into the current release

클러스터에 `clusterrole`, `clusterrolebinding` 리소스가 이미 존재할 경우 다음 명령어를 통해 해당 리소스를 삭제하세요.

```
kubectl delete clusterrolebinding whatap
kubectl delete clusterrole whatap
```

`whatap-monitoring` 네임스페이스를 사용할 수 없는 경우

기본적으로 와탭 쿠버네티스 에이전트는 `whatap-monitoring` 네임스페이스에 설치됩니다. 그러나 해당 네임스페이스를 사용할 수 없거나 또는 사용자가 특정 네임스페이스를 지정해야 하는 경우, 다음 과정을 참고하여 원하는 네임스페이스에 에이전트 설치를 진행하세요.

사전 준비

- **차트 버전**: whatap/kube 1.7.11 이상

  다음 명령어를 통해 차트 버전을 확인하세요.

  ```
  helm search repo whatap/kube --versions
  ```

  1.7.11 버전 미만인 경우 다음 명령어를 통해 차트 버전을 업데이트하세요.

  ```
  helm repo update
  ```

- **Helm 버전**: Helm 3 이상

설치 과정

사용하고자 하는 네임스페이스 이름을 `<namespace>` 자리에 입력한 후 다음 명령어를 실행하여 에이전트를 설치하세요.

```
helm install whatap-kube-agent whatap/kube --create-namespace --namespace <namespace> --set
namespaceOverrideEnabled=true --wait -f values.yaml
```

업데이트 및 삭제

- **업데이트**

  다음 명령어를 통해 에이전트를 업데이트하세요.

```
helm update whatap-kube-agent whatap/kube --namespace <namespace> --wait
```

- **삭제**

  다음 명령어를 통해 에이전트를 삭제하세요.

```
helm uninstall whatap-kube-agent whatap/kube --namespace <namespace>
```

에이전트 기동 시 발생하는 문제 해결

구형 YAML 파일을 통해 와탭 쿠버네티스 모니터링 설치한 경우

> **Unrecognized VM option 'UseCGroupMemoryLimitForHeap'**
> **Error: Could not create the Java Virtual Machine.**
> **Error: A fatal exception has occurred. Program will exit.**

2022년 8월 24일 이전에 다운로드한 설치 YAML을 통해 와탭 쿠버네티스 모니터링을 설치한 경우 노드 에이전트(whatap-node-agent) 기동 시 오류가 발생하며 기동에 실패하는 경우가 있습니다.

원인은 다음과 같습니다. 2022년 8월 24일 이전에 다운로드한 설치 YAML 파일 내 `UseCGroupMemoryLimitForHeap` JVM 옵션을 사용한 노드 에이전트 실행 스크립트가 포함되어 있습니다.

와탭 노드 에이전트 **1.2.0 이상**(2022년 11월 21일 배포) 버전부터 에이전트 내부에서 사용하는 JDK가 업그레이드 되었으며 업그레이드된 JDK 버전은 더 이상 `UseCGroupMemoryLimitForHeap` 옵션을 제공하지 않습니다. **2022년 8월 24일 이전**에 다운로드한 YAML 파일로 와탭 쿠버네티스 모니터링을 설치한 상태로 노드 에이전트 1.2.0 이상 버전을 기동하는 경우 `Unrecognized VM option 'UseCGroupMemoryLimitForHeap'` 메세지가 발생하며 기동에 실패하게 됩니다.

해결 방법

이 문제를 해결하기 위해 다음과 같이 쿠버네티스 모니터링을 재설치하세요.

1. **관리** > **에이전트 설치** 화면에서 최신 YAML 파일을 다운로드하세요.

2. 사용자 쿠버네티스 환경에서 다운로드한 YAML 파일을 사용하여 에이전트를 설치하세요.
3. 사용자 쿠버네티스 환경에서 설치 과정 중 오류 출력 여부를 확인하세요.
4. 와탭 모니터링 에이전트 Pod가 Running 상태인지 확인하세요.

   > ⓘ 사용자 쿠버네티스 환경 및 **워크로드** > **Pod 목록** 메뉴에서 확인하세요.

5. **관리** > **쿠버네티스 에이전트 목록** 메뉴에서 각 모니터링 에이전트에서 발생하는 로그를 확인하세요.

DockerHub 이미지 pull limit으로 에이전트가 정상적으로 설치되지 않는 경우

여러 노드에서 동시에 이미지를 pull할 경우 아래와 같은 오류가 발생하며 설치에 실패할 수 있습니다. 문제가 발생한 경우 아래 명령어로 yaml에 이미지를 변경해 재설치가 필요합니다.

에러 조회

```
kubectl describe pod/<와탭 k8s 에이전트 파드명> -n whatap-monitoring
```

조회 결과

```
> > Failed to pull image "whatap/kube_mon:latest": rpc error: code = Unknown desc = toomanyrequests: You have reached your pull rate limit. Please try again later or log in to increase your rate limit.
```

이 경우, `yaml` 파일 내 DockerHub 이미지를 public.ecr.aws 레포로 변경한 후 재설치가 필요합니다.

- 에이전트 설치 화면에서 기본 설치용 yaml을 다운로드한 경우 아래 명령어를 사용해 이미지를 변경할 수 있습니다. 명령어 수행 시 yaml에 있는 기존 dockerHub 이미지가 public.ecr.aws 리포지토리 이미지로 대체됩니다.

  ```
  sed -i 's|image: "whatap/kube_mon|image: "public.ecr.aws/whatap/kube_mon|g' kube_agent.yaml
  ```

애플리케이션 연동 문제 해결

컨테이너 맵에 Java 모니터링 데이터가 표시되지 않는 경우

사용자 쿠버네티스 환경에서 Cgroups의 설정 및 관리 방식에 따라 특정 상황에서 컨테이너 ID를 확보하지 못하는 문제가 발생할 수 있습니다. 와탭은 이러한 상황에 대응하기 위해 쿠버네티스 모니터링의 컨테이너와 애플리케이션 간 매핑 메커니즘을 변경하였습니다. 해당 변경 사항의 경우 와탭 **쿠버네티스 에이전트 1.8.8** 버전과 **Java 에이전트 2.2.42** 버전부터 적용되어 Cgroups 영향으로 컨테이너 ID가 누락되는 문제를 해결합니다.

해결 방법

에이전트 업데이트

쿠버네티스 환경에서 발생하는 컨테이너 ID 식별 문제를 해결하기 위해서는 와탭 쿠버네티스 에이전트와 APM이 서로 주고받는 데이터를 정확히 식별할 수 있도록 지원하는 최신 버전을 사용해야 합니다. **쿠버네티스 에이전트**는 **1.8.8 버전 이상**을 설치하고 **Java 에이전트**는 **2.2.42 버전 이상**을 설치하세요.

> ⓘ 현재는 Java 에이전트만 지원하고 있으며 다른 언어의 APM은 추후 지원할 예정입니다.

추가 환경변수 설정

에이전트 버전 업데이트 이후에도 **컨테이너 맵**에 Java 애플리케이션 정보가 표시되지 않는 경우 사용자 환경변수를 추가하여 문제를 해결할 수 있습니다. 이 설정은 Java APM이 설치된 컨테이너 매니페스트(YAML)의 환경변수에 해당 컨테이너 내 Java 에이전트의 절대 경로를 나타내는 `WHATAP_JAVA_AGENT_PATH` 옵션을 추가합니다.

다음은 Java APM이 설치된 컨테이너 환경변수에 해당 옵션을 추가한 예시입니다.

```
env:
- name: NODE_IP
  valueFrom: {fieldRef: {fieldPath: status.hostIP}}
- name: NODE_NAME
  valueFrom: {fieldRef: {fieldPath: spec.nodeName}}
- name: POD_NAME
```

```
    valueFrom: {fieldRef: {fieldPath: metadata.name}}
  - name: OKIND
    value: {YOUR_OKIND_NAME}
  - name: license
    value: <Project Access Key>
  - name: whatap.server.host
    value: <수집서버 IP 주소>
  - name: whatap.micro.enabled
    value: "true"
  - name: WHATAP_JAVA_AGENT_PATH
    value: "/whatap/whatap.agent-2.2.35.jar"
```

컨테이너와 APM 간 정상 매핑 확인 방법

컨테이너와 애플리케이션이 정상 매핑되었는지 확인하는 방법은 다음과 같습니다.

- **대시보드** > **컨테이너 맵** 메뉴에서 **컨테이너 맵** 블록 내부에 흰색 마이크로 컨테이너가 존재할 경우 정상적으로 매핑된 상태입니다.
- **분석** > **메트릭스 조회** 메뉴에서 `app_counter` 카테고리의 `container` 필드에 애플리케이션이 실행 중인 컨테이너 ID가 기록된 경우 정상적으로 매핑된 상태입니다.

쿠버네티스 에이전트 옵션 설정

- 와탭 쿠버네티스 에이전트 배포 시 컨테이너 `env` 필드를 통해 설정할 수 있습니다.
- 컨테이너 내부 whatap.conf 파일을 수정해 설정할 수 있습니다.
- whatap.conf 파일 내 옵션이 컨테이너 `env` 필드에 설정된 옵션보다 높은 우선순위를 가집니다.
- whatap.conf 또는 컨테이너 `env` 에 설정된 값이 없는 경우 기본값이 사용됩니다.

> **컨테이너 내 애플리케이션 에이전트 옵션 설정**
>
> 컨테이너 내 Java, Python, Node.js 등 애플리케이션의 모니터링을 위해 필요 설정을 whatap.conf 파일에 작성합니다. whatap.conf 파일을 통해 설정할 수 있는 옵션에 대한 자세한 내용은 다음 링크를 참조하세요.
>
> - Java - whatap.conf 설정하기
> - Node.js - whatap.conf 설정하기
> - Python - whatap.conf 설정하기

컨테이너별 옵션 설정

whatap-node-helper

- **collect_volume_detail_enabled** bool

  기본값 `true`

  노드 에이전트에서 모든 컨테이너에 대한 `exec df` 명령어를 이용해 컨테이너 볼륨 정보 조회를 활성화합니다. 노드에 실행 중인 모든 컨테이너의 스토리지 사용량 및 볼륨 정보를 수집하는 기능을 활성화합니다.

- **use_cached_mount_info_enabled** bool

  기본값 `false`

  마운트 정보(mountinfo) 캐시 사용 여부를 설정합니다. `true` 로 설정하면 노드 헬퍼가 첫 디스크 스캔 시점의 mountinfo를 캐시해 재사용합니다. 이 경우 에이전트 기동 후 새로 마운트된 Pod PVC 볼륨이 영구적으로 누락될 수 있어 기본값은 `false` 입니다. mountinfo 재읽기 비용은 수집 주기당 미미합니다.

- **inject_container_id_to_apm_agent_enabled** bool

  기본값 `true`

  와탭 APM 컨테이너에 `exec` 를 이용한 container ID 주입을 활성화합니다. 와탭 APM으로 식별되는 컨테이너의 ID를 자동으로 추출합니다. 이를 APM 도구에 주입해 성능 모니터링 데이터와 컨테이너를 연결할 수 있습니다.

- **debug** bool

  기본값 `false`

  디버그 모드의 활성화 여부를 설정합니다. 디버그 모드 활성화 시 로그 레벨을 `DEBUG` 로, 그렇지 않을 경우 `INFO` 로 설정합니다.

- **kube_config_path** string

  기본값 `~/.kube/config`

  Kubernetes 설정 파일 경로를 지정합니다. 미설정 시~/.kube/config 를 사용합니다.

- **log_sys_out** bool

  기본값 `true`

  로그 시스템 출력을 설정합니다. 옵션 값을 `true` 로 설정 시 표준 출력(stdOut)과 파일에 동시에 로깅을 남깁니다. 변경 시 로깅 시스템에 즉시 반영됩니다.

- **collect_kube_node_process_metric_enabled** bool

  기본값 `true`

  노드에서 Kubernetes 관련 프로세스 모니터링을 활성화하거나 비활성화하는 플래그입니다. `true` 로 설정하면 모니터링이 활성화되고, `false` 로 설정하면 모니터링이 비활성화됩니다.

- **collect_kube_node_process_metric_target_list** list

  기본값

  ```
  "kubelet,containerd-shim,containerd,docker,dockerd,crio,metrics-server, coredns,aws-k8s-agent,kube-proxy"
  ```

  기본적으로 Kubernetes 관련 프로세스를 모니터링합니다. 사용자가 특정 프로세스 리스트를 수정하여 임의로 모니터링할 프로세스를 설정할 수 있습니다.

  > ⓘ **와탭에서 기본 설정된 모니터링 대상 프로세스 목록**
  >
  > - **kubelet**

각 노드에서 실행되는 컨트롤러로, 컨테이너의 상태를 지속적으로 모니터링하며 Kubernetes 마스터와 통신하여 워크로드가 올바르게 스케줄링되고 실행되도록 합니다. 컨테이너 상태 체크 및 노드 자원 관리를 담당합니다.

- **containerd**

  컨테이너 실행을 위한 고성능 컨테이너 런타임입니다. 이미지 관리, 컨테이너 생성 및 실행, 네트워킹 등을 처리합니다.

- **dockerd**

  Docker 컨테이너의 생성을 관리하는 주요 프로세스입니다. 이 프로세스는 컨테이너 이미지의 빌드, 배포, 실행, 네트워크 및 데이터 관리를 포함합니다.

- **crio**

  Kubernetes를 위해 설계된 경량화된 컨테이너 런타임으로, CRI(Container Runtime Interface)를 구현하여 컨테이너 워크로드를 효율적으로 관리합니다.

- **coredns**

  Kubernetes 클러스터 내부에서 DNS 서비스를 제공하는 프로세스로, 클러스터 내 서비스에 대한 네임 리졸루션을 처리합니다. 또한 서비스 디스커버리 및 로드 밸런싱 기능을 수행하며, 클러스터 내 서비스 간 통신을 원활하게 지원합니다.

- **aws-k8s-agent**

  AWS 환경에서 Kubernetes 네트워크 인터페이스(VPC CNI)를 관리하는 에이전트입니다. 이 프로세스는 포드 간 통신을 위해 AWS 네트워크 자원(ENI)을 할당하고 해제하는 역할을 합니다. 문제가 발생하면 네트워크 연결 장애나 리소스 할당 오류가 발생할 수 있습니다.

- **kube-proxy**

  Kubernetes 클러스터 내에서 네트워크 트래픽을 라우팅하는 프로세스입니다. 이 프로세스는 각 노드에서 실행되며, 서비스와 포드 간의 통신을 가능하게 합니다. 또한 iptables 혹은 IPVS와 같은 리눅스 네트워킹 기술을 사용하여 클러스터 내부의 요청을 올바른 포드로 전달합니다. 이를 통해 Kubernetes 클러스터 내의 서비스 디스커버리 및 로드 밸런싱을 지원합니다.

- **kube-apiserver**

Kubernetes API 서버로, 클러스터의 모든 요청을 처리하고 리소스 상태를 관리하는 중앙 구성 요소입니다. 인증, 권한 부여, 애드미션 컨트롤 등 클러스터의 전반적인 요청 처리를 담당합니다.

- **etcd**

  Kubernetes 클러스터 상태 정보를 저장하는 분산 키-값 데이터베이스입니다. 클러스터의 상태 저장 및 복구에 중요한 역할을 합니다.

- **kube-controller**

  여러 컨트롤러를 관리하고 실행하는 프로세스로, 클러스터 리소스의 상태를 유지하고 자동화 작업을 수행합니다. 노드 및 파드 상태 관리, 레플리케이션 유지 등 다양한 역할을 담당합니다.

- **kube-scheduler**

  클러스터에서 파드를 적절한 노드에 배치하는 역할을 합니다. 파드의 요구 사항과 각 노드의 상태를 고려하여 최적의 노드를 선택합니다.

whatap-node-agent

- **shutdown** bool

  기본값 `false`

  수집 서버와 연결을 강제로 중단합니다. 통신 중지를 의미합니다.

- **license** string

  프로젝트에 발급된 라이센스를 사용해 에이전트를 식별합니다.

- **whatap.server.host** string

  기본값 `13.124.11.223` 혹은 `13.209.172.35`

  수집 서버의 IP를 설정합니다. `delimiter`의 경우 `/:`을 사용합니다.

- **whatap.server.port** number

  기본값 `6600`

  수집 서버의 데이터 수신 Port를 설정합니다.

- **ONODE_NAME** string

  기본값 `NODE_NAME` or `NODE_IP` 환경 변수값

  에이전트 노드 식별을 위한 이름을 설정합니다.

- **tcp_connection_timeout** number

  기본값 `3000`

  whatap-master-agent와 수집 서버 소켓 연결 시 타임아웃 시간을 설정합니다.

- **tcp_so_timeout** number

  기본값 `60000`

  소켓 연결 후 데이터 송수신 타임아웃 시간을 설정합니다.

- **debug** bool

  기본값 `false`

  디버그 모드의 활성화 여부를 설정합니다.

- **gpuEnabled** bool

  기본값 `false`

  GPU 모니터링의 활성화 여부를 설정합니다.

- **count_interval** number

  기본값 `5000`

  메트릭스(`kube_pod` , `container`)를 수집 서버에 전송하는 주기를 설정합니다. 기본값의 단위는 ms입니다.

  > ⓘ 해당 옵션은 화면상 데이터 갱신 주기에 영향을 줄 수 있습니다. 자세한 내용은 다음 메트릭스별 영향을 받는 화면 구성요소를 참고하세요.

- **collect_interval** number

  기본값 `5000`

  에이전트가 apiserver에 요청해 메트릭스(`kube_pod` , `container`)를 수집하는 주기를 설정합니다.

  > ⓘ 해당 옵션은 화면상 데이터 갱신 주기에 영향을 줄 수 있습니다. 자세한 내용은 다음 메트릭스별 영향을 받는 화면

> ⓘ 구성요소를 참고하세요.

- **collect_apiserver_metric_by_leader** bool

  기본값 `false`

  해당 옵션이 활성화 된 경우 노드 에이전트는 메트릭 수집을 마스터 에이전트에 위임합니다. whatap-master-agent의 `collect_apiserver_metric_by_leader` 옵션 활성화 선행이 필요합니다. 메트릭 수집 위임을 통해 기존 whatap-node-agent가 발생시킨 apiserver 부하를 낮출 수 있습니다.

- **kubelet_root_dir** string

  기본값 `/var/lib/kubelet`

  kubelet 루트 디렉터리 경로를 설정합니다. 노드 에이전트는 이 경로를 기준으로 컨테이너 로그 경로의 심볼릭 링크를 해석합니다. Huawei Cloud Container Engine(CCE)과 같이 kubelet 루트 위치가 기본값과 다른 환경에서 사용합니다. 쉽게 말하면 표준 경로를 따르지 않는 환경에서 에이전트가 로그 파일을 찾도록 안내해 주는 옵션입니다.

- **container_log_path** string

  기본값 없음

  컨테이너 로그 디렉터리 경로를 직접 지정합니다. 미설정 시 노드 에이전트는 `kubelet_root_dir` 기반의 기본 경로와 `/var/log/pods` 같은 표준 경로를 차례로 탐색합니다. CCE 환경처럼 로그 경로가 비표준이어서 자동 탐색이 어려울 경우에만 이 옵션으로 명시적으로 지정합니다.

  > ⓘ `kubelet_root_dir`과 `container_log_path`는 컨테이너 런타임이나 클라우드 사업자가 표준과 다른 로그 경로를 사용할 때만 설정합니다. 일반적인 EKS·GKE·AKS·온프레미스 환경에서는 기본값으로 두어도 정상 동작합니다.

- **collect_conntrack_enabled** bool

  기본값 `true`

  노드 conntrack 사용 현황 수집의 활성화 여부를 설정합니다. 활성화 시 `kube_node_perf` 카테고리에 `conntrack_count`, `conntrack_max`, `conntrack_pct` 필드를 전송합니다. `nf_conntrack` 모듈이 로드되지 않은 노드는 필드를 전송하지 않습니다. 노드 에이전트 1.9.18 버전 이상에서 지원합니다.

- **collect_pod_emptydir_enabled** bool

  기본값 `true`

emptyDir 볼륨 사용량 수집의 활성화 여부를 설정합니다. 활성화 시 노드 에이전트가 kubelet 볼륨 디렉터리를 60초 주기로 조회해 `kube_pod_volume` 카테고리로 전송합니다. `false`로 설정하면 기존 `df` 기반 수집 경로가 emptyDir 볼륨을 담당합니다. 노드 에이전트 1.9.22 버전 이상에서 지원합니다.

- **pod_emptydir_max_files** number

  기본값 `20000`

  emptyDir 볼륨 하나에서 조회할 최대 파일 수를 설정합니다. 상한에 걸린 볼륨은 `truncated` 필드를 `1`로 전송해 값이 실제보다 작다는 것을 표시합니다.

- **pod_emptydir_max_volumes** number

  기본값 `300`

  한 노드에서 조회할 최대 emptyDir 볼륨 수를 설정합니다.

- **net_send_buffer_enabled** bool

  기본값 `true`

  수집 서버와의 네트워크가 끊긴 구간에서 전송 큐를 초과한 데이터의 보관 및 재전송 여부를 설정합니다. 활성화 시 전송 큐가 가득 차면 데이터를 폐기하는 대신 보관 버퍼로 옮기고, 재연결 후 오래된 데이터부터 순서대로 전송합니다. 노드 에이전트 1.9.17 버전 이상에서 지원합니다.

- **net_send_buffer_minutes** number

  기본값 `15`

  보관 버퍼의 데이터 보관 시간을 설정합니다. 기본값의 단위는 분입니다. 보관 시간을 넘긴 데이터는 오래된 것부터 폐기합니다.

- **net_send_buffer_max_mb** number

  기본값 `64`

  보관 버퍼가 사용할 최대 메모리를 설정합니다. 기본값의 단위는 MB입니다. 상한을 넘기면 오래된 데이터부터 폐기하며, 이때 폐기된 데이터만 `pack_dropped`로 집계합니다.

whatap-control-plane-helper

- **port** string

  기본값 `9496`

whatap-control-plane-helper의 웹 서버 포트를 설정합니다.

- **cycle** number

  기본값 `5`

  control-plane-helper가 지표를 스크래핑하는 주기를 설정합니다. 기본값의 단위는 s입니다.

- **collect_control_plane_monitoring_enabled** bool

  기본값 `false`

  control-plane-monitoring 기능의 활성화 여부를 설정합니다.

- **kube_config_path** string

  기본값 `~.kube/config`

  Kubernetes 설정 파일 경로를 지정합니다. 미설정 시 ~/.kube/config 를 사용합니다.

- **kube_master_url** string

  기본값 `~/.kube/config`

  스크래핑을 위해 요청할 control-plane 의 url을 설정합니다. 미설정 시 ~/.kube/config 를 사용합니다.

- **debug** bool

  기본값 `false`

  whatap-control-plane-helper의 디버깅(로깅)의 활성화 여부를 설정합니다.

- **kube_client_tls_verify** bool

  기본값 `true`

  옵션값을 `false`로 설정하면 TLS 인증을 강제하지 않고, 인증 절차를 생략할 수 있습니다.

whatap-master-agent

- **shutdown** bool

  기본값 `false`

  수집 서버와 연결을 강제로 중단합니다. 통신 중지를 의미합니다.

- **license** string

프로젝트에 발급된 라이센스를 사용해 에이전트를 식별합니다.

- **whatap.server.host** string

  기본값 `13.124.11.223` 혹은 `13.209.172.35`

  수집 서버의 IP를 설정합니다. `delimiter` 의 경우 `/:` 을 사용합니다.

- **whatap.server.port** number

  기본값 `6600`

  수집 서버의 데이터 수신 Port를 설정합니다.

- **tcp_connection_timeout** number

  기본값 `3000`

  whatap-master-agent와 수집 서버 소켓 연결 시 타임아웃 시간을 설정합니다.

- **tcp_so_timeout** number

  기본값 `60000`

  whatap-master-agent와 수집 서버 소켓 통신 시 타임아웃 시간을 설정합니다.

- **debug** bool

  기본값 `false`

  whatap-master-agent에서 `DEBUG` 레벨 로깅 활성화 여부를 설정합니다.

- **collect_apiserver_metric_by_leader** bool

  기본값 `false`

  whatap-master-agent가 스크래퍼의 리더로 동작 활성화 여부를 설정합니다. 활성화 시 whatap-master-agent 단독으로 apiserver에 메트릭을 요청합니다.

- **collect_control_plane_monitoring_enabled** bool

  기본값 `false`

  컨트롤 플레인 모니터링 기능의 활성화 여부를 설정합니다.

- **master_to_control_plane_connection_timeout** number

  기본값 `5000`

  whatap-master-agent가 whatap-control-plane-helper와 통신 시 connection timeout이 발생할 때까지의 시간을

설정합니다. 기본값의 단위는 ms입니다.

- **master_to_control_plane_read_timeout** number

  기본값 `20000`

  whatap-master-agent가 whatap-control-plane-helper와 통신 시 read timeout이 발생할 때까지의 시간을 설정합니다. 기본값의 단위는 ms입니다.

- **control_plane_debug** bool

  기본값 `false`

  컨트롤 플레인 모니터링에 대한 whatap-master-agent에서의 모든 디버그 로그 기능의 활성화 여부를 설정합니다.

- **store_k8s_object_manifest_enabled** bool

  기본값 `true`

  오브젝트 매니페스트 수집 기능의 활성화 여부를 설정합니다. 활성화 시 하루에 한번 클러스터 오브젝트들의 매니페스트 정보를 수집합니다.

- **store_k8s_object_manifest_timezone** string

  기본값 `UTC`

  오브젝트 매니페스트 수집을 위한 타임존을 설정합니다. 별도 설정이 없을 경우 기본값 UTC 기준 자정에 수집을 시작합니다.

  예, 한국 시간 기준으로 자정에 실행하고 싶을 경우 `GMT+9` 혹은 `Asia/Seoul`과 같이 타임존 설정

- **store_k8s_object_manifest_max_fetch_limit** number

  기본값 `50`

  한 번에 수집할 오브젝트의 조회 개수를 설정합니다. 별도 설정이 없을 경우 기본값 `50` 기준으로 1회 요청당 50개씩 요청합니다.

  예, 전체 Pod 개수가 90개인 클러스터에서 해당 옵션 `30` 설정 시 Pod 정보를 30개씩 3회 요청

- **collect_network_policy_enabled** bool

  기본값 `true`

  NetworkPolicy 오브젝트의 이벤트 기반 수집 활성화 여부를 설정합니다. 활성화 시 NetworkPolicy가 추가, 변경, 삭제될 때마다 `kube_network_policy` 카테고리로 스냅숏을 전송합니다. ClusterRole에 `networkpolicies` 조회 권한이 필요합니다.

- **net_send_buffer_enabled** bool

기본값 `true`

수집 서버와의 네트워크가 끊긴 구간에서 전송 큐를 초과한 데이터의 보관 및 재전송 여부를 설정합니다. 노드 에이전트와 마스터 에이전트에 공통으로 적용되는 옵션입니다. 마스터 에이전트 1.9.17 버전 이상에서 지원합니다.

- **net_send_buffer_minutes** number

  기본값 `15`

  보관 버퍼의 데이터 보관 시간을 설정합니다. 기본값의 단위는 분입니다.

- **net_send_buffer_max_mb** number

  기본값 `64`

  보관 버퍼가 사용할 최대 메모리를 설정합니다. 기본값의 단위는 MB입니다.

메트릭스별 영향을 받는 화면 구성요소

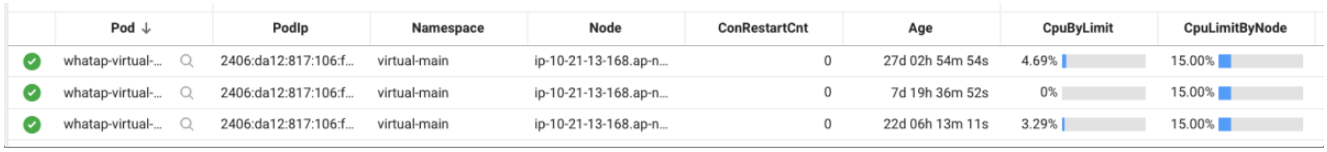
kube_pod

- **대시보드 > 컨테이너 맵**

  다음 예시 이미지의 Pod 뷰 기준 데이터는 `kube_pod` 메트릭스 수집 주기 및 수집 서버 전송 주기에 영향을 받습니다.

- **워크로드 > Pod 목록**

  다음 예시의 데이터는 `kube_pod` 메트릭스 수집 주기 및 수집 서버 전송 주기에 영향을 받습니다.



Pod	PodIp	Namespace	Node	ConRestartCnt	Age	CpuByLimit	CpuLimitByNode
whatap-virtual-...	2406:da12:817:106:f...	virtual-main	ip-10-21-13-168.ap-n...	0	27d 02h 54m 54s	4.69%	15.00%
whatap-virtual-...	2406:da12:817:106:f...	virtual-main	ip-10-21-13-168.ap-n...	0	7d 19h 36m 52s	0%	15.00%
whatap-virtual-...	2406:da12:817:106:f...	virtual-main	ip-10-21-13-168.ap-n...	0	22d 06h 13m 11s	3.29%	15.00%

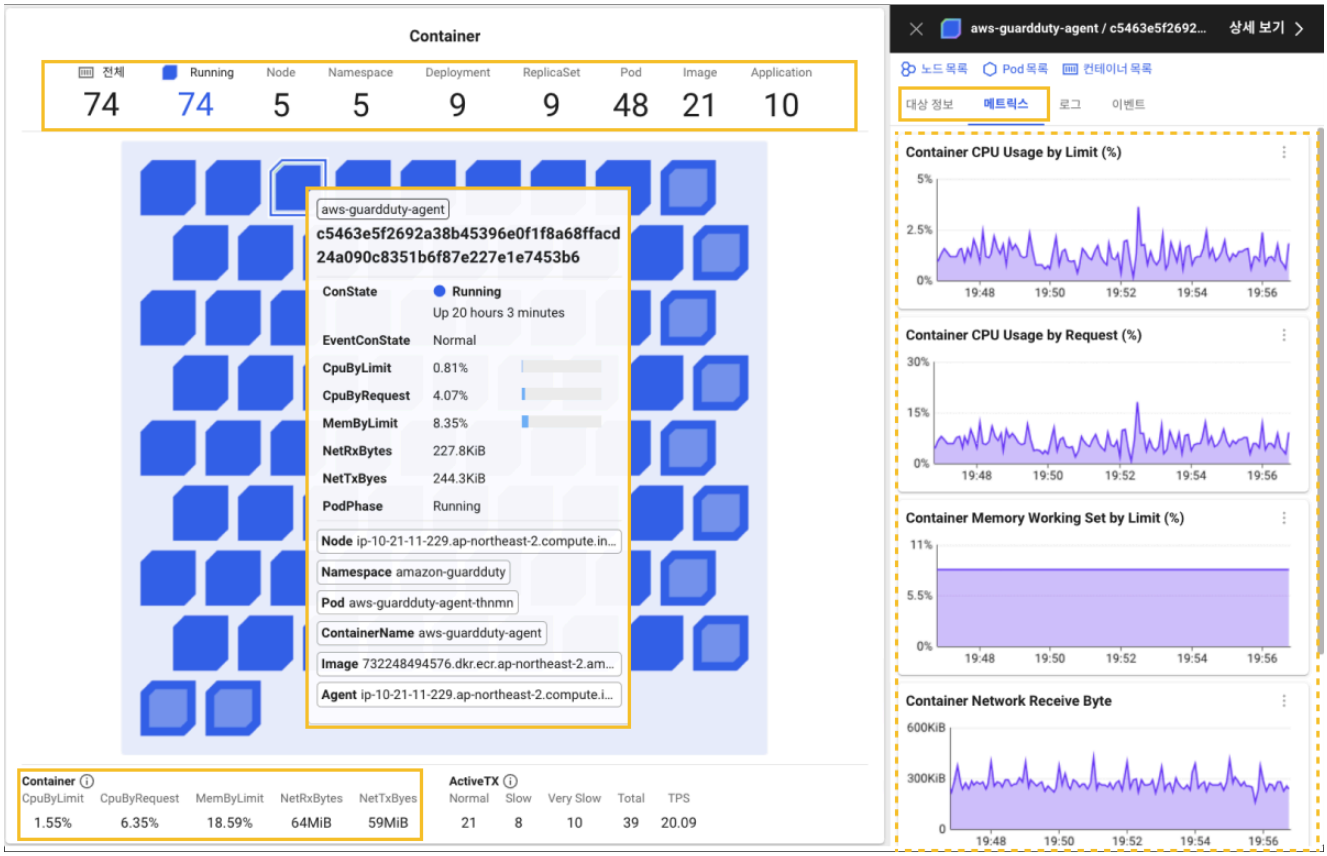
- **대시보드 > 퍼포먼스 요약**

  `kube_pod` 메트릭스를 이용하는 **에이전트 설치 노드에서 실행 중인 Pod 수** 차트 갱신 주기에 영향을 받습니다.

container

- **대시보드 > 컨테이너 맵**

  다음 예시의 컨테이너 뷰 기준 화면 데이터 갱신 주기는 `container` 메트릭스의 영향을 받습니다.



- **컨테이너 > 컨테이너 이미지**

  다음 예시의 데이터는 `container` 메트릭스의 영향을 받습니다.

Registry	Name	Tag	Repo Digest	Containers	Total Container CPU	Total Container Memory
732248494576.dkr.ecr.ap-nor...	aws-guardduty-agent	v1.5.0	sha256:0393d044ad76b911d...	5	544m	605.55MiB
602401143452.dkr.ecr.ap-nor...	eks/aws-ebs-csi-driver	v1.24.0	sha256:3dba81d0c3c8ce51a...	7	0m	81.15MiB
602401143452.dkr.ecr.ap-nor...	eks/livenessprobe	v2.10.0-eks-1-28-6	sha256:2fbd620d25bda985b...	7	4m	48.63MiB
602401143452.dkr.ecr.ap-nor...	eks/csi-node-driver-regi...	v2.9.0-eks-1-28-6	sha256:16ebeda509ca6f641...	5	0m	29.07MiB

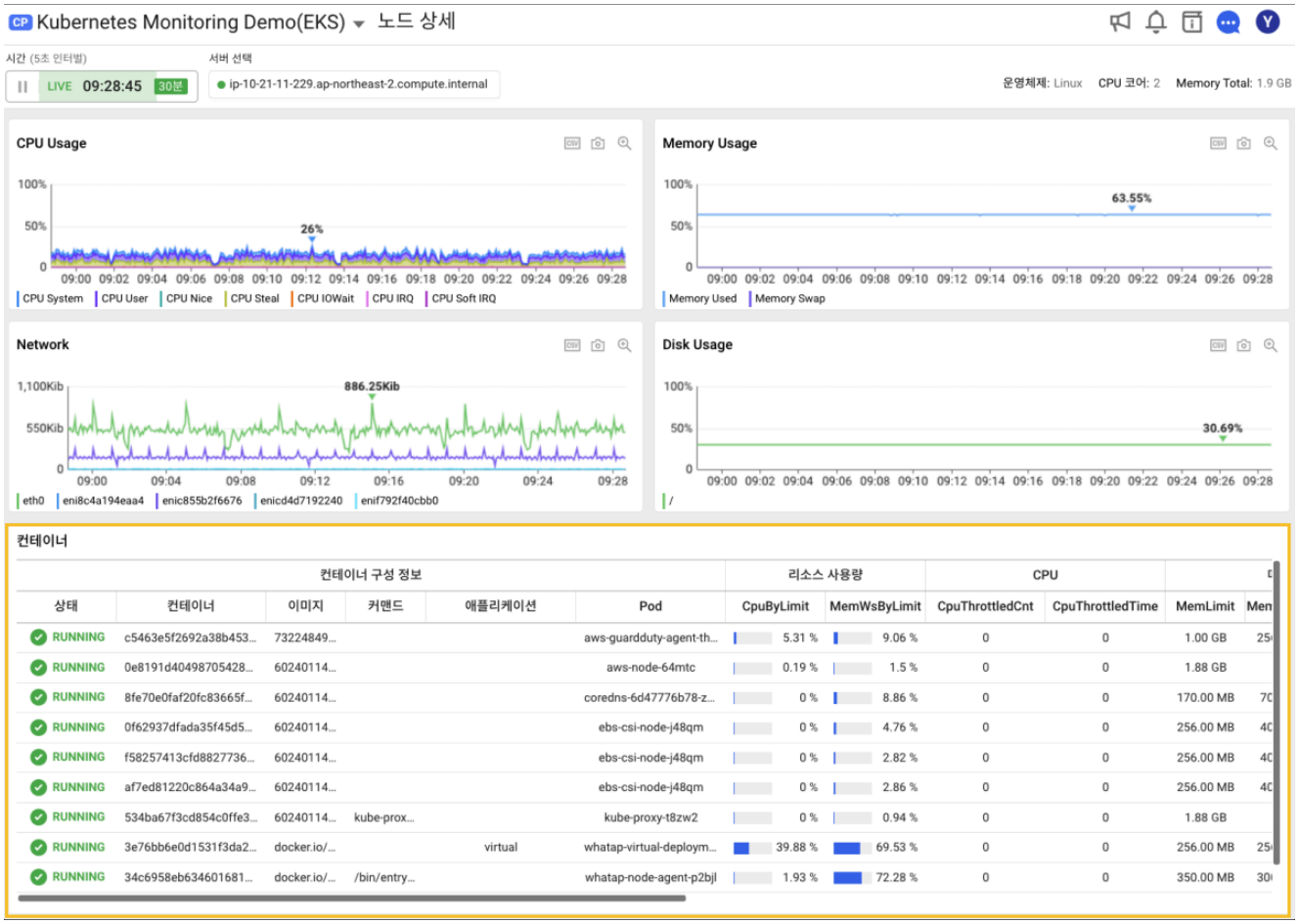
- **대시보드 > 퍼포먼스 요약**

  `container` 메트릭스를 이용하는 다음 차트의 갱신 주기에 영향을 받습니다.

  - **에이전트 설치 노드에서 실행 중인 컨테이너 수**
  - **에이전트 설치 노드 수**
  - **Total Container CPU Request**
  - **Total Container CPU Usage**

  - **Node CPU**
  - **Node Memory**
  - **Container CPU Usage by Request**
  - **Container CPU Usage by Node**
  - **Container Memory Working Set by Request**
  - **Container Memory Failure Count**
  - **Container CPU Throttling Time**
  - **Container Network Receive Byte**
  - **Container Network Transmit Byte**
- **클러스터** > **노드 상세**

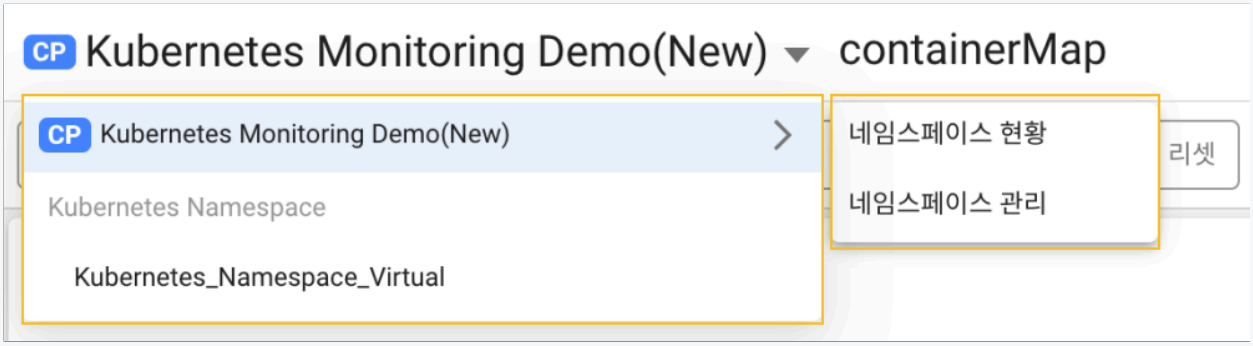
  다음 예시의 **노드 상세** 화면에서 **컨테이너** 목록의 갱신 주기는 `container` 메트릭스의 영향을 받습니다.


Kubernetes Monitoring Demo(EKS)
노드 상세
시간 (5초 인터벌)
LIVE 09:28:45 30분
서버 선택
ip-10-21-11-229.ap-northeast-2.compute.internal
운영체제: Linux
CPU 코어: 2
Memory Total: 1.9 GB
CPU Usage
26%
CPU System
CPU User
CPU Nice
CPU Steal
CPU IOWait
CPU IRQ
CPU Soft IRQ
Memory Usage
63.55%
Memory Used
Memory Swap
Network
886.25Kib
eth0
eni8c4a194eaa4
enic855b2f6676
enicd4d7192240
enif792f40cbb0
Disk Usage
30.69%
/
컨테이너
컨테이너 구성 정보
리소스 사용량
CPU
상태
컨테이너
이미지
커맨드
애플리케이션
Pod
CpuByLimit
MemWsByLimit
CpuThrottledCnt
CpuThrottledTime
MemLimit
RUNNING
aws-guardduty-agent-th...
aws-node-64mtc
coredns-6d47776b78-z...
ebs-csi-node-j48qm
kube-proxy-t8zw2
virtual
whatap-virtual-deploym...
whatap-node-agent-p2bjl

네임스페이스 관리

클러스터 프로젝트(CP)에서 **네임스페이스 관리** 메뉴를 통해 네임스페이스 프로젝트를 생성하고 관리할 수 있습니다.
네임스페이스 프로젝트에서는 해당 메뉴로 진입할 수 없습니다.

> 클러스터 프로젝트의 **대시보드** > **컨테이너 맵** 메뉴 화면 위에는 하위 네임스페이스 또는 **네임스페이스 현황** 및 **네임스페이스 관리** 메뉴로 이동할 수 있는 리스트 박스를 제공합니다.

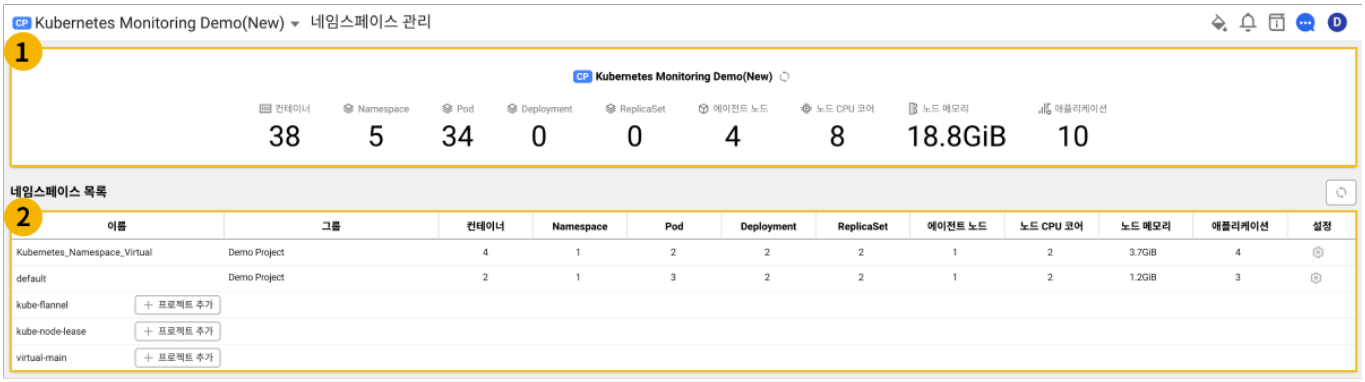


네임스페이스 프로젝트 분리를 고려하는 경우

다음과 같은 상황에서는 클러스터 프로젝트를 네임스페이스 단위로 분리하는 것을 고려하세요.

- 단일 클러스터 프로젝트에 존재하는 리소스 수가 많고, 쿠버네티스 Namespace 기준으로 분리할 수 있는 경우
- 쿠버네티스 Namespace마다 부서 또는 담당자가 달라 특정 Namespace만 모니터링해야 하는 경우

네임스페이스 프로젝트 관리

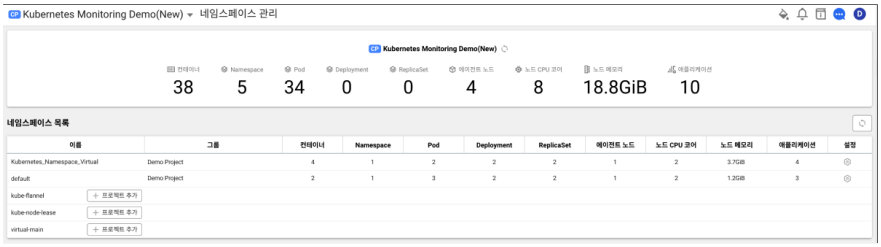


상단의 1 영역에서 대상 전체 개수 및 주요 리소스 개수를 확인할 수 있습니다. 2 **네임스페이스 목록** 영역에서 네임스페이스 프로젝트를 조회하고 추가 및 제거할 수 있습니다.

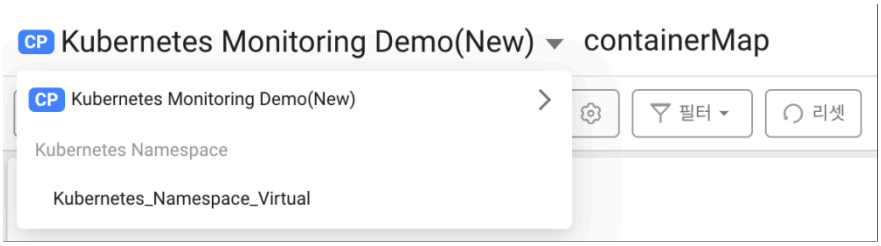
네임스페이스 프로젝트 생성

1. **네임스페이스 목록**에서 프로젝트를 생성할 항목의 **+ 프로젝트 추가** 버튼을 클릭하세요.
2. **프로젝트 추가** 창이 나타나면 네임스페이스 이름을 확인한 다음 **확인** 버튼을 클릭하세요.

3. 생성한 네임스페이스 프로젝트는 별도의 와탭 프로젝트로 분리되어 클러스터 프로젝트의 **네임스페이스 목록**에서 더 이상 표시되지 않습니다.



4. 분리된 네임스페이스 프로젝트는 클러스터 프로젝트의 **대시보드** > **컨테이너 맵** 화면 상단 프로젝트 목록 박스에서 선택해 이동할 수 있습니다.



> ⓘ 네임스페이스 프로젝트 추가 시 해당 네임스페이스는 클러스터 프로젝트(CP)에서 별도의 와탭 프로젝트로 분리됩니다. 추가된 네임스페이스 프로젝트에는 새로운 **프로젝트 액세스 키**가 부여됩니다.
>
> 분리된 네임스페이스에 애플리케이션 모니터링이 설치되어 있는 경우, 애플리케이션 모니터링 설정 내 **프로젝트 액세스 키**를 새로 추가된 네임스페이스 프로젝트의 **프로젝트 액세스 키**로 변경해야 합니다.

네임스페이스 프로젝트 삭제

1. **네임스페이스 목록**에서 삭제하려는 프로젝트 항목의 가장 오른쪽에 ⚙ **설정** 아이콘을 클릭하세요.
2. **프로젝트 관리** 메뉴로 이동하세요.
3. 화면 오른쪽 위에 **프로젝트 삭제** 버튼을 클릭하세요.

> ⓘ 네임스페이스 프로젝트를 삭제할 경우 사용자 목록 정보와 그동안 수집한 데이터는 모두 삭제합니다. 삭제한 데이터는 다시 복구할 수 없기 때문에 주의하세요.

> ⓘ 프로젝트 관리에 대한 자세한 내용은 다음 문서를 참조하세요.

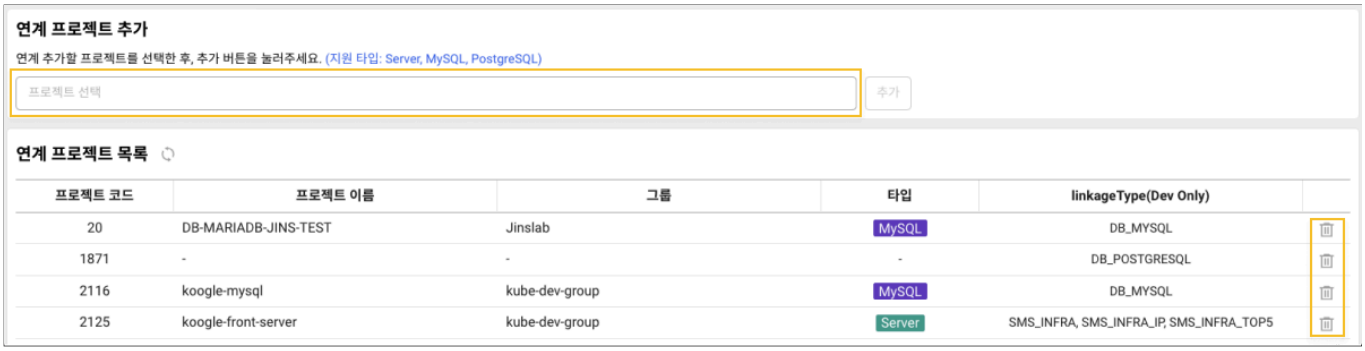
연계 프로젝트 관리

본 문서는 **연계 프로젝트 관리** 메뉴를 안내합니다. 와탭은 프로젝트간 데이터 연계를 바탕으로 통합 가시성을 제공합니다. 태그 정보를 통해 연계 프로젝트를 검색하고, 해당 프로젝트에서 문제의 원인이 될 수 있는 주요 지표를 빠르게 확인할 수 있습니다.

- 연계 프로젝트를 조회 및 추가, 삭제 등 관리 기능을 제공합니다.
- 추가한 프로젝트간 연계 분석이 가능합니다.

기본 화면 안내

홈 화면 > 프로젝트 선택 > **관리** > **연계 프로젝트 관리**



다음은 **연계 프로젝트 관리** 메뉴의 기본 화면 구성을 안내합니다. 상단의 **연계 프로젝트 추가**와 하단의 **연계 프로젝트 목록** 화면으로 구성되어있습니다.

연계 프로젝트 추가

연계 프로젝트를 추가할 수 있습니다. **프로젝트 선택** 창 클릭 시 다음과 같이 연계 등록할 수 있는 프로젝트를 조회할 수 있습니다.

- **프로젝트 선택** 창에서 **프로젝트 코드** 및 **프로젝트 이름**으로 검색할 수 있습니다.
- 프로젝트 선택 시 **추가** 버튼이 활성화됩니다. 여러 개의 프로젝트를 선택할 수 있습니다.
- 프로젝트 선택 후 **추가** 버튼 클릭 시 안내 메시지와 함께 하단의 **연계 프로젝트 목록**에 해당 프로젝트가 추가된 것을 확인할 수 있습니다.

연계 프로젝트 삭제

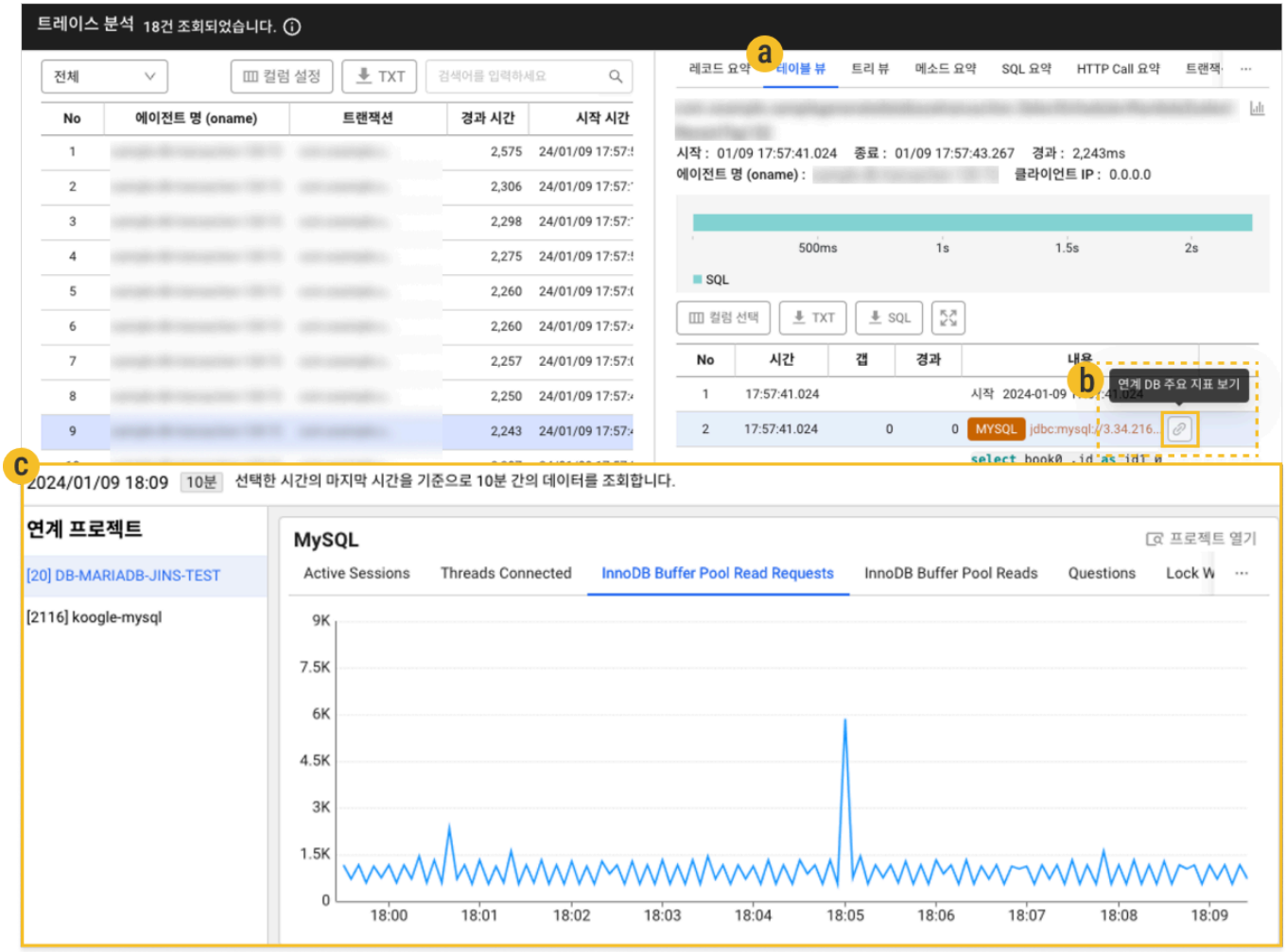
추가한 프로젝트를 연계 대상에서 제외하고 싶은 경우 **연계 프로젝트** 목록 우측의 **삭제** 아이콘을 선택해 삭제하세요.

> - 현재 Server, PostgreSQL, MySQL 제품을 지원합니다.
> - 추가한 연계 프로젝트가 목록에 표시되지 않을 경우 잠시 기다린 후 **새로 고침** 아이콘을 선택해 새로고침 하세요.
> - **삭제**의 경우 **프로젝트 수정 권한**이 필요합니다.

연계 분석 활용 예시

히트맵을 드래그해 **트레이스 분석** 창으로 이동 > 트랜잭션 항목 선택 > 오른쪽 상세 분석 화면의 **테이블 뷰** 선택

다음은 연계 프로젝트를 기반으로 연계 분석 활용 예시를 안내합니다.



1. **애플리케이션 서비스 대시보드**의 **히트맵**을 드래그해 **트레이스 분석** 창으로 이동하세요.
2. 조회를 원하는 트랜잭션을 선택 후 오른쪽 상세 분석 화면에서 a **테이블 뷰**를 선택하세요.
3. 목록에서 b 영역의 **연계 아이콘** 선택 시 c 연계 프로젝트의 주요 지표를 확인할 수 있습니다. 연계 시나리오가 다수일 경우 드롭다운 방식으로 원하는 분석 시나리오 목록을 제공합니다.

> **조회 시간**
>
> 사용자가 조회한 마지막 시간을 기준으로 10분 간의 데이터를 표시합니다.
>
> 예, **히트맵**의 34분~36분 범위 데이터 드래그 시 26분~36분간의 데이터를 표시합니다.

쿠버네티스 에이전트

쿠버네티스 내에 실행 중인 애플리케이션의 모니터링을 위해 설치한 에이전트 구성에 대해 알아봅니다.

에이전트 구성

와탭 쿠버네티스 설치 시 기본적으로 사용자 쿠버네티스 환경의 <whatap-monitoring> Namespace 하위에 에이전트용 리소스를 구성합니다.

마스터 에이전트는 Deployment, 노드 에이전트는 DaemonSet 형태입니다. 다음은 사용자 쿠버네티스 환경에 설치된 와탭 에이전트를 확인하는 방법입니다.

- **마스터 에이전트**

```
kubectl get deployments -n whatap-monitoring
kubectl get pods -l=name=whatap-master-agent -n whatap-monitoring
```

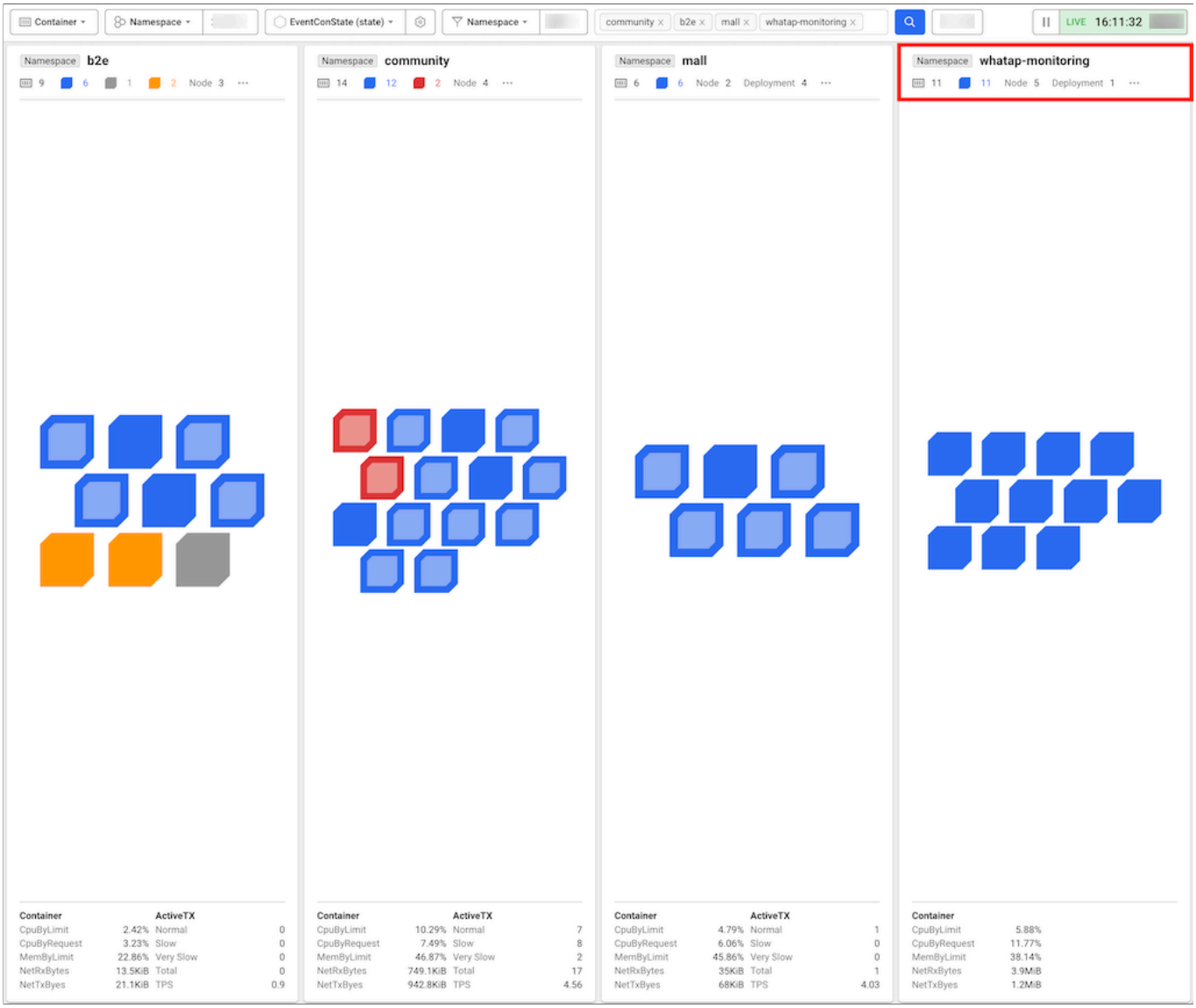
- **노드 에이전트**

```
kubectl get daemonsets -n whatap-monitoring
kubectl get pods -l=name=whatap-node-agent -n whatap-monitoring
```

다음은 <whatap-monitoring> Namespace 내에서 실행 중인 마스터 에이전트, 노드 에이전트 형태의 예입니다.

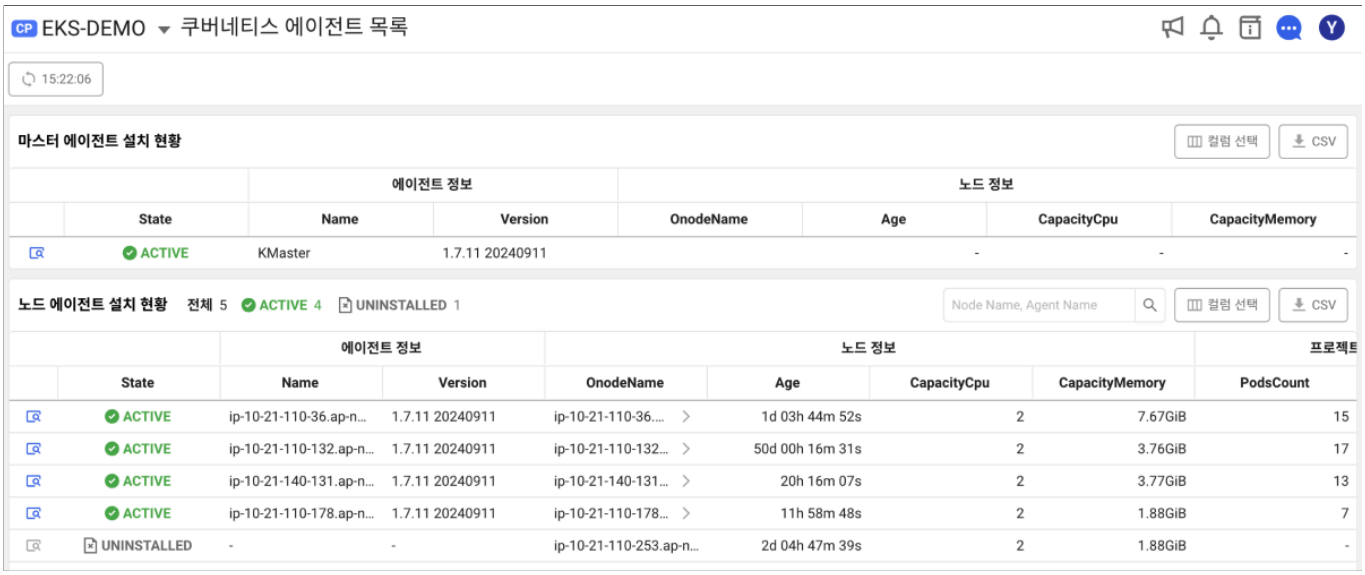
```
$ kubectl get pods -n whatap-monitoring
NAME                                READY   STATUS    RESTARTS   AGE
whatap-master-agent-57f4499567-ltbbr   1/1     Running   0          3d19h
whatap-node-agent-5q8rj             2/2     Running   0          21h
whatap-node-agent-6nb2z             2/2     Running   0          30h
whatap-node-agent-8m2vk             2/2     Running   0          21h
whatap-node-agent-9thmw             2/2     Running   0          3h31m
whatap-node-agent-kpv7d             2/2     Running   0          21h
```

다음은 와탭 쿠버네티스 컨테이너 맵 상에 표시된 <whatap-monitoring> Namespace 입니다.


Container
Namespace
EventConState (state)
Namespace
community
b2e
mall
whatap-monitoring
LIVE 16:11:32
Namespace b2e
9 6 1 2 Node 3
Namespace community
14 12 2 Node 4
Namespace mall
6 6 Node 2 Deployment 4
Namespace whatap-monitoring
11 11 Node 5 Deployment 1
Container
CpuByLimit 2.42%
CpuByRequest 3.23%
MemByLimit 22.86%
NetRxBytes 13.5KiB
NetTxBytes 21.1KiB
ActiveTX
Normal 0
Slow 0
Very Slow 0
Total 0
TPS 0.9
Container
CpuByLimit 10.29%
CpuByRequest 7.49%
MemByLimit 46.87%
NetRxBytes 749.1KiB
NetTxBytes 942.8KiB
ActiveTX
Normal 7
Slow 8
Very Slow 2
Total 17
TPS 4.56
Container
CpuByLimit 4.79%
CpuByRequest 6.06%
MemByLimit 45.86%
NetRxBytes 35KiB
NetTxBytes 68KiB
ActiveTX
Normal 1
Slow 0
Very Slow 0
Total 1
TPS 4.03
Container
CpuByLimit 5.88%
CpuByRequest 11.77%
MemByLimit 38.14%
NetRxBytes 3.9MiB
NetTxBytes 1.2MiB

쿠버네티스 에이전트 목록

클러스터 프로젝트(CP)의 **관리** 메뉴 하위에서 **쿠버네티스 에이전트 목록** 메뉴로 이동할 수 있습니다.



- 쿠버네티스 에이전트의 설치 현황을 확인할 수 있습니다. 일부 에이전트가 미설치(**UNINSTALLED**) 혹은 비활성화(**INACTIVE**) 상태일 경우 데이터 수집에 왜곡이 생길 수 있어 빠른 확인이 필요합니다.

  예, **State** 컬럼의 **UNINSTALLED**, **ACITVE**, **INACTIVE**

- 마스터 에이전트에 할당된 **CPU Limit**과 설치된 **노드 에이전트 수**를 확인할 수 있습니다.

  예, **노드 에이전트 수** 50개 이상일 경우 마스터 에이전트 **CPU Limit** 500 밀리코어 이상으로 설정 권장

- 에이전트가 설치된 노드에 할당된 CPU 및 메모리 정보를 확인할 수 있습니다.

  예, **노드 정보** 컬럼 하위의 **CapacityCpu**, **CapacityMemory**

- 에이전트가 설치되지 않은 노드들을 확인할 수 있습니다.
- 에이전트의 주요 정보를 확인할 수 있습니다.
- 노드 에이전트 버전과 관련된 안내를 확인할 수 있습니다.

  예, 노드간 버전 불일치 시 상단에 버전 불일치 안내

> **마스터 에이전트 자원 할당**
>
> 사용자의 쿠버네티스 환경 규모 즉 노드의 수에 따라 와탭 에이전트가 수집 및 처리하는 데이터의 양이 달라지기 때문에 마스터 에이전트에 충분한 자원 할당이 필요합니다. 예를 들어 **노드 에이전트 수**가 많을 경우 마스터 에이전트 **CPU Limit** 상향을 권장합니다.

새로고침

상단의 **새로고침** 아이콘 선택 시 현재 시간을 기준으로 데이터를 갱신할 수 있습니다.

컬럼 선택

그룹	컬럼명	설명
에이전트 정보	Name	에이전트 이름 또는 에이전트 별칭을 설정한 경우 에이전트 별칭을 의미합니다.
	Original Name	에이전트의 이름입니다. 일반적으로 노드의 이름과 동일합니다.
	Alias	사용자가 정의한 에이전트 별칭을 의미합니다.
	Initial	에이전트 지칭 시 사용하는 약어입니다.
	Id	고유하게 부여하는 에이전트의 식별자(ID)입니다.
	Start Time	해당 에이전트가 처음 실행을 시작한 시각입니다.
	Active Time	에이전트가 현재 활성화되어 모니터링 데이터를 수집하기 시작한 시각입니다.
	Inactive Time	에이전트가 비활성화되어 데이터 수집이 중단된 시각입니다.
	Version	현재 에이전트에 적용된 소프트웨어 버전입니다.
	IP	에이전트가 설치된 서버의 IP 주소를 의미합니다.
	CPU Cores	에이전트가 설치된 서버의 CPU 코어 수를 의미합니다.
노드 정보	OnodeName	에이전트가 설치된 노드의 이름입니다.

그룹	컬럼명	설명
	Onode	에이전트가 설치된 노드의 이름을 변환한 해시값입니다.
	Age	노드가 생성된 이후 경과한 시간을 의미합니다.
	CapacityCpu	해당 노드가 보유한 총 CPU 용량을 의미합니다.
	CapacityMemory	해당 노드가 보유한 총 메모리 용량을 의미합니다.
프로젝트 리소스	**PodsCount**	노드 에이전트가 설치된 노드에서 실행 중인 Pod의 수를 의미합니다. 분리된 네임스페이스는 제외됩니다.
	ContainersCount	노드 에이전트가 설치된 노드에서 실행 중인 컨테이너의 수를 의미합니다. 분리된 네임스페이스는 제외됩니다.

컬럼 선택 옵션 활용 시 원하는 에이전트 정보를 선택해 에이전트 목록을 조회할 수 있습니다.

- 선택한 컬럼 목록은 메뉴 재진입 시 유지됩니다.
- **초기화** 버튼 선택 시 사전 정의된 기본 주요 지표 컬럼이 선택됩니다. 이 경우 기존 컬럼 선택 내역은 삭제됩니다.
- 컬럼 선택 후 우측 하단 **적용** 버튼을 클릭 해 선택 내역을 적용할 수 있습니다.

다운로드

CSV **다운로드** 아이콘 선택 시 **범위** 및 **표시 옵션**을 설정해 해당 데이터를 CSV 파일로 다운로드할 수 있습니다.

상세 보기

마스터 에이전트 에이전트 **상세** 컬럼의 아이콘을 선택하면 오른쪽에 상세 정보 창이 나타납니다. **환경 변수**, **힙 히스토그램**, **스레드 목록/덤프**, **에이전트 로그**를 확인할 수 있습니다.

노드 에이전트 목록에서 조회를 원하는 에이전트 **상세** 컬럼의 아이콘을 선택하면 오른쪽에 해당 노드의 상세 정보 창이 나타납니다.

> ⓘ 노드 상세와 관련한 자세한 내용은 다음 문서를 참조하세요.

에이전트 업데이트

> ⓘ
> - EKS Fargate는 지원 예정입니다.
> - 와탭 쿠버네티스 에이전트의 최신 버전은 hub.docker.com에서 확인할 수 있습니다.

에이전트 업데이트가 필요한 경우 다음 명령을 실행하세요.

- **마스터 에이전트**

```
kubectl set image -n whatap-monitoring deployment/whatap-master-agent whatap-master-agent=whatap/
kube_mon:{agent_version}
```

- **노드 에이전트**

```
kubectl set image -n whatap-monitoring daemonset/whatap-node-agent whatap-node-agent=whatap/
kube_mon:{agent_version}
```

- **Helm 차트 기준 설치 시 에이전트 애플리케이션 업데이트**

```
# release 업그레이드
helm upgrade whatap-kube-agent whatap/kube -f values.yaml
```

에이전트 삭제

쿠버네티스 에이전트 삭제

에이전트 설치 단계에서 다운로드한 쿠버네티스 버전과 Container Runtime에 맞는 YAML 파일명을 확인하고 절차에 따라 쿠버네티스 에이전트 삭제 과정을 진행하세요.

쿠버네티스 에이전트 YAML 파일명

Docker	YAML 파일명
쿠버네티스 버전 1.16 이상	whatap_kube_1.16.yaml
쿠버네티스 버전 1.16 이상(GPU 버전)	whatap_kube_1.16_gpu.yaml
쿠버네티스 버전 1.16 미만	whatap_kube.yaml
쿠버네티스 버전 1.16 미만(GPU 버전)	whatap_kube_gpu.yaml

Containerd	YAML 파일명
쿠버네티스 버전 1.16 이상	whatap_kube_containerd_1.16.yaml
쿠버네티스 버전 1.16 이상(GPU 버전)	whatap_kube_containerd_1.16_gpu.yaml
쿠버네티스 버전 1.16 미만	whatap_kube_containerd.yaml
쿠버네티스 버전 1.16 미만(GPU 버전)	whatap_kube_containerd_gpu.yaml

CRIO YAML	파일명
쿠버네티스 버전 1.16 이상	whatap_kube_crio_1.16.yaml
쿠버네티스 버전 1.16 이상(GPU 버전)	whatap_kube_crio_1.16_gpu.yaml

CRIO YAML	파일명
쿠버네티스 버전 1.16 미만	whatap_kube_crio.yaml
쿠버네티스 버전 1.16 미만(GPU 버전)	whatap_kube_crio_gpu.yaml

YAML 파일이 있을 경우

1. 설치 단계에서 다운로드한 YAML 파일명으로 설치된 쿠버네티스 에이전트를 삭제합니다. 다음 명령어의 표기 부분을 사용자의 YAML 파일명으로 수정 후 실행하세요.

```
kubectl delete -f < 사용자의 YAML 파일명 >
```

2. 쿠버네티스 에이전트 삭제가 정상적으로 완료되었을 경우 다음과 같은 메시지를 확인할 수 있습니다.

```
namespace "whatap-monitoring" deleted
serviceaccount "whatap" deleted
clusterrole.rbac.authorization.k8s.io "whatap" deleted
configmap "node-start-script" deleted
clusterrolebinding.rbac.authorization.k8s.io "whatap" deleted
daemonset.apps "whatap-node-agent" deleted
configmap "master-start-script" deleted
deployment.apps "whatap-master-agent" deleted
service "whatap-master-agent" deleted
```

YAML 파일이 없을 경우

> whatap-monitoring 네임스페이스에 직접 사용자 정의 리소스를 생성한 경우 해당 리소스도 함께 삭제됩니다.

1. 다음 명령어를 실행해 쿠버네티스 에이전트를 삭제하세요.

```
kubectl delete namespace whatap-monitoring
kubectl delete clusterrolebinding whatap
```

```
kubectl delete clusterrole whatap
```

2. 쿠버네티스 에이전트 삭제가 정상적으로 완료되었을 경우 다음과 같은 메시지를 확인할 수 있습니다.

```
namespace "whatap-monitoring" deleted
clusterrolebinding.rbac.authorization.k8s.io "whatap" deleted
clusterrole.rbac.authorization.k8s.io "whatap" deleted
```

Helm 차트 기준 설치 시 에이전트 삭제

다음 명령어를 실행해 와탭 쿠버네티스 에이전트 애플리케이션을 삭제하세요.

```
# release 삭제
helm uninstall whatap-kube-agent
```

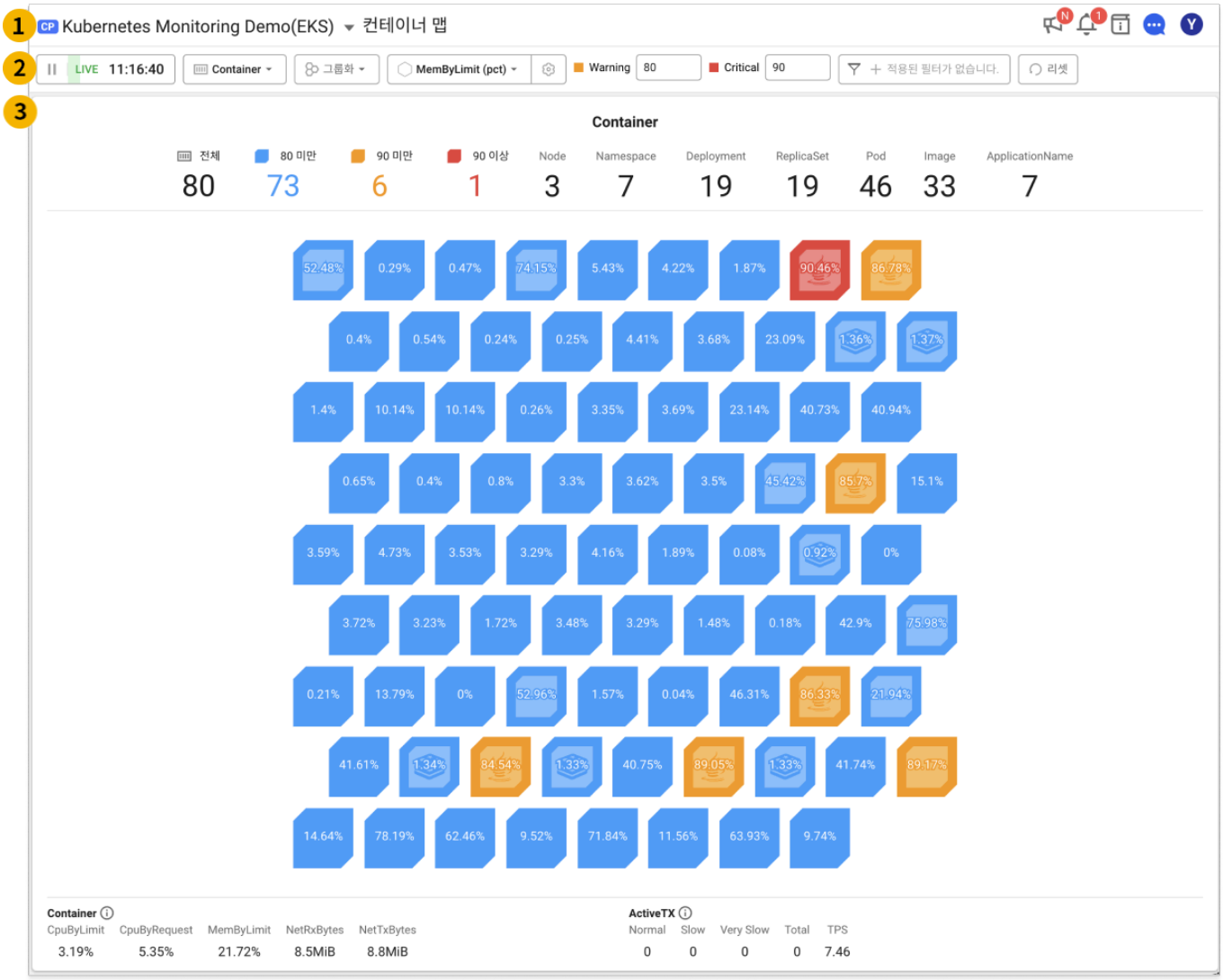
컨테이너 맵

와탭 쿠버네티스의 **컨테이너 맵**은 쿠버네티스 모니터링의 시작점입니다. 사용자 설정 보기에 따라 각 개체는 하나의 컨테이너, 또는 하나의 Pod를 나타냅니다.

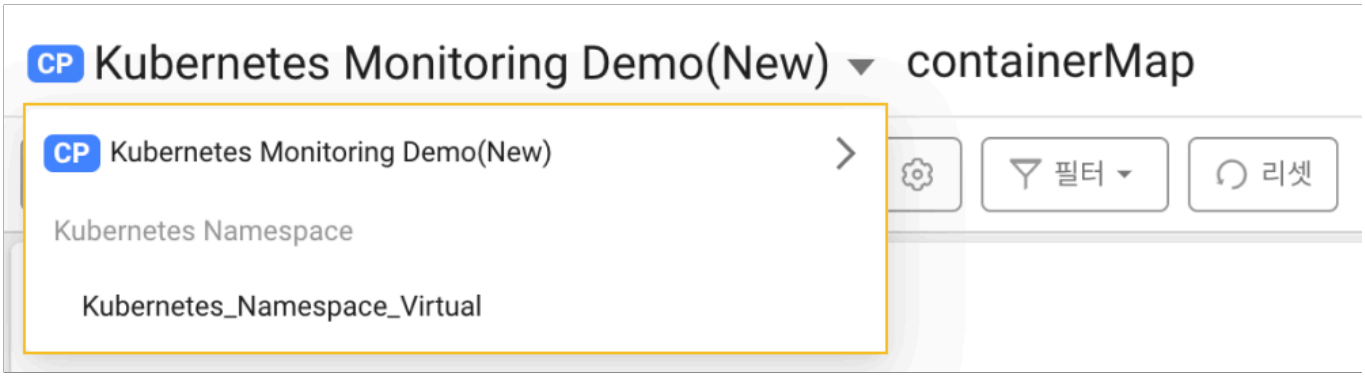
- 컨테이너 맵을 통해 분석 대상을 선택하고 그룹을 구성할 수 있습니다.
- 상태 지표 및 임계치를 설정해 개별 컨테이너 또는 Pod의 상태를 한 눈에 볼 수 있습니다.
- 개별 컨테이너 또는 Pod를 선택해 상세 정보를 확인할 수 있습니다. **트레이스**, **메트릭스**, **로그**, **이벤트**와 같은 핵심 모니터링 데이터를 제공합니다.

다음 동영상은 **컨테이너 맵** 메뉴에 대한 기본 안내입니다.

기본 화면 안내


1
Kubernetes Monitoring Demo(EKS)
컨테이너 맵
2
LIVE 11:16:40
Container
그룹화
MemByLimit (pct)
Warning 80
Critical 90
적용된 필터가 없습니다.
리셋
3
Container
전체 80
80 미만 73
90 미만 6
90 이상 1
Node 3
Namespace 7
Deployment 19
ReplicaSet 19
Pod 46
Image 33
ApplicationName 7
52.48% 0.29% 0.47% 74.15% 5.43% 4.22% 1.87% 90.46% 86.78%
0.4% 0.54% 0.24% 0.25% 4.41% 3.68% 23.09% 1.36% 1.37%
1.4% 10.14% 10.14% 0.26% 3.35% 3.69% 23.14% 40.73% 40.94%
0.65% 0.4% 0.8% 3.3% 3.62% 3.5% 45.42% 85.7% 15.1%
3.59% 4.73% 3.53% 3.29% 4.16% 1.89% 0.08% 0.92% 0%
3.72% 3.23% 1.72% 3.48% 3.29% 1.48% 0.18% 42.9% 75.98%
0.21% 13.79% 0% 52.96% 1.57% 0.04% 46.31% 86.33% 21.94%
41.61% 1.34% 84.54% 1.33% 40.75% 89.05% 1.33% 41.74% 89.17%
14.64% 78.19% 62.46% 9.52% 71.84% 11.56% 63.93% 9.74%
Container
CpuByLimit 3.19%
CpuByRequest 5.35%
MemByLimit 21.72%
NetRxBytes 8.5MiB
NetTxBytes 8.8MiB
ActiveTX
Normal 0
Slow 0
Very Slow 0
Total 0
TPS 7.46

1 클러스터 프로젝트 - 네임스페이스 프로젝트 이동



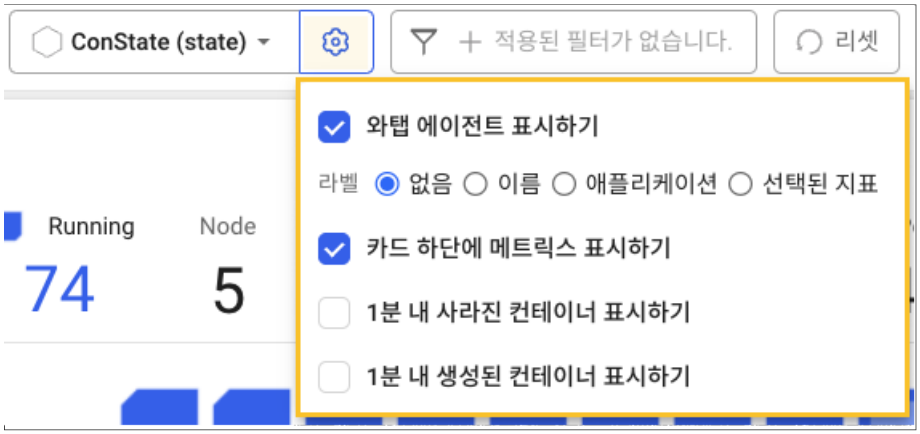
클러스터 프로젝트 CP 하위에 네임스페이스 프로젝트를 생성한 경우 컨테이너 맵 화면 위의 ▾ **더보기** 아이콘 선택 시 프로젝트 리스트 박스가 나타납니다. 프로젝트 리스트 박스를 통해 클러스터 프로젝트를 포함한 프로젝트 목록을 확인하고 프로젝트 간 이동할 수 있습니다.

> ⓘ 네임스페이스 프로젝트 생성 및 관리에 관한 자세한 내용은 다음 문서를 참조하세요.

2 사용자 설정

컨테이너 맵이 제공하는 여러 사용자 설정을 통해 원하는 화면 보기를 생성할 수 있습니다. 화면 순서대로 사용할 수 있는 기능은 다음과 같습니다.

- 카테고리: 분석 대상을 선택할 수 있습니다. **Container** 또는 **Pod**를 지정할 수 있습니다.
- **그룹화**: **그룹화**(클러스터링) 옵션을 통해 클러스터 구성 요소 간의 계층관계를 총 2단계까지 표현할 수 있습니다. 자세한 내용은 다음 문서를 참조하세요.
- **ConState(state)**: 차트 지표를 선택할 수 있습니다.
- ⚙ 버튼을 클릭하면 각 개체에 컨테이너 또는 Pod의 정보를 표시할 수 있는 옵션을 선택할 수 있습니다.



  - **와탭 에이전트 표시하기**: 컨테이너 또는 Pod의 정보를 표시할 수 있습니다. 원하는 **라벨** 옵션을 선택하세요. 라벨 옵션에 대한 자세한 내용은 다음 문서를 참조하세요.
  - **카드 하단에 메트릭스 표시하기**: 화면 하단에 메트릭스 정보를 표시할 지 여부를 선택할 수 있습니다.
  - **1분 내 생성된 컨테이너** 또는 **Pod 표시하기**: 1분 이내 생성된 컨테이너 또는 Pod를 표시하는 옵션입니다.
  - **1분 내 사라진 컨테이너** 또는 **Pod 표시하기**: 1분 이내 사라진 컨테이너 또는 Pod를 표시하는 옵션입니다.
- **필터**: 사용자가 선택한 항목에 따라 필터링해 컨테이너 또는 Pod의 상태를 확인할 수 있습니다.
- **리셋**: 사용자가 설정한 화면 보기를 초기화합니다.

지표 선택 및 임계치 설정

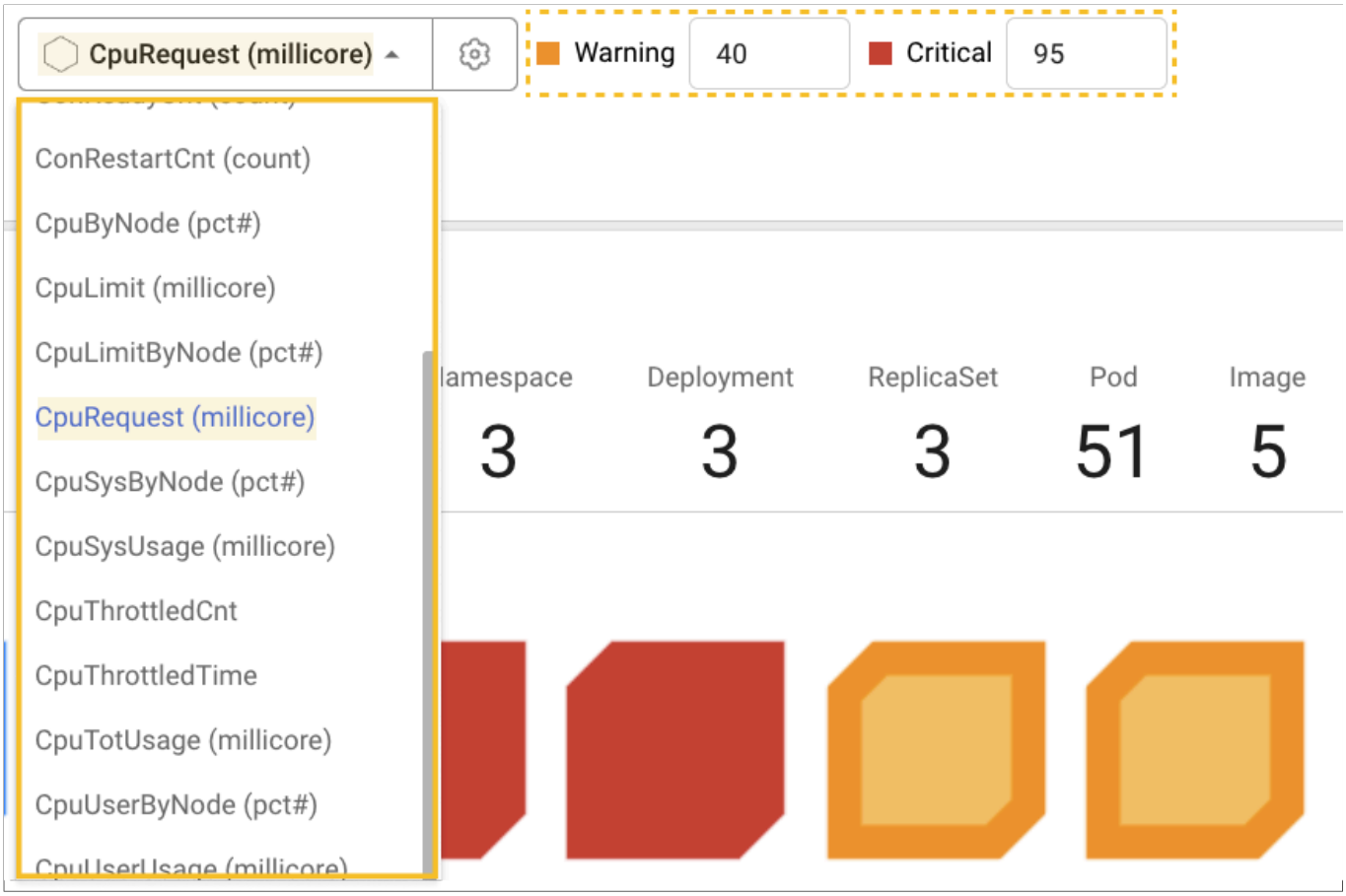


차트 지표 선택 및 임계치 설정은 컨테이너 맵의 상태 표현을 결정합니다. 선택한 지표의 단위가 숫자인 경우 임계치를 설정할 수 있습니다. 설정한 임계치는 다음 범주에 따라 색상으로 표현됩니다.

- 0 이상 Warning 이하: 파란색
- Warning 초과 Critical 이하: 주황색
- Critical 초과: 빨간색

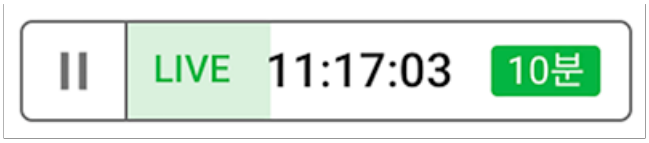
> ⓘ
> - 수치형 지표 단위로는 **pct**(percentage), **bps**, **millicore** 등이 있습니다. **pct#**는 100% 이상의 값이 나올 수 있는 지표 단위입니다.
> - 차트 지표 선택과 임계치 설정에 대한 자세한 내용은 다음 문서를 참조하세요.

라벨 옵션

- **없음**: 블럭에 라벨을 표시하지 않습니다.
- **이름**: 컨테이너/POD의 이름을 표시합니다.
- **애플리케이션**: 컨테이너/POD 내의 애플리케이션을 표시합니다.
- **선택된 지표**: **ConState(state)** 항목에서 선택한 차트 지표를 표시합니다.

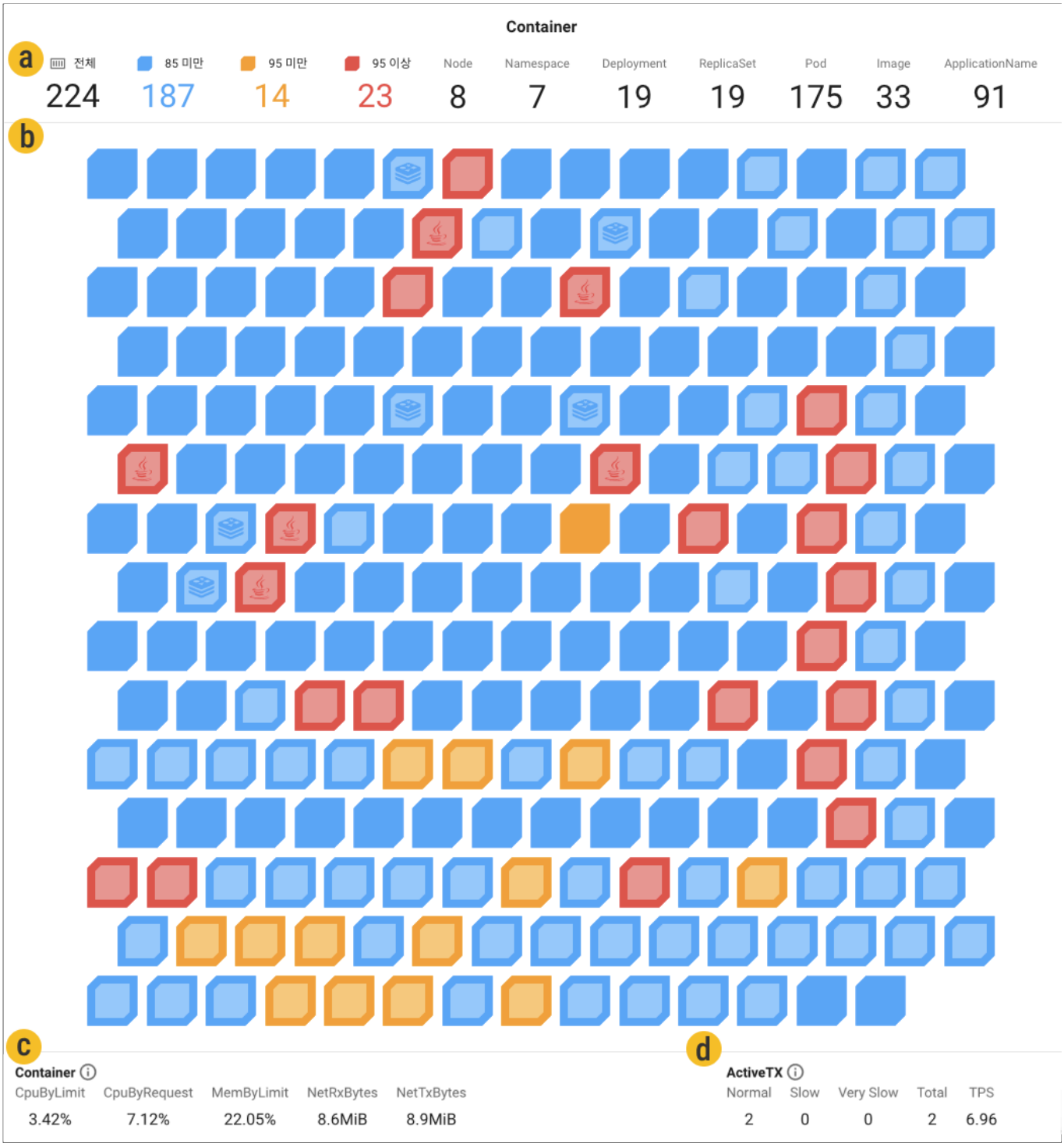
실시간 조회 및 과거 조회

컨테이너 맵은 기본적으로 실시간 조회 기반이지만 여러 시점에 따른 조회 기능을 제공합니다. 실시간 조회 변경은 화면 상단 왼쪽의 시간 선택자를 이용하세요.



조회 상태	설명
실시간 조회 (LIVE)	기본 모드 10초 인터벌로 실시간 상태 변경
일시정지 조회 (PAUSE)	실시간 상태 변경 OFF 현재 상태를 자세히 확인할 때 사용
과거시점 조회	사용자가 과거 특정 시점을 지정하여 확인 년/월/일/시/분 지정 가능

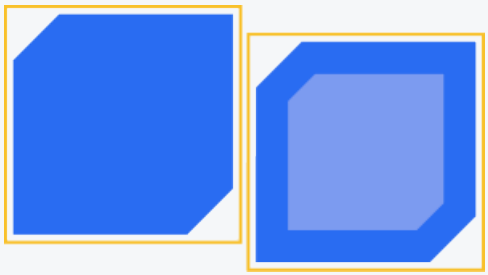
3 컨테이너 맵 차트


Container
a
전체
85 미만
95 미만
95 이상
Node
Namespace
Deployment
ReplicaSet
Pod
Image
ApplicationName
224
187
14
23
8
7
19
19
175
33
91
b
c
Container
CpuByLimit
CpuByRequest
MemByLimit
NetRxBytes
NetTxBytes
3.42%
7.12%
22.05%
8.6MiB
8.9MiB
d
ActiveTX
Normal
Slow
Very Slow
Total
TPS
2
0
0
2
6.96

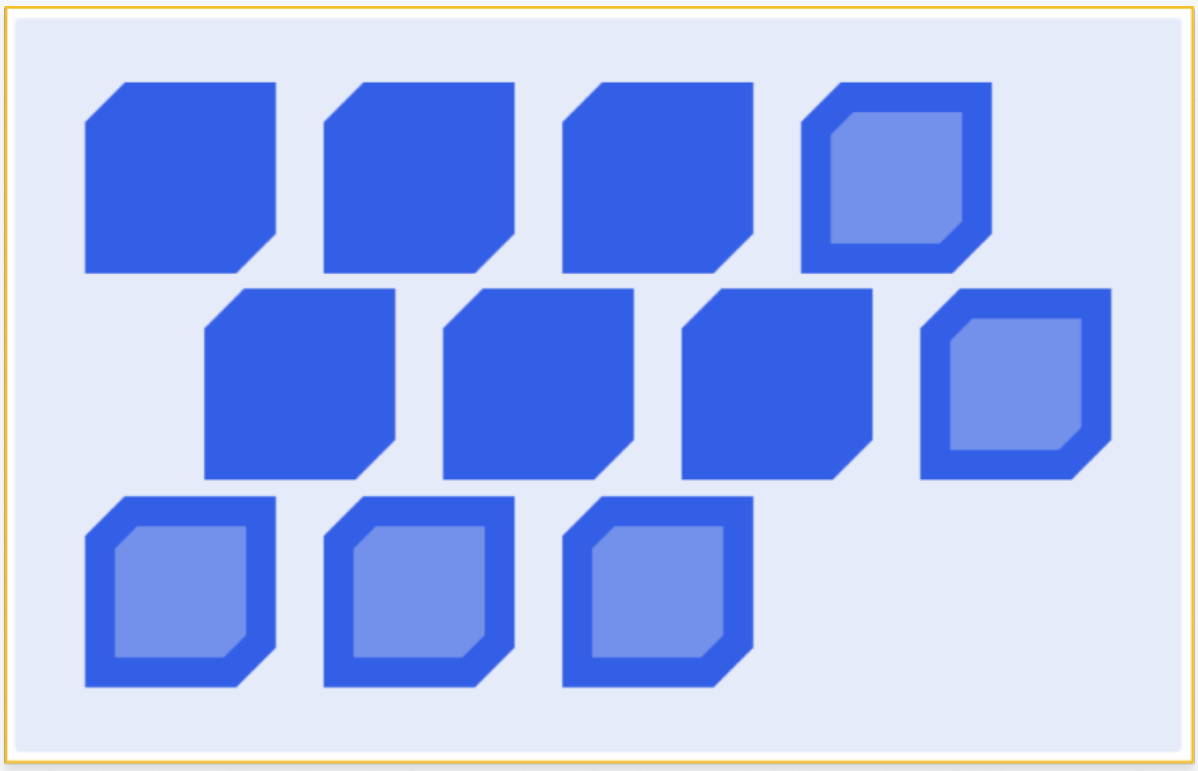
- a 그룹화된 대상 전체 개수 및 주요 리소스 개수를 나타냅니다.
- b 각 개체의 모양과 색상은 컨테이너 또는 Pod의 특성과 상태를 나타냅니다.
  - **외부 프로젝트 연동**

    블록 내에 로고를 통해 연동된 외부 프로젝트를 확인할 수 있습니다. 자세한 내용은 다음 문서를 참조하세요.
  - **색상**

    컨테이너 또는 Pod의 상태를 의미합니다. Normal, Warning, Critical 외 상태에 대한 자세한 내용은 다음 문서를 참조하세요.
  - **테두리**
    - 개체 중에는 테두리가 표현된 것이 있습니다. 이는 컨테이너에 Java, Node.js, Python 등 마이크로 에이전트를 설치한 대상입니다.
    - 만약 1개 Pod 내에 2개 컨테이너가 있고, 2개 중 1개 컨테이너에만 마이크로 에이전트를 설치했더라도 테두리가 표시됩니다. 즉 Pod 내 컨테이너 중에 하나라도 마이크로 에이전트가 설치되어 있다면 테두리 표시가 됩니다.

> ⓘ **블록과 마이크로 블록**
>
> 
>
> **컨테이너 맵**에서는 다각형 하나를 **블록**(Block)이라고 합니다. 이때 에이전트가 있는 블록(블록 안에 블록)은 **마이크로 블록**(Micro Block)이라고 합니다.

> ⓘ **그룹**



컨테이너 맵 화면에서 전체 블록을 선택한 경우 해당 다중 블록을 그룹이라고 합니다.

- c 그룹화된 대상의 CPU 및 Memory 관련 지표를 나타냅니다.
- d 그룹화된 대상 내부 애플리케이션의 트랜잭션 정보를 나타냅니다.

컨테이너 맵 대상 정보

사용자는 관심있는 컨테이너 또는 Pod를 선택해 요약 정보 및 상세 정보를 확인할 수 있습니다. 해당 정보에는 다음과 같은 핵심 모니터링 데이터를 포함하고 있습니다.

- **대상 정보**

  블록 클릭으로 대상 특정 시 선택한 대상과 관련된 리소스 정보 및 해당 메뉴 링크 이동을 제공합니다.

- **리소스 관계**

선택한 대상과 관련된 리소스 정보를 시작적으로 표현한 관계 차트로 제공합니다.

- **트레이스**

  컨테이너 안에 와탭 애플리케이션 에이전트가 설치된 경우 대시보드 상에서 실시간으로 확인할 수 있도록 해당 애플리케이션의 트랜잭션 트레이스 정보를 제공합니다.

- **메트릭스**

  선택한 컨테이너 또는 Pod의 CPU 사용량, 메모리 사용량, 네트워크 트래픽 등의 메트릭 정보를 대시보드에서 제공합니다.

- **로그**

  각 컨테이너의 표준 출력 로그(Standard Output) 및 컨테이너와 애플리케이션에서 생성하여 저장된 로그 정보를 제공합니다.

- **이벤트**

  쿠버네티스에서 발생한 이벤트와 와탭 이벤트 정보를 제공합니다.

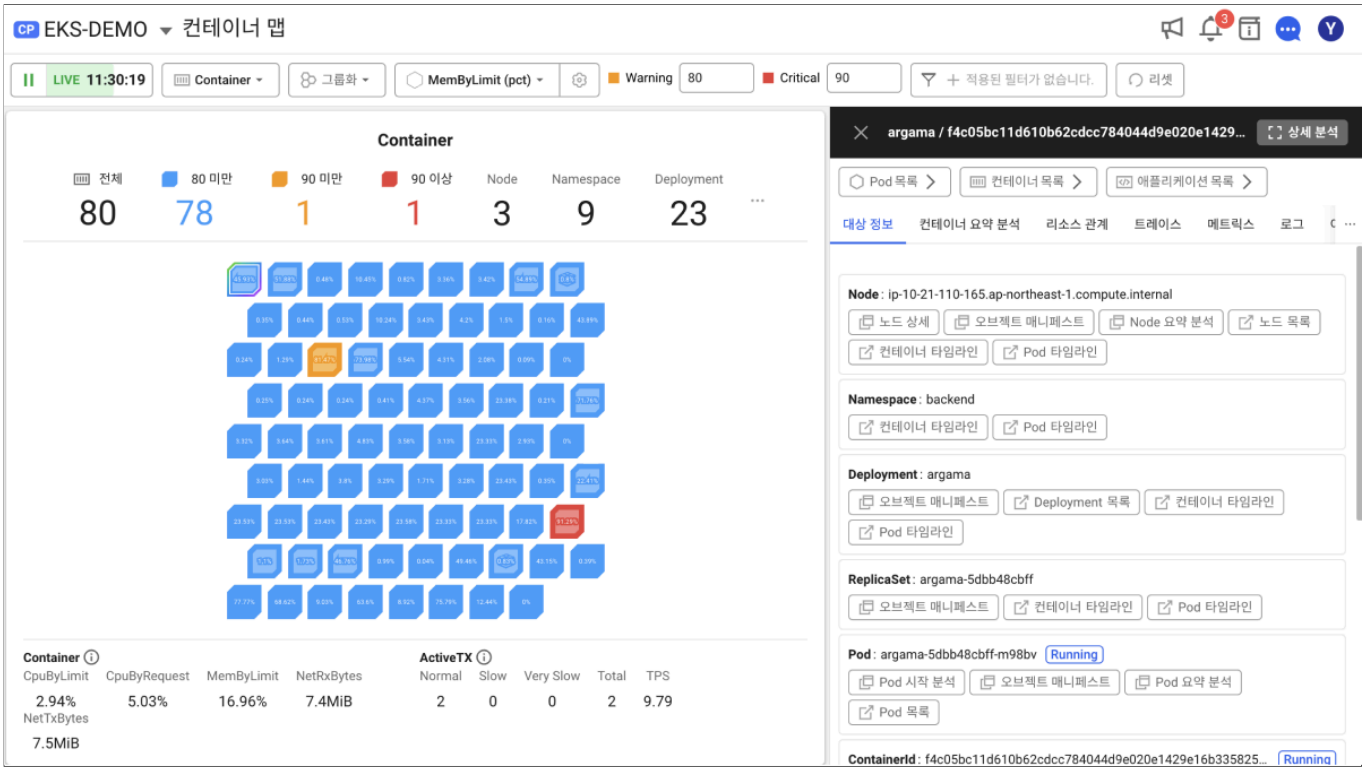
- **호출 정보**

  선택한 컨테이너 내에서 실행 중인 애플리케이션의 호출 흐름을 시각화한 차트를 제공합니다.

이러한 데이터들을 기반으로 사용자는 쿠버네티스 환경의 Observability를 확보할 수 있습니다. 다음의 **요약 보기** 및 **상세 분석** 화면 모두 위 4개 항목을 중심으로 정보를 제공합니다.

요약 보기

컨테이너 맵에서 특정 개체 혹은 개체 집합을 클릭하세요. 오른쪽에 나타나는 **요약 보기** 화면을 통해 해당 객체의 요약 정보를 확인할 수 있습니다.



- **대상 정보**

  선택한 Pod 또는 컨테이너에 대한 기본 정보를 제공합니다. 해당 리소스가 실행 중인 노드, 소속된 네임스페이스 등 기본적인 대상 정보를 확인하고 목록 메뉴로 이동할 수 있습니다.

ⓘ
- **Node**: 노드 상세, 오브젝트 매니페스트, Nod 요약 분석 정보를 확인하고 노드 목록 메뉴로 이동할 수 있습니다.
- **Deployment**: 오브젝트 매니페스트 정보를 확인하고 Deployment 목록 메뉴로 이동할 수 있습니다.
- **ReplicaSet**: 오브젝트 매니페스트 정보를 확인할 수 있습니다.
- **Pod**: Pod 시작 분석, 오브젝트 매니페스트, Pod 요약 분석 정보를 확인하고 Pod 목록 메뉴로 이동할 수 있습니다.
- **ContainerId**: **컨테이너 정보**, 컨테이너 요약 분석 정보를 확인하고 컨테이너 목록 메뉴로 이동할 수 있습니다.
- **Image**: **컨테이너 이미지** 정보를 확인하고 컨테이너 이미지 메뉴로 이동할 수 있습니다.
- **ServiceName**: 오브젝트 매니페스트 정보를 확인하고 Service 목록 메뉴로 이동할 수 있습니다.
- **Volume**: 컨테이너가 사용하는 볼륨과 마운트 경로 정보를 확인할 수 있습니다.
  - Pod에서 사용하는 a 볼륨의 이름과 b 볼륨 타입을 확인할 수 있습니다.

예, **Volume** a containerddomainsocket b HostPath

- **HostPath** 볼륨 타입을 사용할 경우 c 호스트 시스템의 파일 경로를 확인할 수 있습니다. 이 경로는 파드와 호스트 시스템 간의 파일 시스템을 연결하는 경로입니다.

  예, **Host Path** c /run/containerd/containerd.sock

- **리소스 관계**

  컨테이너 맵의 **리소스 관계** 탭을 통해 쿠버네티스 리소스 간 연결 관계를 시각적으로 확인할 수 있습니다.

- **트레이스**

  컨테이너 맵의 **트레이스** 탭을 통해 컨테이너 상에서 실행되는 애플리케이션의 개별 트랜잭션을 추적할 수 있습니다.

- **메트릭스**

  컨테이너 맵의 **메트릭스** 탭을 통해 각 컨테이너의 자원 할당량, 사용량 등의 각종 지표 데이터를 실시간으로 확인할 수 있습니다.

- **로그**

  컨테이너 맵의 **로그** 탭을 통해 각 컨테이너의 표준 출력 로그 및 컨테이너 상의 애플리케이션이 생성하는 로그를 통합 관리, 조회할 수 있습니다.

- **이벤트**

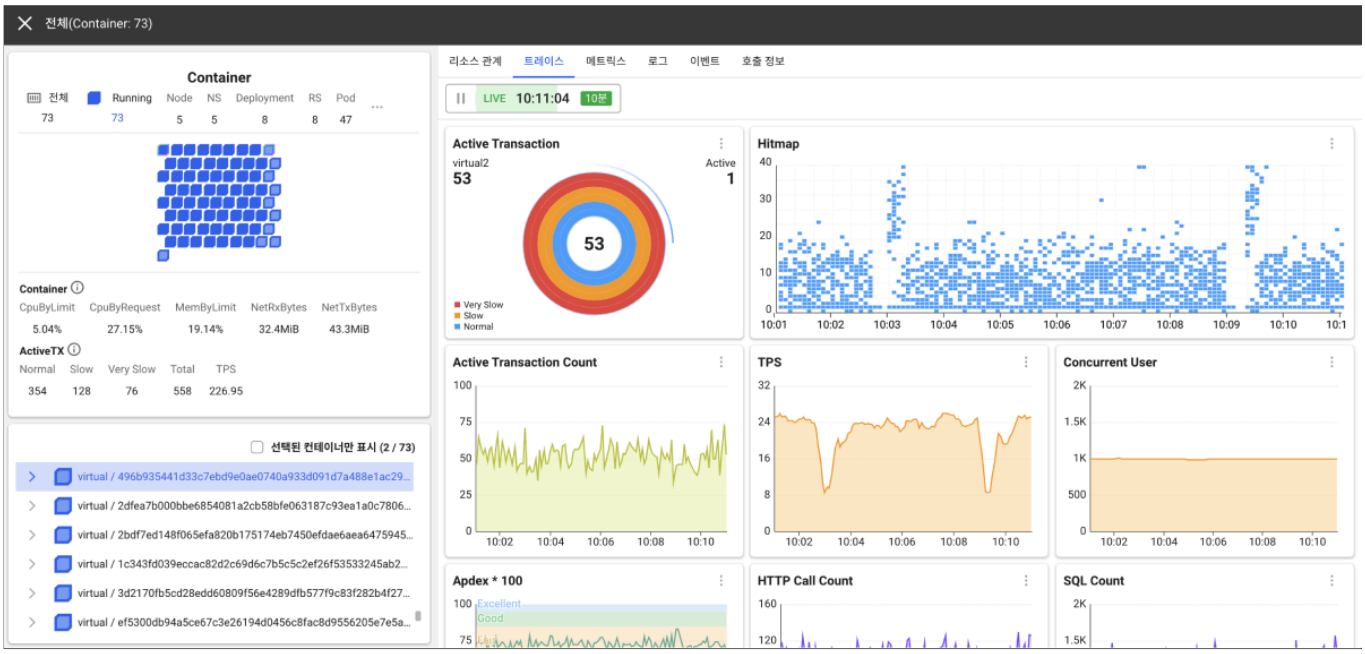
  컨테이너 맵의 **이벤트** 탭을 통해 쿠버네티스 클러스터에서 발생한 이벤트 및 와탭의 경고 기능을 이용한 사용자 임계치 기반 이벤트의 통합 조회가 가능합니다.

- **호출 정보**

  컨테이너 맵의 **호출 정보** 탭을 통해 쿠버네티스 컨테이너 상에서 실행 중인 애플리케이션의 호출 흐름과 외부 호출 대상의 상태를 확인할 수 있습니다.

상세 분석

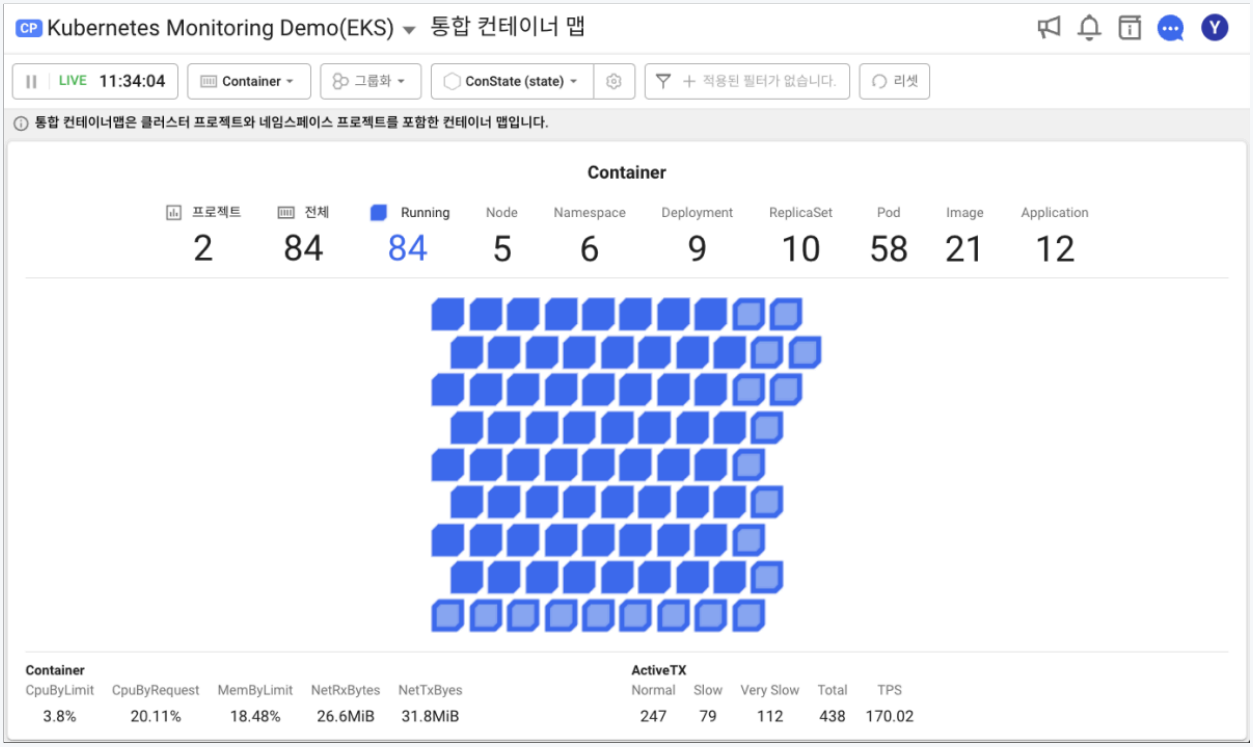
요약 보기 화면의 오른쪽 위에 ⛶ **상세 분석** 버튼을 클릭하세요. **상세 분석** 화면으로 전환합니다. **상세 분석** 화면 구성에 대한 자세한 안내는 다음 문서를 참조하세요.



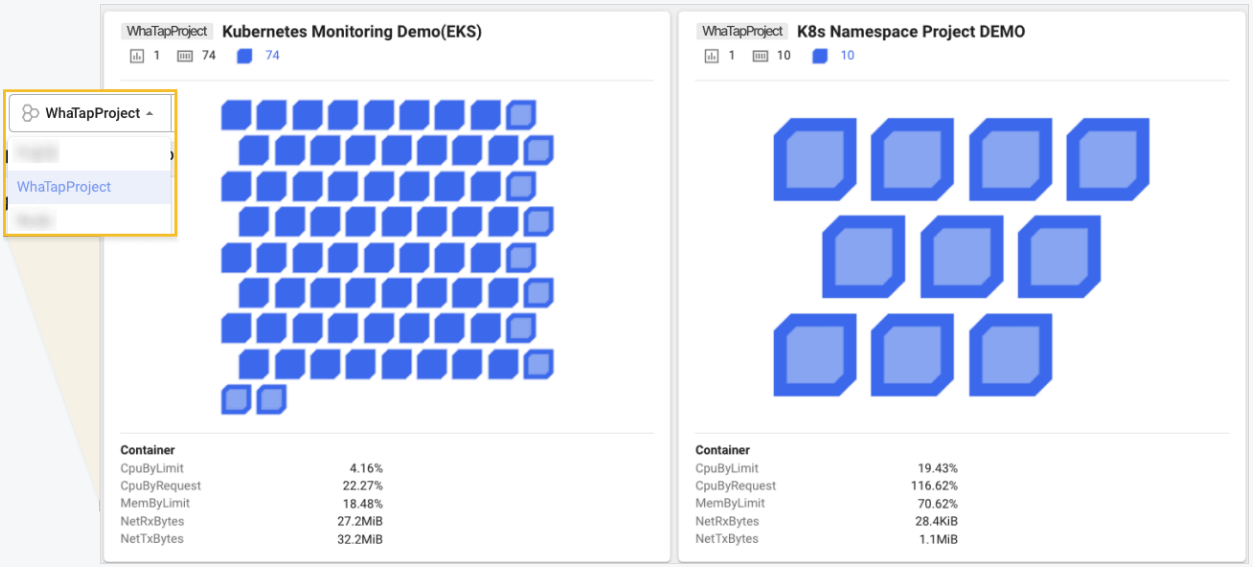
통합 컨테이너 맵

홈 화면 > **클러스터** > **통합 컨테이너 맵**

클러스터 프로젝트(CP)와 네임스페이스 프로젝트(Namespace)를 포함한 컨테이너 맵을 조회하려면 **통합 컨테이너 맵**을 확인하세요.



통합 컨테이너 맵은 프로젝트별 **그룹화** 옵션을 추가로 제공합니다. **그룹화** 옵션 중 **WhaTapProject** 그룹화를 통해 한 화면에서 클러스터 프로젝트 및 네임스페이스 개별 프로젝트 내 컨테이너 맵을 다음과 같이 조회할 수 있습니다.



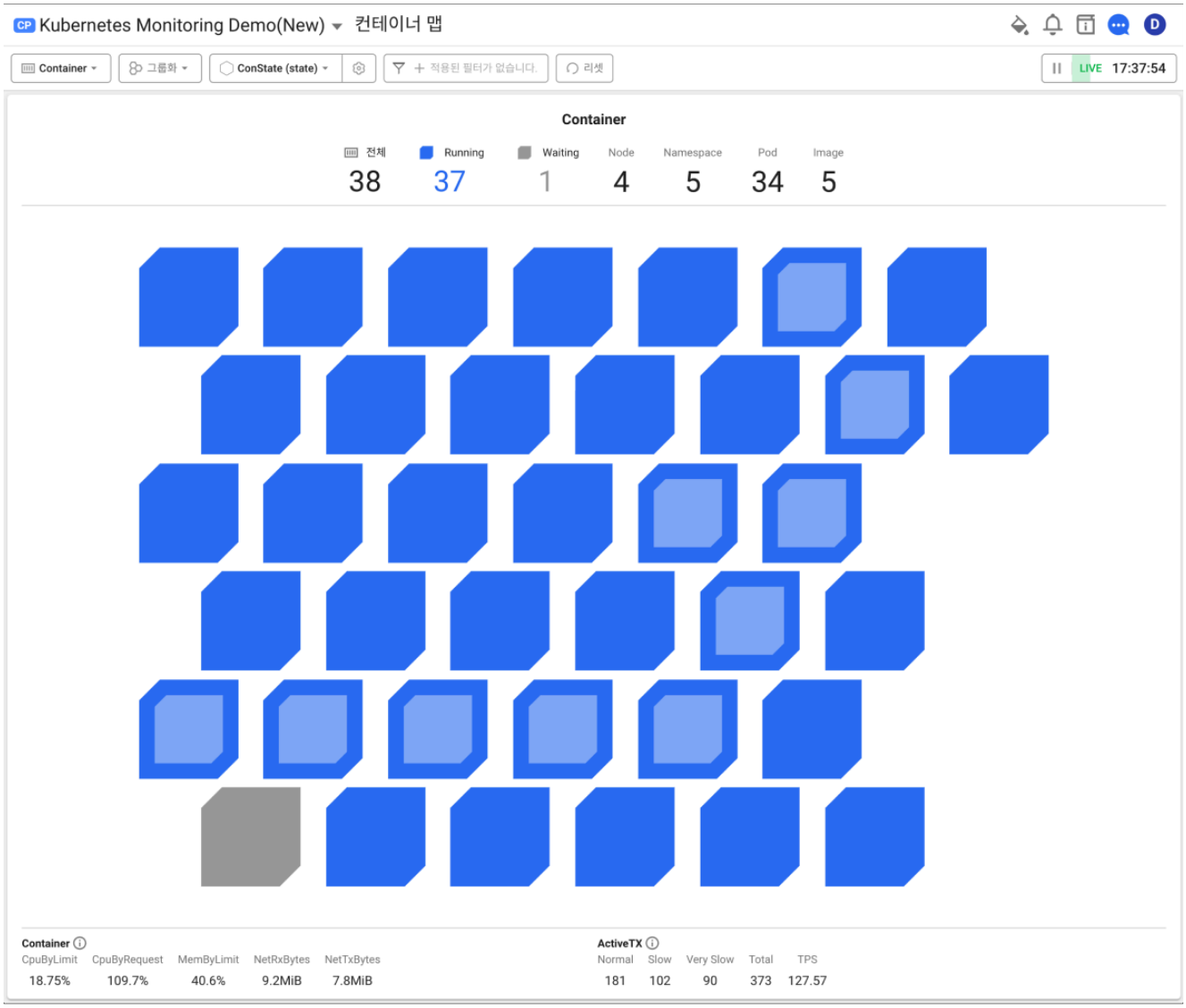
컨테이너 뷰, Pod 뷰

카테고리 선택 (컨테이너 뷰, Pod 뷰)

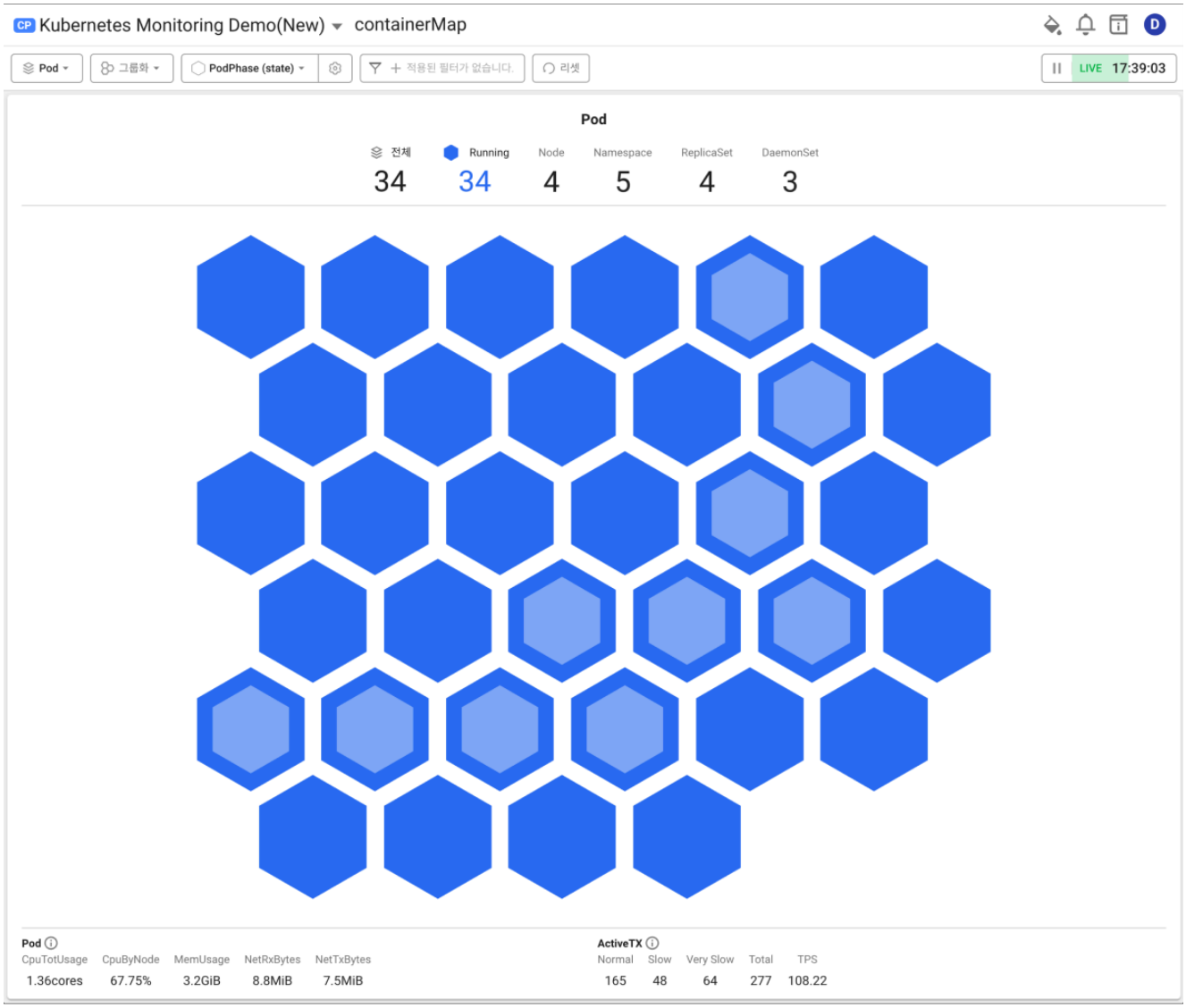
컨테이너 맵 메뉴에 처음 진입하면 사용자 클러스터 프로젝트 내의 개별 컨테이너 혹은 Pod가 모두 펼쳐진 것을 확인할 수 있습니다. 1개 Pod 내에 2개 이상의 컨테이너가 실행될 수 있기 때문에 컨테이너, Pod 각각에 대한 뷰를 가지고 있어야 합니다.

와탭 쿠버네티스의 **컨테이너 맵** 메뉴에서는 **컨테이너 기준 뷰**, **Pod 기준 뷰**의 두가지 모드를 모두 지원하기 때문에 컨테이너 단위의 모니터링, Pod 단위의 모니터링 등 상황에 맞는 모니터링이 가능합니다.

- **컨테이너 기준 뷰**


CP Kubernetes Monitoring Demo(New)
컨테이너 맵
Container
그룹화
ConState (state)
적용된 필터가 없습니다.
리셋
LIVE 17:37:54
Container
전체 Running Waiting Node Namespace Pod Image
38 37 1 4 5 34 5
Container
CpuByLimit CpuByRequest MemByLimit NetRxBytes NetTxBytes
18.75% 109.7% 40.6% 9.2MiB 7.8MiB
ActiveTX
Normal Slow Very Slow Total TPS
181 102 90 373 127.57

- **Pod 기준 뷰**



컨테이너 기준 뷰와 **Pod 기준 뷰**는 **컨테이너 맵** 메뉴 화면 위에 사용자 설정을 통해 전환할 수 있습니다.

ⓘ 구버전 쿠버네티스 에이전트를 사용한다면 Pod 기준 뷰를 확인할 수 없습니다. 에이전트 업그레이드를 수행하세요.

Master

```
kubectl set image deployment/whatap-master-agent whatap-master-agent=whatap/kube_mon:{에이전트 버전} -n whatap-monitoring
```

Node

```
kubectl set image daemonset/whatap-node-agent whatap-node-agent=whatap/kube_mon:{에이전트 버전} -n whatap-monitoring
```

대상 그룹화

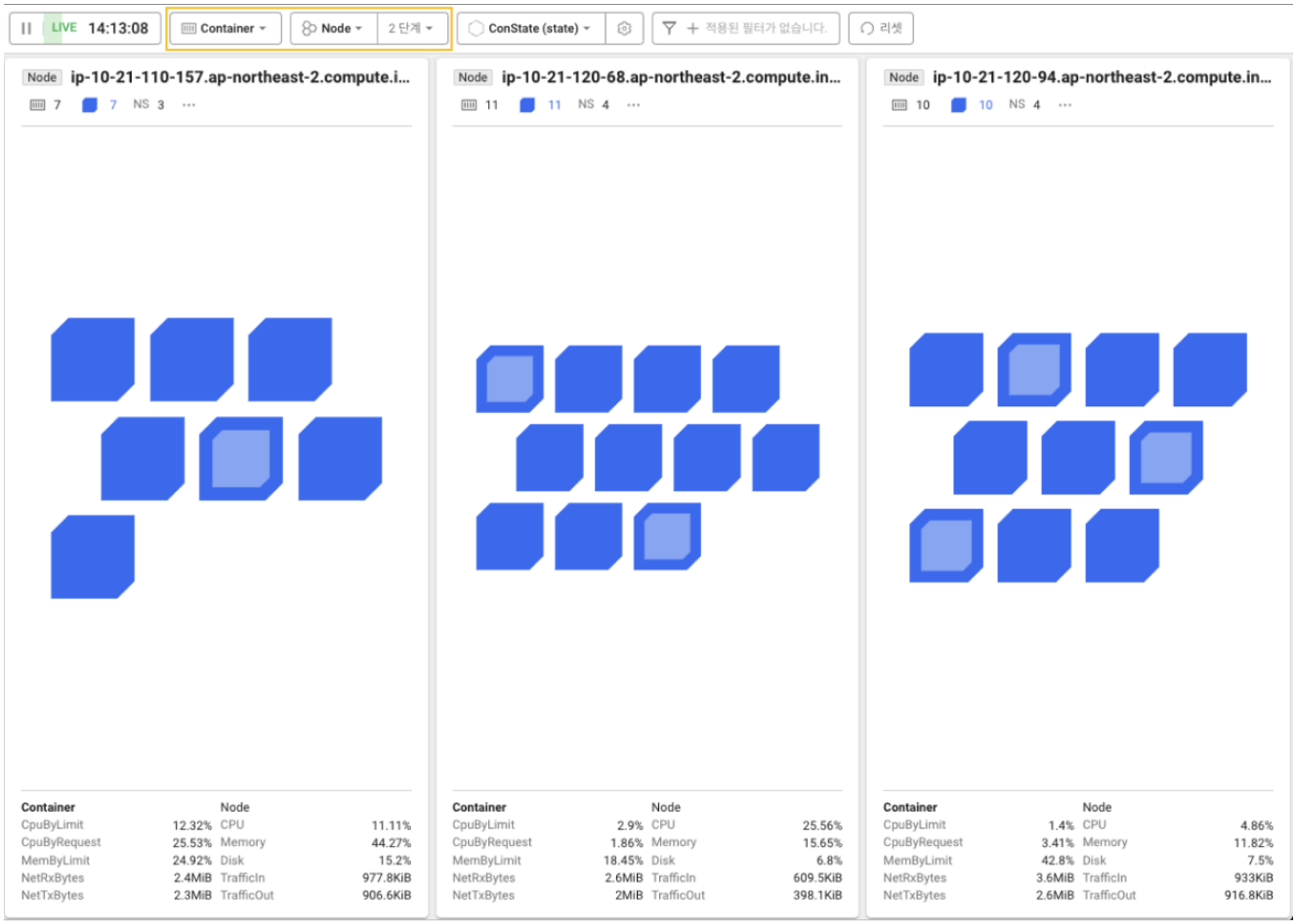
사용자 관심사에 따른 다양한 그룹화

기본 **컨테이너 맵** 메뉴의 화면을 보면 모든 컨테이너 또는 Pod가 펼쳐져 있습니다. 그런데 쿠버네티스의 컨테이너/Pod를 관찰할 때는 특정 기준으로 대상을 선별해 모니터링하는 경우가 많습니다. 물론 대상을 바라보는 기준은 매우 다양합니다. 쿠버네티스 환경을 대하는 사용자의 관심사가 모두 동일하지 않기 때문입니다.

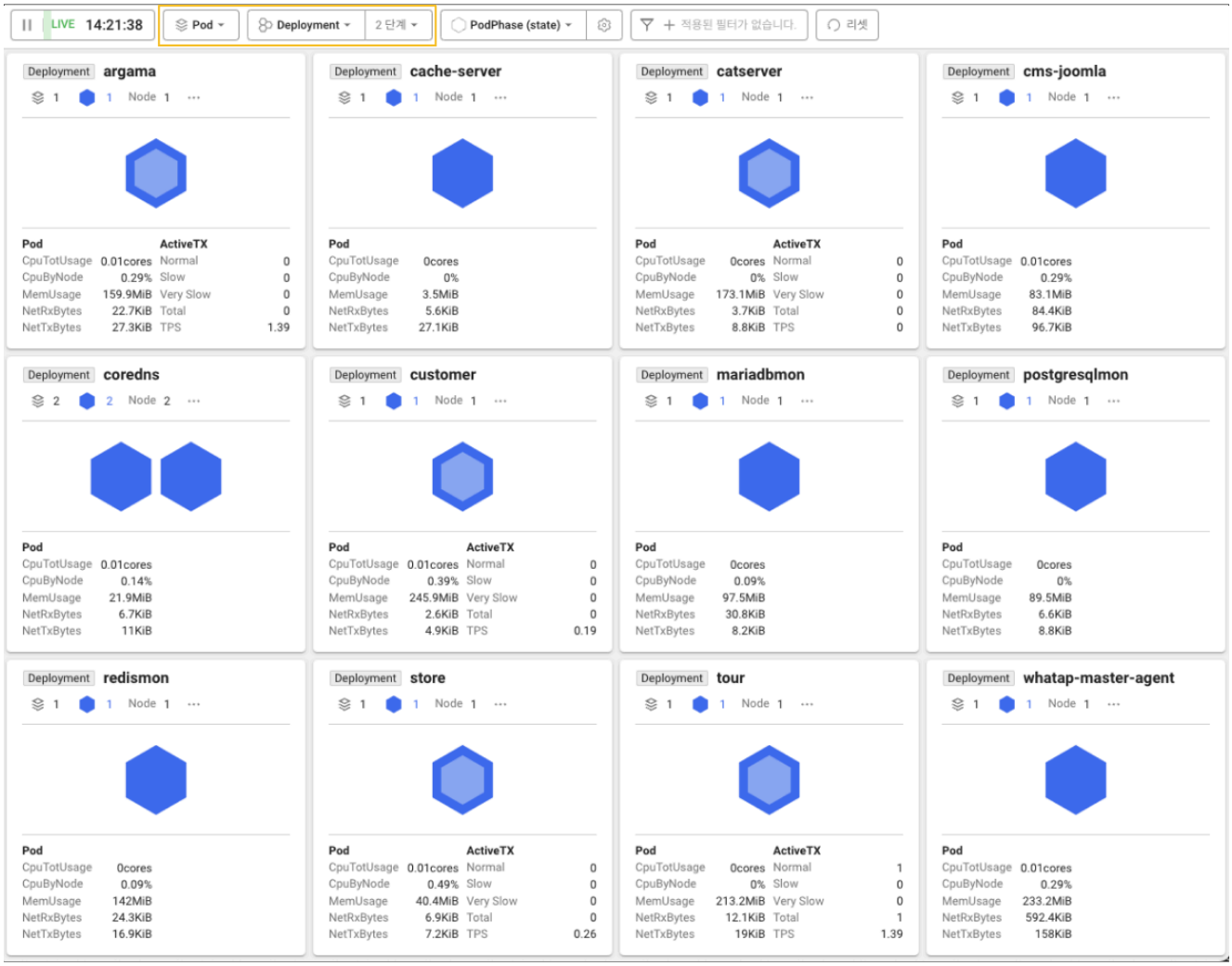
서버 담당자는 쿠버네티스 리소스를 실행하기 위한 Node 상태에 관심이 많습니다. 반면 특정 서비스의 애플리케이션 담당자는 Deployment나 개별 Pod, 혹은 컨테이너 상태에 관심이 많습니다. 만약 신규 버전 배포 이후라면 Deployment 전체적인 상태뿐만 아니라 신규 이미지를 사용하는 컨테이너들을 모니터링하고자 할 것입니다.

그래서 이러한 다양한 사용자 니즈를 충족하고자 제공하는 기능이 **그룹화**입니다.

- **컨테이너 기준 뷰** + **Node 그룹화**



- **Pod 기준 뷰 + Deployment 그룹화**



주요 그룹화 속성

컨테이너 맵의 뷰에 따라 지원하는 그룹화 속성은 다음과 같습니다.

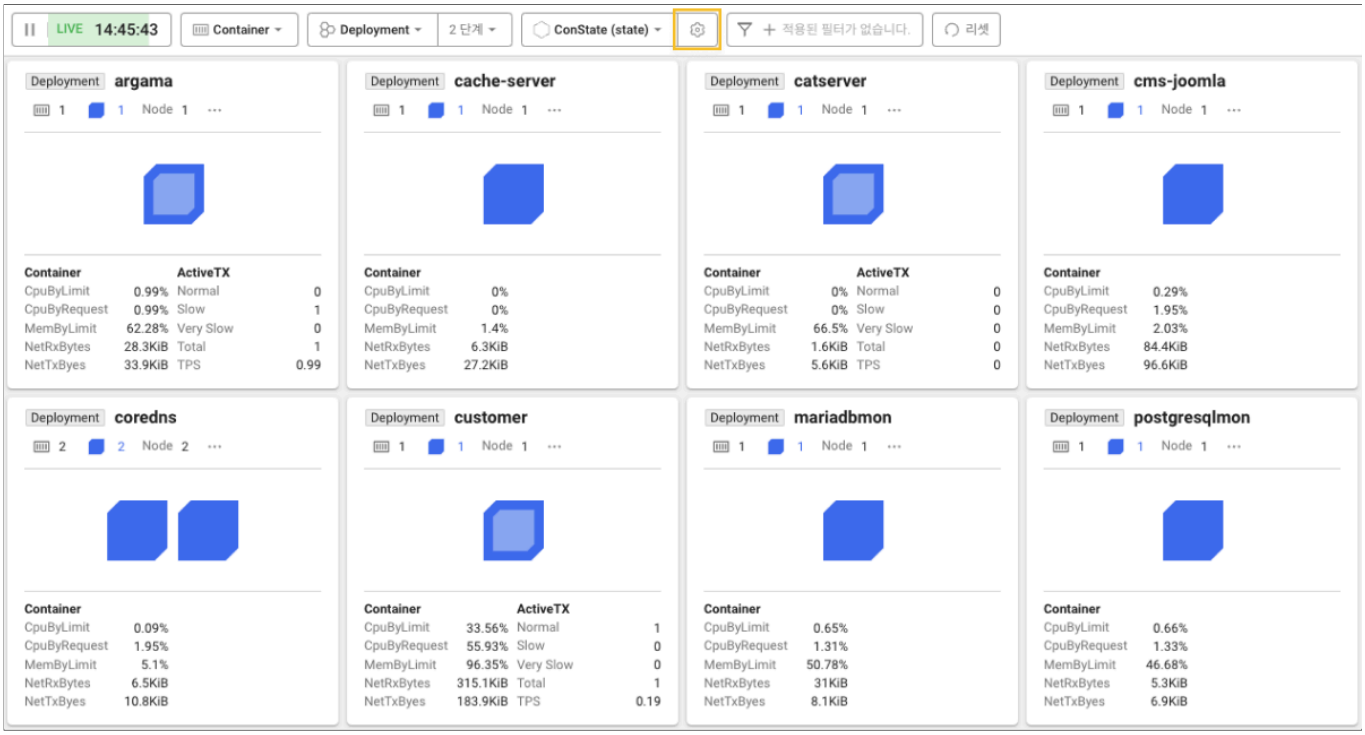
기준 뷰	그룹화 속성
Container	미설정 (All) Node Namespace Deployment

기준 뷰	그룹화 속성
	ReplicaSet ReplicationController Pod ContainerName Image
Pod	미설정 (All) Node Namespace Deployment ReplicaSet ReplicationController DemonSet StatefulSet

ⓘ 화면 위에 설정한 그룹화 설정(속성 필드 및 필터링 옵션 포함) 정보는 사용자 브라우저 내에 저장됩니다. 만약 저장된 설정을 초기화하고 싶다면 ↻ **리셋** 버튼을 선택하세요.

그룹화 & 카드

컨테이너 맵은 1개 이상의 카드로 구성되어 있습니다. 만약 그룹화를 적용하지 않은 상태라면 프로젝트 내 전체 컨테이너/Pod가 하나의 큰 카드 내에 표현됩니다. 그룹화를 적용하면 각 그룹 단위로 카드를 구성합니다.



위 화면은 Deployment를 기준 속성으로 하여 그룹화를 적용한 것입니다. 각 Deployment 단위로 카드가 나열되어 있음을 알 수 있습니다.

그룹화 적용 시 기준 속성으로 그룹화 할 수 없는 대상은 기본적으로 표시되지 않습니다. 하지만 해당 대상들도 화면에 표시할 수 있는 방법이 있습니다. 화면 위에 ⚙ 설정 버튼을 선택하세요. **알 수 없는 그룹 숨기기** 체크박스를 해제하면 그룹화에 해당하지 않는 대상들만 모아 별도의 카드로 표현합니다.

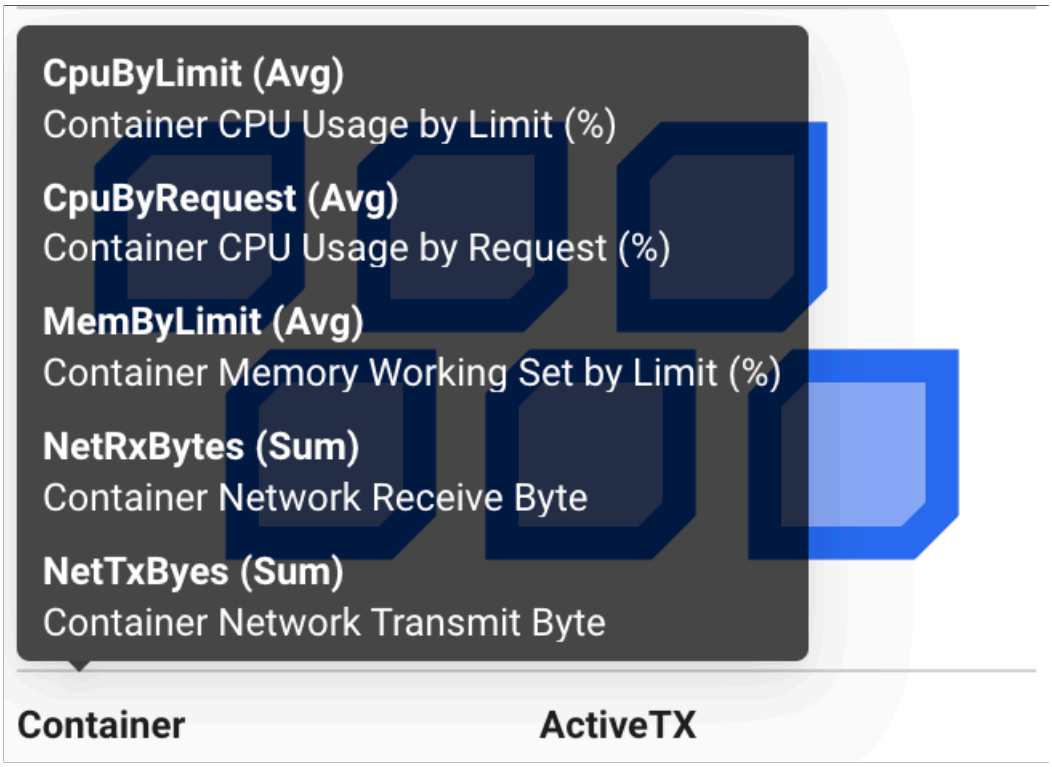
카드별 메트릭 데이터

각 카드 아래에는 그룹화된 대상들에 대한 메트릭 데이터를 표시합니다. 다음은 제공하는 지표 정보입니다.

기준 뷰	리소스 지표	애플리케이션 지표
Container	CpuByLimit CpuByRequest MemByLimit NetRxBytes NetTxBytes	Normal TX Slow TX Very Slow TX Total TX TPS

기준 뷰	리소스 지표	애플리케이션 지표
Pod	CpuTotUsage CpuByNode MemUsage NetRxBytes NetTxBytes	

사용자의 커서가 **Container** (또는 **Pod**), **ActiveTX** 위에 올라가 있으면 각 메트릭에 대한 설명을 팝업으로 확인할 수 있습니다.



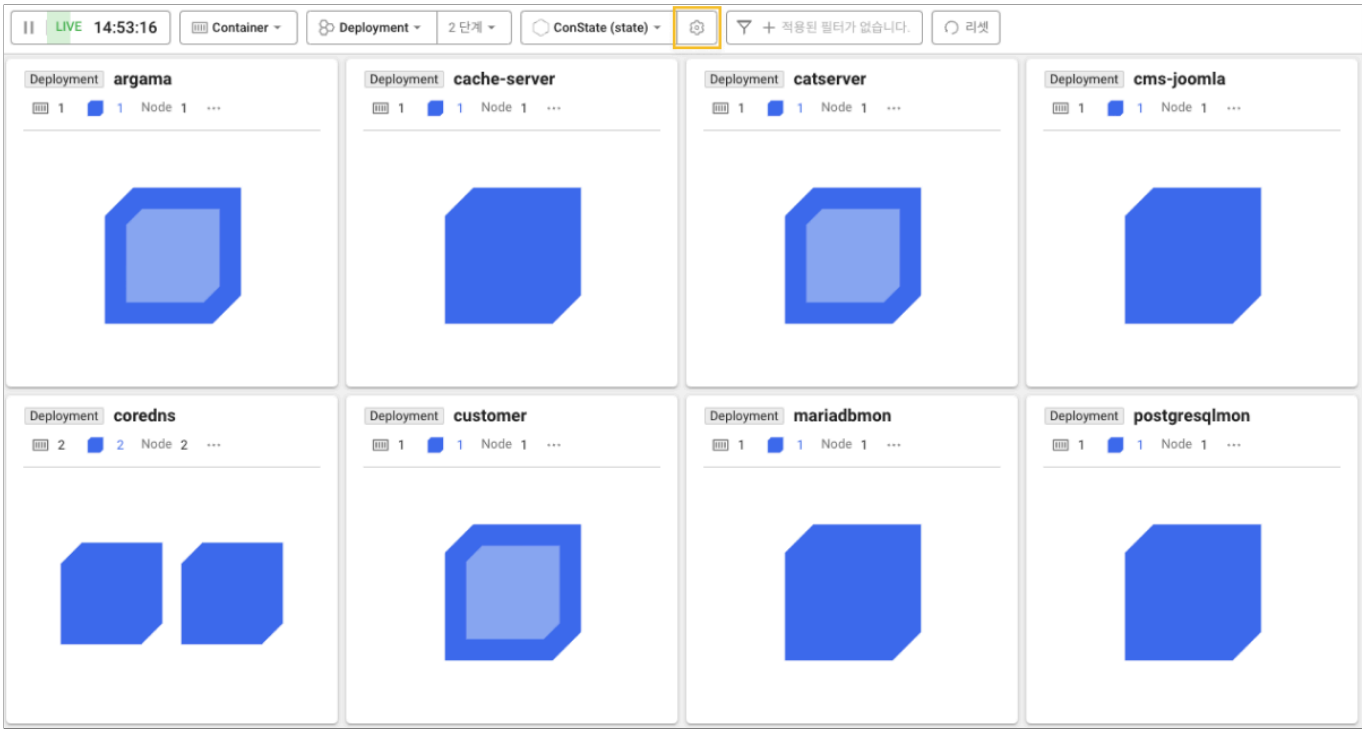
- 팝업에 표시된 지표 중 (Avg)가 붙은 항목은 카드 아래에 표시된 같은 이름의 지표 수치가 그룹 내 개별 대상이 가진 값의 평균 값이라는 의미입니다.
- 팝업에 표시된 지표 중 (Sum)이 붙은 항목은 카드 아래에 표시된 같은 이름의 지표 수치가 그룹 내 개별 대상이 가진 값의 합산 값이라는 의미입니다.

카드의 메트릭 데이터 감추기

카드 아래에 메트릭 데이터는 숨기고 오직 차트에만 집중할 수 있습니다. 화면 위에 ⚙ 설정 버튼을 선택하세요. **카드 하단에**

메트릭스 표시 체크박스를 해제하세요. 이 기능은 카드 수가 많을 때 유용합니다.

다음과 같이 메트릭 데이터를 숨길 수 있습니다.

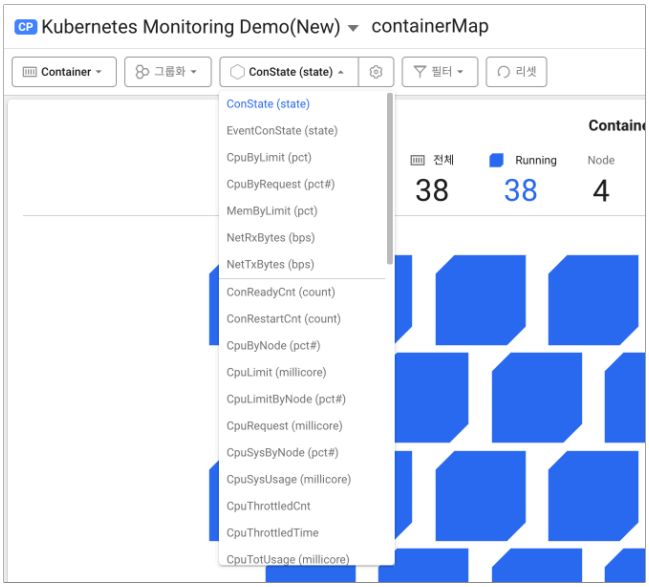


컨테이너 맵 지표와 상태

차트 지표 사용자화

와탭 쿠버네티스 모니터링 서비스의 **컨테이너 맵** 메뉴에서 제공하는 기본 차트 속성은 컨테이너 또는 Pod의 상태(State)입니다. 메트릭 이름으로는 **ConState**(Container State)입니다. 하지만 기본 상태가 아닌 다른 기준으로 맵을 구성할 필요가 있습니다. 예를 들어, 기본 State 상태가 아닌 CPU 사용량(millicore), 메모리 사용률(%), 네트워크 사용량(byte) 등을 기준으로 각 대상들의 상태를 파악해야 하는 경우도 있습니다.

와탭은 컨테이너의 차트 지표 변경 기능을 제공합니다. 변경 가능한 차트 지표 값은 와탭 쿠버네티스가 수집하는 메트릭스(지표)와 같습니다.



> ⓘ 화면 위에 설정한 그룹화 설정(속성 필드 및 필터링 옵션 포함) 정보는 사용자 브라우저 내에 저장됩니다. 만약 저장된 설정을 초기화하고 싶다면 ↻ **리셋** 버튼을 선택하세요.

지표 임계치 설정 및 상태 색상

대상에 부여한 지표의 임계치 설정으로 각 대상의 상태를 색상으로 분류할 수 있습니다. 지표의 단위가 숫자인 경우(pct, bps, millicore)에 임계치 설정이 가능합니다.

Normal, Warning, Critical 등 상태별 색상 및 의미는 다음과 같습니다.

상태	설명
Normal	정상 상태 구간 표시 (~ Warning 미만 구간을 3등분)
Warning	Warning 임계치 구간 내에 있음을 표시 (Warning 초과 ~ Critical 이상)
Critical	Critical 임계치 구간 내에 있음을 표시 (Critical 초과 ~)
Dead	컨테이너 종료 진행 상태
Wait	컨테이너 시작 진행 상태
Pause	컨테이너 일시 중단 상태

이벤트 기반 컨테이너 상태

컨테이너 맵 메뉴에서 컨테이너 뷰는 `EventConState` 라는 상태 값을 제공합니다. `EventConState` 는 `Event-based Container State` 로 이벤트가 반영된 컨테이너 상태를 나타냅니다. `EventConState` 에 반영된 이벤트 종류는 다음과 같습니다.

이벤트	레벨
Kubernetes Event	Warning
WhaTap Event	Warning, Critical

- `Kubernetes Event` 는 쿠버네티스가 제공하는 이벤트 정보입니다. 사용자의 쿠버네티스 환경에서 `kubectl get events` 명령을 통해 확인 가능한 이벤트 정보와 동일합니다.
- `WhaTap Event` 는 사용자가 와탭 이벤트 기능을 통해 설정한 이벤트입니다.

`EventConState` 는 다음과 같은 방식으로 결정됩니다.

상태	상태값	원인
Critical	50	WhaTap 마이크로 에이전트에서 Critical 이벤트 발생 WhaTap 노드 에이전트에서 Critical 이벤트 발생 컨테이너 상태 OOMKILLED 전환
Warning	30	Kubernetes에서 Warning 이벤트 발생 WhaTap 마이크로 에이전트에서 Warning 이벤트 발생 WhaTap 노드 에이전트에서 Warning 이벤트 발생 컨테이너 상태 RESTARTING 전환
Dead	23	컨테이너 상태 DEAD 전환
Pause	22	컨테이너 상태 PAUSE 전환
Waiting	21	컨테이너 상태 WAITING 전환

상태	상태값	원인
Normal	10	컨테이너 상태 Normal 전환

와탭 이벤트 중에는 **이벤트 상태가 해소되면 추가 알림**을 설정한 이벤트가 있습니다. 내부적으로 두가지 유형의 이벤트를 다음과 같이 구분합니다.

- 상태 해결을 추적하지 않는 이벤트 : `stateless` 이벤트
- 상태 해결 시 추가 알림을 설정한 이벤트 : `stateful` 이벤트

컨테이너에서 발생한 `stateless` , `stateful` 이벤트는 `EventConState` 상태를 결정하는데 다음과 같이 사용합니다.

- `stateless` 이벤트 : 컨테이너에서 발생한 최근 1분 이내의 `stateless` 이벤트를 `EventConState` 상태에 반영
- `stateful` 이벤트 : 발생 시점부터 종료 시점까지의 `stateful` 이벤트를 `EventConState` 상태에 반영

외부 프로젝트 연동과 설정

> ⓘ 와탭 쿠버네티스 에이전트 1.7.16 버전 이상이 필요합니다.

와탭은 다양한 모니터링 서비스를 프로젝트 단위로 제공합니다. **외부 프로젝트 연동** 기능을 통해 쿠버네티스 프로젝트의 Pod와 연관된 외부 프로젝트를 연동할 수 있습니다.

외부 프로젝트 연동을 위해서는 쿠버네티스 에이전트 설정 및 연동 대상 Pod 설정이 필요합니다. 쿠버네티스 프로젝트와 외부 프로젝트를 연동하여 통합적인 모니터링 체계를 구축해 보세요.

외부 프로젝트 연동

외부 프로젝트 연동 기능 활성화 시 다음과 같이 연동된 컨테이너 및 Pod에 해당 프로젝트 로고가 표시됩니다. 이를 통해 어떤 외부 프로젝트와 연동되었는지 시각적으로 쉽게 식별할 수 있습니다. 또한 해당 블록을 클릭하여 요약 보기 패널 내 버튼을 통해 해당 프로젝트로 이동할 수 있습니다.

Node ip-10-21-110-117.ap-northeast-1.compute.i...
23 22 1 NS 5 Deployment 5
0.09% 9.09% 4.39% 0% 0%
4.79% 3.21% 0% 2.03% 1.81%
0% 0% 3.03% 0.88% 0.92%
0% 0% 0% 0% 81.2%
6.29% 2.78% 13.87%
redis / adb18ac9ada667a1f3d180673e7db17b658a7e1...
상세 분석
Pod 목록
컨테이너 목록
대상 정보
리소스 관계
메트릭스
로그
이벤트
REDIS 프로젝트
Node : ip-10-21-110-117.ap-northeast-1.compute.internal

외부 프로젝트 설정

에이전트 설정

쿠버네티스 에이전트의 설치 YAML 파일에서 whatap-monitoring 네임스페이스의 whatap-node-agent DaemonSet 설정을 변경해야 합니다. 특히, whatap-node-agent 컨테이너의 env 섹션에 다음 환경 변수를 추가해야 합니다.

```
- name: "collect_foreign_project"
  value: "true"
```

예시

다음은 환경변수가 추가된 YAML 파일의 예시입니다. 실제 운영 환경의 YAML 파일과 많은 차이가 있을 수 있기에 예시를 참고하여 필요한 부분만 수정 후 사용하세요. 하이라이팅된 라인이 추가된 환경 변수입니다.

```
apiVersion: "apps/v1"
kind: "DaemonSet"
metadata:
  labels:
    name: "whatap-node-agent"
  name: "whatap-node-agent"
  namespace: "whatap-monitoring"
spec:
  selector:
    matchLabels:
      name: "whatap-node-agent"
  template:
    metadata:
      labels:
        name: "whatap-node-agent"
    spec:
      containers:
      - command:
        - "/data/agent/node/cadvisor_helper"
        - "-port"
        - "6801"
```

```
    env:
    - name: "NODE_NAME"
      valueFrom:
        fieldRef:
          fieldPath: "spec.nodeName"
    image: "whatap/kube_mon_dev:foreign-2"
    name: "whatap-node-helper"
    ports:
    - containerPort: 6801
      name: "helperport"
    resources:
      limits:
        cpu: "200m"
        memory: "350Mi"
      requests:
        cpu: "100m"
        memory: "100Mi"
    volumeMounts:
    - mountPath: "/rootfs"
      name: "rootfs"
      readOnly: true
    - mountPath: "/sys"
      name: "hostsys"
      readOnly: true
    - mountPath: "/dev/disk"
      name: "hostdiskdevice"
      readOnly: true
    - mountPath: "/run/containerd/containerd.sock"
      name: "containerddomainsocket"
  - command:
    - "/bin/entrypoint.sh"
    env:
    - name: "NODE_IP"
      valueFrom:
        fieldRef:
          fieldPath: "status.hostIP"
    - name: "NODE_NAME"
      valueFrom:
        fieldRef:
          fieldPath: "spec.nodeName"
    - name: "WHATAP_LICENSE"
```

```
      value: "x6050rs80f4eg-x1g0egm9rjb3mg-z7p9533a2ea87e"
    - name: "WHATAP_HOST"
      value: "13.124.11.223/13.209.172.35"
    - name: "WHATAP_PORT"
      value: "6600"
    - name: "HOST_PREFIX"
      value: "/rootfs"
    - name: "WHATP_MEM_LIMIT"
      valueFrom:
        resourceFieldRef:
          containerName: "whatap-node-agent"
          resource: "limits.memory"
    - name: "collect_foreign_project"
      value: "true"
    image: "whatap/kube_mon_dev:foreign-2"
    name: "whatap-node-agent"
    ports:
    - containerPort: 6600
      name: "nodeport"
    resources:
      limits:
        cpu: "200m"
        memory: "350Mi"
      requests:
        cpu: "100m"
        memory: "300Mi"
    volumeMounts:
    - mountPath: "/rootfs"
      name: "rootfs"
      readOnly: true
    - mountPath: "/bin/entrypoint.sh"
      name: "start-script-volume"
      readOnly: true
      subPath: "entrypoint.sh"
    - mountPath: "/whatap_conf"
      name: "whatap-config-volume"
  initContainers:
  - command:
    - "/data/agent/tools/whatap_debugger"
    - "run"
    image: "whatap/kube_mon_dev:foreign-2"
```

```
      name: "whatap-node-debug"
      volumeMounts:
      - mountPath: "/rootfs"
        name: "rootfs"
        readOnly: true
    serviceAccount: "whatap"
    tolerations:
    - effect: "NoSchedule"
      key: "node-role.kubernetes.io/master"
    - effect: "NoSchedule"
      key: "node-role.kubernetes.io/control-plane"
    volumes:
    - hostPath:
        path: "/"
      name: "rootfs"
    - hostPath:
        path: "/sys"
      name: "hostsys"
    - hostPath:
        path: "/dev/disk"
      name: "hostdiskdevice"
    - configMap:
        defaultMode: 448
        name: "node-start-script"
      name: "start-script-volume"
    - emptyDir: {}
      name: "whatap-config-volume"
    - hostPath:
        path: "/run/containerd/containerd.sock"
      name: "containerddomainsocket"
```

연동 대상 Pod 설정

연동하려는 Pod의 설정을 수정하세요. 예를 들어 Redis Pod를 연동하려는 경우 Redis 컨테이너의 env 섹션에 다음 환경 변수를 추가해야 합니다.

```
- name: "license"
  value: "here_is_whatap_license"
- name: "whatap_platform"
```

```
value: "redis"
```

- `here_is_whatap_license` : 외부 프로젝트의 라이선스 정보를 입력하세요.
- `whatap_platform` : 연동하려는 외부 프로젝트의 플랫폼을 입력하세요.

> ⓘ 현재 외부 프로젝트 연동을 지원하는 플랫폼은 다음과 같습니다. 플랫폼 입력 시 대소문자 구분없이 입력할 수 있습니다.
>
> **Redis, Java, Node.js, Python, PHP**

예시

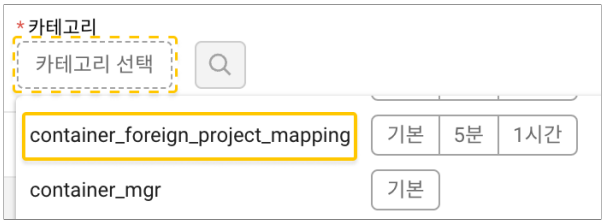
다음은 Redis Pod 설정에 환경 변수를 추가한 YAML 파일의 예시입니다. 하이라이팅된 라인이 추가된 환경 변수입니다.

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: redis-cluster
  namespace: backend
spec:
  serviceName: redis-cluster-svc
  replicas: 6
  selector:
    matchLabels:
      app: redis-cluster
  template:
    metadata:
      labels:
        app: redis-cluster
    spec:
      containers:
      - name: redis
        image: redis:5.0.1-alpine
        ports:
        - containerPort: 6379
          name: client
        - containerPort: 16379
          name: gossip
        command: ["/conf/update-node.sh", "redis-server", "/conf/redis.conf"]
```

```
env:
- name: POD_IP
  valueFrom:
    fieldRef:
      fieldPath: status.podIP
- name: license
  value: x6050s081i709-z2lhh7gpi3bdr4-x7jovr61kumi5k
- name: whatap_platform
  value: redis
volumeMounts:
- name: conf
  mountPath: /conf
  readOnly: false
- name: data
  mountPath: /data
  readOnly: false
```

외부 프로젝트 연동 확인

분석 > 메트릭스 조회



외부 프로젝트 연동 설정 후 **메트릭스 조회** 메뉴로 이동하세요. 예시 이미지와 같이 **카테고리 선택** 버튼 클릭하세요.

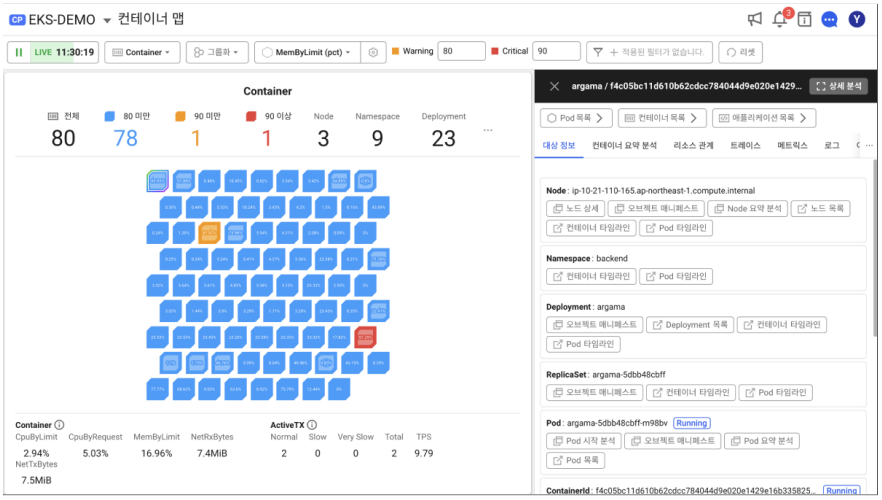
다음 카테고리가 표시된다면 연동이 정상적으로 적용된 상태입니다.

`container_foreign_project_mapping`

컨테이너 맵 상세 분석

컨테이너 맵 메뉴에서 선택한 대상에 대한 자세한 정보를 **상세 분석** 화면에서 확인할 수 있습니다.

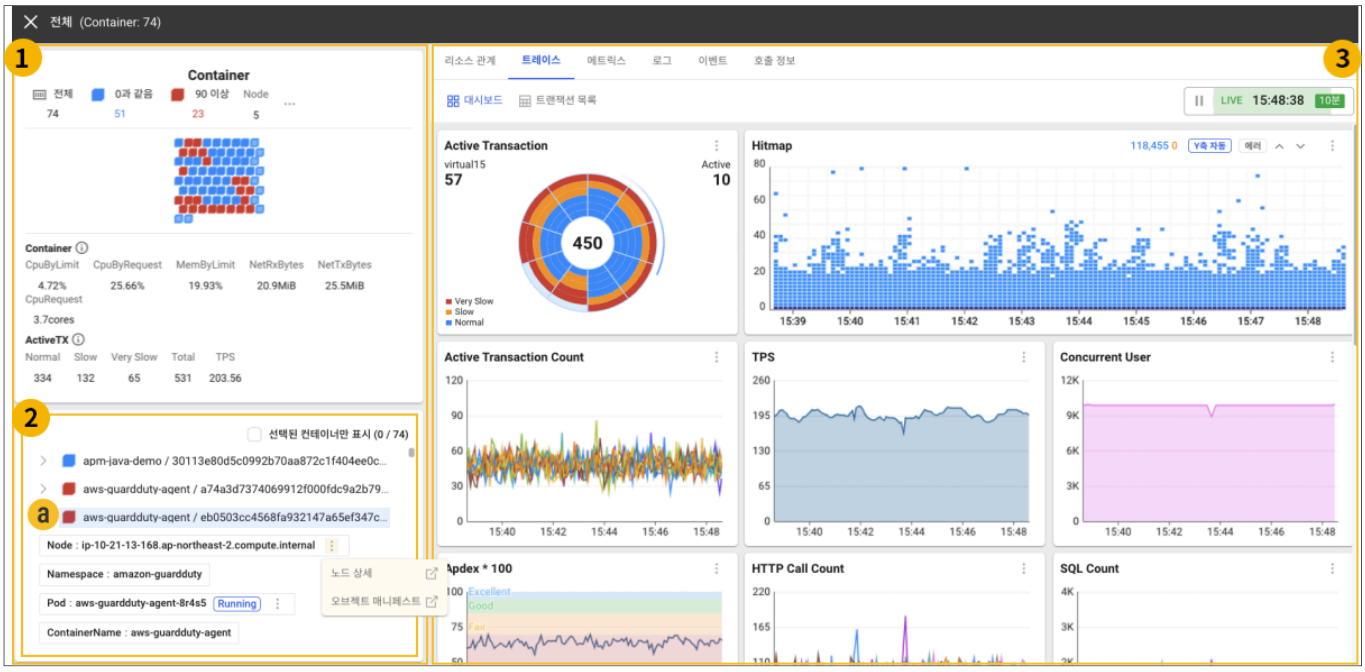
1. **컨테이너 맵** 메뉴 화면에서 조회를 원하는 특정 블록(컨테이너 또는 Pod)을 선택하세요.
2. 화면 오른쪽에 다음과 같이 **요약 보기** 패널이 나타납니다.



3. **요약 보기** 패널 오른쪽 상단에 **상세 분석** 아이콘을 선택하면 더욱 자세한 정보를 확인할 수 있습니다.

기본 화면 안내

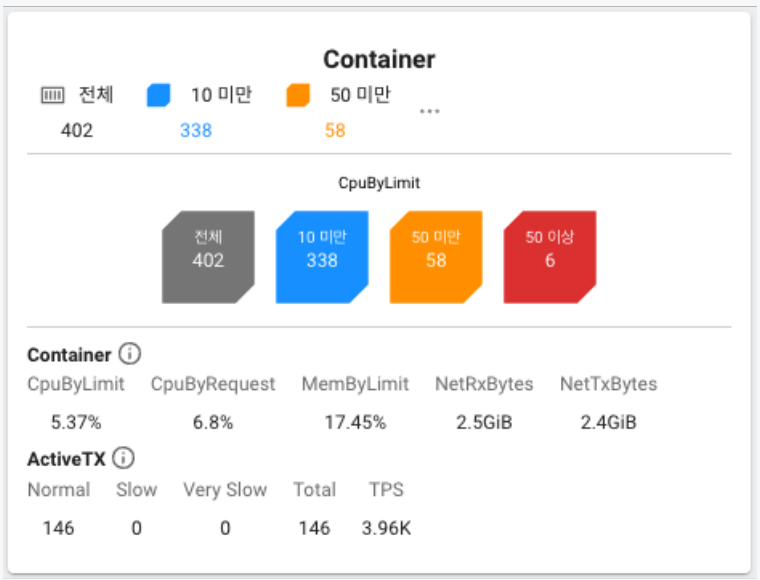
컨테이너 맵 **상세 분석** 화면은 다음과 같이 구성되어 있습니다. 기본 정보는 카드와 목록 형식으로 확인할 수 있습니다. 핵심 모니터링 데이터를 조회할 수 있도록 **트레이스**, **메트릭스**, **로그**, **이벤트** 탭을 제공합니다.



- **1 요약 정보 카드** (컨테이너 또는 Pod)

  선택 대상의 그룹 정보를 확인할 수 있습니다. 대상 추가 선택 시 3 영역에서 선택한 대상에 대한 지표를 확인할 수 있습니다. 컨테이너 맵에서 선택한 카드와 동일한 뷰를 가지고 있습니다. 자세한 내용은 다음 문서를 참조하세요.

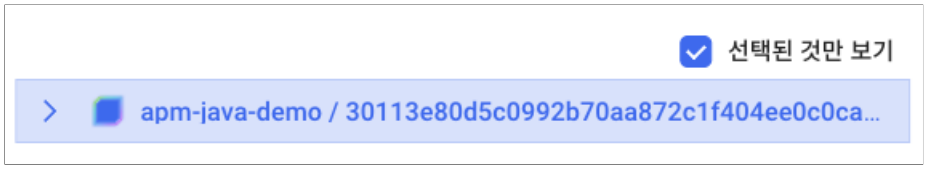
  > ⓘ 요약 정보 카드 조회 시 블록의 개수가 많은 경우 다음과 같이 표기 방식이 변경됩니다. 전체 블록, 정상, 경고, 위험 상태 블록 개수를 확인해 비교할 수 있습니다.



- **2 대상 정보 목록**

  선택 대상의 그룹 정보를 목록 형식으로 확인할 수 있습니다. 대상 추가 선택 시 3 영역에서 선택한 대상에 대한 지표를 확인할 수 있습니다.

  - 2 목록 상단의 **선택된 것만 보기** 체크 박스 선택 시 다음과 같이 선택된 컨테이너 또는 Pod만 표시됩니다.

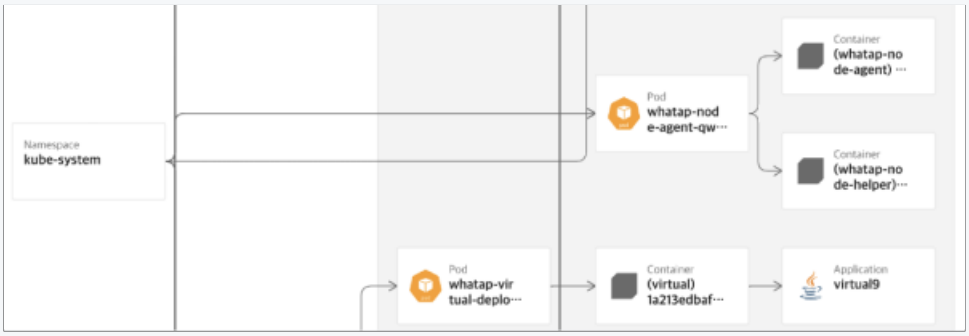
  - 목록에서 선택한 대상 좌측의 > **더보기** 아이콘을 선택하면 a 이하와 같이 **Node**, **Namespace**, **Deployment**, **ReplicaSet**, **Pod**, **ContainerName**, **ContainerId**, **Image**, **Agent** 등의 지표 정보를 조회할 수 있습니다. 자세한 내용은 다음 문서를 참조하세요.

- **3 리소스 관계, 트레이스, 메트릭스, 로그, 이벤트, 호출 정보**

  - **리소스 관계**

    리소스 관계 탭을 통해 쿠버네티스 리소스 간 연결 관계를 시각적으로 확인할 수 있습니다.

선택한 블록과 관련된 **Namespace**, **Deployment**, **ReplicaSet(DaemonSet)**, **Pod**, **Container**, **Application** 들의 관계를 표현한 차트를 다음과 같이 제공합니다.



- **트레이스**

  트레이스 탭을 통해 쿠버네티스 컨테이너 상에서 실행 중인 애플리케이션의 트랜잭션 상세 정보를 모니터링할 수 있습니다. **대시보드** 뷰와 **트랜잭션 목록** 뷰를 제공합니다. 자세한 내용은 다음 문서를 참조하세요.

- **메트릭스**

  메트릭스 탭을 통해 선택된 컨테이너 또는 Pod의 주요 메트릭 정보와 리소스 사용 패턴을 확인할 수 있습니다. 이를 통해 리소스 관련 문제점을 빠르게 식별하고 최적화 포인트를 찾을 수 있습니다. 자세한 내용은 다음 문서를 참조하세요.

- **로그**

  로그 탭을 통해 쿠버네티스 컨테이너 및 컨테이너 애플리케이션에서 발생하는 로그를 조회할 수 있습니다. 로그 기능은 컨테이너의 실시간 로그 조회 기능과 특정 시점의 과거 로그 검색 기능으로 나누어집니다. 자세한 내용은 다음 문서를 참조하세요.

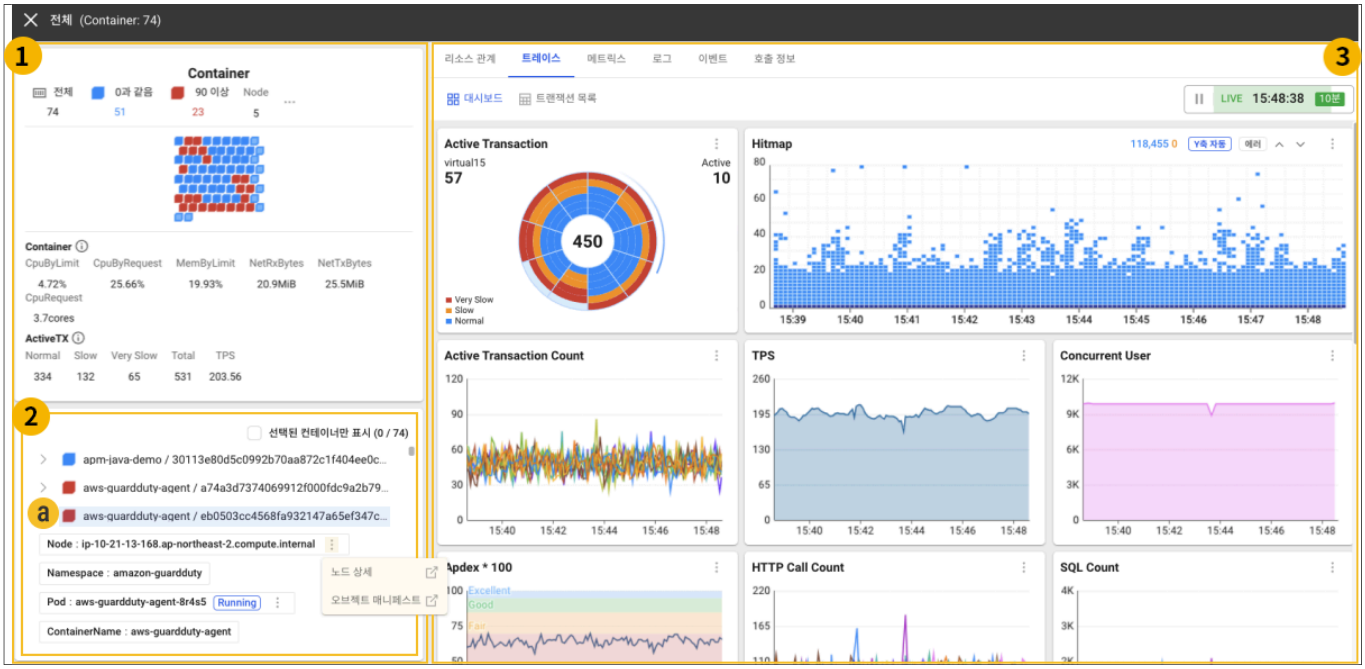
- **이벤트**

  이벤트 탭을 통해 쿠버네티스 클러스터에서 발생하는 **Kubernetes** 이벤트와 사용자가 직접 설정한 조건에 의한 **WhaTap** 이벤트를 조회할 수 있습니다. 자세한 내용은 다음 문서를 참조하세요.

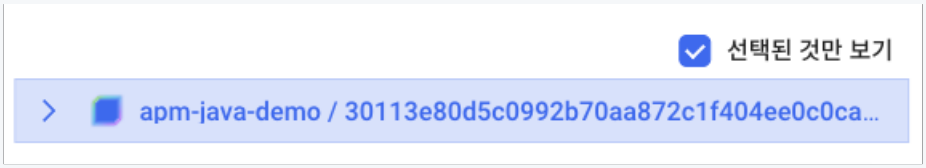
- **호출 정보**

  호출 정보 탭을 통해 쿠버네티스 컨테이너 상에서 실행 중인 애플리케이션의 호출 흐름과 외부 호출 대상의 상태를 확인할 수 있습니다. 자세한 내용은 다음 문서를 참조하세요.

Target Information



컨테이너 뷰 또는 Pod 뷰에서 블록은 각 컨테이너 또는 Pod를 의미합니다. 1 **요약 정보 카드** 내에서 대상 추가 선택 시 2 **대상 정보 목록**과 3 영역에서 각각의 상세 탭을 통해 선택한 대상에 대한 정보를 확인할 수 있습니다.

> ⓘ 2 **대상 정보 목록** 상단의 **선택된 것만 보기** 체크 박스 선택 시 다음과 같이 선택된 컨테이너 또는 Pod만 표시됩니다.
>
> 
>

블록 선택 후 **상세 분석** 내 2 **대상 정보 목록**을 통해 a 이하와 같이 선택한 컨테이너 또는 Pod가 실행 중인 노드, 소속된 네임스페이스 및 오브젝트 매니페스트와 같은 세부 대상 정보를 확인하거나 목록 메뉴로 이동할 수 있습니다.

- **Node**: 노드 상세, 오브젝트 매니페스트, Nod 요약 분석 정보를 확인하고 노드 목록 메뉴로 이동할 수 있습니다.
- **Deployment**: 오브젝트 매니페스트 정보를 확인하고 Deployment 목록 메뉴로 이동할 수 있습니다.
- **ReplicaSet**: 오브젝트 매니페스트 정보를 확인할 수 있습니다.

- **Pod**: Pod 시작 분석, 오브젝트 매니페스트, Pod 요약 분석 정보를 확인하고 Pod 목록 메뉴로 이동할 수 있습니다.
- **ContainerId**: **컨테이너 정보**, 컨테이너 요약 분석 정보를 확인하고 컨테이너 목록 메뉴로 이동할 수 있습니다.
- **Image**: **컨테이너 이미지** 정보를 확인하고 컨테이너 이미지 메뉴로 이동할 수 있습니다.
- **ServiceName**: 오브젝트 매니페스트 정보를 확인하고 Service 목록 메뉴로 이동할 수 있습니다.
- **Volume**: 컨테이너가 사용하는 볼륨과 마운트 경로 정보를 확인할 수 있습니다.
  - Pod에서 사용하는 ⓐ 볼륨의 이름과 ⓑ 볼륨 타입을 확인할 수 있습니다.

    예, **Volume** ⓐ containerddomainsocket ⓑ HostPath
  - **HostPath** 볼륨 타입을 사용할 경우 ⓒ 호스트 시스템의 파일 경로를 확인할 수 있습니다. 이 경로는 파드와 호스트 시스템 간의 파일 시스템을 연결하는 경로입니다.

    예, **Host Path** ⓒ /run/containerd/containerd.sock

> ⓘ
> - 와탭 쿠버네티스 에이전트에서 제공하는 볼륨 타입은 다음과 같습니다.
>
> HostPath , ConfigMap , Projected , EmptyDir , PersistentVolumeClaim (PVC), Secret
>
> - 쿠버네티스에서 지원하는 볼륨의 유형과 안내는 다음 쿠버네티스 공식 문서에서 확인할 수 있습니다.

쿠버네티스 상태 요약 분석

쿠버네티스 환경에서 각 컨테이너, Pod, 노드의 성능과 자원에 대한 요약 분석 보고서를 제공합니다. 클러스터와 네임스페이스의 컨테이너, Pod, 노드의 자원 할당 및 사용 현황을 실시간으로 모니터링하고 보고서 내 제안을 참고하여 효과적으로 대응할 수 있습니다.

> ⓘ 현재 노드 요약 분석 기능은 국문으로만 제공하고 있습니다.

컨테이너 요약 분석

- 기본 정보: 컨테이너의 기동 시간, 재기동 횟수, 속한 파드 및 네임스페이스 정보를 제공합니다.
- 컨테이너 이미지: 현재 사용 중인 컨테이너 이미지와 동일한 이미지를 사용하는 다른 컨테이너의 수 및 전체 리소스 사용량을 확인할 수 있습니다.

- 노드 정보: 컨테이너가 실행되는 노드의 운영체제, IP 주소, CPU 코어 수, 메모리 크기 등 하드웨어 정보를 제공합니다.
- 리소스 사용 현황: 컨테이너의 CPU 및 메모리 리소스 사용량, 설정된 한도(Limit) 및 요청(Request) 값과 비교한 사용률을 제공합니다.
- 서비스 네트워킹 정보: 연결된 Service 및 Ingress 정보를 확인할 수 있습니다.

Pod 요약 분석

- 기본 정보: Pod의 클러스터 이름, 네임스페이스, 기동 시간, 현재 상태(Phase) 등을 확인할 수 있습니다.
- 쿠버네티스 이벤트: 최근 파드와 관련된 이벤트 로그를 통해 문제가 발생했는지 확인할 수 있습니다.
- 실행 주체: 파드를 생성한 컨트롤러(DaemonSet, Deployment 등)를 식별합니다.
- 노드 정보: 파드가 실행 중인 노드의 운영체제, 커널 버전, CPU, 메모리 등의 상세 사양을 제공합니다.
- 리소스 사용 현황: 파드 내 모든 컨테이너의 CPU와 메모리 사용량, 설정된 리소스 한도(Limit)와 비교한 사용률을 보여줍니다.
- 서비스 네트워킹 정보: 연결된 Service 및 Ingress 정보를 확인할 수 있습니다.

Node 요약 분석

- 기본 정보: 노드의 클러스터 버전, 기동 시간, 현재 상태, 스케줄링 가능 여부, 내부 IP 주소, 그리고 리전 및 존 정보를 제공합니다.
- 운영체제 및 하드웨어 정보: 노드가 실행되는 운영체제, OS 이미지, CPU 아키텍처, 코어 수, 총 메모리 크기를 제공합니다. 노드의 메모리와 CPU에 대한 할당 가능 및 현재 사용량을 확인할 수 있습니다.
- 리소스 사용 현황: 노드에서 실행 중인 모든 컨테이너의 요청(Request)과 할당 가능 자원(CPU, 메모리)의 차이를 통해 잔여 자원을 파악합니다. 실제 CPU 사용률과 메모리 사용률을 포함하여 리소스 사용 상태를 요약합니다.
- 디스크 사용 현황: 노드에 연결된 디스크 정보, 파일 시스템 유형, 디바이스 ID, 마운트 지점, 총 용량 및 사용 중인 용량, 그리고 사용률을 제공합니다.
- Pod 상태 및 리소스 여유: 노드의 할당 가능한 전체 Pod 수와 현재 실행 중인 Pod 수를 비교하여, 추가 실행 가능한 Pod 수를 확인할 수 있습니다. 또한 MemoryPressure, DiskPressure, PidPressure 등의 상태를 통해 노드의 안정성을 점검합니다.

Trace

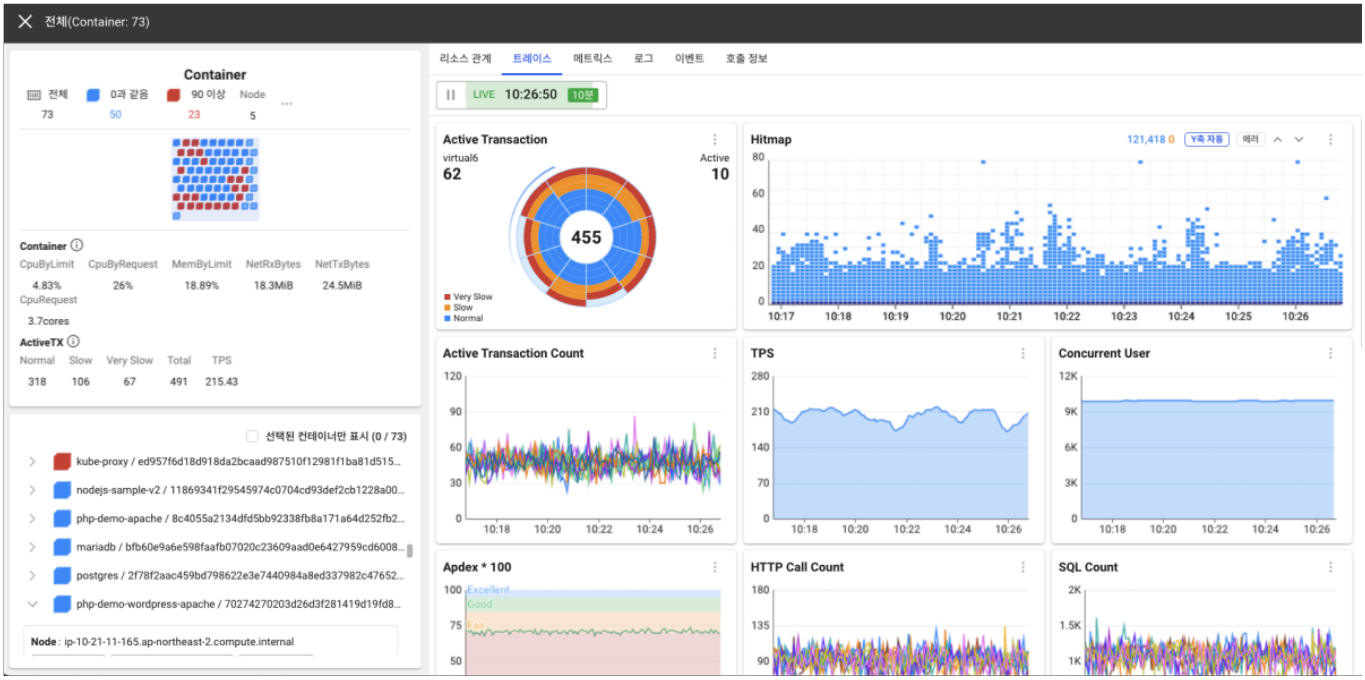
컨테이너화된 애플리케이션 트레이스

쿠버네티스 모니터링 시 쿠버네티스 환경 내부에서 실행 중인 애플리케이션에 대한 추적 없이 단지 클러스터만 모니터링해서는 현재 상태나 문제의 원인을 정확히 파악하기 힘듭니다.

문제의 원인을 추적하다보면 특정 트랜잭션의 SQL 처리 지연, 비즈니스 로직 오류 등 애플리케이션 레벨의 문제인 경우가 많기 때문입니다.

와탭 쿠버네티스의 트레이스 기능은 컨테이너 상에서 실행 중인 애플리케이션 상태 모니터링 정보를 제공합니다.

> ⓘ 쿠버네티스 상에서 실행 중인 애플리케이션을 추적하기 위해서는 Java, Node.js, Python 등의 각 애플리케이션 컨테이너에 와탭 모니터링 에이전트가 설치되어야 합니다. 에이전트 설치에 대한 자세한 내용은 다음 문서를 참조하세요.



대시보드

대시보드는 다음과 같은 애플리케이션의 핵심 정보를 한눈에 제공합니다.

- **Active Transaction**: 현재 실행 중인 트랜잭션으로, 트랜잭션 실행 시간에 따라 색으로 분류
  - 빨간색 (Very slow) : 8초 이상
  - 주황색 (Slow) : 3초 ~ 8초
  - 파란색 (Normal) : 0초 ~ 3초
- **Active Transaction Count**(액티브 트랜잭션 추이)
- **Htimap**(히트맵): 종료된 트랜잭션 응답시간의 분포도
  - 가로축: 트랜잭션 종료 시간
  - 세로축: 수행 시간
  - 수행 시간이 긴 트랜잭션은 분포도 상단에 위치
  - 특정 영역 드래그 시 트랜잭션 정보 확인 가능
- **TPS**(Transaction Per Second): 초당 처리된 트랜잭션 건수
- **Conrurrent User**: 동시 사용자
- **Apdex * 100**(Application Performance Index): 웹 애플리케이션의 사용자 만족도 측정
- **HTTP Call Count**: HTTP 호출 건수
- **HTTP Call Error Count**: HTTP 호출 에러 건수
- **SQL Count**
- **SQL Error Count**
- **Transaction Total Conunt**: 트랜잭션 건수
- **Transaction Error Conunt**: 트랜잭션 Error 건수
- **Response Time**: 응답시간
- **Heap Memory Usage**: Heap Memory 사용량, 사용 가능한 최대/현재 메모리
  - 시간에 따른 메모리 사용량 변화를 실시간 파악 가능
  - Python의 경우 Process Total 메모리 사용량

- **ThreadCount**: Thread 건수, Node.js, Python 제외
- **GC Count**: GC 횟수, Node.js, Python 제외
- **GC Old Generation Count**: Old 영역 GC 횟수, Node.js, Python 제외

트레이스 분석

대시보드 내 **Hitmap**에서 조회를 원하는 구간 드래그 시 **트레이스 분석**을 확인할 수 있습니다. 트랜잭션 목록에서 원하는 트랜잭션 선택 후 각 트랜잭션 하위의 스텝 정보를 한 번에 확인할 수 있습니다.

> ⓘ 트레이스 분석에 대한 자세한 내용은 다음 문서를 참조하세요.

Metrics

쿠버네티스 메트릭스

쿠버네티스는 각 컨테이너의 리소스 사용량을 통제하고 지속적으로 모니터링하면서 최적의 상태를 유지합니다. 따라서 리소스 사용량을 확인하고, 해당 데이터를 기반으로 컨테이너 리소스 사용량이 적절히 설정되었는지를 확인하는 과정은 쿠버네티스 모니터링의 기본적인 패턴이라고 할 수 있습니다.

CPU와 Memory 메트릭

대표적인 쿠버네티스 리소스 설정은 컨테이너의 CPU Request/Limit, Memory Request/Limit입니다. 다음은 설정 예시입니다.

```
apiVersion: v1
kind: Pod
metadata:
  name: <Pod명>
spec:
  containers:
  - name: <컨테이너 1명>
    image: <컨테이너 1 이미지>
    resources:
      requests:
        memory: "64Mi"
        cpu: "100m"
      limits:
        memory: "128Mi"
        cpu: "200m"
  - name: <컨테이너 2명>
    image: <컨테이너 2 이미지>
    resources:
      requests:
        memory: "128Mi"
        cpu: "250m"
      limits:
```

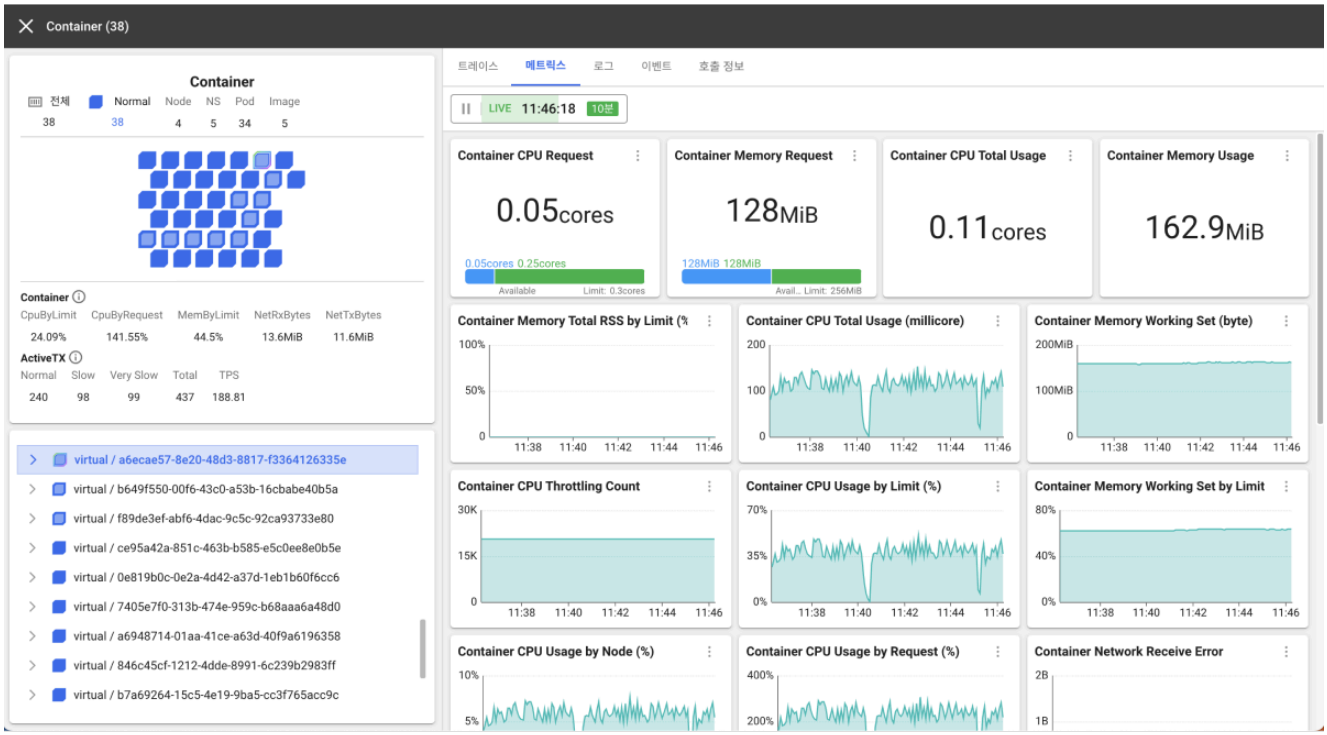
```
23      memory: "256Mi"
24      cpu: "500m"
```

위 설정을 통해 해당 Pod 내에는 2개의 컨테이너가 존재합니다. 각 컨테이너의 리소스 설정은 다음과 같습니다.

대상 컨테이너	리소스 구분	리소스 설정	값
컨테이너1	CPU	Request	100m
		Limit	200m
	Memory	Request	64Mi
		Limit	128Mi
컨테이너2	CPU	Request	250m
		Limit	500m
	Memory	Request	128Mi
		Limit	256Mi

다양한 컨테이너의 리소스 설정 정보 및 현재 사용량을 쉽게 파악하기란 쉽지 않습니다. 그래서 와탭 쿠버네티스 모니터링 서비스는 특정 단일 컨테이너, 사용자가 선택한 복수의 컨테이너들, 또는 Deployment 전체를 대상으로 리소스 설정 및 현재 사용량을 쉽게 확인할 수 있는 기능을 제공합니다.

- **상세보기** 화면 > **메트릭스** - 리소스 사용량 모니터링 및 분석



- 특정 컨테이너를 선택해 상태를 확인해 볼 수 있습니다.

위 데이터를 보면 CPU는 Request 0.2cores(200m), Limit 0.4cores(400m)로 설정되어 있습니다. 현재 CPU 사용량은 0.02cores(20m) (5초 평균) 입니다. 우측의 CPU 관련 추이 그래프를 확인하면 전반적으로 여유가 있지만 간헐적으로 CPU 사용량이 증가하는 현상을 확인할 수 있습니다.

Memory는 Request 128MiB, Limit 256MiB로 설정되어 있습니다. 그런데 현재 Memory 사용량은 184MiB입니다. 우측의 Memory 관련 추이 그래프를 봐도 계속 Request 기준으로 100%를 초과하였고 Limit 기준으로도 70% 이상을 사용하고 있습니다.

CPU, Memory 관련 주요 와탭 메트릭은 다음과 같습니다.

대상 리소스	Shortname	Name	설명
CPU	CpuTotUsage	**Container CPU Usage (millicore)**	컨테이너 CPU 사용률
	CpuByRequest	**Container CPU Usage by Request (%)**	컨테이너 CPU 요청 대비 사용률
	CpuByLimit	**Container CPU Usage by Limit (%)**	CPU Limit 기준 CPU 전체 사용률
Memory	MemWs	**Container Memory Working Set (byte)**	컨테이너 메모리 working set
	MemWsByRequest	**Container Memory Working Set by Request (%)**	컨테이너 메모리 요청량 기준 working set 사용량
	MemWsByLimit	**Container Memory Working Set by Limit (%)**	컨테이너 메모리 할당량 기준 working set 사용량

ⓘ Pod가 실행 중인 노드에 사용 가능한 리소스가 충분하면 컨테이너에 설정된 Request 이상의 리소스를 사용할 수 있습니다. 와탭 메트릭 중 **Container CPU Usage by Request (%)**, **Container Memory Working Set by Request (%)**의 값이 100%를 넘어설 수 있는 이유입니다.

예를 들어, 컨테이너의 Memory Request를 128MiB로 설정했다고 가정하겠습니다. 컨테이너(Pod)가 실행 중인 노드의 총 메모리가 4GiB이고, 해당 노드에 다른 Pod가 없는 경우 이 컨테이너는 128MiB이상의 메모리를 사용할 수 있습니다.

그 외 메트릭

와탭은 CPU, Memory 외에도 다양한 메트릭을 제공합니다.

- **Container Block I/O Read Byte** (IoReadBytes)
- **Container Block I/O Write Byte** (IoWriteBytes)
- **Container Network Receive Byte** (NetRxBytes)
- **Container Network Transmit Byte** (NetTxByes)

와탭 쿠버네티스 메트릭스에 대한 자세한 내용은 다음 문서를 참조하세요.

Log

쿠버네티스 로그

와탭 쿠버네티스 모니터링 서비스는 쿠버네티스에서 발생하는 로그를 수집하고 저장합니다. 쿠버네티스는 다음 두 종류의 로그가 있습니다.

- 각 컨테이너의 Standard Out(표준 출력) 로그
- 각 컨테이너 내부에서 실행 중인 애플리케이션이 출력하는 로그

이러한 로그는 컨테이너의 상태나 메트릭 정보로는 확인하기 힘든 모니터링의 사각지대를 보완하는 역할을 합니다. 이 로그를 확인하는 방법은 다음과 같습니다.

- 각 컨테이너의 Standard Out 로그의 경우는 컨테이너 내부에 진입하지 않아도 쿠버네티스가 제공하는 `kubectl logs` 명령을 통해 확인할 수 있습니다.
- 각 컨테이너 내부의 애플리케이션 로그는 컨테이너 내부에 진입하여 확인하거나 별도의 로그 수집 방안을 수립해야 합니다.

최근 로그 메시지 (컨테이너 Stdout/Stderr)

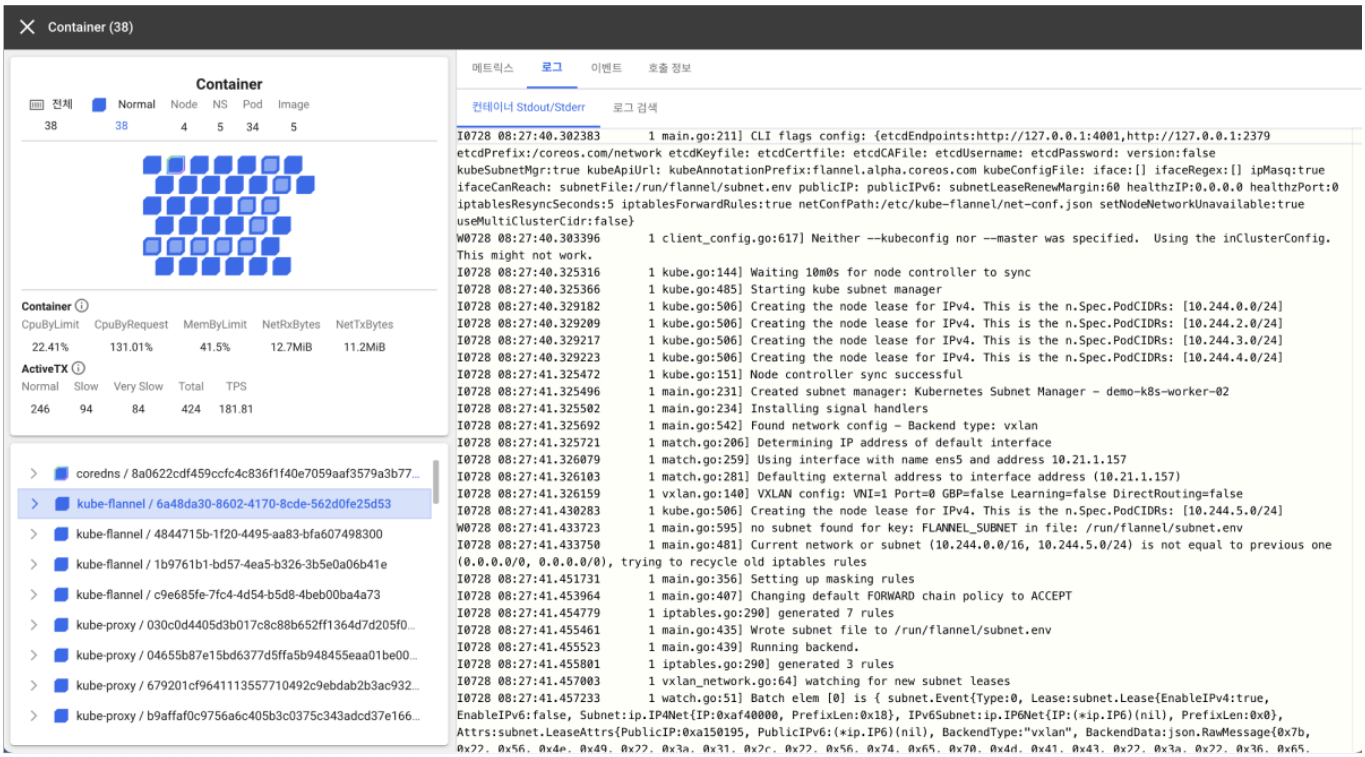
현재 컨테이너 로그를 바로 확인할 수 있는 기능입니다. 사용자의 클러스터 환경에서 직접 로그를 가져와 선택한 컨테이너의 실시간 로그 또는 선택한 Pod 내 애플리케이션의 실시간 로그를 확인할 수 있습니다. 일반적으로 컨테이너 로그를 확인하기 위해서는 다음 명령어를 사용합니다.

```
kubectl logs pod/<Pod명> -n <Namespace명>
```

Pod 내에 2개 이상의 컨테이너가 존재한다면 개별 컨테이너를 선택해 로그를 확인합니다.

```
kubectl logs pod/<Pod명> -c <컨테이너명> -n <Namespace명>
```

로그를 확인하기 위해 매번 터미널 화면에 접근해 커맨드 라인 도구를 실행하는 것은 매우 불편합니다. 와탭 쿠버네티스 모니터링의 **로그** 기능을 사용하면 편리하게 현재 로그를 확인할 수 있습니다.



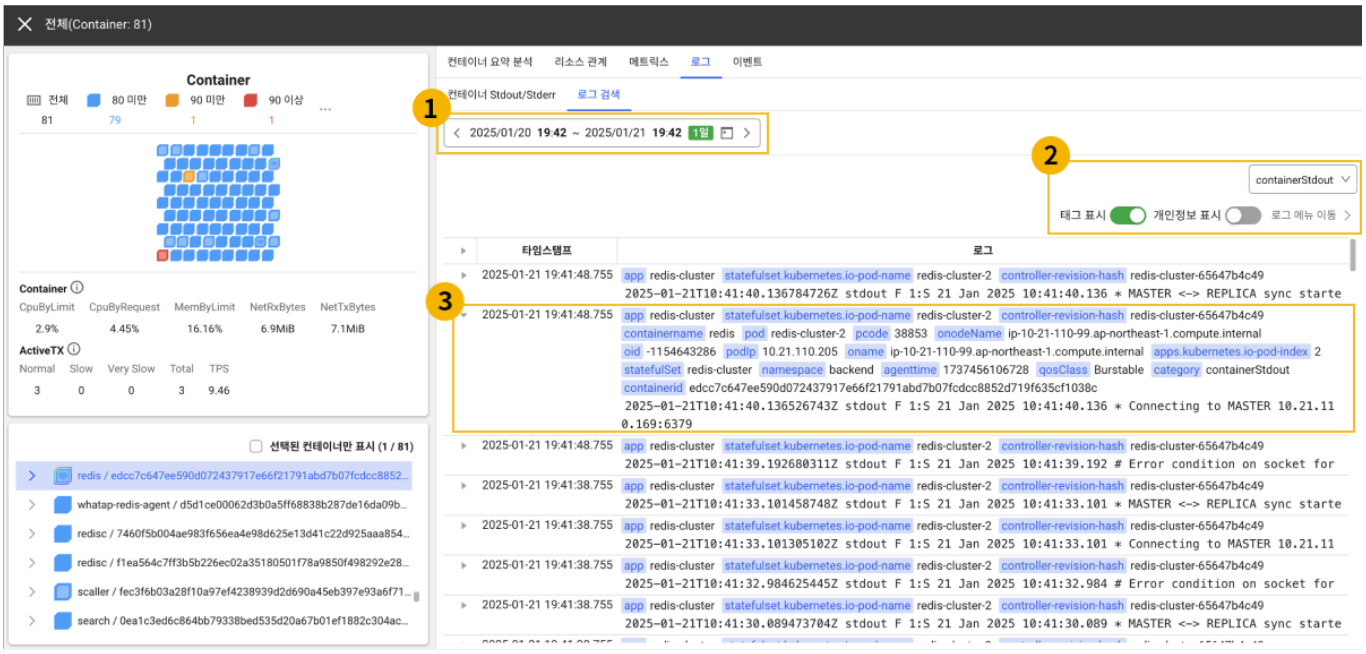
다만 생각해봐야 할 점이 있습니다. 우선 컨테이너를 종료하면 더 이상 로그를 확인할 수 없습니다. 또한 컨테이너가 계속 실행 중이더라도 일정 기간 또는 일정 용량의 로그만 확인할 수 있다는 단점이 있습니다. 와탭의 최근 로그 메시지(**컨테이너 Stdout/Stderr**) 기능은 사용자 쿠버네티스 환경의 로그를 와탭을 통해 볼 수 있게 제공하는 기능으로 사용자 환경에서 사라진 로그는 와탭을 통해서도 확인하기 어렵습니다.

따라서 로그 데이터를 안정적이고 지속적으로 관리하기 위해서는 별도의 관리 방안이 필수적입니다. 특히 마이크로서비스 아키텍처 도입이 확산되면서 비즈니스를 구성하는 컨테이너 수량이 증가함에 따라 로그의 확인, 관리 방안은 더욱 복잡해지고 있습니다.

좋은 방법이 있을까요?

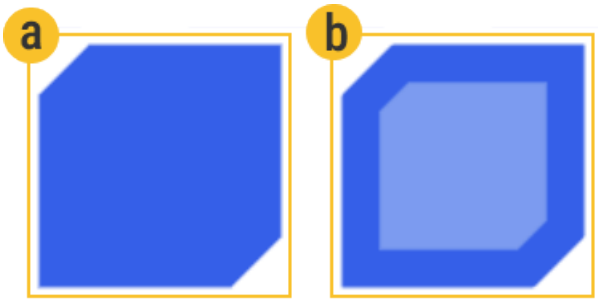
로그 검색

컨테이너 종료 시 관련 로그가 모두 사라지지만 **로그 검색** 기능을 통해 선택한 컨테이너 또는 Pod에서 과거에 발생한 로그를 확인할 수 있습니다. 컨테이너 Standard Out 로그 저장 시 원하는 만큼 저장 기간을 설정할 수 있고 특정 대상별 정보를 확인하거나 상호 연계 분석을 할 수 있습니다.



- 1 시간 선택자를 통해 데이터 조회를 원하는 시점을 선택할 수 있습니다.
- 2 카테고리 옵션을 통해 카테고리를 지정해 조회할 수 있습니다. 조회할 수 있는 카테고리에 대한 설명은 다음의 블록과 그룹 안내를 참조하세요.
- 2 **태그 표시** 토글을 비활성화하면 로그 기록 중 태그 정보를 제외한 **타임스탬프**와 **메시지**를 확인할 수 있습니다.
- 2 **개인정보 표시** 토글을 활성화하면 마스킹된 데이터의 원본 데이터를 조회할 수 있습니다. **개인정보 비식별화** 기능에 대한 자세한 내용은 다음 문서를 참조하세요.
- 2 **로그 검색** 버튼을 클릭하면 필터 기능 등을 통해 상세 조회할 수 있는 **로그** > **로그 검색** 메뉴로 이동합니다.
- 3 ▶ **더보기** 아이콘을 선택하면 해당 로그의 파싱된 키와 값 및 로그 메시지를 확인할 수 있습니다. 자세한 내용은 다음 문서를 참조하세요.

일반 블록과 마이크로 블록



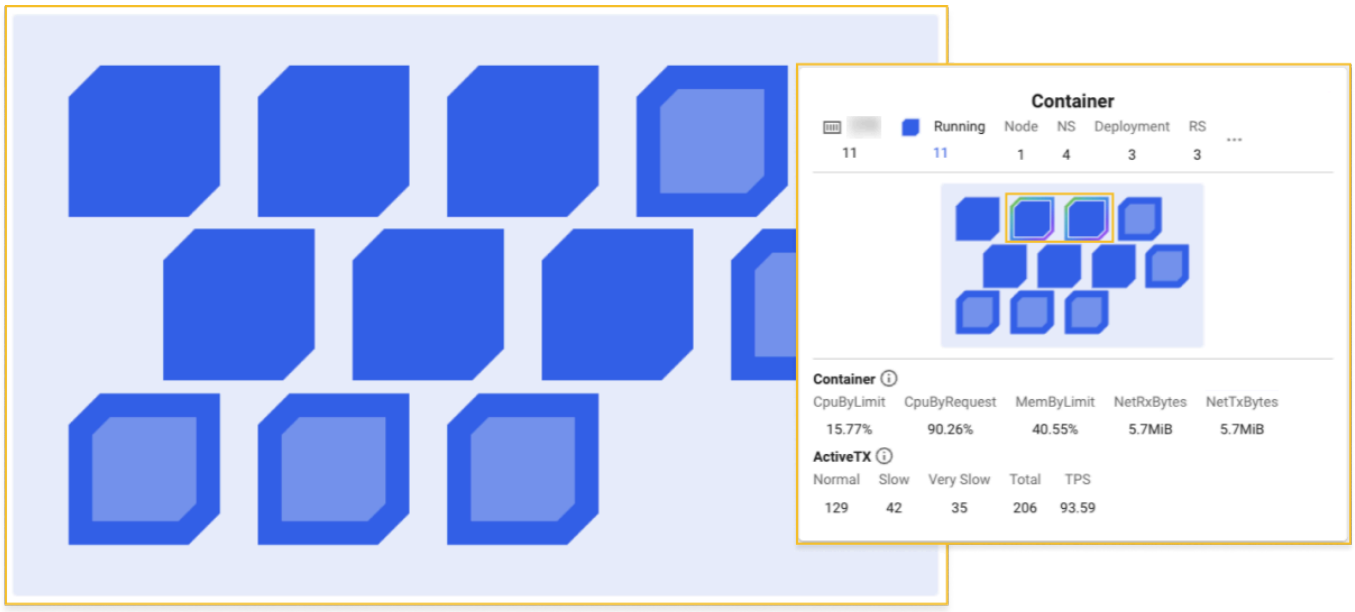
- ⓐ **일반 블록**

  컨테이너 맵의 다각형 하나를 블록이라고 합니다. 로그의 `containerStdout` 카테고리에서 `containerid` 가 일치하는 정보를 확인할 수 있습니다.

- ⓑ **마이크로 블록**

  에이전트가 설치된 블록을 마이크로 블록이라고 합니다. `containerStdout` 카테고리 뿐 아니라 `AppLog` , `AppStdOut` , `AppStdErr` 등 `App*` 관련 카테고리도 확인할 수 있습니다. 해당 카테고리를 선택할 경우 `microOid` (블록 내 에이전트 id)가 일치하는 로그 정보를 확인할 수 있습니다.

그룹



컨테이너 맵 화면에서 전체 블록을 선택한 경우 혹은 **상세 보기** 화면의 카드에서 블록을 다중 선택한 경우 해당 다중 블록을 그룹이라고 합니다. 그룹 선택 시 **로그 검색** 탭에서 `containerStdout` 카테고리와 `App*` 카테고리의 모든 로그를 확인할 수 있습니다.

Standard Out 저장을 위한 노드 에이전트 설정 추가

와탭 로그 서비스를 통해 컨테이너의 Standard Output을 저장하기 위해서는 와탭 쿠버네티스 노드 에이전트에 `logsink_enabled=true` 옵션을 설정해야 합니다. 설정 및 와탭 로그 모니터링 서비스에 대한 자세한 내용은 다음 문서를 참조하세요.

컨테이너 내부 애플리케이션 로그 수집을 위한 애플리케이션 에이전트 설정 추가

와탭 로그 서비스를 통해 컨테이너 상에서 실행되는 애플리케이션의 로그를 수집하려면 다음 문서를 참조하세요. Java, Python, Go를 지원합니다.

> 와탭 로그 서비스를 활성화하고 컨테이너의 Standard Output 로그 저장 기능을 사용하면 그에 따른 추가 비용이 청구됩니다. 관련한 요금에 대한 자세한 내용은 다음 문서를 참조하세요.

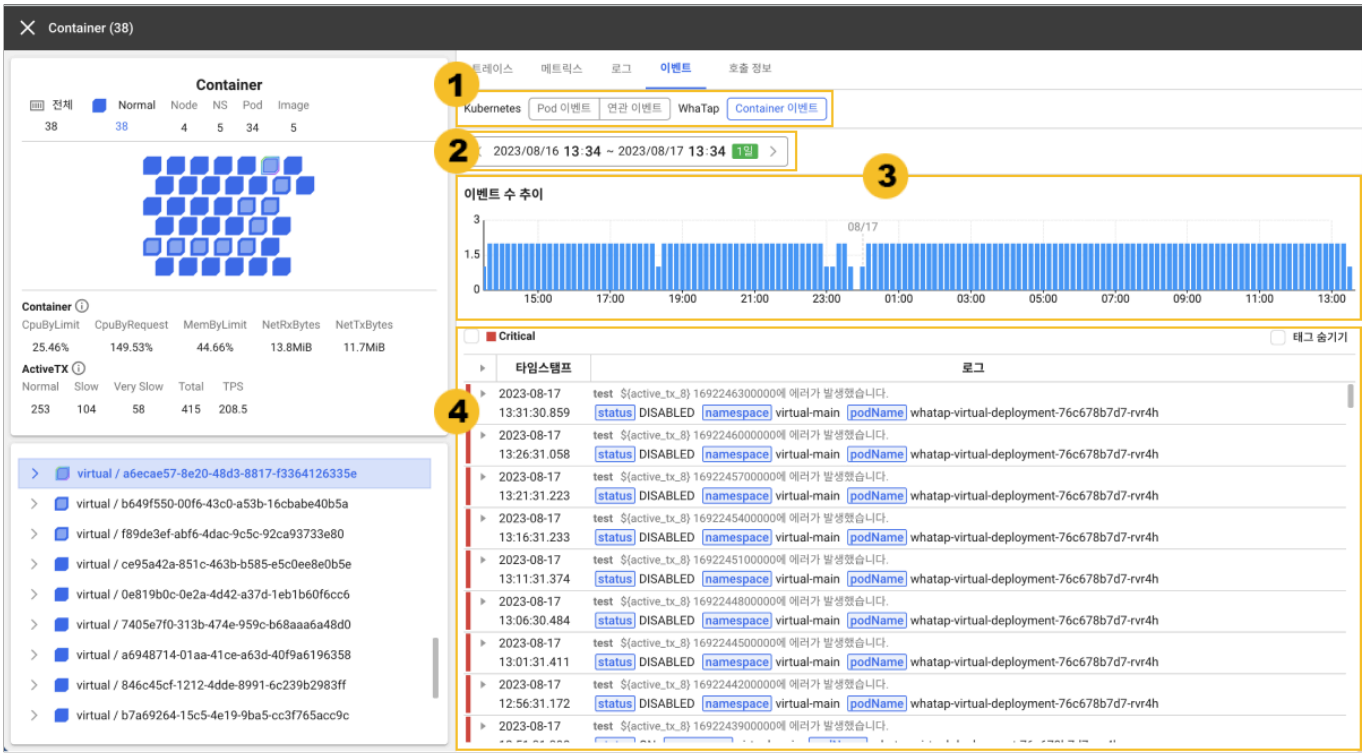
> 일부 구 버전 에이전트의 경우는 컨테이너 Standard Output 로그 저장 기능을 제공하지 않습니다. 에이전트 업그레이드 방법은 다음과 같습니다.
>
> Agent update
>
> ```
> kubectl set image daemonset/whatap-node-agent whatap-node-agent=whatap/kube_mon:<에이전트버전> -n whatap-monitoring
> ```
>
> 와탭 쿠버네티스 에이전트 버전은 다음 문서를 통해 확인할 수 있습니다.

Event

리소스 수명 주기 동안 발생하는 모든 이벤트에 대한 기록을 제공합니다. 와탭 쿠버네티스 모니터링에서 제공하는 이벤트 정보는 다음과 같습니다.

- 쿠버네티스 이벤트(**Kubernetes**)
- 와탭 이벤트(**WhaTap**)

기본 화면 안내



- 1 쿠버네티스 이벤트(**Kubernetes**) 또는 와탭 이벤트(**WhaTap**)를 선택해 조회할 수 있습니다.
  - 전체 또는 다중 블록 선택

    컨테이너 맵 및 왼쪽의 카드에서 전체 또는 다중 블록 선택 시 **Kubernetes 전체 이벤트** 또는 **WhaTap 전체 이벤트**를 조회할 수 있습니다.

- 클러스터링 또는 개별 블록 선택

  컨테이너 맵 상단의 **2단계** 그룹화 선택 또는 개별 블록 선택 시 해당 이벤트를 조회할 수 있습니다. **그룹화**(클러스터링)에 관한 자세한 내용은 다음 문서를 참조하세요.

  > ⓘ **상세 보기** 화면에 진입하기 전 **컨테이너 맵** 상단에서 선택한 뷰 및 대상 그룹화에 따라 조회할 수 있는 이벤트가 달라집니다.

- 2 시간 선택자를 통해 이벤트 기록을 조회하고자 하는 시간대를 지정할 수 있습니다. 기본 설정은 **1일**입니다.
- 3 선택한 시간대의 **이벤트 수 추이**를 바 차트를 통해 살펴볼 수 있습니다.
- 4 목록 형식으로 이벤트 기록 정보를 확인할 수 있습니다.
  - 이벤트 기록 목록 오른쪽 상단의 **태그 숨기기**를 선택하면 이벤트 기록 중 태그 정보를 제외한 **타임스탬프**와 **메시지**를 확인할 수 있습니다.
  - **Kubernetes** 이벤트
    - Normal 또는 Warning 중 조회를 원하는 이벤트 타입을 선택해 확인할 수 있습니다. 이 이벤트 타입은 쿠버네티스가 제공하는 이벤트 타입 정보와 동일합니다.
    - 각 이벤트의 왼쪽 띠 역시 이벤트 타입을 나타냅니다. 회색은 Normal, 주황색은 Warning 상태를 의미합니다.
  - **WhaTap** 이벤트
    - Critical, Info, Warning 중 조회를 원하는 이벤트 레벨을 선택해 확인할 수 있습니다.
    - 각 이벤트의 왼쪽 띠 역시 이벤트 레벨을 나타냅니다. 회색은 Info, 주황색은 Warning, 빨간색은 Critical 상태를 의미합니다.

쿠버네티스 이벤트

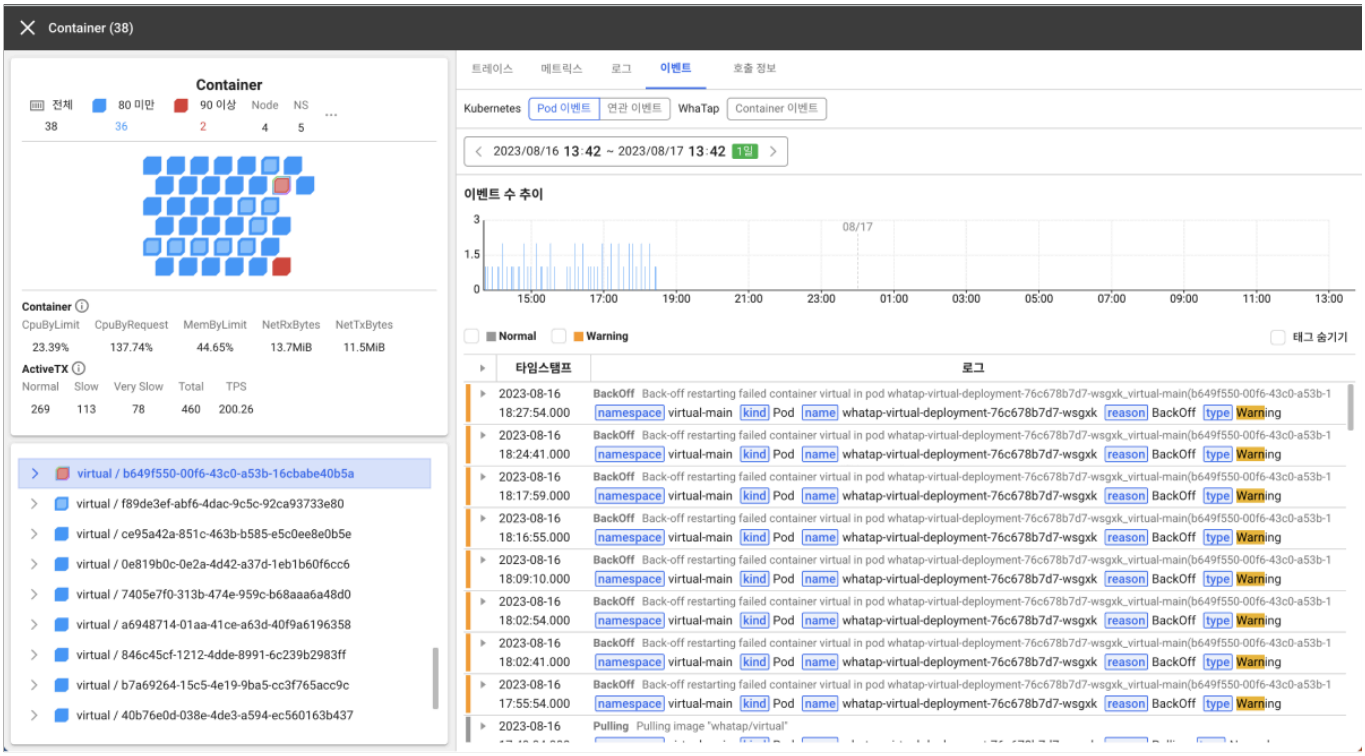
Kubernetes 이벤트는 쿠버네티스 자체적으로 제공하는 이벤트 정보입니다. 사용자 쿠버네티스 환경에서 `kubectl get events` 명령을 사용하면 사용자 쿠버네티스 클러스터의 이벤트 정보를 확인할 수 있습니다.

```
$ kubectl get events -A
 NAMESPACE   LAST SEEN   TYPE     REASON              OBJECT                    MESSAGE
 community   15m         Normal   ScalingReplicaSet   deployment/baekdusan      Scaled up replica set
```

```
baekdusan-59554d4859 to 5
 community   12m         Normal    ScalingReplicaSet       deployment/suraksan             Scaled down replica set
suraksan-66fc4bf889 to 3
 community   5m18s       Normal    ScalingReplicaSet       deployment/baekdusan            Scaled down replica set
baekdusan-59554d4859 to 3
 community   7m10s       Normal    Created                 pod/suraksan-66fc4bf889-h2cw8   Created container suraksan
 community   6m56s       Normal    Pulled                  pod/suraksan-66fc4bf889-h2cw8   Container image
"123456789123.dkr.ecr.ap-northeast-2.amazonaws.com/suraksan:0.48" already present on machine
 community   7m9s        Normal    Started                 pod/suraksan-66fc4bf889-h2cw8   Started container suraksan
 community   2m7s        Warning   BackOff                 pod/suraksan-66fc4bf889-h2cw8   Back-off restarting failed container
 community   7m12s       Normal    Started                 pod/suraksan-66fc4bf889-t27rp   Started container suraksan
 community   6m54s       Normal    Pulled                  pod/suraksan-66fc4bf889-t27rp   Container image
"123456789123.dkr.ecr.ap-northeast-2.amazonaws.com/suraksan:0.48" already present on machine
 community   7m12s       Normal    Created                 pod/suraksan-66fc4bf889-t27rp   Created container suraksan
 community   2m46s       Warning   BackOff                 pod/suraksan-66fc4bf889-t27rp   Back-off restarting failed container
 community   7m19s       Normal    ScalingReplicaSet       deployment/mountain             Scaled up replica set
mountain-6795bfbc54 to 5
 community   27m         Normal    ScalingReplicaSet       deployment/mountain             Scaled down replica set
mountain-6795bfbc54 to 4
 community   6m53s       Normal    Pulled                  pod/suraksan-66fc4bf889-mjbsj   Container image
"123456789123.dkr.ecr.ap-northeast-2.amazonaws.com/suraksan:0.48" already present on machine
 community   7m10s       Normal    Created                 pod/suraksan-66fc4bf889-mjbsj   Created container suraksan
 community   7m10s       Normal    Started                 pod/suraksan-66fc4bf889-mjbsj   Started container suraksan
 community   2m10s       Warning   BackOff                 pod/suraksan-66fc4bf889-mjbsj   Back-off restarting failed container
 community   5m18s       Warning   FailedToUpdateEndpoint  endpoints/baekdusan-svc         Failed to update endpoint
community/baekdusan-svc: Operation cannot be fulfilled on endpoints "baekdusan-svc": the object has been modified;
please apply your changes to the latest version and try again
```

하지만 이러한 이벤트 정보는 지속적으로 저장되지 않습니다. 따라서 일정 시간이 지난 이벤트는 더 이상 정보를 확인할 수 없습니다.

와탭 쿠버네티스는 사용자 클러스터 환경에서 발생한 쿠버네티스 이벤트 정보를 모두 수집하여 저장합니다. 그리고 사용자가 확인하고자 하는 대상에 따라 Pod, ReplicaSet, Deployment, Namespace 등의 이벤트 정보를 제공합니다.



각 이벤트의 왼쪽 띠는 이벤트 타입을 나타냅니다. 회색은 Normal, 주황색은 Warning 상태를 의미합니다. 사용자 선택 대상에 따른 제공 쿠버네티스 이벤트는 다음과 같습니다.

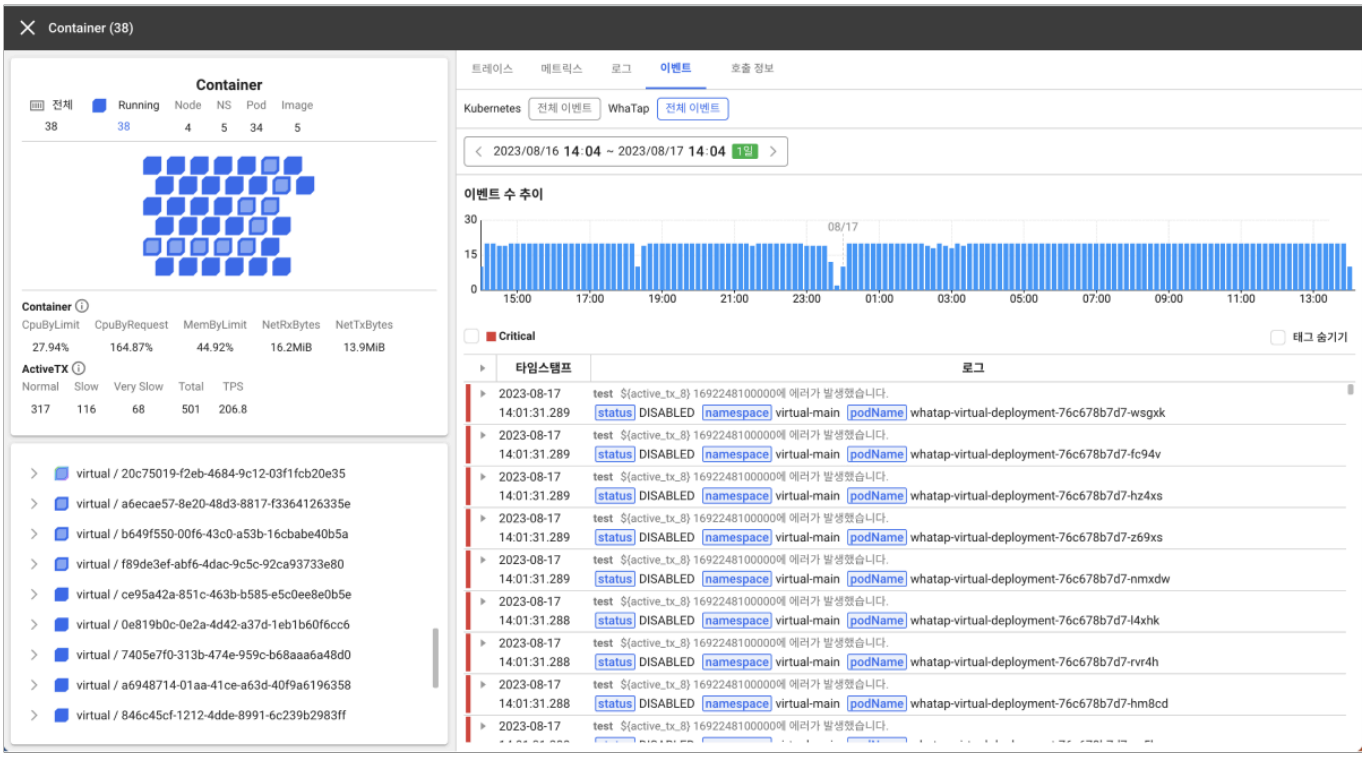
선택 대상	제공 이벤트	제공 연관 이벤트
전체 (All)	프로젝트 전체 이벤트	
Namespace	Namespace 하위 전체 이벤트	
Node	해당 Node 이벤트 또는 노드 내 Pod (컨테이너) 전체 이벤트	
Deployment	① `kind`가 Deployment ② `name`이 사용자 Deployment 이름과 일치	`name`이 사용자 Deployment 이름으로 시작
ReplicaSet	① `kind`가 ReplicaSet ② `name`이 사용자 ReplicaSet 이름과 일치	`name`이 사용자 ReplicaSet 이름으로 시작

선택 대상	제공 이벤트	제공 연관 이벤트
Pod (컨테이너)	① `kind`가 Pod ② `name`이 사용자 Pod 이름과 일치	해당 Pod가 소속된 Deployment 이름으로 시작 또는 ReplicaSet 이름으로 시작

와탭에 저장된 쿠버네티스 이벤트는 `message`, `namespace`, `kind`, `name`, `reason`, `type` 등의 항목으로 구성되어 있습니다.

와탭 이벤트

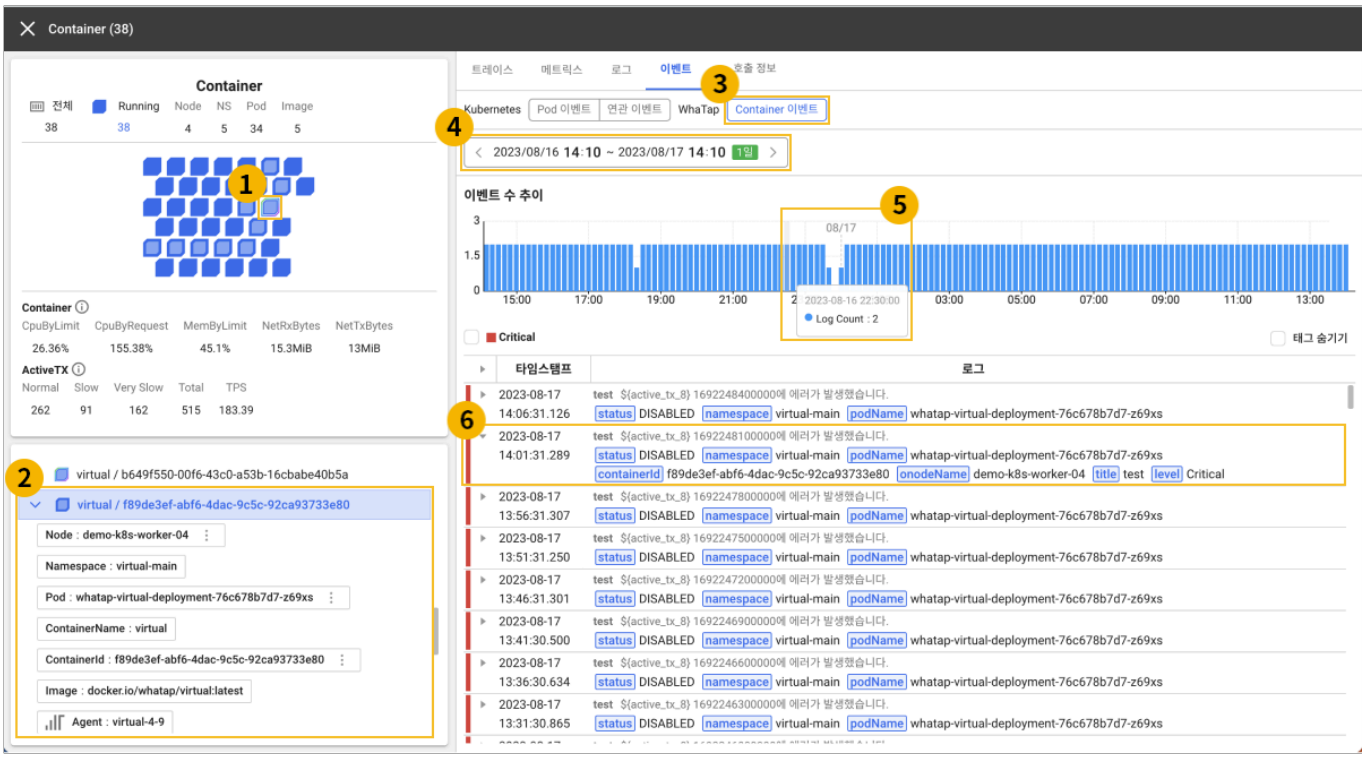
WhaTap 이벤트는 와탭의 이벤트 설정을 통해 등록한 이벤트 조건에 따라 발생한 이벤트입니다. **컨테이너 맵** 상세 보기 화면의 **이벤트** 탭에서 사용자의 와탭 경고 설정에 따른 이벤트 목록을 확인할 수 있습니다.



각 이벤트의 왼쪽 띠는 이벤트 레벨을 나타냅니다. 회색은 Info, 주황색은 Warning, 빨간색은 Critical 상태를 의미합니다. 사용자 선택 대상에 따른 제공 와탭 이벤트는 다음과 같습니다.

선택 대상	제공 이벤트
전체(All)	프로젝트 전체 이벤트
Namespace	Namespace 하위 전체 이벤트
Node	`onodeName`이 사용자 Node 이름과 일치
Deployment	`deployment`가 Deployment 이름과 일치
ReplicaSet	`replicaSetName`이 사용자 ReplicaSet 이름과 일치
Pod	`podName`이 사용자 Pod 이름과 일치
Container	`containerId`가 사용자 Container ID와 일치

와탭 이벤트는 `message`, `status`, `title`, `level` 등의 항목으로 구성되어 있습니다. 다음 예시 화면은 **WhaTap Containter 이벤트** 기록을 조회하고 있습니다.



- 1 **상세 보기** 왼쪽 요약 정보 카드에서 조회하려는 대상을 선택하세요.
- 2 요약 정보 목록에서 선택 대상의 정보를 확인하세요.
- 3 **이벤트** 탭에서 **WhaTap** 이벤트를 선택하세요.

  상세 보기 화면에 진입하기 전 **컨테이너 맵** 상단 뷰 옵션을 컨테이너 뷰로 지정해 **Container 이벤트**를 조회할 수 있습니다.

- 4 시간 선택자를 통해 조회하려는 시간을 선택하세요. 기본 설정은 **1일** 입니다.
- 5 바 차트를 통해 지정한 시간대의 **이벤트 수 추이**를 살펴보세요.

  지정 시간대 기준으로 이벤트 수 추이를 살펴보면 이벤트 2개가 카운트된 것을 확인할 수 있습니다.

- 6 이벤트 기록 목록에서 이벤트 기록 정보를 확인하세요. 목록 왼쪽 띠를 통해 이벤트별 레벨을 확인할 수 있습니다. 이벤트 메시지 및 태그 전체를 확인하려면 ▾ 아이콘을 선택하세요.

  이벤트 기록 목록에서 `컨테이너 Limit 기준 CPU 사용률이 85% 이상`, `GC 횟수 이상` Warning 레벨 이벤트 2개를 조회할 수 있습니다.

Call Information

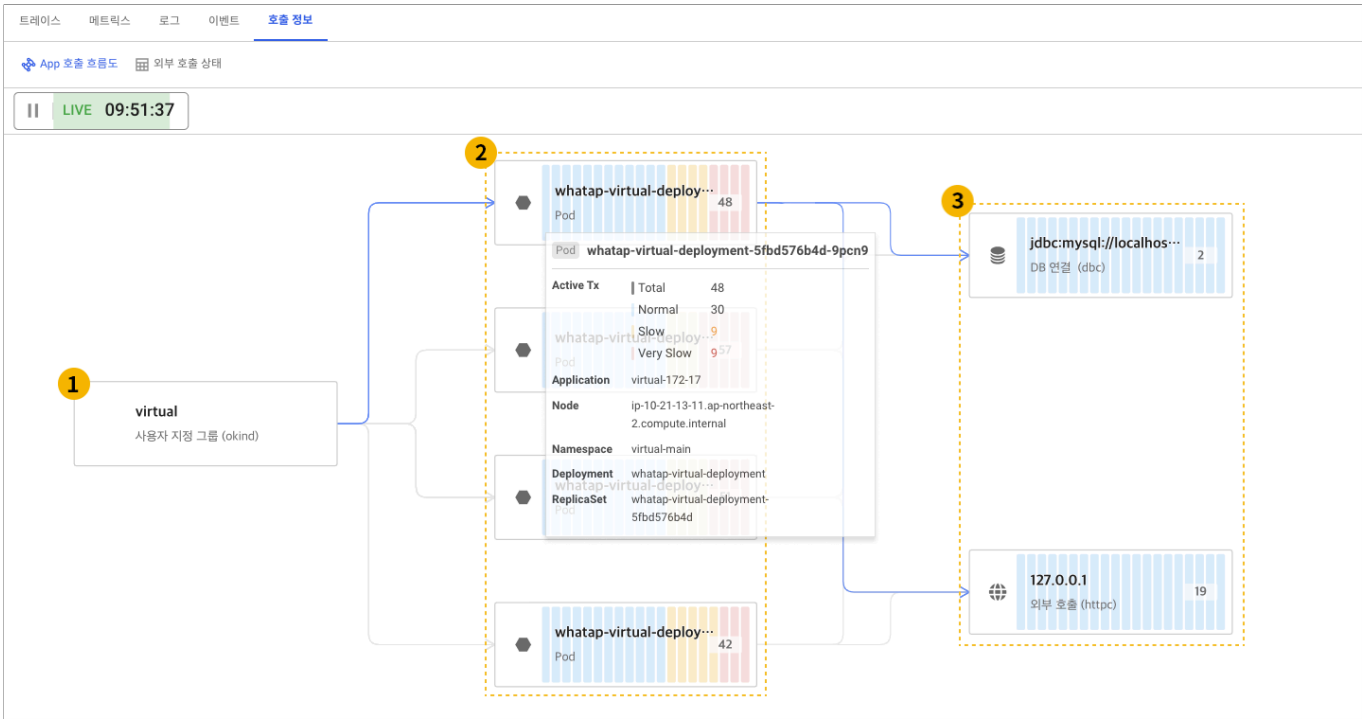
> **외부 호출 상태** 조회 기능은 애플리케이션이 Java인 경우만 제공합니다.

홈 화면 > 프로젝트 선택 > **대시보드** > **컨테이너 맵** > 대상 선택 후 **상세 분석** 화면 진입 > **호출 정보** 탭 선택

쿠버네티스 컨테이너 상에서 실행 중인 애플리케이션의 호출 흐름과 외부 호출 대상의 상태를 확인할 수 있습니다. 와탭 쿠버네티스 모니터링에서 제공하는 호출 정보는 다음과 같습니다.

- **App 호출 흐름도**
- **외부 호출 상태**

App 호출 흐름도



Pod 내부에서 실행 중인 컨테이너 애플리케이션의 호출 흐름을 표현한 차트를 제공합니다. 사용자 지정 그룹(Okind), Pod, DB 연결(DBC) 및 외부 호출(HTTPC) 세 가지의 노드간 연결 관계를 확인할 수 있습니다. 각 노드는 다음과 같은 정보를 제공합니다.

1 사용자 지정 그룹(okind)

사용자가 지정한 그룹 명칭을 의미합니다. 그룹(okind)을 지정하지 않은 경우 해당 노드는 표시되지 않습니다.

2 Pod

컨테이너 애플리케이션이 속해 있는 Pod 명칭을 의미합니다. 컨테이너 애플리케이션 내 액티브 트랜잭션 총 개수와 처리 속도를 확인할 수 있습니다. 선택 시 **액티브 트랜잭션** 메뉴로 이동합니다.

- **Active Tx**: 컨테이너 애플리케이션 내 액티브 트랜잭션을 의미합니다.
  - **Total**: 컨테이너 애플리케이션 내 총 액티브 트랜잭션의 개수를 의미합니다.
  - Normal: 0~3초 내 처리되는 트랜잭션의 개수를 의미합니다.
  - Slow: 3~8초 내 처리되는 트랜잭션의 개수를 의미합니다.
  - Very Slow: 처리에 8초 이상 소요되는 트랜잭션의 개수를 의미합니다.
- **Application**: Pod 내부의 Application 명칭을 의미합니다.
- **Node**: Pod가 속한 Node 명칭을 의미합니다.
- **Namespace**: Pod가 속한 Namespace 명칭을 의미합니다.
- **ReplicaSet**: Pod가 속한 ReplicaSet 명칭을 의미합니다.
- **Deployment**: Pod가 속한 Deployment 명칭을 의미합니다.

3 DB 연결 및 외부 호출

- **DB 연결(dbc)**: Pod 내부의 컨테이너 애플리케이션이 호출한 DB Connection을 의미합니다. 선택 시 **외부 호출 상태** 조회 탭으로 이동합니다.
  - **Active Tx**: DB Connection 중인 컨테이너 애플리케이션 내 액티브 트랜잭션을 의미합니다.
  - **Total**: 컨테이너 애플리케이션 내 총 액티브 트랜잭션의 개수를 의미합니다.
  - Normal: 0~3초 내 처리되는 트랜잭션의 개수를 의미합니다.
  - Slow: 3~8초 내 처리되는 트랜잭션의 개수를 의미합니다.
  - Very Slow: 처리에 8초 이상 소요되는 트랜잭션의 개수를 의미합니다.
- **외부 호출(httpc)**: Pod 내부의 컨테이너 애플리케이션이 호출한 외부 API를 의미합니다. 선택 시 **외부 호출 상태** 조회 탭으로 이동합니다.

- **Active Tx**: 외부 API 호출 중인 컨테이너 애플리케이션 내 액티브 트랜잭션을 의미합니다.
- **Total**: 컨테이너 애플리케이션 내 총 액티브 트랜잭션의 개수를 의미합니다.
- Normal: 0~3초 내 처리되는 트랜잭션의 개수를 의미합니다.
- Slow: 3~8초 내 처리되는 트랜잭션의 개수를 의미합니다.
- Very Slow: 처리에 8초 이상 소요되는 트랜잭션의 개수를 의미합니다.

외부 호출 상태

외부 호출 로그를 이용해 모니터링 대상이 호출하는 목적지의 연결 상태 정보를 기반으로 외부 호출 대상의 정상 여부 정보를 제공합니다. **외부 호출 상태** 조회 기능을 통해 접속 불가 대상의 가시성을 제공합니다. 이를 통해 서비스 이슈 발생 시 문제 발생 지점을 용이하게 파악할 수 있습니다. 또한 지속적인 접속 실패에 의한 스레드와 커넥션 증가 등의 자원 잠식 현상을 사전에 감지하고 이에 대응할 수 있습니다.

외부 호출 로그

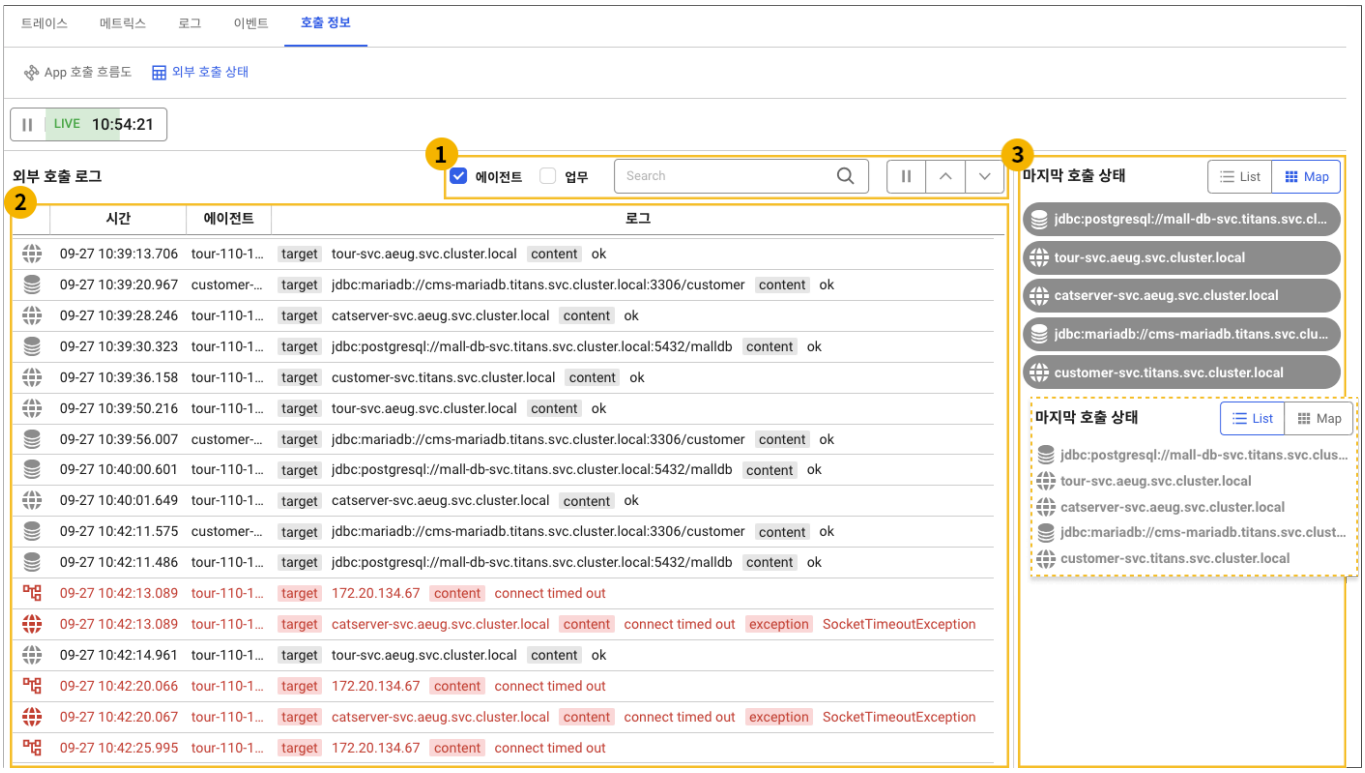
애플리케이션 에이전트 설정을 통해 각 모니터링 대상별 외부 호출에 대한 성공 및 실패 여부를 기록한 로그입니다.

- 외부 호출 유형은 다음과 같습니다.

  `httpcall` , `db` , `socket`

- 외부 호출 로그를 수집하려면 **에이전트 설정**에 다음 옵션을 추가해야 합니다.

```
logsink_rt_enabled=true
logsink_enabled=true
```



외부 호출 상태 탭에서 외부 호출 로그를 목록 형식으로 조회할 수 있습니다. 최근 24시간 범위의 로그를 최대 500개 불러와 표시합니다. 실시간 조회 시 데이터는 3초 간격으로 업데이트됩니다.

1 상단 바 옵션

- **에이전트** 또는 **업무** 컬럼의 표시 여부를 선택할 수 있습니다.
- 검색어 입력창을 통해 해당 검색어가 속한 에이전트, 업무, 로그가 포함된 열을 필터링할 수 있습니다.
- ▶ **재생**: 로그가 업데이트 되었을 시 최하단으로 이동해 가장 최근 로그를 확인할 수 있습니다.
- ⏸ **일시 정지**: 자동 스크롤 이동을 일시적으로 중지합니다.
- ▲ ▼ : 최상단 또는 최하단으로 이동합니다.

2 외부 호출 로그 목록

아이콘과 태그를 통해 외부 호출 로그의 주요 내용을 쉽게 확인할 수 있습니다. 에러 로그의 경우 빨간색으로 표시됩니다.

- 외부 호출 유형은 아이콘 (**db**), (**httpcall**), (**socket**)을 통해 안내합니다.
- `target` : 외부 호출 및 연결 대상을 의미합니다.

- `content` : 로그의 내용을 의미합니다.
- `exception` : 에러 발생 시 기록된 로그의 내용을 의미합니다.

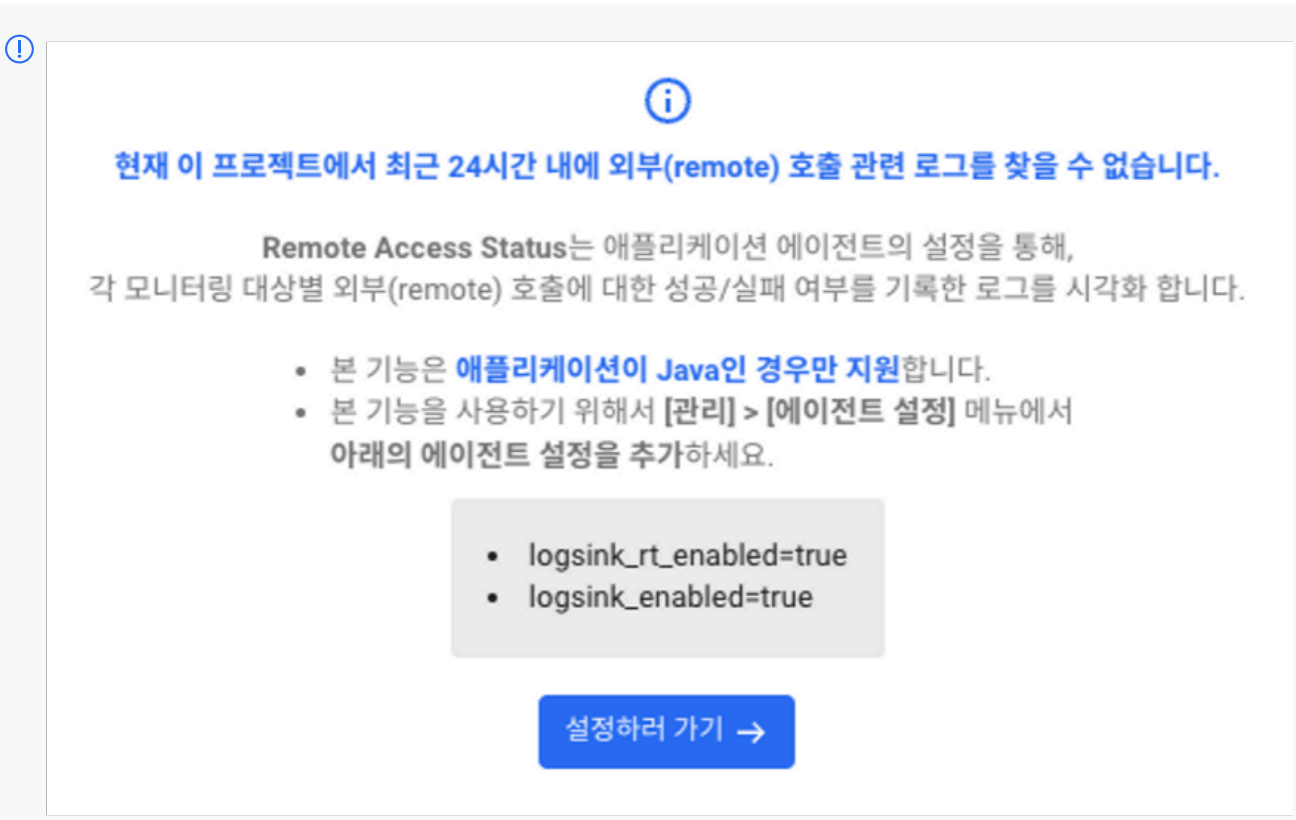
3 마지막 호출 상태

조회된 모든 로그의 외부 호출 대상 목록을 확인합니다. 가장 최근 연결 상태에 따라 성공 또는 실패로 표시됩니다.

- 조회 모드
  - **List**: 기본 목록 뷰입니다. 전체 목록의 길이가 화면 높이를 넘어갈 경우 스크롤이 생깁니다.
  - **Map**: 목록이 화면을 넘어가게되면 목록의 길이를 줄여 가로 방향으로 외부 호출 대상을 나열합니다.
- 필터

  외부 호출 대상을 선택 시 해당 외부 호출 대상에 대한 로그만 필터링해 외부 호출 로그 목록을 조회할 수 있습니다. 재선택 시 해제됩니다.

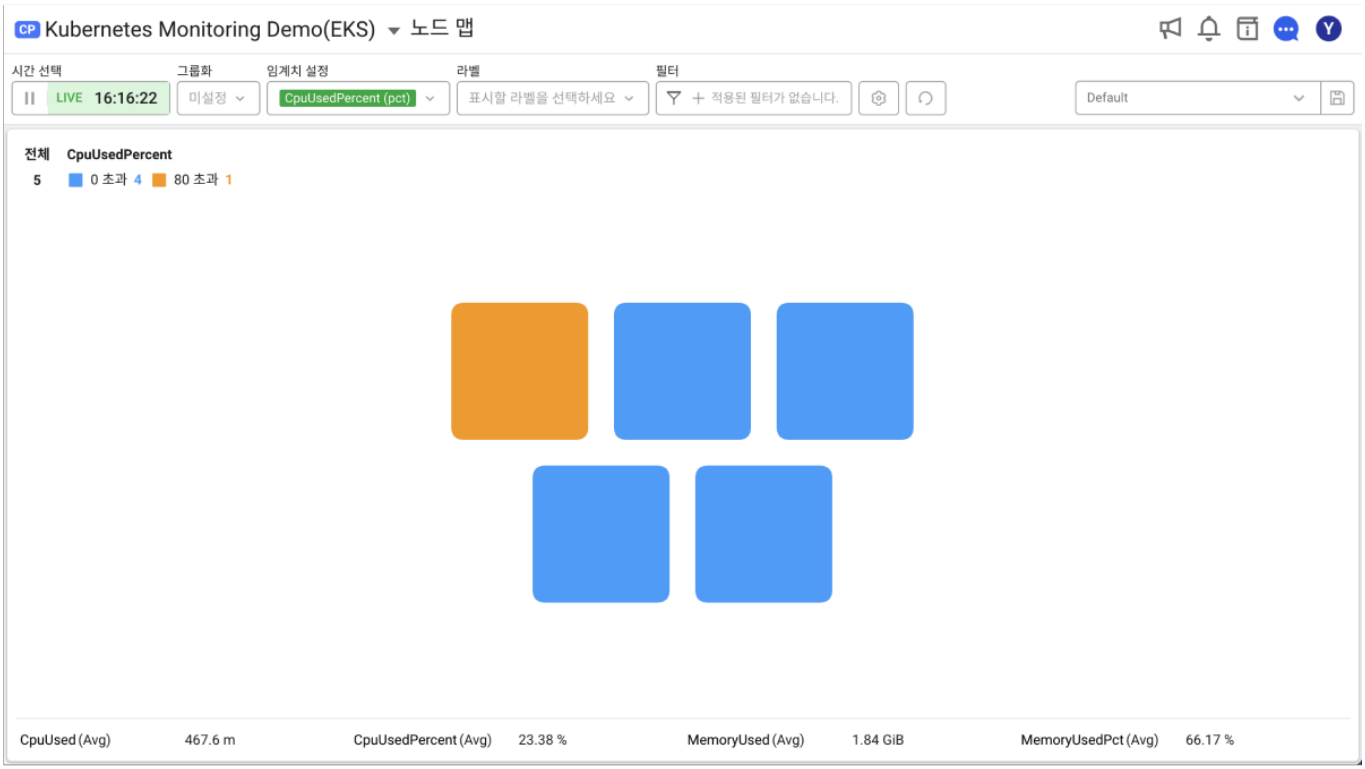
> ⓘ **데이터가 없는 경우**
>
> 실시간 조회 시 최근 24시간 동안 외부 호출 로그가 수집되었는지 확인합니다. 24시간 동안 외부 호출 로그가 수집되지 않은 경우 다음과 같은 안내 메시지를 제공합니다.


현재 이 프로젝트에서 최근 24시간 내에 외부(remote) 호출 관련 로그를 찾을 수 없습니다.
Remote Access Status는 애플리케이션 에이전트의 설정을 통해,
각 모니터링 대상별 외부(remote) 호출에 대한 성공/실패 여부를 기록한 로그를 시각화 합니다.
본 기능은 애플리케이션이 Java인 경우만 지원합니다.
본 기능을 사용하기 위해서 [관리] > [에이전트 설정] 메뉴에서
아래의 에이전트 설정을 추가하세요.
logsink_rt_enabled=true
logsink_enabled=true
설정하러 가기

노드 맵

> **요구 사항**
>
> - 와탭 쿠버네티스 에이전트 1.4.1 버전 이상이 필요합니다.
> - 쿠버네티스 클러스터 프로젝트(CP)일 경우 해당 메뉴를 조회할 수 있습니다.

노드 맵 메뉴는 클러스터 관점에서 노드의 상태를 모니터링하는 기능을 제공합니다. 노드의 상태, 리소스 사용량, 에이전트 상태 정보를 시각화하여 사용자가 직관적으로 이해할 수 있도록 돕습니다. 또한, 과거 시점의 노드 상태를 조회할 수 있어, 이전의 상태를 쉽게 확인할 수 있는 기능을 지원합니다.

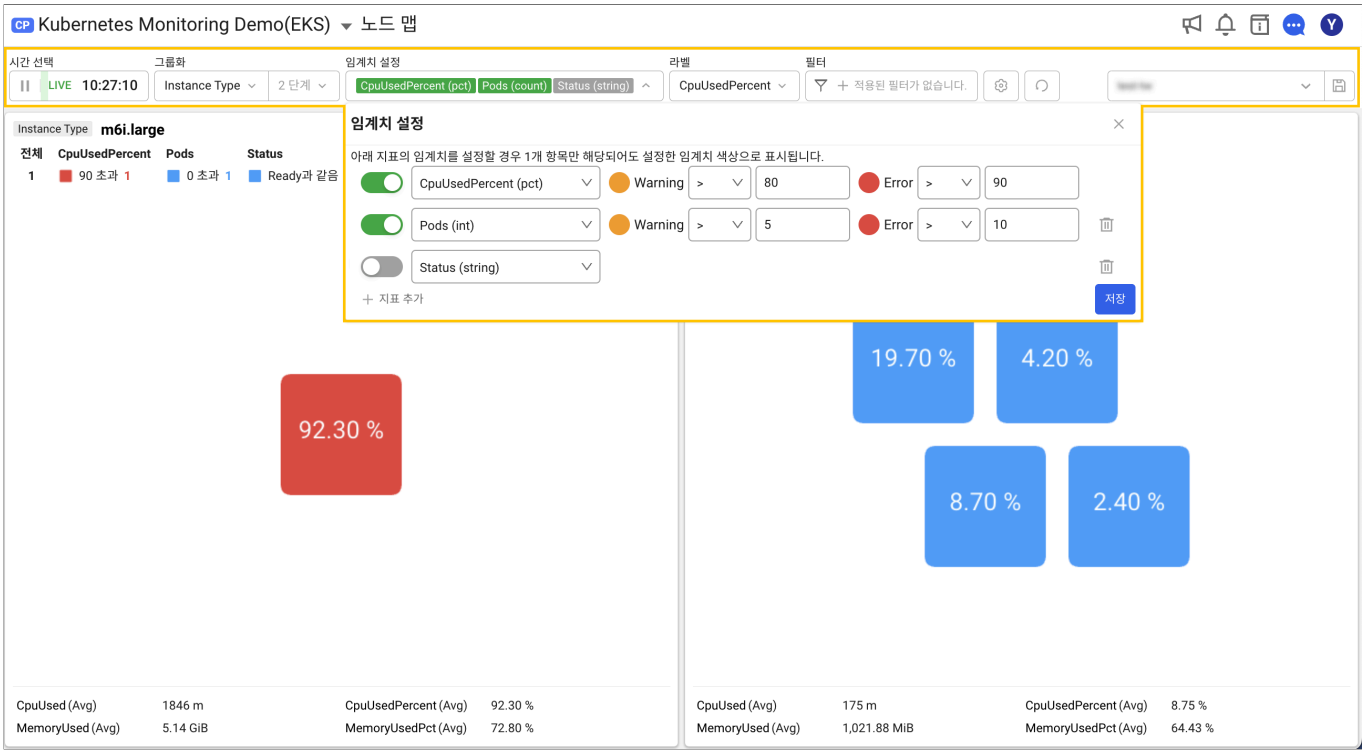


기본 화면 안내

노드 맵 메뉴는 크게 **사용자 설정** 영역과 **노드 맵 차트** 영역으로 구성되어 있습니다.

사용자 설정

노드 맵 상단의 **사용자 설정** 영역에서는 **그룹화**, **임계치 설정**, **라벨**, **필터**, 프리셋 및 ⚙ 추가 설정을 구성할 수 있습니다. 다음 예시 이미지는 **그룹화**, **임계치 설정**, **라벨**이 적용되어있습니다.



그룹화

사용자는 다양한 조건에 따라 노드를 **그룹화**하여 표시할 수 있습니다. 이를 통해 각 그룹 내 노드의 관계를 쉽게 파악할 수 있습니다. **그룹화** 기능은 최대 2단계까지 지원됩니다. **노드 맵 그룹화**를 통해 사용자는 특정 기준으로 노드를 선별하여 맞춤형 모니터링 환경을 구성할 수 있습니다.

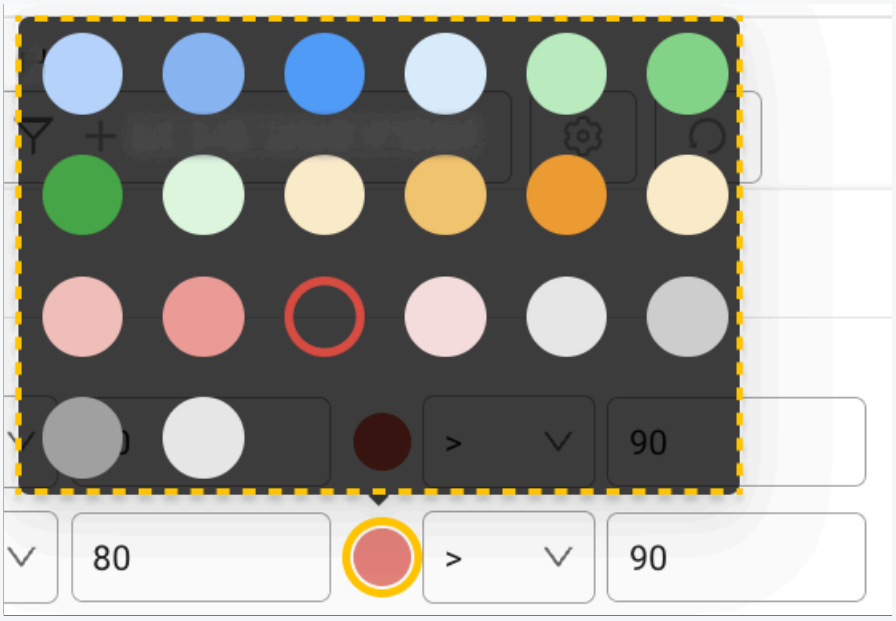
임계치 설정

사용자는 **임계치 설정**을 통해 노드 상태를 색상으로 표현할 수 있습니다. 지표는 하나 이상 설정 가능하며 각 지표별로 경고(Warning)와 오류(Error) 구간을 지정할 수 있습니다. 정상(Normal) 구간은 파란색으로 표시됩니다. 설정한 범위에 따라 해당 상태에 맞는 색상이 적용됩니다.

1. **임계치 설정** 옵션을 클릭하세요.

2. 원하는 지표를 선택 후 각 Warning, Error 구간을 설정하세요.
3. 지표 추가를 원하는 경우 하단의 **+ 지표 추가** 버튼을 클릭하세요.
4. 원하는 지표별 임계치 설정을 마쳤으면 하단의 **저장** 버튼을 클릭하세요.

> ⓘ 이상 상태 구간인 Warning, Error는 기본적으로 주황색, 빨간색으로 설정되어 있습니다. **임계치 설정**에서 각 상태에 대한 색상을 사용자가 지정할 수 있습니다.



문자열 타입 지표

문자열 타입(String)의 지표일 경우 특정 문자열이 포함될 때 Error로 지정됩니다. 예를 들어 **Status (string)** 지표에 `NotReady` 라는 텍스트가 포함될 경우 해당 노드는 Error로 지정되어 빨간색으로 표시됩니다.

지표 명	값	Error 상태
Status	`Ready *` / `NotReady *`	`NotReady *`
Ready	`true` / `false`	`false`
PIDPressure	`true` / `false`	`true`

지표 명	값	Error 상태
DiskPressure	`true` / `false`	`true`
MemoryPressure	`true` / `false`	`true`
Unschedulable	`true` / `false`	`true`

노드 상태 표시 우선 순위

상태 표시 우선 순위는 Error 구간이 가장 높습니다. 여러 지표 중 하나의 항목이라도 Error 구간에 속할 경우 해당 노드는 우선적으로 빨간색으로 표시됩니다. 다음 예시의 경우 노드 a와 노드 b 모두 Error로 표시됩니다.

라벨

사용자는 노드 블록 내부에 텍스트 라벨을 설정하여 지정한 주요 정보를 한눈에 볼 수 있습니다. **라벨**을 **CpuUsedPercent** 지표 기준으로 설정한 경우 왼쪽 예시 이미지와 같이 노드 블록 내부에서 해당 정보를 확인할 수 있습니다.

필터

사용자는 필터를 통해 특정 조건을 만족하는 노드를 선택하여 집중적으로 모니터링할 수 있습니다.

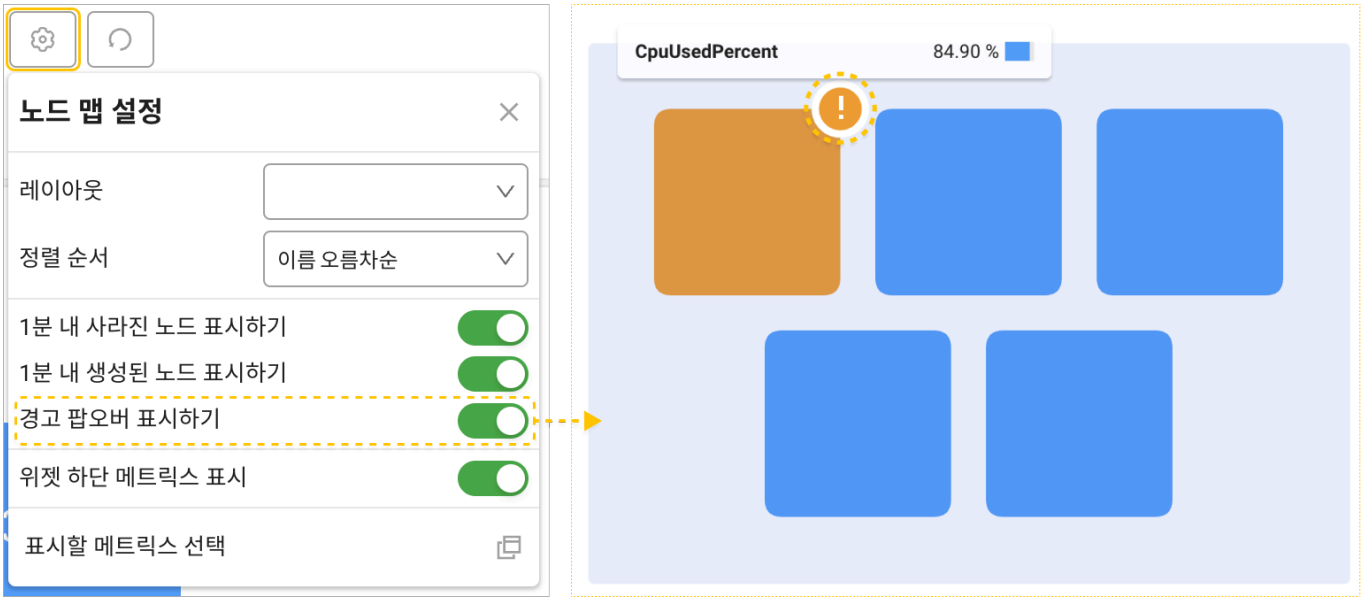
프리셋

사용자는 설정한 옵션을 프리셋으로 저장하고 필요 시 불러올 수 있습니다. 프리셋은 프로젝트 단위로 저장됩니다. 이를 통해 프로젝트 멤버들이 동일한 설정으로 클러스터를 모니터링할 수 있습니다. 기본 프리셋(**Default**)을 제공하며 기본 프리셋의 경우 삭제할 수 없습니다.

> **기본 프리셋**
>
> **노드 맵**의 기본 프리셋은 **CpuUsedPercent (pct)** 지표의 임계치가 다음과 같이 설정되어있습니다.
>
> - Normal: 값이 `80` 이하
> - Warning: 값이 `80` 초과 `90` 이하
> - Error: 값이 `90` 초과

추가 설정

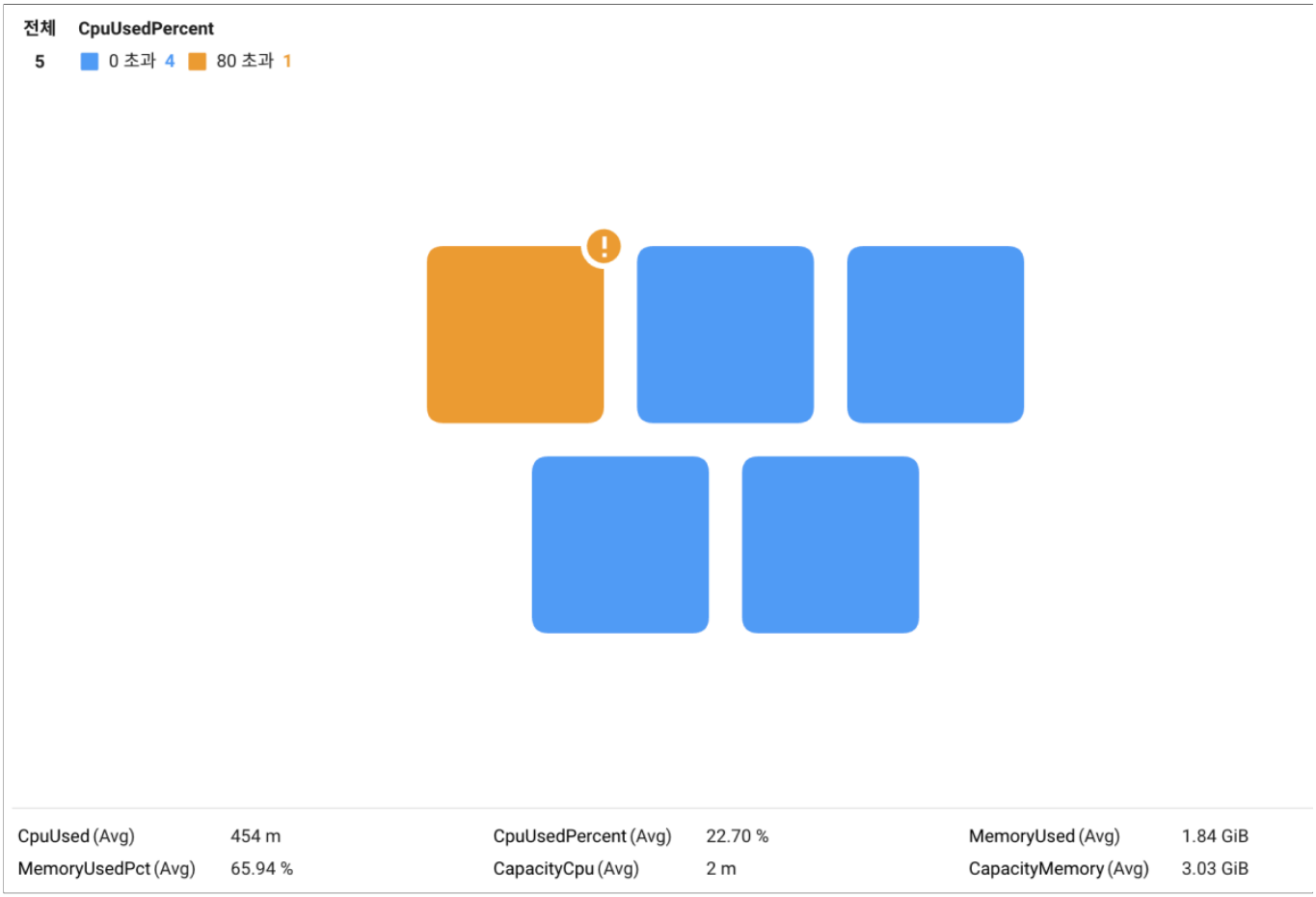
노드 맵 오른쪽 상단의 ⚙ 아이콘을 통해 다음과 같은 추가적인 세부 설정을 조정할 수 있습니다.



- **레이아웃**: 그룹화된 위젯의 컬럼을 고정합니다.
- **정렬 순서**: 그룹화된 위젯을 노드 수 또는 이름 기준으로 정렬합니다.
- **1분 내 사라진 노드 표시하기**: 최근 1분 이내에 사라진 노드를 표시합니다.
- **1분 내 생성된 노드 표시하기**: 최근 1분 이내에 생성된 노드를 표시합니다.
- **경고 팝오버 표시하기**: 특정 노드가 **임계치 설정**에 따라 Warning 또는 Error 상태일 때 경고 팝오버를 표시합니다.
- **위젯 하단 메트릭스 표시**: 위젯 하단에 평균, 합계, 최대값, 최소값 표시 여부를 설정합니다.
- **표시할 메트릭스 선택**: 위젯 하단에 표시할 지표를 선택합니다. 특정 지표를 선택하고 그룹화된 모든 노드에 대하여 해당 지표를 평균, 합계, 최대값, 최소값 중 원하는 방식으로 데이터를 표시하도록 설정합니다.

노드 맵 차트

노드 맵 차트는 다음 세 가지 영역으로 구성되어있습니다.



1 위젯 상단 영역

노드의 전체 개수 또는 그룹화된 경우 그룹회된 항목들의 총 개수와 선택한 메트릭스의 상태별 개수를 표시합니다.

2 중앙 시각화 영역

전체 노드 상태와 개수를 직관적으로 확인할 수 있는 시각화된 대시보드를 제공합니다.

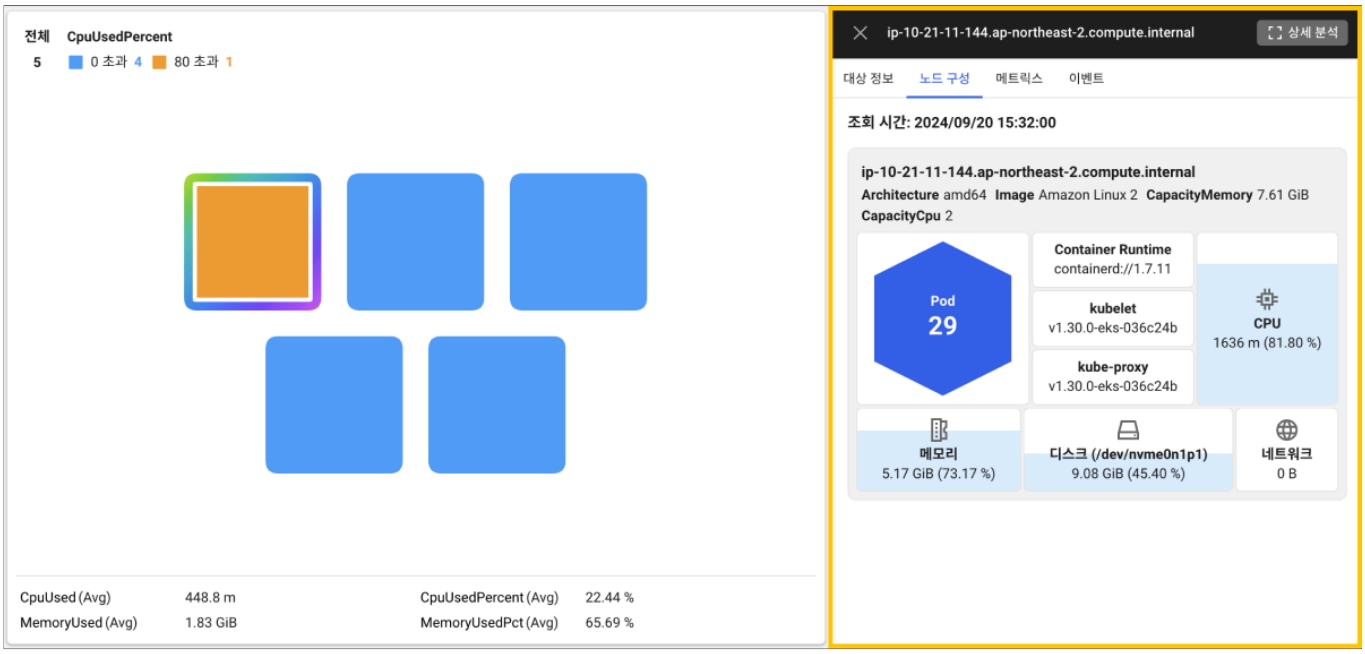
3 위젯 하단 영역

선택한 메트릭스의 통계값을 표시합니다. 추가 설정의 **표시할 메트릭스 선택**을 통해 설정한대로 합계, 최대값, 최소값 중 원하는 방식으로 데이터를 표시합니다.

> ⓘ **노드 맵**에서는 노드 에이전트가 설치된 노드만 표시합니다.

요약 보기 및 상세 분석

노드 맵에서 조회를 원하는 노드를 선택하세요. 오른쪽에 나타나는 요약 보기 화면을 통해 해당 객체의 요약 정보를 확인할 수 있습니다. 요약 보기 화면에서 **상세 분석** 버튼을 클릭해 상세 분석 화면을 확인할 수 있습니다.



대상 정보

대상 정보 탭에서 사용자가 선택한 노드의 주요 속성 정보를 확인할 수 있습니다. 또한 노드 목록 및 컨테이너 맵 메뉴로 이동할 수 있습니다.

> ⓘ **Labels** 지표는 쿠버네티스 에이전트 버전 1.7.9 이상부터 지원합니다.

노드 요약 분석

노드의 자원 할당 및 사용 현황 요약 정보를 **노드 요약 분석** 탭을 통해 확인할 수 있습니다.

- 기본 정보: 노드의 클러스터 버전, 기동 시간, 현재 상태, 스케줄링 가능 여부, 내부 IP 주소, 그리고 리전 및 존 정보를

제공합니다.

- 운영체제 및 하드웨어 정보: 노드가 실행되는 운영체제, OS 이미지, CPU 아키텍처, 코어 수, 총 메모리 크기를 제공합니다. 노드의 메모리와 CPU에 대한 할당 가능 및 현재 사용량을 확인할 수 있습니다.
- 리소스 사용 현황: 노드에서 실행 중인 모든 컨테이너의 요청(Request)과 할당 가능 자원(CPU, 메모리)의 차이를 통해 잔여 자원을 파악합니다. 실제 CPU 사용률과 메모리 사용률을 포함하여 리소스 사용 상태를 요약합니다.
- 디스크 사용 현황: 노드에 연결된 디스크 정보, 파일 시스템 유형, 디바이스 ID, 마운트 지점, 총 용량 및 사용 중인 용량, 그리고 사용률을 제공합니다.
- Pod 상태 및 리소스 여유: 노드의 할당 가능한 전체 Pod 수와 현재 실행 중인 Pod 수를 비교하여, 추가 실행 가능한 Pod 수를 확인할 수 있습니다. 또한 MemoryPressure, DiskPressure, PidPressure 등의 상태를 통해 노드의 안정성을 점검합니다.

노드 구성

노드 구성 탭에서 선택한 노드 구성 및 전반적인 리소스 사용량 정보를 요약해서 제공합니다. 해당 노드의 하드웨어 및 소프트웨어 구성 요소를 종합적으로 확인할 수 있습니다. 사용자는 노드의 리소스 사용 현황 및 각 구성 요소의 상태를 파악하고 성능 및 운영 안정성을 모니터링할 수 있습니다.

- **노드 시스템 사양**
  - **Architecture**: 노드가 사용하고 있는 하드웨어 아키텍처
  - **Image**: 노드가 실행 중인 운영 체제 이미지 정보
  - **CapacityMemory**: 노드의 총 메모리 용량
  - **CapacityCpu**: 노드의 총 CPU 용량
- **소프트웨어 구성**

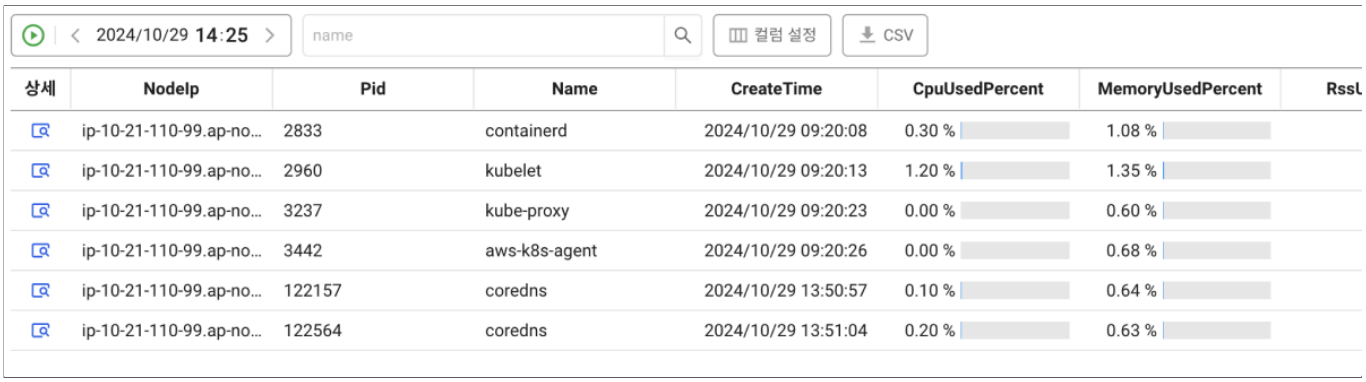
- **Container Runtime**: 노드에서 사용되는 컨테이너 런타임의 버전 정보
- **kubelet**: 노드에서 실행 중인 쿠버네티스 kubelet의 버전 정보
- **kube-proxy**: 네트워크 관리 역할을 하는 kube-proxy의 버전 정보

- **리소스 사용 현황**
  - **CPU**: 노드의 CPU 사용량
  - **메모리**: 노드의 메모리 사용량
  - **디스크**: 노드에서 사용 중인 디스크 용량과 사용 비율
  - **네트워크**: 네트워크 전송량

> - CPU, 메모리, 디스크, 네트워크 클릭 시 조회 기간 동안 사용량 추이를 확인할 수 있는 **메트릭스** 탭으로 이동합니다.
> - **AVG**, **MAX** 버튼을 클릭하여 조회 기간동안 평균 사용량과 최대 사용량을 확인할 수 있습니다.

프로세스

프로세스 탭에서 선택한 노드 내부에 있는 주요 쿠버네티스 관리 프로세스 정보를 확인할 수 있습니다. 프로세스 문제로 발생하는 클러스터 장애를 예방하고 다운타임을 최소화할 수 있습니다.



상세	NodeIp	Pid	Name	CreateTime	CpuUsedPercent	MemoryUsedPercent	RssU
	ip-10-21-110-99.ap-no...	2833	containerd	2024/10/29 09:20:08	0.30 %	1.08 %	
	ip-10-21-110-99.ap-no...	2960	kubelet	2024/10/29 09:20:13	1.20 %	1.35 %	
	ip-10-21-110-99.ap-no...	3237	kube-proxy	2024/10/29 09:20:23	0.00 %	0.60 %	
	ip-10-21-110-99.ap-no...	3442	aws-k8s-agent	2024/10/29 09:20:26	0.00 %	0.68 %	
	ip-10-21-110-99.ap-no...	122157	coredns	2024/10/29 13:50:57	0.10 %	0.64 %	
	ip-10-21-110-99.ap-no...	122564	coredns	2024/10/29 13:51:04	0.20 %	0.63 %	

프로세스 상세

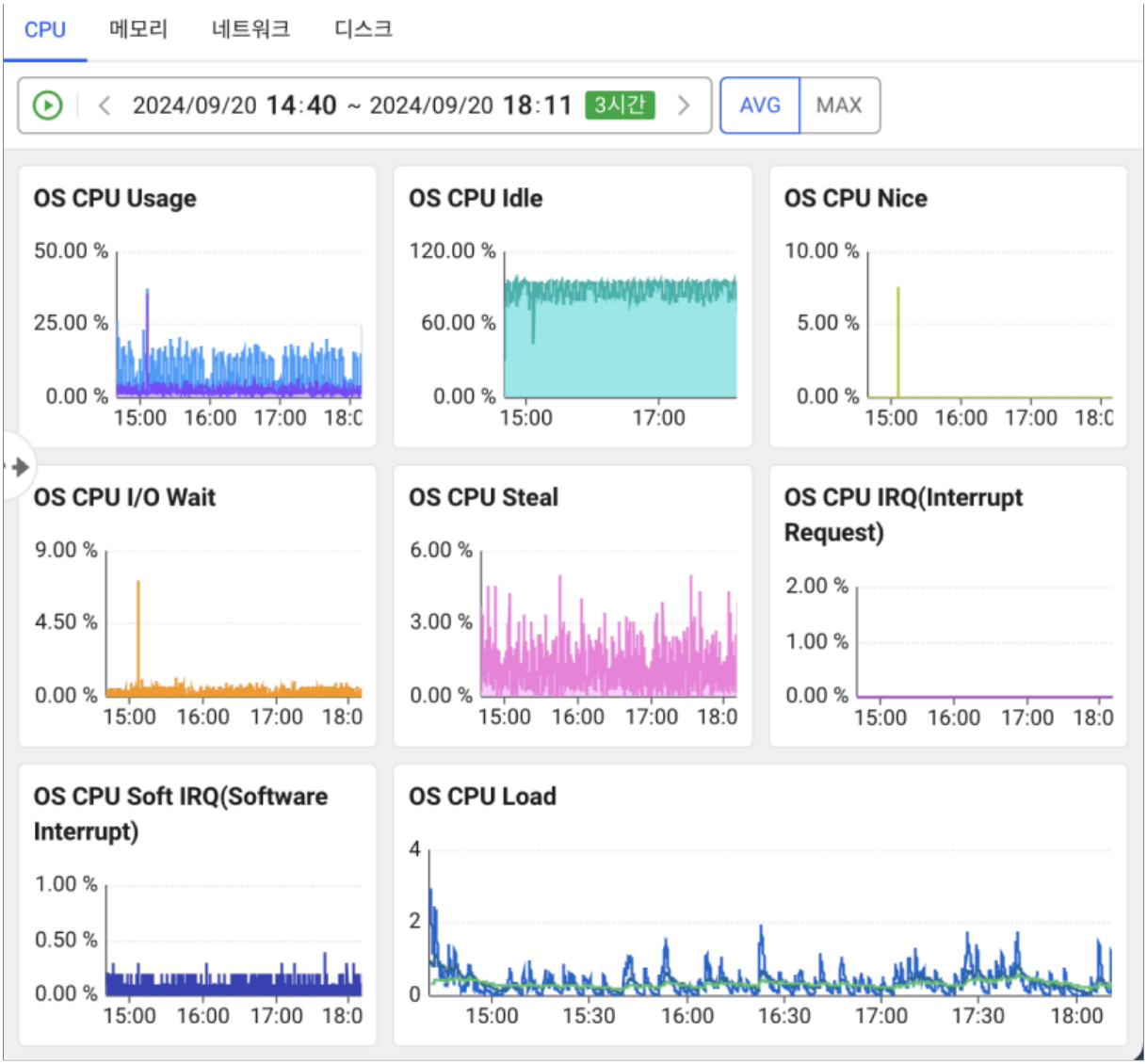
- 프로세스 목록 왼쪽 **상세** 컬럼에서 아이콘 선택 시 해당 프로세스에 대한 상세 정보를 확인할 수 있습니다.
- **AVG** 또는 **MAX** 버튼을 클릭하여 3일 이상의 장기 데이터 조회 시 불러올 통계 데이터의 합산 방식을 선택할 수 있습니다.

- 프로세스 주요 정보 값 클릭 시 해당 값이 복사됩니다.

> ⓘ **프로세스** 탭은 쿠버네티스 에이전트 버전 1.7.14 이상부터 지원합니다.

메트릭스

메트릭스 탭에서 선택한 노드의 리소스 사용량 추이를 확인할 수 있습니다. 조회 기간 동안의 평균(**AVG**), 최대값(**MAX**)을 기준으로 **CPU**, **메모리**, **네트워크**, **디스크** 사용량의 변화를 조회할 수 있습니다.

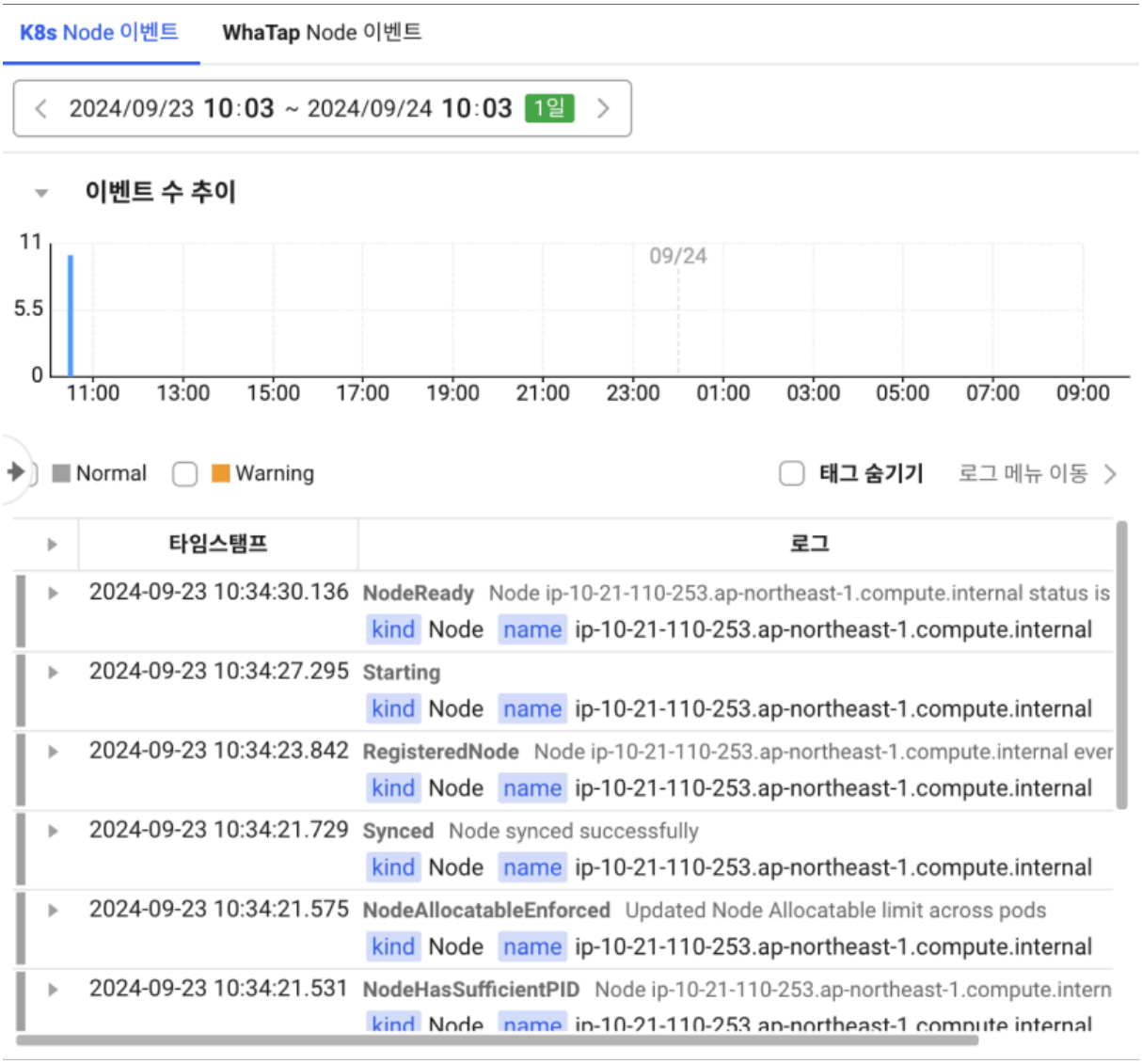


- **AVG**: 조회 기간 동안 평균 사용량
- **MAX**: 조회 기간 동안 최대 사용량
- **Total**: 여러 네트워크 인터페이스의 전체 트래픽 사용량 합산
- **In/Out**: 특정 네트워크 인터페이스의 Inbound(수신) 및 Outbound(송신) 트래픽 사용량

> ⓘ **K8s CPU Request / Allocatable CPU (%)** 위젯은 쿠버네티스 에이전트 버전 1.7.10 이상부터 지원합니다.

이벤트

이벤트 탭에서 선택한 노드에서 발생한 **K8s Node 이벤트**와 **WhaTap Node 이벤트**를 확인할 수 있습니다. **K8s Node 이벤트**는 노드의 상태 변화를 포함한 클러스터 내의 중요한 작업 기록을 포함하며, **WhaTap Node 이벤트**는 사용자가 설정한 경고 알림 기준에 따른 이벤트 발생 정보를 제공합니다.


K8s Node 이벤트
WhaTap Node 이벤트
2024/09/23 10:03 ~ 2024/09/24 10:03 1일
이벤트 수 추이
11
5.5
0
09/24
11:00
13:00
15:00
17:00
19:00
21:00
23:00
01:00
03:00
05:00
07:00
09:00
Normal
Warning
태그 숨기기
로그 메뉴 이동
타임스탬프
로그
2024-09-23 10:34:30.136 NodeReady Node ip-10-21-110-253.ap-northeast-1.compute.internal status is
kind Node name ip-10-21-110-253.ap-northeast-1.compute.internal
2024-09-23 10:34:27.295 Starting
kind Node name ip-10-21-110-253.ap-northeast-1.compute.internal
2024-09-23 10:34:23.842 RegisteredNode Node ip-10-21-110-253.ap-northeast-1.compute.internal ever
kind Node name ip-10-21-110-253.ap-northeast-1.compute.internal
2024-09-23 10:34:21.729 Synced Node synced successfully
kind Node name ip-10-21-110-253.ap-northeast-1.compute.internal
2024-09-23 10:34:21.575 NodeAllocatableEnforced Updated Node Allocatable limit across pods
kind Node name ip-10-21-110-253.ap-northeast-1.compute.internal
2024-09-23 10:34:21.531 NodeHasSufficientPID Node ip-10-21-110-253.ap-northeast-1.compute.intern
kind Node name ip-10-21-110-253.ap-northeast-1.compute.internal

퍼포먼스 요약

최근 시간별 차트를 통하여 프로젝트 내의 와탭 노드 에이전트 및 애플리케이션 에이전트가 설치된 대상의 정보를 표시합니다.
네임스페이스 프로젝트로 분리된 정보는 네임스페이스 프로젝트에서 확인할 수 있습니다.

- **바로 가기 옵션**

  위젯별 오른쪽 상단에 위치한 ⋮ **더보기** 아이콘을 선택하여 🔗 **바로 가기** 옵션을 확인할 수 있습니다.

- **라인 차트 설정 옵션**

  라인 차트 위젯별 오른쪽 상단에 위치한 ⋮ **더보기** 아이콘을 선택 후 또는 바로 ⚙ 설정 옵션을 확인할 수 있습니다. 라인 차트의 경우 설정 옵션을 통해 id가 되는 값을 변경할 수 있습니다. 해당 설정은 저장되지 않습니다.

  예, **Node CPU (%)** 위젯의 경우 **Node**별 데이터 조회가 가능합니다. 메트릭스를 변경해 **ContainerName**별로 조회할 수 있습니다.

> ⓘ 에이전트 설치가 정상적으로 완료되지 않은 경우 데이터가 표시되지 않을 수 있습니다.

노드 정보

위젯으로 제공하는 노드 정보는 다음과 같습니다.

- 최근 30분 차트
  - **Node CPU (%)**, **Node Memory (%)** 등

> ⓘ
> - 노드 수량 위젯 오른쪽 위에 ⋮ **더보기** 아이콘 > 🔗 **바로 가기** 아이콘을 차례로 선택하면 **클러스터** > **노드 목록** 메뉴로 이동합니다.
> - **Node CPU (%)** 위젯과 **Node Memory (%)** 위젯 오른쪽 위에 ⋮ **더보기** 아이콘 > 🔗 **바로 가기** 아이콘을 차례로 선택하면 **분석** > **메트릭스 차트** 메뉴로 이동합니다. 해당 메뉴를 사용하면 더욱 자세한 조건을 통해 차트를 확인할 수 있습니다.

컨테이너 정보

위젯으로 제공하는 컨테이너 정보는 다음과 같습니다.

- 최근 10분 차트
  - **Container Status**, **Container CPU (%)**, **Container Memory (%)** 등

애플리케이션 정보

위젯으로 제공하는 애플리케이션 정보는 다음과 같습니다.

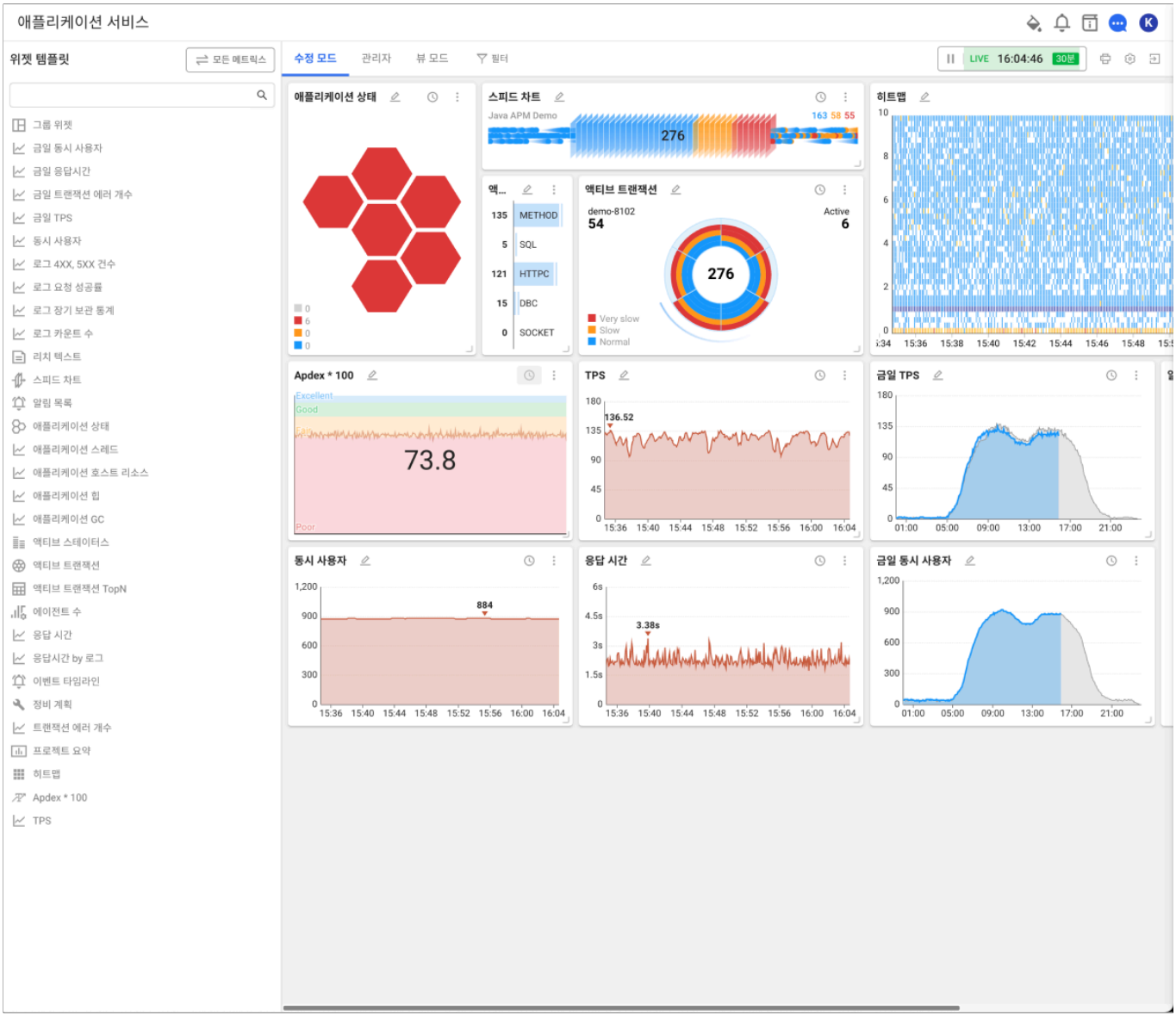
- 최근 10분 차트
  - **Active TX**, **TPS**, **Response Time** 등

> ⓘ 애플리케이션 정보는 애플리케이션 에이전트 설치 대상이 존재할 때 표시합니다.

Flex 보드

Flex 보드는 사용자 정의 통합 대시보드입니다. 애플리케이션, 서버, 데이터베이스, 컨테이너 등 프로젝트의 데이터로 실시간 대시보드를 만들 수 있습니다.

사전 구성된 템플릿으로 초기 설정을 빠르게 마치고 원하는 형태의 대시보드를 만들 수 있습니다. 데이터 위젯을 추가하거나 속성을 수정해 필요한 형식으로 표시할 수 있으며, 필터링 기능으로 모니터링 대상을 간추릴 수도 있습니다. 시간 범위를 지정하면 특정 시점의 데이터를 확인할 수 있고, 보조 차트를 활용해 다양한 방식으로 분석할 수 있습니다. 대시보드는 즐겨찾기에 등록해 쉽게 접근할 수 있으며, 개인화한 대시보드는 다른 계정으로 복사해 재사용할 수 있습니다.



홈 화면 > 통합 Flex 보드

- 위젯 생성 시 조회 가능한 모든 프로젝트를 선택 옵션으로 제공합니다.
- 사용자 계정에 대시보드가 저장되며 다른 사용자에게 복사하기 기능을 이용해 공유할 수 있습니다.
- 개인 계정 대시보드로 권한에 따른 영향은 없으나 읽기 전용으로 공유된 대시보드의 경우 수정할 수 없습니다.

홈 화면 > 프로젝트 선택 > 대시보드 > Flex 보드

- 위젯 생성 시 해당 프로젝트 정보를 자동 입력합니다.

- 프로젝트 멤버들에게 생성한 Flex 보드가 자동 공유됩니다.
- 프로젝트 **수정** 권한 또는 **프로젝트 플렉스보드 편집** 권한, **Site Admin** 권한을 가진 사용자가 수정할 수 있습니다.

Flex 보드의 수정 권한이 있는 사용자는 다음 기능을 이용할 수 있습니다.

- 대시보드를 JSON 파일로 내보내기/가져오기
- 대시보드 내의 데이터 요청 및 응답 내용 확인
- 위젯 설정 옵션을 JSON 형식으로 조회 및 수정

프로젝트 내 **Flex 보드** 메뉴에서는 대시보드 수정 권한이 있는 사용자만 **수정 모드** 및 **관리자** 모드, **필터** 기능에 접근할 수 있습니다.

- **접근 가능 멤버 권한**
  - **프로젝트 수정** 권한
  - **프로젝트 플렉스보드 편집** 권한
  - **Site Admin** 권한
- **뷰 모드** 및 **필터** 기능에 대한 자세한 내용은 다음 문서를 참고하세요.
- 멤버 권한에 대한 자세한 내용은 다음 문서를 참고하세요.

Flex 보드 사용하기

> 대시보드를 메트릭스 위젯을 포함해 다양한 위젯 템플릿을 활용해 구성할 수 있습니다.
> 각 템플릿의 특징과 사용법은 위젯 템플릿 문서에서 자세히 확인할 수 있습니다.

사전 템플릿 적용하기

사전 구성 템플릿으로 초기 설정을 빠르게 진행하고 필요에 따라 수정해 활용하세요.

1. **Flex 보드** 메뉴를 선택하세요.
2. 왼쪽 **템플릿** 목록에서 원하는 템플릿을 선택하세요.
3. **Flex 보드 관리** 창이 나타나면 **이름 변경**에 원하는 이름을 입력하세요.
    - **레이아웃**에서 위젯을 선택 후, 드래그하여 원하는 위치로 이동할 수 있습니다.
4. **저장** 버튼을 클릭세요. 선택한 템플릿 기준의 Flex 보드를 생성합니다.
    - **Flex 보드**의 보드 목록에서 생성한 보드를 확인할 수 있습니다.

Flex 보드 만들기

Flex 보드를 생성하고 위젯을 배치해 자신만의 대시보드를 만들 수 있습니다.

1. **Flex 보드**에서 오른쪽 위의 **대시보드 생성하기** 버튼을 클릭하세요.
2. **대시보드 생성하기** 창이 나타나면 입력 상자에 대시보드 **이름**을 입력하세요.
3. 위젯 배치 방식을 선택한 후, **대시보드 생성하기** 버튼을 클릭하세요.
    - **고정 레이아웃 플렉스보드**: 픽셀 기반으로 위젯 자유 배치
    - **반응형 플렉스보드**: 브라우저 크기 기준 그리드 반응형 레이아웃
      (그리드 설정한 뒤 **대시보드 생성하기** 버튼 클릭)
4. **위젯 템플릿** 선택 화면에서 대시보드에 배치할 위젯을 선택하세요.

- 메트릭스 위젯을 추가하려면 **위젯 템플릿**의 오른쪽 **모든 메트릭스** 버튼을 클릭하세요.

> ⓘ
> - **위젯 템플릿**: 일반적인 모니터링 상황에서 중요하게 다뤄지는 지표를 간추려 사전 정의된 위젯 목록을 선택할 수 있습니다.
> - **모든 메트릭스**: 사용자의 프로젝트에서 수집 중인 모든 메트릭스 데이터를 기준으로 위젯을 생성합니다. 메트릭스 위젯에 대한 자세한 설명은 다음 문서를 참고하세요.

5. 배치한 위젯의 위치를 이동하거나 크기를 조절하세요.
   - 위젯의 위쪽에 커서를 올려 십자 커서가 보이면, 위젯을 클릭한 채 드래그하여 위치를 이동합니다.
   - 위젯 오른쪽 아래에 커서를 올려 화살표 커서가 보이면, 위젯을 클릭한 채 드래그하여 크기를 조절합니다.
6. 추가할 위젯 배치를 완료하면 **뷰 모드**를 선택해 레이아웃을 확인하세요. 레이아웃을 다시 수정하려면 **수정 모드**를 선택해 위젯의 배치를 변경하세요.
7. 모든 과정을 완료했다면 오른쪽 위 (**목록으로 가기**) 아이콘을 클릭하세요.
   - **Flex 보드**의 **대시보드 목록**에서 생성한 보드를 확인할 수 있습니다. 생성한 보드를 선택해 새로 생성한 Flex 보드를 확인할 수 있습니다.

> ⓘ 고객의 의견을 반영해 대시보드 템플릿 또는 위젯을 추가하려고 합니다. 필요한 대시보드 템플릿 또는 위젯이 있다면 support@whatap.io로 문의해 주세요.

Flex 보드 편집하기

> ⓘ **Flex 보드 편집 권한**
> - **홈 화면** > **통합 Flex 보드**: 개인 계정 대시보드로 권한 영향은 없으나 읽기 전용으로 공유된 대시보드는 수정할 수 없습니다.
> - **홈 화면 > 프로젝트 선택** > **대시보드** > **Flex 보드**: 프로젝트 수정 권한 또는 프로젝트 플렉스보드 편집 권한, Site Admin 권한을 가진 사용자가 수정할 수 있습니다.
>
> 멤버 권한에 대한 자세한 내용은 다음 문서를 참고하세요.

1. **Flex 보드**의 **대시보드 목록**에서 수정 아이콘을 클릭하세요. 화면 오른쪽에 **Flex 보드 관리** 창이 나타납니다.
2. **Flex 보드 관리** 창에서 이름, 프로젝트, 레이아웃, 위젯.json, 옵션.json을 설정합니다.
   - **이름 변경**: 대시보드의 이름을 수정합니다.
   - **프로젝트**: 대시보드 위젯에 데이터를 가져올 프로젝트를 선택합니다.

     > **프로젝트** 옵션은 **홈 화면** > **통합 Flex 보드** 메뉴에서만 설정할 수 있습니다.
     > **통합 Flex 보드** 메뉴로 진입한 경우, 위젯마다 데이터를 가져올 프로젝트를 선택할 수 있습니다.
     >
     > 1. 위젯 오른쪽 위 + 버튼을 클릭하고 원하는 프로젝트를 검색해 하나 또는 다중 선택하세요.
     > 2. 프로젝트를 선택한 후 **적용** 버튼을 클릭하세요.

   - **레이아웃**
     - 위젯을 드래그하여 위치를 이동합니다.
     - 위젯 오른쪽 아래를 마우스로 클릭한 상태로 드래그해 위젯의 크기를 조절합니다.
     - 위젯을 삭제하려면 위젯 오른쪽 위에 아이콘을 클릭하세요.
   - **위젯.json**: 대시보드의 위젯 설정을 JSON 형식으로 불러올 수 있습니다. 아이콘을 클릭하면 JSON 내용을 복사합니다.
   - **옵션.json**: 대시보드에 설정된 옵션을 JSON 형식으로 불러올 수 있습니다. 아이콘을 클릭하면 JSON 내용을 복사합니다.

Flex 보드 삭제하기

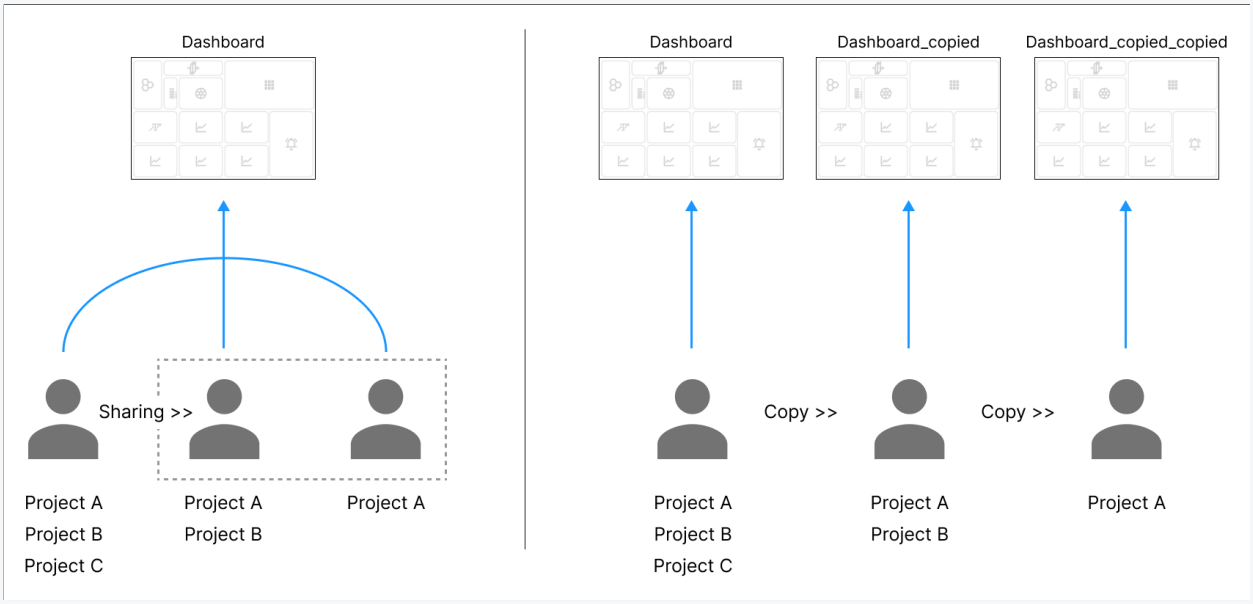
1. **Flex 보드** > **대시보드 목록**에서 삭제할 항목의 아이콘을 클릭하세요.
2. 확인 메시지에서 **삭제** 버튼을 클릭하세요.

> 대시보드 삭제는 소유자와 공유받은 사용자에 따라 다르게 적용됩니다.
> - 대시보드 소유자가 대시보드를 삭제하면, 공유된 모든 사용자 계정에서도 해당 대시보드를 삭제됩니다.
> - 대시보드를 공유받은 사용자가 대시보드를 삭제하면, 해당 사용자의 **대시보드 목록**에서만 삭제됩니다. 원본 대시보드는 유지됩니다.

Flex 보드 공유하기

Flex 보드 공유 기능은 **홈 화면 > 통합 Flex 보드** 메뉴에서만 사용할 수 있습니다.
프로젝트의 **Flex 보드** 메뉴에서는 제공되지 않습니다.

공유와 복사의 차이

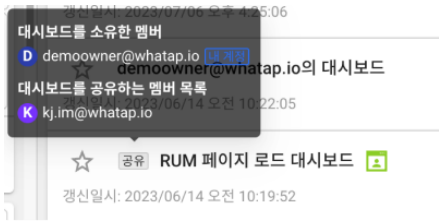


- 공유: 공유한 멤버는 모두 같은 대시보드를 볼 수 있습니다.
- 복사: 대시보드를 다른 멤버 또는 나에게 복사하면, 복사된 별도의 대시보드를 볼 수 있습니다.
- **읽기 전용**로 공유/복사받은 대시보드는 수정할 수 없습니다. 단, **수정 모드**로 공유/복사된 경우는 수정할 수 있습니다.
- **읽기 전용**로 공유받은 멤버도 해당 대시보드를 **읽기 전용으로 재공유**하거나 **복사**할 수 있습니다.
- 대시보드에 포함된 프로젝트 중 **최소 1개** 프로젝트에 초대된 멤버에게만 공유할 수 있습니다. 공유 또는 복사 받을 멤버의 프로젝트 소속 여부를 확인하세요.

통합 Flex 보드에서 생성한 Flex 보드를 다른 멤버와 공유할 수 있습니다. 소유자가 공유된 대시보드를 수정하면, 공유받은 모든

멤버에게 동일하게 반영됩니다.

1. **통합 Flex 보드**의 **대시보드 목록**에서 공유할 대시보드의 아이콘을 클릭하세요.
2. **대시보드 공유하기** 창이 나타나면 입력창에 공유할 멤버를 한 명 또는 여러명을 선택하거나 계정을 직접 입력하세요.
   - **읽기 전용**를 선택하면 공유받은 멤버는 대시보드를 수정할 수 없습니다.
   - 공유 대상은 대시보드에 포함된 프로젝트 중 최소 1개 프로젝트에 소속되어 있어야 합니다.
3. **공유** 버튼을 클릭하세요.
   - 공유된 항목은 **공유** 태그가 표시됩니다. **공유** 태그에 마우스를 올리면 공유한 멤버 정보를 확인할 수 있습니다.
   - 읽기 전용으로 공유한 항목은 **읽기 전용** 태그가 표시됩니다.

대시보드를 공유받은 경우	대시보드를 공유한 경우(대시보드 소유자)
	

대시보드 공유 조건

- 대시보드 공유 기능은 **홈 화면** > **통합 Flex 보드**에서만 제공합니다.
- 누구나 자신의 대시보드를 공유할 수 있으며, 공유받은 대시보드를 다시 공유하거나 복사할 수 있습니다.
- **읽기 전용**로 공유받은 대시보드는 수정할 수 없지만, **수정 모드**로 공유받은 대시보드는 수정할 수 있습니다.
- **소유자**가 대시보드를 삭제하면, 공유받은 모든 계정에서 해당 대시보드가 삭제됩니다.
- **공유받은 사용자**가 삭제하면, 해당 사용자의 **대시보드 목록**에서만 삭제되며 원본은 유지됩니다.
- 공유받은 대시보드의 프로젝트 중 **조회 분석** 권한이 없는 프로젝트는 데이터를 조회할 수 없습니다.

Flex 보드 복사하기

사용자가 Flex 보드를 복사해 다른 멤버에게 전달할 수 있습니다.

1. **통합 Flex 보드**의 **대시보드 목록**에서 아이콘을 클릭하세요.
2. 복사 옵션을 선택하세요.
    - **나에게 복사**: 자신의 계정으로 복사합니다.

      a. `_copied` 접미어가 붙은 항목이 **대시보드 목록**에 추가됩니다.

    - **다른 사람에게 복사**: 다른 멤버 계정으로 복사합니다.

      a. ＋ **계정**을 클릭하세요.
      b. 복사할 멤버의 이메일을 선택하고 **대시보드 복사하기** 버튼을 클릭하세요.
      c. 해당 멤버의 **대시보드 목록**에 `_copied` 접미어가 붙은 항목이 추가됩니다.

JSON 파일로 공유하기

Flex 보드 설정을 JSON 파일로 다른 멤버에게 전달하거나 다른 멤버의 설정을 가져올 수 있습니다.

내보내기

1. **통합 Flex 보드** 또는 프로젝트의 **Flex 보드** 메뉴로 이동하세요.
2. **대시보드 목록**에서 버튼을 클릭하세요.
3. 내보내기 옵션을 선택하세요.
    - **프로젝트 정보 포함**: 대시보드에 설정된 프로젝트 정보를 JSON 파일에 포함해 공유
    - **프로젝트 정보 미포함**: 대시보드에 설정된 프로젝트 정보를 JSON 파일에 포함하지 않고 공유
4. JSON 파일을 다운로드하고 공유할 멤버에게 전달하세요.

가져오기

1. **통합 Flex 보드** 또는 프로젝트의 **Flex 보드** 메뉴로 이동하세요.

2. 화면 오른쪽 위 **가져오기** 버튼을 클릭하세요.
3. 다운로드한 JSON 파일을 선택하세요.

> ⓘ 이 기능은 프로젝트의 **수정** 권한을 소유한 멤버만 이용할 수 있습니다.

Flex 보드 화면 모드

Flex 보드의 화면 모드에 따라 제공하는 기능을 설명합니다.

> ⓘ **접근 가능한 멤버 권한**
>
> 프로젝트 내 **Flex 보드** 메뉴에서 **수정 권한**이 있는 사용자만 **수정 모드**, **관리자**, **필터** 기능에 접근할 수 있습니다. 멤버 권한에 대한 자세한 내용은 다음 문서를 참조하세요.
>
> - **프로젝트 수정** 권한
> - **프로젝트 플렉스보드 편집** 권한
> - **Site Admin** 권한

모드 종류

Flex 보드는 화면 모드에 따라 설정할 수 있는 기능이 다릅니다.

- **수정 모드**: 위젯 템플릿 목록에서 Flex 보드의 레이아웃에 위젯을 배치하거나 위젯을 수정, 삭제, 위치 조정할 수 있습니다.
- **관리자**: 위젯의 세부 속성을 json 형태로 관리할 수 있습니다. 레이아웃에 배치한 위젯의 오른쪽 위에 위치한 🔧 버튼을 선택하세요.
- **뷰 모드**: 위젯 배치, 크기 조절, 세부 옵션 등의 설정을 완료한 다음 대시보드와 같이 각 위젯의 데이터를 실시간으로 모니터링할 수 있습니다.

공통 기능

- 데이터 필터링

- 실시간 데이터 조회 범위 설정
- 인쇄 모드
- Flex 보드 관리
- 데이터 병합 옵션

데이터 필터링

데이터 필터링을 통해 모니터링 대상을 간추려 효율적인 Flex 보드를 구성할 수 있습니다.

1. 화면 위 ▽ **필터** 탭을 클릭하세요.
2. 화면 오른쪽에 나타나는 **필터** 창에서 필터링 대상을 선택하세요. 필터링 대상을 모두 선택하려면 **전체 선택**을 선택하세요.
   - **에이전트** 옵션: 에이전트 종류 또는 애플리케이션 목록 선택, **검색** 기능 제공
   - **메트릭스** 옵션: 프로젝트 또는 에이전트 유형 선택, **+ And** 조건 추가하거나 **포함**, **제외**로 문자열 기반 자동 선택 가능
3. 필터링 대상을 선택했다면 **적용** 버튼을 클릭하세요.
4. 자주 사용하는 조건은 **필터 저장** 버튼으로 저장하세요.

   a. 필터 이름을 입력하고 **저장**을 선택하세요. **저장된 필터**는 **필터** 창 하단에 나타납니다.

필터링 조건

필터링 조건은 에이전트 타입과 메트릭스 타입으로 구분합니다.
에이전트는 기본 모니터링 대상만 선택 옵션으로 제공하며, 메트릭스는 프로젝트에서 수집한 모든 메트릭스 데이터의 태그를 선택 옵션으로 제공합니다. 메트릭스 타입은 에이전트가 아닌 형태로 데이터를 수집한 후 필터링을 적용할 때 유용합니다. 메트릭스에 대한 자세한 내용은 다음 문서를 참조하세요.

타입	에이전트	메트릭스
선택 옵션	와탭 프로젝트 모니터링 대상 (애플리케이션, 서버, 데이터베이스, 컨테이너, 쿠버네티스 POD와 NODE)	에이전트를 포함한 모든 메트릭스 태그
적용 범위	모든 위젯	메트릭스 데이터를 사용하는 모든 위젯

타입	에이전트	메트릭스
필터링 조건	체크박스	**일치**, **포함**, **제외** 방식

조회 시간 설정하기

Flex 보드의 화면 오른쪽 위 시간 선택자를 이용해 위젯에 데이터 조회 시간을 설정할 수 있습니다.

- ||: 실시간 모드 멈추고 특정 조회 시간으로 설정
- ▶: 다시 실시간 모드로 전환

인쇄하기

Flex 보드 화면을 PDF로 저장해 인쇄할 수 있습니다.

1. Flex 보드 화면의 오른쪽 위 🖨 아이콘을 클릭해 인쇄 모드로 전환합니다.
2. 옵션을 설정합니다.
    - **필터 옵션 보기**: 화면에 필터 옵션 켜거나 끄기
    - **{mode} 방향으로 변환 / {mode} 방향으로 변환**: 화면을 가로 또는 세로로 변환
3. 완료 후 **PDF다운로드**를 선택하세요.
4. 인쇄 모드를 종료하려면 오른쪽 위 ✕ **인쇄모드 종료**를 클릭하세요.

Flex 보드 관리

Flex 보드의 레이아웃을 변경하거나 위젯의 위치 변경, 크기 조절, 이름 변경 등의 관리 기능을 제공합니다. **Flex 보드 관리** 기능은 **대시보드** > **Flex 보드** 메뉴의 **대시보드 목록**에서 ✎ 아이콘을 클릭해 진입할 수도 있습니다.

1. Flex 보드 화면의 오른쪽 위 ⚙ 아이콘을 클릭하면 **Flex 보드 관리** 창이 열립니다.
    - **이름 변경**: Flext 보드의 이름 변경
    - **레이아웃**: 위젯의 위치 및 크기조절 및 삭제
    - **다른 사용자의 수정을 제한**: 수정 제한 옵션
        - 수정 권한이 있는 멤버는 수정, 삭제, 편집할 수 있으며 옵션을 활성화하면 **다른 사용자의 수정을 제한: On**으로

변경됨

- **위젯.json**: Flex 보드에 포함된 위젯을 JSON 형태로 편집 및 관리
- **옵션.json**: Flex 보드의 설정을 JSON 형태로 편집 및 관리(프로젝트 정보, Flex 보드 필터, 시간 선택 등 설정값 편집)

> **위젯.json**, **옵션.json** 기능은 대시보드 관리자 권한의 사용자에게만 제공합니다.

2. 모든 설정을 완료하고 **저장** 버튼을 클릭하세요.

대시보드 즐겨찾기로 등록

사용자가 생성한 Flex 보드는 즐겨찾기에 등록해 빠르게 접근할 수 있습니다. **대시보드** > **Flex 보드**의 **대시보드 목록**에서 즐겨찾기로 등록할 대시보드의 ☆ 버튼을 클릭하면 ★으로 변경됩니다.

위젯 관리하기

Flex 보드에 배치한 위젯 관리를 통해 대시보드를 사용자가 원하는 대로 꾸밀 수 있습니다.

> 대시보드를 메트릭스 위젯을 포함해 다양한 위젯 템플릿을 활용해 구성할 수 있습니다.
> 각 템플릿의 특징과 사용법은 위젯 템플릿 문서에서 자세히 확인할 수 있습니다.

위젯 추가

이미 만들어진 Flex 보드에 위젯을 추가할 수 있습니다.

1. **Flex 보드** 메뉴의 **대시보드 목록**에서 위젯을 추가할 대시보드를 선택하세요.
2. 화면 왼쪽 위에 **수정 모드**를 선택하세요.
3. 왼쪽에 **위젯 템플릿** 목록에서 원하는 위젯을 선택하세요. 메트릭스 위젯을 추가하려면 **모든 메트릭스**를 선택한 다음 원하는 위젯을 클릭하세요.
    - 홈 화면의 **통합 Flex 보드**로 진입한 경우 위젯을 추가했다면 **프로젝트 선택** 버튼을 클릭해 수집 대상 프로젝트를 선택하세요.
    - 추가할 위젯이 있다면 같은 과정을 반복하세요.
4. 모든 과정을 완료했다면 화면 위에 **뷰 모드**를 선택해 레이아웃을 확인한 후, (**목록으로 가기**) 버튼을 클릭하세요.

위젯 속성 변경

Flex 보드 메뉴의 **대시보드 목록**에서 위젯 속성을 변경할 보드를 선택하세요. **수정 모드**로 진입합니다.

위젯 이름 변경

1. 이름을 변경할 위젯의 이름 오른쪽에 아이콘을 클릭하세요.
2. 위젯의 **제목**, **폰트 크기**, **색상**을 설정한 후, **저장** 버튼을 클릭하세요.

- 저장 시 **전체 위젯에 적용** 토글 버튼의 기능을 활성화하면 폰트 크기와 색상을 다른 위젯에도 적용합니다.
- 위젯의 제목을 잠금 설정하려면 버튼을 선택한 후 **저장** 버튼을 클릭하세요.

시간 설정

1. 시간을 설정하려는 위젯의 오른쪽 위에 버튼을 클릭하세요.
2. **시간 선택** 창에서 시간을 설정하세요.
   - 대시보드 화면에 설정한 시간을 적용하려면 **대시보드 설정을 따름**을 체크하세요. 대시보드의 시간 설정은 오른쪽 상단에 위치합니다.
   - 사용자가 원하는 시간 설정을 적용하려면 **대시보드 설정을 따름**을 체크 해제하세요. 또는 버튼을 이용해 원하는 시간을 설정하세요.
3. 시간 설정을 완료했다면 버튼을 클릭하세요.

> 위젯의 데이터 유형에 따라서 시간 설정 기능을 지원하지 않을 수 있습니다.

출력 데이터 옵션 설정

차트형 위젯의 출력 데이터 옵션을 설정해 사용자가 원하는 데이터를 선택할 수 있습니다.

1. 위젯의 오른쪽 위에 아이콘을 클릭해, 출력 데이터 옵션을 설정하세요.
   - 프로젝트, 에이전트, 애플리케이션 등의 종류를 선택해 데이터를 필터링할 수 있습니다.
   - 개별, 병합 데이터를 선택해 차트에 표시되는 그래프의 유형을 변경할 수 있습니다.
   - > 을 선택하세요. **차트 설정** 창을 통해 세부 옵션을 변경할 수 있습니다. 설정할 수 있는 옵션은 다음과 같습니다.
     - **차트 유형**: 집계하는 현재 수치를 차트에 표시할 수 있는 옵션입니다.
     - **y축 차트 최대값**: y축에 차트 최댓값을 **자동** 또는 **고정**으로 선택할 수 있습니다.
     - **수평선 설정**: **추가** 버튼을 선택해 임계치 또는 범위를 수평선으로 표시할 수 있습니다. 여러 개를 추가해 적용할 수 있습니다. 수평선을 추가한 다음에는 **적용** 버튼을 선택해야 차트에 반영됩니다.
     - **차트 최대값 표시** 차트에 표시되는 그래프의 최댓값을 표시합니다.

- **보조 차트**: 위젯의 주요 차트에 보조 차트를 추가 설정할 수 있습니다. 보조 차트의 종류를 선택한 다음 레이아웃, 크기, 데이터 병합 여부를 선택하세요. 다른 위젯에도 공통 적용하려면 **전체 위젯에 적용** 버튼을 선택하세요. 보조 차트는 데이터를 다양한 방법으로 보고 싶을 때 유용합니다.

2. 옵션 설정을 모두 완료했다면 **차트 설정** 창의 왼쪽 위에 ✕ 버튼을 클릭하거나 **차트 설정** 창 영역 밖을 클릭하세요.

> ⓘ 위젯의 데이터 유형에 따라서 선택할 수 있는 옵션은 다를 수 있습니다.

메트릭스 위젯 데이터 변경

메트릭스 위젯의 데이터 조회 조건을 변경할 수 있습니다.

1. **Flex 보드** > **대시보드 목록**에서 위젯 데이터를 변경할 대시보드를 선택하세요.
2. **수정 모드**에서 변경할 위젯의 오른쪽 위에 ⋮ 아이콘을 클릭하세요.
3. </> 아이콘을 클릭하면, 데이터 설정 편집 창이 나타납니다.
4. 다음을 참조해 설정을 변경한 다음 < **위젯에 적용** 버튼을 클릭해 적용하세요.
   - 위젯 생성 시 기본 생성되는 메트릭스 옵션이 탭에 표시됩니다.
   - 메트릭스 옵션에는 **카테고리**와 **필드**, **태그**가 있습니다.
   - **카테고리** 항목은 현재 변경 기능을 제한합니다.
   - **필드**, **태그** 항목은 복수 선택할 수 있습니다. 선택한 태그 옵션으로 차트 데이터를 그룹화합니다.
   - 🔍 버튼을 선택해 원본데이터를 조회할 수 있습니다.
   - **오브젝트 병합** 및 **시간 병합** 항목에 대한 자세한 내용은 데이터 병합 옵션을 참조하세요.

데이터 병합 옵션

Flex 보드의 **모든 메트릭스** 목록에서 배치한 메트릭스 위젯은 데이터 병합 옵션을 제공합니다. 데이터 병합은 **오브젝트 병합**과 **시간 병합** 방법을 제공합니다.

- **오브젝트 병합**

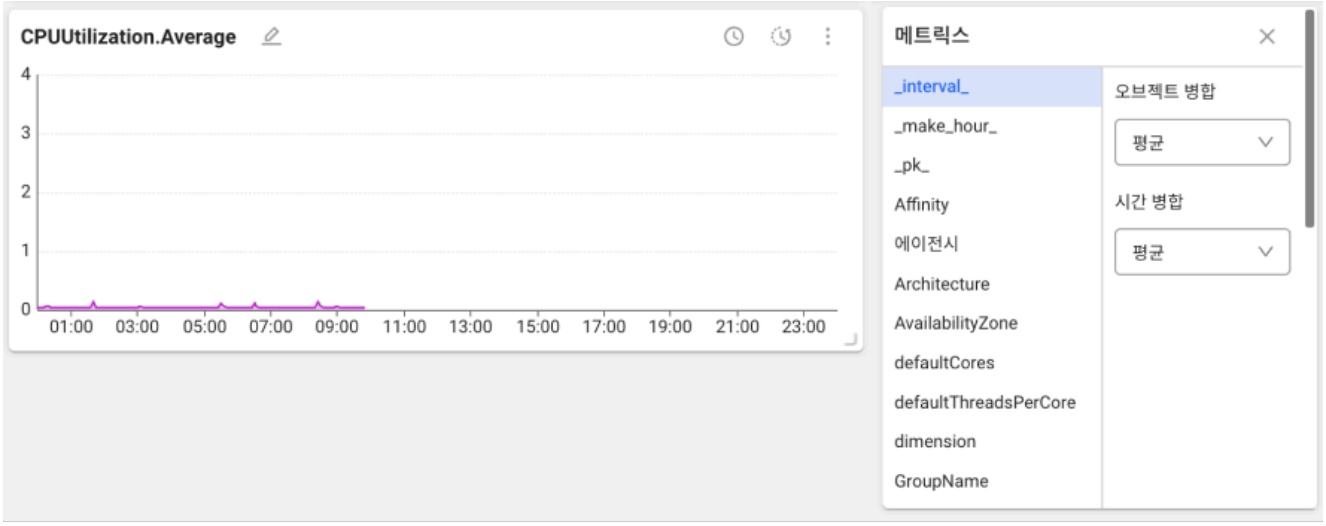
  좌측에서 선택한 태그(예: 에이전트 명)를 기준으로 차트 데이터를 그룹화할 때 사용합니다. 이때 서로 다른 필드(예: CPU)값을 가진 데이터들 중에서 태그가 일치하는 경우 해당 데이터를 병합하는 방법입니다.

- **시간 병합**

  조회 시간이 긴 경우 주로 사용합니다. 원본 데이터에서 필드 값이 같은 데이터끼리 시간 범위에 따라 일정한 시간 간격(5분 또는 1시간)으로 데이터를 병합합니다. 이 병합 방법을 시간 병합이라고 합니다.

- **AWS CloudWatch 메트릭스**

  AWS 모니터링 지표의 경우 통계 방식이 필드명 뒤에 suffix(접미사) 형식으로 추가됩니다. 데이터 병합 기본값은 이 suffix를 참조하고 있습니다.



1. **수정 모드**에서 **모든 메트릭스** 버튼을 클릭해 **위젯 템플릿** 목록을 **모든 메트릭스** 목록으로 변경하세요.
2. 메트릭스 위젯을 레이아웃에 배치하세요.
3. 메트릭스 위젯 오른쪽 위에 ⋮ 버튼을 선택하세요.
4. 태그(예: 에이전트 명) 옵션을 선택하면 데이터 병합 옵션이 나타납니다. 위젯의 크기에 따라 메트릭스 옵션의 위치가 다를 수 있습니다.

위젯 복사

1. 위젯의 오른쪽 위에 ⋮ 버튼을 클릭하세요.
2. 버튼을 클릭하세요.
3. **복사** 창이 나타나면 ＋ **위젯 추가** 버튼을 클릭하세요.
4. 목록에 위젯의 이름을 변경하세요.

5. **적용** 버튼을 클릭하세요.
6. **수정 모드** 화면에서 복사한 위젯을 확인할 수 있습니다.

위젯 삭제

1. 위젯의 오른쪽 위에 ⋮ 아이콘을 클릭하세요.
2. ✕ 버튼을 클릭하세요.
3. 확인 메시지 창이 나타나면 **삭제** 버튼을 클릭하세요. 삭제를 취소하려면 **취소** 버튼을 선택하세요.

템플릿 목록

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

템플릿 목록

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

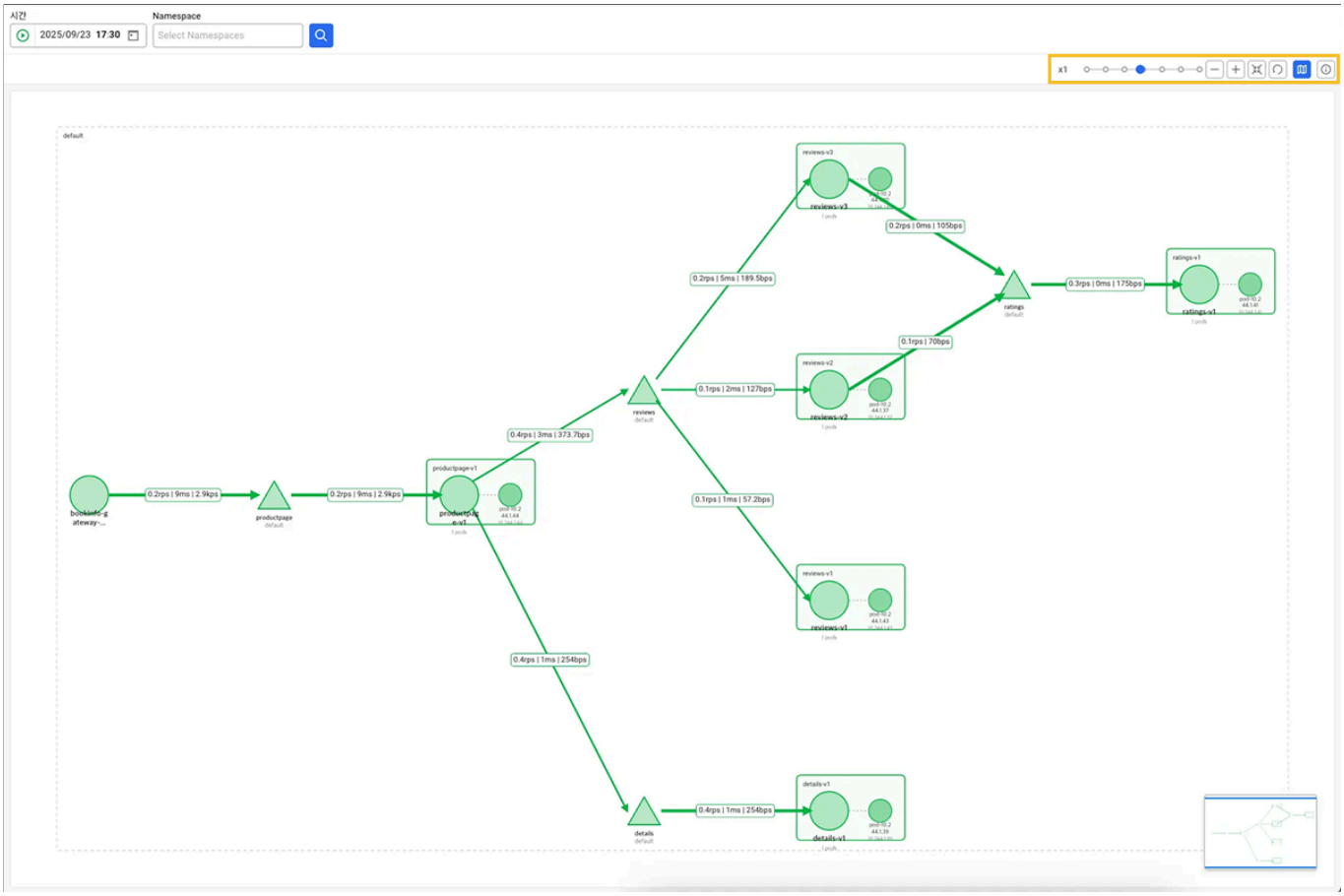
- → **템플릿**: 클릭하면 해당 템플릿 화면으로 바로 이동합니다.
- ⓘ **수집 안내**: 클릭하면 데이터 수집 설정 방법을 제공합니다.

Nginx 템플릿

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

Istio 템플릿

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



범례

화면 오른쪽 상단의 ⓘ 버튼을 클릭하면 범례를 확인할 수 있습니다.

노드 모양

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

노드 상태

HTTP 요청/응답 전체 중 에러(4xx, 5xx)의 비율

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

Pod 노드 상태

쿠버네티스 Pod의 Phase

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

엣지 색상

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

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

OpenMetrics 탐색기

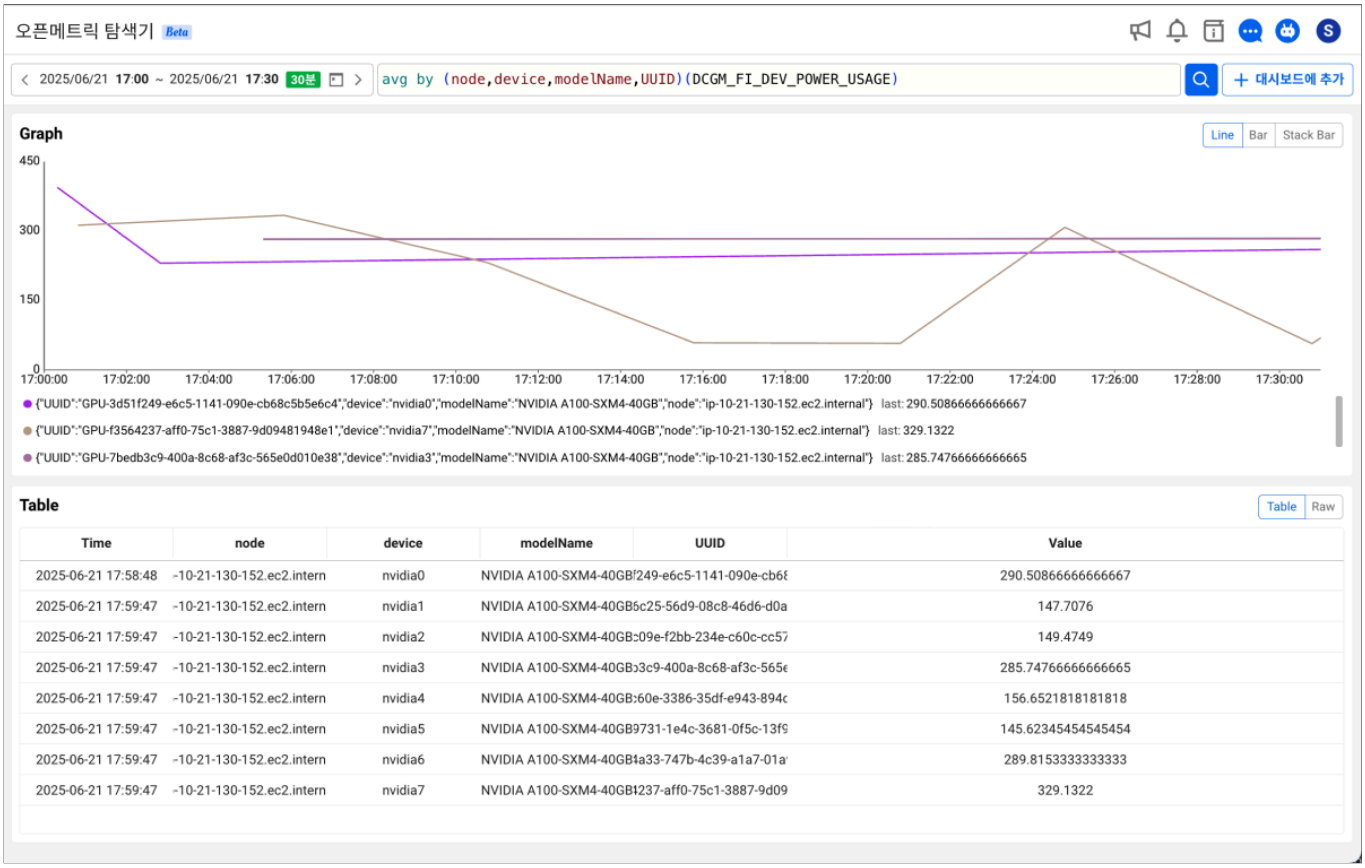
OpenMetrics 탐색기는 Prometheus 기반의 커스텀 메트릭을 PromQL을 통해 직접 탐색하고, 다양한 형태로 시각화하여 메트릭의 상태를 확인하거나 인사이트를 얻을 수 있는 기능입니다.

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

> ⓘ **OpenMetrics 탐색기**를 사용하려면 오픈 에이전트 설치가 필요합니다.
>
> - 와탭 오퍼레이터 설치 문서를 참고하세요.

PromQL 탐색하기

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



Graph

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

Table / Raw

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

PromQL 문법 지원 및 자동완성

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

연산자 목록을 제시합니다.

`{pod="example"}` 형태의 라벨 필터링도 지원합니다.

대시보드에 추가하기

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

추가 방법

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

클러스터 요약

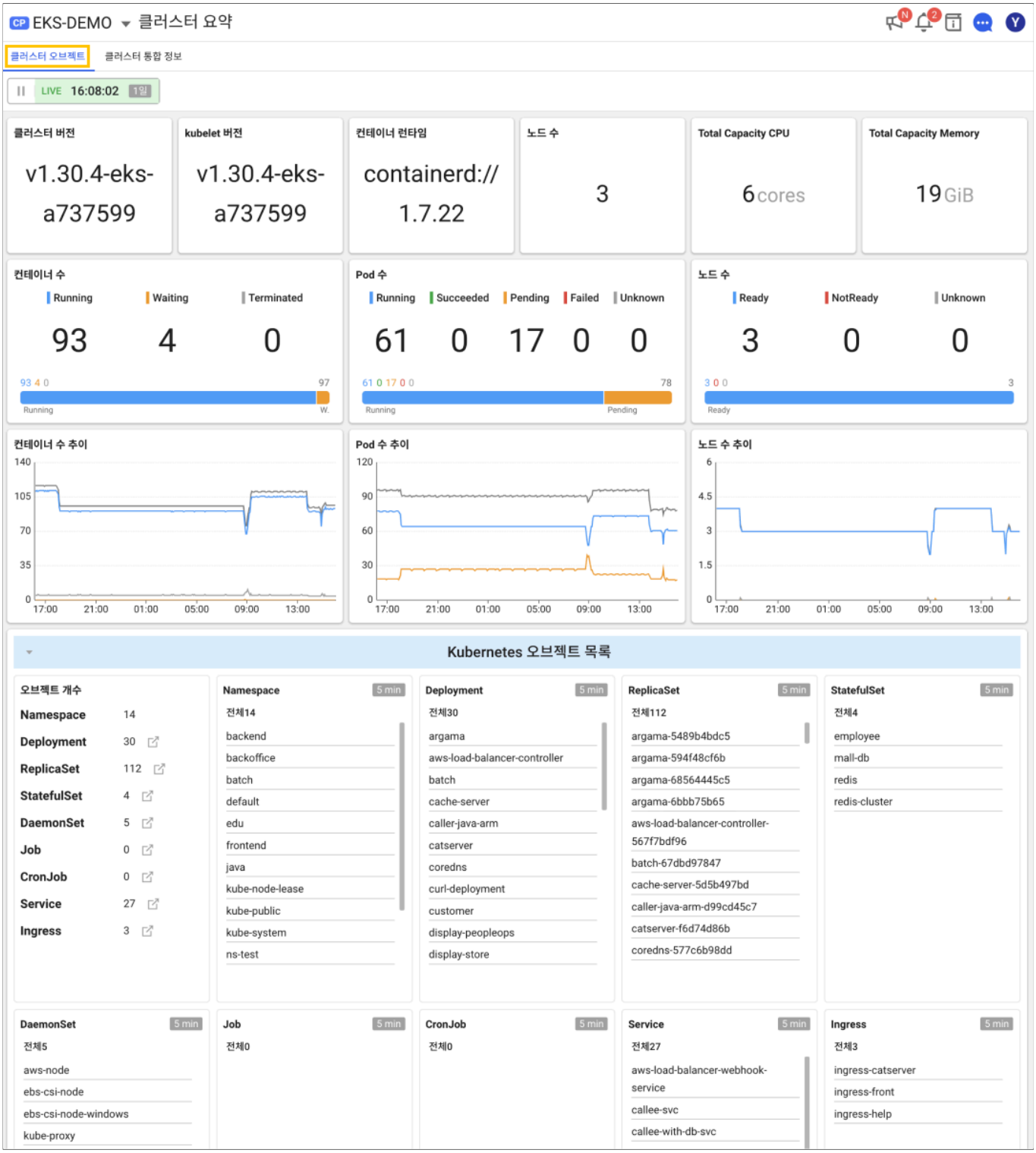
> **요구 사항**
>
> 쿠버네티스 클러스터 프로젝트(CP)인 경우에만 **클러스터 요약** 메뉴를 확인할 수 있습니다.

클러스터 관련 요약 정보를 확인할 수 있습니다. 네임스페이스 프로젝트로 분리된 정보를 통합해 확인할 수 있습니다.

클러스터 오브젝트

클러스터와 관련된 버전 정보와 클러스터 내부의 노드, Pod, 컨테이너 등 쿠버네티스 오브젝트의 개수를 확인할 수 있습니다.

> - 쿠버네티스 에이전트 1.4.1 버전 이상일 때 **클러스터 버전**을 확인할 수 있습니다.
> - 쿠버네티스 에이전트 1.3.6 버전 이상일 때 다음 사항을 확인할 수 있습니다.
>   - **kubelet 버전**
>   - **컨테이너 런타임**


EKS-DEMO
클러스터 요약
클러스터 오브젝트
클러스터 통합 정보
LIVE 16:08:02 1일
클러스터 버전
v1.30.4-eks-a737599
kubelet 버전
v1.30.4-eks-a737599
컨테이너 런타임
containerd://1.7.22
노드 수
3
Total Capacity CPU
6 cores
Total Capacity Memory
19 GiB
컨테이너 수
Running 93
Waiting 4
Terminated 0
Pod 수
Running 61
Succeeded 0
Pending 17
Failed 0
Unknown 0
노드 수
Ready 3
NotReady 0
Unknown 0
컨테이너 수 추이
Pod 수 추이
노드 수 추이
Kubernetes 오브젝트 목록
오브젝트 개수
Namespace 14
Deployment 30
ReplicaSet 112
StatefulSet 4
DaemonSet 5
Job 0
CronJob 0
Service 27
Ingress 3
Namespace 5 min
전체14
backend
backoffice
batch
default
edu
frontend
java
kube-node-lease
kube-public
kube-system
ns-test
Deployment 5 min
전체30
argama
aws-load-balancer-controller
batch
cache-server
caller-java-arm
catserver
coredns
curl-deployment
customer
display-peopleops
display-store
ReplicaSet 5 min
전체112
argama-5489b4bdc5
argama-594f48cf6b
argama-68564445c5
argama-6bbb75b65
aws-load-balancer-controller-567f7bdf96
batch-67dbd97847
cache-server-5d5b497bd
caller-java-arm-d99cd45c7
catserver-f6d74d86b
coredns-577c6b98dd
StatefulSet 5 min
전체4
employee
mall-db
redis
redis-cluster
DaemonSet 5 min
전체5
aws-node
ebs-csi-node
ebs-csi-node-windows
kube-proxy
Job 5 min
전체0
CronJob 5 min
전체0
Service 5 min
전체27
aws-load-balancer-webhook-service
callee-svc
callee-with-db-svc
Ingress 5 min
전체3
ingress-catserver
ingress-front
ingress-help

클러스터 버전

- 5초 주기의 데이터입니다. 실시간 데이터 조회 시 5초에 한 번씩 데이터를 호출합니다.

kubelet 버전

- 5초 주기의 데이터입니다. 실시간 데이터 조회 시 5초에 한 번씩 데이터를 호출합니다.

컨테이너 런타임

- 5초 주기의 데이터입니다. 실시간 데이터 조회 시 5초에 한 번씩 데이터를 호출합니다.

컨테이너 수

- 상태별 컨테이너의 개수를 확인할 수 있습니다.
- 5초 주기의 데이터입니다. 실시간 데이터 조회 시 5초에 한 번씩 데이터를 호출합니다.

Pod 수

- 상태별 Pod의 개수를 확인할 수 있습니다.
- 5초 주기의 데이터입니다. 실시간 데이터 조회 시 5초에 한 번씩 데이터를 호출합니다.

노드 수

- 상태별 노드의 개수를 확인할 수 있습니다.
- 5초 주기의 데이터입니다. 실시간 데이터 조회 시 5초에 한 번씩 데이터를 호출합니다.

Kubernetes 오브젝트 목록

- 쿠버네티스 오브젝트 목록을 확인할 수 있습니다.
- 5분 주기의 데이터입니다. 실시간 데이터 조회 시 3분에 한 번씩 데이터를 호출합니다.

오브젝트 개수 추이

- 쿠버네티스 오브젝트의 개수가 24시간 동안 어떻게 변했는지 확인할 수 있습니다.

- 5분 주기의 데이터입니다. 실시간 데이터 조회 시 3분에 한 번씩 데이터를 호출합니다.

클러스터 통합 정보

클러스터 프로젝트 정보와 네임스페이스 프로젝트로 분리된 Pod, 컨테이너, 로그, 이벤트 수, 이벤트 목록을 조회할 수 있습니다.

> ⓘ 데이터를 조회하기 위해서는 해당 네임스페이스 프로젝트 **읽기** 권한이 필요합니다.

Pod/컨테이너 수

컨테이너 개수와 Pod 개수를 프로젝트 단위로 확인할 수 있습니다.

- **컨테이너 수**
  - **전체**: 클러스터에 속한 전체 컨테이너의 개수를 표시합니다.
  - **State 별**: 상태별로 진행 중인 컨테이너의 개수를 표시합니다.
- **Pod 수**
  - **전체**: 클러스터에 속한 전체 Pod의 개수를 표시합니다.
  - **Phase 별**: 상태별로 진행 중인 Pod의 개수를 표시합니다.

로그/이벤트 수

로그, Kubernetes 이벤트, WhaTap 알람의 개수를 프로젝트 단위로 확인할 수 있습니다.

- **로그 수**
  - **최근 1시간**: 최근 1시간 동안 쌓인 로그 라인의 개수를 표시합니다.
  - **최근 24시간**: 하루 동안 쌓인 로그 라인의 개수를 표시합니다.
- **Kubernetes 이벤트 수**
  - **최근 1시간**: 최근 1시간 동안 쌓인 이벤트 라인의 개수를 표시합니다.
  - **최근 24시간**: 하루 동안 쌓인 이벤트 라인의 개수를 표시합니다.
- **WhaTap 알람 수**
  - **최근 1시간**: 최근 1시간 동안 발송된 알람 개수를 표시합니다.
  - **최근 24시간**: 하루 동안 발송된 알람 개수를 표시합니다.

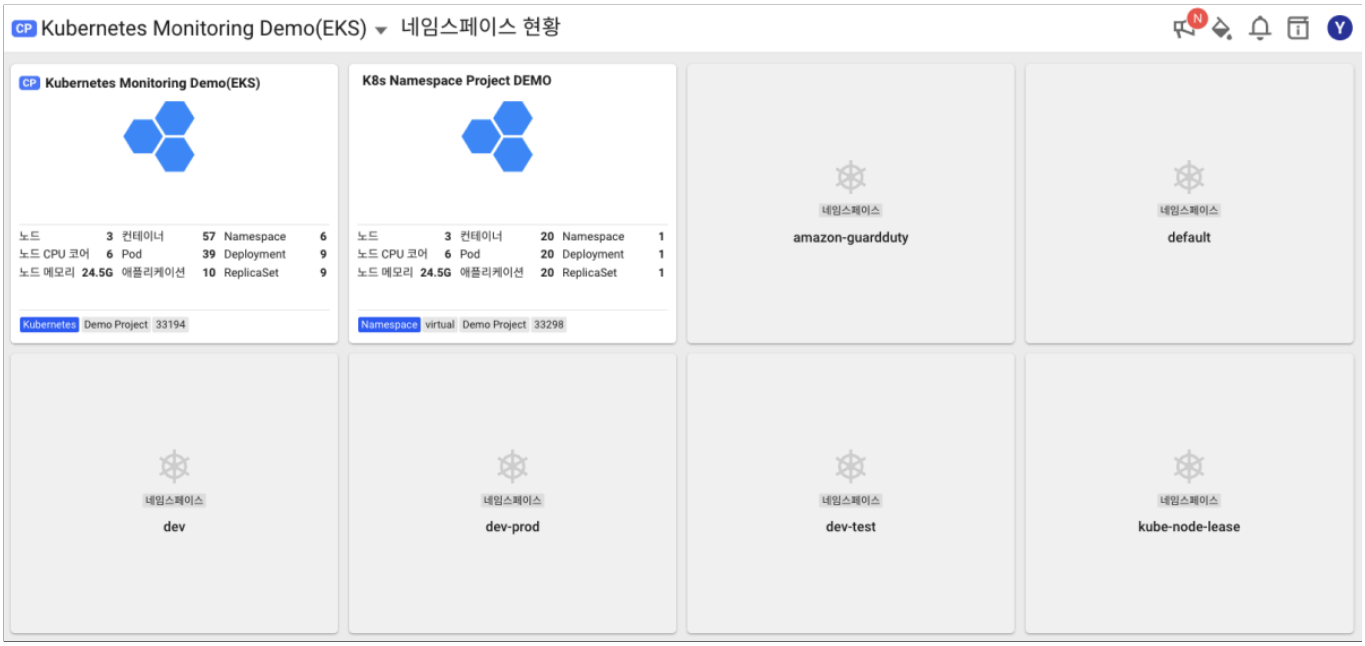
Kubernetes 이벤트 5

클러스터 프로젝트와 네임스페이스 프로젝트에서 발생한 최근 5개 쿠버네티스 이벤트 목록을 확인할 수 있습니다.

네임스페이스 현황

> ⓘ 쿠버네티스 클러스터 프로젝트(CP)일 경우 해당 메뉴를 조회할 수 있습니다.

네임스페이스 현황 메뉴를 통해 와탭 쿠버네티스 클러스터 프로젝트 및 하위 네임스페이스 프로젝트 현황을 확인할 수 있습니다. 네임스페이스 프로젝트에서는 접근할 수 없는 메뉴입니다.



네임스페이스 현황 메뉴는 프로젝트 목록 화면과 유사하지만 다음과 같은 차이가 있습니다.

- **프로젝트 목록**

  네임스페이스 프로젝트 생성 후 해당 프로젝트를 다른 그룹으로 이동할 수 있지만 네임스페이스를 한 눈에 파악하기 어렵습니다.

- **네임스페이스 현황**
  - 클러스터 프로젝트와 동일한 그룹에 속하지 않는 네임스페이스도 확인할 수 있습니다.
  - 한 눈에 관련 프로젝트를 모두 확인할 수 있습니다.

클러스터 프로젝트의 경우 Kubernetes 플랫폼을 나타내는 파란색 Kubernetes 태그가 표시됩니다. 네임스페이스

프로젝트에는 Namespace 태그가 표시됩니다. 위젯 왼쪽 위에 타이틀은 프로젝트 이름이며, 네임스페이스 이름은 가장 아래에 Namespace 태그 옆에 표시됩니다.

노드 목록

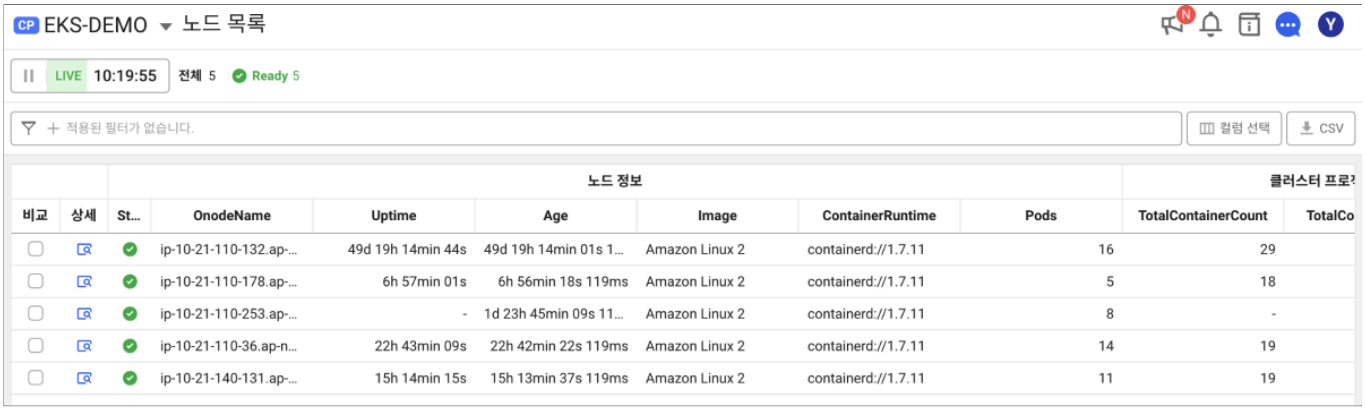
> 특정 컬럼의 데이터가 표시되지 않는 것은 쿠버네티스 노드 에이전트가 구 버전이기 때문일 수 있습니다. 이런 경우 충분한 데이터가 수집될 수 있도록 쿠버네티스 노드 에이전트 버전을 1.3.7 이상으로 업데이트하세요.

현재 프로젝트의 노드의 정보를 목록 형태로 확인할 수 있습니다.

- 10초 주기로 실시간 데이터를 업데이트합니다.
- 노드 상태 요약, 비교, 필터 및 CSV 파일 다운로드 기능을 제공합니다.

  CSV 파일명, kubernetes_node_list_{pcode}_{timestamp}.csv

기본 화면 안내

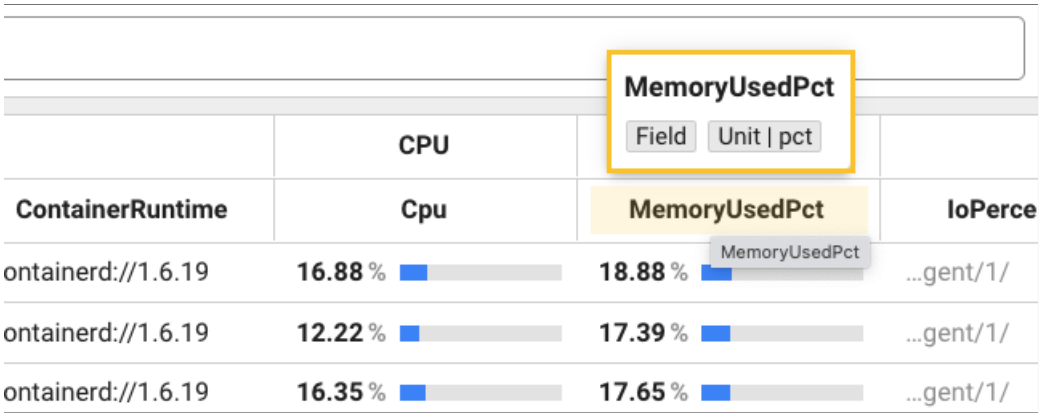


노드 목록 상단에서 원하는 옵션을 선택하여 데이터를 조회할 수 있습니다.

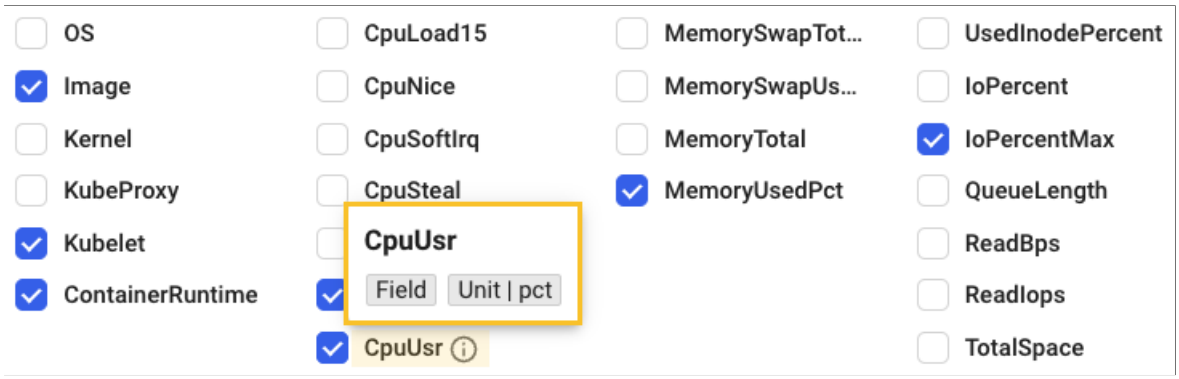
- 전체 노드 개수 및 상태별 노드 개수를 확인할 수 있습니다.
- 원하는 컬럼 선택 시 해당 컬럼 기준으로 **노드 목록**을 정렬해 조회할 수 있습니다.
- **노드 목록** 칼럼 좌측 확장 버튼을 클릭하면 디스크 정보를 확인할 수 있습니다.

메트릭 정보

- 원하는 컬럼 위에 마우스 커서를 위치시키면 다음과 같이 해당 컬럼이 제공하는 메트릭 정보를 확인할 수 있습니다.



- **컬럼 선택** 옵션 화면 내 컬럼 목록에서 컬럼별 메트릭 정보를 확인할 수 있습니다. 원하는 컬럼 위에 마우스 커서를 위치시키면 컬럼 이름 우측에 ⓘ **인포 아이콘**이 나타납니다. ⓘ **인포 아이콘** 위에 마우스 커서를 위치시키면 다음과 같이 메트릭 정보를 확인할 수 있습니다.



필터

필터 옵션 활용 시 원하는 노드 목록을 선택해 조회할 수 있습니다. `and` 조건을 통해 여러 개의 필터를 적용할 수 있습니다.

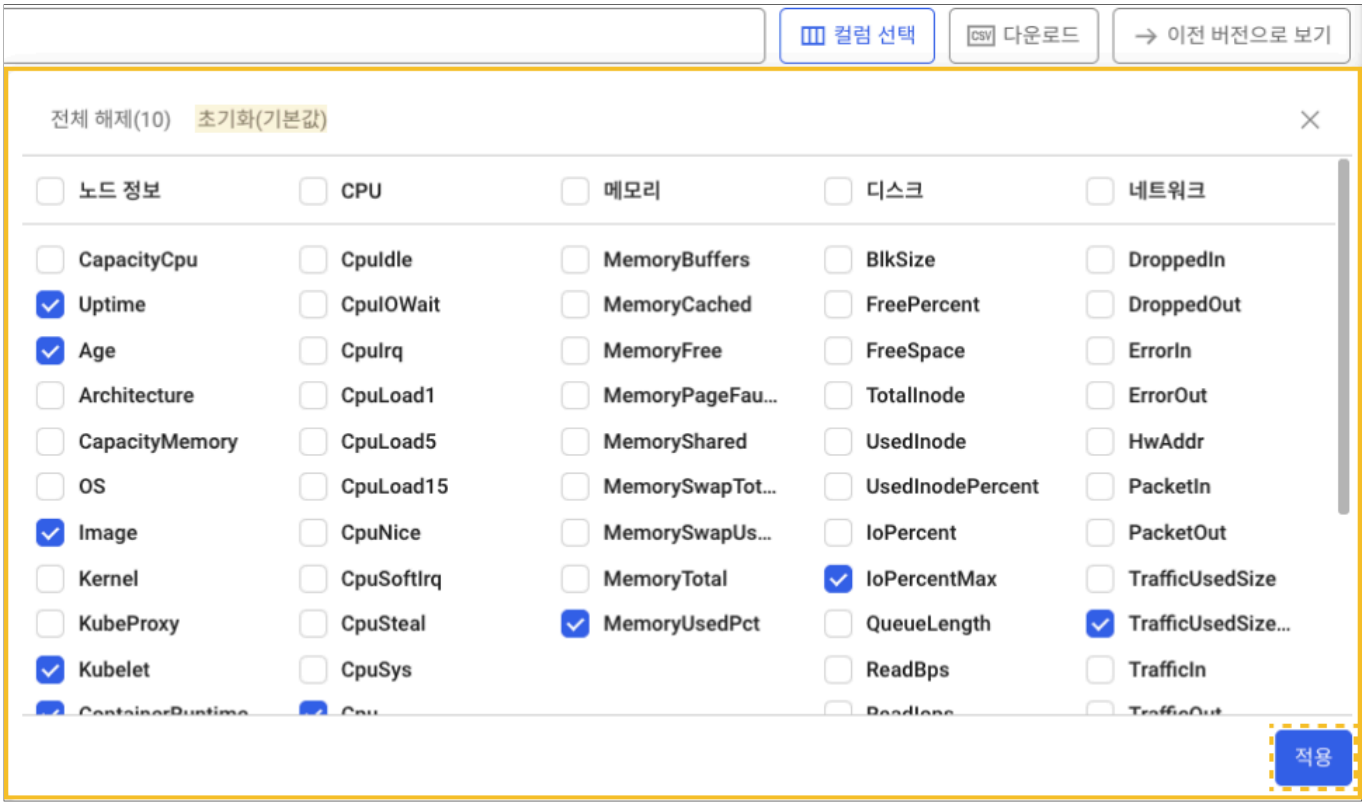
1. **필터** 입력 창에서 + **더하기 아이콘** 선택 시 다음과 같이 **필터 추가하기** 창이 나타납니다.

- **필터 키**와 **조건**을 선택하세요.
- 조건에 맞는 값을 선택하세요.

2. 필터 설정 후 **적용** 버튼을 선택해 해당 필터를 적용하세요.
3. 여러 개의 필터를 추가하려면 **필터** 입력 창 또는 **필터 추가하기** 창에서 ＋ **더하기 아이콘**을 선택하세요.

컬럼 선택

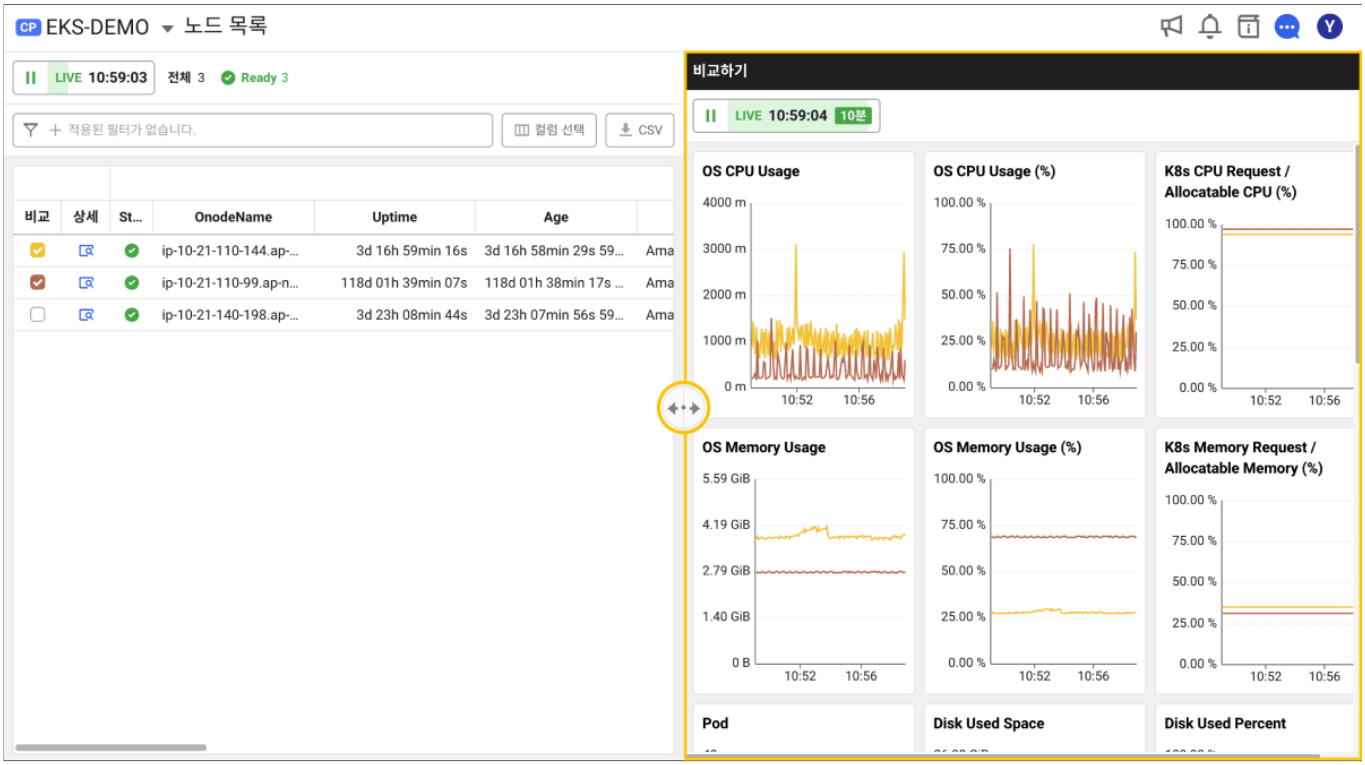
컬럼 선택 옵션 활용 시 원하는 노드 정보를 선택해 목록을 조회할 수 있습니다.



- 선택한 컬럼 목록은 메뉴 재진입 시 유지됩니다.
- **초기화** 버튼 선택 시 사전 정의된 기본 주요 지표 컬럼이 선택됩니다. 이 경우 기존 컬럼 선택 내역은 삭제됩니다.
- 컬럼 선택 후 우측 하단 **적용** 버튼을 클릭 해 선택 내역을 적용할 수 있습니다.

비교하기

비교 옵션을 통해 여러 노드의 주요 지표에 대한 비교 차트를 확인할 수 있습니다. 목록에서 조회를 원하는 노드의 **비교** 컬럼의 체크 박스를 선택하면 다음과 같이 오른쪽에 해당 노드의 상세 정보를 비교하는 창이 나타납니다.



상세 보기

노드 상세는 **상세** 옵션을 통해 선택한 노드에 대한 상세 정보를 확인할 수 있습니다. 목록에서 조회를 원하는 노드 **상세** 컬럼의 아이콘을 선택하면 오른쪽에 해당 노드의 상세 정보 창이 나타납니다.

노드 구성

노드 구성 탭에서 선택한 노드 구성 및 전반적인 리소스 사용량 정보를 요약해서 제공합니다. 해당 노드의 하드웨어 및 소프트웨어 구성 요소를 종합적으로 확인할 수 있습니다. 사용자는 노드의 리소스 사용 현황 및 각 구성 요소의 상태를 파악하고 성능 및 운영 안정성을 모니터링할 수 있습니다.

- **노드 시스템 사양**
  - **Architecture**: 노드가 사용하고 있는 하드웨어 아키텍처
  - **Image**: 노드가 실행 중인 운영 체제 이미지 정보
  - **CapacityMemory**: 노드의 총 메모리 용량
  - **CapacityCpu**: 노드의 총 CPU 용량
- **소프트웨어 구성**

- **Container Runtime**: 노드에서 사용되는 컨테이너 런타임의 버전 정보
- **kubelet**: 노드에서 실행 중인 쿠버네티스 kubelet의 버전 정보
- **kube-proxy**: 네트워크 관리 역할을 하는 kube-proxy의 버전 정보

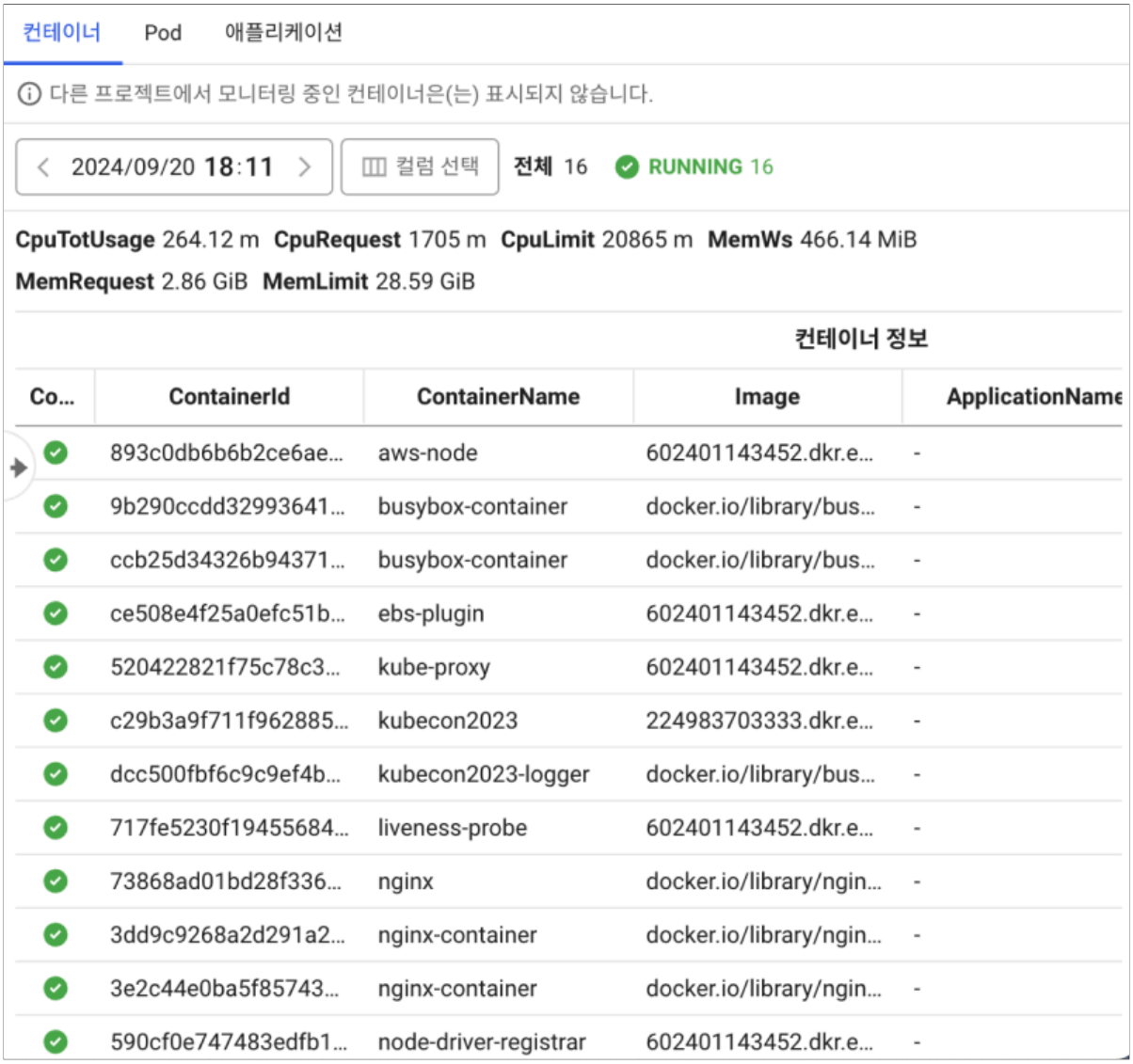
- **리소스 사용 현황**
  - **CPU**: 노드의 CPU 사용량
  - **메모리**: 노드의 메모리 사용량
  - **디스크**: 노드에서 사용 중인 디스크 용량과 사용 비율
  - **네트워크**: 네트워크 전송량

> ⓘ
> - CPU, 메모리, 디스크, 네트워크 클릭 시 조회 기간 동안 사용량 추이를 확인할 수 있는 **메트릭스** 탭으로 이동합니다.
> - **AVG**, **MAX** 버튼을 클릭하여 조회 기간동안 평균 사용량과 최대 사용량을 확인할 수 있습니다.

> ⓘ **노드 구성** 탭은 클러스터 프로젝트(CP)일 경우 조회할 수 있습니다.

리소스

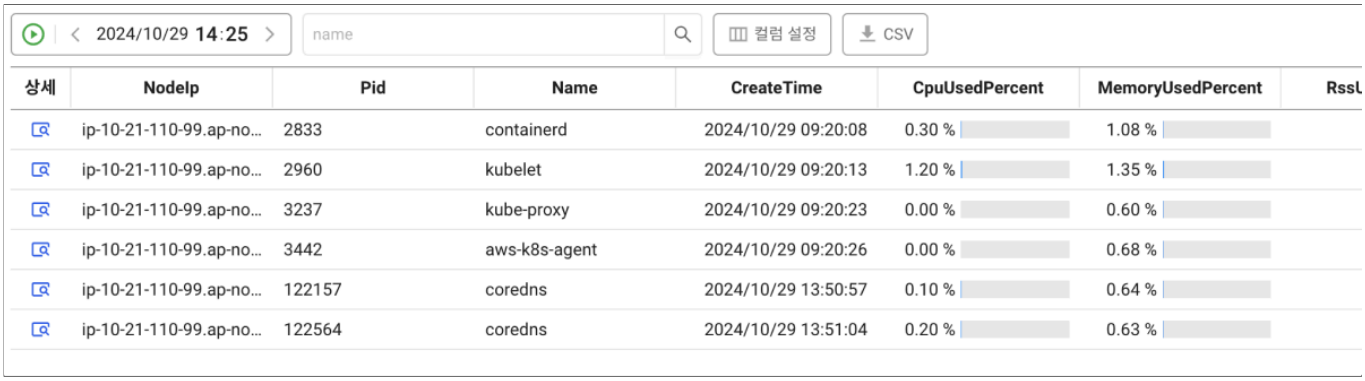
리소스 탭에서 선택한 노드에 스케줄링된 **Pod** 및 해당 Pod에서 실행 중인 **컨테이너**와 **애플리케이션** 목록을 확인할 수 있습니다. 해당 정보는 선택한 시간 구간의 마지막 시점을 기준으로 1분간의 리소스 데이터를 반영하여 표시합니다. 이를 통해 해당 시점의 노드 상태와 각 Pod, 컨테이너, 애플리케이션의 동작 상태를 보다 명확하게 파악할 수 있습니다.



요약 컬럼 선택 시 컨테이너 및 Pod 요약 분석을 확인할 수 있습니다.

프로세스

프로세스 탭에서 선택한 노드 내부에 있는 주요 쿠버네티스 관리 프로세스 정보를 확인할 수 있습니다. 프로세스 문제로 발생하는 클러스터 장애를 예방하고 다운타임을 최소화할 수 있습니다.



상세	NodeIp	Pid	Name	CreateTime	CpuUsedPercent	MemoryUsedPercent	RssU
	ip-10-21-110-99.ap-no...	2833	containerd	2024/10/29 09:20:08	0.30 %	1.08 %	
	ip-10-21-110-99.ap-no...	2960	kubelet	2024/10/29 09:20:13	1.20 %	1.35 %	
	ip-10-21-110-99.ap-no...	3237	kube-proxy	2024/10/29 09:20:23	0.00 %	0.60 %	
	ip-10-21-110-99.ap-no...	3442	aws-k8s-agent	2024/10/29 09:20:26	0.00 %	0.68 %	
	ip-10-21-110-99.ap-no...	122157	coredns	2024/10/29 13:50:57	0.10 %	0.64 %	
	ip-10-21-110-99.ap-no...	122564	coredns	2024/10/29 13:51:04	0.20 %	0.63 %	

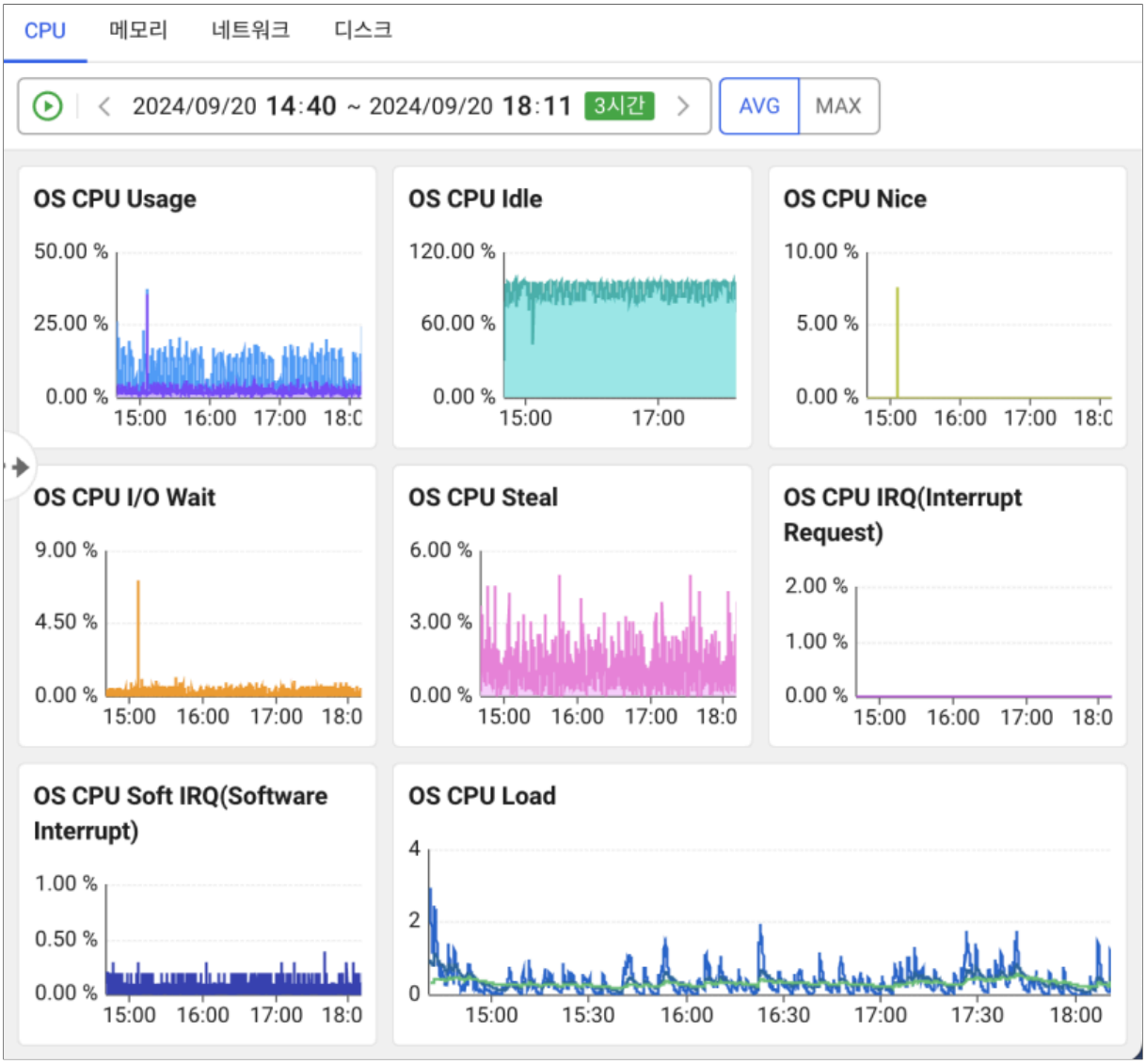
프로세스 상세

- 프로세스 목록 왼쪽 **상세** 컬럼에서 아이콘 선택 시 해당 프로세스에 대한 상세 정보를 확인할 수 있습니다.
- **AVG** 또는 **MAX** 버튼을 클릭하여 3일 이상의 장기 데이터 조회 시 불러올 통계 데이터의 합산 방식을 선택할 수 있습니다.
- 프로세스 주요 정보 값 클릭 시 해당 값이 복사됩니다.

> **프로세스** 탭은 쿠버네티스 에이전트 버전 1.7.14 이상부터 지원합니다.

메트릭스

메트릭스 탭에서 선택한 노드의 리소스 사용량 추이를 확인할 수 있습니다. 조회 기간 동안의 평균(**AVG**), 최대값(**MAX**)을 기준으로 **CPU**, **메모리**, **네트워크**, **디스크** 사용량의 변화를 조회할 수 있습니다.

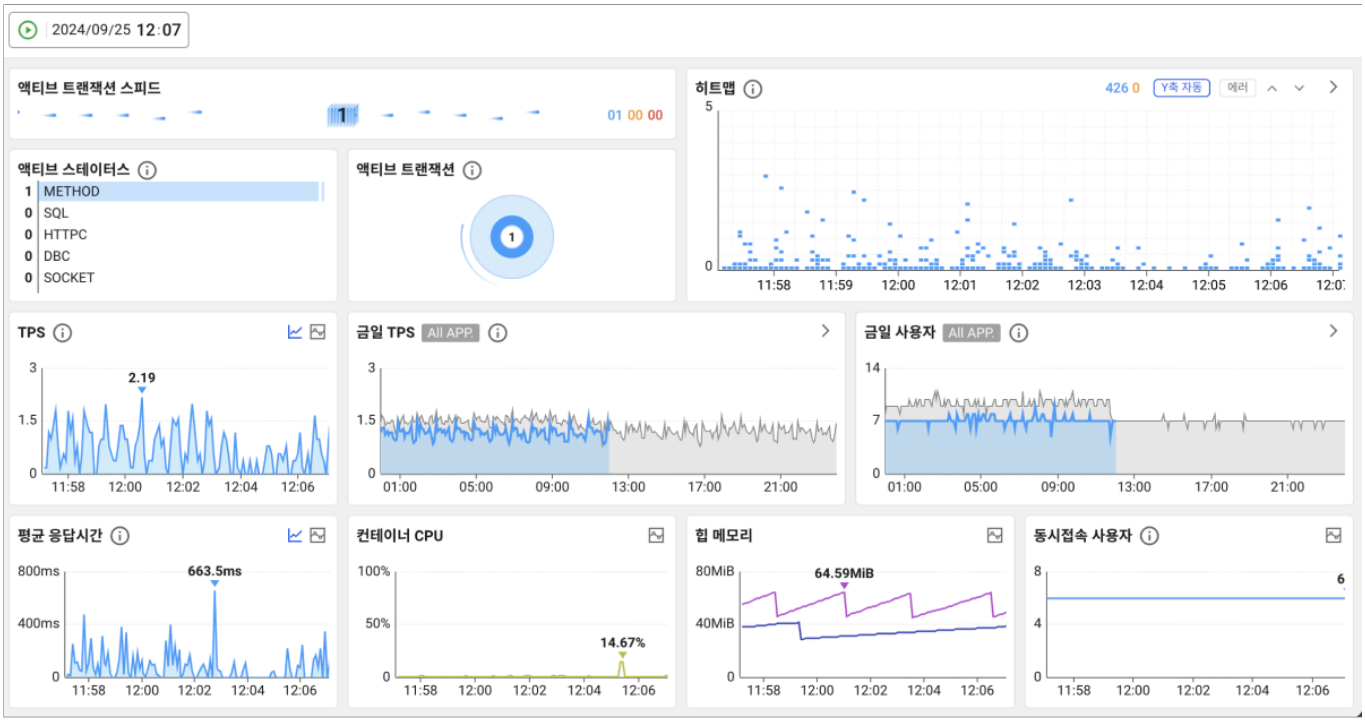


- **AVG**: 조회 기간 동안 평균 사용량
- **MAX**: 조회 기간 동안 최대 사용량
- **Total**: 여러 네트워크 인터페이스의 전체 트래픽 사용량 합산
- **In/Out**: 특정 네트워크 인터페이스의 Inbound(수신) 및 Outbound(송신) 트래픽 사용량

> ⓘ **K8s CPU Request / Allocatable CPU (%)** 위젯은 쿠버네티스 에이전트 버전 1.7.10 이상부터 지원합니다.

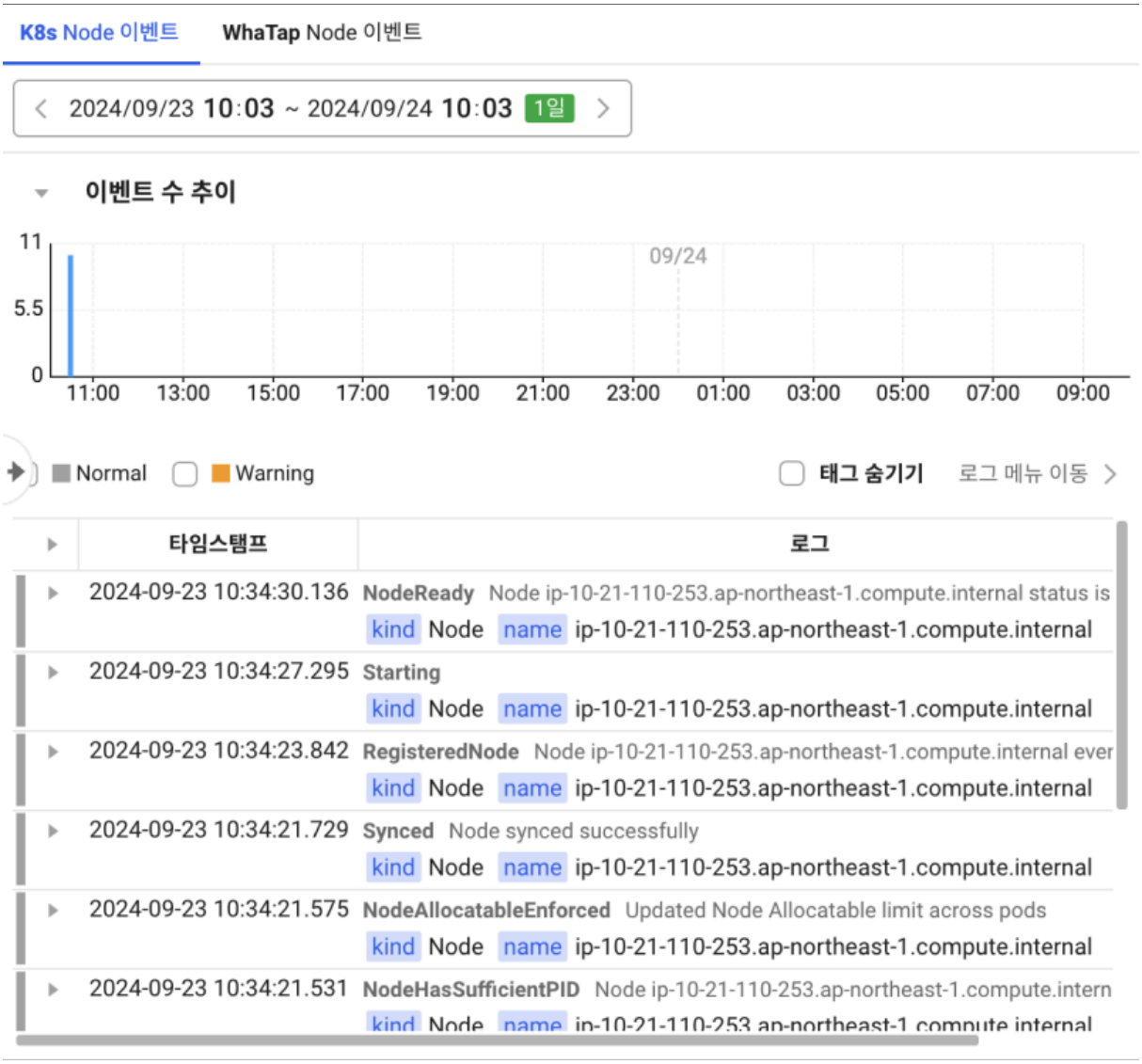
트레이스

트레이스 탭에서 선택한 노드에서 실행 중인 애플리케이션 트레이스 정보를 확인할 수 있습니다.



이벤트

이벤트 탭에서 선택한 노드에서 발생한 **K8s Node 이벤트**와 **WhaTap Node 이벤트**를 확인할 수 있습니다. **K8s Node 이벤트**는 노드의 상태 변화를 포함한 클러스터 내의 중요한 작업 기록을 포함하며, **WhaTap Node 이벤트**는 사용자가 설정한 경고 알림 기준에 따른 이벤트 발생 정보를 제공합니다.


K8s Node 이벤트
WhaTap Node 이벤트
2024/09/23 10:03 ~ 2024/09/24 10:03 1일
이벤트 수 추이
11
5.5
0
09/24
11:00
13:00
15:00
17:00
19:00
21:00
23:00
01:00
03:00
05:00
07:00
09:00
Normal
Warning
태그 숨기기
로그 메뉴 이동
타임스탬프
로그
2024-09-23 10:34:30.136 NodeReady Node ip-10-21-110-253.ap-northeast-1.compute.internal status is
kind Node name ip-10-21-110-253.ap-northeast-1.compute.internal
2024-09-23 10:34:27.295 Starting
kind Node name ip-10-21-110-253.ap-northeast-1.compute.internal
2024-09-23 10:34:23.842 RegisteredNode Node ip-10-21-110-253.ap-northeast-1.compute.internal ever
kind Node name ip-10-21-110-253.ap-northeast-1.compute.internal
2024-09-23 10:34:21.729 Synced Node synced successfully
kind Node name ip-10-21-110-253.ap-northeast-1.compute.internal
2024-09-23 10:34:21.575 NodeAllocatableEnforced Updated Node Allocatable limit across pods
kind Node name ip-10-21-110-253.ap-northeast-1.compute.internal
2024-09-23 10:34:21.531 NodeHasSufficientPID Node ip-10-21-110-253.ap-northeast-1.compute.intern
kind Node name ip-10-21-110-253.ap-northeast-1.compute.internal

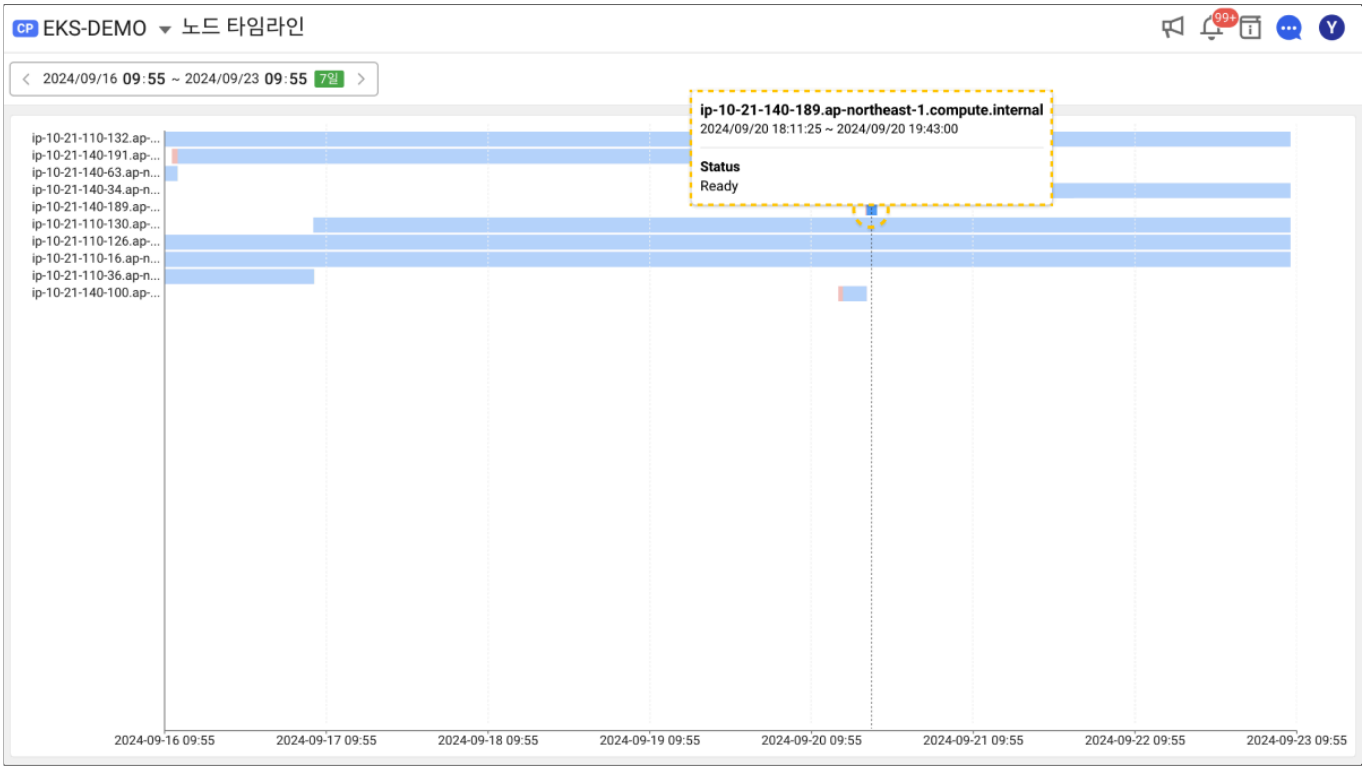
노드 타임라인

> **요구 사항**
>
> - 와탭 쿠버네티스 에이전트 1.7.11 버전 이상이 필요합니다. 타임라인 데이터는 1.7.11 버전 이상에서 수집된 시점부터 제공됩니다. 즉 에이전트가 업데이트된 이후 수집된 데이터부터 반영합니다.
> - 쿠버네티스 클러스터 프로젝트(CP)일 경우 해당 메뉴를 조회할 수 있습니다.

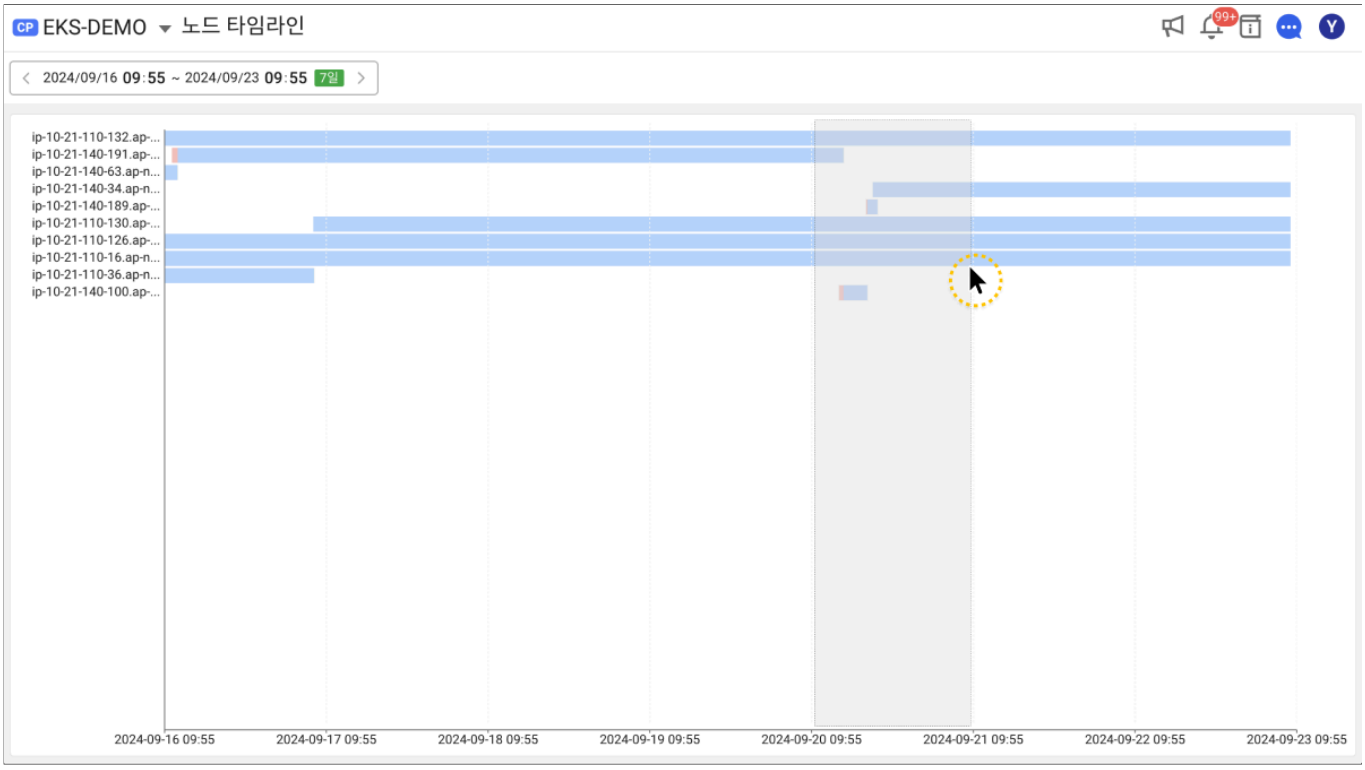
노드 타임라인 메뉴는 노드 상태의 변화를 시간에 따라 추적할 수 있는 기능을 제공합니다. 이를 통해 이미 사라진 노드가 언제 제거되었는지, 해당 시점에 노드가 어떤 상태였는지를 시각적으로 파악할 수 있습니다. 클러스터 운영 시 발생한 이슈를 분석하거나 과거 상황을 확인하는 데 유용합니다.

기본 화면 안내

노드 타임라인 메뉴에 진입 시 다음과 같이 최근 7일 동안 수집된 노드 목록이 나타납니다. 시간 축을 따라 노드의 상태 변화를 시각적으로 확인할 수 있습니다. 각 시간 구간에 마우스를 올리면 해당 구간의 시작 시간과 종료 시간, 그리고 노드의 상태(**Ready**, **NotReady** 등)가 툴팁으로 표시됩니다.



타임라인 상에서 구간을 드래그하여 시간 범위를 조정할 수 있습니다. 원하는 기간을 설정하여 과거 데이터를 확인해 보세요.

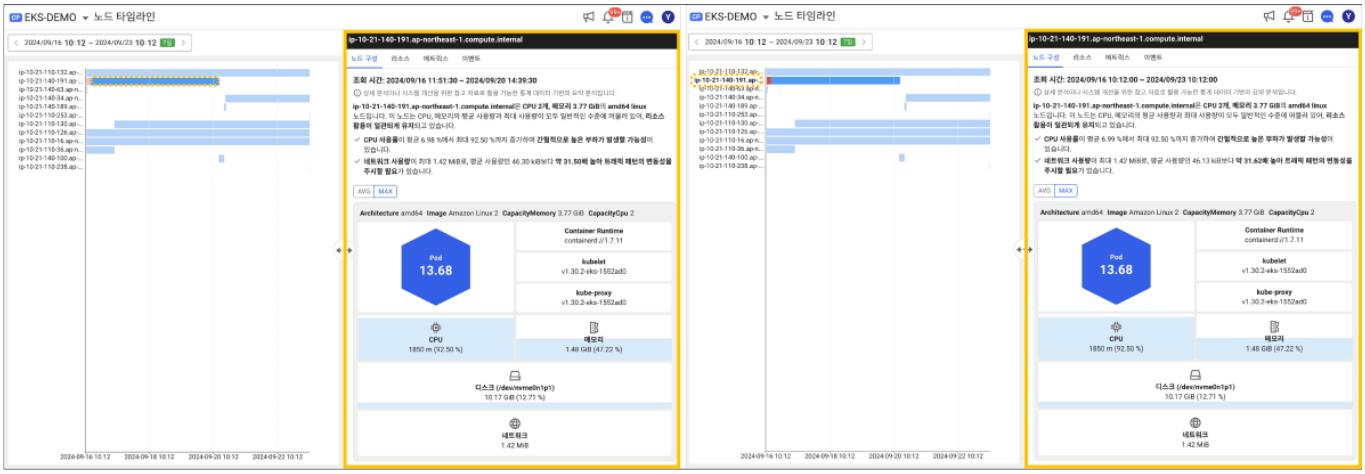


또한 특정 기간 선택 시 해당 기간 동안의 노드 상태를 **상세 보기**를 통해 자세히 확인할 수 있습니다. 자세한 내용은 다음 노드 타임라인 상세 안내를 참조하세요.

> • 노드 상태에 대한 자세한 내용은 다음 문서를 참조하세요.

노드 타임라인 상세

노드 타임라인 상세는 다음과 같이 특정 구간 또는 특정 노드를 선택하여 확인할 수 있습니다.



- 세부 정보를 조회하고자 하는 구간을 선택하여 오른쪽 **상세 보기** 창을 통해 해당 시점의 노드에 대한 상세를 조회할 수 있습니다.
- 차트 왼쪽 목록에서 조회하고자 하는 노드 이름을 선택하여 오른쪽 **상세 보기** 창을 통해 조회 기간 내 해당 노드의 전체 상태를 볼 수 있습니다.

노드 구성

노드 구성 탭에서 선택한 노드 구성 및 전반적인 리소스 사용량 정보를 요약해서 제공합니다. 해당 노드의 하드웨어 및 소프트웨어 구성 요소를 종합적으로 확인할 수 있습니다. 사용자는 노드의 리소스 사용 현황 및 각 구성 요소의 상태를 파악하고 성능 및 운영 안정성을 모니터링할 수 있습니다.

- **노드 시스템 사양**
  - **Architecture**: 노드가 사용하고 있는 하드웨어 아키텍처
  - **Image**: 노드가 실행 중인 운영 체제 이미지 정보
  - **CapacityMemory**: 노드의 총 메모리 용량
  - **CapacityCpu**: 노드의 총 CPU 용량
- **소프트웨어 구성**

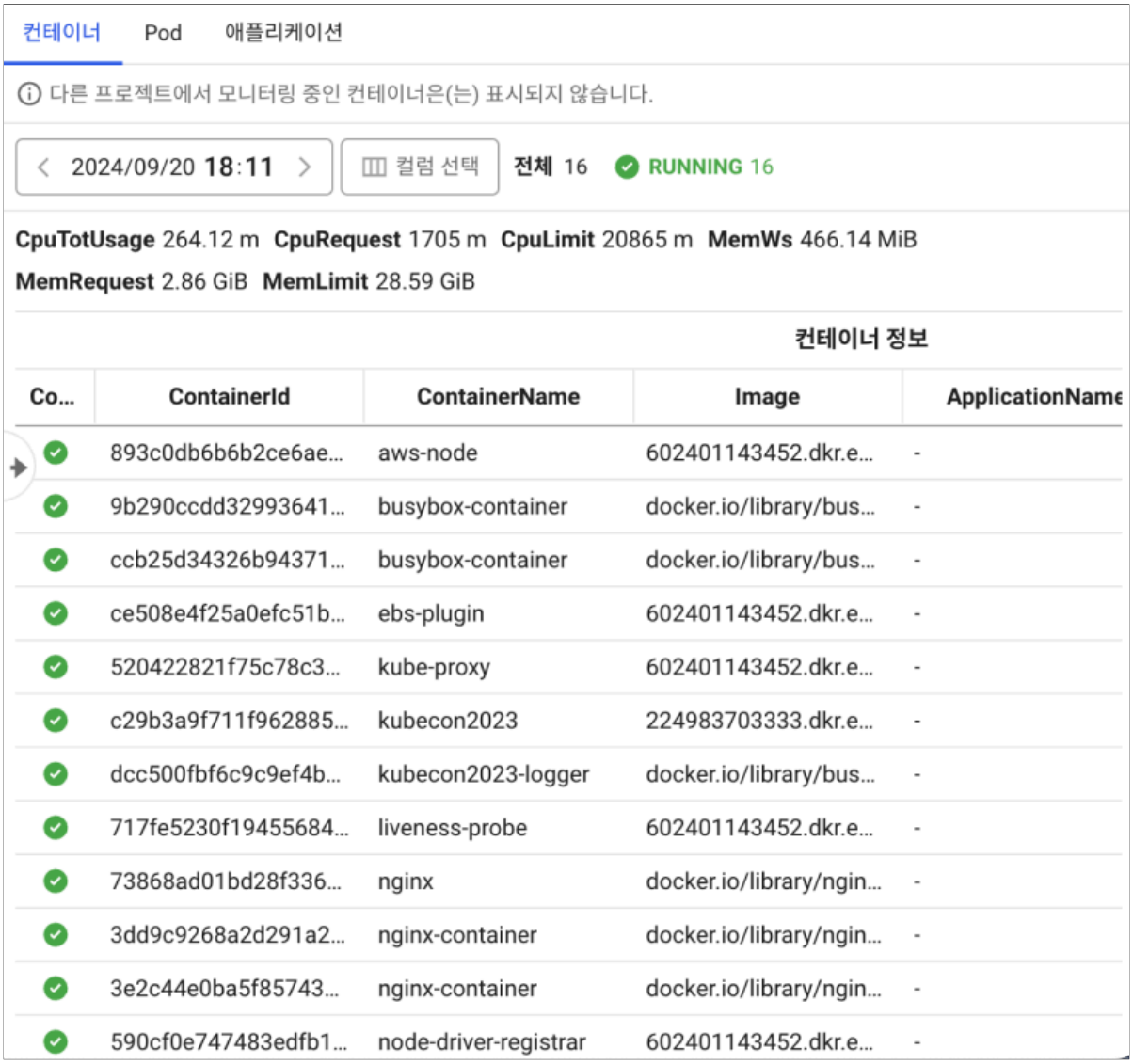
- **Container Runtime**: 노드에서 사용되는 컨테이너 런타임의 버전 정보
- **kubelet**: 노드에서 실행 중인 쿠버네티스 kubelet의 버전 정보
- **kube-proxy**: 네트워크 관리 역할을 하는 kube-proxy의 버전 정보

- **리소스 사용 현황**
  - **CPU**: 노드의 CPU 사용량
  - **메모리**: 노드의 메모리 사용량
  - **디스크**: 노드에서 사용 중인 디스크 용량과 사용 비율
  - **네트워크**: 네트워크 전송량

> ⓘ **AVG**, **MAX** 버튼을 클릭하여 조회 기간동안 평균 사용량과 최대 사용량을 확인할 수 있습니다.

리소스

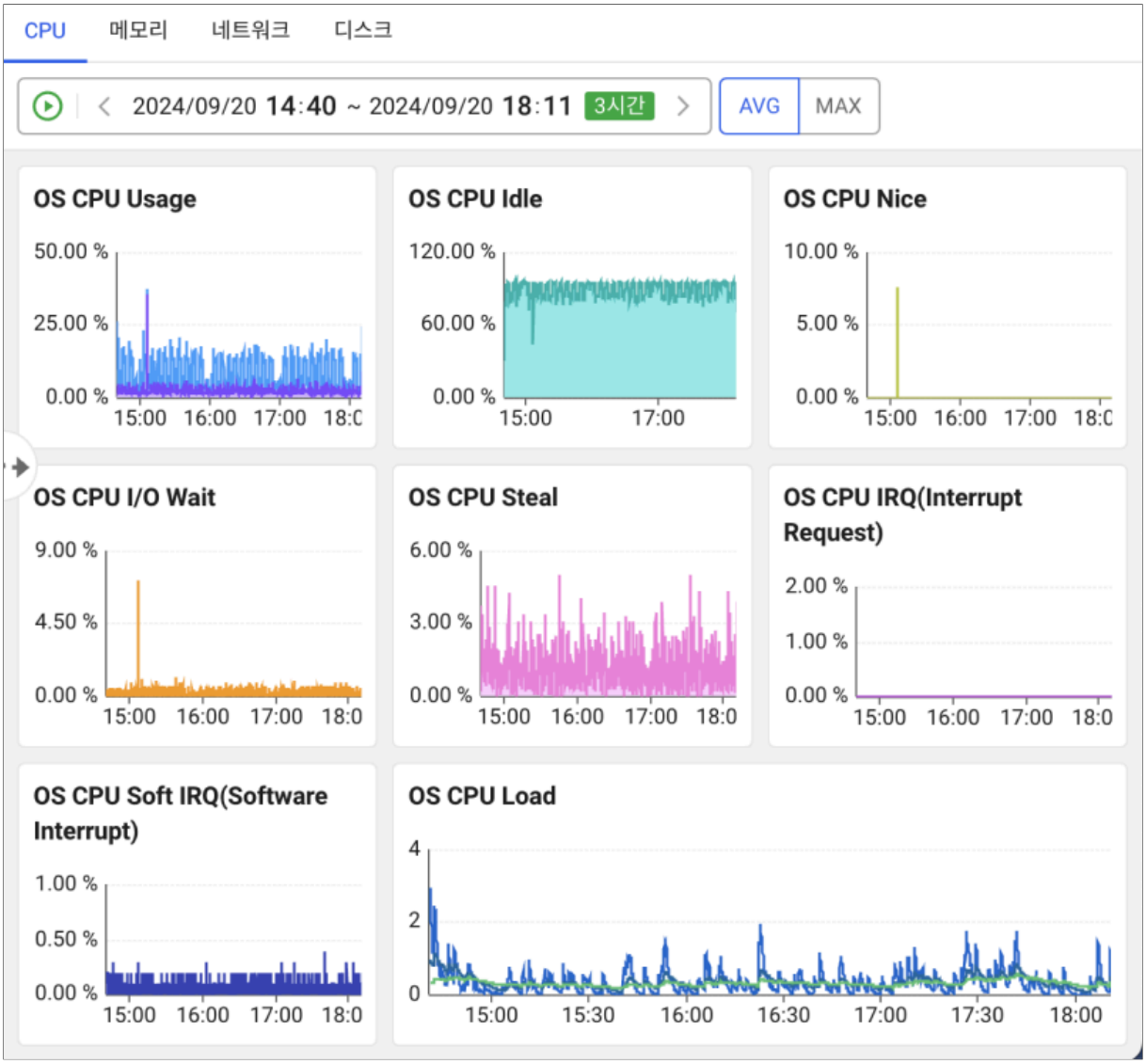
리소스 탭에서 선택한 노드에 스케줄링된 **Pod** 및 해당 Pod에서 실행 중인 **컨테이너**와 **애플리케이션** 목록을 확인할 수 있습니다. 해당 정보는 선택한 시간 구간의 마지막 시점을 기준으로 1분간의 리소스 데이터를 반영하여 표시합니다. 이를 통해 해당 시점의 노드 상태와 각 Pod, 컨테이너, 애플리케이션의 동작 상태를 보다 명확하게 파악할 수 있습니다.



요약 컬럼 선택 시 컨테이너 및 Pod 요약 분석을 확인할 수 있습니다.

메트릭스

메트릭스 탭에서 노드의 리소스 사용량 추이를 제공합니다. 조회 기간 동안의 평균(**AVG**), 최대값(**MAX**)을 기준으로 **CPU**, **메모리**, **네트워크**, **디스크** 사용량의 변화를 확인할 수 있습니다.

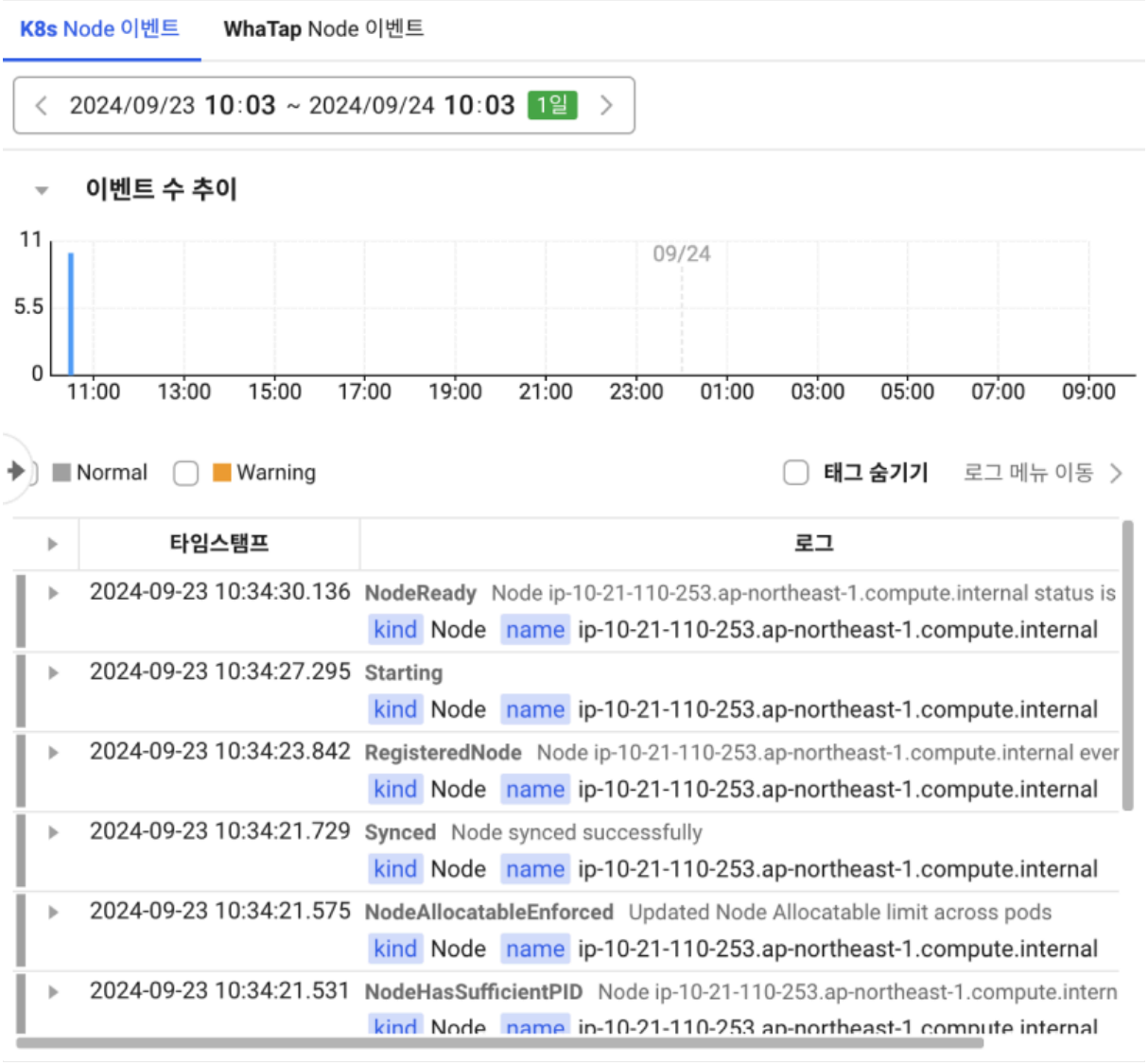


- **AVG**: 조회 기간 동안 평균 사용량
- **MAX**: 조회 기간 동안 최대 사용량
- **Total**: 여러 네트워크 인터페이스의 전체 트래픽 사용량 합산
- **In/Out**: 특정 네트워크 인터페이스의 Inbound(수신) 및 Outbound(송신) 트래픽 사용량

이벤트

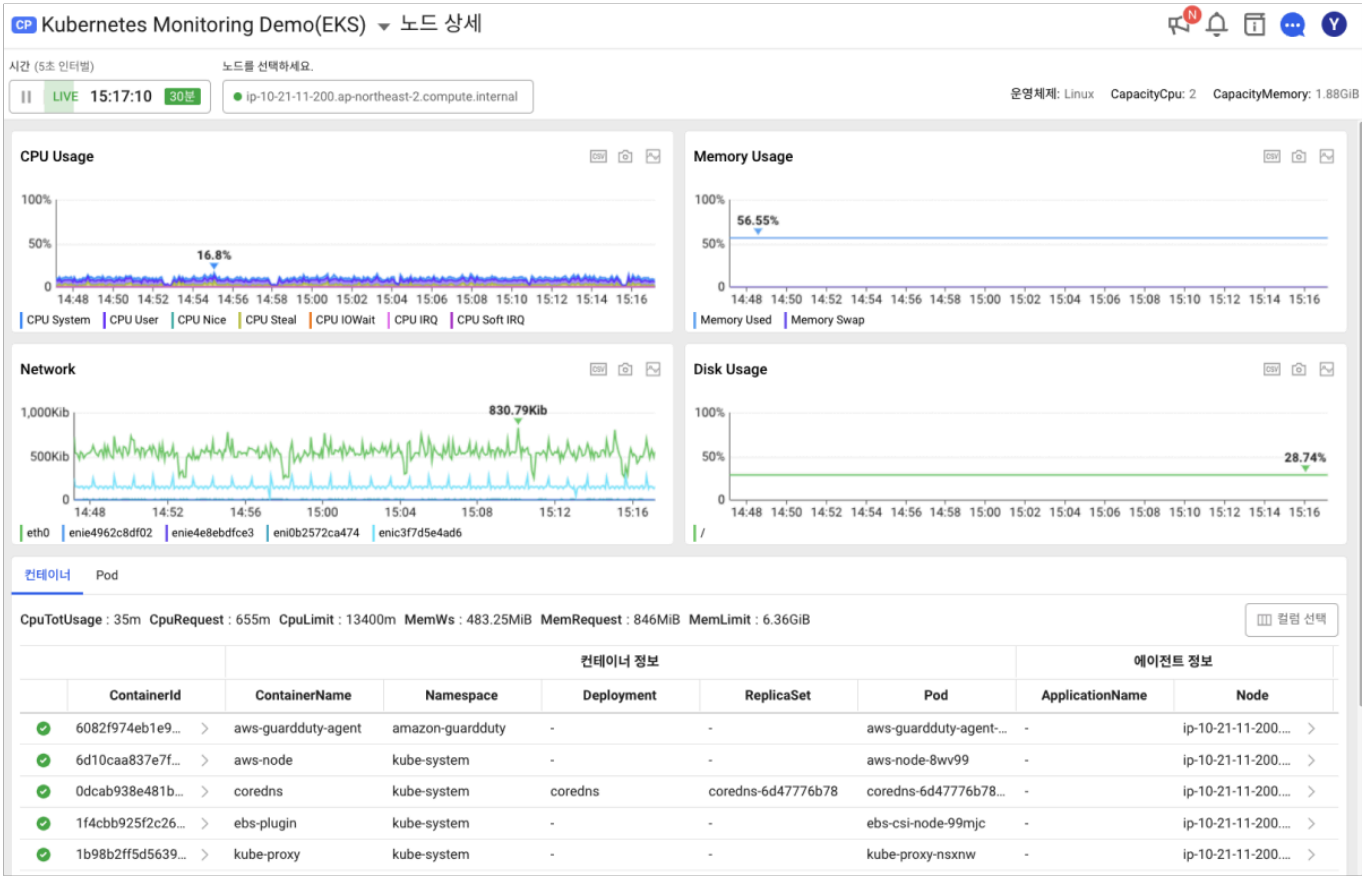
이벤트 탭에서 타임라인에 표시된 시간 동안 발생한 **K8s Node 이벤트**와 **WhaTap Node 이벤트**를 확인할 수 있습니다. **K8s**

Node 이벤트는 노드의 상태 변화를 포함한 클러스터 내의 중요한 작업 기록을 포함하며, **WhaTap Node 이벤트**는 사용자가 설정한 경고 알림 기준에 따른 이벤트 발생 정보를 제공합니다.



노드 상세

노드 상세 화면은 상단의 4개 차트와 노드 내 **컨테이너** 및 **Pod** 상세 정보로 구성되어 있습니다. 노드 셀렉터를 통해 조회를 원하는 노드를 선택할 수 있습니다.



> ⓘ 5초 주기로 최근 1분 데이터를 업데이트합니다.

상단 차트

다음과 같은 4개의 차트 정보를 확인할 수 있습니다.

- **CPU Usage**: 노드의 CPU 사용량을 확인할 수 있습니다.

- **Memory Usage**: 노드의 메모리 사용량을 확인할 수 있습니다.
- **Network**: 노드의 네트워크 인터페이스의 입출력 트래픽과 오류 패킷을 확인할 수 있습니다.
- **Disk Usage**: 노드의 Disk 사용량을 확인할 수 있습니다.

컨테이너 및 Pod 상세 정보

컨테이너

컨테이너 탭 선택 시 노드 내 컨테이너 상세 정보 상단에서 `cpu_quota`, `cpu_request`, `cpu_total_milli`, `mem_limit`, `mem_request`, `mem_working_set` 지표의 합계를 확인할 수 있습니다.

- 원하는 컬럼 선택 시 해당 컬럼 기준으로 컨테이너 목록을 정렬해 조회할 수 있습니다.
- **요약** 컬럼 선택 시 **컨테이너 요약 분석**을 확인할 수 있습니다.
- 컨테이너 선택 시 **컨테이너 목록** 메뉴로 이동합니다. 자세한 내용은 다음 문서를 참조하세요.

Pod

Pod 탭 선택 시 Pod 상세 정보를 조회합니다. 상세 정보 상단에서 `cpu_quota`, `cpu_request`, `cpu_total_milli`, `mem_limit`, `mem_request`, `mem_working_set` 지표의 합계를 확인할 수 있습니다.

- 원하는 컬럼 선택 시 해당 컬럼 기준으로 Pod 목록을 정렬해 조회할 수 있습니다.
- **요약** 컬럼 선택 시 **Pod 요약 분석**을 확인할 수 있습니다.
- Pod 선택 시 **Pod 목록** 메뉴로 이동합니다. 자세한 내용은 다음 문서를 참조하세요.

노드 디스크 목록

> **요구 사항**
>
> - 와탭 쿠버네티스 에이전트 1.7.8 버전 이상이 필요합니다.
> - 쿠버네티스 클러스터 프로젝트(CP)일 경우 해당 메뉴를 조회할 수 있습니다.

노드 디스크 목록을 조회합니다. 와탭 쿠버네티스 모니터링의 **노드 디스크 목록**에서 각 노드에 마운트된 개별 디스크의 현재 사용량, 여유 공간, 성능 지표를 실시간으로 모니터링하여 디스크의 상태를 쉽게 확인할 수 있습니다. 이를 통해 클러스터의 디스크 자원을 효율적으로 관리하고, 잠재적인 문제를 사전에 파악하여 대응할 수 있습니다.

기본 화면 안내

현재 클러스터에 존재하는 노드와 해당 노드에 마운트된 디스크 목록을 다음과 같이 확인할 수 있습니다. 조회 시간 기준 최근 1분간 수집된 디스크 정보를 기준으로 목록을 표시합니다.

비교	OnodeName	DeviceId	FileSystem	BlkSize	TotalSpace	UsedSpace	UsedPercent	FreeSpace
	ip-10-21-11-144....	/dev/nvme0n1p1	xfs	4.00 kiB	19.99 GiB	9.30 GiB	46.50 %	10.69 GiB
	ip-10-21-11-188....	/dev/nvme0n1p1	xfs	4.00 kiB	19.99 GiB	5.20 GiB	26.03 %	14.79 GiB
	ip-10-21-11-98.a...	/dev/nvme0n1p1	xfs	4.00 kiB	19.99 GiB	4.74 GiB	23.72 %	15.25 GiB
	ip-10-21-13-239....	/dev/nvme0n1p1	xfs	4.00 kiB	19.99 GiB	4.64 GiB	23.21 %	15.35 GiB
	ip-10-21-13-36.a...	/dev/nvme0n1p1	xfs	4.00 kiB	19.99 GiB	8.30 GiB	41.53 %	11.69 GiB

미표시 노드 디스크

와탭 쿠버네티스 에이전트가 기동되면 각 노드에 마운트된 디스크 목록이 표시됩니다. 다음과 같은 경우 목록에 표시되지 않습니다.

- 디스크가 특정 경로에 마운트되어 있어야 목록에 표시됩니다. **마운트 포인트가 존재하지 않는 디스크**는 목록에 표시되지 않습니다.
- 와탭 쿠버네티스 에이전트 **기동 후 추가된 디스크**는 목록에 표시되지 않습니다.

필터

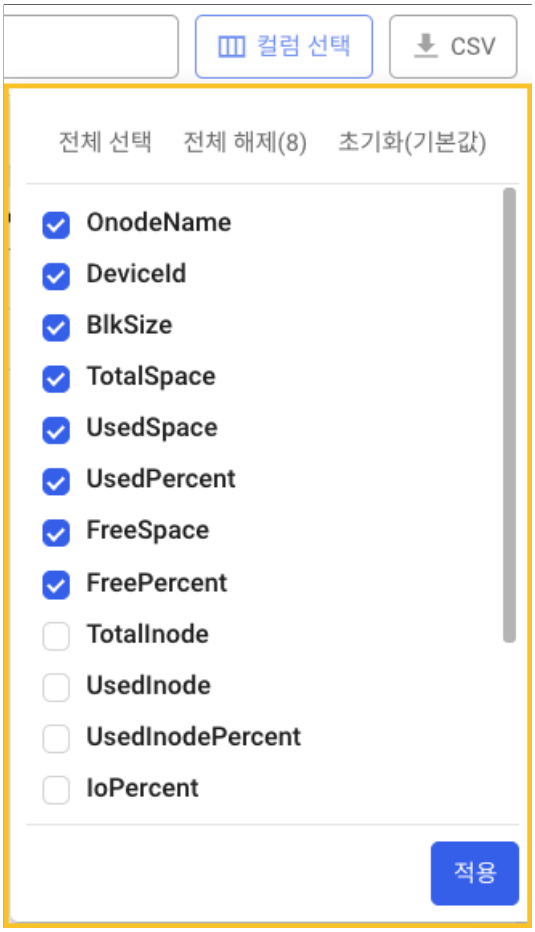
필터 옵션 활용 시 원하는 을 선택해 조회할 수 있습니다. `and` 조건을 통해 여러 개의 필터를 적용할 수 있습니다.

1. **필터** 입력 창에서 + **더하기 아이콘** 선택 시 다음과 같이 **필터 추가하기** 창이 나타납니다.

   - **필터 키**와 **조건**을 선택하세요.
   - 조건에 맞는 값을 선택하세요.
2. 필터 설정 후 **적용** 버튼을 선택해 해당 필터를 적용하세요.
3. 여러 개의 필터를 추가하려면 **필터** 입력 창 또는 **필터 추가하기** 창에서 + **더하기 아이콘**을 선택하세요.

컬럼 선택

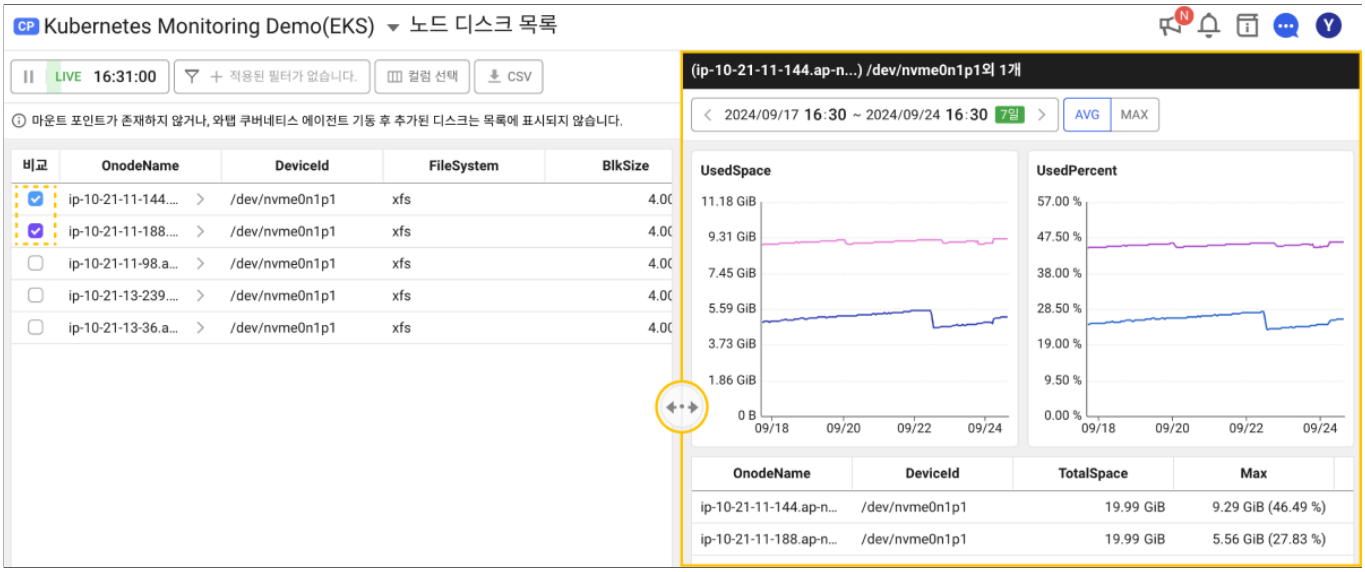
컬럼 선택 옵션 활용 시 원하는 정보를 선택해 목록을 조회할 수 있습니다.



- 선택한 컬럼 목록은 메뉴 재진입 시 유지됩니다.
- **초기화** 버튼 선택 시 사전 정의된 기본 주요 지표 컬럼이 선택됩니다. 이 경우 기존 컬럼 선택 내역은 삭제됩니다.
- 컬럼 선택 후 우측 하단 **적용** 버튼을 클릭 해 선택 내역을 적용할 수 있습니다.

비교하기

비교 옵션을 통해 디스크간 사용량을 비교할 수 있습니다. 목록에서 조회를 원하는 노드 디스크 **비교** 컬럼의 체크 박스를 선택하면 오른쪽에 해당 디스크들의 세부 정보를 비교하는 창이 나타납니다.



예를 들어 **UsedSpace** 및 **UsedPercent** 그래프를 통해 각 디스크에 대해 시간별 사용량 및 사용률을 확인할 수 있습니다. 이를 통해 특정 시간 구간에서의 디스크 사용량 변화를 직관적으로 확인할 수 있습니다. 또한 하단의 상세 데이터 목록을 통해 각 디스크의 최대와 최소 사용량 및 증감률 등의 세부 데이터를 확인할 수 있습니다. 컬럼 정보는 다음과 같습니다.

- **OnodeName**: 해당 디스크가 연결된 노드의 이름을 나타냅니다.
- **DeviceId**: 디스크의 장치 ID를 의미하며, 디스크의 구체적인 장치를 구분할 수 있습니다.
- **Max**: 조회 기간 동안 기록된 디스크의 최대 사용량과 최대 사용률을 나타냅니다.
- **Min**: 조회 기간 동안 기록된 디스크의 최소 사용량과 최소 사용률을 보여줍니다.
- **First / Last**: 조회 기간 중 처음 수집된 데이터와 마지막 수집된 데이터를 통해 사용량 변화를 알 수 있습니다.
- **Increase**: 선택된 기간 동안 해당 디스크의 사용량 증감률을 보여줍니다.
- **Increase(Per Day)**: 조회 기간 동안 하루 평균 사용량 증감률을 계산한 값으로, 디스크 사용량 변동 추세를 확인할 수 있습니다. 조회 기간이 하루 미만일 경우 이 값은 표시되지 않습니다.
- **PredictedSpace(N days later)**: 하루 평균 증감률을 기준으로 N일 후의 예상 사용량을 예측하는 값입니다. 조회 기간이 하루 미만일 경우 이 값은 표시되지 않습니다.

  예시, 조회 기간이 7일일 경우, 7일 후 예상 사용량을 표시합니다.

> ⓘ
> - **AVG**, **MAX** 버튼을 클릭하여 해당 구간의 평균 사용량과 최대 사용량을 확인할 수 있습니다.
> - ↔ 왼쪽 또는 오른쪽으로 드래그해서 화면의 크기를 조절할 수 있습니다.

WhaTap GPU 모니터링

WhaTap GPU 모니터링은 서버와 쿠버네티스 환경 전반에서 GPU 자원의 상태와 활용 현황을 통합적으로 분석할 수 있도록 지원합니다. GPU는 서버를 구성하는 여러 리소스 중 하나이지만, CPU나 메모리 대비 높은 비용과 함께 AI/ML, LLM, HPC 워크로드의 성능에 직접적인 영향을 미치는 핵심 자원입니다.

단순히 **GPU가 장착되어 있고, 살아 있는지** 확인하는 수준을 넘어, 현재 얼마나 활용되고 있는지, 어떤 작업이 점유하고 있는지, 이상 징후는 없는지, 자원 배치는 적절한지까지 확인할 수 있어야 합니다.

WhaTap GPU 모니터링 범위

> ⓘ **WhaTap GPU 모니터링 지원 환경**
>
> - Server 환경에서 GPU 모니터링
> - Kubernetes 환경에서 GPU 모니터링

서버 환경

- GPU 서버 모니터링은 GPU 사용률, 메모리 사용량, 온도, 전력, 클럭, 에러, PCIe/NVLink 통신 상태 등의 핵심 지표를 주기적으로 수집·저장하여, 서버 인프라 관점에서 GPU 장비의 상태를 종합적으로 확인할 수 있도록 지원합니다.
- GPU를 사용하는 프로세스 점유 현황을 함께 제공하여, 어떤 작업이 어떤 GPU를 얼마나 사용하고 있는지를 파악할 수 있습니다. 이를 통해 단순 장비 상태 확인을 넘어, 실제 서버 내 GPU 자원 활용 방식까지 분석할 수 있습니다.
- 모니터링을 통해 과열, 스로틀링, 전력 제한, 비정상 에러, PCIe/NVLink 통신 이상 등의 징후를 조기에 감지할 수 있으며, 장애 예방과 신속한 대응 체계를 마련할 수 있습니다. 이는 서비스 안정성 확보와 운영 효율 향상에 직접적으로 기여합니다.

쿠버네티스 환경

- 쿠버네티스 환경에서 GPU가 주로 AI 학습/추론 워크로드, 배치 작업, 고성능 연산 애플리케이션에 할당되어 사용됩니다. 이때 단순히 노드 단위 GPU 상태만 보는 것으로는 실제 운영 상황을 충분히 파악하기 어렵습니다.
- 와탭의 쿠버네티스 GPU 모니터링은 클러스터, 노드, 파드/컨테이너 단위로 GPU 자원 사용 현황을 확인할 수 있도록 지원합니다. 이를 통해 어떤 워크로드가 GPU를 점유하고 있는지, 특정 노드에 사용량이 편중되어 있는지, GPU 요청/할당

대비 실제 사용 효율은 어떤지 등을 파악할 수 있습니다.

- 쿠버네티스 관점의 GPU 모니터링은 스케줄링 적정성, 자원 편중, 과소/과대 할당 여부를 분석하는 데에도 유용합니다. 결과적으로 GPU 기반 워크로드 운영의 안정성을 높이고, 클러스터 전체 자원의 활용 효율을 개선할 수 있습니다.

엔터프라이즈 환경

- 엔터프라이즈 환경에서 다수의 GPU 서버와 쿠버네티스 클러스터가 함께 운영됩니다. 단일 서버나 개별 쿠버네티스 환경 단위의 GPU 모니터링만으로 전체 자원 상태와 활용 현황을 한눈에 파악하기 어렵기 때문에, 통합 관점의 관리 체계가 필요합니다.
- 수백~수천 장 규모의 GPU를 운영하는 기업에서 팀 또는 업무별로 할당된 GPU 자원이 얼마나 효과적으로 사용되고 있는지를 지속적으로 확인하고 싶어 합니다. 이러한 데이터가 있어야 유휴 자원 재배치, 신규 자원 증설, 추가 구매 여부 등에 대한 합리적인 의사 결정에 도움이 됩니다.
- 다수의 쿠버네티스 기반 GPU 환경과 베어메탈 GPU 서버 환경을 단일 모니터링 체계로 통합함으로써, 분산된 GPU 자원을 하나의 관점에서 관리할 수 있습니다. 이를 통해 조직 전체의 GPU 자원 현황을 통합적으로 파악하고, Capacity Planning까지 연결할 수 있는 엔터프라이즈형 GPU 모니터링을 제공합니다.
- 팀 또는 프로젝트 단위로 Kubernetes 파드 라벨을 GPU 메트릭에 연결해 사용량을 그룹핑하는 방법은 GPU 서비스 그룹핑문서를 참고하세요.

WhaTap GPU 모니터링 제공 가치

GPU 자원 가시성 확보

서버, 클러스터, 노드, 파드, 프로세스 수준까지 GPU 사용 현황을 입체적으로 파악할 수 있습니다.

장애 조기 감지 및 신속 대응

온도 상승, 전력 제한, 클럭 저하, 에러 발생, 통신 이상 등의 징후를 빠르게 식별하여 장애로 확산되기 전에 대응할 수 있습니다.

GPU 활용률 최적화

유휴 GPU, 편중된 사용, 과소 활용 자원을 확인하여 재배치 및 운영 최적화가 가능합니다.

운영 효율 향상

서버와 쿠버네티스 환경을 분리해서 보지 않고 통합적으로 관리함으로써 운영 복잡도를 낮출 수 있습니다.

합리적인 증설 및 투자 판단 지원

실제 사용 데이터를 기반으로 GPU 증설 필요성을 검토하고, 예산 및 구매 계획의 근거 자료로 활용할 수 있습니다.

Capacity Planning 지원

단기 장애 대응뿐 아니라 중장기적인 자원 수요 예측과 운영 전략 수립까지 연결할 수 있습니다.

GPU 장치 이상 탐지 방법

GPU 장치 및 워크로드에서 발생할 수 있는 이상 징후를 탐지하는 방법입니다.

1. Xid 기반 이상 탐지

NVIDIA GPU는 장애 또는 비정상 상태 발생 시 Xid 이벤트를 로그에 기록합니다. 따라서 시스템 로그나 드라이버 로그에서 Xid 키워드 및 코드 발생 여부를 탐지하면 GPU 장치 이상을 비교적 빠르게 식별할 수 있습니다.

예를 들어, 특정 Xid는 GPU 연산 오류, 메모리 접근 문제, 드라이버/하드웨어 이상, 리셋 발생 등의 징후를 의미할 수 있으므로, 운영 환경에서는 Xid 이벤트를 중요한 장애 신호로 활용할 수 있습니다.

2. 상태 지표 기반 이상 탐지

GPU 이상은 반드시 에러 로그로만 드러나지 않을 수도 있습니다. 실제 운영 환경에서는 GPU 사용률, 메모리 사용량, 온도, 전력, 클럭, PCIe/NVLink 통신 상태와 같은 핵심 상태 지표의 변화 패턴을 함께 모니터링함으로써 이상 징후를 더 조기에 감지할 수 있습니다.

다만 이러한 지표는 **절대값만으로 판단하기보다, 업무 특성, 작업 유형, 시간대별 패턴, 평상시 베이스라인을 함께 고려하여 해석하는 것이 중요**합니다. 예를 들어 학습, 추론, 배치 처리, 데이터 전처리와 같은 워크로드는 각각 정상 동작 시의 GPU 사용 패턴이 다르기 때문에, 동일한 수치라도 업무 맥락에 따라 의미가 달라질 수 있습니다.

이러한 관점에서 다음과 같은 패턴 변화는 주요한 점검 대상이 됩니다.

- 평소 업무 패턴과 비교해 GPU 사용률이 과도하게 높거나, 반대로 할당된 자원이 충분히 활용되지 않는 상태가 지속되는 경우
- 작업 종료 이후에도 메모리 점유가 비정상적으로 유지되거나, 특정 시점부터 메모리 사용량이 급격히 증가하는 경우
- 온도가 지속적으로 상승하여 평소 운영 수준을 벗어나거나, 장시간 고온 상태가 유지되는 경우
- 전력 사용량이 제한값 근처에 장시간 머물러 전력 제한에 따른 성능 제약이 의심되는 경우
- 부하 수준 대비 클럭이 기대치보다 낮게 유지되어 스로틀링 또는 비정상 제어 상태가 의심되는 경우
- PCIe/NVLink 통신량이 업무 특성 대비 비정상적으로 증가·감소하거나, infiniband 에 에러 지표가 함께 관측되는 경우

상태 지표를 개별 수치가 아니라 업무 맥락 속 변화 패턴으로 해석하면, 단순 사용 현황을 넘어 성능 저하, 과열, 스로틀링, 전력 제한, 통신 병목, 비정상 동작과 같은 문제를 보다 정확하게 식별할 수 있습니다.

3. 프로세스/워크로드 관점의 이상 탐지

GPU를 점유한 프로세스, Pod, Container 등의 워크로드 정보를 함께 보면, 장비 지표만으로는 파악하기 어려운 이상 상황을 보다 명확하게 해석할 수 있습니다. 다만 이 역시 단순 점유 여부만으로 판단하기보다, 업무 유형, 실행 단계, 반복 주기, 평상시 점유 패턴을 함께 고려해야 합니다.

학습, 추론, 배치 처리, 데이터 전처리와 같은 워크로드는 GPU 사용 방식이 서로 다르기 때문에, 동일한 점유 상태라도 어떤 업무인지에 따라 정상과 비정상의 기준이 달라질 수 있습니다.

이러한 관점에서 다음과 같은 패턴은 주요한 점검 대상이 됩니다.

- 작업 특성에 비해 특정 프로세스나 워크로드가 GPU 메모리를 과도하게 점유한 채 예상보다 오래 해제하지 않는 경우
- GPU 사용률은 낮은데 메모리 점유만 장시간 유지되어, 작업 정체·비정상 대기·메모리 누수 가능성이 의심되는 경우
- 특정 작업이 반복적으로 재시작되거나 실패하면서 GPU 자원을 비효율적으로 점유하는 경우
- 일부 GPU에만 특정 워크로드가 지속적으로 몰려, 사용 불균형이나 국소적 과열이 발생하는 경우
- 동일 계열 업무 대비 특정 프로세스만 과도하게 긴 실행 시간, 높은 메모리 점유, 비정상적인 점유 패턴을 보이는 경우
- Pod 또는 Container 재배치 이후 GPU 점유 패턴이 급격히 변해, 스케줄링 비효율이나 자원 편중이 의심되는 경우

이러한 패턴은 단순 장비 이상뿐 아니라 애플리케이션 또는 워크로드 운용상의 문제까지 함께 식별하는 데 도움될 수 있습니다.

GPU 대시보드

GPU 자원의 할당부터 실제 활용까지의 흐름과 디바이스별 상태를 맵으로 추적할 수 있는 쿠버네티스 관점의 **GPU 대시보드**입니다.

- Total부터 Effective까지 단계별 자원 흐름을 퍼널로 제공하여 할당 대비 실제 활용 수준을 한눈에 파악할 수 있습니다.
- 디바이스 단위 GPU Map으로 클러스터·노드·네임스페이스 기준의 자원 분포와 상태를 시각적으로 확인할 수 있습니다.
- 낭비 자원만 필터링하는 토글을 제공하여 할당되었으나 활용되지 않는 GPU를 빠르게 식별할 수 있습니다.

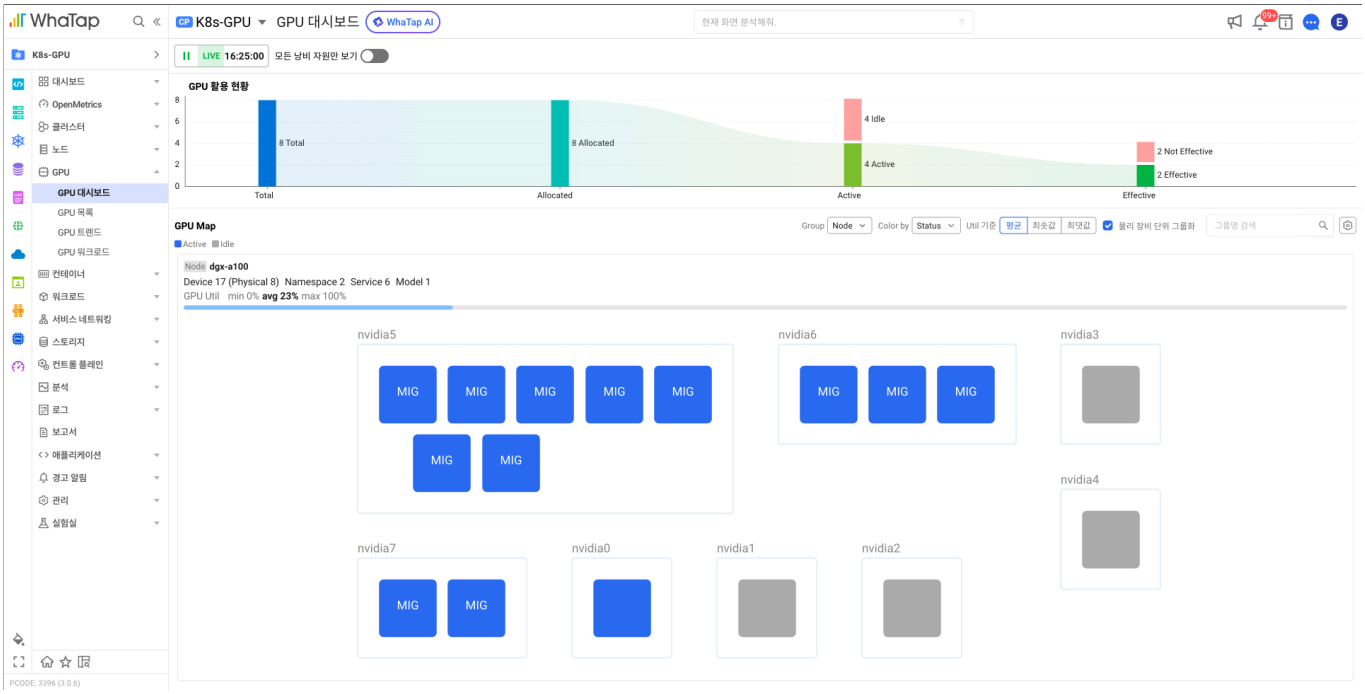
> ⓘ GPU 메뉴는 **GPU 대시보드**, **GPU 목록**, **GPU 트렌드**, **GPU 워크로드** 페이지로 구성됩니다. 이 문서는 자원 활용 현황과 디바이스 맵을 제공하는 **GPU 대시보드** 페이지를 다룹니다.

권한 및 조건

- **지원 환경**: 쿠버네티스 클러스터 프로젝트
- **에이전트 버전**: 쿠버네티스 에이전트 버전 1.8.7 이상
- 오픈 에이전트 설치 필요

기본 화면

클러스터 내 GPU 자원의 활용 현황과 디바이스별 상태를 한눈에 파악할 수 있는 시각화 대시보드입니다.



화면 위쪽에는 LIVE 시각과 **모든 낭비 자원만 보기** 토글이 있습니다. 토글을 켜면 GPU Map에 낭비 자원으로 분류된 디바이스만 표시됩니다.

- **모든 낭비 자원만 보기** 토글은 GPU Map의 표시 대상을 필터링하며, **GPU 활용 현황** 퍼널의 집계 값에는 영향을 주지 않습니다.
- 화면 위쪽 시간 선택자에서 ⏸ 버튼을 선택하면 과거 시간의 데이터를 조회할 수 있습니다.

GPU 활용 현황

GPU 자원이 할당된 후 실제로 활용되기까지의 흐름을 퍼널 차트로 제공합니다. 단계는 **Total**, **Allocated**, **Active**, **Effective** 순으로 좁혀지며, 각 단계에서 활용되지 못한 자원은 **Idle**, **Not Effective**로 분리되어 표시됩니다.

Stage	Description
Total	전체 물리 장치
Allocated	프로세스 또는 Pod가 할당된 장치

Stage	Description
Active	5분 평균 GPU 사용률(Utilization)이 1% 이상인 장치
Idle	5분 평균 GPU 사용률이 1% 미만인 장치
Effective	GPU 사용률이 50%를 초과하는 장치
Not Effective	GPU 사용률이 50% 이하인 장치

Allocated 판정 기준이 프로세스 실행 여부가 아니라 GPU 메트릭의 Pod 라벨 존재 여부로 변경되었습니다. GPU가 할당된 Pod가 CrashLoop 등으로 프로세스를 띄우지 못한 경우에도 할당된 장치로 정확히 집계됩니다. 이에 따라 위 표의 Allocated 설명은 "GPU가 할당된 Pod가 존재하는 장치"로 이해할 수 있습니다.

위 화면 예시에서는 Total 8개 중 8개가 Allocated, 그중 Active 4개·Idle 4개로 나뉘고, Active 중 Effective 2개·Not Effective 2개로 분류됩니다.

Idle과 Not Effective는 할당되었으나 충분히 활용되지 못한 장치로, 자원 낭비를 점검하는 기준이 됩니다.

GPU Map

조회 시점에 수집된 GPU 디바이스를 맵으로 표시합니다. 노드 단위 카드 안에 각 디바이스가 박스로 배치되며, 카드 위쪽에서 해당 노드의 디바이스·네임스페이스·서비스·모델 수와 GPU 사용률 요약을 확인할 수 있습니다.

컨트롤

Control	Option	Description
Group	Cluster / Node / Namespace / Service / Model	디바이스를 묶는 기준
Color by	Status / Utilization / VRAM(%) / Temp	디바이스 박스 색상의 기준 지표
Util 기준	평균 / 최솟값 / 최댓값	사용률 표시에 사용할 통계값
물리 장비 단위 그룹화	켜기 / 끄기	물리 장비 단위로 디바이스를 묶어 표시

Control	Option	Description
그룹명 검색	텍스트 입력	그룹명으로 표시 대상 필터링

노드 카드와 디바이스 박스

각 노드 카드는 해당 노드에 장착된 디바이스를 박스로 표시합니다. 카드 위쪽 요약에는 다음 정보가 표시됩니다.

- **Device n (Physical n)**: 디바이스 수와 물리 장치 수
- **Namespace n / Service n / Model n**: 노드에 연관된 네임스페이스·서비스·모델 수
- **GPU Util (min / avg / max)**: 노드 GPU 사용률의 최솟값·평균·최댓값

디바이스 박스는 상태에 따라 다르게 표시됩니다.

Display	Meaning
MIG 표기 박스	MIG 인스턴스로 분할된 디바이스
색상 박스	Active 디바이스, Color by 기준에 따라 색상 결정
회색 박스	Idle 디바이스(5분 평균 GPU 사용률 1% 미만)

범례에서는 **Active**와 **Idle**을 색상으로 구분합니다.

디바이스 박스에 마우스를 올리면 디바이스 종류(Physical/MIG), GPU Utilization, Status, Device, Pod, Node, Namespace, Service 정보를 확인할 수 있습니다.

> ⓘ
> - **Color by** 옵션을 변경하면 같은 디바이스라도 기준 지표(상태·사용률·VRAM·온도)에 따라 색상이 달라집니다.
> - MIG로 분할된 디바이스는 물리 장치 1개가 여러 인스턴스로 표시되므로, 디바이스 수와 물리 장치 수가 다를 수 있습니다.

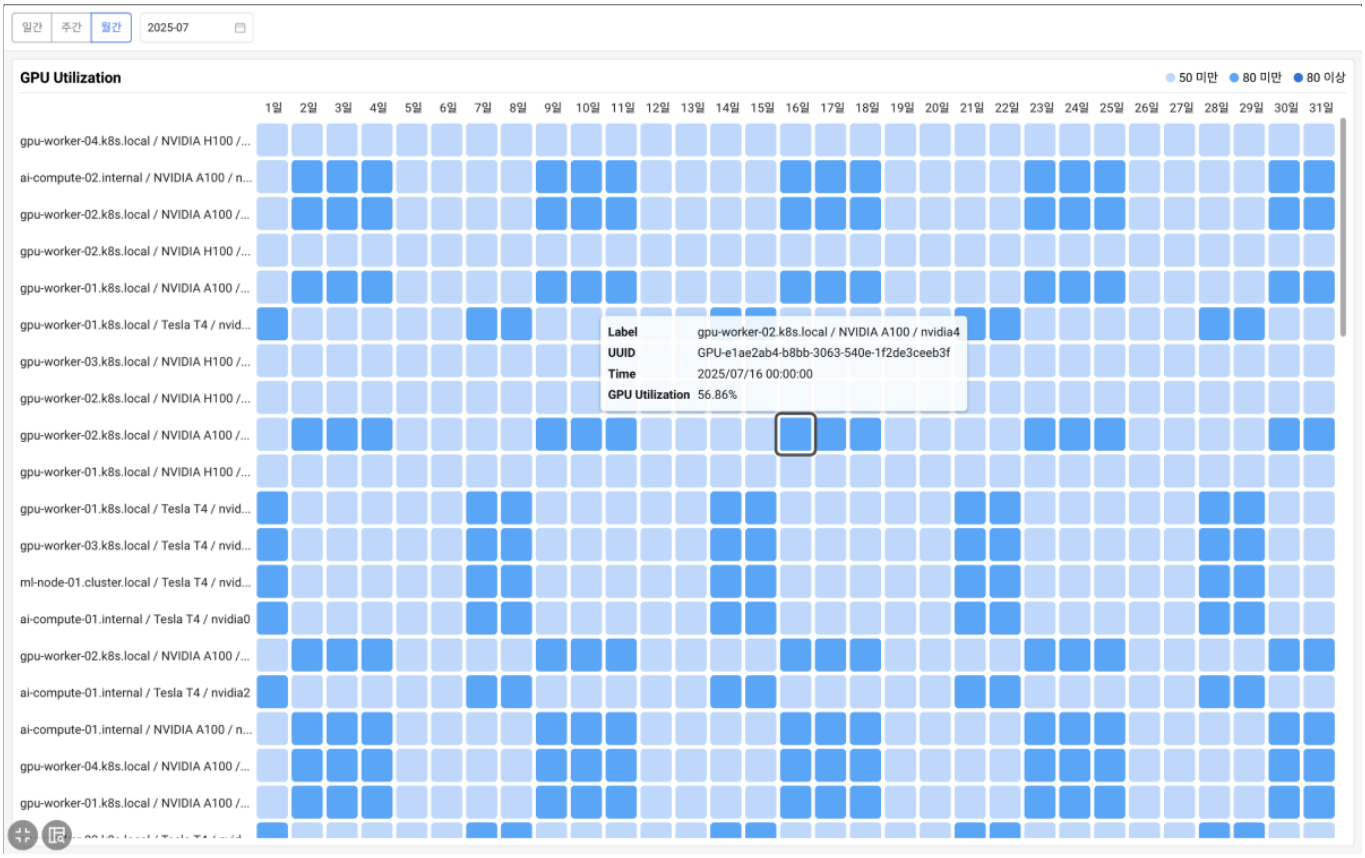
GPU 트렌드

GPU 트렌드는 GPU 장비별 사용량을 장기 통계 데이터로 시각화하여, 특정 시점에 어떤 장비가 비효율적으로 사용되었는지 빠르게 식별할 수 있습니다.

기본 화면

GPU 장기 사용량 통계 및 패턴을 확인할 수 있습니다. 조회 단위는 **일간, 주간, 월간**을 지원합니다. 장비별 GPU 사용률이 블록 형태로 표시되며, 색상 진하기에 따라 사용량 수준을 구분할 수 있습니다.

- **50% 미만**: 연한 파랑
- **80% 미만**: 파랑
- **80% 이상**: 진한 파랑



상세 정보 확인

각 블록에 마우스를 올리면 해당 시점과 장비의 상세 정보를 툴팁으로 확인할 수 있습니다.

- **Label**: 장비명 및 GPU 모델
- **UUID**: GPU 고유 식별자
- **Time**: 기준 시각
- **GPU Utilization**: 해당 시점의 GPU 사용률(%)

GPU 서비스 그룹핑

> GPU 모니터링 전반에 대한 개요는 WhaTap GPU 모니터링 문서를 참고하세요.

WhaTap Operator의 GPU 모니터링 기능에서 **애플리케이션(GPU 사용) 파드의 Kubernetes Label을 GPU 메트릭에 부착**하여, WhaTap에서 서비스/프로젝트 단위로 GPU 사용량을 그룹핑·조회하는 방법을 설명합니다.

- 적용 대상: `WhatapAgent` CR (`monitoring.whatap.com/v2alpha1`)
- 관련 필드: `spec.features.k8sAgent.gpuMonitoring.groupLabel` , `groupLabelAllowlistRegex`
- 핵심 결과 라벨: `whatap_kube_label_gpu_group`

> **어떤 문제를 해결하나요?**
>
> 기본 DCGM 메트릭(`DCGM_FI_DEV_GPU_UTIL` 등)에는 GPU/노드/파드 식별자는 있지만 **사용자가 운영하는 "서비스" 단위의 정보**(예: 프로젝트 ID, 팀, 워크로드 종류)는 들어있지 않습니다.
>
> - "프로젝트 A가 사용하는 전체 GPU 사용률 합계" 같은 질의가 어렵습니다.
> - 멀티테넌트(여러 팀이 한 클러스터 공유) 환경에서 비용/사용량 정산이 어렵습니다.
>
> 이 기능을 사용하면 파드에 붙인 라벨(`prjId=projectA` 등)이 **DCGM 메트릭 라벨로 자동 전파**되고, WhaTap의 공통 라벨 `whatap_kube_label_gpu_group` 으로 정규화되어 대시보드/검색에서 그룹핑 키로 바로 사용할 수 있습니다.

빠른 시작

GPU 모니터링 + 그룹 라벨 활성화

```
apiVersion: monitoring.whatap.com/v2alpha1
kind: WhatapAgent
metadata:
  name: whatap
  namespace: whatap-monitoring
```

```
spec:
  features:
    k8sAgent:
      gpuMonitoring:
        enabled: true
        # 그룹핑 키로 사용할 파드 라벨 키
        groupLabel: "prjId"
        # (선택) 노출 허용 라벨 키 정규식. 미설정 시 ^(prjId)$ 자동 적용
        # groupLabelAllowlistRegex: "^(prjId|team)$"
        # (선택) 모든 GPU 메트릭에 cluster 라벨 부여
        clusterName: "prod-gpu-cluster"
```

GPU 사용 파드에 라벨 부여

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: ml-train
  namespace: ml
spec:
  replicas: 2
  selector:
    matchLabels:
      app: ml-train
  template:
    metadata:
      labels:
        app: ml-train
        prjId: "projectA"       # ◀ 그룹핑 키 (groupLabel 과 동일)
    spec:
      containers:
        - name: trainer
          image: my-registry/trainer:latest
          resources:
            limits:
              nvidia.com/gpu: 1
```

ⓘ 라벨은 **Pod 레벨**(`spec.template.metadata.labels`)에 부여해야 합니다. Deployment 라벨이 아닙니다.

사용 예시 (활용 시나리오)

프로젝트(prjId) 단위 사용량

- `groupLabel: prjId`
- 파드 라벨: `prjId: projectA` , `prjId: projectB`
- WhaTap 대시보드에서 `whatap_kube_label_gpu_group` 으로 group by → 프로젝트별 GPU 사용률/메모리 사용량 비교

팀(team) 단위 정산

```
gpuMonitoring:
  enabled: true
  groupLabel: "team"
  groupLabelAllowlistRegex: "^(team)$"
```

- 파드 라벨: `team: search` , `team: recsys` , ...

여러 라벨을 함께 노출하고 싶은 경우

`groupLabel` 로 정규화되는 라벨은 한 개(`whatap_kube_label_gpu_group`)지만, 다른 파드 라벨도 메트릭 라벨로 같이 노출하고 싶다면 `groupLabelAllowlistRegex` 로 확장할 수 있습니다.

```
gpuMonitoring:
  enabled: true
  groupLabel: "prjId"
  groupLabelAllowlistRegex: "^(prjId|team|env)$"
```

- `whatap_kube_label_gpu_group` ← `prjId` 값
- `team` , `env` 라벨도 DCGM 메트릭에 그대로 노출 (정규화는 되지 않음)

검증 방법

dcgm-exporter 환경변수 확인

```
kubectl -n whatap-monitoring get ds whatap-node-agent \
  -o jsonpath='{.spec.template.spec.containers[?(@.name=="dcgm-exporter")].env}' | jq
```

출력 결과에 다음 두 환경변수가 포함되어 있어야 파드 라벨 수집이 활성화된 상태입니다.

- `DCGM_EXPORTER_KUBERNETES_ENABLE_POD_LABELS=true`
- `DCGM_EXPORTER_KUBERNETES_POD_LABEL_ALLOWLIST_REGEX=^(prjId)$`

메트릭 직접 조회 (파드에 라벨 있어야 함)

```
# dcgm-exporter Pod 하나에 접속
POD=$(kubectl -n whatap-monitoring get pod -l name=whatap-node-agent -o name | head -1)
kubectl -n whatap-monitoring exec "$POD" -c dcgm-exporter -- \
  wget -qO- localhost:9400/metrics | grep DCGM_FI_DEV_GPU_UTIL | head
```

```
DCGM_FI_DEV_GPU_UTIL{gpu="0", ..., pod="ml-train-xxxx", namespace="ml", prjId="projectA"} 87
```

OpenAgent scrape 설정 확인

```
kubectl -n whatap-monitoring get configmap whatap-open-agent-config -o yaml | grep -A2 whatap_kube_label_gpu_group
```

출력에 다음 relabel 블록이 포함되어 있어야 합니다. `source_labels`의 값은 설정한 `groupLabel` 키와 일치해야 합니다.

```
- action: replace
  source_labels: [prjId]
  target_label: whatap_kube_label_gpu_group
```

트러블슈팅

증상	원인 / 확인 포인트
메트릭에 `prjId` 라벨이 안 보임	(1) Pod 라벨에 `prjId` 부여되었는지 (`spec.template.metadata.labels`), (2) `dcgm-exporter`가 재시작되어 새 env 반영되었는지
`prjId`는 보이는데 `whatap_kube_label_gpu_group`이 안 보임	OpenAgent ConfigMap 갱신 후 OpenAgent 파드 재기동 필요. `kubectl rollout restart deploy/whatap-open-agent -n whatap-monitoring`
`groupLabel` 값에 `-`, `.`, `/` 포함	Prometheus 라벨 키 규칙상 `_`로 치환된 키로 source_labels가 구성됨. allowlist 정규식도 원본 키 기준이므로 그대로 설정 가능
멀티 라벨을 노출하고 싶음	`groupLabelAllowlistRegex`를 `^(prjId|team|env)$`처럼 명시
모든 GPU 메트릭에 클러스터 식별이 필요	`clusterName` 필드를 함께 설정 (메트릭에 `cluster=<값>` 부착)

관련 필드 요약

필드	설명	기본값
`gpuMonitoring.enabled`	GPU 모니터링(dcgm-exporter) 활성화	`false`
`gpuMonitoring.groupLabel`	그룹핑에 사용할 Pod 라벨 키. 설정 시 `whatap_kube_label_gpu_group`으로 정규화	-
`gpuMonitoring.groupLabelAllowlistRegex`	dcgm-exporter가 노출할 Pod 라벨 키 정규식	`^(<groupLabel>)$`
`gpuMonitoring.clusterName`	모든 GPU 메트릭에 `cluster` 라벨 부착	-

참고

- git repo: https://github.com/whatap/whatap-operator.git
- 구현 코드: `internal/controller/install_agents.go` (`addDcgmExporterToNodeAgent` , GPU scrape config 생성부)
- CRD 정의: `api/v2alpha1/whatapagent_types.go` (`GpuMonitoringSpec`)
- 예제: `examples/whatap-agent-all-options.yaml` , `examples/whatap-agent-gpu-custom-env.yaml`

컨테이너 목록

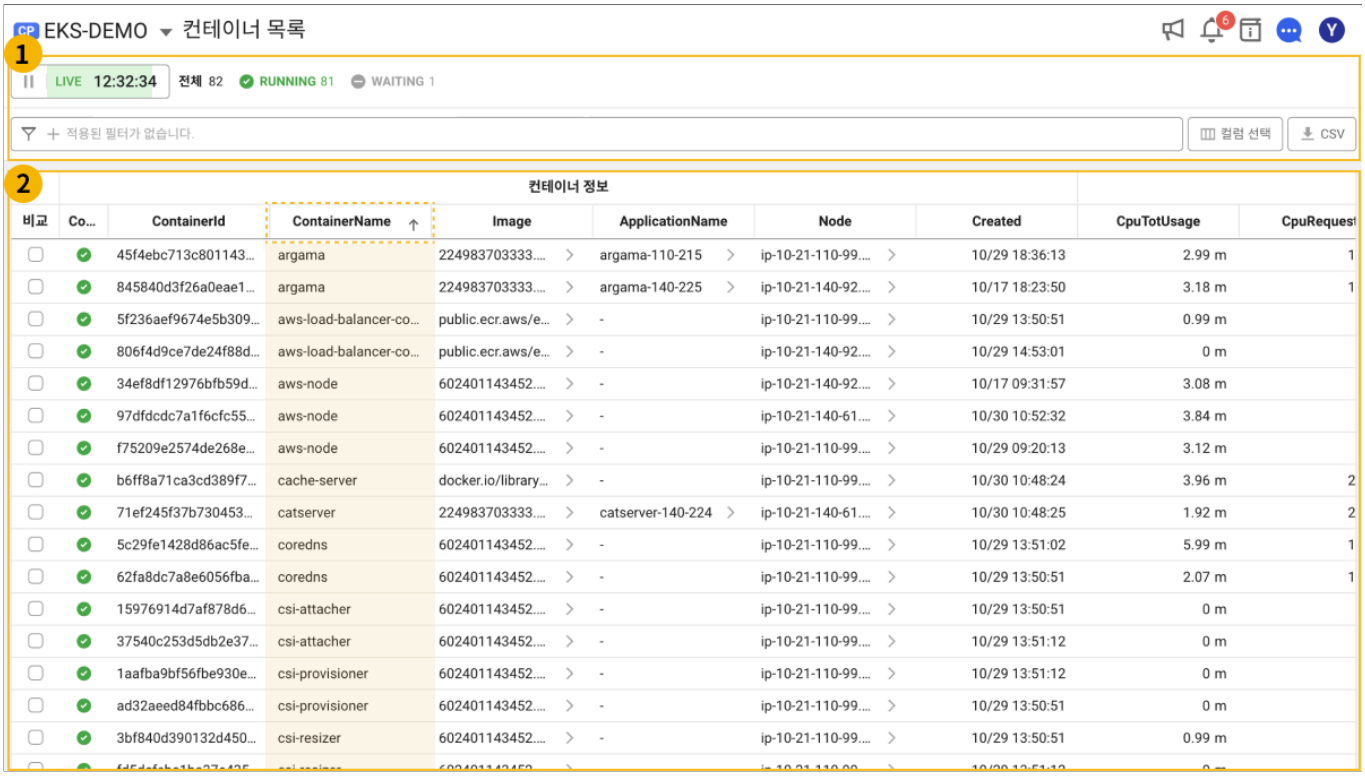
> ⓘ 특정 컬럼의 데이터가 표시되지 않는 것은 쿠버네티스 노드 에이전트가 구 버전이기 때문일 수 있습니다. 이런 경우 충분한 데이터가 수집될 수 있도록 쿠버네티스 에이전트 버전을 업데이트하세요.

컨테이너 목록 메뉴를 통해 현재 프로젝트의 컨테이너의 상세 정보를 목록 형식으로 확인할 수 있습니다.

- 10초 주기로 실시간 데이터를 업데이트합니다.
- 컨테이너 상태 요약, 비교, 필터 및 CSV 파일 다운로드 기능을 제공합니다.

  CSV 파일명, kubernetes_contianer_list_ `{pcode}` _ `{timestamp}` .csv

기본 화면 안내

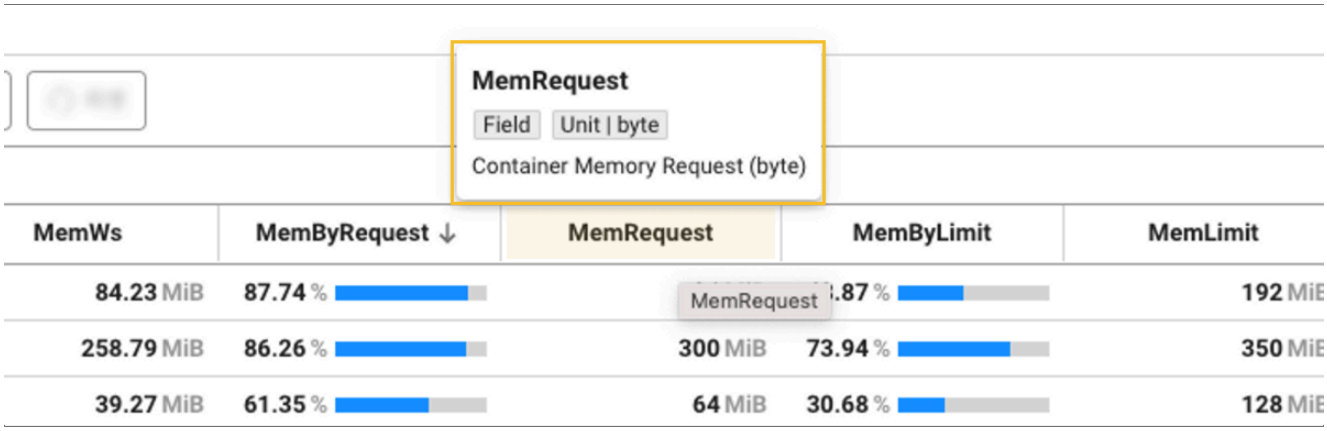


1 영역에서 원하는 옵션을 선택해 2 영역에서 데이터를 조회할 수 있습니다.

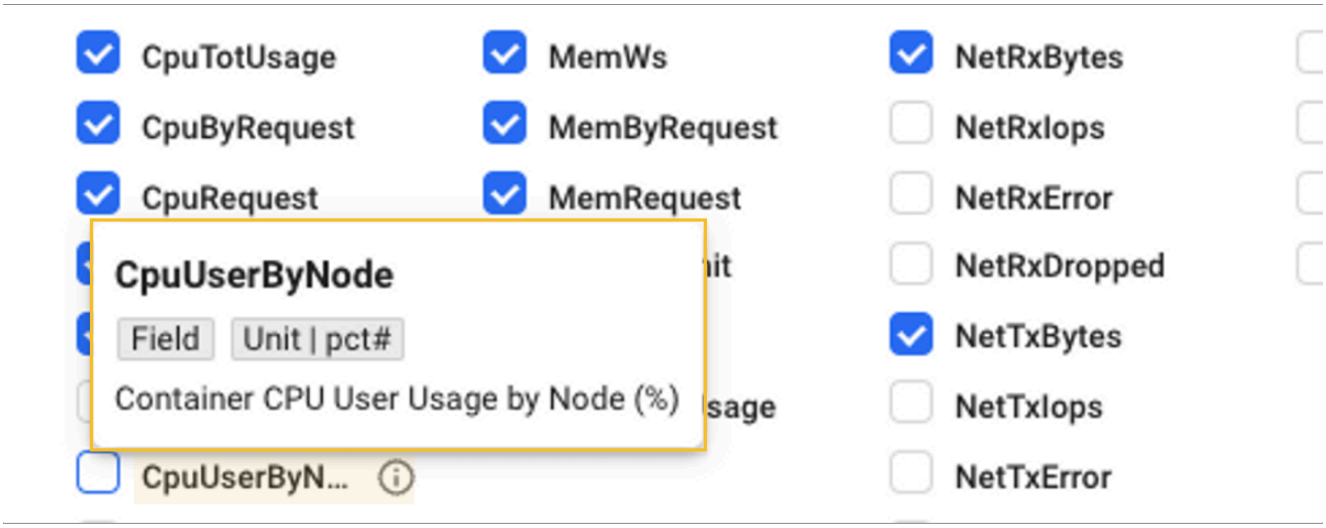
- 1 영역 좌측 상단에서 전체 컨테이너 개수와 상태별 컨테이너 개수를 확인할 수 있습니다.
- 2 영역에서 원하는 컬럼을 선택 시 해당 컬럼 기준으로 컨테이너 목록을 정렬해 조회할 수 있습니다.
- 2 영역 **Node** 컬럼에서 **노드 정보 보기** 아이콘 선택 시 **노드 목록** 메뉴로 이동합니다.

메트릭 정보

- 원하는 컬럼 위에 마우스 커서를 위치시키면 다음과 같이 해당 컬럼이 제공하는 메트릭 정보를 확인할 수 있습니다.



- **컬럼 선택** 옵션 화면 내 컬럼 목록에서 컬럼별 메트릭 정보를 확인할 수 있습니다. 원하는 컬럼 위에 마우스 커서를 위치시키면 컬럼 이름 우측에 ⓘ **인포 아이콘**이 나타납니다. ⓘ **인포 아이콘** 위에 마우스 커서를 위치시키면 다음과 같이 메트릭 정보를 확인할 수 있습니다.



필터

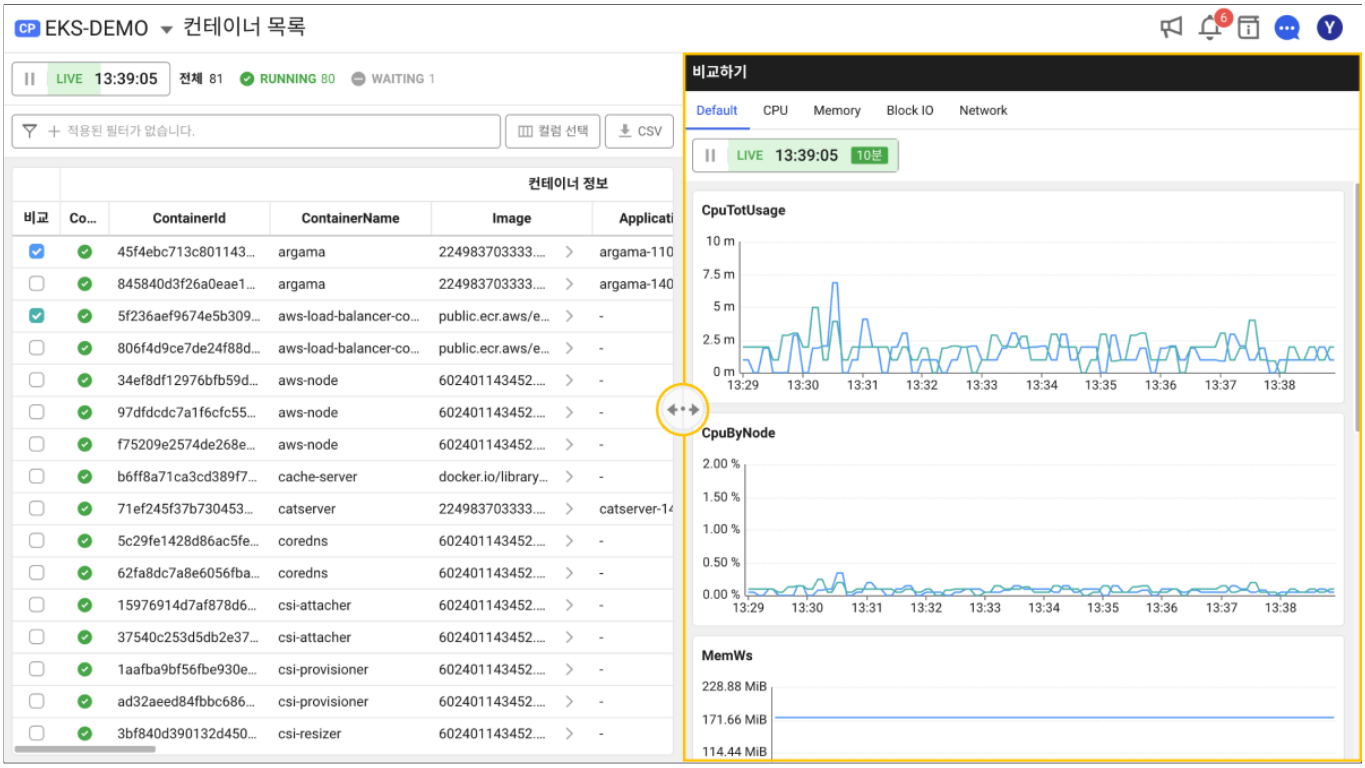
필터 옵션 활용 시 원하는 컨테이너 목록을 선택해 조회할 수 있습니다. `and` 조건을 통해 여러 개의 필터를 적용할 수 있습니다.

1. **필터** 입력 창에서 + **더하기 아이콘** 선택 시 다음과 같이 **필터 추가하기** 창이 나타납니다.

   - **필터 키**와 **조건**을 선택하세요.
   - 조건에 맞는 값을 선택하세요.
2. 필터 설정 후 **적용** 버튼을 선택해 해당 필터를 적용하세요.
3. 여러 개의 필터를 추가하려면 **필터** 입력 창 또는 **필터 추가하기** 창에서 + **더하기 아이콘**을 선택하세요.

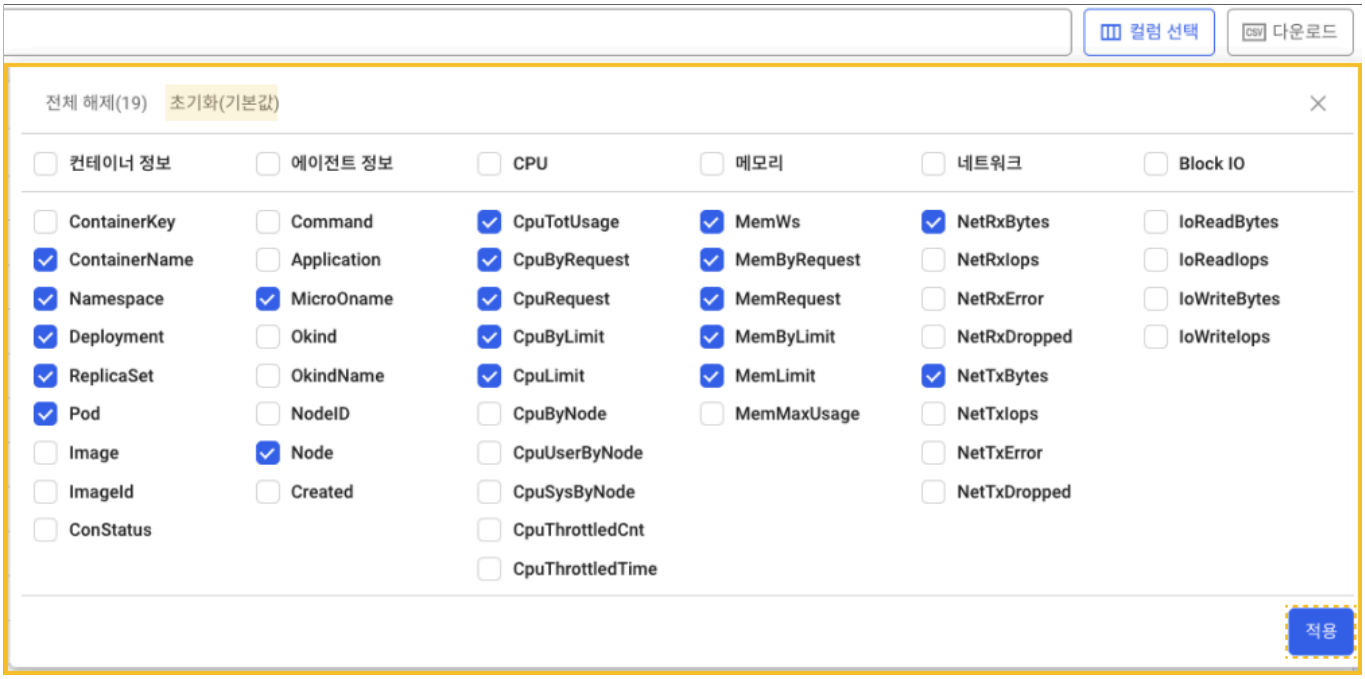
비교하기

비교 옵션을 통해 여러 컨테이너의 주요 지표에 대한 비교 차트를 확인할 수 있습니다. 목록에서 조회를 원하는 컨테이너의 **비교** 컬럼의 체크 박스를 선택하면 다음과 같이 오른쪽에 해당 컨테이너의 상세 정보를 비교하는 창이 나타납니다.



컬럼 선택

컬럼 선택 옵션 활용 시 원하는 컨테이너 정보를 선택해 목록을 조회할 수 있습니다.



- 선택한 컬럼 목록은 메뉴 재진입 시 유지됩니다.
- **초기화** 버튼 선택 시 사전 정의된 기본 주요 지표 컬럼이 선택됩니다. 이 경우 기존 컬럼 선택 내역은 삭제됩니다.
- 컬럼 선택 후 우측 하단 **적용** 버튼을 클릭 해 선택 내역을 적용할 수 있습니다.

컨테이너 타임라인

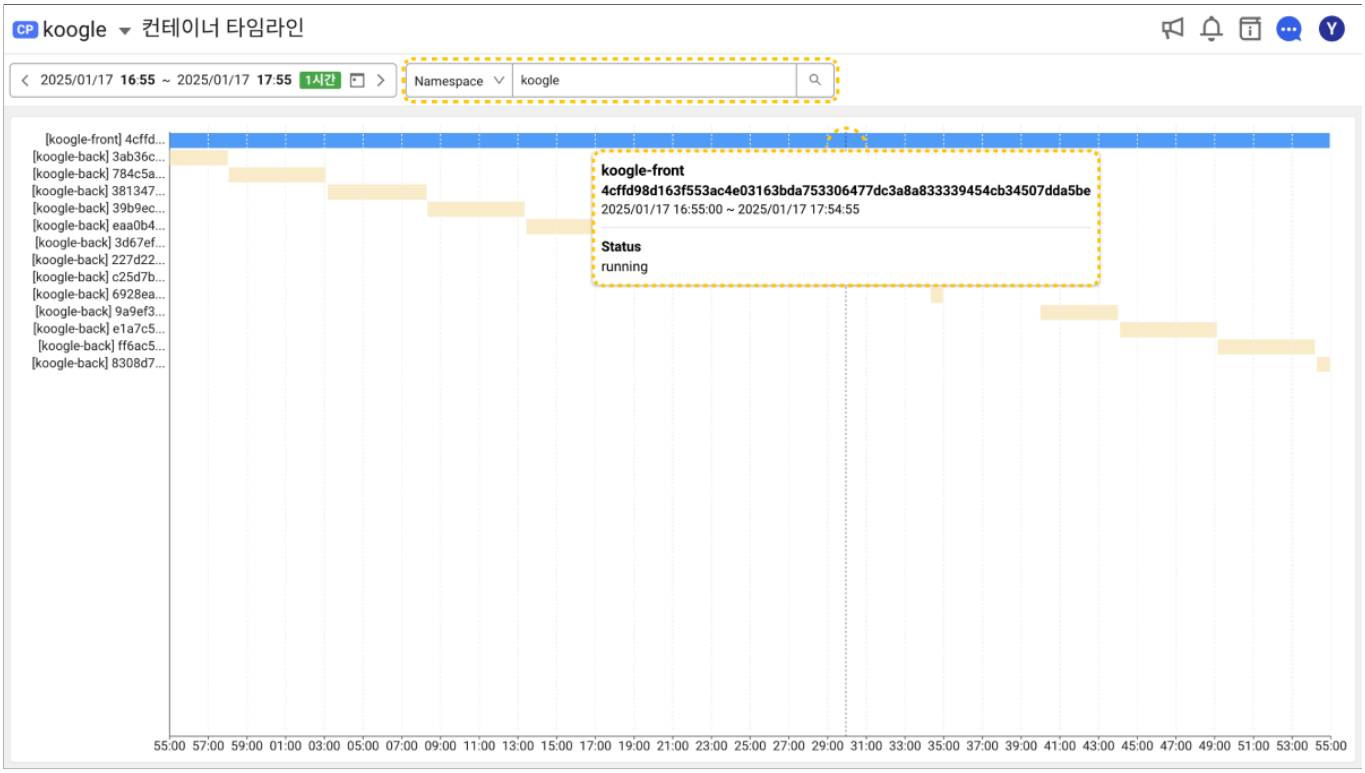
> **요구 사항**
>
> 와탭 쿠버네티스 에이전트 1.7.6 버전 이상이 필요합니다. 타임라인 데이터는 1.7.6 버전 이상에서 수집된 시점부터 제공됩니다.

컨테이너 타임라인 메뉴는 시간의 흐름에 따른 컨테이너의 상태 변화를 시각적으로 추적할 수 있는 기능을 제공합니다. 이미 종료된 컨테이너가 언제 종료되었는지, 특정 시점에 컨테이너가 어떤 상태였는지를 시각적으로 파악할 수 있습니다. 이를 통해 사용자는 과거 상태를 분석하고, 클러스터에서 발생한 이슈의 원인 파악 및 운영 안정성을 모니터링할 수 있습니다.

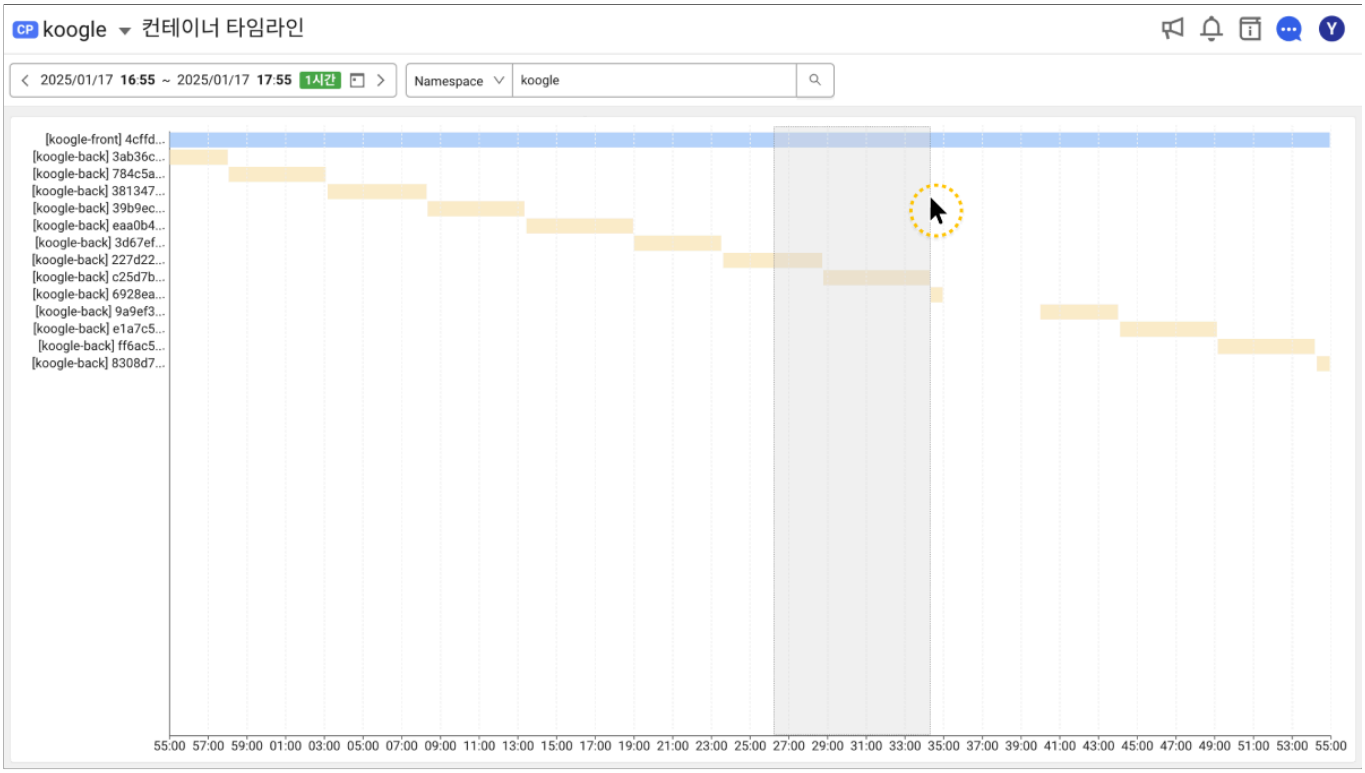
기본 화면 안내

컨테이너 타임라인 메뉴에 진입 후 **필터 키**와 **필터 값**을 선택하여 데이터를 조회하세요. 최근 1시간 동안 수집된 컨테이너 목록을 확인할 수 있습니다. 시간 축을 따라 컨테이너의 상태 변화를 시각적으로 확인할 수 있습니다.

각 시간 구간에 마우스를 올리면 해당 구간의 시작 시간과 종료 시간, 그리고 컨테이너의 상태(**running**, **waiting**, **terminated**)가 표시됩니다.



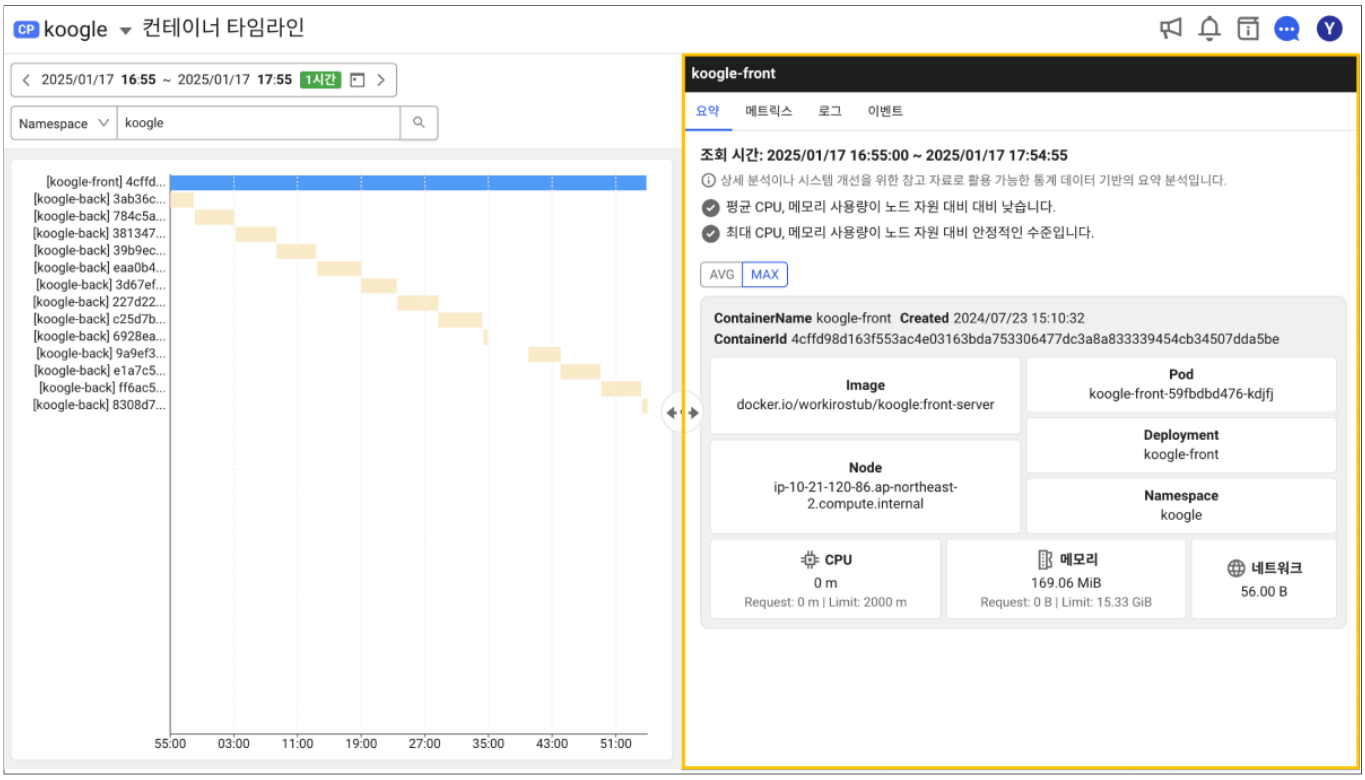
타임라인 상에서 구간을 드래그하여 시간 범위를 조정할 수 있습니다. 원하는 기간을 설정하여 과거 데이터를 확인해 보세요.



또한 특정 기간 선택 시 해당 기간 동안의 컨테이너 상태를 **상세 보기**를 통해 자세히 확인할 수 있습니다. 자세한 내용은 다음 컨테이너 타임라인 상세 안내를 참조하세요.

컨테이너 타임라인 상세

컨테이너 타임라인 상세는 특정 구간 또는 특정 컨테이너를 선택하여 확인할 수 있습니다.



- 세부 정보를 조회하고자 하는 구간을 선택하여 오른쪽 **상세 보기** 창을 통해 해당 시점의 컨테이너에 대한 상세를 조회할 수 있습니다.
- 차트 왼쪽 목록에서 조회하고자 하는 컨테이너 이름을 선택하여 오른쪽 **상세 보기** 창을 통해 해당 컨테이너의 전체 상태를 볼 수 있습니다.

> ⓘ 상세 보기 창의 메트릭스, 이벤트, 로그 탭의 경우 컨테이너 맵 상세 보기와 동일합니다.

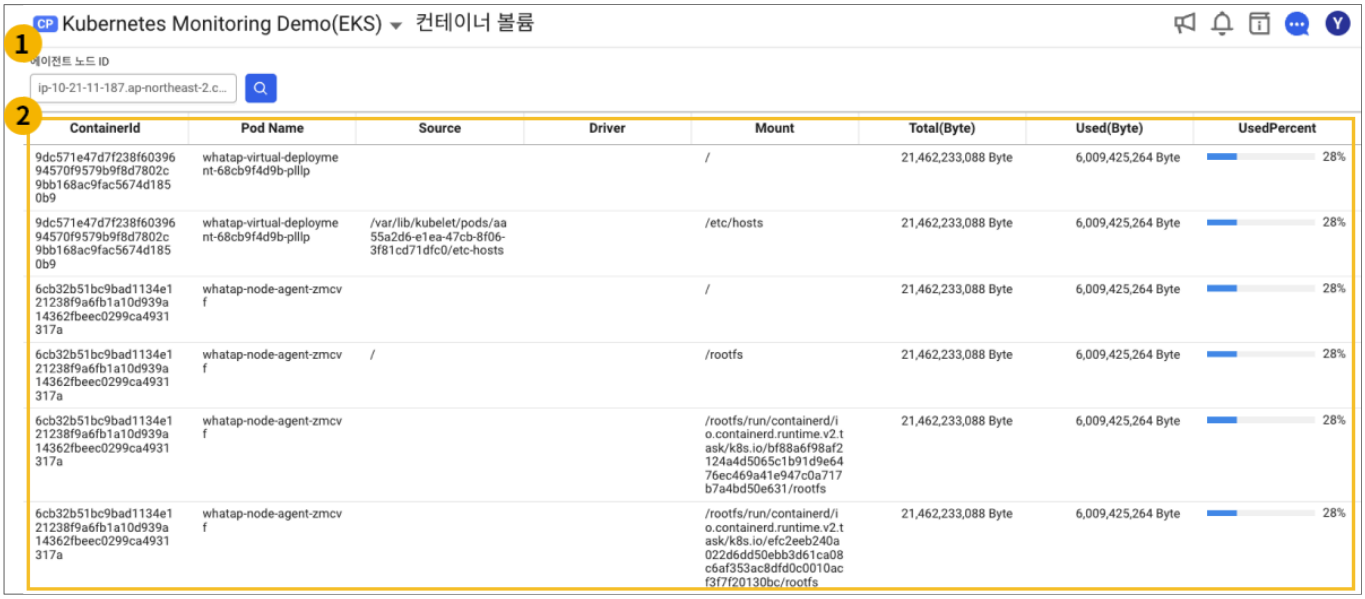
요약

요약 탭에서 선택한 컨테이너 정보 및 전반적인 리소스 사용량 정보를 요약해서 제공합니다. 해당 컨테이너의 이미지, 컨테이너가 실행되고 있는 노드, 컨테이너가 실행되고 있는 Pod의 `ownerReference` 등 컨테이너 정보를 종합적으로 확인할 수 있습니다. 사용자는 컨테이너의 리소스 사용 현황 및 상태를 파악하고 성능 및 운영 안정성을 모니터링할 수 있습니다. 또한 **AVG**, **MAX** 버튼을 클릭하여 조회 기간동안 평균 사용량과 최대 사용량을 확인할 수 있습니다.

컨테이너 볼륨

컨테이너 볼륨 메뉴를 통해 컨테이너에 마운트된 볼륨 및 해당 용량 상태를 확인할 수 있습니다.

기본 화면 안내

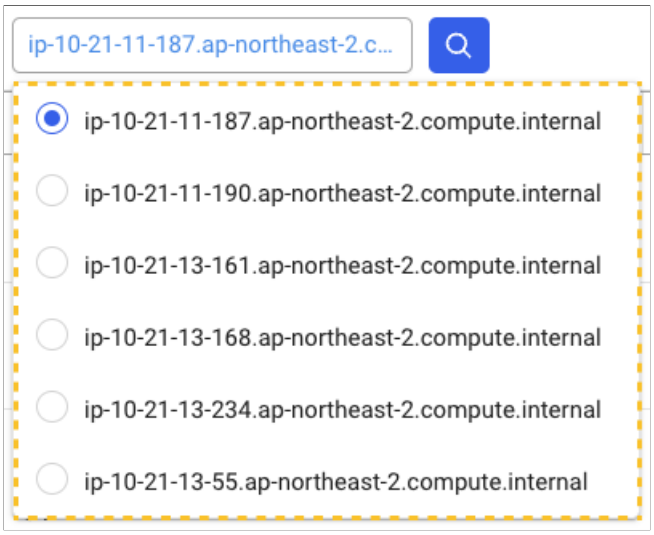


ContainerId	Pod Name	Source	Driver	Mount	Total(Byte)	Used(Byte)	UsedPercent
9dc571e47d7f238f6039694570f9579b9f8d7802c9bb168ac9fac5674d1850b9	whatap-virtual-deployment-68cb9f4d9b-pllljp			/	21,462,233,088 Byte	6,009,425,264 Byte	28%
9dc571e47d7f238f6039694570f9579b9f8d7802c9bb168ac9fac5674d1850b9	whatap-virtual-deployment-68cb9f4d9b-pllljp	/var/lib/kubelet/pods/aa55a2d6-e1ea-47cb-8f06-3f81cd71dfc0/etc-hosts		/etc/hosts	21,462,233,088 Byte	6,009,425,264 Byte	28%
6cb32b51bc9bad1134e121238f9a6fb1a10d939a14362fbeec0299ca4931317a	whatap-node-agent-zmcvf			/	21,462,233,088 Byte	6,009,425,264 Byte	28%
6cb32b51bc9bad1134e121238f9a6fb1a10d939a14362fbeec0299ca4931317a	whatap-node-agent-zmcvf	/		/rootfs	21,462,233,088 Byte	6,009,425,264 Byte	28%
6cb32b51bc9bad1134e121238f9a6fb1a10d939a14362fbeec0299ca4931317a	whatap-node-agent-zmcvf			/rootfs/run/containerd/io.containerd.runtime.v2.task/k8s.io/bf88a6f98af2124a4d5065c1b91d9e6476ec469a41e947c0a717b7a4bd50e631/rootfs	21,462,233,088 Byte	6,009,425,264 Byte	28%
6cb32b51bc9bad1134e121238f9a6fb1a10d939a14362fbeec0299ca4931317a	whatap-node-agent-zmcvf			/rootfs/run/containerd/io.containerd.runtime.v2.task/k8s.io/efc2eeb240a022d6dd50ebb3d61ca08c6af353ac8dfd0c0010acf3f7f20130bc/rootfs	21,462,233,088 Byte	6,009,425,264 Byte	28%

1 영역에서 조회를 원하는 에이전트 노드를 선택 후 2 영역에서 데이터를 확인할 수 있습니다.

노드별 조회

입력창 선택 시 다음과 같이 노드 목록창이 나타납니다. 조회를 원하는 에이전트 노드를 선택 후 **돋보기** 버튼을 클릭하세요.



컨테이너 볼륨 정보

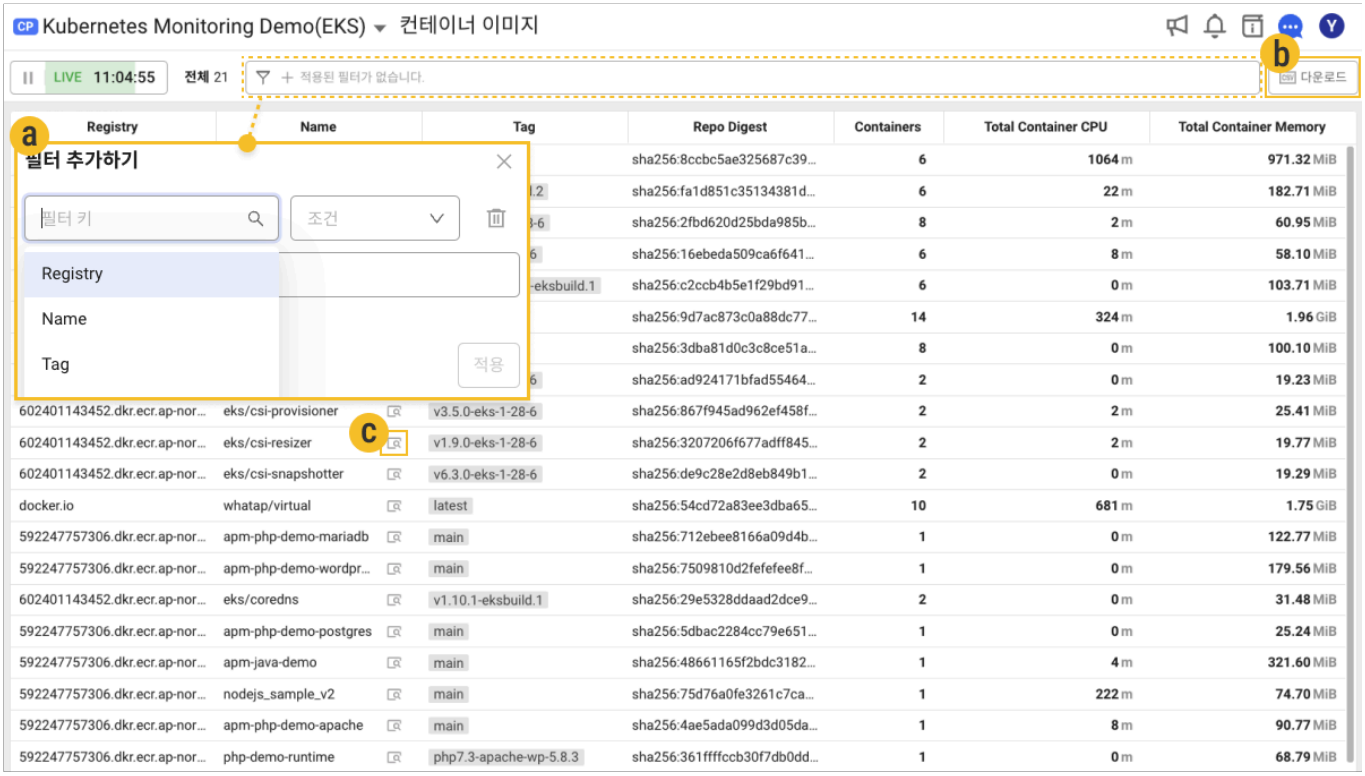
컨테이너 볼륨 메뉴에서 제공하는 정보 항목을 다음과 같이 안내합니다.

항목	설명
ContainerId	컨테이너의 고유 식별자인 컨테이너 ID입니다. 이 ID는 컨테이너 런타임에 의해 생성됩니다. 클러스터 내에서 컨테이너를 식별하고 관리하기 위해 필수적입니다.
Pod Name	컨테이너가 속한 Pod의 이름입니다. 쿠버네티스 클러스터에서는 여러 컨테이너가 하나의 Pod 내에서 실행될 수 있으며, 이 이름을 통해 해당 컨테이너가 어떤 Pod에 속해 있는지 식별할 수 있습니다.
MountType	볼륨이 마운트된 방식을 나타냅니다. 단, `tmpfs` 는 메모리를 사용하는 임시적인 파일 시스템이기 때문에 해당 **컨테이너 볼륨** 정보 목록에 **표시되지 않습니다.**
	- `bind` : 호스트 상 특정 디렉토리가 마운트 된 것을 의미합니다. - `volume` : 호스트 상 Docker가 관리하는 특정 파일 시스템(예: /var/lib/docker/volumes/)이 마운트 된 것을 의미합니다.
Source	호스트 상에서의 해당 볼륨의 경로입니다. 이 경로는 볼륨 데이터가 실제로 저장되는 위치를 의미합니다.
Driver	해당 볼륨 생성 시 사용된 볼륨 드라이버입니다. 볼륨 드라이버는 데이터의 저장 방식과 볼륨의 관리 방법을

항목	설명
	결정합니다. 현재 이 정보는 주로 도커 런타임에서 제공됩니다. containerd와 CRI-O와 같은 더 낮은 수준의 컨테이너 런타임에서는 이러한 고수준 세부 정보를 직접 제공하지 않습니다. 모니터링 대시보드는 런타임 API를 통해 이러한 볼륨 데이터를 제한적으로 제공합니다.
Mount	컨테이너 내에서 볼륨이 마운트된 경로입니다. 이 경로를 통해 컨테이너 내의 애플리케이션이 볼륨에 저장된 데이터에 접근할 수 있습니다.
Total(Byte)	볼륨의 전체 용량을 바이트 단위로 나타냅니다. 이는 볼륨에 저장할 수 있는 데이터의 최대 크기를 의미합니다.
Used(Byte)	현재 볼륨에서 사용 중인 용량을 바이트 단위로 나타냅니다. 이는 볼륨 내에 실제로 저장된 데이터의 크기를 의미합니다.
UsedPercent	볼륨 용량의 사용률을 백분율로 나타냅니다. 이 비율은 볼륨의 전체 용량 대비 현재 사용 중인 용량의 비율을 보여줍니다. 사용률이 높아질수록 볼륨 공간이 부족해지고 있음을 의미합니다. 이는 성능 저하나 데이터 손실 위험을 알리는 중요한 지표가 될 수 있습니다.

컨테이너 이미지

컨테이너 이미지 메뉴를 통해 컨테이너에서 사용하고 있는 이미지 정보와 해당 이미지를 기반으로 실행 중인 컨테이너의 개수 및 CPU, Memory 사용 총량을 목록 형식으로 확인할 수 있습니다.



a 필터

필터 창에서 **필터 키** 선택 시 **Registry**, **Name**, **Tag**, **Repo Digest**에 대한 필터 기능을 제공합니다.

b CSV 다운로드

상단 오른쪽의 **CSV 다운로드** 버튼 선택 시 원본 데이터를 CSV 형식의 파일로 다운로드할 수 있습니다. 파일명은 `kubernetes_container_images_` 뒤에 프로젝트 코드와 데이터 조회 시점의 타임스탬프를 덧붙여 생성합니다.

예, kubernetes_container_images_{pcode}_{timestamp}.csv

C 컨테이너 상세 보기

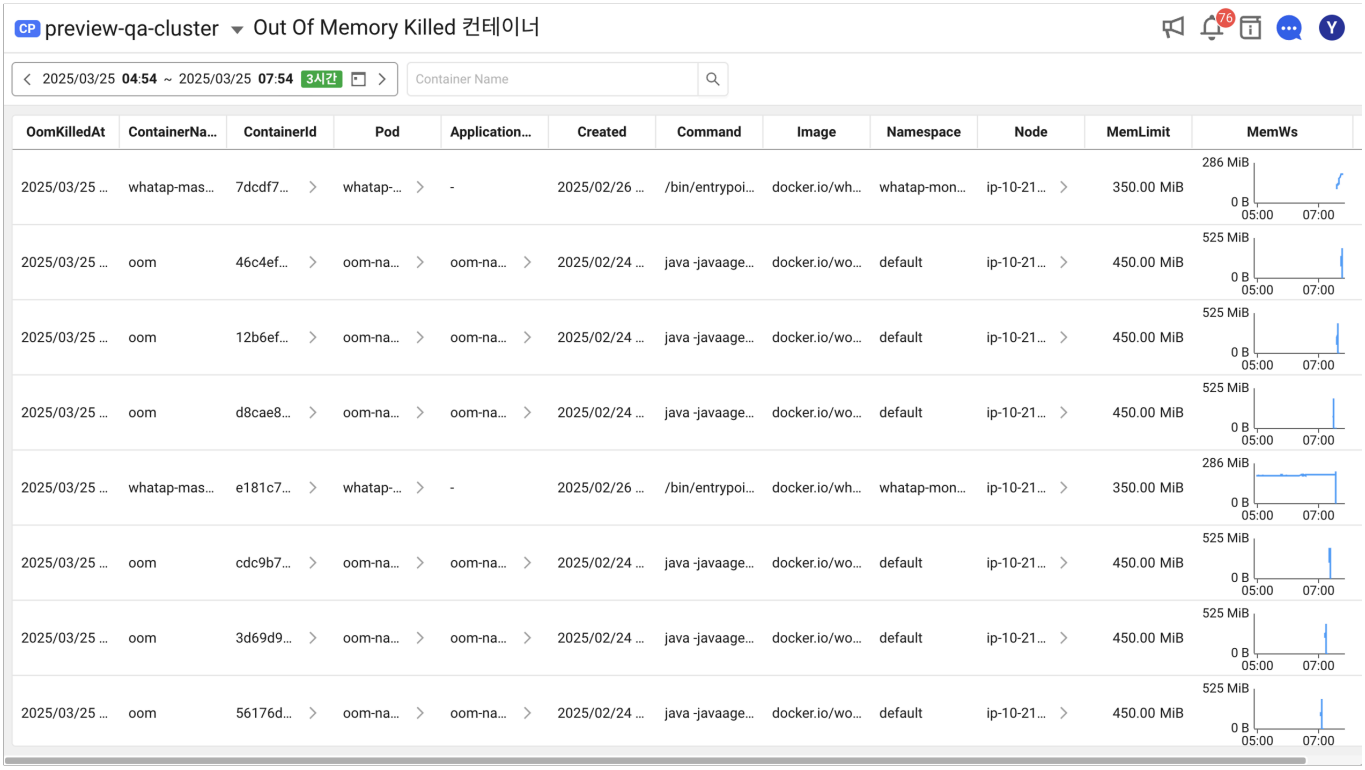
Name 컬럼 오른쪽 **컨테이너 상세 보기** 아이콘 선택 시 해당 이미지가 검색된 **컨테이너 목록** 메뉴로 이동합니다.

Out Of Memory Killed 컨테이너

Out Of Memory Killed 컨테이너 메뉴를 통해 **Out Of Memory(OOM)**가 발생한 컨테이너를 빠르게 검색하고 해당 시점의 메모리 사용량의 변화 추이를 확인할 수 있습니다. 변화 추이는 급격하거나 점진적일 수 있습니다. 이를 확인해 적절한 POD 수와 메모리 임계점에 대한 객관적 판단이 가능합니다.

> **OOM 발생 예시**
>
> 컨테이너의 메모리 사용량이 메모리 제한을 초과하거나, 컨테이너가 실행 중인 노드의 메모리가 부족할 경우 Out Of Memory가 발생합니다. 만약 컨테이너가 코드 `137` 로 종료되면 Out Of Memory에 의해 `SIGKILL` 을 수신한 것입니다.

기본 화면 안내



- 왼쪽 상단의 시간 선택자를 클릭해 조회를 원하는 시간대를 선택할 수 있습니다. 최대 2일까지 조회할 수 있습니다.

- 오른쪽 상단의 검색 입력창에서 컨테이너 이름을 입력하여 조회를 원하는 특정 컨테이너 정보를 빠르게 찾을 수 있습니다.
- 목록 오른쪽 상세 아이콘 클릭 시 선택한 컨테이너의 메모리 사용 추이를 확인할 수 있습니다.
- OOM이 발생한 컨테이너 정보를 목록 형식으로 제공합니다. 컨테이너가 자세한 내용은 다음 안내를 참조하세요.

OOM Killed 컨테이너 정보

Out Of Memory Killed 컨테이너 메뉴에서 제공하는 컬럼 정보를 다음과 같이 안내합니다.

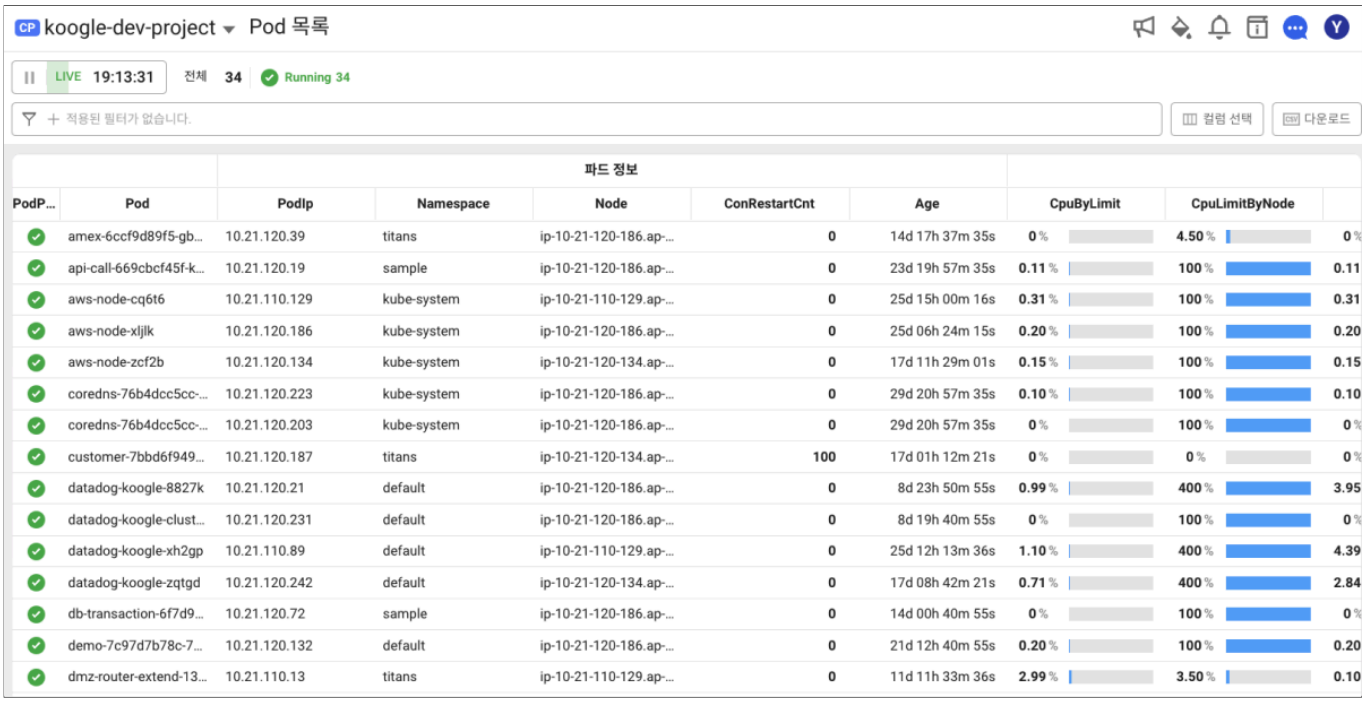
컬럼	의미
OomKilledAt	Out Of Memory Killed가 발생된 TimeStamp
ContainerName	컨테이너의 이름
ContainerId	컨테이너의 ID
Pod	컨테이너가 속한 Pod의 이름
ApplicationName	컨테이너에서 실행중인 애플리케이션 이름
Created	컨테이너가 생성된 TimeStamp
Command	컨테이너의 실행 명령어
Image	컨테이너의 이미지의 이름
Namespace	컨테이너의 Pod가 속한 네임스페이스의 이름
Node	컨테이너가 작동 중인 노드 이름
MemLimit	컨테이너의 메모리 Limit 크기
MemWs	컨테이너에서 비활성화 된 메모리를 제외한 실제 사용 중인 메모리

ⓘ • **ContainerId** 클릭 시 컨테이너 목록 메뉴로 이동합니다.

- **Pod** 클릭 시 Pod 목록 메뉴로 이동합니다.
- **ApplicationName** 클릭 시 해당 애플리케이션이 검색된 컨테이너 애플리케이션 목록 메뉴로 이동합니다.
- **Node** 클릭 시 노드 목록 메뉴로 이동합니다.

Pod 목록

Pod 목록을 조회합니다. 원하는 컬럼 선택 시 해당 컬럼 기준으로 Pod 목록을 정렬해 조회할 수 있습니다.



	파드 정보							
PodP...	Pod	PodIp	Namespace	Node	ConRestartCnt	Age	CpuByLimit	CpuLimitByNode
	amex-6ccf9d89f5-gb...	10.21.120.39	titans	ip-10-21-120-186.ap-...	0	14d 17h 37m 35s	0 %	4.50 % 0 %
	api-call-669cbcf45f-k...	10.21.120.19	sample	ip-10-21-120-186.ap-...	0	23d 19h 57m 35s	0.11 %	100 % 0.11
	aws-node-cq6t6	10.21.110.129	kube-system	ip-10-21-110-129.ap-...	0	25d 15h 00m 16s	0.31 %	100 % 0.31
	aws-node-xljlk	10.21.120.186	kube-system	ip-10-21-120-186.ap-...	0	25d 06h 24m 15s	0.20 %	100 % 0.20
	aws-node-zcf2b	10.21.120.134	kube-system	ip-10-21-120-134.ap-...	0	17d 11h 29m 01s	0.15 %	100 % 0.15
	coredns-76b4dcc5cc-...	10.21.120.223	kube-system	ip-10-21-120-186.ap-...	0	29d 20h 57m 35s	0.10 %	100 % 0.10
	coredns-76b4dcc5cc-...	10.21.120.203	kube-system	ip-10-21-120-186.ap-...	0	29d 20h 57m 35s	0 %	100 % 0 %
	customer-7bbd6f949...	10.21.120.187	titans	ip-10-21-120-134.ap-...	100	17d 01h 12m 21s	0 %	0 % 0 %
	datadog-koogle-8827k	10.21.120.21	default	ip-10-21-120-186.ap-...	0	8d 23h 50m 55s	0.99 %	400 % 3.95
	datadog-koogle-clust...	10.21.120.231	default	ip-10-21-120-186.ap-...	0	8d 19h 40m 55s	0 %	100 % 0 %
	datadog-koogle-xh2gp	10.21.110.89	default	ip-10-21-110-129.ap-...	0	25d 12h 13m 36s	1.10 %	400 % 4.39
	datadog-koogle-zqtgd	10.21.120.242	default	ip-10-21-120-134.ap-...	0	17d 08h 42m 21s	0.71 %	400 % 2.84
	db-transaction-6f7d9...	10.21.120.72	sample	ip-10-21-120-186.ap-...	0	14d 00h 40m 55s	0 %	100 % 0 %
	demo-7c97d7b78c-7...	10.21.120.132	default	ip-10-21-120-186.ap-...	0	21d 12h 40m 55s	0.20 %	100 % 0.20
	dmz-router-extend-13...	10.21.110.13	titans	ip-10-21-110-129.ap-...	0	11d 11h 33m 36s	2.99 %	3.50 % 0.10

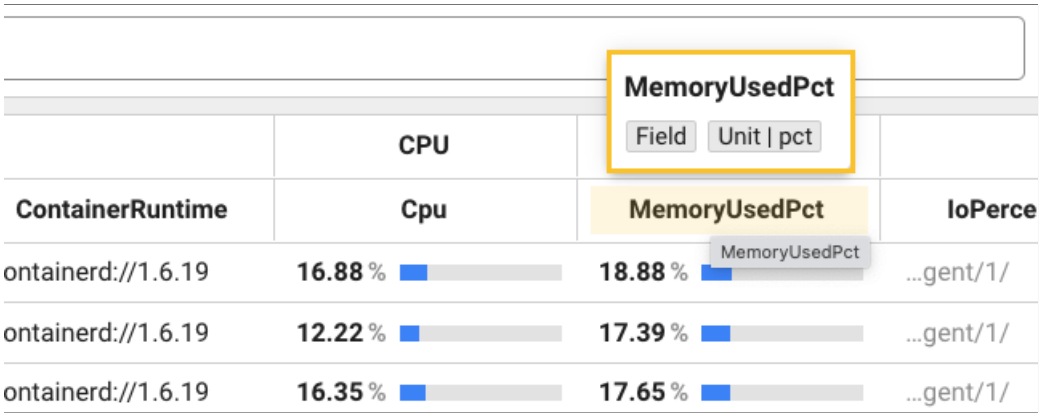
- 10초 주기로 실시간 데이터를 업데이트합니다.
- 좌측 상단에서 전체 Pod 개수와 상태별 Pod 개수를 확인할 수 있습니다.
- 검색, 컬럼 정렬, 상세 보기, CSV 파일 다운로드를 제공합니다.

  CSV 파일명, kubernetes_pod_list_ `{pcode}` _ `{timestamp}` .csv

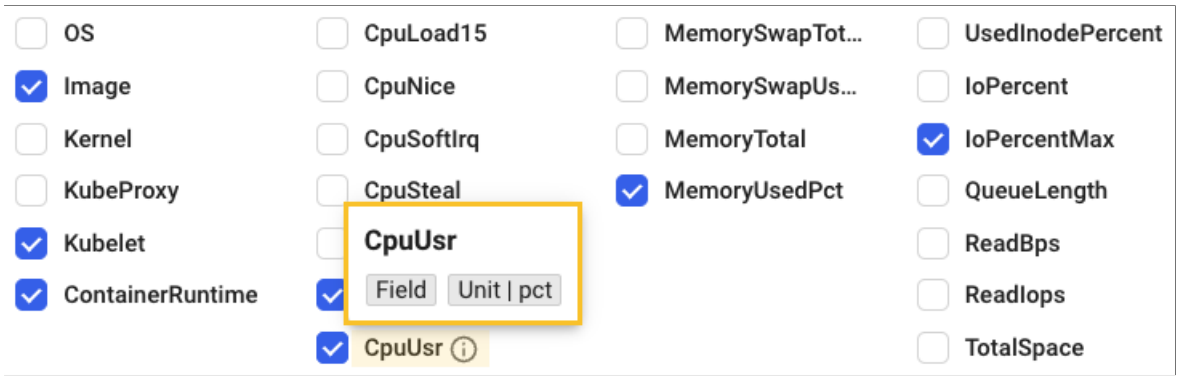
> ⓘ 노드 목록과 컨테이너 목록 및 Pod 목록과 컨테이너 애플리케이션 목록 등 목록형 메뉴의 **메트릭 정보**, **필터**, **컬럼 선택** 안내는 동일합니다.

메트릭 정보

- 원하는 컬럼 위에 마우스 커서를 위치시키면 다음과 같이 해당 컬럼이 제공하는 메트릭 정보를 확인할 수 있습니다.



- **컬럼 선택** 옵션 화면 내 컬럼 목록에서 컬럼별 메트릭 정보를 확인할 수 있습니다. 원하는 컬럼 위에 마우스 커서를 위치시키면 컬럼 이름 우측에 ⓘ **인포 아이콘**이 나타납니다. ⓘ **인포 아이콘** 위에 마우스 커서를 위치시키면 다음과 같이 메트릭 정보를 확인할 수 있습니다.



필터

필터 옵션 활용 시 원하는 Pod 목록을 선택해 조회할 수 있습니다. `and` 조건을 통해 여러 개의 필터를 적용할 수 있습니다.

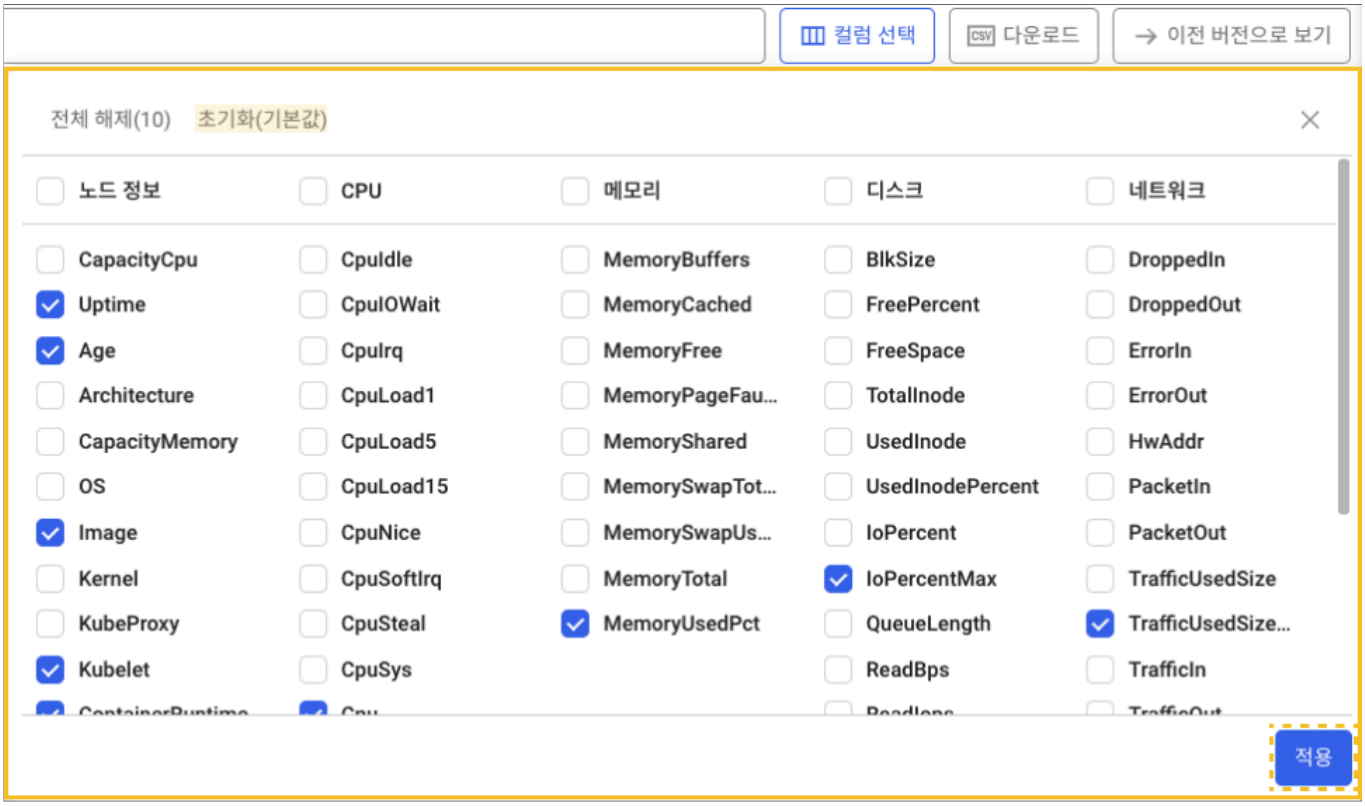
1. **필터** 입력 창에서 + **더하기 아이콘** 선택 시 다음과 같이 **필터 추가하기** 창이 나타납니다.

- **필터 키**와 **조건**을 선택하세요.
- 조건에 맞는 값을 선택하세요.

2. 필터 설정 후 **적용** 버튼을 선택해 해당 필터를 적용하세요.
3. 여러 개의 필터를 추가하려면 **필터** 입력 창 또는 **필터 추가하기** 창에서 ＋ **더하기 아이콘**을 선택하세요.

컬럼 선택

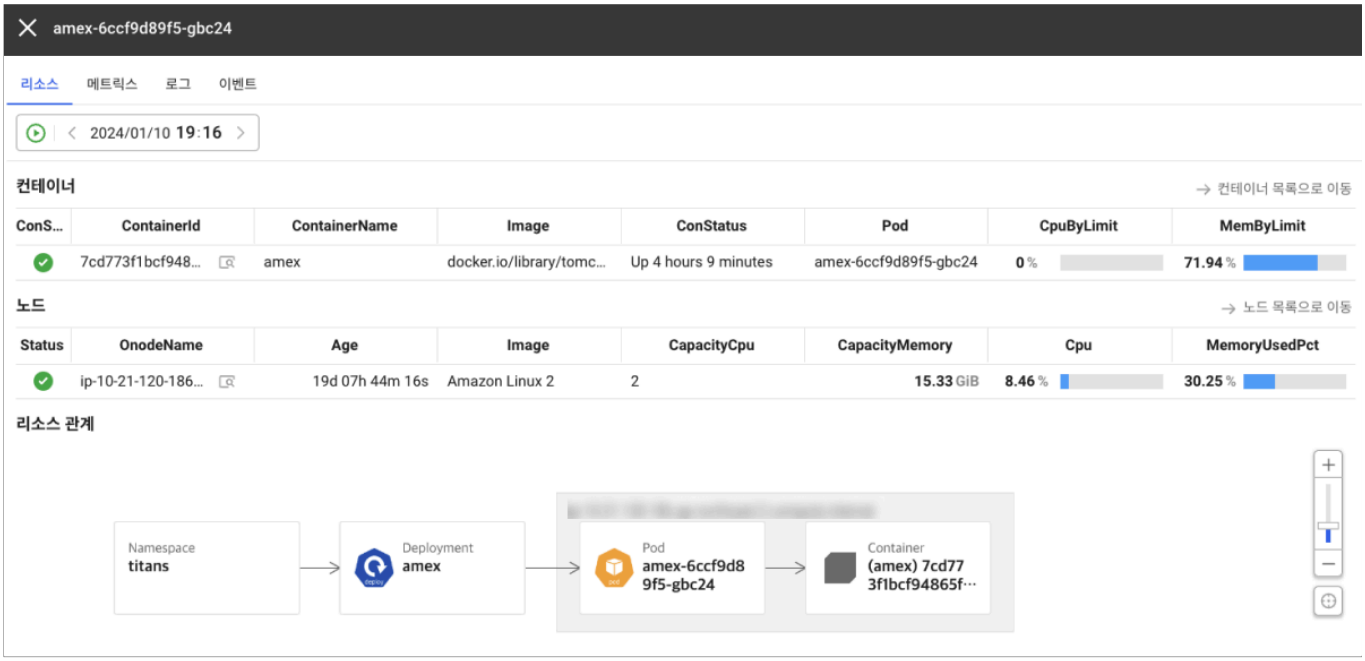
컬럼 선택 옵션 활용 시 원하는 Pod 정보를 선택해 목록을 조회할 수 있습니다.



- 선택한 컬럼 목록은 메뉴 재진입 시 유지됩니다.
- **초기화** 버튼 선택 시 사전 정의된 기본 주요 지표 컬럼이 선택됩니다. 이 경우 기존 컬럼 선택 내역은 삭제됩니다.
- 컬럼 선택 후 우측 하단 **적용** 버튼을 클릭 해 선택 내역을 적용할 수 있습니다.

상세 보기

목록에서 조회를 원하는 Pod의 🔍 **상세 보기** 아이콘 선택 시 해당 Pod에 대한 상세 정보를 확인할 수 있습니다.



> ⓘ 상세 보기 창의 트레이스, 메트릭스, 로그, 이벤트, Pod 요약 분석 탭의 경우 컨테이너 맵 상세 보기와 동일합니다.

리소스

대상 선택 후 상세 보기의 **리소스** 탭에서 선택한 리소스와 관련된 리소스 목록 및 관계도를 조회할 수 있습니다.

요약 정보

선택한 리소스와 관련된 리소스 목록의 요약 정보를 제공합니다. 리소스 명 옆에 상세보기 아이콘을 선택 시 해당 리소스 목록 화면으로 이동합니다.

- **컨테이너**: 해당 컨테이너만 필터링된 상태의 **컨테이너 목록**으로 이동합니다.
- **Pod**: 해당 Pod만 필터링된 **Pod 목록**으로 이동합니다.
- **노드**: 해당 노드 이름이 선택된 상태의 **노드 상세**로 이동합니다.

리소스 관계도

리소스간 연결 관계와 상태를 시각적으로 확인할 수 있습니다. 목록과 동일하게 Pod, 컨테이너, 애플리케이션에 마우스 오버 시

상세 지표를 확인할 수 있습니다. 리소스 선택 시 해당 리소스 목록 페이지로 이동합니다.

Pod 요약 분석

- 기본 정보: Pod의 클러스터 이름, 네임스페이스, 기동 시간, 현재 상태(Phase) 등을 확인할 수 있습니다.
- 쿠버네티스 이벤트: 최근 파드와 관련된 이벤트 로그를 통해 문제가 발생했는지 확인할 수 있습니다.
- 실행 주체: 파드를 생성한 컨트롤러(DaemonSet, Deployment 등)를 식별합니다.
- 노드 정보: 파드가 실행 중인 노드의 운영체제, 커널 버전, CPU, 메모리 등의 상세 사양을 제공합니다.
- 리소스 사용 현황: 파드 내 모든 컨테이너의 CPU와 메모리 사용량, 설정된 리소스 한도(Limit)와 비교한 사용률을 보여줍니다.
- 서비스 네트워킹 정보: 연결된 Service 및 Ingress 정보를 확인할 수 있습니다.

> ⓘ 현재 요약 분석 기능은 국문으로만 제공하고 있습니다.

Pod 타임라인

홈 화면 > 프로젝트 선택 > **워크로드** > **Pod 타임라인**

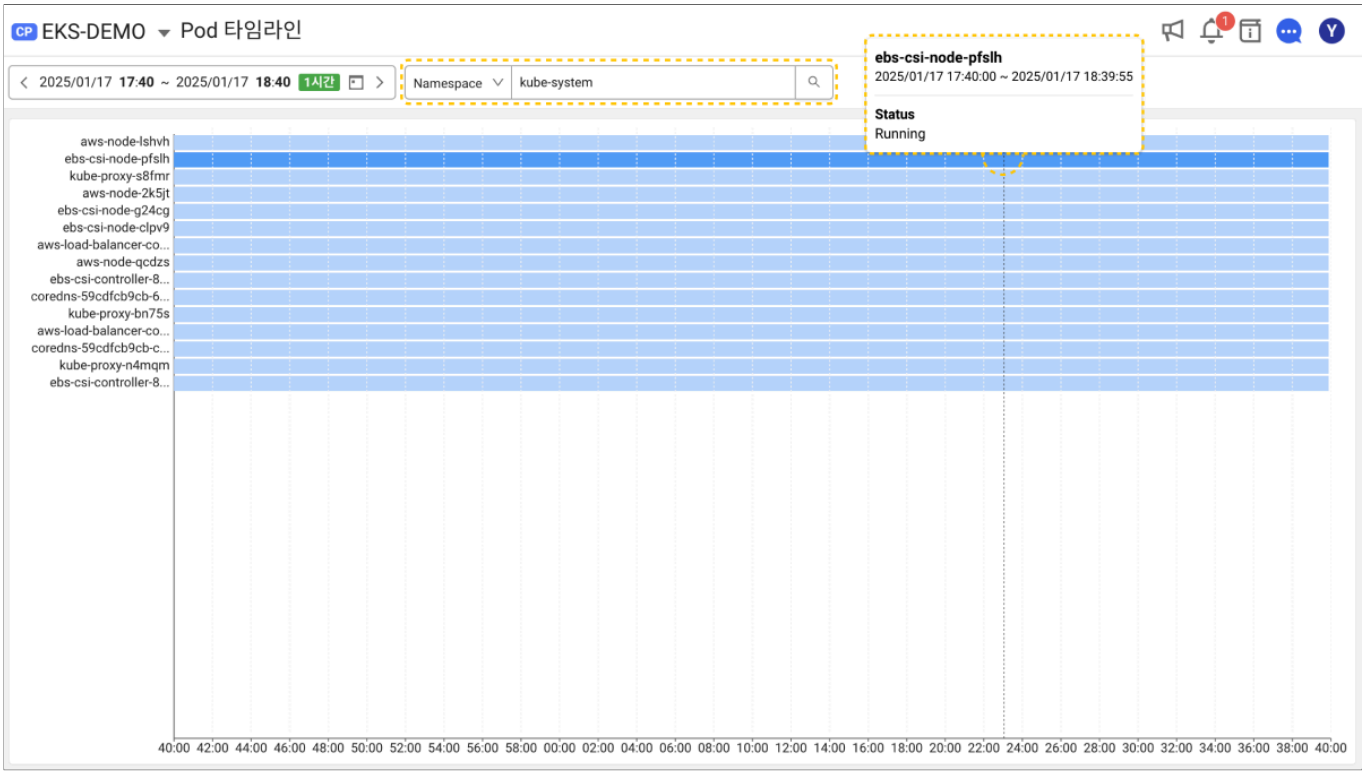
> **요구 사항**
>
> 와탭 쿠버네티스 에이전트 1.7.6 버전 이상이 필요합니다. 타임라인 데이터는 1.7.6 버전 이상에서 수집된 시점부터 제공됩니다.

Pod 타임라인 메뉴는 시간의 흐름에 따른 Pod의 상태 변화를 시각적으로 추적할 수 있는 기능을 제공합니다. 이미 종료된 Pod가 언제 종료되었는지, 특정 시점에 Pod가 어떤 상태였는지를 시각적으로 파악할 수 있습니다. 이를 통해 사용자는 과거 상태를 분석하고, 클러스터에서 발생한 이슈의 원인 파악 및 운영 안정성을 모니터링할 수 있습니다.

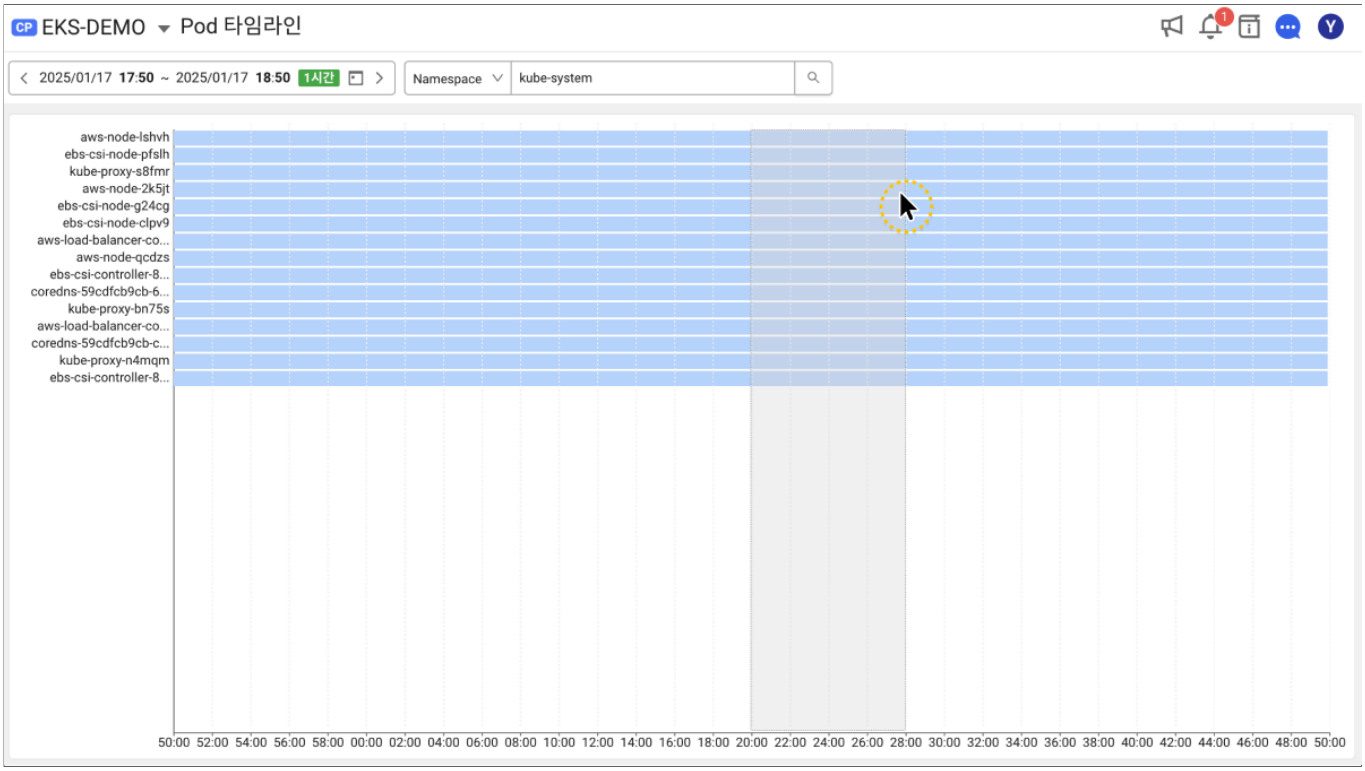
기본 화면 안내

Pod 타임라인 메뉴에 진입 후 **필터 키**와 **필터 값**을 선택하여 데이터를 조회하세요. 최근 1시간 동안 수집된 Pod 목록을 확인할 수 있습니다. 시간 축을 따라 Pod의 상태 변화를 시각적으로 확인할 수 있습니다.

각 시간 구간에 마우스를 올리면 해당 구간의 시작 시간과 종료 시간, 그리고 Pod의 상태(**Running**, **Pending**, **Succeeded**, **Failed**)가 표시됩니다.



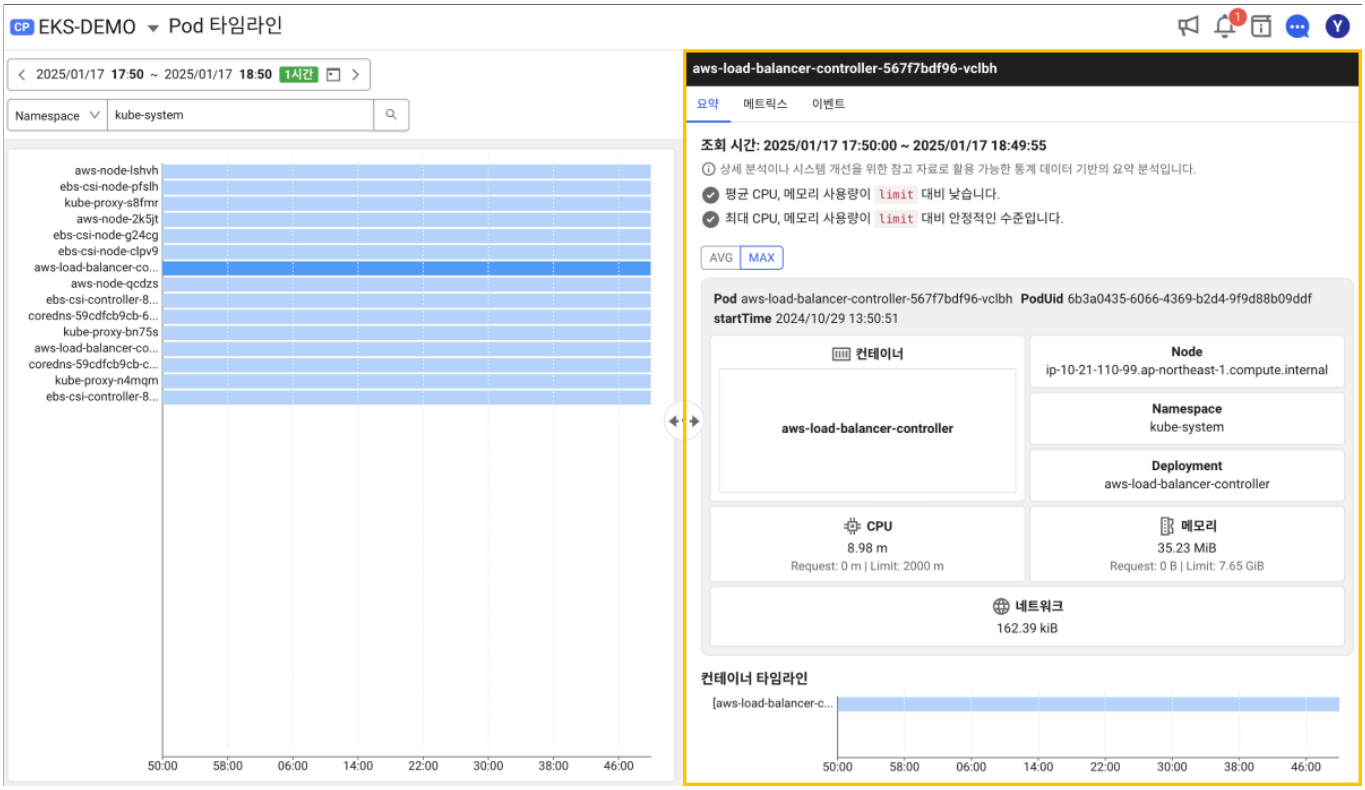
타임라인 상에서 구간을 드래그하여 시간 범위를 조정할 수 있습니다. 원하는 기간을 설정하여 과거 데이터를 확인해 보세요.



또한 특정 기간 선택 시 해당 기간 동안의 Pod 상태를 **상세 보기**를 통해 자세히 확인할 수 있습니다. 자세한 내용은 다음 Pod 타임라인 상세 안내를 참조하세요.

Pod 타임라인 상세

Pod 타임라인 상세는 특정 구간 또는 특정 Pod를 선택하여 확인할 수 있습니다.



- 세부 정보를 조회하고자 하는 구간을 선택하여 오른쪽 **상세 보기** 창을 통해 해당 시점의 Pod에 대한 상세를 조회할 수 있습니다.
- 차트 왼쪽 목록에서 조회하고자 하는 Pod 이름을 선택하여 오른쪽 **상세 보기** 창을 통해 해당 Pod의 전체 상태를 볼 수 있습니다.

> ⓘ 상세 보기 창의 메트릭스, 이벤트 탭의 경우 컨테이너 맵 상세 보기와 동일합니다.

요약

요약 탭에서 선택한 Pod 정보 및 전반적인 리소스 사용량 정보를 요약해서 제공합니다. 해당 Pod에서 실행중인 컨테이너, Pod가 실행되고 있는 노드, `ownerReference` 등 Pod 정보를 종합적으로 확인할 수 있습니다. 사용자는 Pod의 리소스 사용 현황 및 상태를 파악하고 성능 및 운영 안정성을 모니터링할 수 있습니다.

- **AVG**, **MAX** 버튼을 클릭하여 조회 기간동안 평균 사용량과 최대 사용량을 확인할 수 있습니다.
- Pod 내에서 실행된 컨테이너의 타임라인을 제공합니다. 컨테이너 타임라인을 선택 시 해당 타임라인이 선택된 **컨테이너**

타임라인 메뉴가 새 탭으로 열립니다.

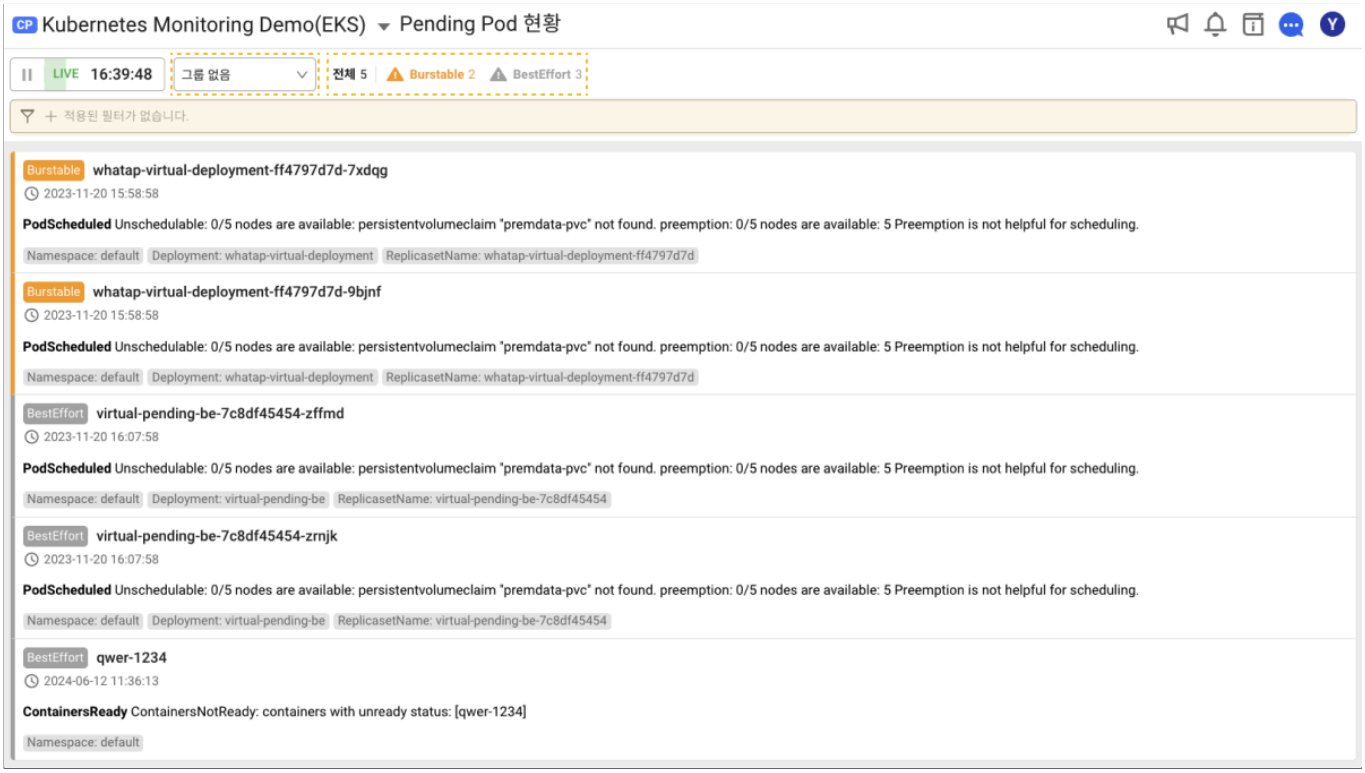
Pending Pod 현황

> **요구 사항**
>
> 와탭 쿠버네티스 에이전트 1.7.0 버전 이상이 필요합니다.

Pending 상태의 Pod 현황을 조회하여 각 Pod의 대기 원인을 종합적으로 분석할 수 있습니다. 클러스터 내에 많은 Pending Pod가 있을수록 시스템의 효율성과 안정성이 저하될 수 있습니다. Pending 상태의 Pod를 모니터링해 이러한 문제를 조기에 발견하고 해결함으로써 클러스터의 안정성을 유지하세요.

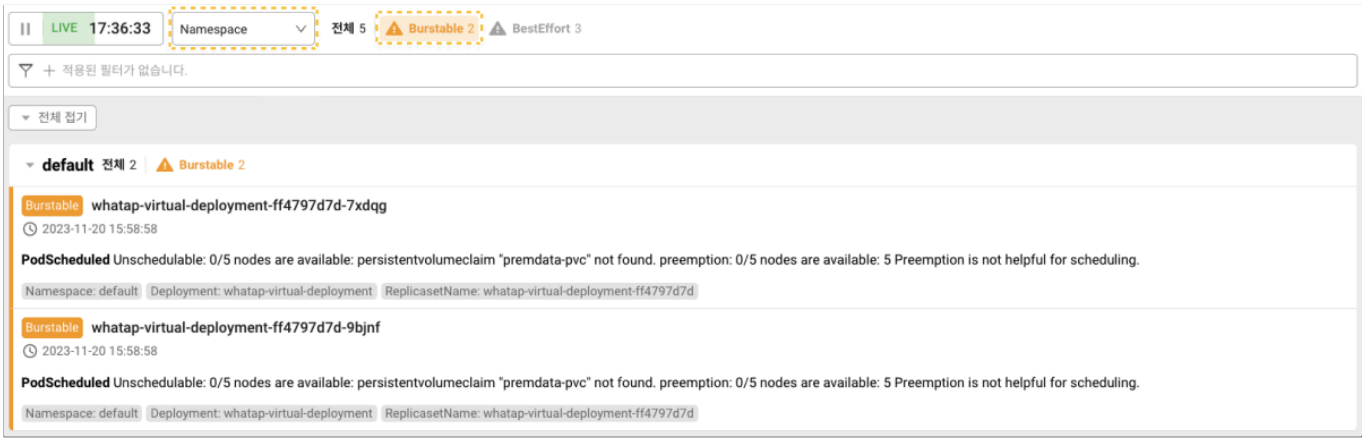
Pending Pod 목록

현재 Pending 상태인 Pod 목록을 다음과 같이 조회할 수 있습니다. 각 Pending Pod마다 QoS Class, Pod 이름, 타임스탬프, 이벤트 메시지, 네임스페이스 및 해당 Pod의 배포 정보를 간략히 확인할 수 있습니다. Pod 선택 시 Pending Pod 상세를 조회할 수 있습니다. 자세한 내용은 다음 문서를 참조하세요.



그룹화와 QoS Class 조회

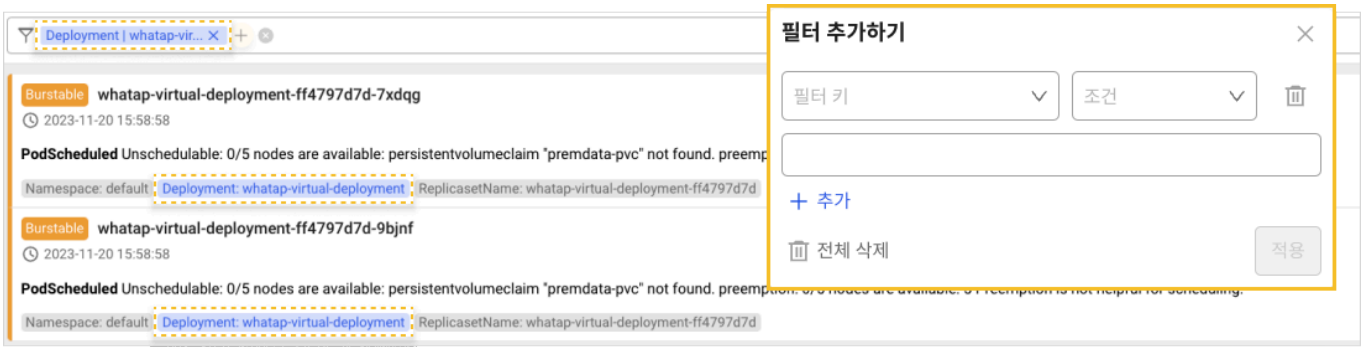
오브젝트별 **그룹화** 기능을 제공합니다. 특정 네임스페이스, 노드 또는 레플리카셋에 관련된 모든 Peding Pod의 상태 정보를 쉽게 확인해 문제 식별 및 분석에 용이합니다. 또한 **QoS Class별 Pending 상태의 Pod 개수**를 확인할 수 있습니다. QoS Class 선택 시 해당 QoS Class에 속하는 Pod를 목록에서 조회할 수 있습니다.



> **QoS Class**(Quality of Service Class)에 대한 자세한 내용은 다음 문서를 참조하세요.

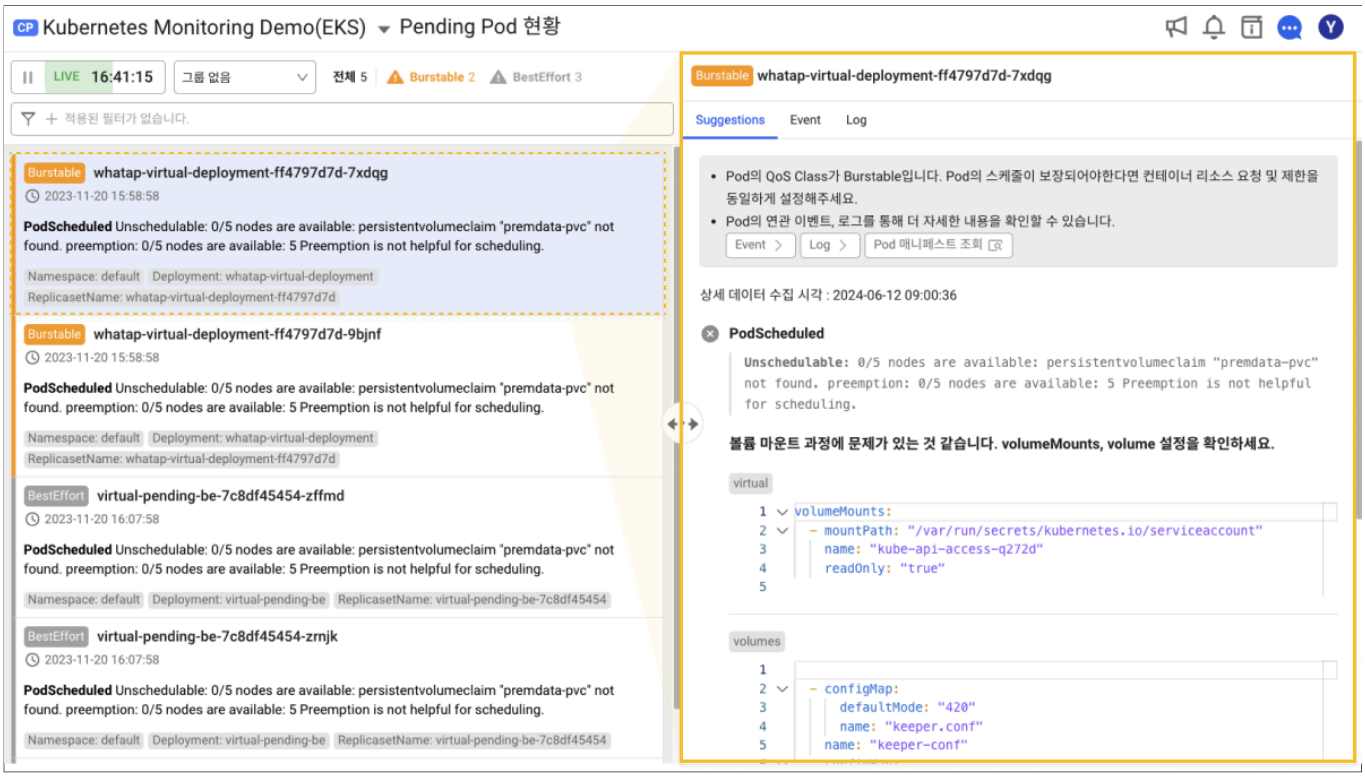
필터

필터 기능을 통해 추가적인 조건별 검색이 가능합니다. 상단 필터 선택창 외에 목록의 각 Pending Pod 하단의 태그를 통해 필터 기능을 활용할 수 있습니다. 태그 선택 시 해당 오브젝트에 대한 일치 조건이 추가되고 조건에 맞는 태그는 파란색으로 표시됩니다.



Pending Pod 상세

Pending Pod 목록에서 Pod 선택 시 해당 Pod에 대한 상세 정보를 **Suggestions**, **Event**, **Log** 탭을 통해 다음과 같이 확인할 수 있습니다.



Suggestions

현재 Pod가 Pending 상태에 있는 원인과 문제 해결을 위한 다양한 정보를 다음과 같이 제공합니다.

- **QoS Class 분석**

  QoS Class를 분석해 스케줄링 권장 사항을 확인할 수 있습니다.

- **상세 데이터 수집 시각**

  해당 상세 정보가 수집된 시간을 확인할 수 있습니다.

- **Pending Pod 이벤트 메시지 및 문제 해결 제안**

이벤트 메시지를 확인하고 이를 기반으로 문제 식별 및 해결책을 제안합니다.

- **문제 관련 정보**

  발생한 문제와 관련한 정보를 제공합니다. 예를 들어 볼륨 마운트 과정에 문제가 있다면 볼륨 관련 설정을 안내하고, 노드 리소스에 문제가 있다면 해당 리소스를 확인할 수 있는 관련 정보를 안내합니다.

Event

선택한 Pod에서 실행 중인 컨테이너의 이벤트 정보를 제공합니다. **컨테이너 맵**의 **이벤트** 탭과 동일합니다. 자세한 내용은 다음 문서를 참조하세요.

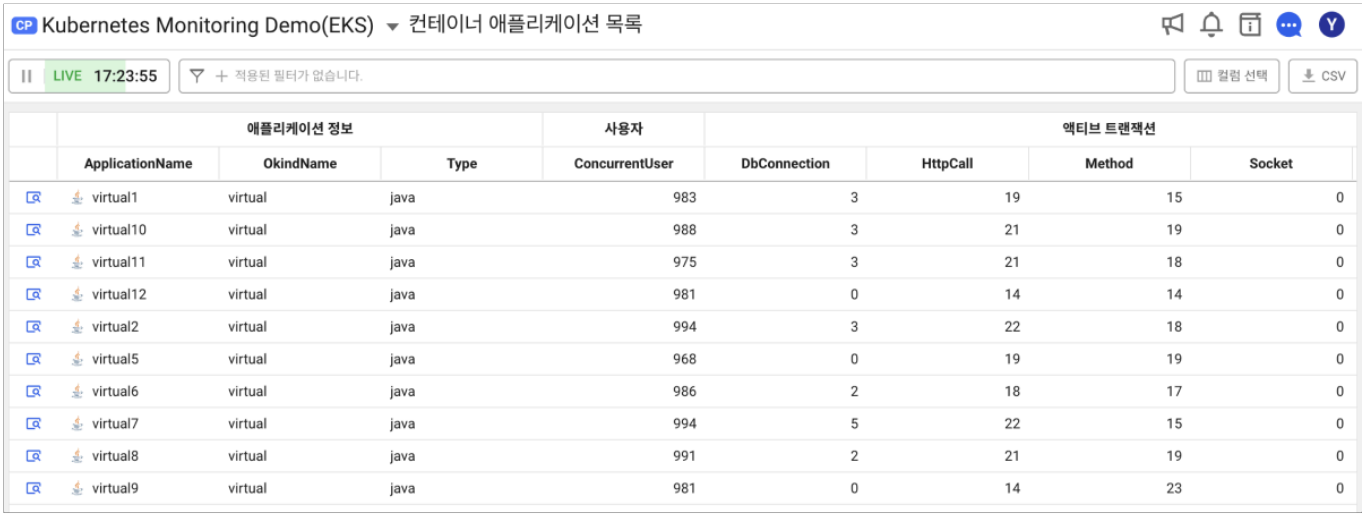
Log

선택한 Pod에서 실행 중인 컨테이너의 로그 정보를 제공합니다. **컨테이너 맵**의 **Log** 탭과 동일합니다. 자세한 내용은 다음 문서를 참조하세요.

> ⓘ 컨테이너 로그는 컨테이너가 실행된 이후에만 확인할 수 있습니다. 만일 Pod가 컨테이너 실행 전 Pending 상태에 머물러 있다면 해당 Pod의 컨테이너 로그를 확인하는 것이 불가능합니다.

컨테이너 애플리케이션 목록

워크로드 > **컨테이너 애플리케이션 목록**에서 컨테이너화된 애플리케이션의 목록을 조회합니다. 원하는 컬럼 선택 시 해당 컬럼 기준으로 애플리케이션 목록을 정렬해 조회할 수 있습니다.



애플리케이션 정보			사용자	액티브 트랜잭션			
ApplicationName	OkindName	Type	ConcurrentUser	DbConnection	HttpCall	Method	Socket
virtual1	virtual	java	983	3	19	15	0
virtual10	virtual	java	988	3	21	19	0
virtual11	virtual	java	975	3	21	18	0
virtual12	virtual	java	981	0	14	14	0
virtual2	virtual	java	994	3	22	18	0
virtual5	virtual	java	968	0	19	19	0
virtual6	virtual	java	986	2	18	17	0
virtual7	virtual	java	994	5	22	15	0
virtual8	virtual	java	991	2	21	19	0
virtual9	virtual	java	981	0	14	23	0

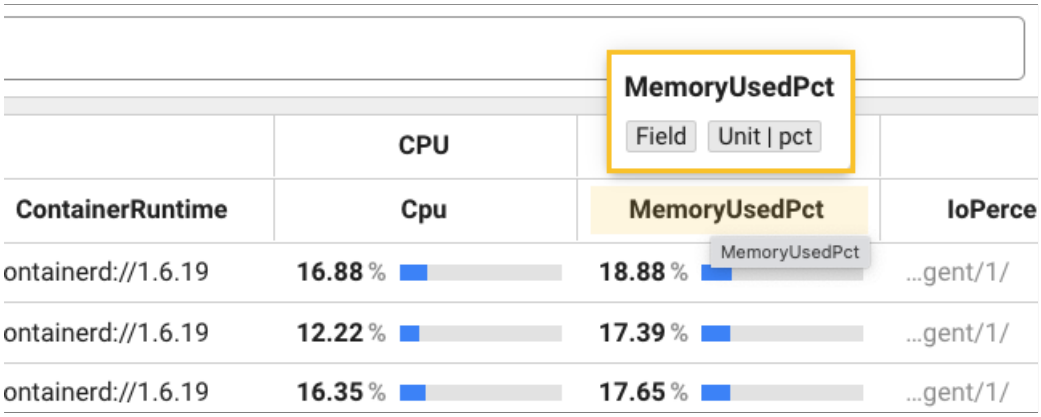
- 10초 주기로 실시간 데이터를 업데이트합니다.
- 검색, 컬럼 정렬, 상세 보기, CSV 파일 다운로드를 제공합니다.

  CSV 파일명, kubernetes_application_list_ `{pcode}` _ `{timestamp}` .csv

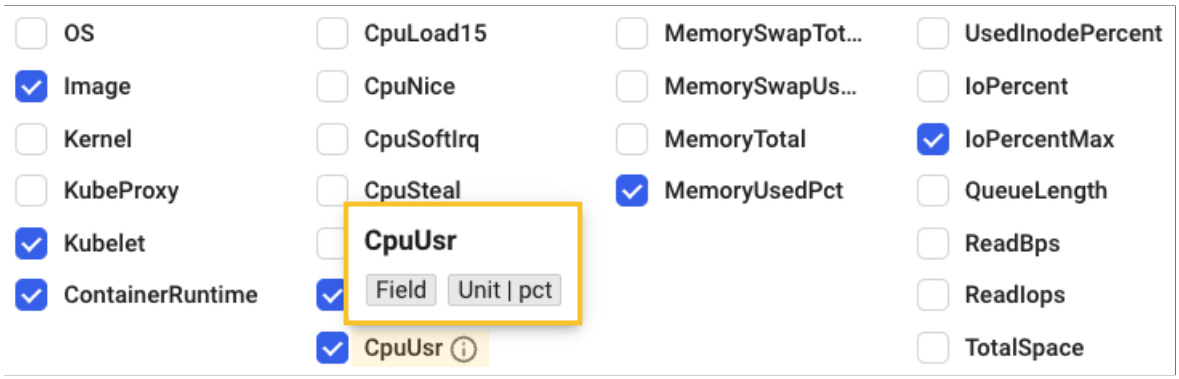
> ⓘ 노드 목록과 컨테이너 목록 및 Pod 목록과 컨테이너 애플리케이션 목록 등 목록형 메뉴의 **메트릭 정보**, **필터**, **컬럼 선택** 안내는 동일합니다.

메트릭 정보

- 원하는 컬럼 위에 마우스 커서를 위치시키면 다음과 같이 해당 컬럼이 제공하는 메트릭 정보를 확인할 수 있습니다.



- **컬럼 선택** 옵션 화면 내 컬럼 목록에서 컬럼별 메트릭 정보를 확인할 수 있습니다. 원하는 컬럼 위에 마우스 커서를 위치시키면 컬럼 이름 우측에 ⓘ **인포 아이콘**이 나타납니다. ⓘ **인포 아이콘** 위에 마우스 커서를 위치시키면 다음과 같이 메트릭 정보를 확인할 수 있습니다.



필터

필터 옵션 활용 시 원하는 애플리케이션 목록을 선택해 조회할 수 있습니다. `and` 조건을 통해 여러 개의 필터를 적용할 수 있습니다.

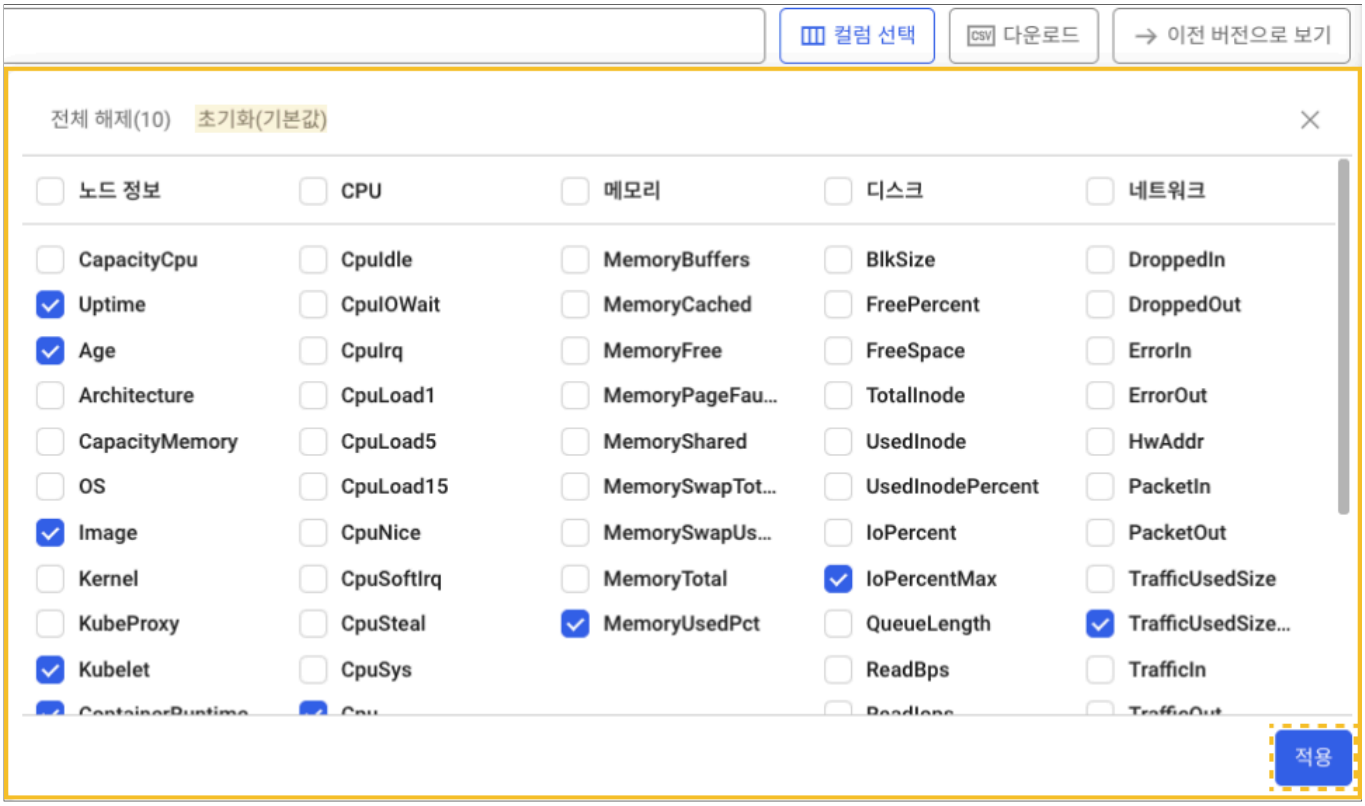
1. **필터** 입력 창에서 + **더하기 아이콘** 선택 시 다음과 같이 **필터 추가하기** 창이 나타납니다.

- **필터 키**와 **조건**을 선택하세요.
- 조건에 맞는 값을 선택하세요.

2. 필터 설정 후 **적용** 버튼을 선택해 해당 필터를 적용하세요.
3. 여러 개의 필터를 추가하려면 **필터** 입력 창 또는 **필터 추가하기** 창에서 ＋ **더하기 아이콘**을 선택하세요.

컬럼 선택

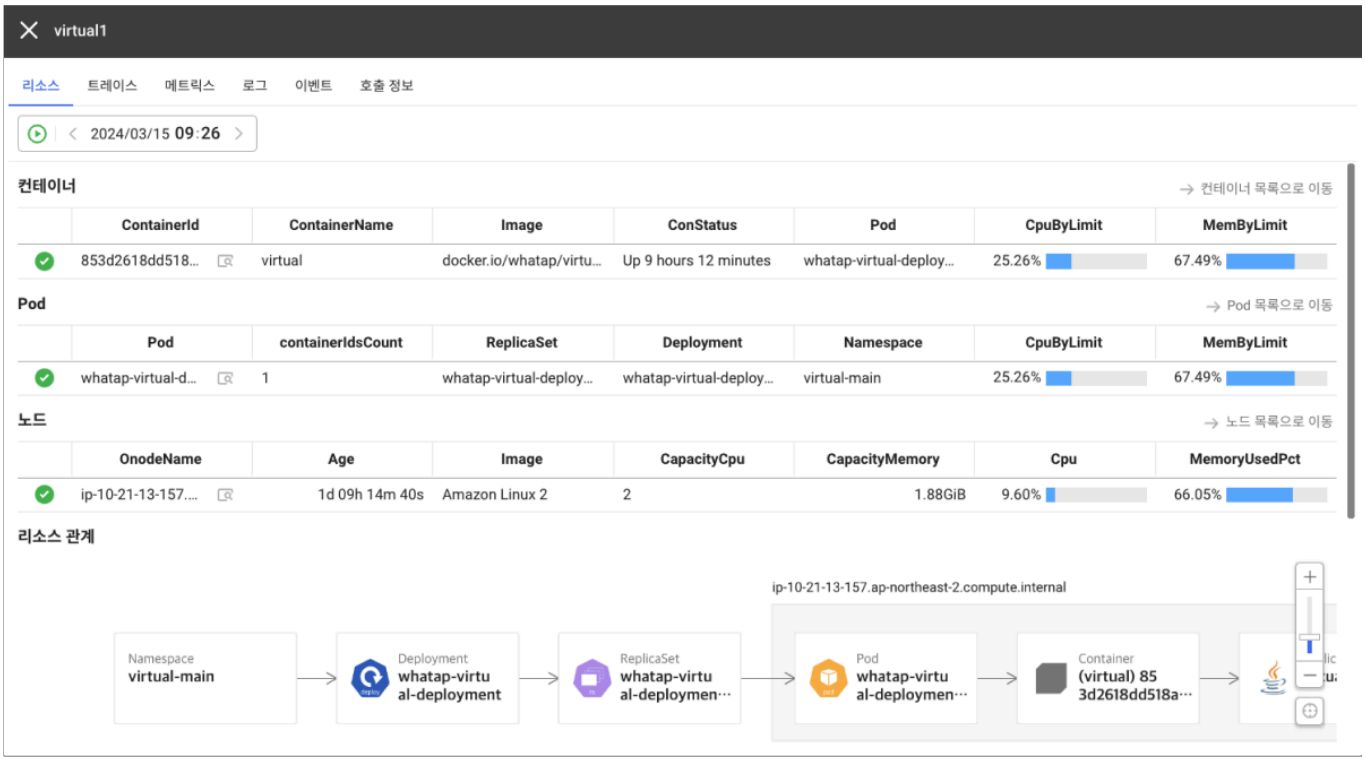
컬럼 선택 옵션 활용 시 원하는 애플리케이션 목록 정보를 선택해 목록을 조회할 수 있습니다.



- 선택한 컬럼 목록은 메뉴 재진입 시 유지됩니다.
- **초기화** 버튼 선택 시 사전 정의된 기본 주요 지표 컬럼이 선택됩니다. 이 경우 기존 컬럼 선택 내역은 삭제됩니다.
- 컬럼 선택 후 우측 하단 **적용** 버튼을 클릭 해 선택 내역을 적용할 수 있습니다.

상세 보기

목록에서 조회를 원하는 애플리케이션의 **상세 보기** 아이콘 선택 시 해당 애플리케이션에 대한 상세 정보를 확인할 수 있습니다.



> ⓘ 상세 보기 창의 컨테이너 요약 분석, 트레이스, 메트릭스, 로그, 이벤트, 호출 정보 탭의 경우 컨테이너 맵 상세 보기와 동일합니다.

리소스

대상 선택 후 상세 보기의 **리소스** 탭에서 선택한 리소스와 관련된 리소스 목록 및 관계도를 조회할 수 있습니다.

요약 정보

선택한 리소스와 관련된 리소스 목록의 요약 정보를 제공합니다. 리소스 명 옆에 상세보기 아이콘을 선택 시 해당 리소스 목록 화면으로 이동합니다.

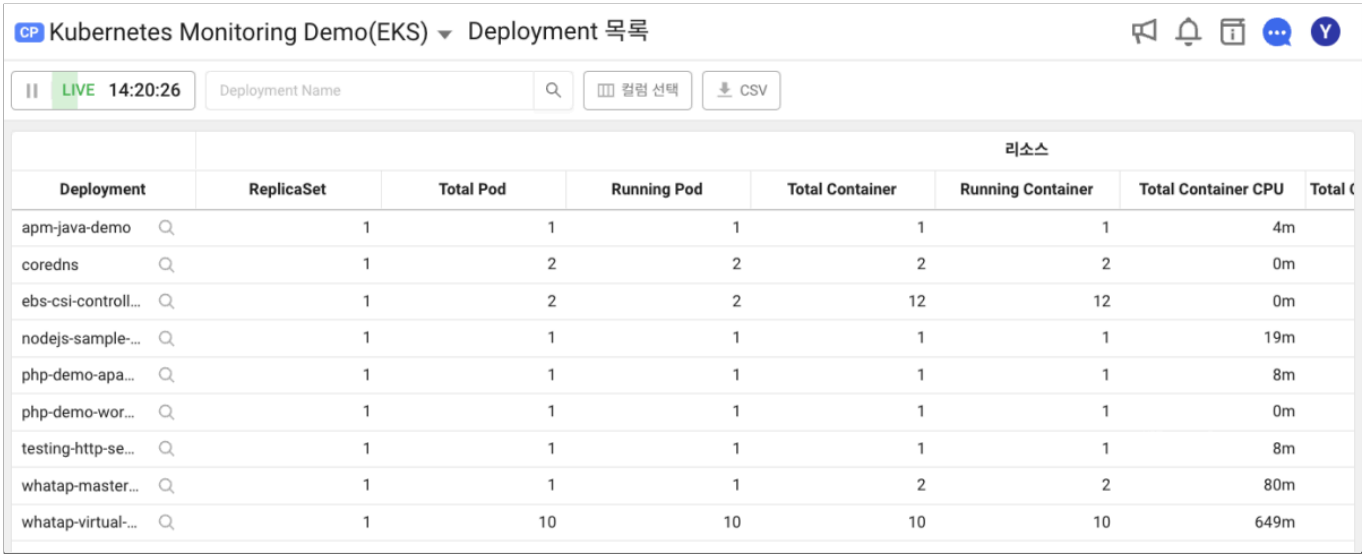
- **컨테이너**: 해당 컨테이너만 필터링된 상태의 **컨테이너 목록**으로 이동합니다.
- **Pod**: 해당 Pod만 필터링된 **Pod 목록**으로 이동합니다.
- **노드**: 해당 노드 이름이 선택된 상태의 **노드 상세**로 이동합니다.

리소스 관계도

리소스간 연결 관계와 상태를 시각적으로 확인할 수 있습니다. 목록과 동일하게 Pod, 컨테이너, 애플리케이션에 마우스 오버 시 상세 지표를 확인할 수 있습니다. 리소스 선택 시 해당 리소스 목록 페이지로 이동합니다.

Deployment 목록

Deployment의 목록을 조회합니다. 여러 Pod로 구성된 애플리케이션의 전체적인 운영 상태를 모니터링해 시스템의 안정성과 성능을 유지하도록 돕습니다. **Deployment 목록** 내 원하는 컬럼 선택 시 해당 컬럼 기준으로 목록을 정렬해 조회할 수 있습니다.



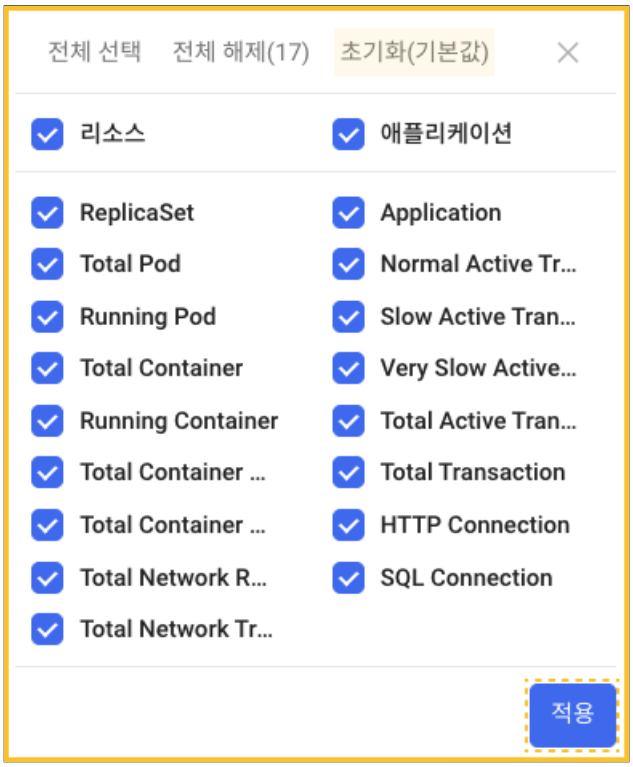
	리소스						
Deployment	ReplicaSet	Total Pod	Running Pod	Total Container	Running Container	Total Container CPU	Total
apm-java-demo	1	1	1	1	1	4m	
coredns	1	2	2	2	2	0m	
ebs-csi-controll...	1	2	2	12	12	0m	
nodejs-sample-...	1	1	1	1	1	19m	
php-demo-apa...	1	1	1	1	1	8m	
php-demo-wor...	1	1	1	1	1	0m	
testing-http-se...	1	1	1	1	1	8m	
whatap-master...	1	1	1	2	2	80m	
whatap-virtual-...	1	10	10	10	10	649m	

- 10초 주기로 실시간 데이터를 업데이트합니다.
- 검색, 컬럼 정렬, 상세 보기, CSV 파일 다운로드를 제공합니다.

  CSV 파일명, kubernetes_deployment_list_ `{pcode}` _ `{timestamp}` .csv

컬럼 선택

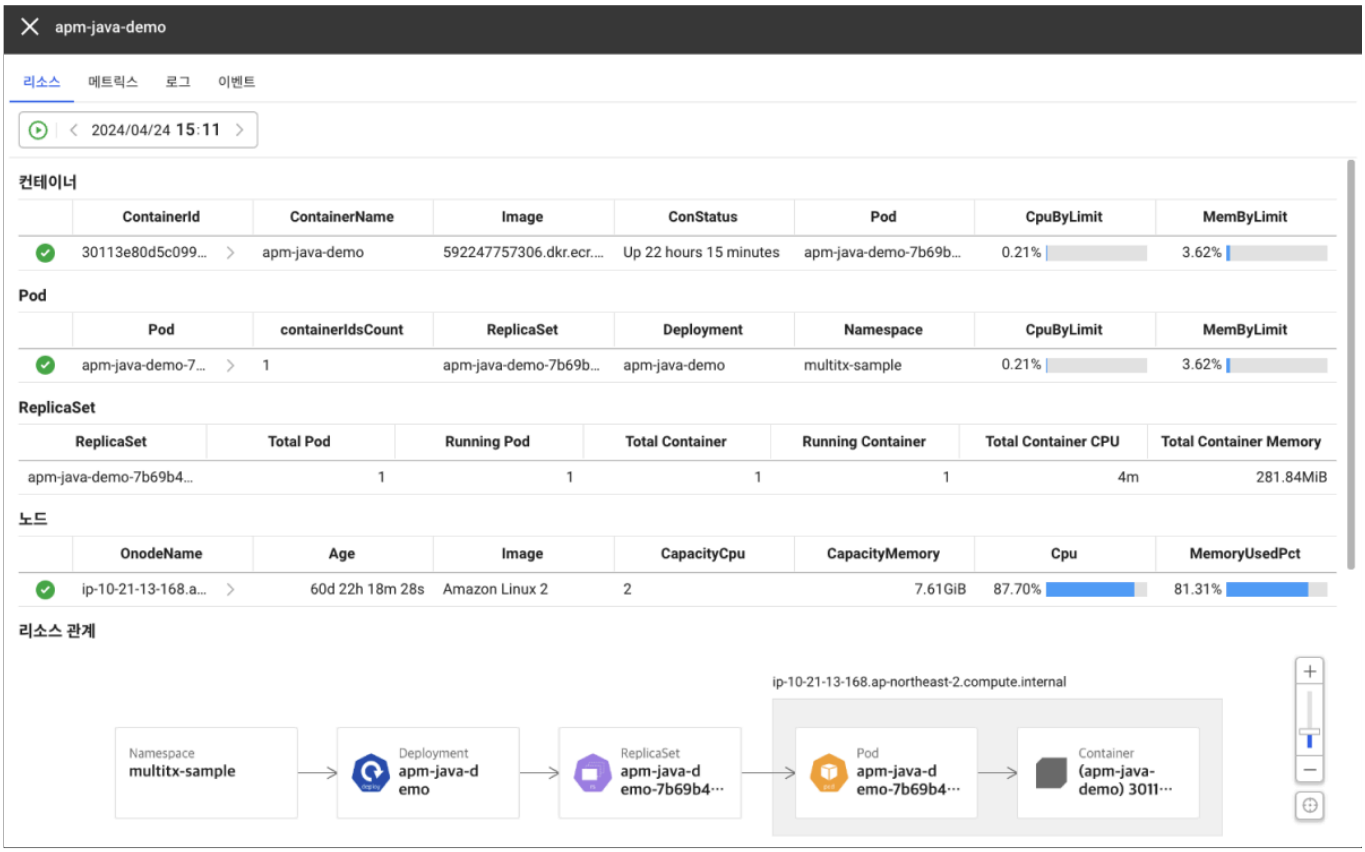
컬럼 선택 옵션 활용 시 원하는 Deployment 정보를 선택해 목록을 조회할 수 있습니다.



- 선택한 컬럼 목록은 메뉴 재진입 시 유지됩니다.
- **초기화** 버튼 선택 시 사전 정의된 기본 주요 지표 컬럼이 선택됩니다. 이 경우 기존 컬럼 선택 내역은 삭제됩니다.
- 컬럼 선택 후 우측 하단 **적용** 버튼을 클릭 해 선택 내역을 적용할 수 있습니다.

상세 보기

목록에서 조회를 원하는 Deployment의 🔍 **상세 보기** 아이콘 선택 시 해당 Deployment에 대한 상세 정보를 확인할 수 있습니다.



> ⓘ 상세 보기 창의 트레이스, 메트릭스, 로그, 이벤트 탭의 경우 컨테이너 맵 상세 보기와 동일합니다.

리소스

대상 선택 후 상세 보기의 **리소스** 탭에서 선택한 리소스와 관련된 리소스 목록 및 관계도를 조회할 수 있습니다.

요약 정보

선택한 리소스와 관련된 리소스 목록의 요약 정보를 제공합니다. 리소스 명 옆에 상세보기 아이콘을 선택 시 해당 리소스 목록 화면으로 이동합니다.

- **컨테이너**: 해당 컨테이너만 필터링된 상태의 **컨테이너 목록**으로 이동합니다.
- **Pod**: 해당 Pod만 필터링된 **Pod 목록**으로 이동합니다.
- **노드**: 해당 노드 이름이 선택된 상태의 **노드 상세**로 이동합니다.

- **ReplicaSet**: **ReplicaSet**을 기준으로 실행중인 Pod와 컨테이너의 요약 정보를 제공합니다.

리소스 관계도

리소스간 연결 관계와 상태를 시각적으로 확인할 수 있습니다. 목록과 동일하게 Pod, 컨테이너, 애플리케이션에 마우스 오버 시 상세 지표를 확인할 수 있습니다. 리소스 선택 시 해당 리소스 목록 페이지로 이동합니다.

Service 목록

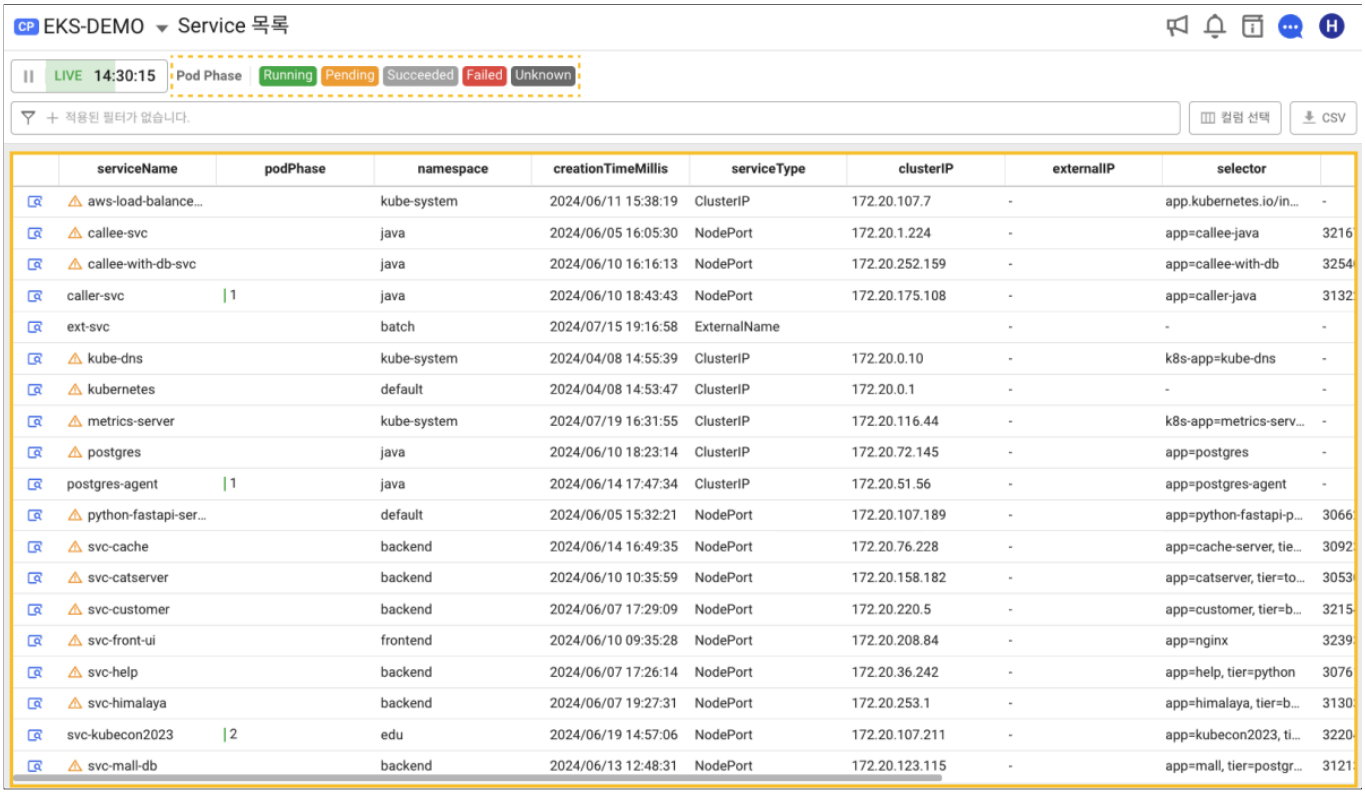
> ⓘ 와탭 쿠버네티스 에이전트 1.7.8 버전 이상이 필요합니다.

Service 목록을 조회합니다. 와탭 쿠버네티스 모니터링의 **Service 목록**을 통해 클러스터 내 네트워크 구성을 쉽게 파악하고 각 Service의 구성과 현재 상태를 효율적으로 모니터링할 수 있습니다.

쿠버네티스에서 **Service**는 클러스터 내부 및 외부에서 Pod에 접속하기 위한 오브젝트입니다. **Service 목록** 메뉴에서 각 Service의 세부 구성을 한눈에 확인할 수 있으며, 연결된 Pod의 상태를 실시간으로 모니터링할 수 있습니다. 이를 통해 Service와 Pod 간의 비정상적인 연결 상태를 신속하게 파악하고 해결할 수 있습니다.

기본 화면 안내

현재 클러스터에 존재하는 **Service 목록** 정보를 다음과 같이 확인할 수 있습니다. 조회 시간 기준 최근 1분간 수집된 Service 정보를 기준으로 목록을 표시합니다. 네임스페이스, 레이블 셀렉터 등 다양한 기준으로 리소스를 구분하고 관리할 수 있으며 Service의 다양한 속성을 명확하게 확인할 수 있습니다.



serviceName	podPhase	namespace	creationTimeMillis	serviceType	clusterIP	externalIP	selector	
aws-load-balance...		kube-system	2024/06/11 15:38:19	ClusterIP	172.20.107.7	-	app.kubernetes.io/in...	-
callee-svc		java	2024/06/05 16:05:30	NodePort	172.20.1.224	-	app=callee-java	3216
callee-with-db-svc		java	2024/06/10 16:16:13	NodePort	172.20.252.159	-	app=callee-with-db	3254
caller-svc	1	java	2024/06/10 18:43:43	NodePort	172.20.175.108	-	app=caller-java	3132
ext-svc		batch	2024/07/15 19:16:58	ExternalName		-	-	-
kube-dns		kube-system	2024/04/08 14:55:39	ClusterIP	172.20.0.10	-	k8s-app=kube-dns	-
kubernetes		default	2024/04/08 14:53:47	ClusterIP	172.20.0.1	-	-	-
metrics-server		kube-system	2024/07/19 16:31:55	ClusterIP	172.20.116.44	-	k8s-app=metrics-serv...	-
postgres		java	2024/06/10 18:23:14	ClusterIP	172.20.72.145	-	app=postgres	-
postgres-agent	1	java	2024/06/14 17:47:34	ClusterIP	172.20.51.56	-	app=postgres-agent	-
python-fastapi-ser...		default	2024/06/05 15:32:21	NodePort	172.20.107.189	-	app=python-fastapi-p...	3066
svc-cache		backend	2024/06/14 16:49:35	NodePort	172.20.76.228	-	app=cache-server, tie...	3092
svc-catserver		backend	2024/06/10 10:35:59	NodePort	172.20.158.182	-	app=catserver, tier=to...	3053
svc-customer		backend	2024/06/07 17:29:09	NodePort	172.20.220.5	-	app=customer, tier=b...	3215
svc-front-ui		frontend	2024/06/10 09:35:28	NodePort	172.20.208.84	-	app=nginx	3239
svc-help		backend	2024/06/07 17:26:14	NodePort	172.20.36.242	-	app=help, tier=python	3076
svc-himalaya		backend	2024/06/07 19:27:31	NodePort	172.20.253.1	-	app=himalaya, tier=b...	3130
svc-kubecon2023	2	edu	2024/06/19 14:57:06	NodePort	172.20.107.211	-	app=kubecon2023, ti...	3220
svc-mall-db		backend	2024/06/13 12:48:31	NodePort	172.20.123.115	-	app=mall, tier=postgr...	3121

> ⓘ **Pod Phase**
>
> **podPhase** 컬럼은 해당 Service에 연결된 Pod의 상태별 개수를 표시합니다. Pod의 상태는 색상으로 표기합니다. Pod 상태 관련해 상단 **Pod Phase** 안내를 참조하세요.

> ⓘ 목록형 메뉴의 **필터**, **컬럼 선택** 안내는 동일합니다.

필터

필터 옵션 활용 시 원하는 을 선택해 조회할 수 있습니다. `and` 조건을 통해 여러 개의 필터를 적용할 수 있습니다.

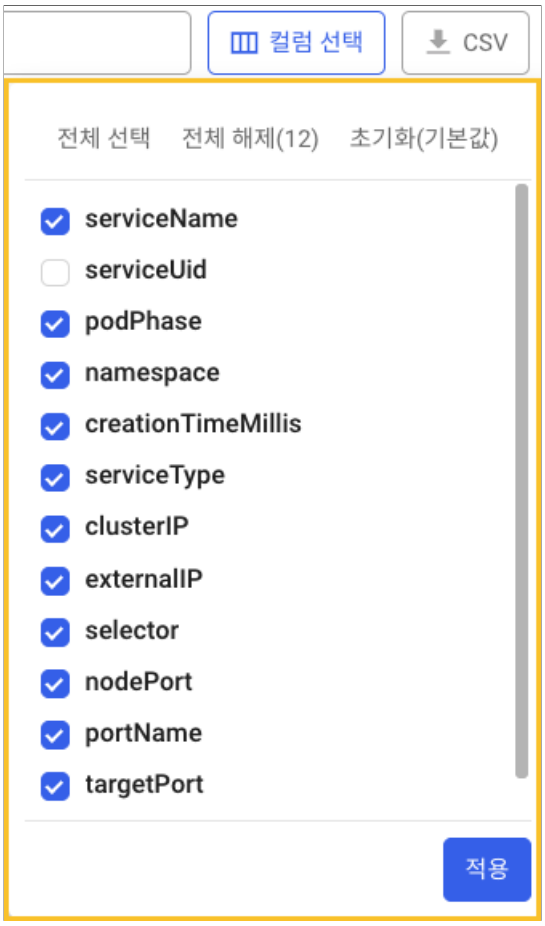
1. **필터** 입력 창에서 ＋ **더하기 아이콘** 선택 시 다음과 같이 **필터 추가하기** 창이 나타납니다.

- **필터 키**와 **조건**을 선택하세요.
- 조건에 맞는 값을 선택하세요.

2. 필터 설정 후 **적용** 버튼을 선택해 해당 필터를 적용하세요.
3. 여러 개의 필터를 추가하려면 **필터** 입력 창 또는 **필터 추가하기** 창에서 ＋ **더하기 아이콘**을 선택하세요.

컬럼 선택

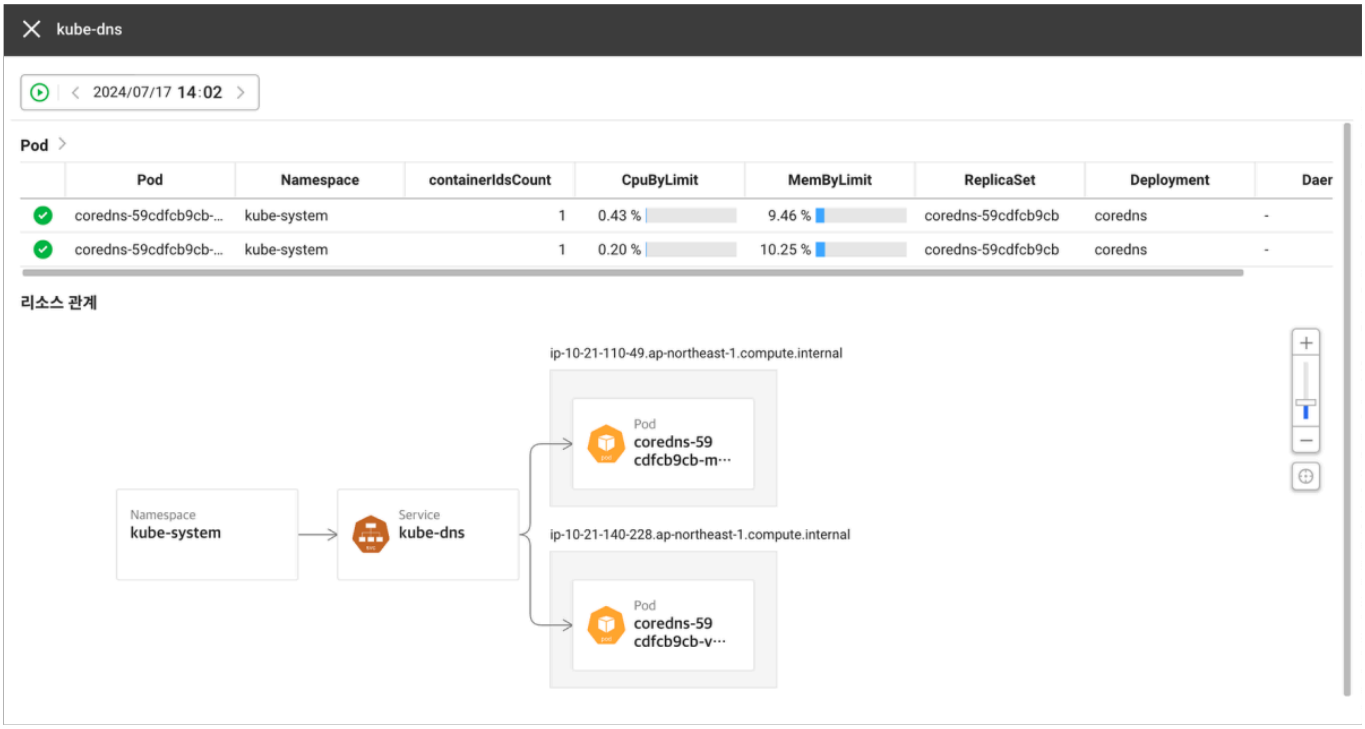
컬럼 선택 옵션 활용 시 원하는 정보를 선택해 목록을 조회할 수 있습니다.



- 선택한 컬럼 목록은 메뉴 재진입 시 유지됩니다.
- **초기화** 버튼 선택 시 사전 정의된 기본 주요 지표 컬럼이 선택됩니다. 이 경우 기존 컬럼 선택 내역은 삭제됩니다.
- 컬럼 선택 후 우측 하단 **적용** 버튼을 클릭 해 선택 내역을 적용할 수 있습니다.

상세 보기

목록에서 조회를 원하는 Service의 🔍 **상세 보기** 아이콘 선택 시 해당 Service에 연결된 Pod 목록을 확인할 수 있습니다.


kube-dns
2024/07/17 14:02
Pod
Pod
Namespace
containerIdsCount
CpuByLimit
MemByLimit
ReplicaSet
Deployment
Daer
coredns-59cdfcb9cb-...
kube-system
1
0.43 %
9.46 %
coredns-59cdfcb9cb
coredns
-
coredns-59cdfcb9cb-...
kube-system
1
0.20 %
10.25 %
coredns-59cdfcb9cb
coredns
-
리소스 관계
ip-10-21-110-49.ap-northeast-1.compute.internal
Pod
coredns-59
cdfcb9cb-m···
Namespace
kube-system
Service
kube-dns
ip-10-21-140-228.ap-northeast-1.compute.internal
Pod
coredns-59
cdfcb9cb-v···

Ingress 목록

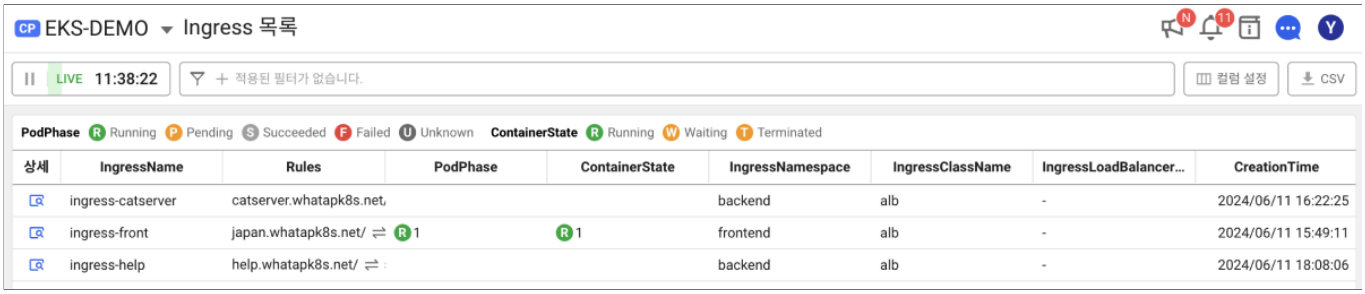
> 와탭 쿠버네티스 에이전트 1.7.13 버전 이상이 필요합니다.

Ingress 목록을 조회합니다. 쿠버네티스 외부에서 유입되는 요청은 **Ingress**를 통해 컨테이너 내부의 애플리케이션까지 최종적으로 라우팅되며, Ingress 설정에 따라 특정 Service나 Pod에 연결됩니다. Ingress 모니터링은 쿠버네티스 클러스터로 들어오는 외부 트래픽의 흐름을 파악하고 관리하는 데에 중요한 역할을 합니다.

와탭의 **Ingress 목록**은 Ingress별 라우팅 정책을 시각화한 목록 뷰를 제공합니다. 이를 통해 각 Ingress가 어떻게 트래픽을 분배하고 있는지 한눈에 파악할 수 있습니다.

기본 화면 안내

Ingress 정책 중심으로 주요 Ingress 정보와 해당 정책에 연결된 Pod 및 컨테이너의 상태별 개수를 다음과 같이 조회할 수 있습니다. 목록 및 상세 보기를 통해 Ingress 정책별 연결 상태와 주요 오브젝트 정보를 종합적으로 살펴볼 수 있으며 또한 목록 상단에서 Pod 및 컨테이너의 상태별 아이콘 정보를 확인할 수 있습니다.



> 목록형 메뉴의 **필터**, **컬럼 선택** 안내는 동일합니다.

필터

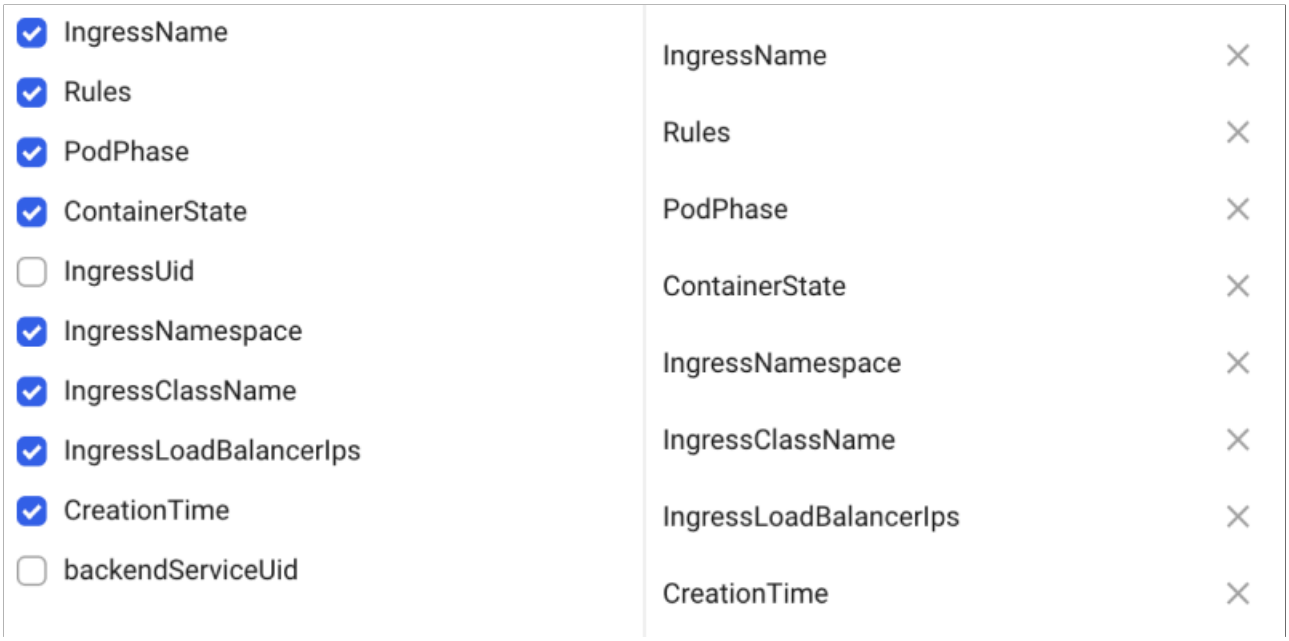
필터 옵션 활용 시 원하는 을 선택해 조회할 수 있습니다. `and` 조건을 통해 여러 개의 필터를 적용할 수 있습니다.

1. **필터** 입력 창에서 + **더하기 아이콘** 선택 시 다음과 같이 **필터 추가하기** 창이 나타납니다.

   - **필터 키**와 **조건**을 선택하세요.
   - 조건에 맞는 값을 선택하세요.
2. 필터 설정 후 **적용** 버튼을 선택해 해당 필터를 적용하세요.
3. 여러 개의 필터를 추가하려면 **필터** 입력 창 또는 **필터 추가하기** 창에서 + **더하기 아이콘**을 선택하세요.

컬럼 선택

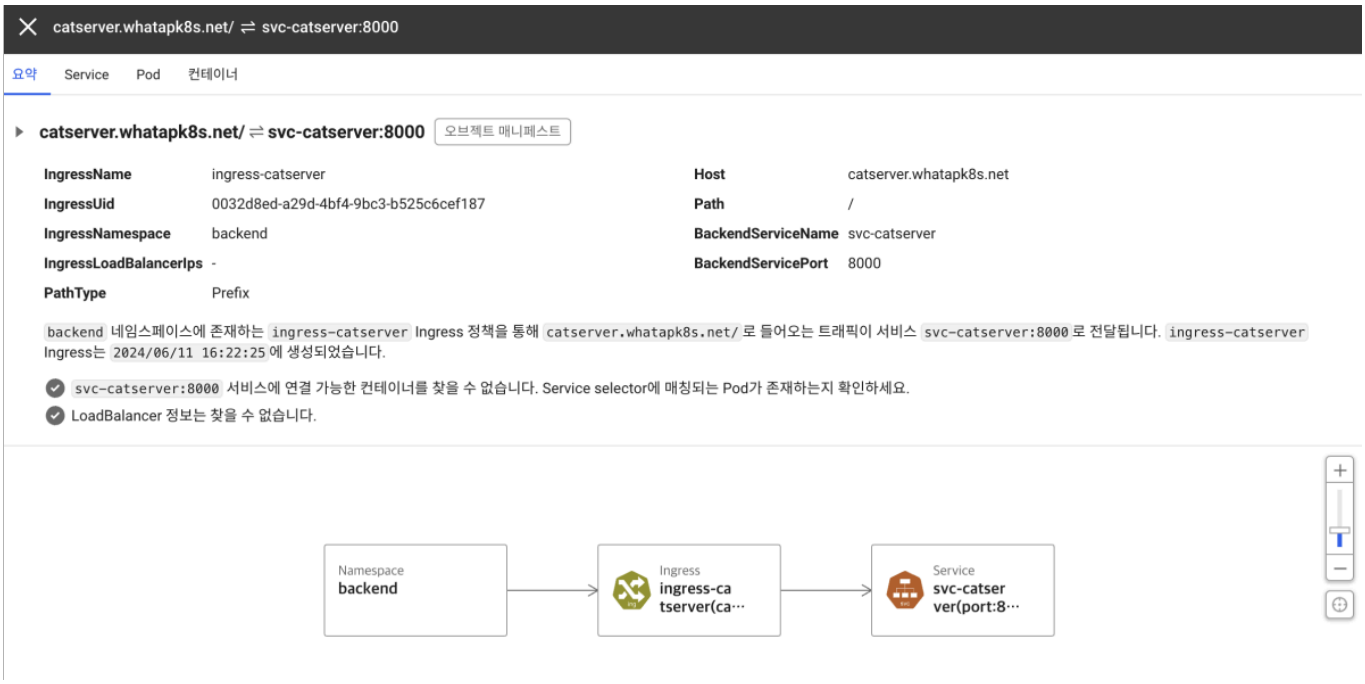
컬럼 선택 옵션 활용 시 원하는 정보를 선택해 목록을 조회할 수 있습니다.



- 선택한 컬럼 목록은 메뉴 재진입 시 유지됩니다.
- **초기화** 버튼 선택 시 사전 정의된 기본 주요 지표 컬럼이 선택됩니다. 이 경우 기존 컬럼 선택 내역은 삭제됩니다.
- 컬럼 선택 후 우측 하단 **적용** 버튼을 클릭 해 선택 내역을 적용할 수 있습니다.

상세 보기

목록에서 조회를 원하는 Ingress 정책 항목의 🔍 **상세 보기** 아이콘 선택 시 해당 Ingress 정책에 대한 상세 정보를 확인할 수 있습니다.



요약

Ingress 기본 정보 및 선택한 Ingress 정책에 대한 **요약** 분석을 제공합니다. Ingress의 네임스페이스, 연결된 호스트 및 경로 설정, 트래픽이 전달되는 백엔드 서비스 정보 등을 확인할 수 있습니다. 이를 통해 외부 트래픽의 흐름과 Ingress 정책의 구조를 살펴볼 수 있습니다. 또한 해당 정책을 통해 연결된 주요 오브젝트의 관계도를 시각화하여 정책 내 리소스 관계를 한눈에 파악할 수 있습니다.

> ⓘ **요약** 탭 상단에서 오브젝트 매니페스트 정보를 확인할 수 있습니다.

Service

선택한 Ingress 정책과 연결된 **Service** 목록을 제공합니다. 이를 통해 Ingress가 어떤 서비스와 연결되어 있는지 확인할 수 있습니다. 각 서비스의 연결 상태 및 서비스와 관련된 세부 사항을 조회할 수 있습니다.

Pod

선택한 Ingress 정책과 연결된 **Pod** 목록을 제공합니다. 연결된 각 Pod의 상태와 기본 정보 및 요약 분석 정보를 확인할 수 있습니다.

컨테이너

선택한 Ingress 정책과 연결된 **컨테이너** 목록을 제공합니다. Ingress 정책을 통한 트래픽이 전달되는 컨테이너의 상태와 성능 및 요약 분석 정보를 확인할 수 있습니다.

PV/PVC 목록

> **요구 사항**
>
> - 와탭 쿠버네티스 에이전트 1.8.1 버전 이상이 필요합니다.
> - 쿠버네티스 클러스터 프로젝트(CP)일 경우 해당 메뉴를 조회할 수 있습니다.

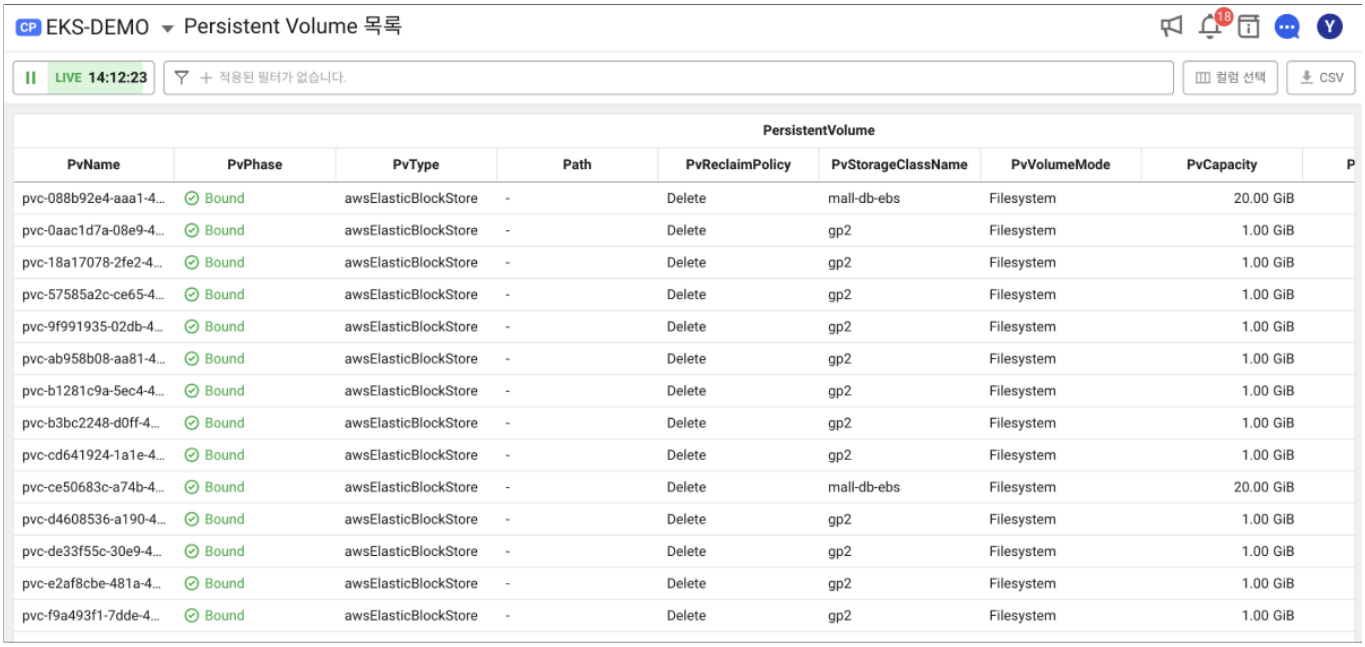
Persistent Volume(PV)은 쿠버네티스 클러스터에서 사용 가능한 물리적 스토리지 리소스를 의미합니다. **Persistent Volume Claim**(PVC)은 Pod의 스토리지 요구사항을 선언적으로 정의하는 객체입니다. PVC는 PV와 연결되어 이를 통해 Pod가 데이터를 읽고 쓸 수 있는 접근 경로를 제공합니다.

와탭 쿠버네티스는 **Persistent Volume 목록**과 **Persistent Volume Claim 목록** 메뉴를 통해 클러스터의 저장소 상태를 모니터링할 수 있습니다. 해당 메뉴는 저장소 리소스의 사용 상태와 연결 관계를 직관적으로 제공하며, 저장소 용량, 접근 모드, 이벤트 상태 등의 정보를 한눈에 확인할 수 있습니다.

Persistent Volume 목록

홈 화면 > 프로젝트 선택 > **스토리지** > **Persistent Volume 목록**

Persistent Volume 목록 메뉴에서 현재 클러스터에 존재하는 PV를 조회할 수 있습니다. 조회 시간 기준 최근 1분 동안 수집된 PV 정보 기반으로 목록을 표시합니다. PVC와의 연결 상태 및 PV의 바인딩 상태를 직관적으로 확인할 수 있습니다. 또한 해당 PV와 연결된 PVC 상세 정보를 간편하게 확인할 수 있습니다.



PersistentVolume

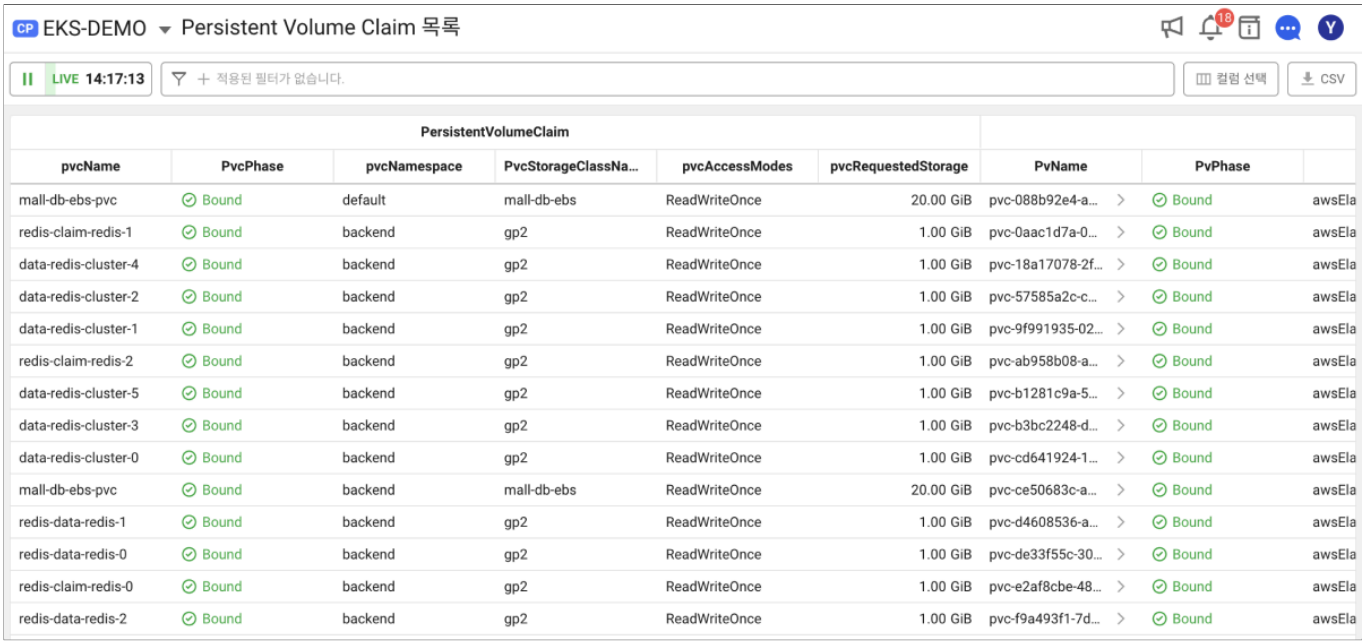
PvName	PvPhase	PvType	Path	PvReclaimPolicy	PvStorageClassName	PvVolumeMode	PvCapacity	P
pvc-088b92e4-aaa1-4...	Bound	awsElasticBlockStore	-	Delete	mall-db-ebs	Filesystem	20.00 GiB	
pvc-0aac1d7a-08e9-4...	Bound	awsElasticBlockStore	-	Delete	gp2	Filesystem	1.00 GiB	
pvc-18a17078-2fe2-4...	Bound	awsElasticBlockStore	-	Delete	gp2	Filesystem	1.00 GiB	
pvc-57585a2c-ce65-4...	Bound	awsElasticBlockStore	-	Delete	gp2	Filesystem	1.00 GiB	
pvc-9f991935-02db-4...	Bound	awsElasticBlockStore	-	Delete	gp2	Filesystem	1.00 GiB	
pvc-ab958b08-aa81-4...	Bound	awsElasticBlockStore	-	Delete	gp2	Filesystem	1.00 GiB	
pvc-b1281c9a-5ec4-4...	Bound	awsElasticBlockStore	-	Delete	gp2	Filesystem	1.00 GiB	
pvc-b3bc2248-d0ff-4...	Bound	awsElasticBlockStore	-	Delete	gp2	Filesystem	1.00 GiB	
pvc-cd641924-1a1e-4...	Bound	awsElasticBlockStore	-	Delete	gp2	Filesystem	1.00 GiB	
pvc-ce50683c-a74b-4...	Bound	awsElasticBlockStore	-	Delete	mall-db-ebs	Filesystem	20.00 GiB	
pvc-d4608536-a190-4...	Bound	awsElasticBlockStore	-	Delete	gp2	Filesystem	1.00 GiB	
pvc-de33f55c-30e9-4...	Bound	awsElasticBlockStore	-	Delete	gp2	Filesystem	1.00 GiB	
pvc-e2af8cbe-481a-4...	Bound	awsElasticBlockStore	-	Delete	gp2	Filesystem	1.00 GiB	
pvc-f9a493f1-7dde-4...	Bound	awsElasticBlockStore	-	Delete	gp2	Filesystem	1.00 GiB	

- **CSV** 버튼 클릭 시 CSV 파일을 다운로드할 수 있습니다.

Persistent Volume Claim 목록

홈 화면 > 프로젝트 선택 > **스토리지** > **Persistent Volume Claim 목록**

Persistent Volume Claim 목록 메뉴에서 현재 클러스터에 존재하는 PVC를 조회할 수 있습니다. 조회 시간 기준 최근 1분 동안 수집된 PVC 정보 기반으로 목록을 표시합니다. PV와의 연결 상태 및 PVC의 바인딩 상태를 직관적으로 확인할 수 있습니다. 또한 해당 PVC와 연결된 PV 상세 정보를 간편하게 확인할 수 있습니다.



PersistentVolumeClaim

pvcName	PvcPhase	pvcNamespace	PvcStorageClassNa...	pvcAccessModes	pvcRequestedStorage	PvName	PvPhase	
mall-db-ebs-pvc	Bound	default	mall-db-ebs	ReadWriteOnce	20.00 GiB	pvc-088b92e4-a...	Bound	awsEla
redis-claim-redis-1	Bound	backend	gp2	ReadWriteOnce	1.00 GiB	pvc-0aac1d7a-0...	Bound	awsEla
data-redis-cluster-4	Bound	backend	gp2	ReadWriteOnce	1.00 GiB	pvc-18a17078-2f...	Bound	awsEla
data-redis-cluster-2	Bound	backend	gp2	ReadWriteOnce	1.00 GiB	pvc-57585a2c-c...	Bound	awsEla
data-redis-cluster-1	Bound	backend	gp2	ReadWriteOnce	1.00 GiB	pvc-9f991935-02...	Bound	awsEla
redis-claim-redis-2	Bound	backend	gp2	ReadWriteOnce	1.00 GiB	pvc-ab958b08-a...	Bound	awsEla
data-redis-cluster-5	Bound	backend	gp2	ReadWriteOnce	1.00 GiB	pvc-b1281c9a-5...	Bound	awsEla
data-redis-cluster-3	Bound	backend	gp2	ReadWriteOnce	1.00 GiB	pvc-b3bc2248-d...	Bound	awsEla
data-redis-cluster-0	Bound	backend	gp2	ReadWriteOnce	1.00 GiB	pvc-cd641924-1...	Bound	awsEla
mall-db-ebs-pvc	Bound	backend	mall-db-ebs	ReadWriteOnce	20.00 GiB	pvc-ce50683c-a...	Bound	awsEla
redis-data-redis-1	Bound	backend	gp2	ReadWriteOnce	1.00 GiB	pvc-d4608536-a...	Bound	awsEla
redis-data-redis-0	Bound	backend	gp2	ReadWriteOnce	1.00 GiB	pvc-de33f55c-30...	Bound	awsEla
redis-claim-redis-0	Bound	backend	gp2	ReadWriteOnce	1.00 GiB	pvc-e2af8cbe-48...	Bound	awsEla
redis-data-redis-2	Bound	backend	gp2	ReadWriteOnce	1.00 GiB	pvc-f9a493f1-7d...	Bound	awsEla

- **CSV** 버튼 클릭 시 CSV 파일을 다운로드할 수 있습니다.
- **PvName** 컬럼 오른쪽 **>** 버튼 클릭 시 **Persistent Volume 목록** 메뉴로 이동합니다.

필터

필터 옵션 활용 시 원하는 을 선택해 조회할 수 있습니다. `and` 조건을 통해 여러 개의 필터를 적용할 수 있습니다.

1. **필터** 입력 창에서 **더하기 아이콘** 선택 시 다음과 같이 **필터 추가하기** 창이 나타납니다.

   - **필터 키**와 **조건**을 선택하세요.
   - 조건에 맞는 값을 선택하세요.

2. 필터 설정 후 **적용** 버튼을 선택해 해당 필터를 적용하세요.
3. 여러 개의 필터를 추가하려면 **필터** 입력 창 또는 **필터 추가하기** 창에서 ＋ **더하기 아이콘**을 선택하세요.

컬럼

그룹	컬럼명	설명
PersistentVolume	**PvUid**	Persistent Volume의 고유 식별자입니다.
	PvName	Persistent Volume의 이름입니다.
	PvPhase	Persistent Volume의 현재 상태를 표시합니다.
	PvType	Persistent Volume의 스토리지 유형입니다.
	Path	Persistent Volume이 마운트된 경로를 의미합니다.
	PvReclaimPolicy	Persistent Volume의 반환 정책을 의미합니다.
	PvStorageClassName	해당 PV에 할당된 스토리지 클래스 이름입니다.
	PvVolumeMode	Persistent Volume의 볼륨 모드 속성을 의미합니다.
	PvCapacity	Persistent Volume의 총 저장 용량을 표시합니다.
	PvMessage	현재 Persistent Volume이 특정 Phase를 가지게 된 원인을 설명하는 메시지입니다.
	PvReason	Persistent Volume과 관련된 오류를 나타내는 키워드로 문제의 원인이나 현재 상태를 간략하게 표시합니다.
PersistentVolumeClaim	**PvcUid**	Persistent Volume Claim의 고유 식별자입니다.
	PvcName (PVC)	Persistent Volume Claim의 이름입니다.
	PvcNamespace	Persistent Volume Claim이 속한 네임스페이스를 의미합니다.

그룹	컬럼명	설명
	PvcStorageClassName	Persistent Volume Claim에 설정된 스토리지 클래스의 이름입니다.
	PvcAccessModes	Persistent Volume Claim의 접근 모드를 표시합니다.
	PvcRequestedStorage	Persistent Volume Claim이 요청한 스토리지 용량을 나타냅니다.

- Persistent Volume의 반환 정책에 대한 자세한 설명은 다음 문서를 참조하세요.

  Retain, **Delete**

- Persistent Volume의 볼륨 모드에 대한 자세한 설명은 다음 문서를 참조하세요.

  Filesystem, **Block**

- Persistent Volume Claim 접근 모드에 대한 자세한 설명은 다음 문서를 참조하세요.

  ReadWriteOnce, **ReadOnlyMany**, **ReadWriteMany**

Phase

PvPhase

PvPhase 항목은 Persistent Volume의 상태를 표시합니다.

Phase	설명
Available	PV가 아직 어떤 PVC와도 바인딩되지 않은 상태입니다.
Bound	PV가 특정 PVC에 바인딩되어 사용 중인 상태입니다.
Released	PVC가 삭제되었으나 PV가 아직 반환되지 않은 상태입니다.
Failed	PV가 자동 반환에 실패하여 사용할 수 없는 상태입니다.

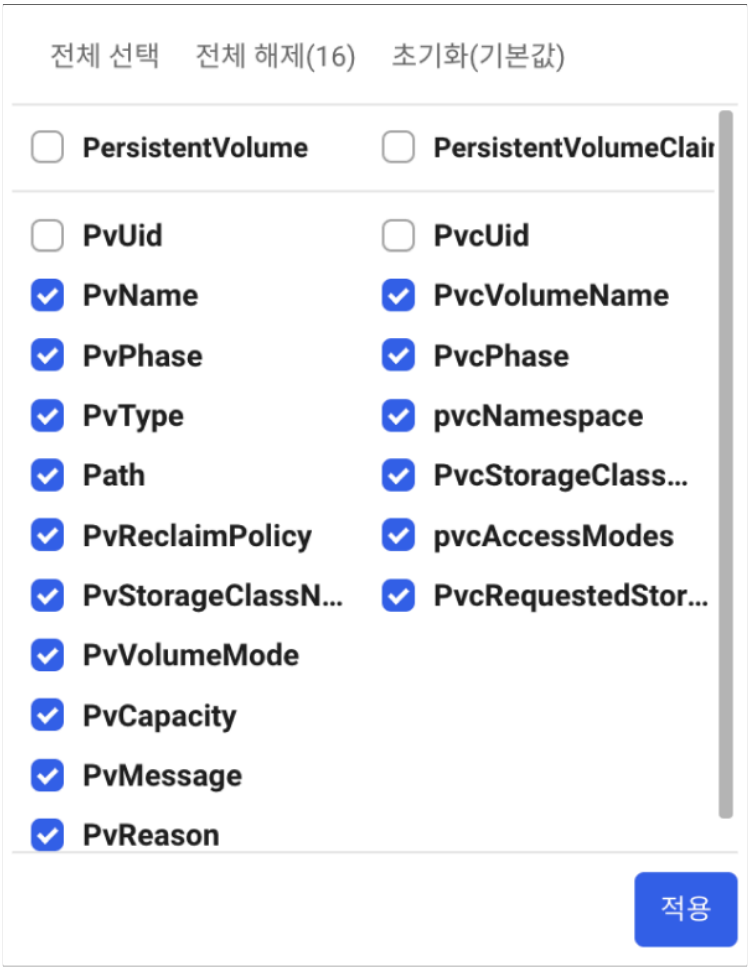
PvcPhase

PvcPhase 항목은 Persistent Volume Claim의 상태를 표시합니다.

Phase	설명
Pending	PVC가 PV와 아직 바인딩되지 않았거나 PV 할당 대기 중인 상태입니다.
Bound	PVC가 PV와 바인딩되어 사용 중인 상태입니다.
Lost	연결된 PV가 삭제되거나 더이상 유효하지 않아 사용할 수 없는 상태입니다.

컬럼 선택

컬럼 선택 옵션 활용 시 원하는 정보를 선택해 목록을 조회할 수 있습니다.



- 선택한 컬럼 목록은 메뉴 재진입 시 유지됩니다.
- **초기화** 버튼 선택 시 사전 정의된 기본 주요 지표 컬럼이 선택됩니다. 이 경우 기존 컬럼 선택 내역은 삭제됩니다.
- 컬럼 선택 후 우측 하단 **적용** 버튼을 클릭 해 선택 내역을 적용할 수 있습니다.

> ⓘ **PersistentVolume** 또는 **PersistentVolumeClaim** 컬럼 그룹을 일괄 선택 및 해제할 수 있습니다.

kube-apiserver 대시보드

> **요구 사항**
>
> - 와탭 쿠버네티스 에이전트 1.5.6 버전 이상이 필요합니다.
> - 쿠버네티스 클러스터 프로젝트(CP)일 경우 해당 메뉴를 조회할 수 있습니다.

kube-apiserver는 **컨트롤 플레인**(Control Plane) 내/외부 요청을 처리하는 컴포넌트입니다. **kube-apiserver 대시보드** 메뉴를 통해 kube-apiserver의 상태와 성능 메트릭을 수집 후 kube-apiserver의 응답 시간, 처리량, 요청 실패 등을 추적해 클러스터의 가용성과 성능을 쉽게 모니터링할 수 있습니다.

기본 화면 안내

kube-apiserver 대시보드 메뉴는 최근 1시간 동안의 kube-apiserver 주요 지표를 1분 주기로 반영합니다. 다음과 같은 지표를

확인할 수 있습니다. 지표 상세는 하단 메트릭 안내를 참조하세요.

- 클러스터 부하 및 사용량 모니터링을 위해 메소드, 리소스, 응답 코드별 요청 횟수 현황 및 추이를 제공합니다.
- 클러스터 부하 상태를 파악할 수 있도록 현재 처리 중인 API 요청 수를 제공합니다.
- 생성된 OS 스레드와 현재 존재하는 고루틴의 수를 제공합니다.

kube-apiserver 모니터링 활성화

와탭 쿠버네티스 에이전트 설치 시 다운로드한 yaml 파일을 다음 안내에 따라 수정해 kube-apiserver 모니터링을 활성화하세요.

1. `whatap-master-agent` deployment의 spec 하위에 kube-apiserver 모니터링을 위한 에이전트 컨테이너 spec을 추가하세요.

```
containers:
  - name: whatap-control-plane-helper
    image: whatap/kube_mon
    imagePullPolicy: Always
    ports:
      - containerPort: 9496
    command: ["/data/agent/master/whatap_control_plane_helper"]
    args: ["-collect_control_plane_monitoring_enabled=true"]
```

다음 예시를 참조하세요.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: whatap-master-agent
  namespace: whatap-monitoring
spec:
  replicas: 1
  selector:
    matchLabels:
      name: whatap-master-agent
  template:
    metadata:
      labels:
```

```
    name: whatap-master-agent
  spec:
    containers:
      - name: whatap-control-plane-helper
        image: whatap/kube_mon
        imagePullPolicy: Always
        ports:
          - containerPort: 9496
        command: ["/data/agent/master/whatap_control_plane_helper"]
        args: ["-collect_control_plane_monitoring_enabled=true"]
      - name: whatap-master-agent
        image: whatap/kube_mon
        imagePullPolicy: Always
        resources:
          requests:
            memory: 300Mi
            cpu: 100m
          limits:
            memory: 350Mi
            cpu: 200m
        ports:
          - containerPort: 6600
        command: ['/bin/entrypoint.sh']
        env:
          - name: WHATAP_LICENSE
            value: "xxxxxxxx-12345xxxxxx-x00e000000xxx8"
          - name: WHATAP_HOST
            value: "13.124.11.223/13.209.172.35"
          - name: WHATAP_PORT
            value: "6600"
          - name: WHATP_MEM_LIMIT
            valueFrom:
              resourceFieldRef:
                containerName: whatap-master-agent
                resource: limits.memory
        volumeMounts:
          - name: start-script-volume
            mountPath: /bin/entrypoint.sh
            readOnly: true
```

```
          subPath: entrypoint.sh
        - mountPath: /whatap_conf
          name: whatap-config-volume
    volumes:
      - name: start-script-volume
        configMap:
          defaultMode: 0700
          name: master-start-script
      - name: whatap-config-volume
        emptyDir: {}
    serviceAccount: whatap
```

2. `whatap-master-agent` deployment의 `whatap-master-agent` 컨테이너 spec에서 env 설정에 다음 내용을 추가하세요.

```
env:
  - name: collect_control_plane_monitoring_enabled
    value: "true"
```

다음 예시를 참조하세요.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: whatap-master-agent
  namespace: whatap-monitoring
spec:
  replicas: 1
  selector:
    matchLabels:
      name: whatap-master-agent
  template:
    metadata:
      labels:
        name: whatap-master-agent
    spec:
      containers:
        - name: whatap-control-plane-helper
```

```
    image: whatap/kube_mon
    imagePullPolicy: Always
    ports:
      - containerPort: 9496
    command: ["/data/agent/master/whatap_control_plane_helper"]
    args: ["-collect_control_plane_monitoring_enabled=true"]
  - name: whatap-master-agent
    image: whatap/kube_mon
    imagePullPolicy: Always
    resources:
      requests:
        memory: 300Mi
        cpu: 100m
      limits:
        memory: 350Mi
        cpu: 200m
    ports:
      - containerPort: 6600
    command: ['/bin/entrypoint.sh']
    env:
      - name: collect_control_plane_monitoring_enabled
        value: "true"
      - name: WHATAP_LICENSE
        value: "xxxxxxxx-12345xxxxxx-x00e000000xxx8"
      - name: WHATAP_HOST
        value: "13.124.11.223/13.209.172.35"
      - name: WHATAP_PORT
        value: "6600"
      - name: WHATP_MEM_LIMIT
        valueFrom:
          resourceFieldRef:
            containerName: whatap-master-agent
            resource: limits.memory
    volumeMounts:
      - name: start-script-volume
        mountPath: /bin/entrypoint.sh
        readOnly: true
        subPath: entrypoint.sh
      - mountPath: /whatap_conf
```

```
          name: whatap-config-volume
      volumes:
        - name: start-script-volume
          configMap:
            defaultMode: 0700
            name: master-start-script
        - name: whatap-config-volume
          emptyDir: {}
      serviceAccount: whatap
```

3. 컨트롤 플레인 모니터링 성능 지표를 수집하기 위해 `whatap` ClusterRole에 다음의 권한을 추가하세요.

```
rules:
  - nonResourceURLs: ["/metrics"]
    verbs: ["*"]
```

다음 예시를 참조하세요.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  annotations:
    rbac.authorization.kubernetes.io/autoupdate: "true"
  name: whatap
  namespace: whatap-monitoring
rules:
  - apiGroups: ["*"]
    resources: ["namespaces", "pods", "pods/log", "configmaps","services",
"endpoints","daemonsets","ingresses","deployments","nodes","persistentvolumes","persistentvolumeclaims",
"events", "replicasets", "roles", "rolebindings", "clusterroles", "clusterrolebindings", "jobs", "cronjobs",
"statefulsets", "serviceaccounts", "configmaps", "storageclasses", "horizontalpodautoscalers" ]
    verbs: ["get","list","watch"]
  - apiGroups: [""]
    resources: ["pods/exec"]
    verbs: ["create"]
  - apiGroups: [""]
    resources: ["configmaps"]
    verbs: ["*"]
```

```
- nonResourceURLs: ["/metrics"]
  verbs: ["*"]
---
```

ⓘ kubectl apply 또는 kubectl edit 명령을 통해 kube-apiserver 모니터링을 활성화할 수도 있습니다.

kube-apiserver 메트릭

분류	메트릭	설명
API GET 요청	분당 GET 요청 현황	kube-apiserver 의 GET 요청에 대한 분당 요청 수
	초당 GET 요청 수 추이	kube-apiserver 의 GET 요청에 대한 초당 요청 수 추이
	Resource & Subresource 별 분당 GET 요청 수 추이	kube-apiserver 에 들어온 GET 요청의 Resource 와 Subresource 별 분당 요청 수 추이
API POST 요청	분당 POST 요청 현황	kube-apiserver 의 POST 요청에 대한 분당 요청 수
	초당 POST 요청 수 추이	kube-apiserver 의 POST 요청에 대한 초당 요청 수 추이
	Resource & Subresource 별 POST 분당 요청 수 추이	kube-apiserver 에 들어온 POST 요청에 대해 Resource 와 Subresource 별로 그룹화 한 분당 요청 수 추이
API 응답 코드별 요청	[2xx] 분당 응답 요청 현황	kube-apiserver 요청 중 200 이상 300 미만의 응답 코드가 내려진 요청 수에 대한 분당 현황
	[2xx] 분당 응답 코드별 요청 수 추이	kube-apiserver 요청 중 200 이상 300 미만의 응답 코드가 내려진 요청 수에 대한 응답 코드별 분당 현황
	[2xx] 분당 Resource & Subresource 별 요청 수 추이	kube-apiserver 요청 중 200 이상 300 미만의 응답 코드가 내려진 요청에 대해 Resource 와 Subresource 별로 그룹화 한 분당 요청 수 추이

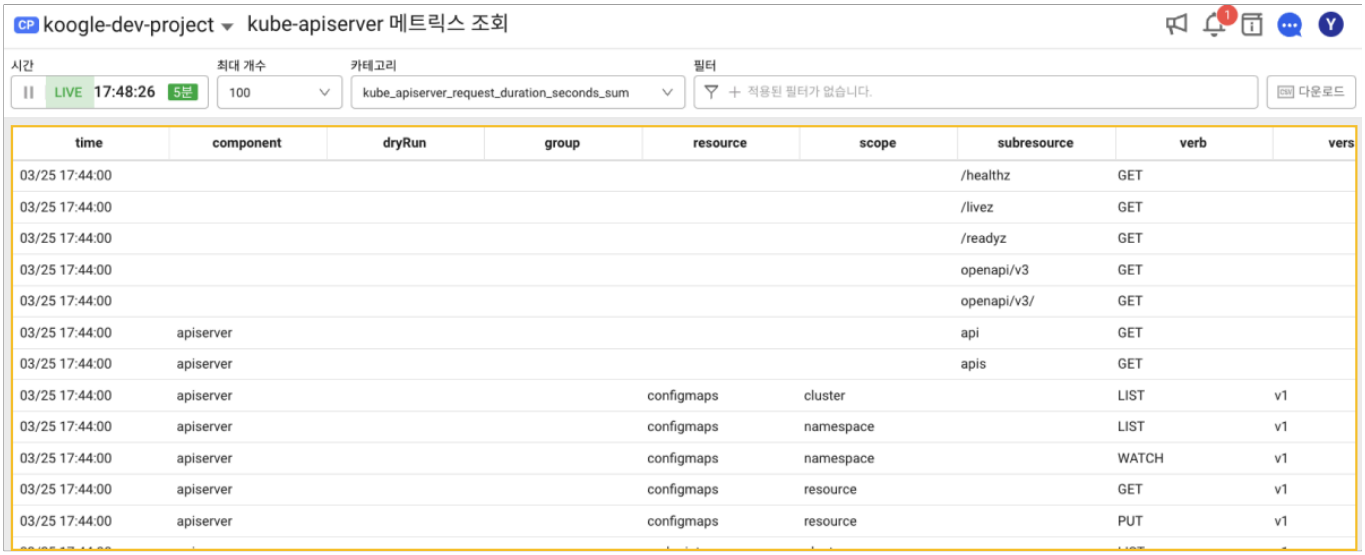
분류	메트릭	설명
	[3xx] 분당 응답 요청 현황	kube-apiserver 요청 중 300 이상 400 미만의 응답 코드가 내려진 요청 수에 대한 분당 현황
	[3xx] 분당 응답 코드별 요청 수 추이	kube-apiserver 요청 중 300 이상 400 미만의 응답 코드가 내려진 요청 수에 대한 응답 코드별 분당 현황
	[3xx] 분당 Resource & Subresource 별 요청 수 추이	kube-apiserver 요청 중 300 이상 400 미만의 응답 코드가 내려진 요청에 대해 Resource 와 Subresource 별로 그룹화 한 분당 요청 수 추이
	[400~] 분당 응답 요청 현황	kube-apiserver 요청 중 400 이상의 응답 코드가 내려진 요청 수에 대한 분당 현황
	[400~] 분당 응답 코드별 요청 수 추이	kube-apiserver 요청 중 400 이상의 응답 코드가 내려진 요청 수에 대한 응답 코드별 분당 현황
	[400~] 분당 Resource & Subresource 별 요청 수 추이	kube-apiserver 요청 중 400 이상의 응답 코드가 내려진 요청에 대해 Resource 와 Subresource 별로 그룹화 한 분당 요청 수 추이
Go Metrics	고루틴 수	현재 존재하는 고루틴 수
	스레드 수	현재 존재하는 OS 스레드 수
현재 실행 중인 요청 수 (1초)	최근 1초 내 실행 중인 요청 수	데이터 수집 시 kube-apiserver 에서 1초 내에 진행되고 있던 요청 수

kube-apiserver 메트릭스 조회

> **요구 사항**
>
> - 와탭 쿠버네티스 에이전트 1.5.6 버전 이상이 필요합니다.
> - 쿠버네티스 클러스터 프로젝트(CP)일 경우 해당 메뉴를 조회할 수 있습니다.

kube-apiserver는 **컨트롤 플레인**(Control Plane) 내/외부 요청을 처리하는 컴포넌트입니다. **kube-apiserver 메트릭스 조회** 메뉴를 통해 kube-apiserver에서 수집되는 지표의 원본 데이터 정합성(Data Consistency)을 확인할 수 있습니다.

기본 화면 안내



time	component	dryRun	group	resource	scope	subresource	verb	vers
03/25 17:44:00						/healthz	GET	
03/25 17:44:00						/livez	GET	
03/25 17:44:00						/readyz	GET	
03/25 17:44:00						openapi/v3	GET	
03/25 17:44:00						openapi/v3/	GET	
03/25 17:44:00	apiserver					api	GET	
03/25 17:44:00	apiserver					apis	GET	
03/25 17:44:00	apiserver			configmaps	cluster		LIST	v1
03/25 17:44:00	apiserver			configmaps	namespace		LIST	v1
03/25 17:44:00	apiserver			configmaps	namespace		WATCH	v1
03/25 17:44:00	apiserver			configmaps	resource		GET	v1
03/25 17:44:00	apiserver			configmaps	resource		PUT	v1

kube-apiserver 메트릭스 조회 메뉴는 kube-apiserver 대상으로 수집한 모니터링 원본 데이터를 **카테고리**별로 조회하는 기능을 제공합니다. 최근 5분 동안 수집한 데이터가 1분 간격으로 반영됩니다.

- 상단 왼쪽 **시간 선택자**를 통해 과거 특정 시점의 데이터를 조회할 수 있습니다. 과거 데이터 조회 시 원하는 옵션 선택 후 **검색** 아이콘을 클릭하세요.
- 조회할 **최대 개수**를 설정할 수 있습니다.
- 메트릭스를 필터링해 원본 데이터를 필요에 따라 가공할 수 있습니다.

- 상단 오른쪽 **다운로드** 아이콘을 선택해 원하는 데이터를 CSV 형식의 파일로 다운로드할 수 있습니다.

필터

필터 옵션 활용 시 원하는 을 선택해 조회할 수 있습니다. `and` 조건을 통해 여러 개의 필터를 적용할 수 있습니다.

1. **필터** 입력 창에서 + **더하기 아이콘** 선택 시 다음과 같이 **필터 추가하기** 창이 나타납니다.

   - **필터 키**와 **조건**을 선택하세요.
   - 조건에 맞는 값을 선택하세요.
2. 필터 설정 후 **적용** 버튼을 선택해 해당 필터를 적용하세요.
3. 여러 개의 필터를 추가하려면 **필터** 입력 창 또는 **필터 추가하기** 창에서 + **더하기 아이콘**을 선택하세요.

kube-apiserver 모니터링 활성화

와탭 쿠버네티스 에이전트 설치 시 다운로드한 yaml 파일을 다음 안내에 따라 수정해 kube-apiserver 모니터링을 활성화하세요.

1. `whatap-master-agent` deployment의 spec 하위에 kube-apiserver 모니터링을 위한 에이전트 컨테이너 spec을 추가하세요.

```
containers:
  - name: whatap-control-plane-helper
    image: whatap/kube_mon
    imagePullPolicy: Always
    ports:
      - containerPort: 9496
```

```
  command: ["/data/agent/master/whatap_control_plane_helper"]
  args: ["-collect_control_plane_monitoring_enabled=true"]
```

다음 예시를 참조하세요.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: whatap-master-agent
  namespace: whatap-monitoring
spec:
  replicas: 1
  selector:
    matchLabels:
      name: whatap-master-agent
  template:
    metadata:
      labels:
        name: whatap-master-agent
    spec:
      containers:
        - name: whatap-control-plane-helper
          image: whatap/kube_mon
          imagePullPolicy: Always
          ports:
            - containerPort: 9496
          command: ["/data/agent/master/whatap_control_plane_helper"]
          args: ["-collect_control_plane_monitoring_enabled=true"]
        - name: whatap-master-agent
          image: whatap/kube_mon
          imagePullPolicy: Always
          resources:
            requests:
              memory: 300Mi
              cpu: 100m
            limits:
              memory: 350Mi
              cpu: 200m
          ports:
```

```
      - containerPort: 6600
      command: ['/bin/entrypoint.sh']
      env:
        - name: WHATAP_LICENSE
          value: "xxxxxxxx-12345xxxxxx-x00e000000xxx8"
        - name: WHATAP_HOST
          value: "13.124.11.223/13.209.172.35"
        - name: WHATAP_PORT
          value: "6600"
        - name: WHATP_MEM_LIMIT
          valueFrom:
            resourceFieldRef:
              containerName: whatap-master-agent
              resource: limits.memory
      volumeMounts:
        - name: start-script-volume
          mountPath: /bin/entrypoint.sh
          readOnly: true
          subPath: entrypoint.sh
        - mountPath: /whatap_conf
          name: whatap-config-volume
    volumes:
      - name: start-script-volume
        configMap:
          defaultMode: 0700
          name: master-start-script
      - name: whatap-config-volume
        emptyDir: {}
    serviceAccount: whatap
```

2. `whatap-master-agent` deployment의 `whatap-master-agent` 컨테이너 spec에서 env 설정에 다음 내용을 추가하세요.

```
env:
  - name: collect_control_plane_monitoring_enabled
    value: "true"
```

다음 예시를 참조하세요.

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: whatap-master-agent
  namespace: whatap-monitoring
spec:
  replicas: 1
  selector:
    matchLabels:
      name: whatap-master-agent
  template:
    metadata:
      labels:
        name: whatap-master-agent
    spec:
      containers:
        - name: whatap-control-plane-helper
          image: whatap/kube_mon
          imagePullPolicy: Always
          ports:
            - containerPort: 9496
          command: ["/data/agent/master/whatap_control_plane_helper"]
          args: ["-collect_control_plane_monitoring_enabled=true"]
        - name: whatap-master-agent
          image: whatap/kube_mon
          imagePullPolicy: Always
          resources:
            requests:
              memory: 300Mi
              cpu: 100m
            limits:
              memory: 350Mi
              cpu: 200m
          ports:
            - containerPort: 6600
          command: ['/bin/entrypoint.sh']
          env:
            - name: collect_control_plane_monitoring_enabled
              value: "true"
```

```
        - name: WHATAP_LICENSE
          value: "xxxxxxxx-12345xxxxxx-x00e000000xxx8"
        - name: WHATAP_HOST
          value: "13.124.11.223/13.209.172.35"
        - name: WHATAP_PORT
          value: "6600"
        - name: WHATP_MEM_LIMIT
          valueFrom:
            resourceFieldRef:
              containerName: whatap-master-agent
              resource: limits.memory
      volumeMounts:
        - name: start-script-volume
          mountPath: /bin/entrypoint.sh
          readOnly: true
          subPath: entrypoint.sh
        - mountPath: /whatap_conf
          name: whatap-config-volume
    volumes:
      - name: start-script-volume
        configMap:
          defaultMode: 0700
          name: master-start-script
      - name: whatap-config-volume
        emptyDir: {}
    serviceAccount: whatap
```

3. 컨트롤 플레인 모니터링 성능 지표를 수집하기 위해 `whatap` ClusterRole에 다음의 권한을 추가하세요.

```
rules:
  - nonResourceURLs: ["/metrics"]
    verbs: ["*"]
```

다음 예시를 참조하세요.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
```

```
  annotations:
    rbac.authorization.kubernetes.io/autoupdate: "true"
  name: whatap
  namespace: whatap-monitoring
rules:
  - apiGroups: ["*"]
    resources: ["namespaces", "pods", "pods/log", "configmaps","services",
"endpoints","daemonsets","ingresses","deployments","nodes","persistentvolumes","persistentvolumeclaims",
"events", "replicasets", "roles", "rolebindings", "clusterroles", "clusterrolebindings", "jobs", "cronjobs",
"statefulsets", "serviceaccounts", "configmaps", "storageclasses", "horizontalpodautoscalers" ]
    verbs: ["get","list","watch"]
  - apiGroups: [""]
    resources: ["pods/exec"]
    verbs: ["create"]
  - apiGroups: [""]
    resources: ["configmaps"]
    verbs: ["*"]
  - nonResourceURLs: ["/metrics"]
    verbs: ["*"]
---
```

ⓘ `kubectl apply` 또는 `kubectl edit` 명령을 통해 kube-apiserver 모니터링을 활성화할 수도 있습니다.

kube-apiserver 메트릭

분류	메트릭	설명
API GET 요청	분당 GET 요청 현황	kube-apiserver 의 GET 요청에 대한 분당 요청 수
	초당 GET 요청 수 추이	kube-apiserver 의 GET 요청에 대한 초당 요청 수 추이
	Resource & Subresource 별 분당 GET 요청 수 추이	kube-apiserver 에 들어온 GET 요청의 Resource 와 Subresource 별 분당 요청 수 추이

분류	메트릭	설명
API POST 요청	분당 POST 요청 현황	kube-apiserver 의 POST 요청에 대한 분당 요청 수
	초당 POST 요청 수 추이	kube-apiserver 의 POST 요청에 대한 초당 요청 수 추이
	Resource & Subresource 별 POST 분당 요청 수 추이	kube-apiserver 에 들어온 POST 요청에 대해 Resource 와 Subresource 별로 그룹화 한 분당 요청 수 추이
API 응답 코드별 요청	[2xx] 분당 응답 요청 현황	kube-apiserver 요청 중 200 이상 300 미만의 응답 코드가 내려진 요청 수에 대한 분당 현황
	[2xx] 분당 응답 코드별 요청 수 추이	kube-apiserver 요청 중 200 이상 300 미만의 응답 코드가 내려진 요청 수에 대한 응답 코드별 분당 현황
	[2xx] 분당 Resource & Subresource 별 요청 수 추이	kube-apiserver 요청 중 200 이상 300 미만의 응답 코드가 내려진 요청에 대해 Resource 와 Subresource 별로 그룹화 한 분당 요청 수 추이
	[3xx] 분당 응답 요청 현황	kube-apiserver 요청 중 300 이상 400 미만의 응답 코드가 내려진 요청 수에 대한 분당 현황
	[3xx] 분당 응답 코드별 요청 수 추이	kube-apiserver 요청 중 300 이상 400 미만의 응답 코드가 내려진 요청 수에 대한 응답 코드별 분당 현황
	[3xx] 분당 Resource & Subresource 별 요청 수 추이	kube-apiserver 요청 중 300 이상 400 미만의 응답 코드가 내려진 요청에 대해 Resource 와 Subresource 별로 그룹화 한 분당 요청 수 추이
	[400~] 분당 응답 요청 현황	kube-apiserver 요청 중 400 이상의 응답 코드가 내려진 요청 수에 대한 분당 현황
	[400~] 분당 응답 코드별 요청 수 추이	kube-apiserver 요청 중 400 이상의 응답 코드가 내려진 요청 수에 대한 응답 코드별 분당 현황
	[400~] 분당 Resource & Subresource 별 요청 수 추이	kube-apiserver 요청 중 400 이상의 응답 코드가 내려진 요청에 대해 Resource 와 Subresource 별로 그룹화 한 분당 요청 수 추이
Go Metrics	고루틴 수	현재 존재하는 고루틴 수

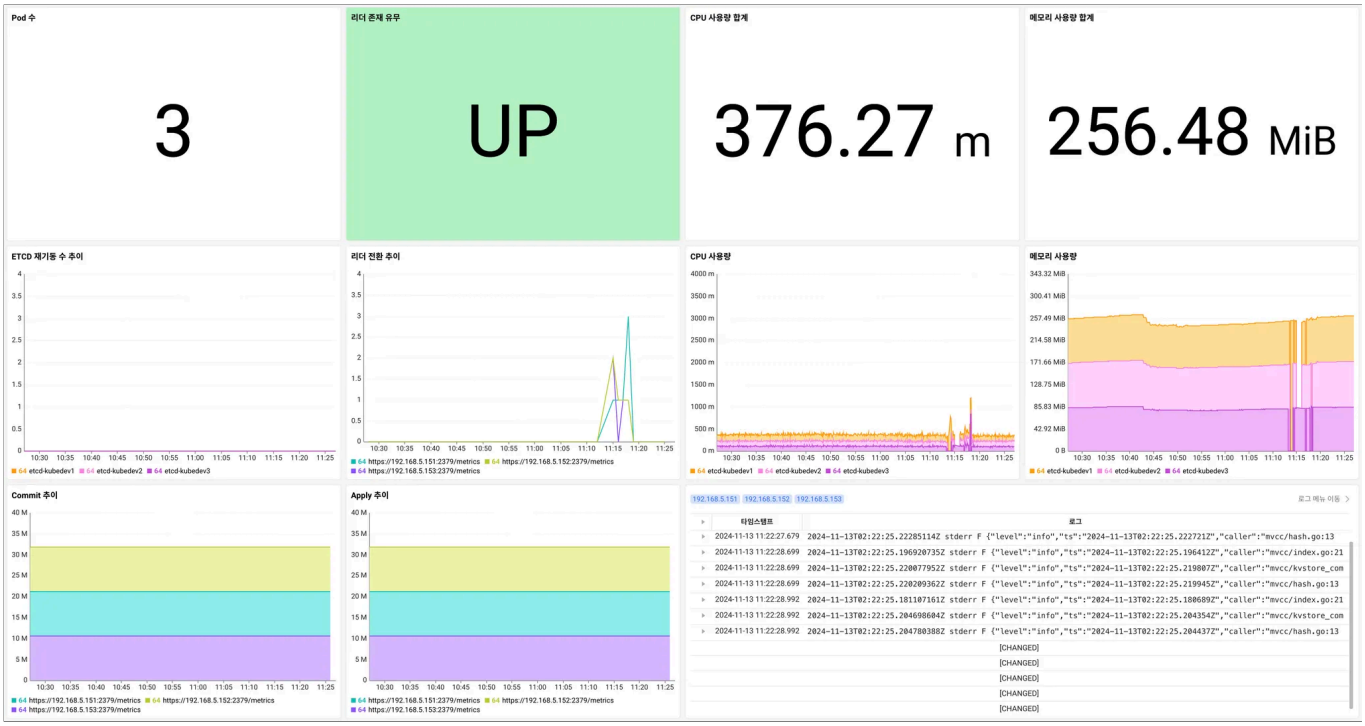
분류	메트릭	설명
	스레드 수	현재 존재하는 OS 스레드 수
현재 실행 중인 요청 수 (1초)	최근 1초 내 실행 중인 요청 수	데이터 수집 시 kube-apiserver 에서 1초 내에 진행되고 있던 요청 수

etcd 대시보드

> **요구 사항**
>
> - 와탭 쿠버네티스 에이전트 1.7.16 버전 이상이 필요합니다.
> - 쿠버네티스 클러스터 프로젝트(CP)일 경우 해당 메뉴를 조회할 수 있습니다.

etcd는 쿠버네티스 환경에서 클러스터 상태와 관련된 데이터를 저장하는 핵심 컴포넌트입니다. 와탭은 etcd의 성능과 상태를 모니터링할 수 있는 **etcd 대시보드**를 제공합니다. 이를 통해 클러스터의 안정성을 높이고 문제 발생 시 신속한 대응이 가능합니다.

기본 화면 안내



etcd 대시보드는 최근 1시간 동안 수집된 주요 etcd 지표를 1분 주기로 제공합니다. 대시보드에서 확인할 수 있는 주요 지표는 다음과 같습니다. 지표 상세는 하단 메트릭 안내를 참조하세요.

- etcd가 사용하는 리소스, Pod 재기동 수, 상세 로그를 제공합니다.

- etcd 내부의 commit 및 apply 횟수를 제공합니다.
- etcd 리더의 존재 여부 및 리더의 변경 이력을 제공합니다.

etcd 모니터링 활성화

와탭 쿠버네티스 에이전트 설치 시 다운로드한 YAML 파일 수정 및 mTLS 인증서를 등록하여 에이전트를 활성화하세요.

인증서 확인 및 Secret 생성

etcd 인증서 확인

etcd의 mTLS 인증서가 필요합니다. etcd 인증서는 일반적으로 Control Plane 노드의 /etc/kubernetes/pki/etcd 경로에 저장됩니다. 파일 소유자는 root 계정이며 필요한 인증서는 다음과 같습니다.

- ca.crt
- server.crt
- server.key

`kubectl` 명령을 실행할 머신에서 다음 명령어를 통해 인증서가 존재하는지 확인하세요.

```
sudo ls /etc/kubernetes/pki/etcd
```

Secret 생성

인증서가 존재하면 다음 명령어를 통해 Secret을 생성하세요.

```
sudo kubectl create secret generic whatap-etcd-cert -n whatap-monitoring --from-file=ca.crt=/etc/kubernetes/pki/etcd/ca.crt
--from-file=server.crt=/etc/kubernetes/pki/etcd/server.crt --from-file=server.key=/etc/kubernetes/pki/etcd/server.key
```

정상적으로 Secret 생성 시 다음과 같은 메시지가 출력됩니다.

```
secret/whatap-etcd-cert created
```

root 계정을 사용할 수 없는 경우

root 계정에서 `kubectl` 명령을 사용할 수 없는 경우 다음 안내를 참조하세요.

1. 인증서 복제 후 접근 가능한 경로로 이동하세요.
2. 복제된 인증서 파일의 소유자를 현재 소유자로 변경하세요.

네임스페이스가 없을 경우

쿠버네티스 에이전트 YAML 적용 전에 Secret을 만드는 경우 `whatap-monitoring` 네임스페이스가 존재하지 않아 Secret 생성에 실패할 수 있습니다. 다음 명령어를 통해 네임스페이스를 생성한 후 다시 Secret을 생성하세요.

```
kubectl create namespace whatap-monitoring
```

와탭 쿠버네티스 에이전트 YAML 수정

와탭 쿠버네티스 에이전트 YAML 파일을 수정하여 컨테이너 설정을 추가 후 Secret을 연결하세요.

컨테이너 설정

1. Deployment의 `whatap-control-plane-helper` 컨테이너 내 args 섹션에 etcd 관련 설정을 추가하세요.

   ```
   -collect_etcd_monitoring_enabled=true
   -etcd_hosts=사용자의_etcd_ip,사용자의_etcd_ip
   ```

   다음 예시를 참조하세요.

   Example

   ```
   -etcd_hosts=192.168.1.45,192.168.1.46,192.168.1.47
   ```

2. Deployment의 `whatap-master-agent` 컨테이너의 환경 변수(env)에 다음 내용을 추가하세요.

   ```
   - name: "collect_etcd_monitoring_enabled"
     value: "true"
   ```

Secret 연결

1. Deployment의 `whatap-master-agent` 내 volumes 섹션에 Secret을 추가하세요.

```
- name: "whatap-etcd-cert"
  secret:
    secretName: "whatap-etcd-cert"
```

2. `whatap-control-plane-helper` 컨테이너에 해당 Secret을 마운트하세요.

```
volumeMounts:
- name: "whatap-etcd-cert"
    readOnly: true
    mountPath: "/etc/kubernetes/pki/etcd"
```

전체 YAML 예시

다음은 모든 설정이 적용된 YAML 파일 예시입니다. 하이라이팅된 부분을 참조하세요.

```
apiVersion: "apps/v1"
kind: "Deployment"
metadata:
  name: "whatap-master-agent"
  namespace: "whatap-monitoring"
spec:
  replicas: 1
  selector:
    matchLabels:
      name: "whatap-master-agent"
  template:
    metadata:
      labels:
        name: "whatap-master-agent"
    spec:
      containers:
      - command:
        - "/bin/entrypoint.sh"
```

```
    env:
    - name: "WHATAP_LICENSE"
      value: "x208gnc8i1kko-z2p19shk7npc4t-z788sgbj3n123a"
    - name: "WHATAP_HOST"
      value: "59.13.101.109"
    - name: "WHATAP_PORT"
      value: "61574"
    - name: "WHATP_MEM_LIMIT"
      valueFrom:
        resourceFieldRef:
          containerName: "whatap-master-agent"
          resource: "limits.memory"
    - name: "collect_control_plane_monitoring_enabled"
      value: "true"
    - name: "collect_etcd_monitoring_enabled"
      value: "true"
    image: "whatap/kube_mon"
    name: "whatap-master-agent"
    ports:
    - containerPort: 6600
    resources:
      limits:
        cpu: "200m"
        memory: "350Mi"
      requests:
        cpu: "100m"
        memory: "300Mi"
    volumeMounts:
    - mountPath: "/bin/entrypoint.sh"
      name: "start-script-volume"
      readOnly: true
      subPath: "entrypoint.sh"
    - mountPath: "/whatap_conf"
      name: "whatap-config-volume"
  - args:
    - "-collect_control_plane_monitoring_enabled=true"
    - "-collect_etcd_monitoring_enabled=true"
    - "-etcd_hosts=192.168.1.45,192.168.1.46,192.168.1.47"
    command:
    - "/data/agent/master/whatap_control_plane_helper"
    image: "whatap/kube_mon"
```

```
    imagePullPolicy: "Always"
    name: "whatap-control-plane-helper"
    ports:
    - containerPort: 9496
    volumeMounts:
      - name: "whatap-etcd-cert"
        readOnly: true
        mountPath: "/etc/kubernetes/pki/etcd"
  serviceAccount: "whatap"
  volumes:
  - configMap:
      defaultMode: 448
      name: "master-start-script"
    name: "start-script-volume"
  - emptyDir: {}
    name: "whatap-config-volume"
  - name: "whatap-etcd-cert"
    secret:
      secretName: "whatap-etcd-cert"
```

etcd 메트릭

구분	설명
Pod 수	현재 존재하는 etcd 파드의 수
etcd 재기동 수 추이	etcd 파드 재기동 횟수 합계 추이
리더 변경 추이	etcd 파드 리더 변경 횟수 추이
리더 존재 유/무	리더 역할을 하는 etcd 파드의 존재 여부
CPU 사용량 합계	etcd 파드 CPU 사용량 합계
CPU 사용량	etcd 파드 CPU 사용량 추이
Memory 사용량 합계	etcd 파드 메모리 사용량 합계

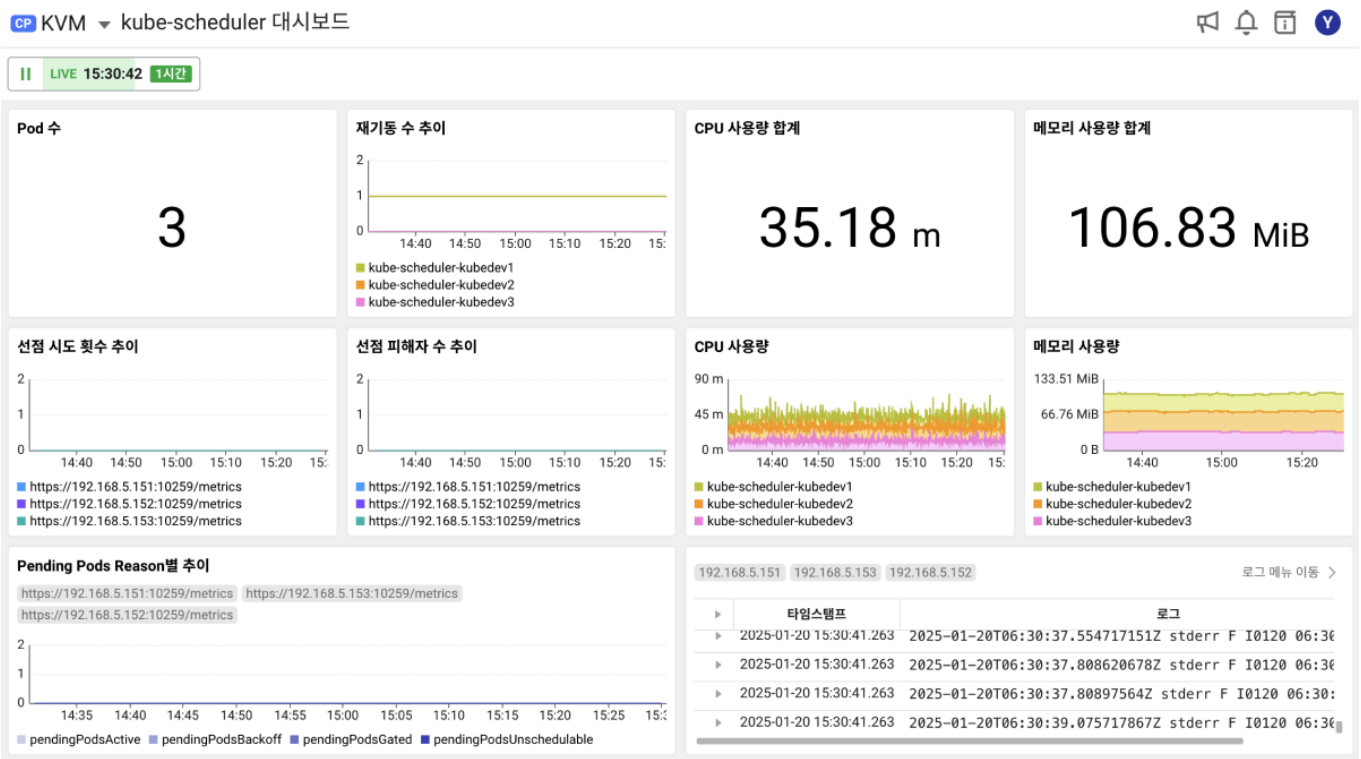
구분	설명
Memory 사용량	etcd 파드 메모리 사용량 추이
Commit 추이	etcd 변경 사항 commit 횟수
Apply 추이	commit 이후 apply 된 횟수
etcd 로그	etcd 파드에서 발생한 로그

kube-scheduler 대시보드

> **요구 사항**
>
> - 와탭 쿠버네티스 에이전트 1.8.1 버전 이상이 필요합니다.
> - 쿠버네티스 클러스터 프로젝트(CP)일 경우 해당 메뉴를 조회할 수 있습니다.
> - 바닐라 쿠버네티스 또는 `kube-scheduler` Pod를 관측할 수 있는 클러스터일 경우 해당 메뉴를 사용할 수 있습니다.

`kube-scheduler`는 쿠버네티스 환경에서 Pod의 스케줄링을 담당하는 주요 컴포넌트입니다. 와탭은 `kube-scheduler`의 성능, 스케줄링 상태를 실시간으로 모니터링할 수 있는 **kube-scheduler 대시보드**를 제공합니다. **kube-scheduler 대시보드**를 통해 클러스터가 스케줄링 불능 상태에 빠지거나 스케줄러의 부하로 인한 문제를 사전에 방지할 수 있습니다.

기본 화면 안내



kube-scheduler 대시보드는 최근 1시간 동안 수집된 주요 지표를 1분 단위로 시각화하여 제공합니다. 대시보드에서 확인할 수 있는 주요 지표는 다음과 같습니다. 지표 상세는 하단 메트릭 안내를 참조하세요.

- `kube-scheduler`가 사용하는 리소스 추이, Pod 재기동 횟수 그리고 상세 로그를 제공합니다.
- `kube-scheduler`의 선점 시도 횟수 및 선점으로 인한 피해자 수 추이를 제공합니다.
- `kube-scheduler` Pod 내부 Pending Queue에 할당된 Pending Pod의 Reason별 추이를 제공합니다.

kube-scheduler 모니터링 활성화

활성화 요구 조건

- `kubectl get pods -n kube-system -l component=kube-scheduler` 명령어를 통해 `kube-scheduler` Pod를 관측할 수 있어야 합니다.
- `kube-scheduler`와의 보안 통신을 위해 `--bind-address` 설정을 외부에서 접근할 수 있도록 변경해야 합니다. 일반적으로 `0.0.0.0`을 사용합니다.

활성화 과정

> ⓘ **활성화 순서**
>
> 1. whatap service account의 RBAC에 `secrets`에 대한 `GET` 권한을 추가하세요.
> 2. whatap service account를 기반으로 `secrets`을 생성하세요.
> 3. `whatap-master-agent` deployment에 `kube-scheduler` 모니터링 활성화 옵션을 적용하세요.

1. **권한 설정**

   `whatap-monitoring` 네임스페이스 내 whatap ClusterRole의 `rules`에서 접근 가능한 리소스에 `secrets` 자원을 추가하세요. 다음 내용을 yaml 파일로 생성 후 `kubectl apply -f <yaml파일명>.yaml`를 실행하여 적용하세요.

   ClusterRole.yaml

```
apiVersion: "rbac.authorization.k8s.io/v1"
kind: "ClusterRole"
metadata:
  annotations:
    rbac.authorization.kubernetes.io/autoupdate: "true"
  name: "whatap"
  namespace: "whatap-monitoring"
rules:
- apiGroups:
  - "*"
  resources:
  - "namespaces"
  - "pods"
  - "pods/log"
  - "configmaps"
  - "services"
  - "endpoints"
  - "daemonsets"
  - "ingresses"
  - "deployments"
  - "nodes"
  - "persistentvolumes"
  - "persistentvolumeclaims"
  - "events"
  - "secrets"
  - "replicasets"
  - "roles"
  - "rolebindings"
  - "clusterroles"
  - "clusterrolebindings"
  - "jobs"
  - "cronjobs"
  - "statefulsets"
  - "serviceaccounts"
  - "configmaps"
  - "storageclasses"
  - "horizontalpodautoscalers"
  - "replicationcontrollers"
  verbs:
```

```
  - "get"
  - "list"
  - "watch"
- apiGroups:
  - ""
  resources:
  - "pods/exec"
  verbs:
  - "create"
- apiGroups:
  - ""
  resources:
  - "configmaps"
  verbs:
  - "*"
- nonResourceURLs:
  - "/metrics"
  verbs:
  - "*"
```

2. **Secret 생성**

   다음 내용을 yaml 파일로 생성 후 `kubectl apply -f yaml파일명.yaml` 를 실행하여 적용하세요.

   secrets.yaml

```
apiVersion: v1
kind: Secret
type: kubernetes.io/service-account-token
metadata:
  name: whatap-scheduler-monitoring-token
  namespace: whatap-monitoring
  annotations:
    kubernetes.io/service-account.name: "whatap"
```

3. **Deployment 수정**

   `whatap-master-agent` deployment의 내용을 수정한 후 `kubectl apply -f <yaml파일명>.yaml` 명령어를 통해 적용하세요.

- **환경 변수 추가**

  `containers` 항목 하위의 `whatap-master-agent` 컨테이너의 `env` 섹션에 다음 내용을 추가하세요.

```
env:
- name: "collect_kube_scheduler_monitoring_enabled"
  value: "true"
```

- **명령어 인자 추가**

  `containers` 항목 하위의 `whatap-control-plane-helper` 컨테이너의 `args` 섹션에 다음 내용을 추가하세요.

```
- args:
  - "-collect_kube_scheduler_monitoring_enabled=true"
```

환경 변수 및 명령어 인자 추가 시 다음 예시를 참조하세요.

```
apiVersion: "apps/v1"
kind: "Deployment"
metadata:
  name: "whatap-master-agent"
  namespace: "whatap-monitoring"
spec:
  replicas: 1
  selector:
    matchLabels:
      name: "whatap-master-agent"
  template:
    metadata:
      labels:
        name: "whatap-master-agent"
    spec:
      priorityClassName: high-priority
      containers:
      - command:
        - "/bin/entrypoint.sh"
        env:
```

```
    - name: "WHATAP_LICENSE"
      value: "c1024123i1kko-241245127abdbe-612632344235"
    - name: "WHATAP_HOST"
      value: "192.168.1.0"
    - name: "WHATP_MEM_LIMIT"
      valueFrom:
        resourceFieldRef:
          containerName: "whatap-master-agent"
          resource: "limits.memory"
    - name: "collect_kube_scheduler_monitoring_enabled"
      value: "true"
    image: "whatap/kube_mon"
    name: "whatap-master-agent"
    ports:
    - containerPort: 6600
    resources:
      limits:
        cpu: "200m"
        memory: "350Mi"
      requests:
        cpu: "100m"
        memory: "300Mi"
    volumeMounts:
    - mountPath: "/bin/entrypoint.sh"
      name: "start-script-volume"
      readOnly: true
      subPath: "entrypoint.sh"
    - mountPath: "/whatap_conf"
      name: "whatap-config-volume"
  - args:
    - "-collect_kube_scheduler_monitoring_enabled=true"
    command:
    - "/data/agent/master/whatap_control_plane_helper"
    image: "whatap/kube_mon"
    imagePullPolicy: "Always"
    name: "whatap-control-plane-helper"
    ports:
    - containerPort: 9496
  serviceAccount: "whatap"
```

```
volumes:
- configMap:
    defaultMode: 448
    name: "master-start-script"
  name: "start-script-volume"
- emptyDir: {}
  name: "whatap-config-volume"
```

kube-scheduler 메트릭

구분	설명
Pod 수	현재 실행 중인 `kube-scheduler` Pod의 수
kube-scheduler 재기동 수 추이	`kube-scheduler` Pod의 재기동 횟수 합계 추이
CPU 사용량 합계	`kube-scheduler` Pod의 총 CPU 사용량
CPU 사용량	`kube-scheduler` Pod의 CPU 사용량 추이
Memory 사용량 합계	`kube-scheduler` Pod의 총 메모리 사용량
Memory 사용량	`kube-scheduler` Pod의 메모리 사용량 추이
kube-scheduler 로그	`kube-scheduler` Pod에서 발생한 로그
선점 시도 횟수 추이	Pod 스케줄링을 위해 선점을 시도한 횟수 추이
선점 피해자 수 추이	선점으로 인해 축출된 Pod 수의 추이
Pending Pod Reason별 추이	스케줄링되지 않고 Pending 상태로 대기 중인 Pod의 원인별 추이

오브젝트 매니페스트

> **요구 사항**
>
> 와탭 쿠버네티스 에이전트 1.5.5 버전 이상이 필요합니다. 1.5.5 버전 미만일 경우 기존 **마스터 메타 정보** 화면이 제공됩니다.

쿠버네티스 마스터 에이전트는 클러스터에 있는 Pod, 서비스, ReplicaSet 등의 오브젝트에 대한 매니페스트를 하루에 한 번씩 자동으로 수집합니다. 해당 데이터는 날짜별로 저장되어 사용자가 특정 날짜 기준의 클러스터 상태를 쉽게 확인할 수 있습니다. 수집한 데이터는 날짜별 비교가 가능해 시간에 따른 클러스터의 변화를 파악하기 용이합니다.

또한 사용자는 쿠버네티스 마스터 에이전트의 Deployment 배포 전 컨테이너 환경변수를 설정해 수집 시간의 타임존을 자유롭게 지정할 수 있습니다. 해당 설정 방법은 다음 컨테이너 환경변수 안내를 참조하세요.

컨테이너 환경변수

- **store_k8s_object_manifest_enabled**

  기본값 `true`

  오브젝트 매니페스트 수집 기능을 활성화합니다. 기본값이 `true` 이기 때문에 별도 설정 없이 수집 기능이 활성화됩니다.

- **store_k8s_object_manifest_timezone**

  기본값 `UTC`

  오브젝트 매니페스트 수집을 위한 타임존을 설정합니다. 별도 설정이 없을 경우 기본값 `UTC` 기준 자정에 수집을 시작합니다.

  - 지역 기반 타임존 또는 GMT 오프셋 형식으로 설정할 수 있습니다.
  - 분 단위의 정밀한 시간 지정을 지원합니다.

    예, `GMT+0810` 과 같이 설정해 특정 지역의 비표준 시간대 지정 가능

- **store_k8s_object_manifest_max_fetch_limit**

  기본값 `50`

  클러스터에서 한 번에 요청할 오브젝트 개수를 설정합니다. 별도 설정이 없을 경우 기본값 `50` 기준으로 1회 요청당 50개씩

요청합니다.

예, 전체 Pod 개수가 90개인 클러스터에서 해당 옵션 `30` 설정 시 Pod 정보를 30개씩 3회 요청

Example

```
containers:
      - name: whatap-master-agent
        image: whatap/kube_mon
        resources:
          requests:
            memory: 300Mi
            cpu: 100m
          limits:
            memory: 350Mi
            cpu: 200m
        ports:
          - containerPort: 6600
        command: ['/bin/entrypoint.sh']
        env:
          - name: WHATAP_LICENSE
            value: {PROJECT_LICENSE_VALUE}
          - name: WHATAP_HOST
            value: "13.124.11.223/13.209.172.35"
          - name: WHATAP_PORT
            value: "6600"
          - name: WHATP_MEM_LIMIT
            valueFrom:
              resourceFieldRef:
                containerName: whatap-master-agent
                resource: limits.memory
          # 추가된 환경 변수
          - name: store_k8s_object_manifest_enabled
            value: "true"
          - name: store_k8s_object_manifest_timezone
            value: "GMT+9"
          - name: store_k8s_object_manifest_max_fetch_limit
            value: "50"
```

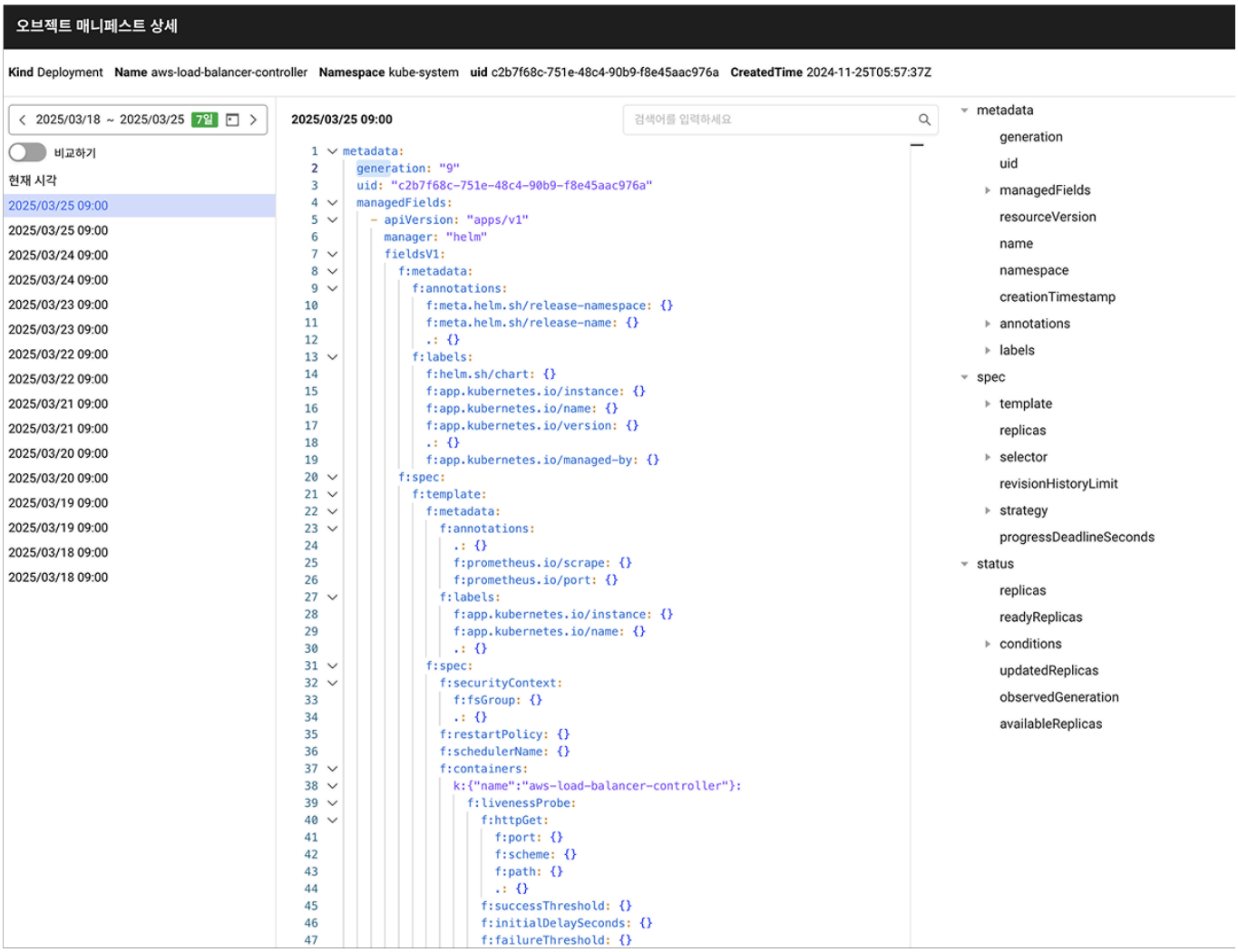
기본 화면 안내

오브젝트 목록

오브젝트 종류 목록에서 원하는 오브젝트 선택 시 해당 오브젝트의 목록을 확인할 수 있습니다. **Name**, **Namespace**, **Uid**, **CreatedTime** 등의 오브젝트 정보가 표시됩니다. 다음 화면은 오브젝트 종류 목록에서 **Deployment** 선택 시 예시 화면입니다.

오브젝트 매니페스트 상세

오브젝트 목록에서 **상세 보기** 버튼 선택 시 **오브젝트 매니페스트 상세** 창에서 선택한 오브젝트의 매니페스트를 조회할 수 있습니다.



오브젝트 매니페스트 저장 시간 목록

- 해당 오브젝트의 매니페스트가 저장된 시간 목록을 좌측에서 확인할 수 있습니다. 오브젝트 매니페스트는 매일 정해진 시간에 자동으로 저장됩니다.
- 오브젝트 매니페스트 저장 시간은 마스터 에이전트의 컨테이너 환경변수 `store_k8s_object_manifest_timezone` 값을 수정해 변경할 수 있습니다.
- 오브젝트 매니페스트는 매일 정해진 저장 시간에 저장될 뿐 아니라 경고 이벤트가 발생한 경우에도 추가로 저장됩니다.

> ⓘ 오브젝트 매니페스트가 저장되는 경고 이벤트는 Warning 타입의 쿠버네티스 이벤트 오브젝트를 의미합니다.

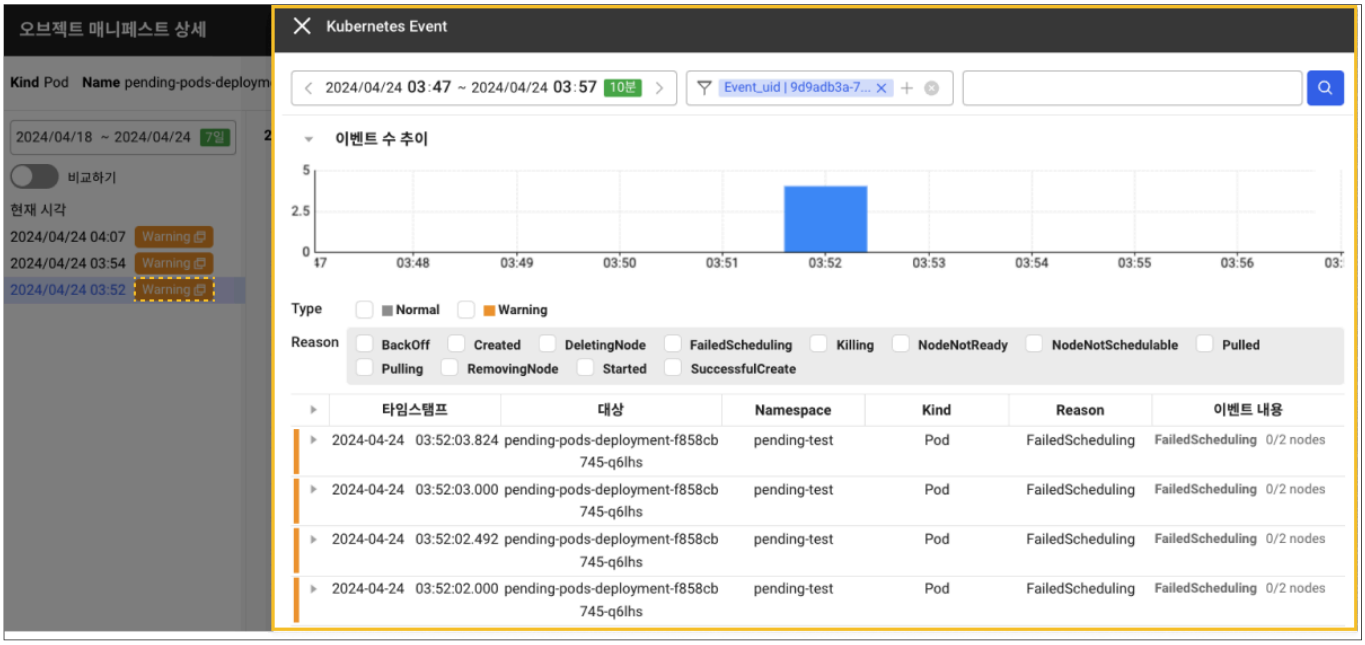
오브젝트 매니페스트 검색과 탐색

오브젝트 매니페스트 상세 창에서 오브젝트 매니페스트 내 특정 키워드를 찾을 수 있도록 검색 기능을 제공합니다. 검색창에 입력한 문자열은 오브젝트 매니페스트 내에 하이라이팅됩니다. 이를 통해 복잡한 매니페스트 구조에서 특정 설정이나 필드를 빠르게 찾을 수 있습니다.

또한 **오브젝트 매니페스트 상세** 창 오른쪽에 매니페스트 전체 구조를 트리 형태로 요약한 탐색 패널을 제공합니다. 이를 통해 대용량 매니페스트 문서를 다룰 때 별도 스크롤 없이 주요 항목을 확인할 수 있습니다.

쿠버네티스 이벤트 연동

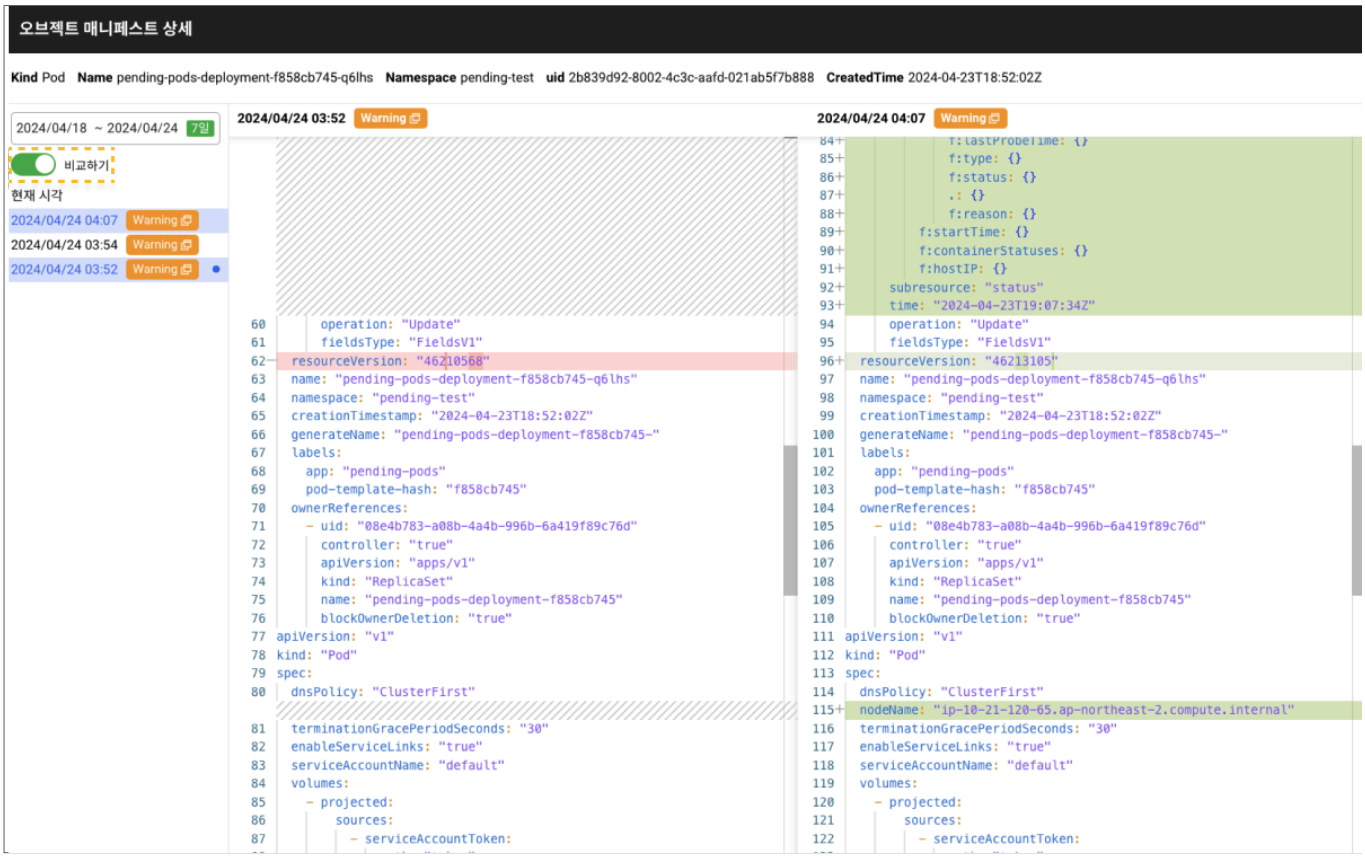
오브젝트 매니페스트 상세 창에서 Warning 버튼 클릭 시 다음과 같이 쿠버네티스 이벤트를 확인할 수 있습니다.



> ⓘ 쿠버네티스 이벤트 연동 기능은 와탭 마스터 에이전트 버전 1.6.1 이상에서 지원합니다.

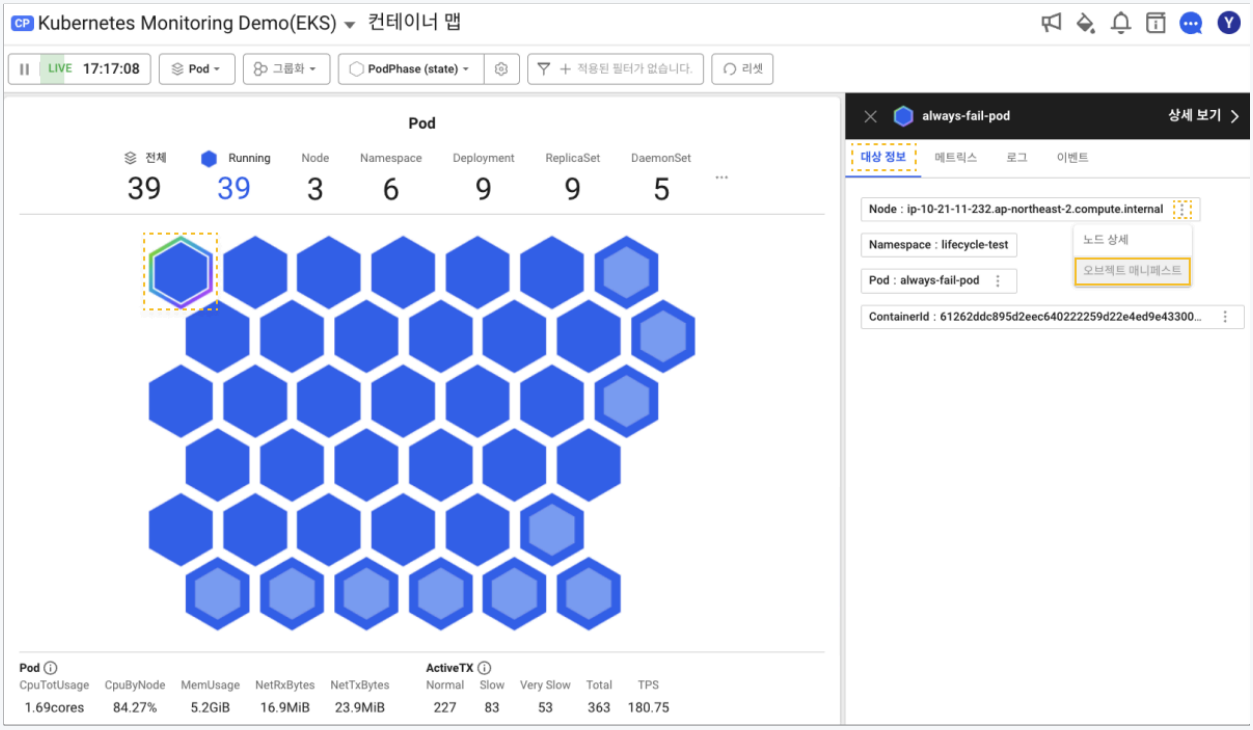
오브젝트 매니페스트 비교

비교하기 토글 활성화 시 저장 시간 목록에서 선택해 다음과 같이 두 개의 오브젝트 매니페스트를 비교할 수 있습니다. 변경 사항이 있을 경우 하이라이팅됩니다.



컨테이너 맵에서 오브젝트 매니페스트 조회

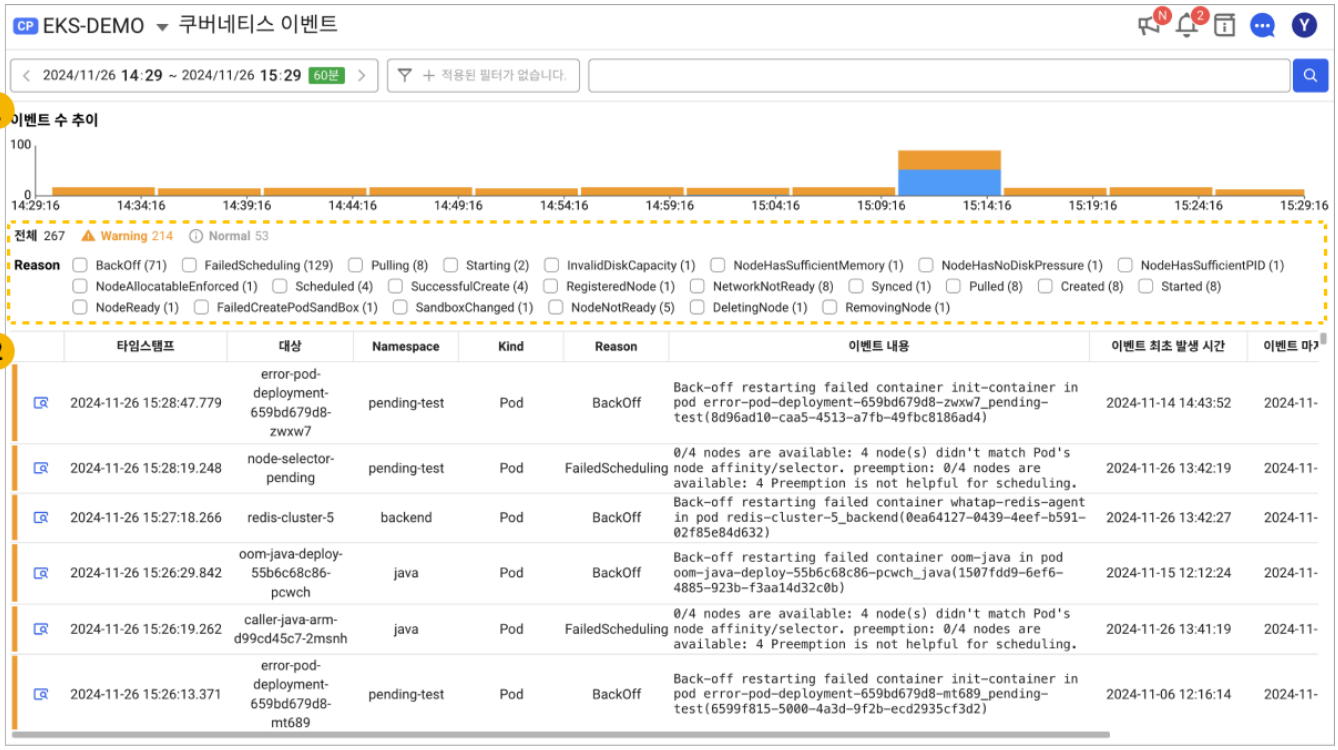
컨테이너 맵에서 **오브젝트 매니페스트**를 확인할 수 있습니다.



- 원하는 블록을 선택하세요.
- 요약보기 창에서 **대상 정보** 탭을 선택해 Pod 또는 컨테이너 정보를 확인하세요.
- **Pod**, **Deployment**, **ReplicaSet**, **Node** 태그의 오른쪽 ⋮ **더보기** 아이콘을 클릭해 **오브젝트 매니페스트**를 선택하세요.

쿠버네티스 이벤트

와탭 **쿠버네티스 이벤트** 메뉴를 통해 명령어 없이 쿠버네티스 이벤트에 대한 다양한 조회와 검색이 가능합니다. 사용자의 클러스터 이벤트를 확인할 수 있으며 시간 선택자를 통해 과거 이벤트 역시 조회할 수 있습니다.



- 필터 옵션 및 검색창을 통해 원하는 이벤트를 확인할 수 있습니다.
- Normal 또는 Warning과 같은 이벤트 유형 및 **Reason** 기준으로 빠르게 필터링할 수 있습니다.

> ⓘ • 쿠버네티스 이벤트에 대한 자세한 내용은 다음 문서를 참조하세요.

이벤트 수 추이

바 차트 형식으로 제공되는 1 **이벤트 수 추이**를 통해 선택한 시간 범위 내에서 발생한 이벤트 수의 변화를 시각적으로 확인할 수 있습니다.

조회를 원하는 시간 구간을 선택하여 해당 구간에 발생한 이벤트를 이벤트 목록에서 바로 확인할 수 있습니다. 이를 통해 이벤트가 집중적으로 발생한 시간대를 빠르게 식별하고 상세 내용을 분석할 수 있습니다.

이벤트 목록

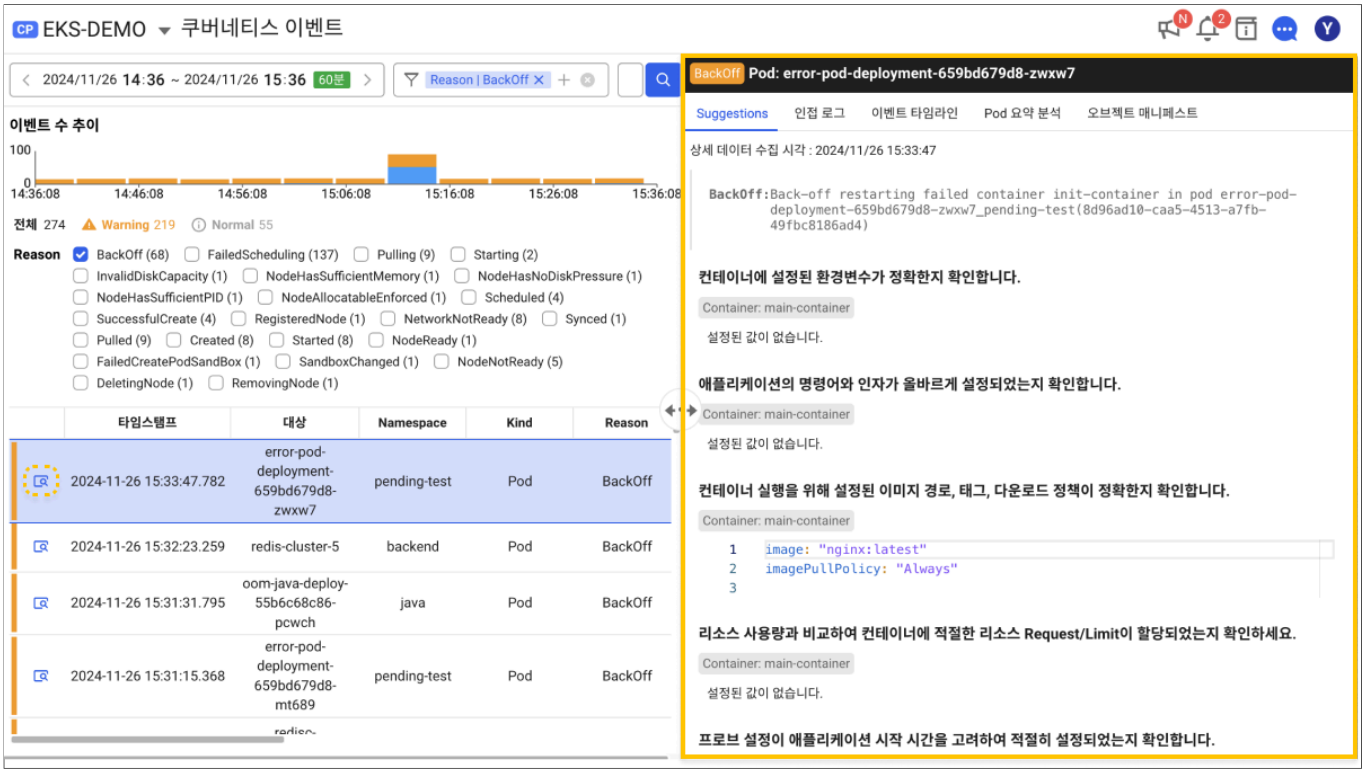
특정 시간대에 발생한 쿠버네티스 이벤트 목록을 2 영역에서 확인할 수 있습니다. 와탭은 최대 30일간의 이벤트 데이터를 저장하며 이를 통해 과거 이벤트를 조회하고 분석할 수 있습니다.

또한 목록 상단에서 이벤트 Type(Warning, Normal) 또는 **Reason**을 선택하여 원하는 이벤트를 빠르게 필터링할 수 있습니다.

항목	설명
타임스탬프	이벤트가 발생한 시간입니다.
대상	이벤트가 발생한 대상입니다. Kind가 Pod인 경우 Pod 이름, Node인 경우 Node 이름을 나타냅니다.
Namespace	이벤트가 발생한 대상이 속한 네임스페이스입니다.
Kind	이벤트가 발생한 대상의 종류입니다.
Reason	이벤트가 발생한 이유입니다.
이벤트 최초 발생 시간	이벤트가 처음 관찰된 시간입니다.
이벤트 발생 건수	일정 시간 동안 지속적으로 발생한 이벤트의 횟수입니다.
이벤트 내용	구체적인 이벤트의 상세 정보입니다.

이벤트 상세

이벤트 목록에서 조회를 원하는 이벤트의 왼쪽 **상세** 아이콘 선택 시 **이벤트 상세**를 확인할 수 있습니다.



Suggestions

Suggestions 탭은 발생한 이벤트 메시지를 기반으로 이벤트 발생 원인을 분석하고 문제를 추적하기 위한 방향을 제시합니다.

> ⓘ 현재 Warning 타입의 **BackOff**, **NodNotReady** 이벤트만 제공되는 기능입니다.

인접 로그

선택한 이벤트 발생 시점을 기준으로 인접한 로그 조회를 통해 이벤트 발생 전후 맥락을 확인할 수 있습니다. 시간 선택 버튼을 통해 원하는 인접 범위를 설정할 수 있습니다.

> ⓘ **로그 조회 권한**
>
> - **로그 조회 권한**이 있는 경우, 인접된 모든 로그를 확인할 수 있습니다.
> - **로그 조회 권한**이 없는 경우, 인접된 쿠버네티스 이벤트만 확인 가능합니다.

이벤트 타임라인

선택한 이벤트의 대상(**Kind**)이 생성된 이후부터 현재까지 발생한 모든 이벤트를 타임라인 형태로 제공합니다. 이를 통해 대상의 이벤트 내역을 한눈에 파악할 수 있습니다. 이를 통해 대상 리소스의 생성, 정상 동작, 경고 발생, 복구 등 라이프사이클 전반을 한눈에 파악할 수 있습니다.

요약 분석

이벤트가 발생한 대상인 Pod 또는 Node의 성능과 자원 상태를 한눈에 파악할 수 있도록 요약 정보를 제공합니다.

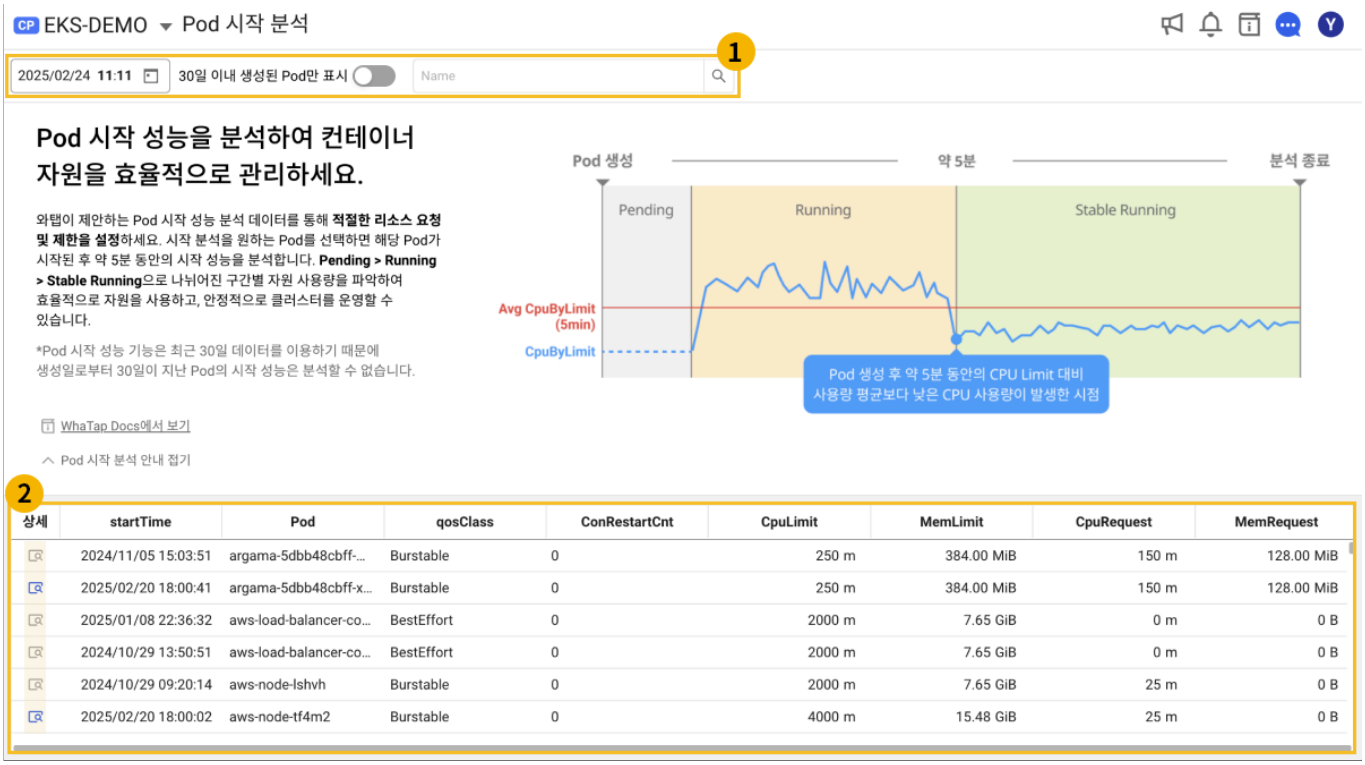
Pod 시작 분석

> ⓘ 와탭 노드 에이전트 버전 **1.4.2 이상**에서 지원합니다. 해당 버전 미만인 경우 기존 메뉴(구 **Pod 초기화 성능**) 화면을 제공합니다.

원활한 클러스터 운영을 위해 Pod가 안정적으로 동작하기까지의 성능을 분석하고 Pod에 할당된 리소스 크기를 최적화할 필요가 있습니다. **Pod 시작 분석** 메뉴는 Pod가 시작한 후 약 5분 동안의 변화를 추적합니다. **Pending** 상태에서 **Running** 상태로, 그리고 안정적인 **Stable Running** 상태로 이어지는 3단계 구간의 자원 사용량을 분석해 리소스 할당량을 최적화할 수 있습니다.

이를 통해 운영자는 자원 효율성 향상 및 비용 절감, 확장성 향상, 스케일 용이성 등의 이점을 얻을 수 있습니다.

기본 화면 안내



상단 옵션 영역

상단 1 영역에서 다음과 같은 옵션을 활용할 수 있습니다.

- 시간 선택자를 통해 시작 시간을 선택할 수 있습니다. 조회를 원하는 시작 시간 선택 시 2 영역의 Pod 목록에서 데이터를 확인할 수 있습니다.
- **30일 이내 생성된 Pod만 표시** 토글을 활성화해 2 영역의 Pod 목록에서 30일 이내의 Pod만을 조회할 수 있습니다.
- Pod 검색 입력창에 문자를 입력해 원하는 Pod를 검색할 수 있습니다.

> - **Pod 시작 분석** 메뉴는 시간 선택자 기능을 와탭 쿠버네티스 에이전트 1.7.6 버전 이상부터 제공합니다.
> - **Pod 시작 분석** 메뉴는 Pod 생성일로부터 최근 30일 이내의 범위 안에서 5분간의 Pod 시작 데이터를 분석합니다. 즉 생성 후 30일이 지난 Pod의 경우 시작 성능을 분석할 수 없습니다.

Pod 목록 영역

2 영역에서 Pod 목록을 확인할 수 있습니다. Pod 목록이 펼쳐진 테이블의 헤더 컬럼은 다음의 정보를 포함하고 있습니다.

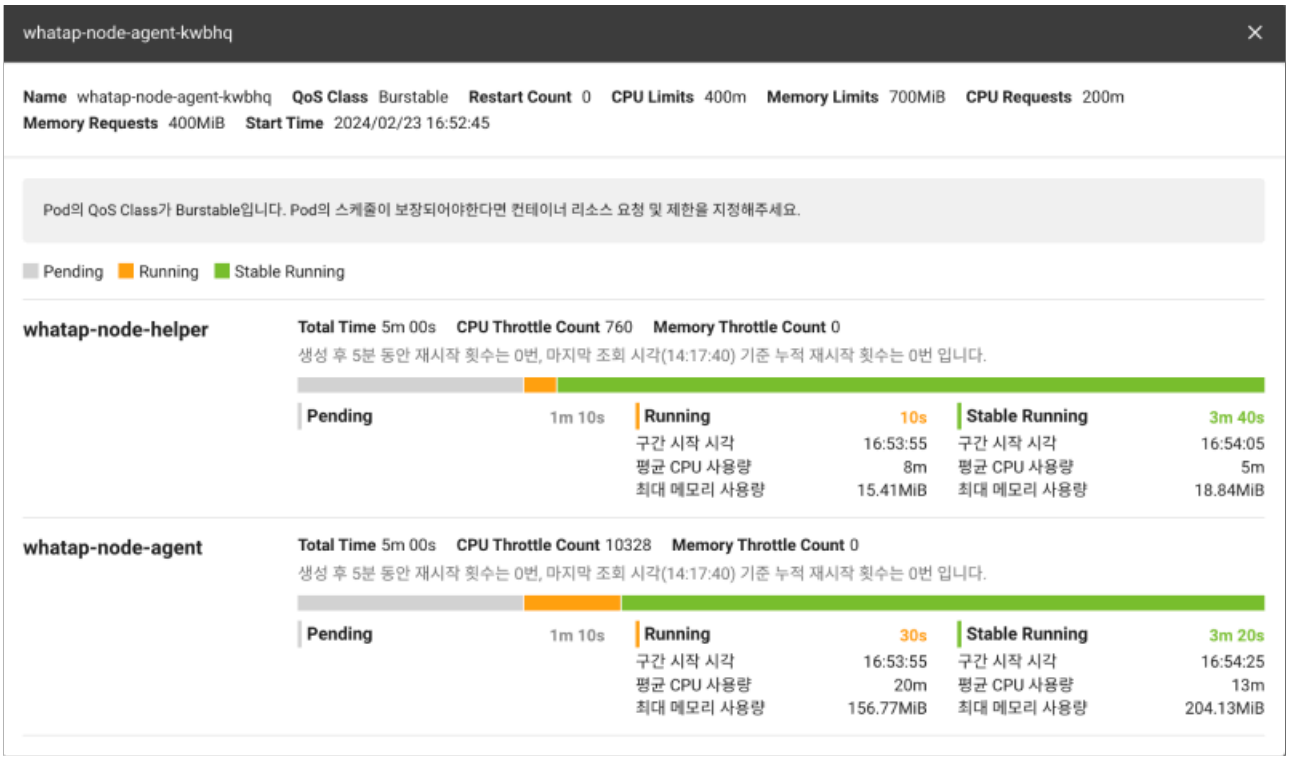
- **QoS Class**: Pod의 Quality of Service를 의미합니다. QoS 클래스는 리소스 부족 시 Pod의 우선 순위를 결정합니다. 우선 순위가 높은 Pod가 우선 순위가 낮은 Pod보다 먼저 종료됩니다.
  - Guaranteed: 노드에서 사용 가능한 리소스를 보장받고 다른 Pod보다 우선적으로 할당합니다. 가장 마지막에 종료됩니다.
    - Pod의 모든 컨테이너가 CPU 및 메모리 request를 가지고 있는 경우입니다.
    - Pod의 모든 컨테이너가 CPU 및 메모리 limit을 가지고 있으며 해당 값이 request와 동일한 경우입니다.
  - Burstable: Pod는 노드 리소스가 부족한 경우에는 일부 리소스를 할당합니다.
    - Guaranteed 클래스의 조건을 충족하지 못하는 경우입니다.
    - Pod의 컨테이너 중 하나라도 CPU 및 메모리에 대한 request 또는 limit이 설정된 경우입니다.
  - BestEffort: 리소스 request나 limit이 없기에 다른 Pod보다 우선 순위가 낮습니다. 가장 먼저 종료됩니다.
    - Pod의 모든 컨테이너가 CPU및 메모리에 대한 request와 limit이 없는 경우입니다.
- **Restart Count**: Pod 내 컨테이너의 총 재시작 횟수입니다.
- **CPU/Memory Limits**: Pod 내 컨테이너에 설정된 CPU/Memory Limits 총합(`resources.limits`)입니다.

- **CPU/Memory Requests**: Pod 내 컨테이너에 설정된 CPU/Memory Requests 총합(`resources.requests`)입니다.
- **Start Time**: Pod 생성 시각(`metadata.startTime`)입니다.

> - QoS Class에 대한 자세한 내용은 다음 링크를 참조하세요.
> - Pod 및 컨테이너 리소스 관린에 대한 자세한 내용은 다음 링크를 참조하세요.

상세 보기 안내

2 영역 Pod 목록에서 분석을 원하는 항목의 가장 왼쪽에 **상세 보기** 버튼을 선택하세요. 선택한 Pod의 컨테이너별 시작 성능을 구간별로 분석한 내용을 다음과 같이 확인할 수 있습니다.

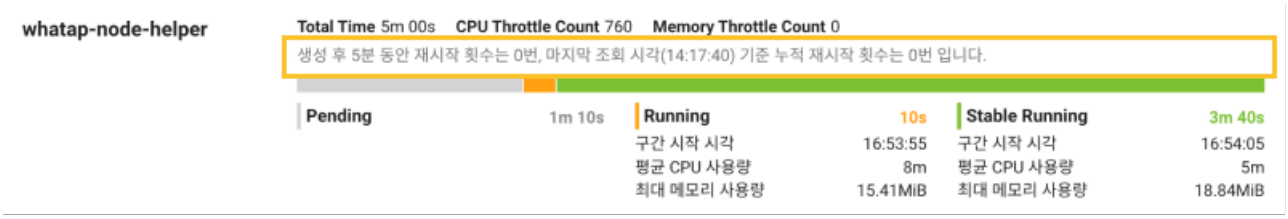


- **Pending**: Pod를 생성한 후 **Running** 상태로 진입하는 구간입니다.
- **Running**: Pod가 **Pending** 상태를 지나 안정화되기 전까지의 구간입니다.
- **Stable Running**: Pod가 **Running** 상태를 지나 안정화된 구간입니다.

> **안정화된 상태(Stable Running)**
>
> 와탭에서는 Pod 생성 직후 약 5분 동안 컨테이너의 CPU Limit 대비 사용량의 평균을 계산합니다. 이 평균값보다 CPU 사용량이 적어지는 시점을 안정화 시작 상태라고 정의합니다.

상세 분석하기

- 5분 동안 한 컨테이너가 여러 번 재시작했다면 가장 처음 시작한 컨테이너의 정보만 표기합니다. 컨테이너별 재시작 횟수를 확인하세요. 재시작 횟수가 많다면 해당 컨테이너가 정상 실행되고 있는지 점검하세요.



- **Pending** 구간이 길어진다면 Pod가 **Running** 상태로 진입하기까지 오랜 시간이 걸렸다는 의미입니다. Pod의 **Pending** 상태를 유발할 수 있는 요인이 있는지 점검해 보세요.

> Pod의 **pending** 상태에 대한 자세한 내용은 다음 링크를 참조하세요.

- **Running** 구간이 길다면 Pod 내 컨테이너가 처음 실행될 때 리소스 사용량이 많은 구간이 있다는 의미입니다. 컨테이너 실행 과정에 문제가 없는지 살펴보세요.
- 전반적인 초기 자원 사용량이 많다면 컨테이너 리소스 요청 및 제한을 늘리는 것을 고려해 보세요.

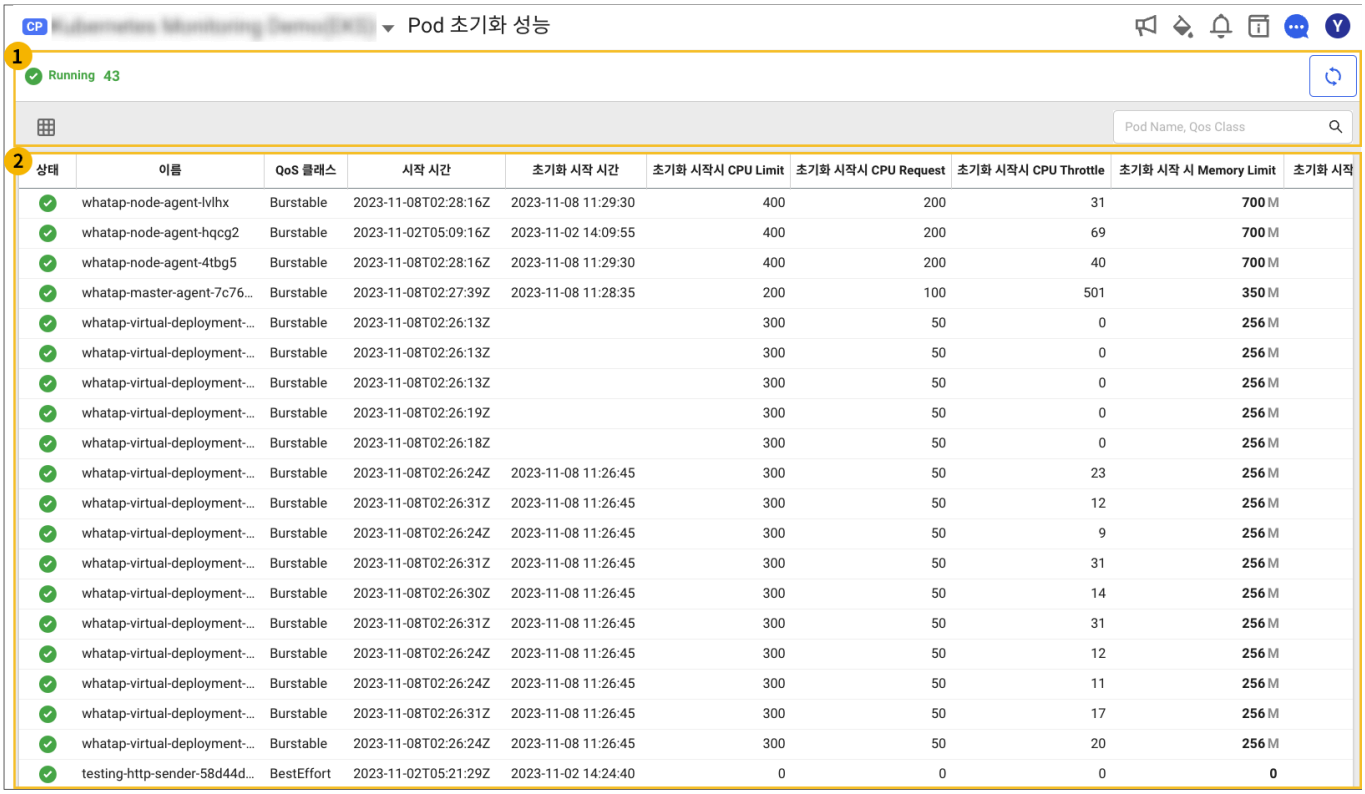
Pod 시작 분석

> **와탭 노드 에이전트 버전 1.4.2 미만**
>
> - 신규 Pod 시작 분석 메뉴 대신 기존 메뉴(구 **Pod 초기화 성능**) 화면을 제공합니다.
> - 클러스터 프로젝트(CP)인 경우만 접근할 수 있습니다.

Springboot를 포함한 JVM 기반 애플리케이션은 초기화 시점에 많은 자원을 사용해 Liveness 체크에 실패하는 경우가 있습니다. 만약 일정 횟수 이상 Liveness 체크에 실패하는 경우 해당 Pod가 삭제되고 무한 재시작 현상이 발생하여 배포 실패로 이어질 수 있습니다.

와탭 쿠버네티스의 **Pod 시작 분석** 기능을 활용하면 Pod 시작 시 피크 사용량과 초기화 시간을 분석하여 최적의 Initial Delay(Liveness 체크 면제 시간) 설정에 도움을 받을 수 있습니다.

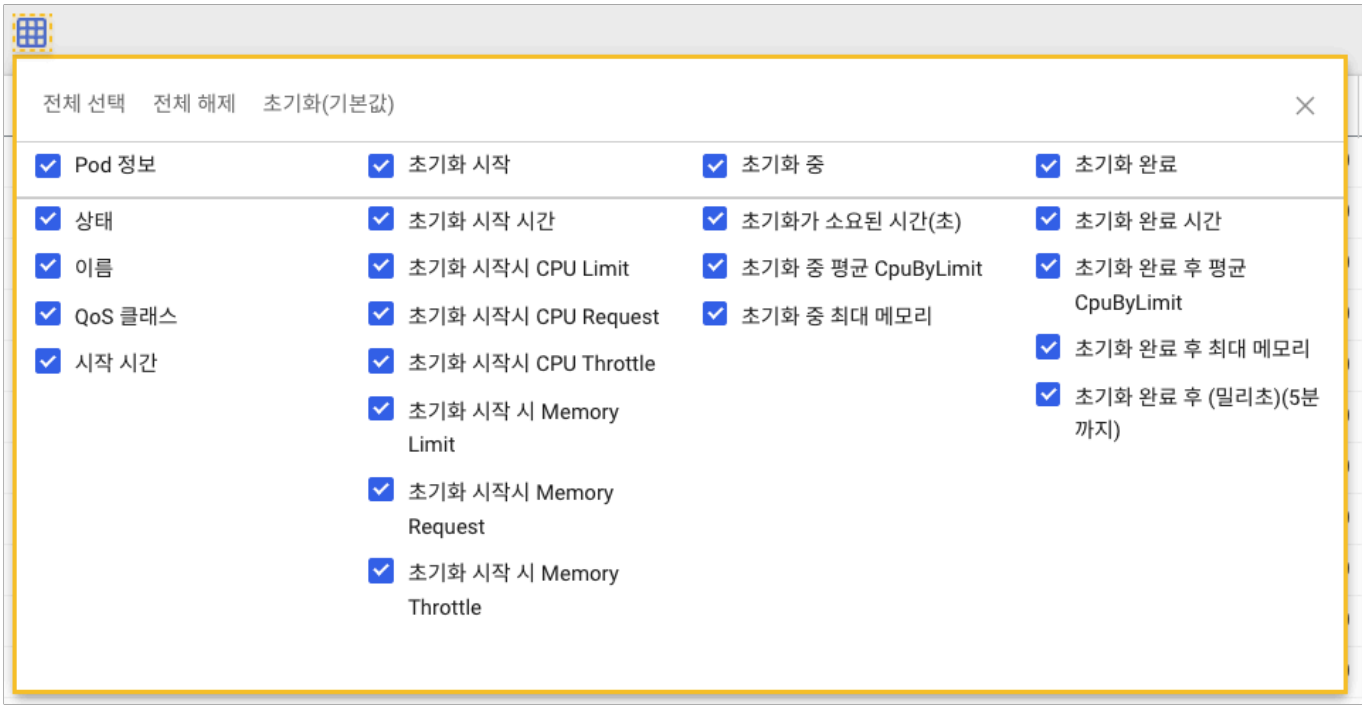
기본 화면 안내



상태	이름	QoS 클래스	시작 시간	초기화 시작 시간	초기화 시작시 CPU Limit	초기화 시작시 CPU Request	초기화 시작시 CPU Throttle	초기화 시작 시 Memory Limit	초기화 시작
	whatap-node-agent-lvlhx	Burstable	2023-11-08T02:28:16Z	2023-11-08 11:29:30	400	200	31	700 M	
	whatap-node-agent-hqcg2	Burstable	2023-11-02T05:09:16Z	2023-11-02 14:09:55	400	200	69	700 M	
	whatap-node-agent-4tbg5	Burstable	2023-11-08T02:28:16Z	2023-11-08 11:29:30	400	200	40	700 M	
	whatap-master-agent-7c76...	Burstable	2023-11-08T02:27:39Z	2023-11-08 11:28:35	200	100	501	350 M	
	whatap-virtual-deployment-...	Burstable	2023-11-08T02:26:13Z		300	50	0	256 M	
	whatap-virtual-deployment-...	Burstable	2023-11-08T02:26:13Z		300	50	0	256 M	
	whatap-virtual-deployment-...	Burstable	2023-11-08T02:26:13Z		300	50	0	256 M	
	whatap-virtual-deployment-...	Burstable	2023-11-08T02:26:19Z		300	50	0	256 M	
	whatap-virtual-deployment-...	Burstable	2023-11-08T02:26:18Z		300	50	0	256 M	
	whatap-virtual-deployment-...	Burstable	2023-11-08T02:26:24Z	2023-11-08 11:26:45	300	50	23	256 M	
	whatap-virtual-deployment-...	Burstable	2023-11-08T02:26:31Z	2023-11-08 11:26:45	300	50	12	256 M	
	whatap-virtual-deployment-...	Burstable	2023-11-08T02:26:24Z	2023-11-08 11:26:45	300	50	9	256 M	
	whatap-virtual-deployment-...	Burstable	2023-11-08T02:26:31Z	2023-11-08 11:26:45	300	50	31	256 M	
	whatap-virtual-deployment-...	Burstable	2023-11-08T02:26:30Z	2023-11-08 11:26:45	300	50	14	256 M	
	whatap-virtual-deployment-...	Burstable	2023-11-08T02:26:31Z	2023-11-08 11:26:45	300	50	31	256 M	
	whatap-virtual-deployment-...	Burstable	2023-11-08T02:26:24Z	2023-11-08 11:26:45	300	50	12	256 M	
	whatap-virtual-deployment-...	Burstable	2023-11-08T02:26:24Z	2023-11-08 11:26:45	300	50	11	256 M	
	whatap-virtual-deployment-...	Burstable	2023-11-08T02:26:31Z	2023-11-08 11:26:45	300	50	17	256 M	
	whatap-virtual-deployment-...	Burstable	2023-11-08T02:26:24Z	2023-11-08 11:26:45	300	50	20	256 M	
	testing-http-sender-58d44d...	BestEffort	2023-11-02T05:21:29Z	2023-11-02 14:24:40	0	0	0	0	

- 1 영역 좌측 상단에서 상태별 Pod 개수를 확인할 수 있습니다.
- 1 영역 우측 상단에서 **새로 고침** 아이콘을 선택해 새로 고침할 수 있습니다.
- 1 영역 좌측에서 **컬럼 선택** 아이콘을 선택해 컬럼을 선택할 수 있습니다.
- 1 영역 우측 입력창에서 Pod **이름** 또는 **QoS 클래스**를 검색할 수 있습니다.
- 2 영역에서 원하는 컬럼을 선택 시 해당 컬럼 기준으로 목록을 정렬해 조회할 수 있습니다.

컬럼 안내



Pod 시작 분석 목록은 다음과 같이 **Pod 정보**와 Pod 초기화 타임라인에 따라 컬럼을 분류할 수 있습니다.

Pod 정보

Pod의 **이름**과 **상태**, **QoS 클래스**, **시작 시간**에 대한 정보를 확인할 수 있습니다.

- **상태**: Pod의 라이프 사이클을 의미합니다.
  - ⏸ Pending: 쿠버네티스 클러스터 내부에서 Pod 시작이 승인되었지만 컨테이너가 생성되지 않은 상태를 의미합니다.
  - ✅ Running: Pod가 노드에 바인딩되고 모든 컨테이너가 생성되고 하나 이상의 컨테이너가 아직 실행 중이거나 시작 또는 재시작 중에 있는 상태를 의미합니다.
  - ✔ Succeeded: Pod의 모든 컨테이너가 성공적으로 종료되었으며 재시작되지 않은 상태를 의미합니다.
  - ⚠ Failed: Pod의 컨테이너 중 하나 이상의 컨테이너가 실패해 종료된 상태를 의미합니다.
  - ? Unknown: 어떤 이유로 Pod와 통신할 수 없는 상태를 의미합니다.
- **QoS 클래스**: Pod의 Quality of Service를 의미합니다. QoS 클래스는 리소스 부족 시 Pod의 우선 순위를 결정합니다. 우선

순위가 높은 Pod가 우선 순위가 낮은 Pod보다 먼저 종료됩니다.

- Guaranteed: 노드에서 사용 가능한 리소스를 보장받고 다른 Pod보다 우선적으로 할당합니다. 가장 마지막에 종료됩니다.
  - Pod의 모든 컨테이너가 CPU 및 메모리 request를 가지고 있는 경우입니다.
  - Pod의 모든 컨테이너가 CPU 및 메모리 limit을 가지고 있으며 해당 값이 request와 동일한 경우입니다.
- Burstable: Pod는 노드 리소스가 부족한 경우에는 일부 리소스를 할당합니다.
  - Guaranteed 클래스의 조건을 충족하지 못하는 경우입니다.
  - Pod의 컨테이너 중 하나라도 CPU 및 메모리에 대한 request 또는 limit이 설정된 경우입니다.
- BestEffort: 리소스 request나 limit이 없기에 다른 Pod보다 우선 순위가 낮습니다. 가장 먼저 종료됩니다.
  - Pod의 모든 컨테이너가 CPU및 메모리에 대한 request와 limit이 없는 경우입니다.

- **시작 시간**: Pod가 시작된 시간 또는 생성된 시간을 의미합니다.

초기화 시작

초기화 시작은 최초 메트릭스 데이터가 수집된 시점을 의미합니다.

- **초기화 시작 시간**: 초기화가 시작된 시점에 대한 데이터로 첫 데이터 수집 시간을 의미합니다.
- **초기화 시작 시 CPU Limit**: 초기화 시작 시 CPU limit을 의미합니다.
- **초기화 시작 시 CPU Request**: 초기화 시작 시 CPU request를 의미합니다.
- **초기화 시작 시 CPU Throttle Count**: 초기화 시작 시 컨테이너에서 CPU 제한을 초과하는 사용 시도 발생 시 증가하는 값을 의미합니다.
- **초기화 시작 시 Memory Limit**: 초기화 시작 시 설정한 메모리 limit을 의미합니다.
- **초기화 시작 시 Memory Request**: 초기화 시작 시 요청한 메모리 request를 의미합니다.
- **초기화 시작 시 Memory Throttle Count**: 초기화 시작 시 컨테이너에서 메모리 제한을 초과하는 사용 시도 발생 시 증가하는 값을 의미합니다.

초기화 중

초기화 중은 초기화 시작 후 5분간 데이터의 CpuByLimit 평균 값보다 낮은 CpuByLimit 값의 데이터가 발생한 이전 시점을 의미합니다.

- **초기화가 소요된 시간(초)**: Pod 초기화에 소요된 시간을 의미합니다.

초기화 시작 시 5분간의 데이터 중 CpuByLimit의 평균 값보다 낮은 CpuByLimit 값이 수집된 시간 - 첫 데이터 수집 시간

- **초기화 중 평균 CpuByLimit**: 초기화 중 사용한 CpuByLimit의 평균 값을 의미합니다.
- **초기화 중 최대 메모리**: 초기화 중 해당 컨테이너에서 사용한 메모리 최대 사용량을 의미합니다.

> ⓘ CpuByLimit(`cpu_per_quota`)은 CPU Limit 기준 CPU 전체 사용률을 의미합니다.

초기화 완료

초기화 완료는 초기화 시작 후 5분간 데이터의 CpuByLimit 평균 값보다 낮은 CpuByLimit 값의 데이터가 발생한 시점을 의미합니다.

- **초기화 완료 시간**: 초기화 완료 시점을 의미합니다.
- **초기화 완료 후 평균 CpuByLimit**: 초기화 완료 후 사용한 CpuByLimit의 평균 값을 의미합니다.
- **초기화 완료 후 최대 메모리**: 초기화 완료 후 해당 컨테이너에서 사용한 메모리 최대 사용량을 의미합니다.
- **초기화 완료 후(초)(5분까지)**: 초기화 완료 후 5분 동안의 밀리초 단위 시간을 의미합니다.

애플리케이션 히트맵

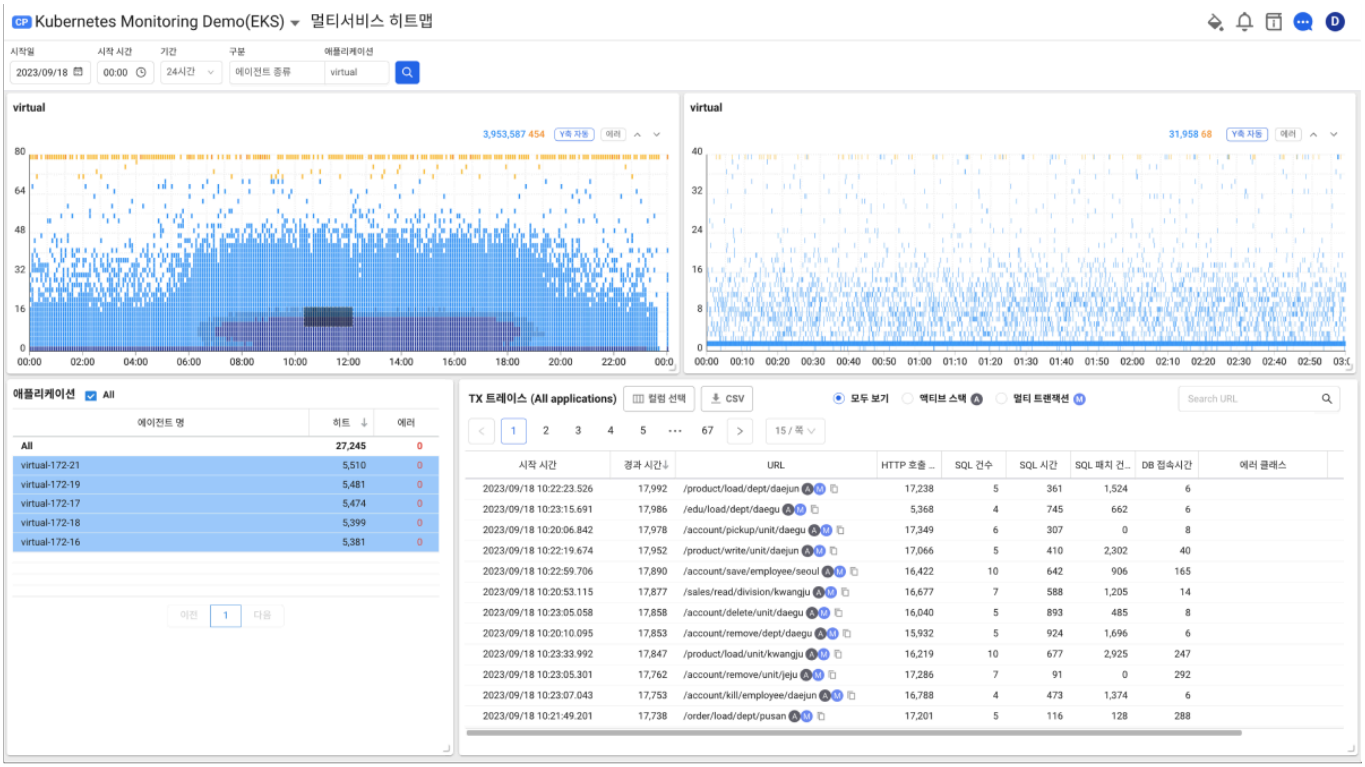
컨테이너 내부 애플리케이션의 히트맵 정보를 확인할 수 있습니다. 히트맵은 트랜잭션 수행 건수 맵으로 응답 시간 분포도를 표시합니다. 검색 조건은 시간 선택, 구분, 애플리케이션으로 구분 조건은 다음과 같습니다.

분류	설명
에이전트	컨테이너 내부에 설치된 애플리케이션 에이전트 이름
에이전트 종류	에이전트가 그룹화된 에이전트 종류 (OKIND)
에이전트 서버	에이전트가 실행 중인 쿠버네티스 노드 이름

히트맵 특정 구간을 드래그하면 트레이스 정보를 확인할 수 있습니다. 트레이스 데이터는 URL 기반의 검색이 가능하며 액티브 스택이 존재하는 트레이스, 멀티 트랜잭션으로 수행하는 트레이스 정보를 필터링하여 확인할 수도 있습니다.

> ⓘ 히트맵에 대한 자세한 내용은 다음 문서를 참조하세요.
>
> - Java
> - Node.js
> - Python

멀티서비스 히트맵



여러 서비스를 대상으로 히트맵 차트를 확인하고 서비스별 상태를 비교 분석할 수 있습니다.

검색 조건은 다음과 같습니다.

항목	설명
시작일	검색날짜
시작시간	3일, 7일, 한 달 검색 시에는 사용할 수 없습니다
기간	• 5분/10분/30분 • 1시간/3시간/24시간 • 3일/7일/한 달

항목	설명
구분	에이전트 종류
애플리케이션	애플리케이션

ⓘ 히트맵에 대한 자세한 내용은 다음 문서를 참조하세요.

- Java
- Node.js
- Python

MSA 분석

MSA 분석이란?

MSA 분석 기능은 와탭의 **멀티 트랜잭션 추적** 기능을 통해서 수집된 트랜잭션들 사이의 호출 비중이 어떠한지를 분석합니다. 이를 기반으로 마이크로 서비스간 의존성을 인스턴스 중심이 아닌 URL 레벨에서 확인할 수 있습니다.

Microservice Architecture Pattern

예를 들어 A,B,C,D,E 라는 5개 마이크로 서비스가 있는데, 사용자가 A를 호출하면 로직에 따라서 다음과 같은 3가지의 호출 패턴이 발생한다고 가정하겠습니다.

1. A→B,C 호출
2. A→B,C 호출 + 다시 B→D 호출
3. A→B,C 호출 + 다시 B→D,E 호출

이 세가지 패턴이 한번씩 수행되었다면 MSA 분석에서는 기준 URL에 따라 연관도 분석 데이터를 확인할 수 있습니다.

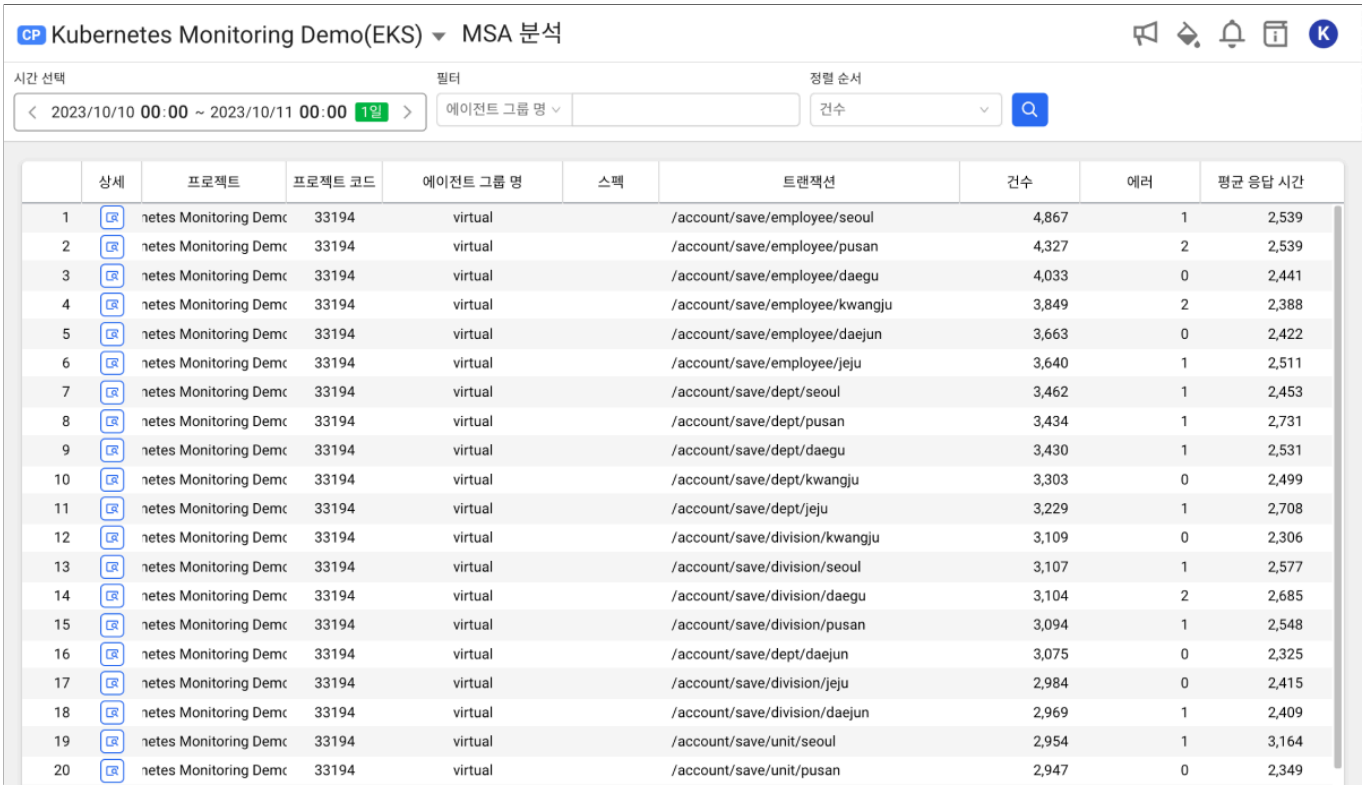
Microservice Architecture Caller & Callee

> ⓘ **멀티 트랜잭션 추적**과 관련한 자세한 내용은 다음 문서를 참조하세요.

MSA 트랜잭션 통계

홈 화면 > 프로젝트 선택 > **분석** > **MSA 분석**

MSA 분석 메뉴는 마이크로 서비스 환경에서 각 서비스의 성능과 안정성을 효과적으로 분석할 수 있는 도구입니다. 트랜잭션의 상세 호출 관계를 분석하고, 이를 기반으로 시스템 성능을 최적화하며 잠재적인 문제를 사전에 식별할 수 있습니다.



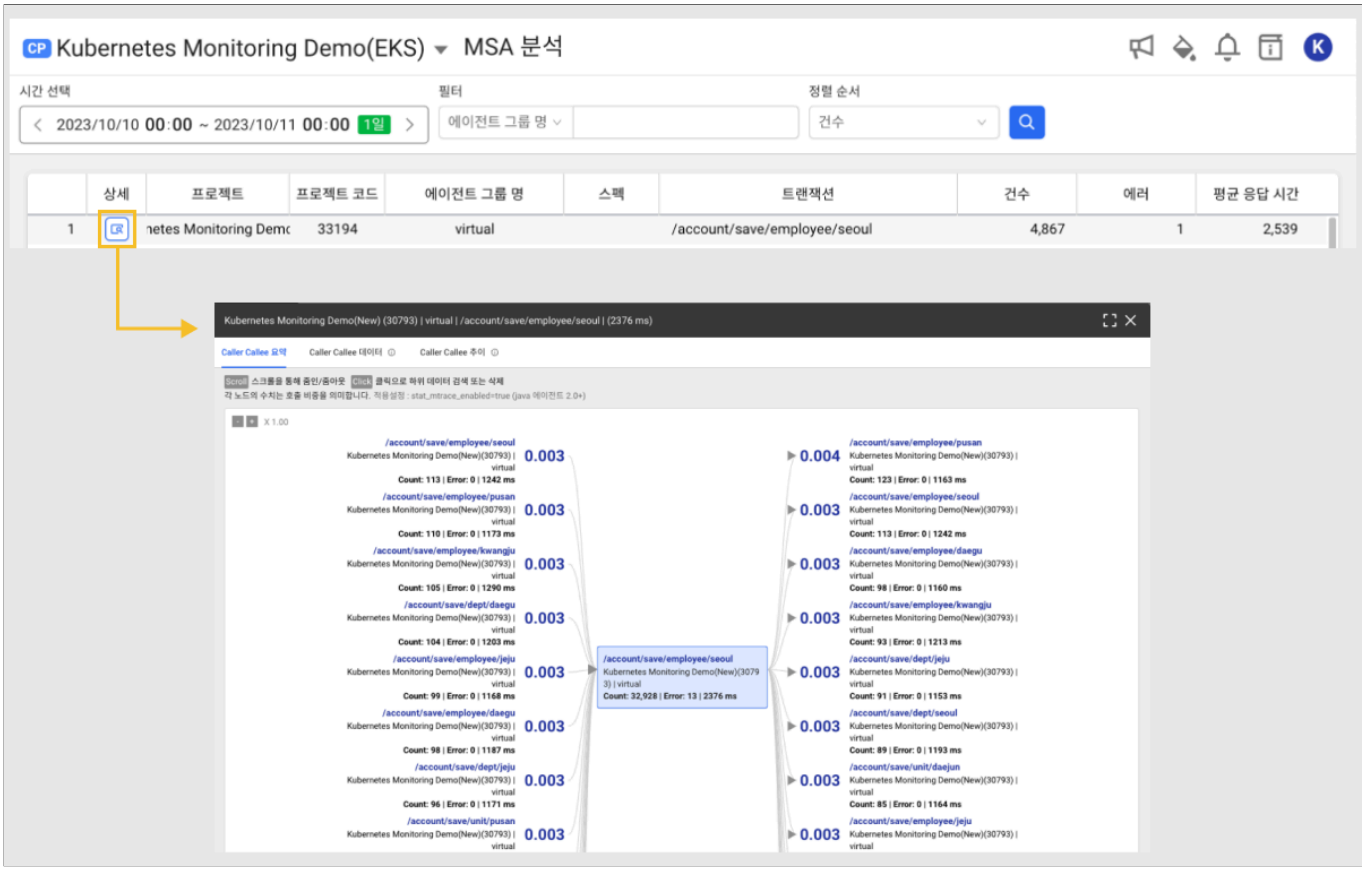
	상세	프로젝트	프로젝트 코드	에이전트 그룹 명	스펙	트랜잭션	건수	에러	평균 응답 시간
1		netes Monitoring Demc	33194	virtual		/account/save/employee/seoul	4,867	1	2,539
2		netes Monitoring Demc	33194	virtual		/account/save/employee/pusan	4,327	2	2,539
3		netes Monitoring Demc	33194	virtual		/account/save/employee/daegu	4,033	0	2,441
4		netes Monitoring Demc	33194	virtual		/account/save/employee/kwangju	3,849	2	2,388
5		netes Monitoring Demc	33194	virtual		/account/save/employee/daejun	3,663	0	2,422
6		netes Monitoring Demc	33194	virtual		/account/save/employee/jeju	3,640	1	2,511
7		netes Monitoring Demc	33194	virtual		/account/save/dept/seoul	3,462	1	2,453
8		netes Monitoring Demc	33194	virtual		/account/save/dept/pusan	3,434	1	2,731
9		netes Monitoring Demc	33194	virtual		/account/save/dept/daegu	3,430	1	2,531
10		netes Monitoring Demc	33194	virtual		/account/save/dept/kwangju	3,303	0	2,499
11		netes Monitoring Demc	33194	virtual		/account/save/dept/jeju	3,229	1	2,708
12		netes Monitoring Demc	33194	virtual		/account/save/division/kwangju	3,109	0	2,306
13		netes Monitoring Demc	33194	virtual		/account/save/division/seoul	3,107	1	2,577
14		netes Monitoring Demc	33194	virtual		/account/save/division/daegu	3,104	2	2,685
15		netes Monitoring Demc	33194	virtual		/account/save/division/pusan	3,094	1	2,548
16		netes Monitoring Demc	33194	virtual		/account/save/dept/daejun	3,075	0	2,325
17		netes Monitoring Demc	33194	virtual		/account/save/division/jeju	2,984	0	2,415
18		netes Monitoring Demc	33194	virtual		/account/save/division/daejun	2,969	1	2,409
19		netes Monitoring Demc	33194	virtual		/account/save/unit/seoul	2,954	1	3,164
20		netes Monitoring Demc	33194	virtual		/account/save/unit/pusan	2,947	0	2,349

- **시간 선택**: 원하는 조회 기간을 선택하세요. 기본 1일 기준으로 Caller Callee를 포함한 트랜잭션 목록을 조회할 수 있습니다.
- **필터**: **에이전트 그룹 명**, **스펙**, **트랜잭션** 기준으로 필터링하여 검색할 수 있습니다.
- **정렬 순서**: **건수**, **에러**, **평균 시간**(평균 응답 시간) 기준으로 목록을 정렬할 수 있습니다.

> - **에이전트 그룹 명**은 에이전트 설정 옵션의 `whatap.okind` 로 분류된 그룹 이름입니다.
> - 조건을 설정한 다음 트랜잭션 목록을 갱신하려면 버튼을 선택하세요.

상세 분석하기

트랜잭션 목록에서 조회를 원하는 항목의 버튼을 선택하면 상세 보기 창이 나타납니다. 상세 보기 창에서 해당 트랜잭션에 대한 호출자(Caller), 피호출자(Callee) 정보를 조회할 수 있습니다.



각 탭에서는 다음의 분석 기능을 제공합니다.

- **Caller Callee 요약**: 트랜잭션의 호출자와 피호출자 간의 관계를 시각화된 다이어그램으로 제공합니다.
- **Caller Callee 데이터**: 트랜잭션의 호출 횟수, 오류 수, 응답 시간을 상세히 제공합니다.
- **Caller Callee 추이**: 시간에 따른 트랜잭션의 호출 횟수 및 비율의 성능 변화를 시계열 차트로 제공합니다.

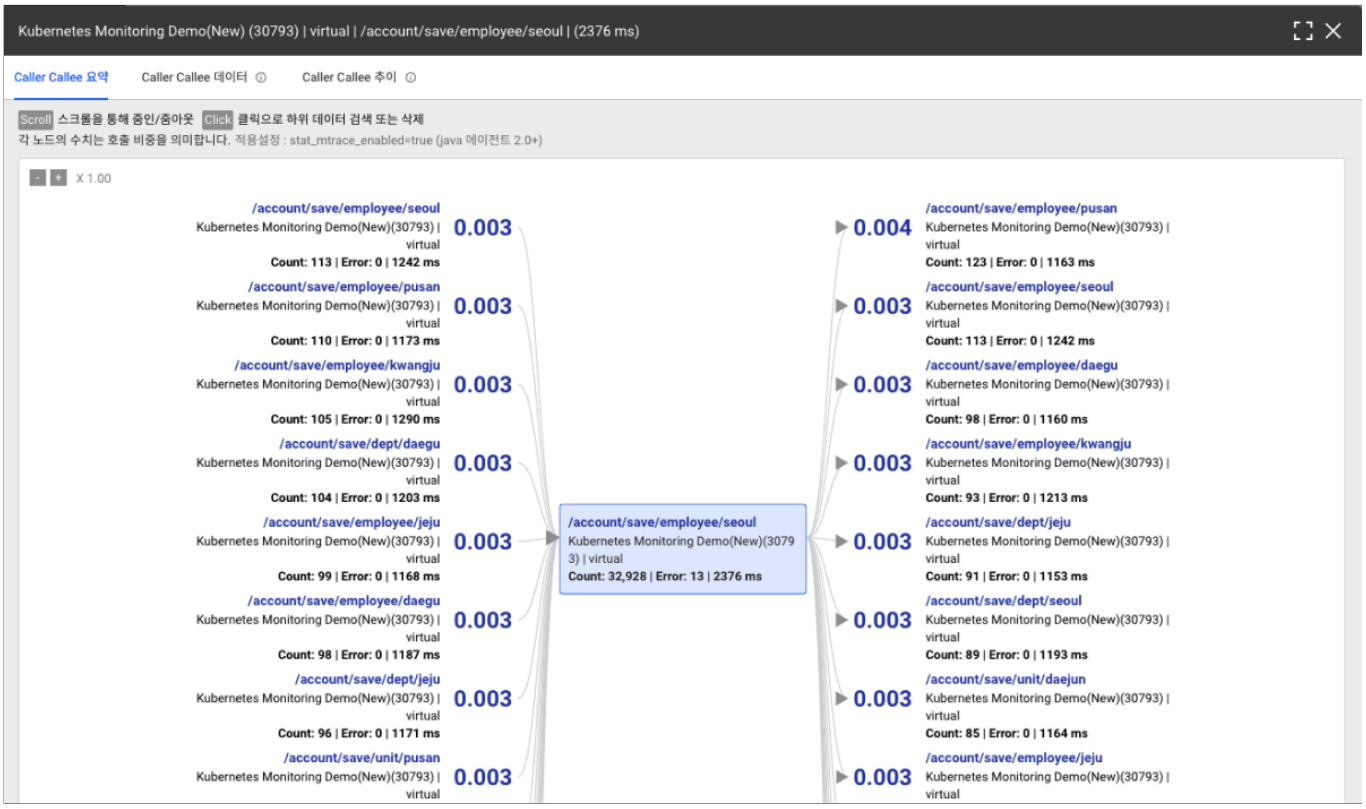
> ⓘ
> - 상세 보기 창에서 오른쪽 위에 ⛶ 버튼을 선택하면 전체 화면으로 볼 수 있습니다.
> - 상세 보기 창을 닫으려면 오른쪽 아래에 **확인** 버튼을 선택하거나 ESC 키를 누르세요.

Caller Callee 요약

Caller Callee 요약 탭에서는 트랜잭션 목록에서 선택한 트랜잭션의 Caller, Callee 사이의 의존성 정보를 시각화된 다이어그램으로 표시합니다. 서비스 간의 호출 관계를 직관적으로 이해할 수 있습니다. 호출 경로와 종속성을 시각적으로 파악할

수 있어 문제 원인 분석에 유리합니다.

호출 관계 분석하기



Caller와 Callee 노드에는 최대 0.004, 0.003 등 최대 1인 숫자 정보를 표시하는데, 이것은 전체 Caller 가운데 차지하는 비중을 의미합니다. 마찬가지로 각 Callee의 비중은 오른쪽에 함께 표시합니다.

위 화면을 통해 /account/save/employee/seoul 이라는 트랜잭션을 기준으로 Caller와 Callee들 사이의 의존성을 분석할 수 있습니다. 트랜잭션이 어떤 서비스에서 호출되고 어떤 서비스로 호출되는지 시각적으로 확인할 수 있습니다.

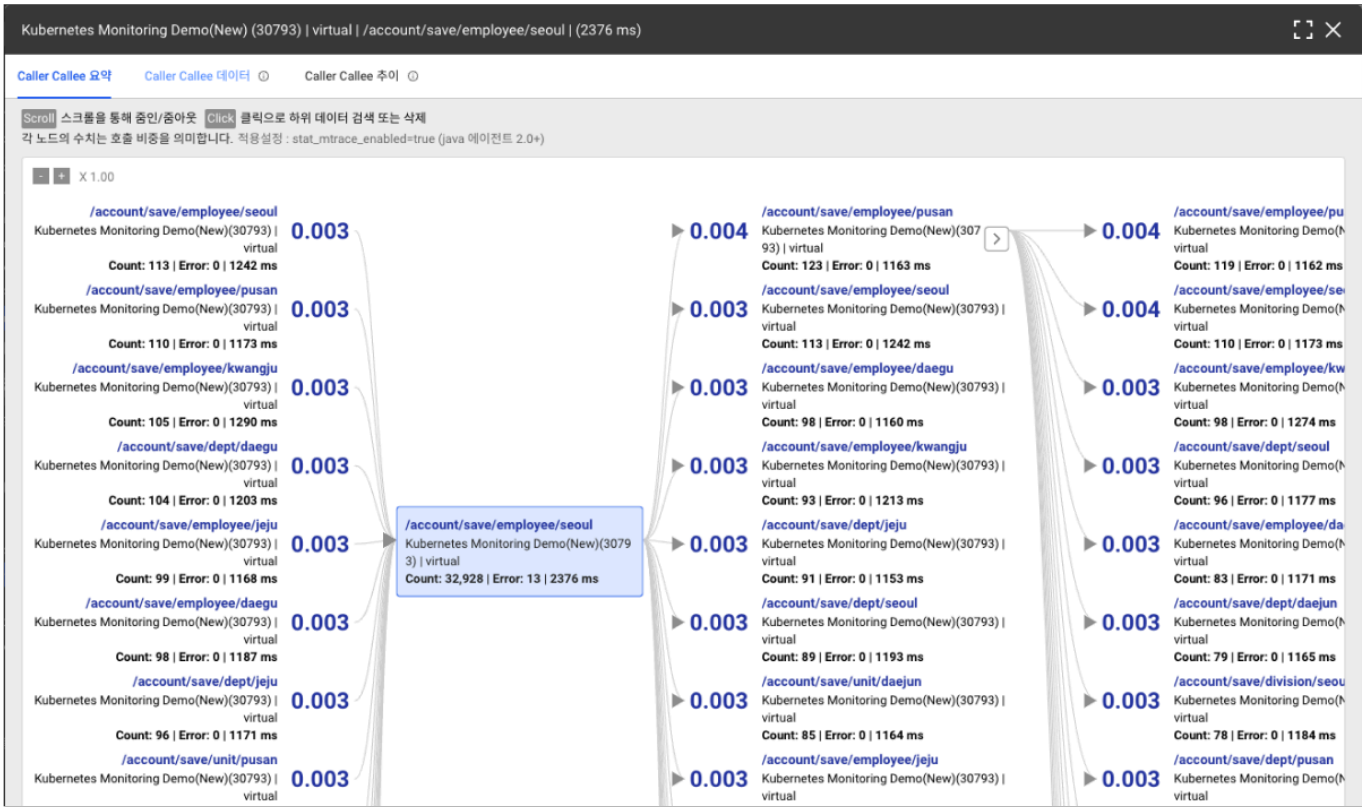
성능 지표 확인하기

각 노드는 트랜잭션 URL, 프로젝트 이름, 에이전트 그룹 이름 그리고 다음 성능 정보로 구성되어 있습니다.

- **Count**: 해당 트랜잭션을 호출한 횟수
- **Error**: 호출 중 발생한 에러 수
- 평균 응답 시간: 해당 트랜잭션의 평균 응답 시간, 밀리초(ms) 단위

트랜잭션 흐름 파악하기

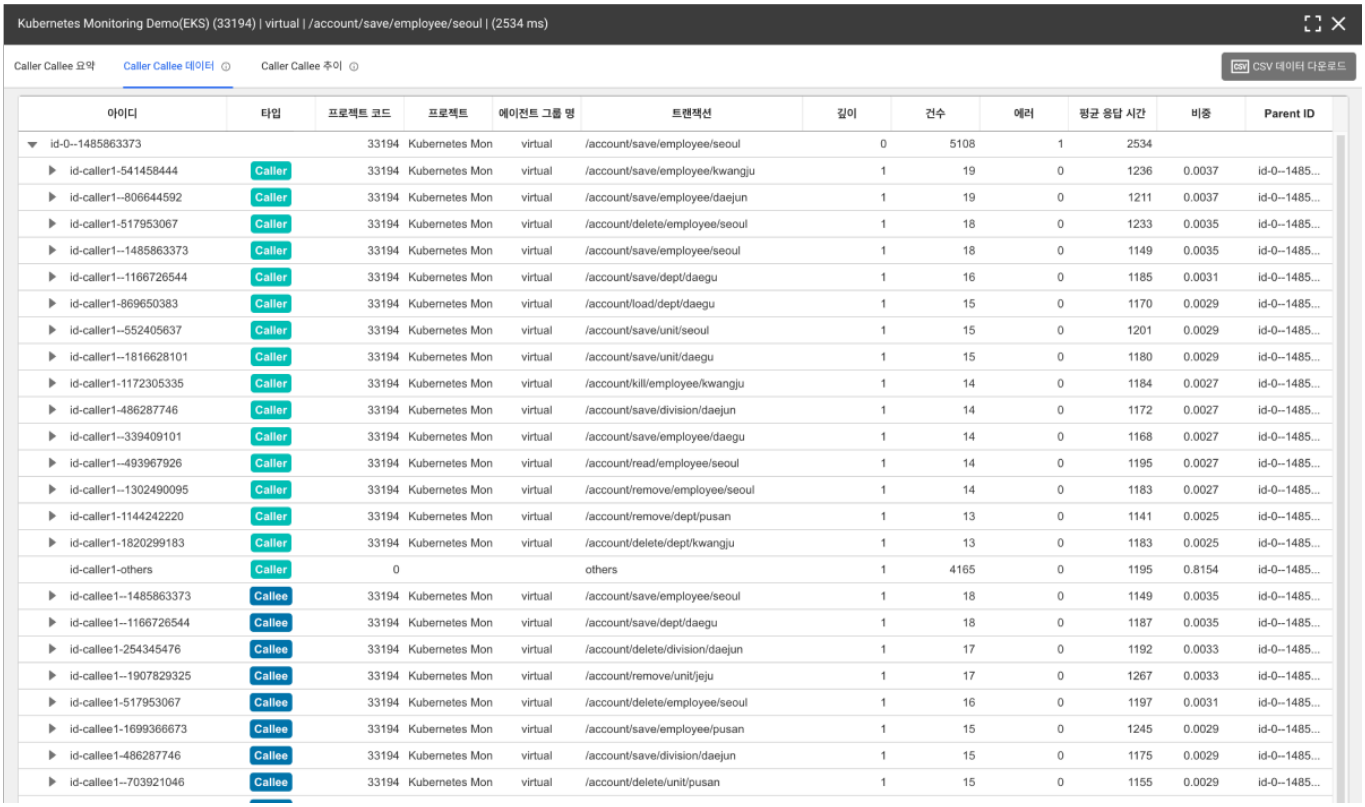
Caller나 Callee 노드에 마우스를 오버하면 > 버튼이 표시됩니다. 이 버튼을 선택하면 하위 노드로 상세 전개할 수 있습니다. 트랜잭션의 전체 흐름을 파악하여 문제 발생 지점을 찾는데 도움을 받을 수 있습니다.



- 하위 노드를 닫으려면 다시 > 버튼을 선택하세요.
- 마우스의 스크롤 기능으로 화면을 확대하거나 축소할 수 있습니다. 또는 화면 왼쪽 위에 + / - 버튼을 클릭하세요.

Caller & Callee 데이터

Caller Callee 데이터 탭에서는 기준 URL(예: /account/save/employee/seoul)에 대한 호출자(Caller)와 피호출자(Callee)의 상관 관계를 표 형식으로 확인할 수 있습니다. 각 트랜잭션의 세부 정보(호출 횟수, 오류 수, 평균 응답 시간 등)를 구체적으로 확인할 수 있어 데이터 기반의 정밀한 분석이 가능합니다.

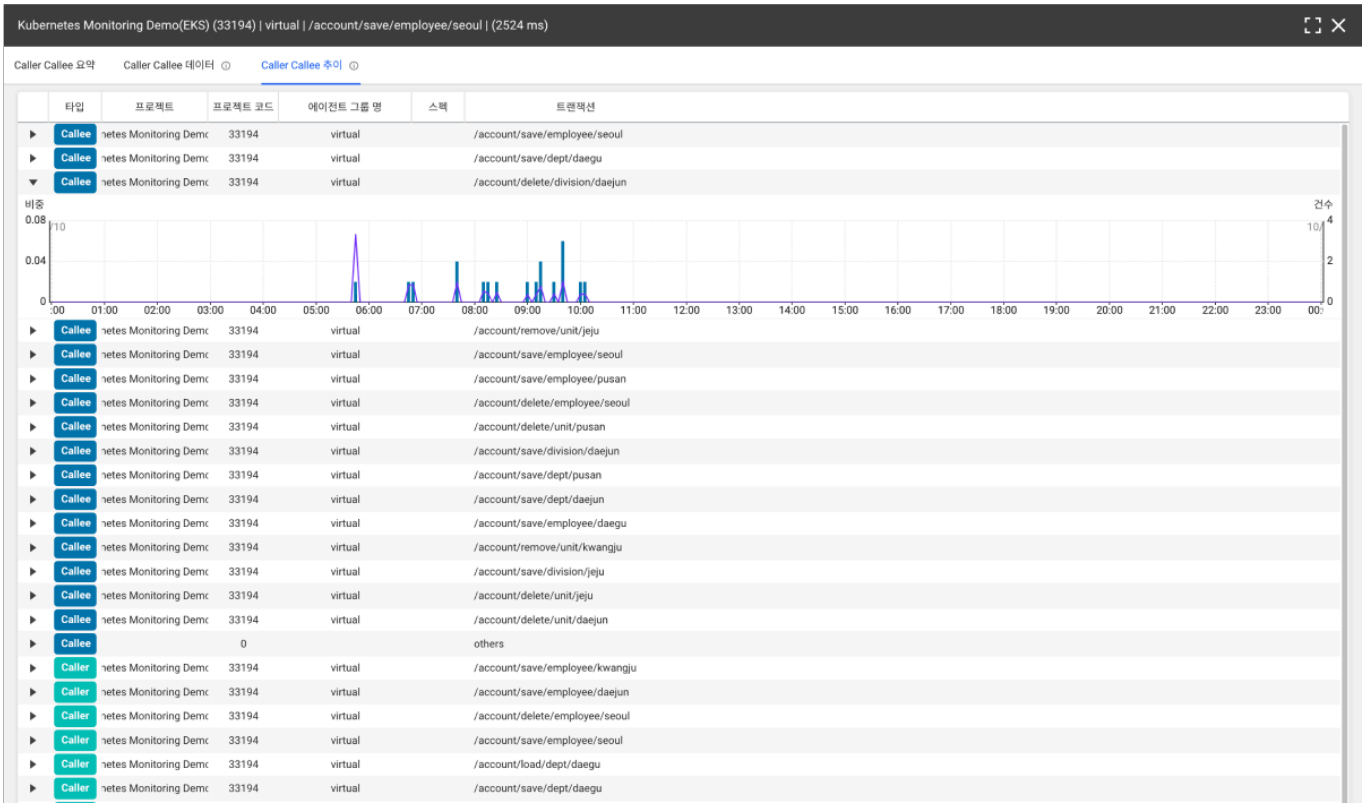


아이디	타입	프로젝트 코드	프로젝트	에이전트 그룹 명	트랜잭션	깊이	건수	에러	평균 응답 시간	비중	Parent ID
id-0--1485863373		33194	Kubernetes Mon	virtual	/account/save/employee/seoul	0	5108	1	2534		
id-caller1-541458444	Caller	33194	Kubernetes Mon	virtual	/account/save/employee/kwangju	1	19	0	1236	0.0037	id-0--1485...
id-caller1--806644592	Caller	33194	Kubernetes Mon	virtual	/account/save/employee/daejun	1	19	0	1211	0.0037	id-0--1485...
id-caller1-517953067	Caller	33194	Kubernetes Mon	virtual	/account/delete/employee/seoul	1	18	0	1233	0.0035	id-0--1485...
id-caller1--1485863373	Caller	33194	Kubernetes Mon	virtual	/account/save/employee/seoul	1	18	0	1149	0.0035	id-0--1485...
id-caller1--1166726544	Caller	33194	Kubernetes Mon	virtual	/account/save/dept/daegu	1	16	0	1185	0.0031	id-0--1485...
id-caller1-869650383	Caller	33194	Kubernetes Mon	virtual	/account/load/dept/daegu	1	15	0	1170	0.0029	id-0--1485...
id-caller1--552405637	Caller	33194	Kubernetes Mon	virtual	/account/save/unit/seoul	1	15	0	1201	0.0029	id-0--1485...
id-caller1--1816628101	Caller	33194	Kubernetes Mon	virtual	/account/save/unit/daegu	1	15	0	1180	0.0029	id-0--1485...
id-caller1-1172305335	Caller	33194	Kubernetes Mon	virtual	/account/kill/employee/kwangju	1	14	0	1184	0.0027	id-0--1485...
id-caller1-486287746	Caller	33194	Kubernetes Mon	virtual	/account/save/division/daejun	1	14	0	1172	0.0027	id-0--1485...
id-caller1--339409101	Caller	33194	Kubernetes Mon	virtual	/account/save/employee/daegu	1	14	0	1168	0.0027	id-0--1485...
id-caller1--493967926	Caller	33194	Kubernetes Mon	virtual	/account/read/employee/seoul	1	14	0	1195	0.0027	id-0--1485...
id-caller1--1302490095	Caller	33194	Kubernetes Mon	virtual	/account/remove/employee/seoul	1	14	0	1183	0.0027	id-0--1485...
id-caller1-1144242220	Caller	33194	Kubernetes Mon	virtual	/account/remove/dept/pusan	1	13	0	1141	0.0025	id-0--1485...
id-caller1-1820299183	Caller	33194	Kubernetes Mon	virtual	/account/delete/dept/kwangju	1	13	0	1183	0.0025	id-0--1485...
id-caller1-others	Caller	0			others	1	4165	0	1195	0.8154	id-0--1485...
id-callee1--1485863373	Callee	33194	Kubernetes Mon	virtual	/account/save/employee/seoul	1	18	0	1149	0.0035	id-0--1485...
id-callee1--1166726544	Callee	33194	Kubernetes Mon	virtual	/account/save/dept/daegu	1	18	0	1187	0.0035	id-0--1485...
id-callee1-254345476	Callee	33194	Kubernetes Mon	virtual	/account/delete/division/daejun	1	17	0	1192	0.0033	id-0--1485...
id-callee1--1907829325	Callee	33194	Kubernetes Mon	virtual	/account/remove/unit/jeju	1	17	0	1267	0.0033	id-0--1485...
id-callee1-517953067	Callee	33194	Kubernetes Mon	virtual	/account/delete/employee/seoul	1	16	0	1197	0.0031	id-0--1485...
id-callee1-1699366673	Callee	33194	Kubernetes Mon	virtual	/account/save/employee/pusan	1	15	0	1245	0.0029	id-0--1485...
id-callee1-486287746	Callee	33194	Kubernetes Mon	virtual	/account/save/division/daejun	1	15	0	1175	0.0029	id-0--1485...
id-callee1--703921046	Callee	33194	Kubernetes Mon	virtual	/account/delete/unit/pusan	1	15	0	1155	0.0029	id-0--1485...

- 각 트랜잭션의 호출 횟수와 비율, 오류 수, 평균 응답 시간 등을 확인하여 성능 문제를 분석하는데 활용할 수 있습니다.
- **타입** 컬럼을 통해서 Caller와 Callee를 쉽게 구분할 수 있으며 호출 관계를 파악할 수 있습니다.
- 각 트랜잭션의 ▶ 버튼을 선택하면 해당 트랜잭션과 관계된 트랜잭션을 추가로 확인할 수 있습니다.
- 데이터 목록을 CSV 형식으로 저장하려면 오른쪽 위에 **CSV 데이터 다운로드** 버튼을 선택하세요.

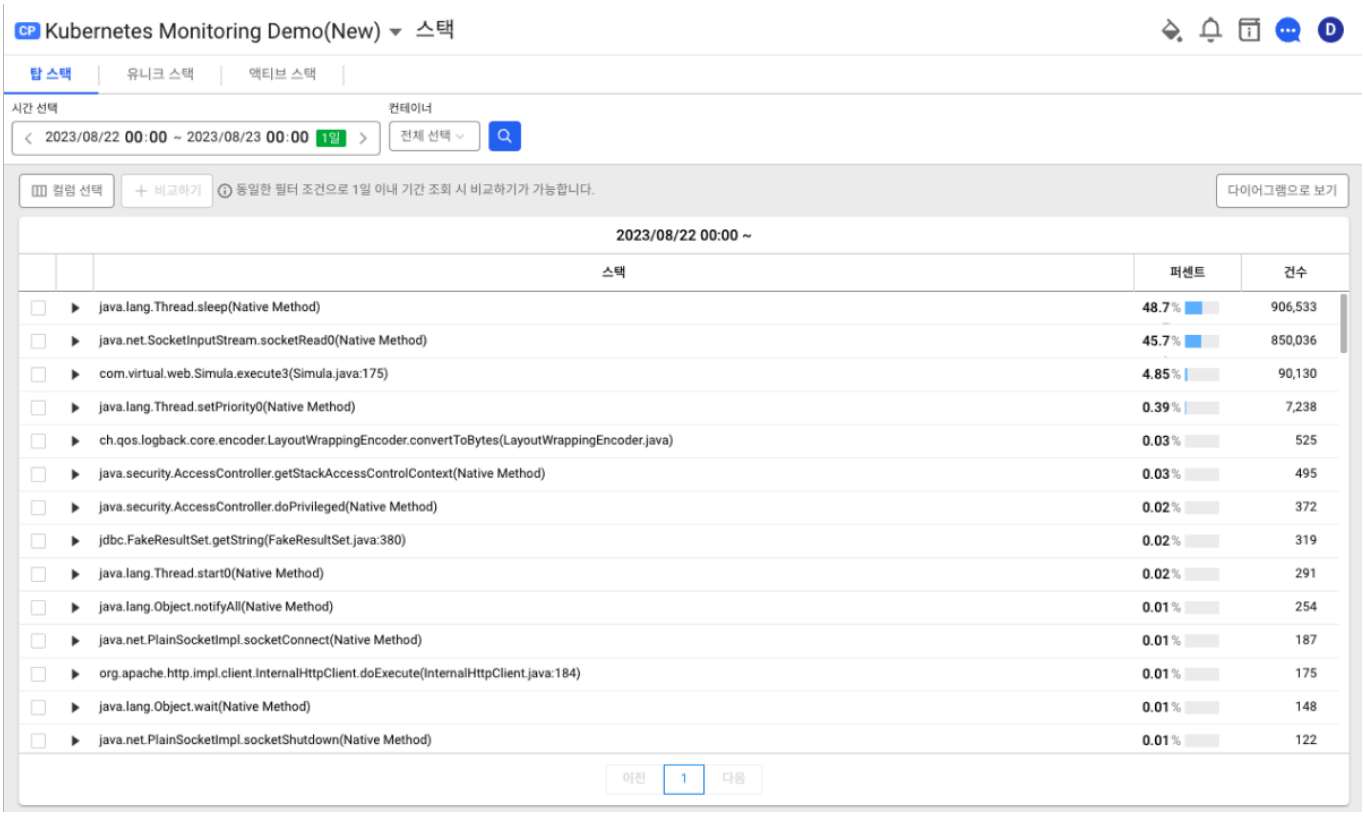
Caller Callee 추이

Caller Callee 추이 탭에서는 기준 URL(/account/save/employee/seoul)의 호출 관계와 성능 데이터를 시간의 흐름에 따라 시각적으로 분석할 수 있습니다. 이를 통해 트랜잭션의 성능 변동 패턴을 파악할 수 있습니다.



- 특정 트랜잭션의 호출 횟수와 비율을 시간대별 그래프로 시각화하여 성능 변화를 한눈에 파악할 수 있습니다.
- **타입** 컬럼을 통해서 Caller와 Callee를 쉽게 구분할 수 있으며 호출 관계를 파악할 수 있습니다.
- 시계열 차트에 마우스를 오버하면 툴팁을 통해서 수치 정보를 확인할 수 있습니다.
- 차트의 특정 시간대를 클릭하면 **분석** > **트랜잭션 검색** 메뉴로 이동합니다. 선택한 시간대의 트랜잭션 목록을 조회할 수 있습니다.

애플리케이션 스택



와탭은 10초(기본값) 간격으로 수집한 스레드 스택을 활용하여 메소드 레벨의 성능 지연 구간을 분석합니다. 이 기능을 통해 각 컨테이너의 **탑 스택**, **유니크 스택**, **액티브 스택**을 확인할 수 있습니다.

> ⓘ **탑 스택**, **유니크 스택**, **액티브 스택**에 대한 자세한 내용은 다음 문서를 참조하세요.
>
> - Java
> - Python

메트릭스

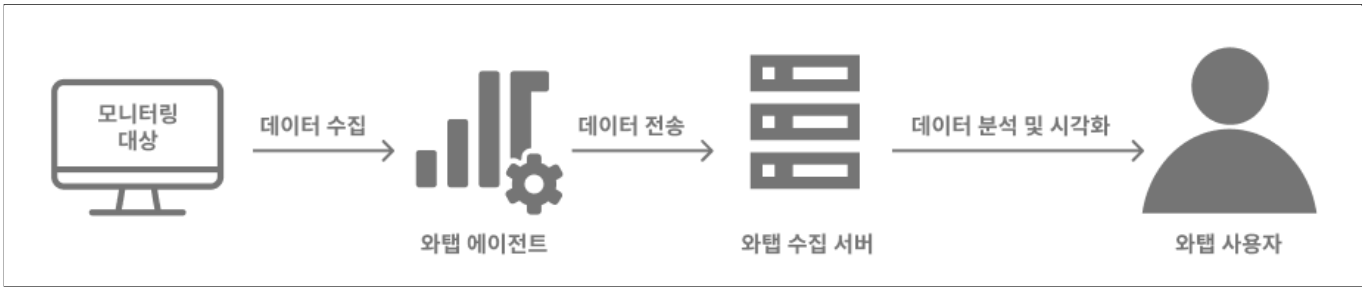
메트릭이란?

메트릭은 모니터링 대상의 상태를 수치로 표현한 시계열 데이터입니다. 와탭 에이전트가 모니터링 대상에서 수집한 성능 데이터를 가공한 결과입니다.

메트릭을 통해 서버별 메모리 사용률, DB 연결 시간 등 주요 성능 지표를 한눈에 파악할 수 있습니다. 문제를 발견한 후에는 로그와 트레이스를 통해 상세 분석할 수 있습니다.

자원 사용량 통계를 기반으로 필요 자원량을 산정할 수 있어, 비용 효율화에도 활용할 수 있습니다.

와탭의 메트릭 수집 방식



메트릭을 수집하려면 모니터링 대상에 와탭 에이전트를 설치해야 합니다. 에이전트가 전송한 메트릭은 와탭 수집 서버에 저장됩니다.

예를 들어 **Kubernetes** 모니터링을 시작하려면 먼저 에이전트를 설치하세요. 수집 가능한 메트릭은 Kubernetes 메트릭에서 확인할 수 있습니다.

와탭의 메트릭 구성 요소

와탭의 **메트릭**은 다음의 요소로 구성되어 있습니다.

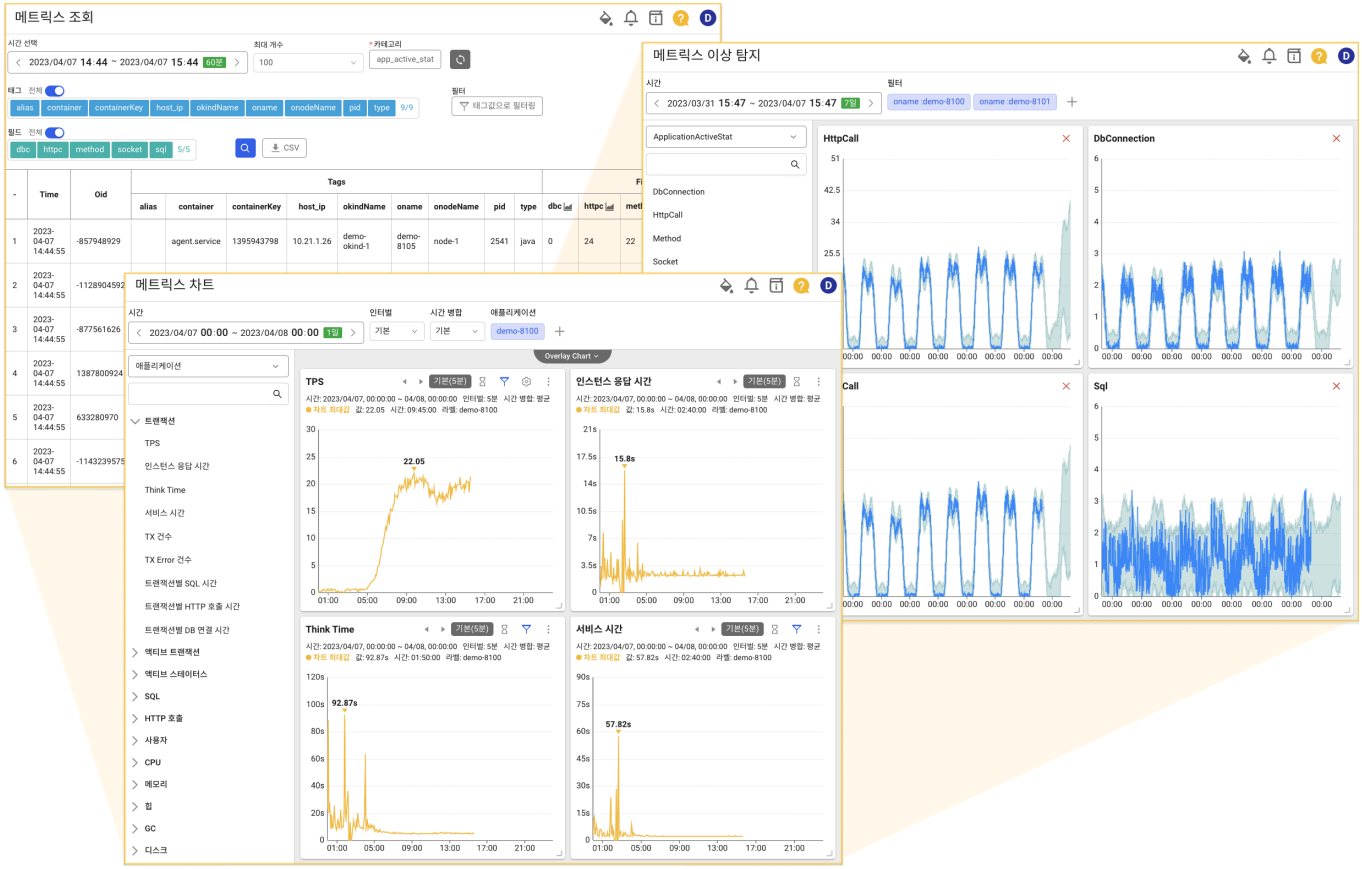
- **Category**: 관련 메트릭을 하나의 그룹으로 묶는 이름입니다.

- **Tags**: 수집 대상을 식별하는 라벨입니다. IP, Oname, Host 등 변경이 드문 고유 정보를 저장합니다. 하나의 메트릭에 여러 태그가 포함될 수 있습니다.
- **Fields**: 실제 측정값을 저장합니다. 하나의 메트릭에 여러 필드가 포함될 수 있습니다.
- **Time**: 메트릭을 수집한 시간입니다.
- **Oid**: 메트릭을 수집한 에이전트의 고유 번호입니다.
- **Oname**: 메트릭을 수집한 에이전트의 이름입니다.

메트릭 데이터 조회 및 시각화

와탭은 수집한 메트릭을 다양한 방식으로 조회할 수 있는 메뉴를 제공합니다.

- **메트릭스 조회**: 메트릭의 원본 데이터를 조회합니다.
- **메트릭스 차트**: 시각화한 차트를 통해 메트릭 데이터를 조회합니다.
- **메트릭스 이상 탐지**: AI가 학습한 메트릭 패턴과 비교해 예상 패턴을 벗어난 이상을 탐지합니다.


메트릭스 조회
메트릭스 이상 탐지
메트릭스 차트
TPS
인스턴스 응답 시간
Think Time
서비스 시간
HttpCall
DbConnection
Sql

쿠버네티스 메트릭

쿠버네티스 모니터링이 수집하는 메트릭을 카테고리별로 정리했습니다. 각 카테고리의 Tags(차원)와 Fields(측정값)를 확인할 수 있습니다.

- 일부 메트릭은 서버 모니터링, 애플리케이션 모니터링과 동일합니다.
- 메트릭을 직접 검색하거나 차트로 시각화하려면 메트릭 검색, 메트릭스 차트 페이지를 참조하세요.

컨테이너(`container`) 메트릭

`container` 카테고리는 컨테이너의 pod에 설정된 모든 사용자 정의 라벨들을 태그로 수집합니다.

- 대상: 클러스터 프로젝트, 네임스페이스 프로젝트
- 수집 간격: 5초
- 통계 데이터: 5분

Tags

Tag	Type	Unit	Description
agentOid	-	-	노드 에이전트 고유 ID(고유값)
agentPcode	-	-	프로젝트 코드(고유값)
command	-	-	실행 명령어
containerId	-	-	컨테이너 ID(고유값)
containerKey	-	-	컨테이너 Key
created	-	-	컨테이너가 생성된 TimeStamp
image	-	-	컨테이너 이미지명

Tag	Type	Unit	Description
imageHash	-	-	이미지 해시 값
imageId	-	-	이미지 ID
k8s-app	-	-	Pod의 레이블 k8s-app에 대한 값
microOid	-	-	컨테이너에 설치된 와탭 APM 에이전트의 고유 ID
name	-	-	컨테이너 이름
namespace	-	-	컨테이너가 소속된 네임스페이스
namespaceHash	-	-	컨테이너가 소속된 네임스페이스 해시 값
okind	-	-	컨테이너에 설치된 와탭 APM 에이전트에 지정한 OKIND 고유 ID
okindName	-	-	컨테이너에 설치된 와탭 APM 에이전트에 지정한 OKIND 이름
oname	-	-	컨테이너에 설치된 와탭 APM 에이전트 이름
onode	-	-	컨테이너가 작동 중인 노드 에이전트의 고유 ID
onodeName	-	-	컨테이너가 작동 중인 노드 이름
podHash	-	-	컨테이너의 Pod 해시 값
podName	-	-	컨테이너의 Pod 이름
replicaSetHash	-	-	컨테이너의 레플리카셋 해시 값
replicaSetName	-	-	컨테이너의 레플리카셋 이름
whatap_project	-	-	컨테이너가 속한 와탭 프로젝트 이름

Fields

Field	Type	Unit	Shortname/Name/Description
blkio_rbps	-	byte	IoReadBytes
			Container Block I/O Read Byte
			컨테이너 전체 블럭 디바이스의 초당 읽은 바이트 합
blkio_riops	-	count	IoReadIops
			Container Block I/O Read IOPS
			컨테이너 전체 블럭 디바이스의 초당 읽은 건수 합
blkio_wbps	-	byte	IoWriteBytes
			Container Block I/O Write Byte
			컨테이너 전체 블럭 디바이스의 초당 쓴 바이트 합
blkio_wiops	-	count	IoWriteIops
			Container Block I/O Write IOPS
			컨테이너 전체 블럭 디바이스의 초당 쓴 건수 합
cpu_per_quota	-	percent	CpuByLimit
			Container CPU Usage by Limit(%)
			CPU Limit 기준 CPU 전체 사용률
cpu_quota	-	millicores	CpuLimit
			Container CPU Limit (core)

Field	Type	Unit	Shortname/Name/Description
			컨테이너 CPU Limit 할당량 - Limit 미설정인 경우 컨테이너가 작동 중인 해당 노드의 CPU 전체 코어가 밀리코어 단위로 표시됨
cpu_quota_percent	-	percent	CpuLimitByNode
			Container CPU Limit by Node(%)
			노드 CPU 대비 컨테이너 CPU Limit 할당량 - Limit 미설정인 경우 컨테이너가 작동 중인 해당 노드의 CPU 전체 코어가 퍼센트로 표시됨
cpu_sys	-	percent	CpuSysByNode
			Container CPU Sys Usage by Node(%)
			노드 CPU 대비 컨테이너 CPU System 사용률
cpu_throttledperiods	-	count	CpuThrottledCnt
			Container CPU Throttling Count
			컨테이너 CPU Throttled 건수
cpu_throttledtime	-	nanosecond(ns)	CpuThrottledTime
			Container CPU Throttling Time
			컨테이너 CPU Throttled 시간
cpu_total	-	percent	CpuByNode
			Container CPU Usage by Node(%)
			노드 CPU 대비 컨테이너 CPU 사용률

Field	Type	Unit	Shortname/Name/Description
cpu_total_milli	-	millicores	CpuTotUsage
			Container CPU Usage (millicore)
			컨테이너 CPU 사용량
cpu_user	-	percent	CpuUserByNode
			Container CPU User Usage by Node(%)
			노드 CPU 대비 컨테이너 CPU User 사용률
cpu_request	-	millicores	CpuRequest
			Container CPU Request (core)
			컨테이너 CPU 요청
cpu_per_request	-	percent	CpuByRequest
			Container CPU Usage by Request(%)
			컨테이너 CPU 요청 대비 사용률 = cpu_total_milli / cpu_request * 100
mem_failcnt	-	count	MemFailCnt
			Container Memory Failure Count
			컨테이너 메모리 Limit 도달 건수
mem_limit	-	byte	MemLimit
			Container Memory Limit (byte)
			컨테이너 메모리 Limit 크기

Field	Type	Unit	Shortname/Name/Description
mem_maxusage	-	byte	MemMaxUsage
			Container Memory Max Usage (byte)
			컨테이너 메모리 최대 사용량 기록값 - 자세한 내용은 하단 안내를 참고하세요.
mem_percent	-	percent	MemWsByLimit
			Container Memory Working Set by Limit(%)
			컨테이너 메모리 Limit 기준 working set 사용량 = mem_working_set / mem_limit * 100
mem_totalcache	-	byte	MemTotCache
			Container Memory Total Cache (byte)
			컨테이너 전체 캐시 크기
mem_totalpgfault	-	count	MemTotPageFaultCnt
			Container Memory Total Page Fault Count
			컨테이너 Page Fault 횟수
mem_totalrss	-	byte	MemTotRss
			Container Memory Total RSS (byte)
			컨테이너 RSS 메모리 전체 크기
mem_totalrss_percent	-	percent	MemTotRssByLimit
			Container Memory Total RSS by Limit (%)

Field	Type	Unit	Shortname/Name/Description
			컨테이너 RSS 메모리 전체 사용률
mem_totalunevictable	-	byte	MemTotUnevictable
			Container Memory Total Unevictable (byte)
			컨테이너 Unevictable Memory 전체 크기
mem_usage	-	byte	MemUsage
			Container Memory Usage (byte)
			컨테이너 메모리 사용량
mem_working_set	-	byte	MemWs
			Container Memory Working Set (byte)
			컨테이너 메모리 working set = mem_usage - inactive file
mem_working_set_percent	-	percent	MemWsByLimit
			Container Memory Working Set by Limit (%)
			컨테이너 메모리 Limit 기준 working set 사용량 = mem_working_set / mem_limit * 100
mem_request	-	byte	MemRequest
			Container Memory Request (byte)
			컨테이너 메모리 Request 크기
mem_per_request	-	percent	MemWsByRequest

Field	Type	Unit	Shortname/Name/Description
			Container Memory Working Set by Request (%)
			컨테이너 메모리 Request 기준 working set 사용량 = mem_working_set / mem_request * 100
network_rbps	-	byte	NetRxBytes
			Container Network Receive Byte
			컨테이너 전체 블럭 디바이스의 초당 읽기 바이트 합
network_rdropped	-	byte	NetRxDropped
			Container Network Receive Dropped
			컨테이너 네트워크 수신 dropped 건수
network_rerror	-	byte	NetRxError
			Container Network Receive Error
			컨테이너 네트워크 수신 에러 건수
network_riops	-	byte	NetRxIops
			Container Network Receive IOPS
			컨테이너 네트워크 수신 건수
network_wbps	-	byte	NetTxByes
			Container Network Transmit Byte
			컨테이너 네트워크 송신 데이터 크기
network_wdropped	-	count	NetTxDropped

Field	Type	Unit	Shortname/Name/Description
			Container Network Transmit Dropped
			컨테이너 네트워크 송신 dropped 건수
network_werror	-	count	NetTxError
			Container Network Transmit Error
			컨테이너 네트워크 송신 에러 건수
network_wiops	-	count	NetTxIops
			Container Network Transmit IOPS
			컨테이너 네트워크 송신 건수
node_cpu	-	percent	ConNodeCpu
			Container Work Node CPU Usage (%)
			컨테이너가 실행 중인 노드의 CPU 사용량
node_mem	-	percent	ConNodeMem
			Container Work Node Memory Usage (%)
			컨테이너가 실행 중인 노드의 메모리 사용량
phase	string	-	Pod 라이프사이클 ① PENDING ② RUNNING ③ SUCCEEDED ④ FAILED ⑤ UNKNOWN
restart_count	integer	-	ConRestartCnt

Field	Type	Unit	Shortname/Name/Description
			Container Restart Count
			컨테이너 재시작 횟수
state	integer	-	ConState
			Container Current State
			컨테이너 상태 코드 ① RUNNING = 114 ② PAUSE = 112 ③ RESTARTING = 101 ④ OOMKILLED = 111 ⑤ DEAD = 100 ⑥ WAITING = 119
status	string	-	ConStatus
			Container Current Status
			컨테이너 상태 정보 ① running 상태: uptime 정보 표시 ② waiting/terminated 상태: 상태에 대한 reason 정보 표시

ⓘ `mem_maxusage` 는 컨테이너가 실행되는 동안 기록된 메모리 최대 사용량을 나타냅니다. 그러나 Linux 커널 버전이 5.19보다 낮은 경우, 이 지표의 원본 데이터가 지원되지 않을 수 있습니다. 이러한 경우 해당 값이 0으로 표시될 수 있습니다. 해당 지표를 정상적으로 수집하기 위해서는 Linux 커널 버전을 5.19 이상으로 업데이트해야 합니다.

쿠버네티스 노드(`kube_node`) 메트릭

`kube_node` 카테고리는 노드에 설정된 모든 사용자 정의 라벨들을 태그로 수집합니다.

- 대상: 클러스터 프로젝트, 네임스페이스 프로젝트

- 수집 간격: 5초
- 통계 데이터: 5분, 1시간

Tags

Tag	Type	Unit	Description
nodeName	-	-	노드 이름

다음 태그가 추가되었습니다.

Tag	Type	Unit	Description
`infrastructureType`	`string`	-	노드 인프라 유형 - `baremetal`, `vm`, `unknown`

Fields

Field	Type	Unit	Description
allocatable_cpu	-	millicores	노드 할당 가능한 CPU량
allocatable_memory	-	byte	노드 할당 가능한 메모리량
allocatable_pods	integer	-	노드 할당 가능한 Pod 수
limit_cpu	-	millicores	노드 CPU Limit 합계
limit_memory	-	byte	노드 메모리 Limit 합계
pods	integer	-	노드 Pod 총 개수
request_cpu	-	millicores	노드 CPU Request 합계
request_memory	-	byte	노드 메모리 Request 합계

노드의 GPU 수량은 디바이스 타입별 병렬 리스트 필드로 수집합니다. `*_types` 와 `*_counts` 는 인덱스가 1:1로 대응하며, `nvidia.com/gpu` (whole-GPU)와 모든 MIG 프로파일(`nvidia.com/mig-*`)을 개별 항목으로 담습니다.

Field	Type	Unit	Description
gpu_capacity_types	list	-	노드 GPU capacity의 디바이스 타입 목록
gpu_capacity_counts	list	count	위 타입별 capacity 수량(인덱스 1:1 대응)
gpu_allocatable_types	list	-	노드 GPU allocatable의 디바이스 타입 목록
gpu_allocatable_counts	list	count	위 타입별 allocatable 수량(인덱스 1:1 대응)

ⓘ `v1.9.14` 부터 whole-GPU 전용 스칼라 필드 `capacity_gpu` , `allocatable_gpu` 는 제거되었습니다. MIG 환경에서 스칼라 필드가 `nvidia.com/gpu` 만 담고 MIG 슬라이스를 누락해 집계가 거짓 합계가 되는 문제를 차단하기 위함입니다. whole-GPU 수량도 리스트의 `nvidia.com/gpu` 항목으로 확인하며, 전체·타입별 집계는 `*_counts` 를 합산합니다.

카테고리	제거된 스칼라 필드	대체 리스트 필드
`kube_node`	`capacity_gpu` , `allocatable_gpu`	`gpu_capacity_types` / `gpu_capacity_counts` , `gpu_allocatable_types` / `gpu_allocatable_counts`

쿠버네티스 노드 성능(`kube_node_perf`) 메트릭

`kube_node_perf` 카테고리는 노드에 스케줄링된 모든 Pod/Container의 리소스 사용량과 할당량을 합산하여 노드 단위 성능 지표를 수집합니다.

- 대상: 클러스터 프로젝트, 네임스페이스 프로젝트
- 수집 간격: 5초
- 통계 데이터: 5분, 1시간

Tags

Tag	Type	Unit	Description
onode	-	-	노드 해시값
onodeName	-	-	노드 이름
agentPcode	-	-	에이전트 프로젝트 코드
agentOid	-	-	에이전트 고유 ID

다음 태그가 추가되었습니다.

Tag	Type	Unit	Description
cgroup_version	-	-	노드 호스트의 cgroup 마운트 모드 - `unified`(cgroup v2), `hybrid`, `legacy`(cgroup v1), `unknown`(접근 불가)

노드 호스트의 `/sys/fs/cgroup` 마운트 모드를 직접 감지해 태그로 수집합니다. 노드별 cgroup 버전 현황을 서버에서 집계할 수 있습니다.

노드 가상화 판별 태그가 추가되었습니다.

Tag	Type	Unit	Description
`virtualization`	`string`	-	노드 가상화 여부 - `baremetal`, `vm`, `unknown`
`hypervisor`	`string`	-	하이퍼바이저 벤더명 - 베어메탈이거나 판별할 수 없으면 `-`

노드 호스트의 DMI 정보(`/sys/class/dmi/id/sys_vendor`, `product_name`)에서 벤더를 먼저 대조하고, 일치하는 벤더가 없으면 `/proc/cpuinfo`의 hypervisor 플래그로 가상 머신 여부만 판정합니다. DMI 정보가 없는 비 x86 노드는 `unknown`으로 수집합니다.

QEMU/KVM, VMware, Hyper-V, Xen, VirtualBox, Parallels, Amazon EC2, Google GCE, OpenStack, Nutanix AHV, oVirt/RHV, Alibaba Cloud 등의 벤더를 인식합니다. 값은 노드 기동 중 변하지 않으므로 최초 1회 읽어 캐시합니다.

제약: Amazon EC2 `.metal` 인스턴스처럼 클라우드 사업자가 제공하는 베어메탈은 DMI 정보만으로 일반 인스턴스와 구분되지 않아 `vm` 으로 판별됩니다.

노드 에이전트 1.9.21 버전 이상에서 수집합니다.

Fields

Field	Type	Unit	Description
cpuRequest	long	millicores	노드 내 전체 컨테이너 CPU Request 합계
cpuLimit	long	millicores	노드 내 전체 컨테이너 CPU Limit 합계
cpuTotal	long	millicores	노드 내 전체 컨테이너 CPU 사용량 합계
cpuPerRequest	float	percent	노드 CPU Request 대비 실제 CPU 사용률
cpuPerLimit	float	percent	노드 CPU Limit 대비 실제 CPU 사용률
memoryRequest	long	byte	노드 내 전체 컨테이너 Memory Request 합계
memoryLimit	long	byte	노드 내 전체 컨테이너 Memory Limit 합계
memTotal	long	byte	노드 내 전체 컨테이너 Memory 사용량 합계
memoryPerRequest	float	percent	노드 Memory Request 대비 실제 Memory 사용률
memoryPerLimit	float	percent	노드 Memory Limit 대비 실제 Memory 사용률

다음 필드가 추가되었습니다.

Field	Type	Unit	Description
`conntrack_count`	`long`	`count`	현재 conntrack 엔트리 수

Field	Type	Unit	Description
conntrack_max	long	count	conntrack 엔트리 상한(nf_conntrack_max)
conntrack_pct	float	percent	상한 대비 사용률

- 엔트리 수는 호스트 network namespace 기준 값을 읽기 위해 /proc/1/net/stat/nf_conntrack 경로에서 수집합니다.
- nf_conntrack 모듈이 로드되지 않은 노드는 필드를 전송하지 않습니다. 값 0 과 미수집을 구분하기 위한 동작입니다.
- 노드 에이전트 1.9.18 버전 이상에서 수집합니다. collect_conntrack_enabled 옵션으로 수집을 끌 수 있습니다.

쿠버네티스 이벤트(kube_event) 메트릭

kube_event 카테고리는 클러스터 프로젝트의 경우 클러스터 전체를 대상으로 데이터를 수집하며 네임스페이스 프로젝트는 해당 네임스페이스에서 발생한 이벤트에 대해서만 수집합니다.

- 대상: 클러스터 프로젝트, 네임스페이스 프로젝트
- 수집 간격: 5초
- 통계 데이터: 5분, 1시간

Tags

Tag	Type	Unit	Description
field_path	-	-	Field Path
kind	-	-	이벤트가 발생한 오브젝트 종류
name	-	-	이벤트가 발생한 쿠버네티스 오브젝트명
namespace	-	-	이벤트가 발생한 네임스페이스명
reason	-	-	이벤트 발생 사유

Tag	Type	Unit	Description
type	-	-	이벤트 유형 - Warning 또는 Normal
uid	-	-	UID - 이벤트가 발생한 오브젝트

Fields

Field	Type	Unit	Description
action	string	-	액션 이름
count	-	count	이벤트 발생 건수
event_time	integer	-	이벤트 최초 발생 TimeStamp
first_timestamp	integer	-	이벤트 최초 발생 시간
last_timestamp	integer	-	이벤트 마지막 발생 시간
message	string	-	이벤트 메시지
reasonFiled	string	-	이벤트 Reason
reporting_component	string	-	현재 이벤트를 보고하는 컴포넌트
reporting_instance	string	-	현재 이벤트를 보고하는 인스턴스
series_last_observed_time	integer	-	series last observed time

쿠버네티스 클러스터(kube_stat) 메트릭

kube_stat 카테고리는 클러스터 프로젝트의 경우 클러스터 전체 대상으로 수집하고 네임스페이스 프로젝트는 해당 네임스페이스와 연관된 오브젝트를 대상으로만 수집합니다.

- 대상: 클러스터 프로젝트, 네임스페이스 프로젝트
- 수집 간격: 5초
- 통계 데이터: 5분, 1시간

Tags

Tag	Type	Unit	Description
name	-	-	kube_stat(고정 값)

다음 태그가 추가되었습니다.

Tag	Type	Unit	Description
namespace	-	-	수집 대상 네임스페이스명

Fields

Field	Type	Unit	Description
alloctable_cpu	-	millicores	클러스터 전체 코어 수(클러스터 프로젝트 Only)
alloctable_ephemeral-storage	-	byte	클러스터 전체 할당 가능한 ephemeral storage(클러스터 프로젝트 Only)
alloctable_hugepages-1gi	-	byte	클러스터 전체 할당 가능한 hugepages-1Gi(클러스터 프로젝트 Only)
alloctable_hugepages-2mi	-	byte	클러스터 전체 할당 가능한 hugepages-2Mi(클러스터 프로젝트 Only)

Field	Type	Unit	Description
alloctable_memory	-	byte	클러스터 전체 할당 가능한 메모리(클러스터 프로젝트 Only)
alloctable_pods	integer	-	할당 가능한 Pod 수
available_pod	integer	-	Phase가 Running 상태인 Pod의 수
desired_pod	integer	-	`metadata.ownerReferences` 없이 배포된 Pod 수와 쿠버네티스 오브젝트(ReplicaSet, Daemonset, StatefulSet)에 정의된 Desired Pod 수의 합
			`kubectl get pods -A`로 조회한 Pod 수와 동일
nodes	integer	-	노드 수
pod_phase_Pending	integer	-	Pending 상태 Pod 수
pod_phase_Running	integer	-	Running 상태 Pod 수
running_containers	integer	-	Running 컨테이너 수
stopped_containers	integer	-	Stopped 컨테이너 수
total_available_cpu	integer	-	할당 가능한 CPU 총합
total_available_memory	integer	-	할당 가능한 Memory 총합
total_limit_cpu	-	millicores	Limit CPU 총합
total_limit_memory	-	byte	Limit Memory 총합
total_request_cpu	-	millicores	Request CPU 총합
total_request_memory	-	byte	Request Memory 총합
unavailable_pod	integer	-	Phase가 Running 상태가 아닌(Pending, Failed, Succedded) Pod 수

Field	Type	Unit	Description
waiting_containers	integer	-	Waiting Container 수

네임스페이스에 설정된 ResourceQuota의 상한과 소진량을 수집합니다. 다음 필드가 추가되었습니다.

Field	Type	Unit	Description
`quota_count`	`long`	`count`	네임스페이스에 설정된 ResourceQuota 개수
`quota_request_cpu_hard`	`long`	`millicores`	CPU Request 쿼터 상한
`quota_request_cpu_used`	`long`	`millicores`	CPU Request 쿼터 소진량
`quota_request_memory_hard`	`long`	`byte`	Memory Request 쿼터 상한
`quota_request_memory_used`	`long`	`byte`	Memory Request 쿼터 소진량
`quota_limit_cpu_hard`	`long`	`millicores`	CPU Limit 쿼터 상한
`quota_limit_cpu_used`	`long`	`millicores`	CPU Limit 쿼터 소진량
`quota_limit_memory_hard`	`long`	`byte`	Memory Limit 쿼터 상한
`quota_limit_memory_used`	`long`	`byte`	Memory Limit 쿼터 소진량
`quota_pods_hard`	`long`	`count`	Pod 수 쿼터 상한
`quota_pods_used`	`long`	`count`	Pod 수 쿼터 소진량

- 쿠버네티스가 계산한 `status.hard`와 `status.used` 값을 그대로 사용합니다.
- 한 네임스페이스에 스코프가 다른 ResourceQuota가 여러 개 있으면 항목별로 합산합니다.
- 쿼터가 설정되지 않은 네임스페이스는 `quota_count`만 `0`으로 전송하고 나머지 필드는 전송하지 않습니다. 쿼터 미설정과 상한 `0`을 구분하기 위한 동작입니다.
- 마스터 에이전트 1.9.17 버전 이상에서 수집합니다.

권한: ClusterRole에 core API `resourcequotas`의 `list` 권한이 필요합니다. 권한이 없으면 ResourceQuota 수집만

생략되고 나머지 `kube_stat` 수집은 계속됩니다.

Pod(`kube_pod`) 메트릭

`kube_pod` 카테고리는 Pod에 설정된 모든 사용자 정의 라벨들을 태그로 수집합니다.

- 대상: 마스터(클러스터) 프로젝트, 네임스페이스 프로젝트
- 수집 간격: 5초
- 통계 데이터: 5분

Tags

Tag	Type	Unit	Description
agentOid	-	-	노드 에이전트 고유 ID(고유값)
agentPcode	-	-	프로젝트 코드(고유값)
command	-	-	실행 명령어
containerIds	-	-	Pod에 속해있는 컨테이너 ID
containerIdsCount	-	-	containerIds의 개수
containerKeys	-	-	Pod 에 속해있는 컨테이너 ID 의 해시 값
containerKeysCount	-	-	containerKeys의 개수
DaemonSet	-	-	Pod의 DaemonSet 이름
Deployment	-	-	Deployment
k8s-app	-	-	Pod의 레이블 k8s-app에 대한 값

Tag	Type	Unit	Description
microOid	-	-	Pod의 컨테이너 내부 애플리케이션에서 실행중인 에이전트의 ID
microOids	-	-	Pod 안의 복수개 컨테이너 내부 애플리케이션에서 실행 중인 에이전트의 복수개 ID
microOidsCount	-	-	microOids의 개수
name	-	-	Pod 이름
onames	-	-	Pod의 컨테이너 내부 애플리케이션에서 실행중인 에이전트의 명칭
onamesCount	-	-	onames의 개수
podName	-	-	Pod 이름
namespace	-	-	Pod가 소속된 네임스페이스
namespaceHash	-	-	Pod가 소속된 네임스페이스 해시 값
replicaSetHash	-	-	Pod의 ReplicaSet 해시 값
replicaSetName	-	-	Pod의 ReplicaSet 이름
whatap_project	-	-	Pod가 속한 와탭 프로젝트 이름

Fields

Field	Type	Unit	Shortname, Name, Description
blkio_rbps	-	byte	IoReadBytes
			Pod Block I/O Read Byte
			Pod 전체 블럭 디바이스의 초당 읽은 바이트 합
blkio_riops	-	count	IoReadIops

Field	Type	Unit	Shortname, Name, Description
			Pod Block I/O Read IOPS
			Pod 전체 블럭 디바이스의 초당 읽은 건수 합
blkio_wbps	-	byte	IoWriteBytes
			Pod Block I/O Write Byte
			Pod 전체 블럭 디바이스의 초당 쓴 바이트 합
blkio_wiops	-	count	IoWriteIops
			Pod Block I/O Write IOPS
			Pod 전체 블럭 디바이스의 초당 쓴 건수 합
cpu_per_limit	-	percent	CpuByLimit
			Pod CPU Usage by Limit (%)
			CPU Limit 기준 CPU 전체 사용률
cpu_per_request	-	percent	CpuByRequest
			Pod CPU Usage by Request (%)
			CPU Request 기준 CPU 전체 사용률
cpu_quota_percent	-	percent	CpuLimitByNode
			Pod CPU Limit by Node (%)
			노드 CPU 대비 Pod CPU Limit 할당량 - Limit 미설정인 경우 Pod가 작동 중인 해당 노드의 CPU 전체 코어가 퍼센트로 표시됨

Field	Type	Unit	Shortname, Name, Description
cpu_sys	-	percent	CpuSysByNode
			Pod CPU Sys Usage by Node (%)
			노드 CPU 대비 Pod CPU System 사용률
cpu_throttledperiods	-	count	CpuThrottledCnt
			Pod CPU Throttling Count
			Pod CPU Throttled 건수
cpu_throttledtime	-	nanosecond(ns)	CpuThrottledTime
			Pod CPU Throttling Time
			Pod CPU Throttled 시간
cpu_total	-	percent	CpuByNode
			Pod CPU Usage by Node (%)
			노드 CPU 대비 Pod CPU 사용률
cpu_total_milli	-	millicores	CpuTotUsage
			Pod CPU Usage (millicore)
			Pod CPU 사용량
cpu_user	-	percent	CpuUserByNode
			Pod CPU User Usage by Node (%)
			노드 CPU 대비 Pod CPU User 사용률

Field	Type	Unit	Shortname, Name, Description
cpu_request	-	millicores	CpuRequest
			Pod CPU Request (core)
			Pod CPU 요청
cpu_per_request	-	percent	CpuByRequest
			Pod CPU Usage by Request (%)
			Pod CPU 요청 대비 사용률 = cpu_total_milli / cpu_request * 100
mem_totalcache	-	byte	MemTotCache
			Pod Memory Total Cache (byte)
			Pod 전체 캐시 크기
mem_totalpgfault	-	count	MemTotPageFaultCnt
			Pod Memory Total Page Fault Count
			Pod Page Fault 횟수
mem_totalrss	-	byte	MemTotRss
			Pod Memory Total RSS (byte)
			Pod RSS 메모리 전체 크기
mem_totalrss_percent	-	percent	MemTotRssByLimit
			Pod Memory Total RSS by Limit (%)
			Pod RSS 메모리 전체 사용률

Field	Type	Unit	Shortname, Name, Description
mem_totalunevictable	-	byte	MemTotUnevictable
			Pod Memory Total Unevictable (byte)
			Pod Unevictable Memory 전체 크기
mem_usage	-	byte	MemUsage
			Pod Memory Usage (byte)
			Pod 메모리 사용량
mem_working_set	-	byte	MemWs
			Pod Memory Working Set (byte)
			Pod 메모리 working set = mem_usage - inactive file
memory_request	-	byte	MemRequest
			Pod Memory Request (byte)
			Pod 메모리 요청량
memory_limit	-	byte	MemLimit
			Pod Memory Limit (byte)
			Pod 메모리 Limit 할당량
memory_per_request	-	percent	MemByRequest
			Pod Memory Working Set By Request (%)
			Pod 메모리 요청량 기준 Working Set 사용량

Field	Type	Unit	Shortname, Name, Description
memory_per_limit	-	percent	MemByLimit
			Pod Memory Working Set By Limit (%)
			Pod 메모리 Limit 기준 Working Set 사용량
network_rbps	-	byte	NetRxBytes
			Pod Network Receive Byte
			Pod 전체 블럭 디바이스의 초당 읽기 바이트 합
network_rdropped	-	byte	NetRxDropped
			Pod Network Receive Dropped
			Pod 네트워크 수신 dropped 건수
network_rerror	-	byte	NetRxError
			Pod Network Receive Error
			Pod 네트워크 수신 에러 건수
network_riops	-	byte	NetRxIops
			Pod Network Receive IOPS
			Pod 네트워크 수신 건수
network_wbps	-	byte	NetTxByes
			Pod Network Transmit Byte
			Pod 네트워크 송신 데이터 크기

Field	Type	Unit	Shortname, Name, Description
network_wdropped	-	count	NetTxDropped
			Pod Network Transmit Dropped
			Pod 네트워크 송신 dropped 건수
network_werror	-	count	NetTxError
			Pod Network Transmit Error
			Pod 네트워크 송신 에러 건수
network_wiops	-	count	NetTxIops
			Pod Network Transmit IOPS
			Pod 네트워크 송신 건수
phase	string	-	Phase
			Pod Current Phase
			Pod 라이프사이클 ① PENDING ② RUNNING ③ SUCCEEDED ④ FAILED ⑤ UNKNOWN

다음 필드는 내부용으로 예약되었습니다.

Field	Type	Unit	Description
kube_sless_normal	-	-	쿠버네티스 정보성 이벤트 발생 건수
kube_sless_warning	-	-	쿠버네티스 경고 이벤트 발생 건수

Field	Type	Unit	Description
micro_sful_critical	-	-	apm 심각 상태 기반 이벤트 발생 건수
micro_sful_info	-	-	apm 정보성 상태 기반 이벤트 발생 건수
micro_sful_warning	-	-	apm 경고 상태 기반 이벤트 발생 건수
micro_sless_critical	-	-	apm 심각 무상태 이벤트 발생 건수
micro_sless_info	-	-	apm 정보성 무상태 이벤트 발생 건수
micro_sless_warning	-	-	apm 경고 무상태 이벤트 발생 건수
sful_critical	-	-	메트릭스 심각 상태 기반 이벤트 발생 건수
sful_info	-	-	메트릭스 정보성 상태 기반 이벤트 발생 건수
sful_warning	-	-	메트릭스 경고 상태 기반 이벤트 발생 건수
sless_critical	-	-	메트릭스 심각 무상태 이벤트 발생 건수
sless_info	-	-	메트릭스 정보성 무상태 이벤트 발생 건수
sless_warning	-	-	메트릭스 경고 무상태 이벤트 발생 건수

Pod의 GPU Request·Limit 수량은 디바이스 타입별 병렬 리스트 필드로 수집합니다. `*_types` 와 `*_counts` 는 인덱스가 1:1로 대응하며, `nvidia.com/gpu` (whole-GPU)와 모든 MIG 프로파일(`nvidia.com/mig-*`)을 개별 항목으로 담습니다.

Field	Type	Unit	Description
gpu_request_types	list	-	Pod GPU Request의 디바이스 타입 목록
gpu_request_counts	list	count	위 타입별 Request 수량(인덱스 1:1 대응)
gpu_limit_types	list	-	Pod GPU Limit의 디바이스 타입 목록
gpu_limit_counts	list	count	위 타입별 Limit 수량(인덱스 1:1 대응)

> v1.9.14 부터 whole-GPU 전용 스칼라 필드 gpu_request , gpu_limit 은 제거되었습니다. MIG 환경에서 스칼라 필드가 nvidia.com/gpu 만 담고 MIG 슬라이스를 누락해 집계가 거짓 합계가 되는 문제를 차단하기 위함입니다. whole-GPU 수량도 리스트의 nvidia.com/gpu 항목으로 확인하며, 전체·타입별 집계는 *_counts 를 합산합니다.

쿠버네티스 Pod 통계(kube_pod_stat) 메트릭

kube_pod_stat 카테고리 클러스터 프로젝트는 클러스터 전체를 대상으로 데이터를 수집하며 네임스페이스 프로젝트는 해당 네임스페이스에 소속된 Pod에 대해서만 수집합니다.

- 대상: 클러스터 프로젝트, 네임스페이스 프로젝트
- 수집 간격: 5초
- 통계 데이터: 5분, 1시간

Tags

Tag	Type	Unit	Description
kind	-	-	유형 - 클러스터 프로젝트의 경우 cluster 고정 값을 가지며 네임스페이스 프로젝트의 경우 Deployment 또는 ReplicaSet만 수집
name	-	-	쿠버네티스 자원 이름 - 클러스터 프로젝트는 name 값이 없으며 네임스페이스 프로젝트는 Deployment 또는 ReplicaSet의 이름

Fields

Field	Type	Unit	Description
available_pod	integer	-	Phase가 Running 상태인 Pod의 수
desired_pod	integer	-	metadata.ownerReferences 없이 배포된 Pod 수와 쿠버네티스

Field	Type	Unit	Description
			오브젝트(ReplicaSet, Daemonset, StatefulSet)에 정의된 Desired Pod 수의 합
			`kubectl get pods -A`로 조회한 Pod 수와 동일
limit_cpu	-	millicores	CPU Limit 사용량
limit_memory	-	byte	Memory Limit 사용량
request_cpu	-	millicores	CPU Request 사용량
request_memory	-	byte	Memory Request 사용량
running_container	integer	-	Running Container 수
stopped_container	integer	-	Stopped Container 수
waiting_container	integer	-	Waiting Container 수

쿠버네티스 Pod 볼륨(`kube_pod_volume`) 메트릭

`kube_pod_volume` 카테고리는 노드에 실행 중인 Pod에 마운트된 볼륨의 용량·i-node 사용량을 수집합니다. 앱 파드의 PVC 볼륨은 디렉토리명이 PV 이름이므로 `volumeName` 태그로 PVC의 `spec.volumeName`과 조인할 수 있습니다.

Tags

Tag	Type	Unit	Description
hostDevice	-	-	볼륨이 위치한 호스트 디바이스
hostFileSystem	-	-	볼륨의 파일 시스템

Tag	Type	Unit	Description
hostMountPath	-	-	호스트 상 볼륨 마운트 경로
volumeName	-	-	볼륨 이름 - PVC 볼륨의 경우 PV 이름(예: `pvc-...`). PVC의 `spec.volumeName`과 조인 가능
volumePlugin	-	-	볼륨 플러그인 - 예: `kubernetes.io~csi`, `kubernetes.io~nfs`, `volume-subpaths`

Fields

Field	Type	Unit	Description
capacity	long	byte	볼륨 전체 용량
available	long	byte	볼륨 사용 가능 용량
availablePercent	float	percent	볼륨 사용 가능 용량 비율
usedPercent	float	percent	볼륨 사용률
free	long	byte	볼륨 여유 용량
freePercent	float	percent	볼륨 여유 용량 비율
totalInode	long	count	볼륨 전체 i-node 수
usedInode	long	count	볼륨 사용 중인 i-node 수
usedInodePercent	float	percent	볼륨 i-node 사용률

emptyDir 볼륨 수집이 추가되었습니다. 기본(디스크) medium emptyDir은 별도 마운트가 아니라 노드 루트 파일 시스템의 디렉터리여서 `df` 기반 수집에 잡히지 않았습니다. 노드 에이전트가 kubelet 볼륨 디렉터리를 직접 조회해 같은 `kube_pod_volume` 카테고리로 전송합니다.

- 조회 경로: `{HOST_PREFIX}{kubelet_root_dir}/pods/{podUid}/volumes/kubernetes.io~empty-`

dir/{volumeName}

- 수집 주기: 60초. 디렉터리 조회 비용이 커서 5초 수집 주기와 분리합니다.
- emptyDir 볼륨의 volumePlugin 태그 값은 kubernetes.io~empty-dir 입니다.

다음 태그가 추가되었습니다.

Tag	Type	Unit	Description
volumeType	string	-	볼륨 종류 - emptyDir 볼륨은 EmptyDir
emptyDirMedium	string	-	emptyDir 저장 매체 - Memory 또는 Disk. 파드 spec의 emptyDir.medium 기준이며 미지정은 Disk
emptyDirSpecConfirmed	string	-	파드 spec 확인 여부 - true는 spec으로 확인된 값, false는 kubelet 경로와 파일 시스템 종류로 추정한 값

다음 필드가 추가되었습니다.

Field	Type	Unit	Description
usedSpace	long	byte	볼륨 사용량 합계
fileCount	long	count	볼륨 내 파일 수
sizeLimit	long	byte	파드 spec의 emptyDir.sizeLimit - 미설정 시 전송하지 않음
truncated	integer	-	수집 상한 초과 여부 - 1이면 상한에 걸려 값이 실제보다 작음

capacity , available , usedPercent 는 볼륨 용량을 확정할 수 있을 때만 전송합니다.

- sizeLimit 이 있으면 그 값을 볼륨 용량으로 사용합니다.
- sizeLimit 이 없고 medium이 Memory 면 kubelet이 만든 tmpfs 총량을 용량으로 사용합니다. tmpfs 총량의 기본값은

노드 메모리의 절반입니다.

- `sizeLimit`이 없는 `Disk` medium은 상한이 없어 용량을 만들지 않습니다. 이 경우 파드의 `container.ephemeral_storage_limit`과 비교하세요.

노드 에이전트 1.9.23 버전 이상에서 위 태그와 필드를 모두 전송합니다. 수집 활성화와 비용 상한 옵션은 쿠버네티스 에이전트 옵션 설정을 참조하세요.

쿠버네티스 Horizontal Pod Autoscaler(HPA)(`kube_hpa_stat`) 메트릭

와탭에서 사용하는 ClusterRole에 HPA가 추가되어야 지표 수집이 시작됩니다.

- 대상: 클러스터 프로젝트
- 수집 간격: 5초
- 통계 데이터: 5분, 1시간

Tags

Tag	Type	Unit	Description
name	-	-	HPA 이름

다음 태그가 추가되었습니다.

Tag	Type	Unit	Description
`scaleTargetKind`	`string`	-	스케일 대상 오브젝트 종류(`spec.scaleTargetRef.kind`)
`scaleTargetName`	`string`	-	스케일 대상 오브젝트 이름(`spec.scaleTargetRef.name`)
`scaleTargetApiVersion`	`string`	-	스케일 대상 오브젝트 API 버전
`creationTimeMillis`	`long`	-	HPA 생성 시각(epoch ms) - 값이 없으면 전송하지 않음

HPA의 스케일 대상은 `spec.scaleTargetRef`에 있습니다. `ownerReferences` 기반인 `owner_kind`, `owner_name`, `owner_id` 태그는 HPA에서 대부분 빈 값이므로 스케일 대상은 `scaleTarget*` 태그로 확인하세요. 기존 태그는 하위 호환을 위해 유지합니다.

마스터 에이전트 1.9.24 버전 이상에서 수집합니다.

Fields

Field	Type	Unit	Description
currentReplicas	integer	count	현재 레플리카 수
desiredReplicas	integer	count	Desired 레플리카 수
lastScaleTime	integer	count	마지막으로 스케일이 변경된 TimeStamp
maxReplicas	integer	count	최대 레플리카 수
minReplicas	integer	count	최소 레플리카 수

쿠버네티스 HPA 목록(`kube_hpa`) 메트릭

`kube_hpa` 카테고리는 `kubectl get hpa -o wide` 결과와 동등한 HPA 목록 데이터를 수집합니다. 시계열 지표 성격인 `kube_hpa_stat`과는 별개 카테고리입니다.

- 대상: 클러스터 프로젝트
- 수집 간격: 30초
- 요구 버전: 마스터 에이전트 1.9.24 이상

Tags

Tag	Type	Unit	Description
hpaUid	string	-	HPA UID(고유값)
hpaName	string	-	HPA 이름
namespace	string	-	HPA가 소속된 네임스페이스
creationTimeMillis	long	-	HPA 생성 시각(epoch ms)
scaleTargetKind	string	-	스케일 대상 오브젝트 종류
scaleTargetName	string	-	스케일 대상 오브젝트 이름
scaleTargetApiVersion	string	-	스케일 대상 오브젝트 API 버전

Fields

Field	Type	Unit	Description
minReplicas	integer	count	최소 레플리카 수
maxReplicas	integer	count	최대 레플리카 수
currentReplicas	integer	count	현재 레플리카 수
desiredReplicas	integer	count	Desired 레플리카 수
targetMetricType	list	-	스케일 기준 메트릭 종류 목록
targetValue	list	-	메트릭별 목표값 목록
currentValue	list	-	메트릭별 현재값 목록

Field	Type	Unit	Description
conditionAbleToScale	string	-	AbleToScale 컨디션 상태
conditionScalingActive	string	-	ScalingActive 컨디션 상태
conditionScalingLimited	string	-	ScalingLimited 컨디션 상태

targetMetricType, targetValue, currentValue는 같은 인덱스가 한 메트릭에 대응하는 병렬 리스트입니다. HPA는 CPU, 메모리, 사용자 정의 메트릭을 동시에 가질 수 있어 값 하나로는 표현되지 않습니다.

- 값은 kubectl 표기와 같은 문자열입니다(예: 50%, 100m).
- metrics-server가 없어 현재값을 확보하지 못하면 <unknown>으로 수집합니다.
- Resource 메트릭 외에 Pods, External, Object 메트릭은 각각 pods:, external:, object: 접두어를 붙여 담습니다.
- autoscaling/v2를 우선 사용하며 미지원 클러스터는 autoscaling/v1로 대체합니다.

ClusterRole에 horizontalpodautoscalers 권한이 이미 포함되어 있어 추가 권한 설정은 필요하지 않습니다.

프로세스(kube_process) 메트릭

> ⓘ 쿠버네티스 에이전트 1.7.12 버전 이상이 필요합니다. 에이전트 업데이트에 대한 자세한 내용은 다음 문서를 참조하세요.

노드에 존재하는 쿠버네티스 관련 프로세스를 모니터링할 때 수집됩니다.

- 대상: 클러스터 프로젝트, 네임스페이스 프로젝트
- 수집 간격: 5초
- 통계 데이터: 5분

Tags

Tag	Type	Unit	Description
ppid	string	-	부모 프로세스 ID `/proc/[pid]/status::PPid`
pid	string	-	프로세스 ID `/proc/[pid]/status::Pid`
cmd1	string	-	명령어 이름 `/proc/[pid]/status::Name`
cmd2	string	-	명령어 라인(전체 명령어와 인자) `/proc/[pid]/cmdline`
user	string	-	사용자 ID 또는 사용자 이름 `/proc/[pid]/status::Uid`
onodeName	string	-	프로세스의 노드 이름 컨테이너 시스템 환경변수 (`NODE_IP`)
createTime	TimeStamp	-	프로세스 시작 시간 `/proc/uptime` 통해 계산된 필드

Fields

Field	Type	Unit	Description
cpu	float	percent(%)	CPU 사용률 - `/proc/[pid]/stat`을 통해 계산된 필드
memory	float	percent(%)	메모리 사용률 - `/proc/[pid]/statm`을 통해 계산된 필드
rss	long	byte	실제 메모리 사용량 (Resident Set Size)

Field	Type	Unit	Description
			- `/proc/[pid]/status`의 VmRSS
uid	string	-	사용자 ID 또는 사용자 이름 - `/proc/[pid]/status`의 Uid
state	string	-	프로세스 상태 - `/proc/[pid]/status`의 State
sharedMemory	long	byte	공유 메모리 크기 - `/proc/[pid]/statm`을 통해 계산된 필드
openFileDescriptors	integer	-	프로세스가 열어둔 파일 디스크립터 수 - `/proc/[pid]/fd`를 통해 계산된 필드
vmSize	long	byte	가상 메모리 크기 (Virtual Memory Size) - `/proc/[pid]/status`의 VmSize
threads	integer	-	프로세스가 생성한 스레드 수 - `/proc/[pid]/status`의 Threads

Kubernetes 환경의 리눅스 프로세스 상태

리눅스에서 /proc/[pid]/status 파일의 `State` 필드가 프로세스의 현재 상태를 표시합니다. 각 상태 값의 의미는 다음과 같습니다.

Code	Description
R (Running)	실행 중 - 프로세스가 실행 중이거나 실행될 준비가 되어 있는 상태
S (Sleeping)	대기 중 - 인터럽트 가능한 수면 상태로, 대기 중이며 이벤트를 기다리고 있는 상태
D (Disk Sleep)	디스크 수면 - 인터럽트 불가능한 수면 상태로, 일반적으로 I/O 작업을 기다리는 중인 상태

Code	Description
R (Zombie)	좀비 상태 - 프로세스가 종료되었으나 아직 부모 프로세스가 종료 상태를 수집하지 않은 상태
T (Stopped)	중지 - 프로세스가 작업 제어 신호에 의해 중지된 상태 (SIGSTOP 등) 또는 디버거에 의해 중지된 상태
t (Tracing stop)	추적 중지 - 디버거에 의해 추적 중인 상태 (소문자 t로 표시됨)
X (Dead)	죽은 상태 - 프로세스가 죽은 상태 (일반적으로 보지 않음)
x (Dead)	죽은 상태 - 커널 쓰레드의 죽은 상태 (일반적으로 보지 않음)
K (WakeKill)	강제 종료 - 깨우기 신호를 무시하고 즉시 죽은 상태
W (Waking)	깨우기 중 - 깨우기 신호를 받아 깨워지는 중인 상태
I (Idle)	유휴 상태 - 커널 스레드가 유휴 상태 (일반적으로 유저 공간 프로세스에는 보이지 않음)

ⓘ Kubernetes는 컨테이너와 노드의 리소스를 효율적으로 관리하기 때문에, 컨테이너 내부에서 실행되는 다수의 프로세스가 실제로 대기 상태를 유지합니다. 이에 따라 대부분의 프로세스는 `Sleeping` 상태일 수 있습니다.

에이전트 상태(`agent_status_summary`) 메트릭

에이전트 상태와 관련한 지표를 10초 간격으로 수집한 카테고리입니다.

Fields

Field	Type	Unit	Description
inActTime	-	milliseconds(ms)	에이전트가 비활성화된 상태로 유지된 시간
isActive	Boolean	-	현재 에이전트의 활성 상태 여부 (true / false)
isRestart	Boolean	-	에이전트가 재시작되었는지 여부 (true / false)
lastActTime	-	milliseconds(ms)	마지막으로 에이전트가 활성화된 상태의 시각 - 0: 비활성화된 경우
oid	-	-	프로젝트에 포함된 각 에이전트의 고유 식별자
oType	-	-	에이전트 종류 - 1: 애플리케이션 에이전트 - 2: subType 참조
startTime	-	milliseconds(ms)	에이전트가 시작된 시점을 나타내는 타임스탬프
subType	-	-	에이전트 종류 - 9: 노드 에이전트 - 10: 마스터 에이전트

Ingress(kube_ingress) 메트릭

> ⓘ 쿠버네티스 에이전트 1.7.13 버전 이상이 필요합니다. 에이전트 업데이트에 대한 자세한 내용은 다음 문서를 참조하세요.

Ingress 리소스에 대한 메타 데이터와 관련 정보를 모니터링할 때 수집됩니다.

- 대상: 클러스터 프로젝트, 네임스페이스 프로젝트

- 수집 간격: 30초
- 통계 데이터: 5분

Tags

Tag	Type	Unit	Description
ingressUid	string	-	Ingress 리소스의 고유 ID
ingressName	string	-	Ingress 리소스의 이름
ingressNamespace	string	-	Ingress 리소스의 네임스페이스
creationTimeMillis	long	milliseconds(ms)	Ingress 리소스 생성 시간
ingressClassName	string	-	Ingress 클래스 이름
ingressLoadBalancerIps	List	-	Ingress 로드밸런서 IP

Fields

Field	Type	Unit	Description
host	List	-	Ingress 리소스가 수신하는 호스트 이름 (`*`인 경우 모든 호스트에 대해 적용됨)
path	List	-	특정 호스트 하위에서 요청 경로
backendServiceName	List	-	백엔드로 전달되는 서비스의 이름
backendServicePort	List	-	백엔드로 전달되는 포트 번호
backendServiceUid	List	-	백엔드로 전달되는 서비스의 uid
pathType	List	-	경로 일치 방식 (예, `Prefix`, `Exact`)

쿠버네티스 크론잡(kube_cronjob) 메트릭

디플로이먼트 리소스에 대한 메타데이터와 상태 정보를 태그로 수집합니다.

Tags

Tag	Type	Unit	Description
cronJobName	string	-	CronJob의 이름 (metadata.name)
namespace	string	-	CronJob이 속한 네임스페이스 (metadata.namespace)
cronJobUid	string	-	CronJob의 UID (metadata.uid)

Fields

Field	Type	Unit	Description
lastScheduleTime	string	-	CronJob이 마지막으로 스케줄된 시간 (status.lastScheduleTime)
lastSuccessfulTime	string	-	CronJob이 마지막으로 성공한 시간 (status.lastSuccessfulTime)
cronJobSpecSchedule	string	-	CronJob의 스케줄 주기 (spec.schedule , 예: "0 */5 * * *")
successfulJobsHistoryLimit	integer	count	성공한 Job을 보존하는 최대 개수 (spec.successfulJobsHistoryLimit)
failedJobsHistoryLimit	integer	count	실패한 Job을 보존하는 최대 개수 (spec.failedJobsHistoryLimit)
concurrencyPolicy	string	-	동시 실행 정책 (Allow , Forbid , Replace) (spec.concurrencyPolicy)

Field	Type	Unit	Description
startingDeadlineSeconds	integer	time(seconds)	스케줄된 시간이 지난 후에도 Job을 실행할 수 있는 최대 지연 시간 (spec.startingDeadlineSeconds)

쿠버네티스 디플로이먼트(kube_deployment) 메트릭

디플로이먼트 리소스에 대한 메타데이터와 상태 정보를 태그로 수집합니다.

Tags

Tag	Type	Unit	Description
deployName	string	-	디플로이먼트의 이름 (metadata.name)
namespace	string	-	디플로이먼트가 속한 네임스페이스 (metadata.namespace)
deployUid	string	-	디플로이먼트의 UID (metadata.uid)

Fields

Field	Type	Unit	Description
deployReadyReplicas	integer	count	준비(Ready) 상태인 파드 수 (status.readyReplicas)
deployTotalReplicas	integer	count	설정된 전체 파드 수 (spec.replicas)
deployUpdatedReplicas	integer	count	최신 버전으로 갱신된 파드 수 (status.updatedReplicas)
deployAvailableReplicas	integer	count	사용 가능한 상태의 파드 수 (status.availableReplicas)
deployUnavailableReplicas	integer	count	사용 불가능한 상태의 파드 수 (status.unavailableReplicas)

Field	Type	Unit	Description
deployCreationTime	integer	milliseconds(ms)	디플로이먼트 생성 시각 (Unix Epoch milliseconds, metadata.creationTimestamp)
deploySelector	string	-	디플로이먼트의 선택자 라벨 (spec.selector.matchLabels)
deployStrategy	string	-	디플로이먼트의 배포 전략 (spec.strategy.type, 예: RollingUpdate, Recreate)

쿠버네티스 데몬셋(kube_daemonset) 메트릭

데몬셋 리소스에 대한 메타데이터와 상태 정보를 태그로 수집합니다.

Tags

Tag	Type	Unit	Description
daemonSetName	string	-	DaemonSet의 이름 (metadata.name)
namespace	string	-	DaemonSet이 소속된 네임스페이스 (metadata.namespace)
daemonSetUid	string	-	DaemonSet의 UID (metadata.uid)
creationTime	integer	milliseconds(ms)	DaemonSet 생성 시각 (metadata.creationTimestamp)
age	string	-	DaemonSet 생성 이후 경과 시간

Fields

Field	Type	Unit	Description
currentNumberScheduled	integer	count	현재 스케줄된 파드 수 (status.currentNumberScheduled)

Field	Type	Unit	Description
desiredNumberScheduled	integer	count	기대하는 파드 수 (status.desiredNumberScheduled)
numberReady	integer	count	Ready 상태인 파드 수 (status.numberReady)
numberAvailable	integer	count	사용 가능한 파드 수 (status.numberAvailable)
numberMisscheduled	integer	count	잘못 스케줄된 파드 수 (status.numberMisscheduled)
updatedNumberScheduled	integer	count	업데이트된 파드 수 (status.updatedNumberScheduled)
daemonSetSelector	string	-	Selector 라벨 (spec.selector.matchLabels)

쿠버네티스 레플리카셋(kube_replicaset) 메트릭

레플리카셋(ReplicaSet) 리소스에 대한 메타데이터와 상태 정보를 태그로 수집합니다.

Tags

Tag	Type	Unit	Description
replicaSetName	string	-	ReplicaSet의 이름 (metadata.name)
namespace	string	-	ReplicaSet이 소속된 네임스페이스 (metadata.namespace)
replicaSetUid	string	-	ReplicaSet의 UID (metadata.uid)
ownerKind	string	-	상위 오브젝트 종류 (예: Deployment) (metadata.ownerReferences.kind)
ownerName	string	-	상위 오브젝트 이름 (metadata.ownerReferences.name)
ownerUid	string	-	상위 오브젝트 UID (metadata.ownerReferences.uid)

Tag	Type	Unit	Description
creationTime	integer	milliseconds(ms)	ReplicaSet 생성 시각 (metadata.creationTimestamp)
age	string	-	ReplicaSet 생성 이후 경과 시간

Fields

Field	Type	Unit	Description
replicaSetReplicas	integer	count	설정된 파드 수 (spec.replicas)
replicaSetReadyReplicas	integer	count	Ready 상태 파드 수 (status.readyReplicas)
replicaSetAvailableReplicas	integer	count	사용 가능한 파드 수 (status.availableReplicas)
replicaSetSelector	string	-	Selector 라벨 (spec.selector.matchLabels)

쿠버네티스 스테이트풀셋(kube_statefulSet) 메트릭

스테이트풀셋(StatefulSet) 리소스에 대한 메타데이터와 상태 정보를 태그로 수집합니다.

Tags

Tag	Type	Unit	Description
statefulSetName	string	-	StatefulSet의 이름 (metadata.name)
namespace	string	-	StatefulSet이 소속된 네임스페이스 (metadata.namespace)
statefulSetUid	string	-	StatefulSet의 UID (metadata.uid)
creationTime	integer	milliseconds(ms)	StatefulSet 생성 시각 (metadata.creationTimestamp)

Tag	Type	Unit	Description
age	string	-	StatefulSet 생성 이후 경과 시간

Fields

Field	Type	Unit	Description
replicas	integer	count	설정된 파드 수 (spec.replicas)
readyReplicas	integer	count	Ready 상태 파드 수 (status.readyReplicas)
currentReplicas	integer	count	현재 생성된 파드 수 (status.currentReplicas)
updatedReplicas	integer	count	업데이트된 파드 수 (status.updatedReplicas)
statefulSetSelector	string	-	Selector 라벨 (spec.selector.matchLabels)

쿠버네티스 잡(kube_job) 메트릭

디플로이먼트 리소스에 대한 메타데이터와 상태 정보를 태그로 수집합니다.

Tags

Tag	Type	Unit	Description
jobName	string	-	Job의 이름 (metadata.name)
namespace	string	-	Job이 소속된 네임스페이스 (metadata.namespace)
jobUid	string	-	Job의 UID (metadata.uid)
ownerKind	string	-	상위 오브젝트의 종류 (예: CronJob)

Tag	Type	Unit	Description
ownerName	string	-	상위 오브젝트의 이름
ownerUid	string	-	상위 오브젝트의 UID

Fields

Field	Type	Unit	Description
succeededPods	integer	count	성공한 파드 수 (status.succeeded)
failedPods	integer	count	실패한 파드 수 (status.failed)
completionTime	string	-	Job이 완료된 시간 (status.completionTime.toString())
parallelism	integer	count	동시에 실행되도록 설정된 파드 수 (spec.parallelism)
completions	integer	count	Job이 완료되기 위해 필요한 파드 실행 횟수 (spec.completions)
backoffLimit	integer	count	실패 시 재시도 가능한 최대 횟수 (spec.backoffLimit)
activeDeadlineSeconds	integer	seconds(sec)	Job이 실행될 수 있는 최대 시간 (spec.activeDeadlineSeconds)

쿠버네티스 NetworkPolicy(kube_network_policy) 메트릭

kube_network_policy 카테고리는 클러스터의 NetworkPolicy 오브젝트를 이벤트 기반으로 수집합니다. 주기 수집이 아니라 오브젝트가 추가, 변경, 삭제될 때마다 해당 시점의 스냅숏을 전송합니다.

- 대상: 클러스터 프로젝트, 네임스페이스 프로젝트
- 수집 방식: 이벤트 기반

Tags

Tag	Type	Unit	Description
networkPolicyUid	string	-	NetworkPolicy UID(고유값)
networkPolicyName	string	-	NetworkPolicy 이름
namespace	string	-	NetworkPolicy가 소속된 네임스페이스
resourceVersion	string	-	오브젝트 리비전(metadata.resourceVersion)

Fields

Field	Type	Unit	Description
generation	long	-	오브젝트 세대(metadata.generation)
creationTimeMillis	long	-	오브젝트 생성 시각(epoch ms)
policyTypes	list	-	정책 유형 목록 - Ingress, Egress. 빈 목록은 API 기본값인 Ingress
selectedPodUids	list	-	정책이 선택한 파드 UID 목록
podSelector	map	-	정책 대상 파드 셀렉터(matchLabels, matchExpressions)
ingressRules	list	-	Ingress 규칙 목록 - 규칙별 from, ports 포함
egressRules	list	-	Egress 규칙 목록 - 규칙별 to, ports 포함
specSchemaVersion	integer	-	spec 필드 스키마 버전
specHash	long	-	정규화한 spec의 해시값

Field	Type	Unit	Description
			- 내용 변경 여부 판별용

규칙은 중첩 구조로 전송해 쿠버네티스 API의 의미를 그대로 보존합니다.

- 한 peer 안의 `namespaceSelector`와 `podSelector`는 AND 관계입니다.
- peer 목록과 규칙 목록은 OR 관계입니다.
- 규칙 안의 빈 `from`, `to`는 모든 peer를, 빈 `ports`는 모든 포트를 의미합니다.
- peer의 `ipBlock`은 `cidr`과 `except`를 담습니다.
- 포트는 `protocol`, `portType`, `portNumber` 또는 `portName`, `endPort`를 담습니다.

ClusterRole에 `networkpolicies` 조회 권한이 필요합니다. `collect_network_policy_enabled` 옵션으로 수집을 끌 수 있습니다.

GPU 메트릭

DCGM Exporter를 통해 수집되는 NVIDIA GPU 메트릭 목록입니다. GPU 수집 방식과 사이드카 패턴은 지원 환경 > NVIDIA GPU 문서를 참조하세요.

Metric	Type	Unit	Description
DCGM_FI_DEV_GPU_UTIL	gauge	%	GPU 코어가 활발하게 연산 작업을 수행한 시간의 비율
DCGM_FI_PROF_GR_ENGINE_ACTIVE	gauge	ratio	그래픽 엔진(3D/Compute)이 활성 상태인 시간의 비율
DCGM_FI_PROF_SM_ACTIVE	gauge	ratio	하나 이상의 워프(Warp)가 활성 상태인 사이클의 비율
DCGM_FI_PROF_SM_OCCUPANCY	gauge	ratio	SM(Streaming Multiprocessor)의 점유율 (SM 당 상주 워프 비율)
DCGM_FI_PROF_PIPE_TENSOR_ACTIVE	gauge	ratio	텐서 코어(HMMA) 파이프가 활성 상태인 사이클의 비율
DCGM_FI_PROF_DRAM_ACTIVE	gauge	ratio	메모리 인터페이스가 활성 상태(데이터 송수신)인 사이클의 비율
DCGM_FI_DEV_FB_USED	gauge	MiB	사용 중인 프레임버퍼(VRAM) 메모리
DCGM_FI_DEV_FB_FREE	gauge	MiB	사용할 수 있는 프레임버퍼(VRAM) 메모리
DCGM_FI_DEV_FB_TOTAL	gauge	MiB	총 프레임버퍼(VRAM) 메모리
DCGM_FI_DEV_FB_USED_PERCENT	gauge	%	사용 중인 프레임버퍼(VRAM) 메모리의 백분율
DCGM_FI_DEV_FB_RESERVED	gauge	MiB	예약된 프레임버퍼 메모리
DCGM_FI_PROF_PCIE_TX_BYTES	counter	byte	PCIe 버스를 통해 전송(Host → Device)된 총 데이터양
DCGM_FI_PROF_PCIE_RX_BYTES	counter	byte	PCIe 버스를 통해 수신(Device → Host)된 총 데이터양
DCGM_FI_DEV_POWER_USAGE	gauge	W	현재 GPU의 전력 소비량

Metric	Type	Unit	Description
DCGM_FI_DEV_GPU_TEMP	gauge	°C	GPU 칩의 현재 온도
DCGM_FI_DEV_SM_CLOCK	gauge	MHz	SM(Streaming Multiprocessor) 클럭 주파수
DCGM_FI_DEV_MEM_CLOCK	gauge	MHz	메모리 클럭 주파수
DCGM_FI_DEV_PSTATE	gauge	P-State	GPU의 현재 성능 상태 (P0가 가장 높음)
DCGM_FI_DEV_ECC_SBE_AGG_TOTAL	counter	count	누적된 단일 비트(Single-Bit) 영구 ECC 오류 수
DCGM_FI_DEV_ECC_DBE_AGG_TOTAL	counter	count	누적된 이중 비트(Double-Bit) 영구 ECC 오류 수

다음 메트릭이 추가되었습니다.

Metric	Type	Unit	Description
DCGM_FI_DEV_WEIGHTED_GPU_UTIL	gauge	ratio	GPU 사용률을 디바이스가 점유한 슬라이스 비율로 가중한 값

MIG 환경과 비 MIG 환경의 GPU 사용률을 같은 기준으로 합산하기 위한 메트릭입니다. MIG 프로파일은 물리 GPU의 일부만 점유하므로 사용률을 그대로 더하면 GPU 단위 총량과 맞지 않습니다.

산출 방식은 calculation_method 라벨로 구분합니다.

Value	Description
weighted_sum	MIG 인스턴스. DCGM_FI_PROF_GR_ENGINE_ACTIVE에 슬라이스 비율(compute_slices / DCGM_FI_DEV_MIG_MAX_SLICES)을 곱해 합산
direct	비 MIG GPU. DCGM_FI_DEV_GPU_UTIL을 100으로 나눈 값
prof_gr_engine_active	H100, H200, H800, H20 계열의 비 MIG GPU. DCGM_FI_PROF_GR_ENGINE_ACTIVE 값을 그대로 사용

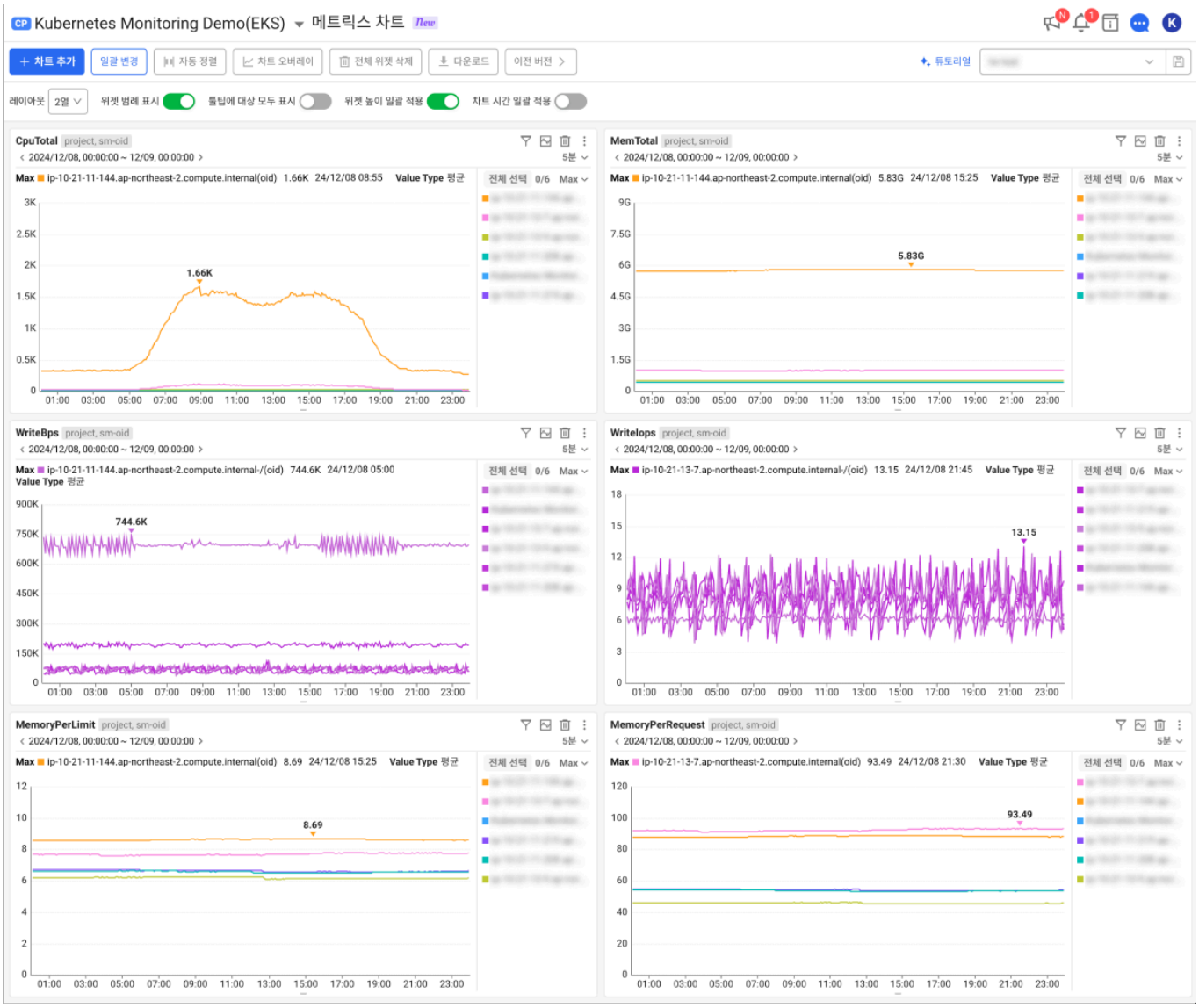
- 값의 범위는 0 부터 1 까지입니다.

- `DCGM_FI_DEV_GPU_UTIL`이 100을 넘는 비정상 값은 100으로 제한한 뒤 계산합니다.
- GPU UUID는 `DCGM_FI_DEV_UUID` 라벨로 함께 제공합니다.

메트릭스 차트

홈 화면 > 프로젝트 선택 > **분석** > **메트릭스 차트** *New*

메트릭스 차트 *New* 메뉴는 와탭 모니터링 솔루션의 핵심 기능으로, 다양한 모니터링 대상으로부터 수집한 데이터를 시계열 차트로 시각화하여 제공합니다. 이를 통해 사용자는 시스템 성능을 효과적으로 모니터링하고 이상 징후를 빠르게 식별하여 문제 해결에 필요한 인사이트를 얻을 수 있습니다.



- 다양한 지표를 시계열 차트를 통해서 시각적으로 확인할 수 있으며, 장애 발생 시점의 여러 지표를 한번에 분석할 수 있습니다.
- 특정 시간 범위를 선택하여 과거 데이터를 조회하고, 해당 기간의 성능을 분석할 수 있습니다.
- 특정 시간대나 조건에 맞는 데이터를 필터링하여 보다 정확한 분석을 수행할 수 있습니다.
- 사용자는 원하는 차트를 화면에 배치하고 레이아웃을 자유롭게 조정할 수 있습니다.
- 사용자가 원하는 지표들을 배치하고 프리셋으로 저장하여 필요시 불러올 수 있습니다.

> **기존 메트릭스 차트 메뉴와의 차이점**
>
> Service 2.8.0 업데이트 이후 변경 사항은 다음과 같습니다. 업데이트 이후 변경된 사항을 화면에서 확인하려면 오른쪽 상단의 **튜토리얼** 버튼을 선택하세요.
>
> - 사용자가 시간, 대상, 인터벌 옵션을 설정한 다음 모니터링하려는 대상의 지표를 특정하고 차트를 추가할 수 있도록 사용자 경험(UX)을 개선했습니다.
> - 대시보드에 배치한 위젯을 삭제하지 않고 차트 조회 시간 범위를 변경할 수 있습니다.
> - 대시보드에 배치한 모든 위젯의 옵션을 일괄로 적용할 수 있는 **일괄 변경** 기능을 제공합니다.
> - 프리셋 기능을 통해 대시보드에 배치한 위젯의 정보를 저장하고 불러올 수 있습니다.
> - 제공하는 다양한 옵션을 통해 대시보드의 레이아웃 및 위젯의 기능을 조정할 수 있습니다.

차트 위젯 추가하기

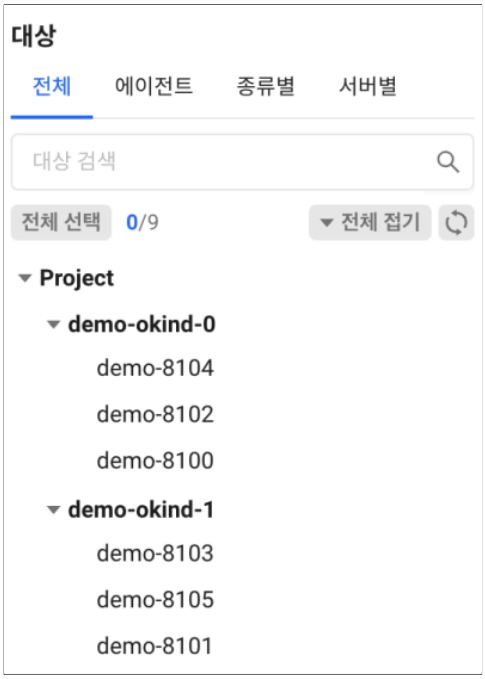
대시보드에 지표를 위젯으로 추가하려면 화면 왼쪽 상단의 **차트 추가** 버튼을 선택하세요.

조회 시간 설정하기

시간에서 지표를 조회할 날짜와 시간을 설정하세요.

대상 선택하기

프로젝트에 포함된 에이전트 중 원하는 대상을 선택하세요. 개별 에이전트를 선택하거나 에이전트 설정을 통해 분류한 그룹 단위로 선택할 수 있습니다.

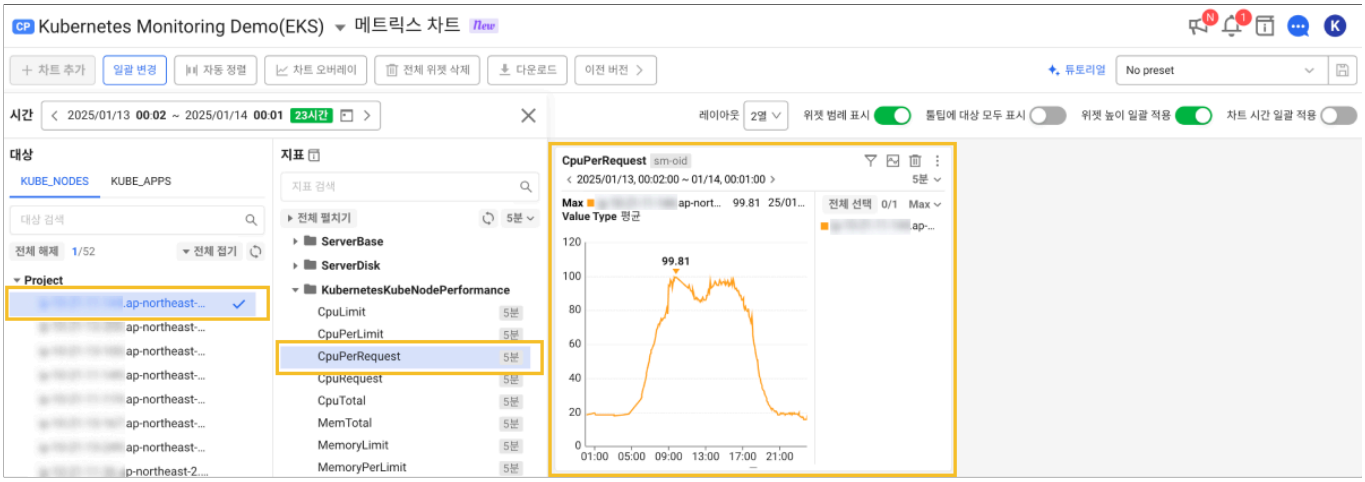


- **에이전트**: 개별 에이전트를 선택하거나 모두 선택할 수 있습니다.
- **종류별**: 에이전트 설정에서 `whatap.okind` 로 분류된 그룹 단위로 에이전트를 모니터링합니다.
- **서버별**: 에이전트 설정에서 `whatap.onode` 로 분류된 그룹 단위로 에이전트를 모니터링합니다.

> ⓘ
> - 설정한 시간과 날짜에 따라 선택할 수 있는 대상이 변경될 수 있습니다.
> - 에이전트 목록에서 마우스 클릭해 대상을 선택하거나 해제할 수 있습니다. `Shift` 키를 이용해 여러 개의 에이전트를 한번에 선택할 수도 있습니다.
> - **전체 선택** 또는 **전체 해제** 버튼을 선택하면 관련 하위 대상들을 한번에 선택 또는 해제할 수 있습니다.
> - 원하는 에이전트가 목록에 표시되지 않는다면 **새로고침** 버튼을 선택하세요.
> - **전체 접기** 또는 **전체 펼치기** 버튼을 선택해 그룹 단위로 목록을 감추거나 펼칠 수 있습니다.

지표 선택하기

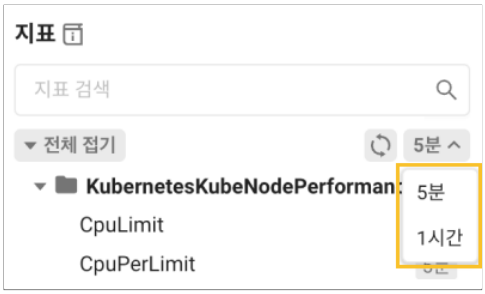
모니터링할 대상 에이전트를 선택한 후 **지표** 패널에 선택할 수 있는 목록이 표시됩니다. 목록에서 원하는 지표를 선택하세요.



선택한 지표에 해당하는 차트 위젯이 오른쪽의 대시보드에 배치됩니다. 다른 지표를 추가로 선택해 차트 위젯을 추가할 수 있습니다. 차트 영역을 확대해 넓게 보려면 **시간** 옵션 오른쪽의 ✕ 버튼을 선택하세요. **대상**과 **지표** 패널이 사라지고 차트 영역만 화면에 표시됩니다.

- **지표 검색** 입력란에 문자열을 입력하면 원하는 지표를 빠르게 검색할 수 있습니다.
- 지표의 카테고리 및 필드 이름을 기준으로도 검색할 수 있습니다. 카테고리 및 필드에 대한 자세한 내용은 다음 문서를 참조하세요.
- **전체 접기** 또는 **전체 펼치기** 버튼을 선택해 카테고리 단위로 지표 목록을 감추거나 펼칠 수 있습니다.
- 원하는 지표 항목이 목록에 표시되지 않는다면 **새로고침** 버튼을 선택하세요.

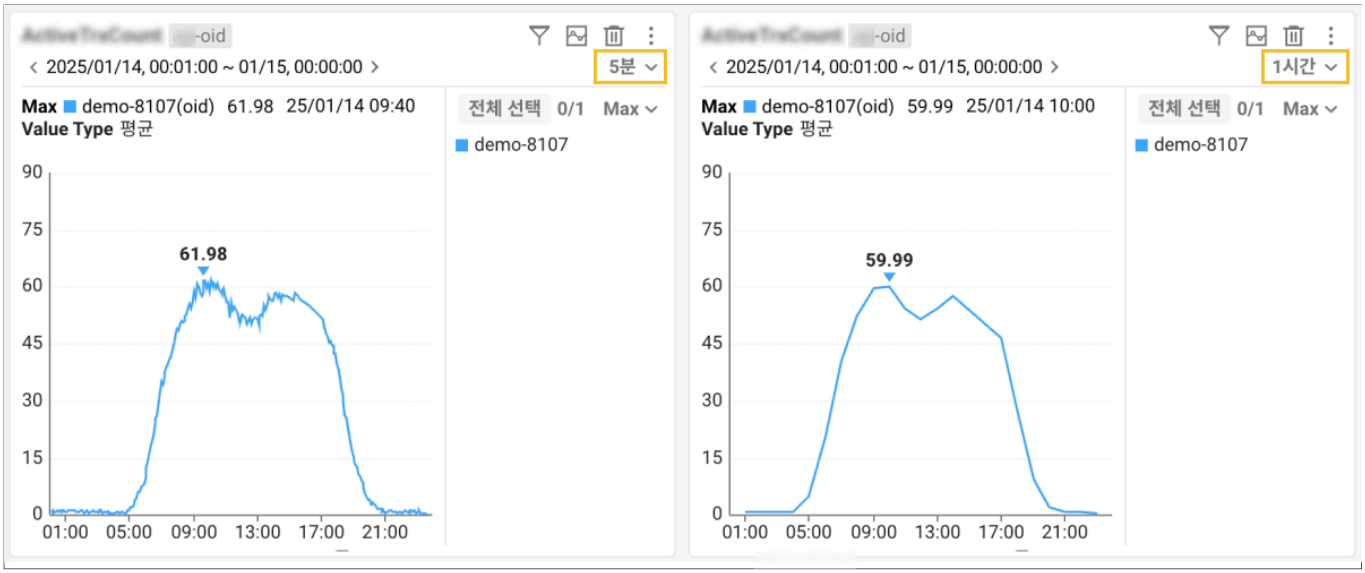
인터벌 설정하기



차트마다 인터벌을 설정할 수 있어 데이터의 집계 단위를 조정하여 다양한 시간 간격의 데이터를 시각화할 수 있습니다. **지표** 패널의 오른쪽 상단에 ⏲ 버튼을 선택하면 인터벌 옵션을 선택할 수 있습니다.

지표마다 기본 설정된 인터벌 시간이 다르며 조회 시간 범위에 따라 인터벌 시간이 변경됩니다.

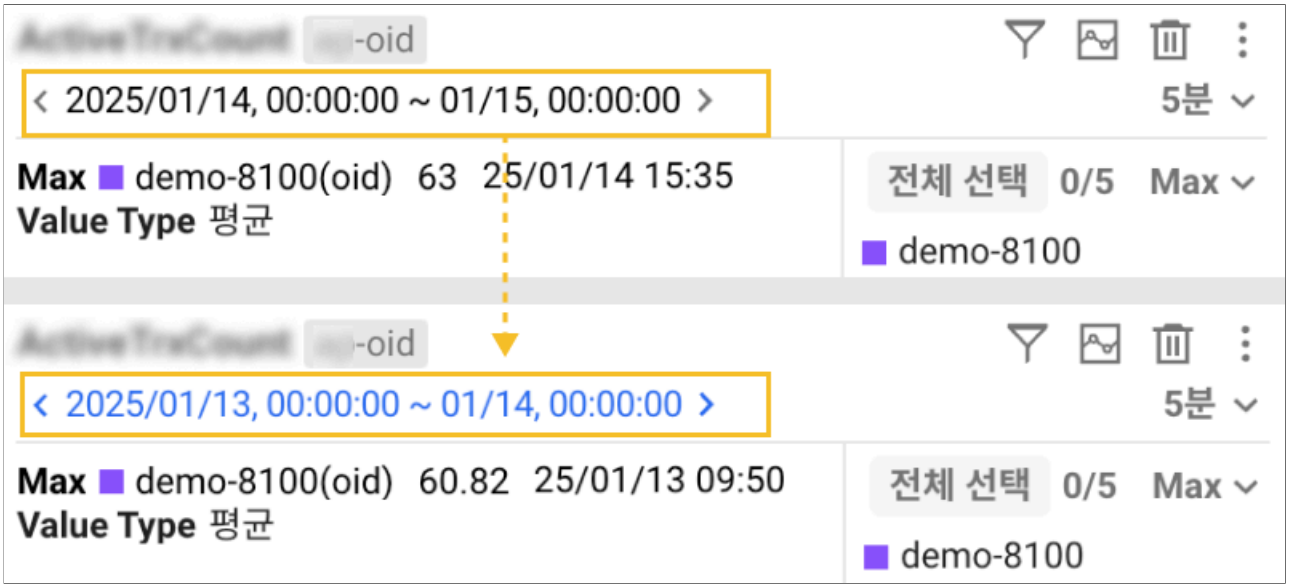
예를 들어, **5분** 인터벌은 5분 단위의 평균값을 표시하고, **1시간** 인터벌은 1시간 단위의 평균값을 표시합니다.



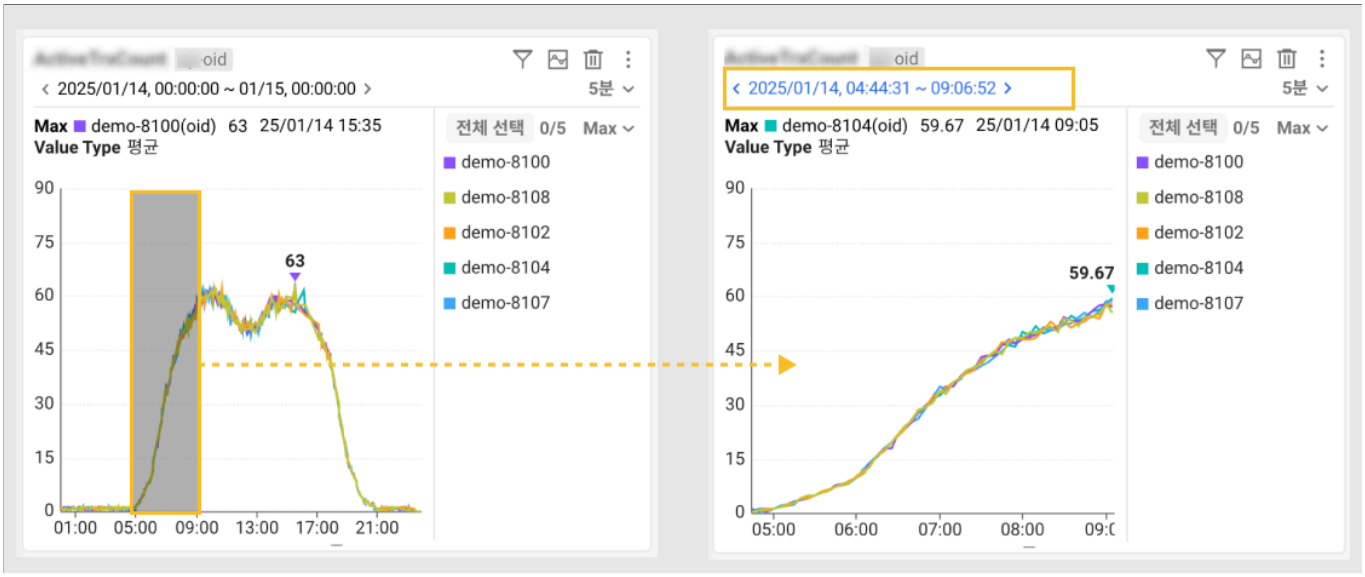
인터벌 옵션을 조정하여 짧은 시간 간격의 세부 데이터부터 긴 시간 간격의 전체 추세까지 다양한 관점에서 데이터를 분석할 수 있습니다.

조회 시간 변경하기

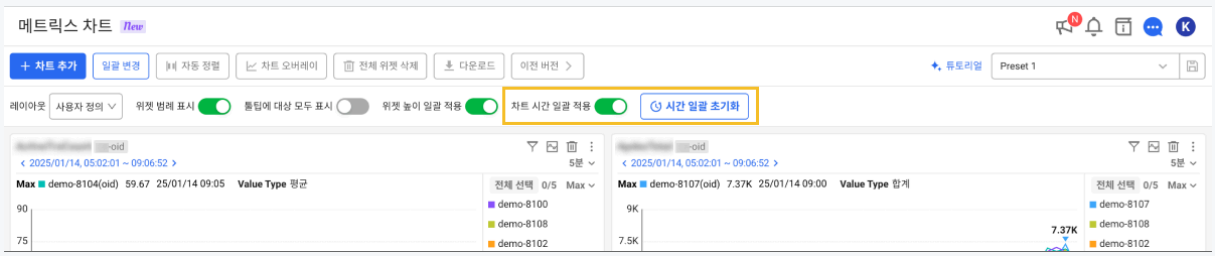
개별 위젯의 차트 조회 시간 범위를 변경하려면 위젯 상단의 < 또는 > 버튼을 선택하세요. 시간 변경 범위는 **시간** 옵션에 설정된 시간 기준과 같습니다. 예를 들어, **시간** 옵션에서 **1일**로 설정되어 있을 때, < 또는 > 버튼을 선택하면 시간을 1일 단위로 변경할 수 있습니다.



또는 차트의 특정 영역을 드래그하면, 드래그한 범위만큼 시간을 변경해 조회할 수도 있습니다.



- 화면 상단에 **차트 시간 일괄 적용** 옵션을 활성화한 상태에서 개별 위젯의 시간을 변경하면 모든 위젯의 시간을 변경할 수 있습니다.



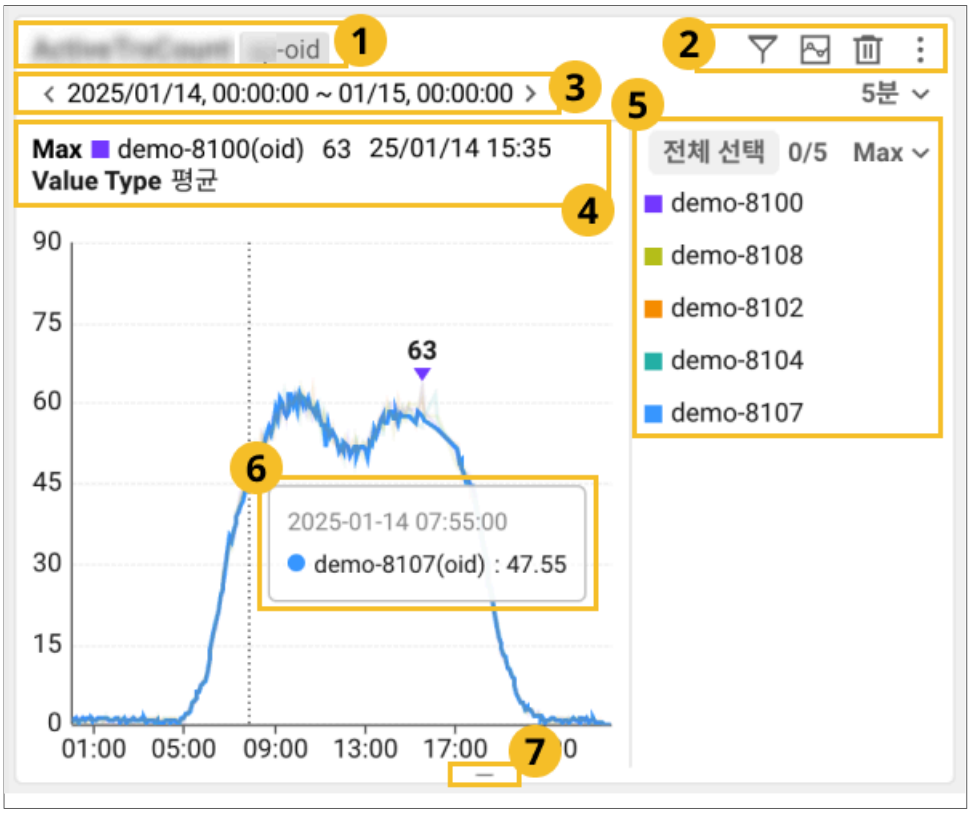
- 조회 시간 범위를 변경한 다음 최근 시간으로 다시 변경하려면 버튼을 선택하세요.

  만약 화면 상단에 **차트 시간 일괄 적용** 옵션을 활성화한 상태라면 **시간 일괄 초기화** 버튼이 표시됩니다. 이 버튼을 선택하면 모든 위젯의 시간을 초기화할 수 있습니다.

일괄 변경 기능을 이용해 전체 위젯의 시간을 변경하고, 모니터링 대상과 인터벌도 수정할 수 있습니다. 자세한 내용은 다음 문서를 참조하세요.

차트 위젯 상세 안내

차트 위젯에 표시되는 각 항목에 대한 자세한 내용을 확인하세요.



- 1 **지표 이름 및 카테고리**: 차트 위젯의 지표 이름을 표시합니다. 지표의 오른쪽에는 조회 대상의 카테고리를 태그 형식으로 표시합니다. 이용하는 제품에 따라 카테고리 명칭은 다를 수 있습니다.
  - `*oid` : 개별 에이전트를 선택했을 때 표시됩니다.
  - `project` : 프로젝트 단위로 대상을 선택했을 때 표시됩니다.
  - `okind` : 에이전트 설정에서 `whatap.okind` 로 분류된 그룹을 선택했을 때 표시됩니다.
  - `onode` : 에이전트 설정에서 `whatap.onode` 로 분류된 그룹을 선택했을 때 표시됩니다.
- 2 **위젯 옵션**: 차트 위젯의 옵션 메뉴입니다. 자세한 내용은 다음 문서를 참조하세요. 인터벌 옵션에 대한 자세한 내용은 다음 문서를 참조하세요.
- 3 **시간**: 차트의 데이터를 조회하는 시간 범위를 표시합니다. < 또는 > 버튼을 선택해 조회 시간 범위를 조정할 수 있습니다.
- 4 **데이터**: 모니터링 대상 중 조회한 데이터의 최댓값(**Max**)과 데이터 병합 방식(**Value Type**)을 확인할 수 있습니다.
- 5 **범례**: 모니터링 대상을 범례로 표시합니다. 범례 표시에 대한 자세한 내용은 다음 문서를 참조하세요.
- 6 **차트**: 조회한 데이터를 시계열 그래프로 표시합니다. 마우스 포인터를 그래프 위로 가져가면 툴팁을 통해 상세 정보를 확인할 수 있습니다. 차트의 특정 영역을 드래그하면, 드래그한 범위만큼 시간을 변경해 조회할 수 있습니다.

- 7 **크기 조절**: 차트 위젯의 가운데 하단 영역을 드래그하면 위젯의 크기를 조절할 수 있습니다.

차트 위젯 옵션 이용하기

차트 위젯의 오른쪽 상단에 위치한 옵션 버튼을 통해 다양한 기능을 이용할 수 있습니다.

대상 필터링하기

화면에 추가한 차트 위젯에 모니터링 대상을 추가하거나 제외할 수 있습니다.

1. 버튼을 선택하면 모니터링 대상을 선택할 수 있는 팝업 메뉴가 나타납니다.

2. 화면의 안내에 따라 원하는 대상을 선택하세요.
3. **적용** 버튼을 선택하세요.

상세 조회하기

차트의 데이터를 더 넓은 화면에서 자세한 정보를 확인하려면 버튼을 선택하세요. **상세 조회** 창이 나타납니다. 개별 대상에 대한 차트와 평균 및 최댓값을 조회할 수 있습니다.

- **1일전**: 하루 전의 데이터와 비교해 확인할 수 있습니다.
- **1주전**: 한 주 전의 데이터와 비교해 확인할 수 있습니다.
- **CSV**: 조회한 데이터를 CSV 형식의 파일로 다운로드할 수 있습니다.
- 그래프 차트 하단의 아이콘을 드래그하면 차트의 크기를 조절할 수 있습니다.

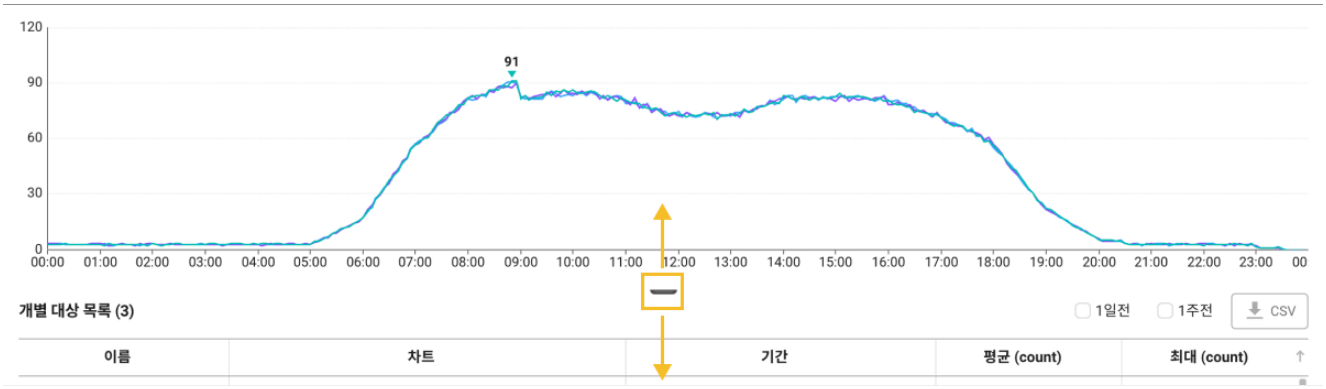
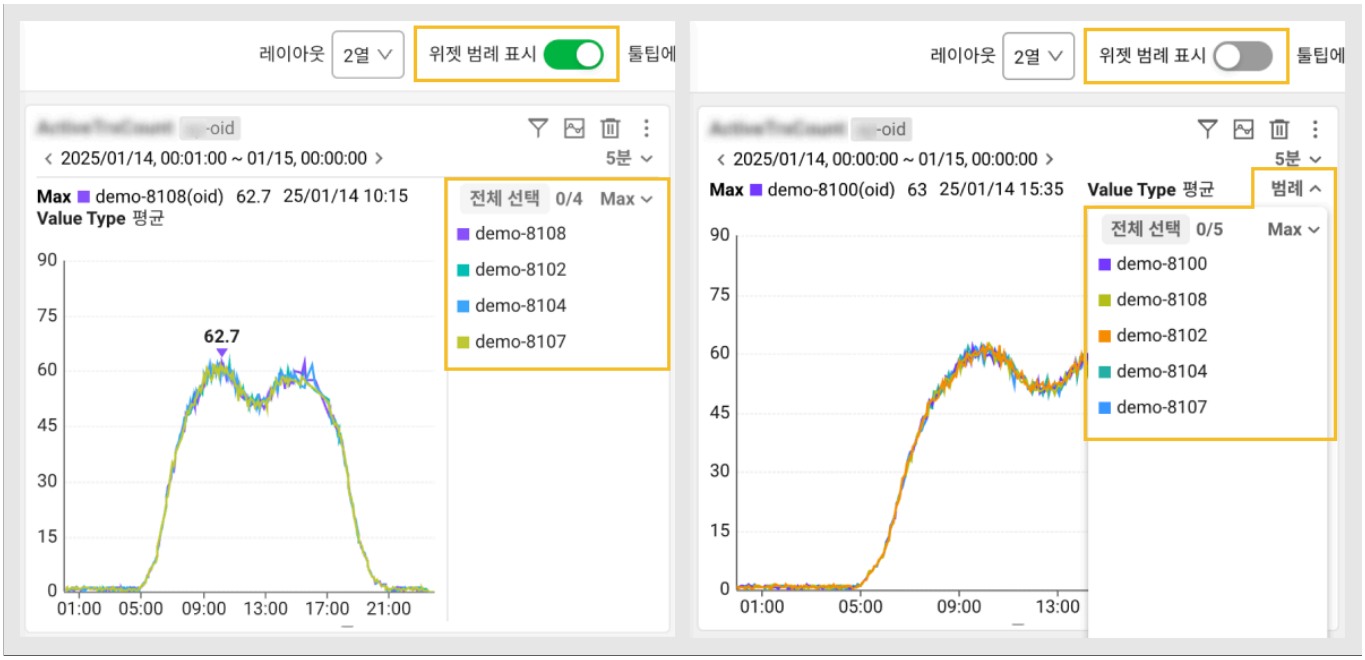


차트 위젯 삭제하기

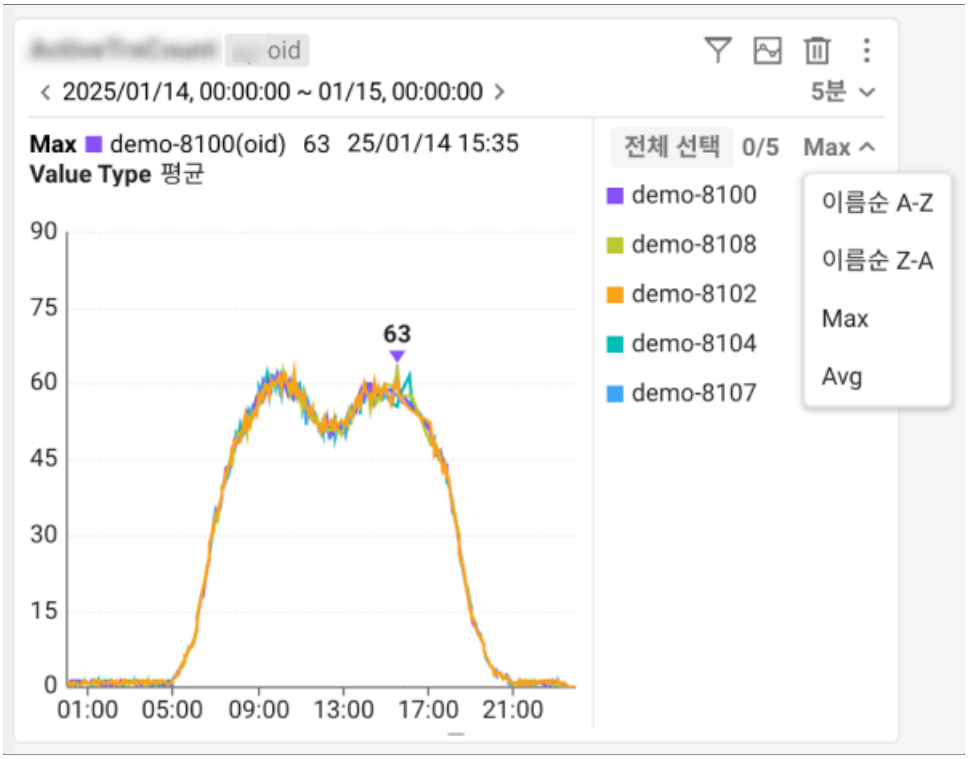
화면에 배치한 차트 위젯을 삭제하려면 버튼을 선택하세요.

범례 활용하기

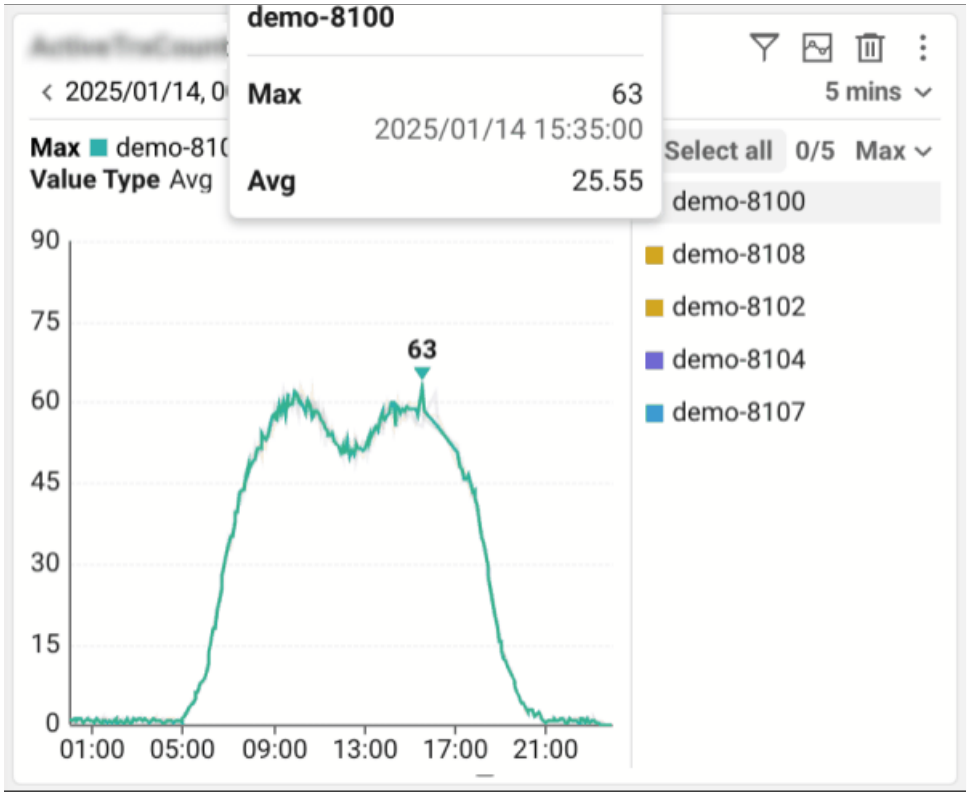
범례를 통해서 차트에 표시되는 모니터링 대상을 확인하고, 원하는 기준으로 정렬하거나 필터링할 수 있습니다. 화면 상단의 **위젯 범례 표시** 옵션을 켜거나 끄면 위젯의 차트 오른쪽에 범례를 표시하거나 숨길 수 있습니다. 옵션을 끈 상태에서는 **범례**를 클릭해 모니터링 대상을 확인하고, 선택할 수 있습니다.



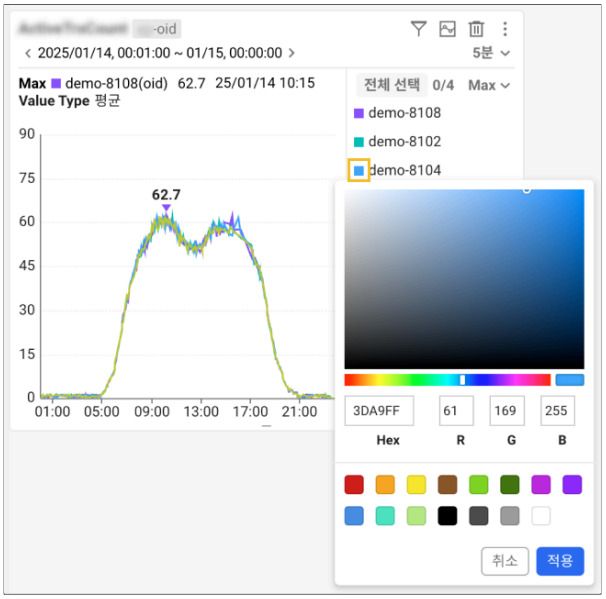
- **전체 선택/전체 해제** 버튼을 선택해 모니터링 대상을 선택하거나 해제할 수 있습니다.
- 범례 목록에서는 모니터링 대상을 이름순, 최댓값, 평균값 기준으로 정렬할 수 있습니다.



- 범례 목록에서 각 항목에 마우스를 오버하면 해당 모니터링 대상에 대한 그래프를 강조해서 확인할 수 있고, Max 값이 기록된 특정 시간을 툴팁으로 표시합니다. 또는 범례 목록의 대상을 선택해 필터링할 수도 있습니다.


demo-8100
Max 63
2025/01/14 15:35:00
Avg 25.55
5 mins
Select all 0/5 Max
demo-8100
demo-8108
demo-8102
demo-8104
demo-8107
Max demo-810
Value Type Avg
90
75
60
45
30
15
0
63
01:00 05:00 09:00 13:00 17:00 21:00

범례 색상 변경하기

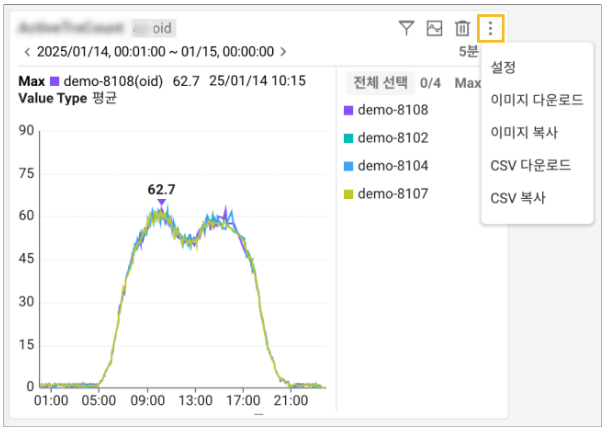

2025/01/14, 00:01:00 ~ 01/15, 00:00:00
5분
Max demo-8108(oid) 62.7 25/01/14 10:15
Value Type 평균
전체 선택 0/4 Max
demo-8108
demo-8102
demo-8104
62.7
3DA9FF
61
169
255
Hex
R
G
B
취소
적용

모니터링 대상에 해당하는 그래프의 색상을 변경할 수 있습니다. 범례 목록에서 색상을 변경하려는 대상의 왼쪽에 네모 상자를 선택하세요. 원하는 색상을 선택한 다음 **적용** 버튼을 클릭하세요.

> ⓘ 설정한 색상은 프리셋 기능을 통해서 저장되지 않습니다. 이 기능은 향후 업데이트를 통해 제공할 예정입니다.

추가 옵션 이용하기

차트 위젯 오른쪽 상단에 ⋮ 버튼을 선택하면 다음의 추가 옵션을 이용할 수 있습니다.



- **설정**
  - **자동 Y축 사용**: 조회한 데이터 범위 내에서 Y축의 크기를 자동 조정할 수 있습니다.
  - **최대값 표시**: 차트의 그래프 영역에 최댓값을 표시할 수 있습니다.
- **이미지 다운로드**: 차트를 이미지 형식의 파일로 다운로드할 수 있습니다.
- **이미지 복사**: 차트를 이미지 형식으로 클립보드에 복사할 수 있습니다.
- **CSV 다운로드**: 차트에서 조회한 데이터를 CSV 형식의 파일로 다운로드할 수 있습니다.
- **CSV 복사**: 차트에서 조회한 데이터를 CSV 형식으로 클립보드에 복사할 수 있습니다.

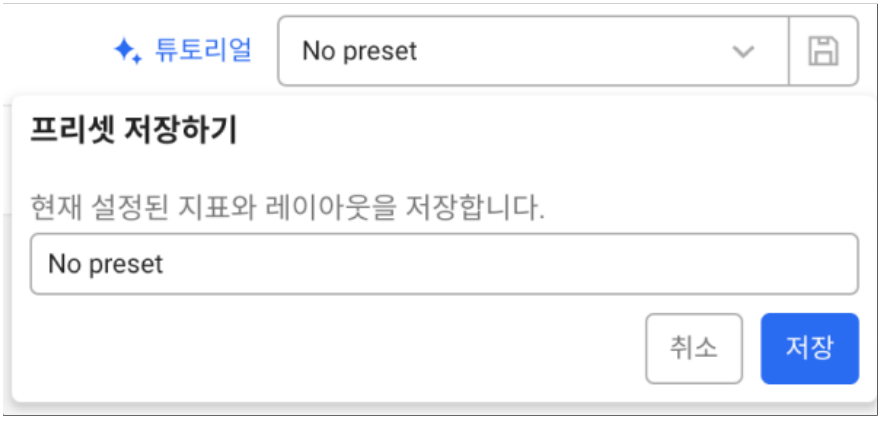
프리셋 설정하기

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

ⓘ **프로젝트 수정** 권한이 있는 멤버만 프리셋을 저장하거나 수정할 수 있습니다. 멤버 권한에 대한 자세한 내용은 다음 문서를 참조하세요.

새로운 프리셋 만들기

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



4. **저장** 버튼을 선택하세요.

프리셋 목록에서 새로 저장한 프리셋을 확인할 수 있습니다.

ⓘ
- 새로 만든 프리셋에 변경 사항이 생겼다면 다시 프리셋을 저장해야 합니다. 버튼을 선택한 다음 같은 이름으로 프리셋을 저장하세요. 기존의 프리셋에 변경 사항을 덮어쓰기합니다.
- 대시보드의 변경 사항을 저장하지 않고 다른 메뉴로 이동하면 변경 사항은 저장되지 않습니다.
- 프리셋은 프로젝트 단위로 저장되어 다른 사용자와 공유할 수 있습니다.
- 프리셋에 저장되는 대상은 대시보드의 레이아웃과 대시보드에 배치한 위젯, 위젯에서 수정한 옵션(**대상**, **인터벌**)입니다. 화면 상단의 **시간** 옵션은 저장되지 않습니다.
- 프리셋을 불러온 다음 위젯의 조회 시간을 변경하려면 **일괄 변경** 기능을 이용하세요. 자세한 내용은 다음 문서를 참조하세요.

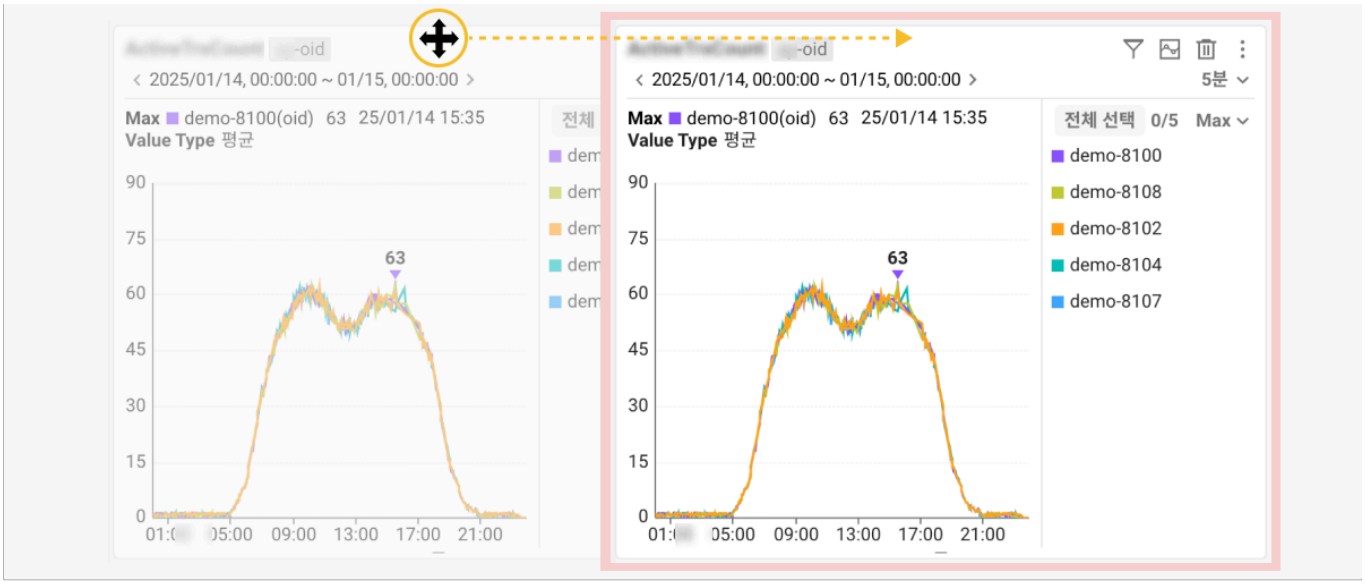
프리셋 삭제하기

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

대시보드 위젯 편집하기

대시보드에 배치한 위젯을 사용자가 원하는 위치에 배치하거나 크기를 조절할 수 있습니다.

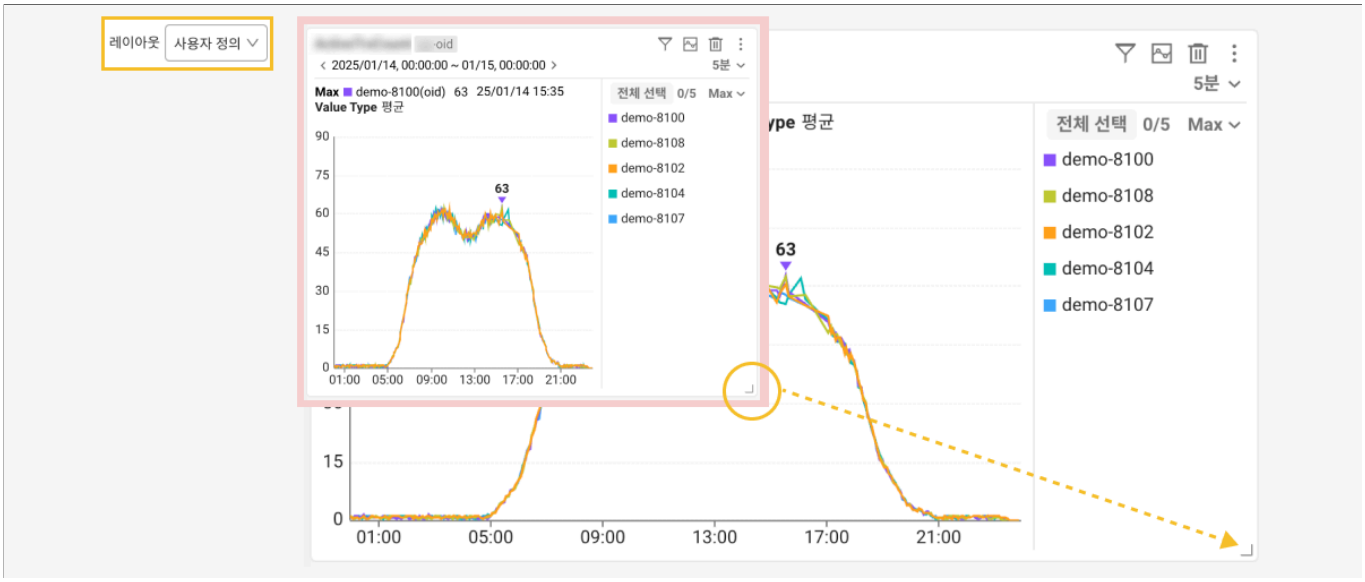
위젯 이동하기



위젯의 상단 부분을 마우스 포인터로 이동하면 ✢ 모양으로 변경됩니다. 이때 마우스로 클릭한 상태로 원하는 위치로 드래그하여 위젯을 이동할 수 있습니다.

위젯 크기 조절하기

대시보드 옵션의 **레이아웃**이 **사용자 정의**으로 설정되어 있으면 위젯의 크기를 조절할 수 있습니다.



위젯의 오른쪽 아래에 ⌟ 요소를 마우스로 클릭한 상태에서 원하는 크기로 드래그하세요.

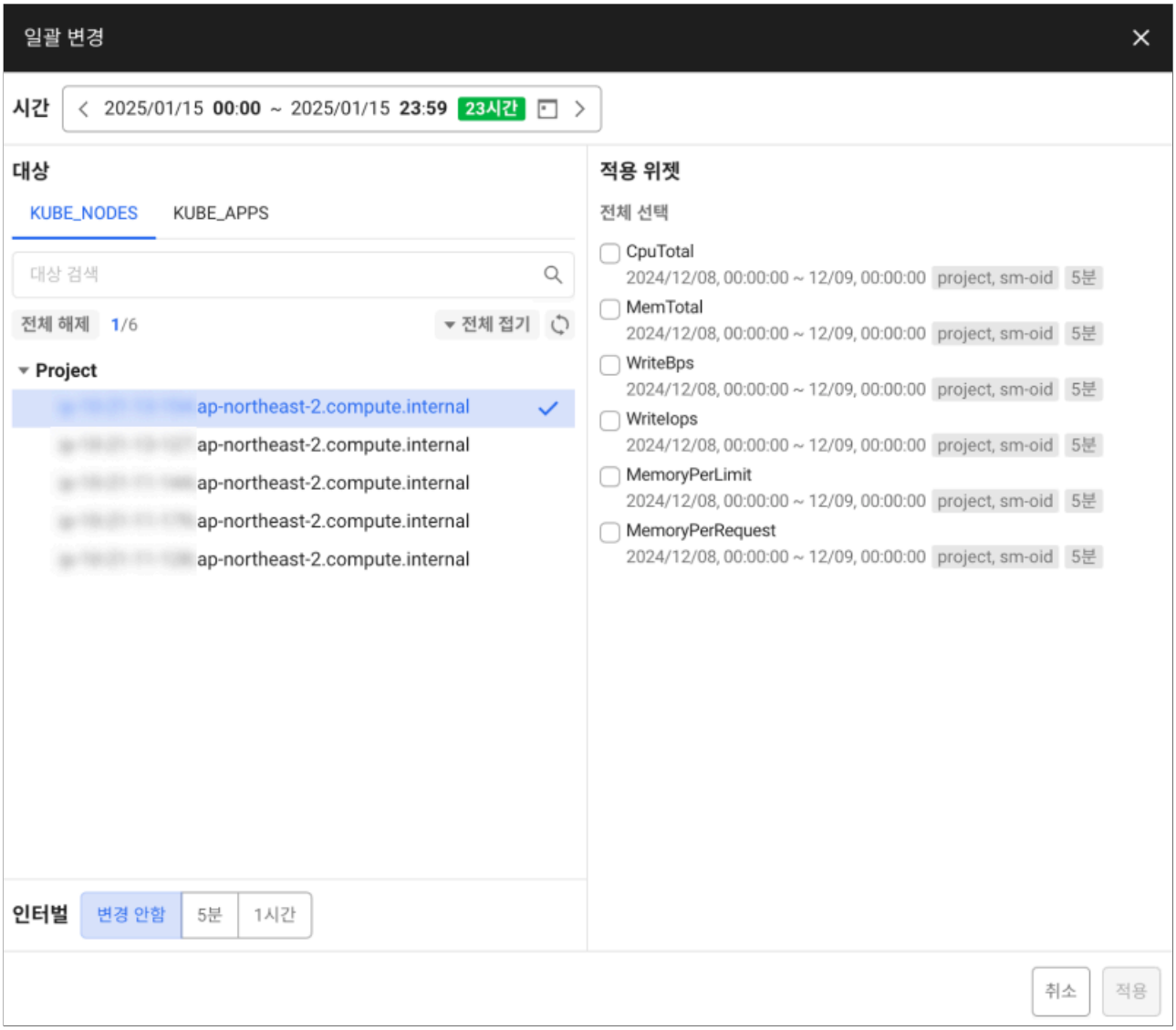
> ⓘ **레이아웃**이 **사용자 정의**로 설정되어 있지 않다면 위젯의 높이만 조절할 수 있습니다. 위젯 하단 가운데 영역을 마우스로 클릭한 상태로 크기를 조절하세요.

대시보드 옵션 이용하기

화면 상단에 위치한 옵션 메뉴를 이용해 차트 위젯에 적용된 다양한 설정을 한번에 적용할 수 있습니다. 또한 대시보드의 레이아웃을 원하는 형태로 설정할 수도 있습니다.

차트 위젯 옵션 일괄 변경하기

화면에 배치된 차트 위젯에 세부 옵션을 한번에 적용할 수 있는 기능을 제공합니다. 화면 왼쪽 상단에 **일괄 변경** 버튼을 선택하세요. **일괄 변경** 창이 나타나면 원하는 옵션을 설정한 다음 **적용** 버튼을 선택하세요.



- **시간**: 차트의 데이터 조회 범위를 설정할 수 있습니다.
- **대상**: 모니터링 대상을 선택하세요. 개별 에이전트를 선택하거나 개별 에이전트를 선택하거나 에이전트 설정을 통해 분류한 그룹 단위로 선택할 수 있습니다.
  - **에이전트**: 개별 에이전트를 선택하거나 모두 선택할 수 있습니다.
  - **종류별**: 에이전트 설정에서 `whatap.okind` 로 분류된 그룹 단위로 에이전트를 모니터링합니다.
  - **서버별**: 에이전트 설정에서 `whatap.onode` 로 분류된 그룹 단위로 에이전트를 모니터링합니다.
- **인터벌**: 다양한 시간 간격으로 데이터의 집계 단위를 조정할 수 있습니다.

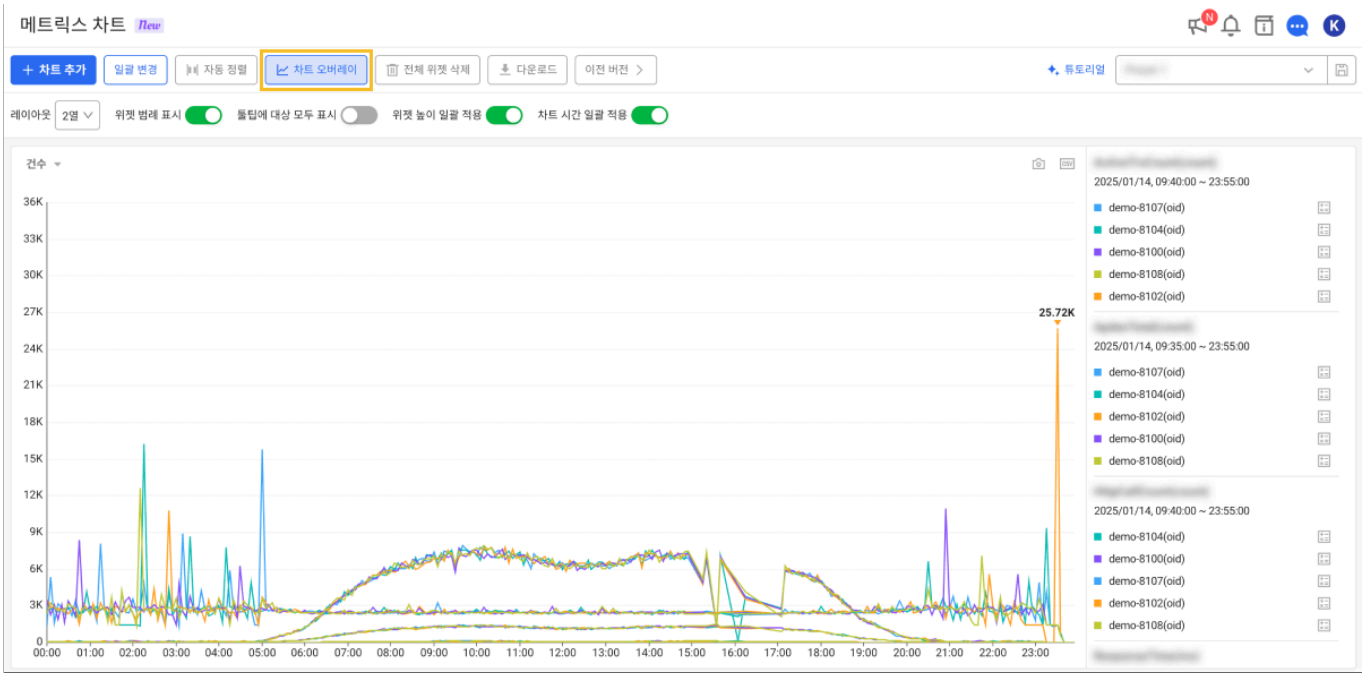
- **적용 위젯**: 변경할 옵션을 적용할 차트 위젯을 선택하세요. 선택한 위젯에 변경한 옵션을 한번에 적용할 수 있습니다.

대시보드 자동 정렬하기

대시보드에 배치한 위젯들의 위치를 자동 정렬하려면 **자동 정렬** 버튼을 선택하세요. 위젯의 너비, 높이가 작은 크기순으로 위젯의 배치를 자동 정렬합니다. 너비가 같으면 높이가 작은 순으로 정렬합니다.

여러 개의 차트 비교하기

대시보드에 배치한 여러 개의 차트 데이터를 하나의 차트에서 확인할 수 있는 기능입니다. 이 기능을 통해서 동시간대의 여러 지표 정보를 다른 지표와 비교해 가며 확인할 수 있습니다. **차트 오버레이** 버튼을 선택하세요. 화면에 오버레이 차트를 배치할 수 있습니다.



- 오버레이 차트 왼쪽 상단에 **건수**를 선택하면 x축 차트의 데이터 기준을 변경할 수 있는 목록이 나타납니다. 원하는 항목을 선택해 차트의 데이터를 비교할 수 있습니다.
- : 현재 조회 중인 데이터를 기준으로 차트를 이미지 형식의 파일로 다운로드할 수 있습니다.
- CSV : 현재 조회 중인 데이터를 기준으로 차트를 CSV 형식의 파일로 다운로드할 수 있습니다.
- 오른쪽 섹션에서는 지표별 에이전트 목록을 확인할 수 있습니다. 개별 에이전트의 색상은 차트의 그래프 색상과 일치합니다.

배치된 위젯 일괄 삭제하기

대시보드에 배치한 여러 개의 위젯을 한번에 삭제할 수 있습니다. **전체 위젯 삭제** 버튼을 선택하세요. 확인 메시지가 나타나면 **삭제** 버튼을 선택하세요.

CSV 파일로 다운로드하기

대시보드에 배치한 위젯의 데이터를 CSV 형식의 파일로 다운로드할 수 있습니다. **다운로드** > **CSV 다운로드**를 선택하세요.
시간대별 지표의 데이터를 확인할 수 있습니다.

이미지 파일로 다운로드하기

대시보드에 배치한 모든 위젯을 하나의 이미지 형식의 파일로 다운로드할 수 있습니다. **다운로드** > **이미지 다운로드**를 선택하세요.
대시보드 화면을 그대로 캡처한 이미지를 저장할 수 있습니다.

이전 버전 이용하기

이전 버전의 **메트릭스 차트** 메뉴를 이용하려면 **이전 버전** 버튼을 선택하세요. 또는 **사이트맵** > **분석** > **메트릭스 차트** *Old* 메뉴 경로에서 확인할 수 있습니다.

> **메트릭스 차트** *Old* 메뉴에 대한 자세한 내용은 다음 문서를 참조하세요.

기타 옵션 이용하기

그 밖의 다음 기타 옵션을 통해 대시보드의 레이아웃을 조정하거나 위젯의 적용된 옵션을 쉽고 간편하게 일괄 설정할 수 있습니다.

- **레이아웃**: 대시보드에서 열의 수, 즉 한 줄에 배치할 수 있는 위젯의 개수를 설정할 수 있습니다. 최대 4열까지 설정할 수 있습니다. **사용자 정의**를 선택하면 열의 수에 상관없이 위젯을 배치할 수 있습니다.

> ⓘ **사용자 정의**에서는 위젯의 너비와 높이를 자유롭게 조절할 수 있습니다. 다른 옵션을 선택하면 위젯의 높이만 조절할 수 있습니다.

- **위젯 범례 표시**: 옵션을 켜거나 끄면 위젯의 차트 오른쪽에 범례를 표시하거나 숨길 수 있습니다. 옵션을 끈 상태에서는 **범례**를 클릭해 모니터링 대상을 확인하고, 선택할 수 있습니다. 범례 활용에 대한 자세한 내용은 다음 문서를 참조하세요.
- **툴팁에 대상 모두 표시**: 차트에 마우스 포인터를 가져가면 해당 그래프의 위치에 해당하는 값을 툴팁을 통해 확인할 수 있습니다. 마우스 포인터가 위치한 시간대의 모든 대상의 데이터를 툴팁을 통해서 확인하려면 옵션을 켜세요.
- **위젯 높이 일괄 적용**: 옵션을 켜면 모든 위젯의 높이를 동일하게 설정할 수 있습니다.
- **차트 시간 일괄 적용**: 개별 차트 위젯의 데이터 조회 시간을 변경할 때, 다른 위젯의 차트 시간도 같이 변경하려면 옵션을 켜세요.

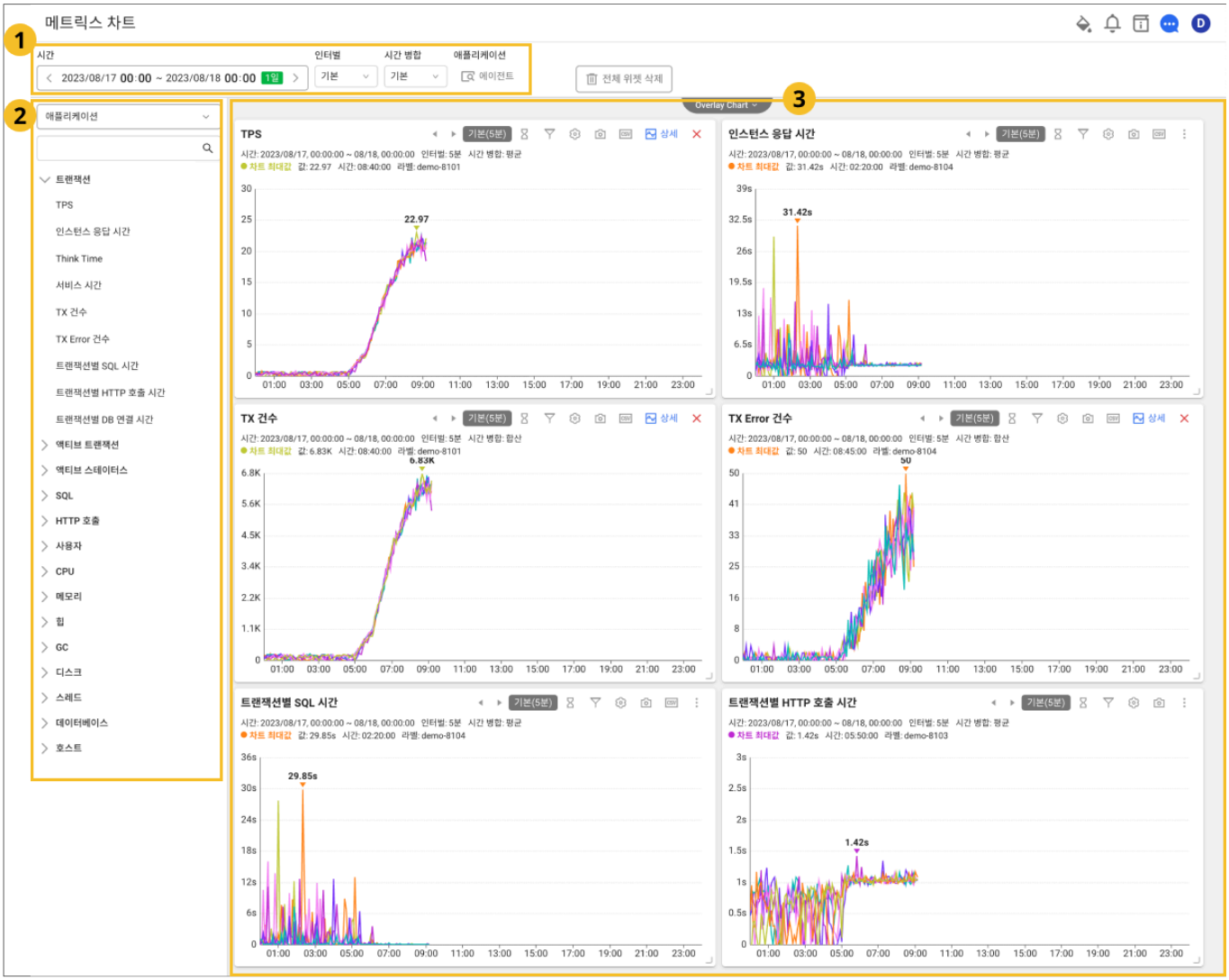
튜토리얼 이용하기

화면 오른쪽 상단에 **튜토리얼** 버튼을 선택하면 Service 2.8.0 업데이트 이후 변경 사항을 확인할 수 있습니다.

메트릭스 차트

사이트맵 > 분석 > 메트릭스 차트 *Old*

메트릭스 차트 메뉴에서 모니터링 대상에서 수집된 메트릭스 데이터를 다음과 같이 차트로 조회할 수 있습니다. 시간과 지표를 지정하는 것은 필수입니다.



상단 옵션

1 영역의 **메트릭스 차트**의 상단 옵션을 통해 차트의 시간 범위와 모니터링 대상 에이전트를 지정할 수 있습니다.

- **시간**: 시간의 총 범위로 X축의 시작과 끝을 지정할 수 있습니다.
- **인터벌**: 시간 간격으로 X축 데이터 간격을 지정할 수 있습니다.
- **시간 병합**: 데이터 병합 방식 중 하나로 인터벌로 지정한 시간 내의 데이터를 병합할 수 있습니다.

  예, **평균**은 1시간 안 데이터의 평균값을 말합니다.

- **에이전트**: 조회할 에이전트를 지정할 수 있습니다. 지정하지 않으면 전체가 조회됩니다.

지표 목록

2 영역은 옵션을 조회할 지표 목록입니다. 먼저 **카테고리**를 선택하세요. 선택한 **카테고리** 하위의 지표를 조회한 후 원하는 지표를 선택하세요. **카테고리**와 지표를 선택하면 1 영역의 상단 메뉴에서 지정한 시간 범위의 데이터를 바탕으로 3 영역에서 차트 위젯을 조회할 수 있습니다.

> **데이터 병합**
>
> 데이터 병합은 **시간 병합**과 **오브젝트 병합**을 제공합니다.
>
> - **시간 병합**은 원본 데이터에서 필드 값이 같은 데이터끼리 일정한 간격으로 데이터를 병합합니다.
> - **오브젝트 병합**은 서로 다른 필드 값을 가진 데이터들 중에서 태그가 일치하는 경우 해당 데이터를 병합합니다.

차트 위젯

3 영역 차트 위젯의 좌측 상단에서 지표명을 확인할 수 있습니다. 차트 위젯의 우측 상단에서 다음과 같은 옵션을 확인할 수 있습니다.

- 시간 이동: ◀ ▶ 왼쪽 화살표 또는 오른쪽 화살표 버튼을 통해 선택한 시간 범위만큼 -1, +1 씩 이동 가능합니다.

예, 시간 범위가 2월 13일 00:00~2월 14일 00:00일 때, ◀ 왼쪽 화살표를 선택하면 2월 12일 00:00~2월 13일 00:00 데이터를 조회할 수 있습니다.

- 인터벌/시간 병합: 1 상단 메뉴에서 지정한 인터벌과 시간 병합을 수정할 수 있습니다.
- 모니터링 대상: 아이콘을 선택해 모니터링 대상을 지정할 수 있습니다. 선택하지 않으면 전체를 대상으로 조회합니다.
- 시간 비교: 아이콘을 선택하면 동일한 지표의 이전 시간대의 추이를 비교할 수 있습니다.
- 스냅샷: 아이콘을 선택해 위젯의 옵션을 제외한 차트를 스냅샷 할 수 있습니다.
- CSV: 아이콘을 선택해 차트를 그리는 데이터를 CSV 파일로 다운로드할 수 있습니다.
- **상세**: 아이콘을 선택해 상세 조회가 가능합니다. 모니터링 대상이 많은 경우 유용하며, 모니터링 대상의 지표 추이를 개별로 확인 할 수 있습니다.

> 메트릭스 차트 위젯 상단에서 옵션이 보이지 않을 경우 ⋮ 아이콘을 선택하세요.

위젯 삭제하기

화면에 배치한 모든 위젯을 삭제하려면 화면 위에 위치한 **전체 위젯 삭제** 버튼을 선택하세요.

메트릭스 이상 탐지

홈 화면 > 프로젝트 선택 > **분석** > **메트릭스 이상 탐지**

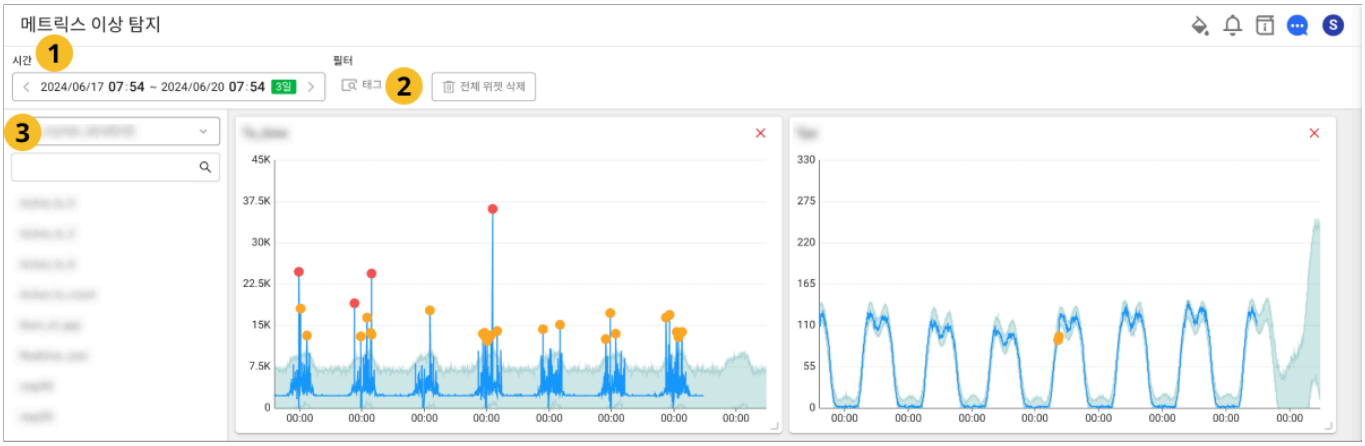
다양한 메트릭의 패턴을 AI가 학습한 예상 패턴과 비교해 볼 수 있습니다. 예상 패턴을 벗어난 이상 탐지를 그래프 차트를 통해 확인할 수 있습니다. 과거 데이터를 바탕으로 반복되는 패턴을 확인하고 향후 지표 값 예측에 활용할 수 있습니다.

- 화면에 배치한 위젯은 다른 메뉴로 이동할 경우 저장되지 않고 초기화합니다.
- 패턴 표시와 이상치 표시 기능을 제외하면 **분석** > **메트릭스 차트** 메뉴와 유사합니다.
- **이상치 탐지(Anomaly Detection)** 경고 알림 기능의 기술 근간은 **메트릭스 이상 탐지**입니다. **이상치 탐지** 경고 알림 설정에 대한 자세한 내용은 다음 문서를 참조하세요.

위젯 확인하기

- 1 밝은 색상의 그래프 영역은 AI가 분석한 예상 패턴입니다.
- 2 파랑색 그래프는 프로젝트의 메트릭 추이입니다.
- AI가 분석한 예상 패턴을 벗어나면 주황색, 빨간색의 단계로 그래프에 점을 표시합니다. 예상 패턴 범위를 크게 벗어난 값을 빨간색으로 표시합니다.

위젯 배치하기



1. 1 **시간**에서 원하는 시간 간격을 설정하세요. 최대 1개월 간격까지 설정할 수 있습니다.
2. 2 **필터**에서 메트릭의 범위를 선택하세요.
3. 아래 목록에서 모니터링하길 원하는 메트릭을 선택하세요.

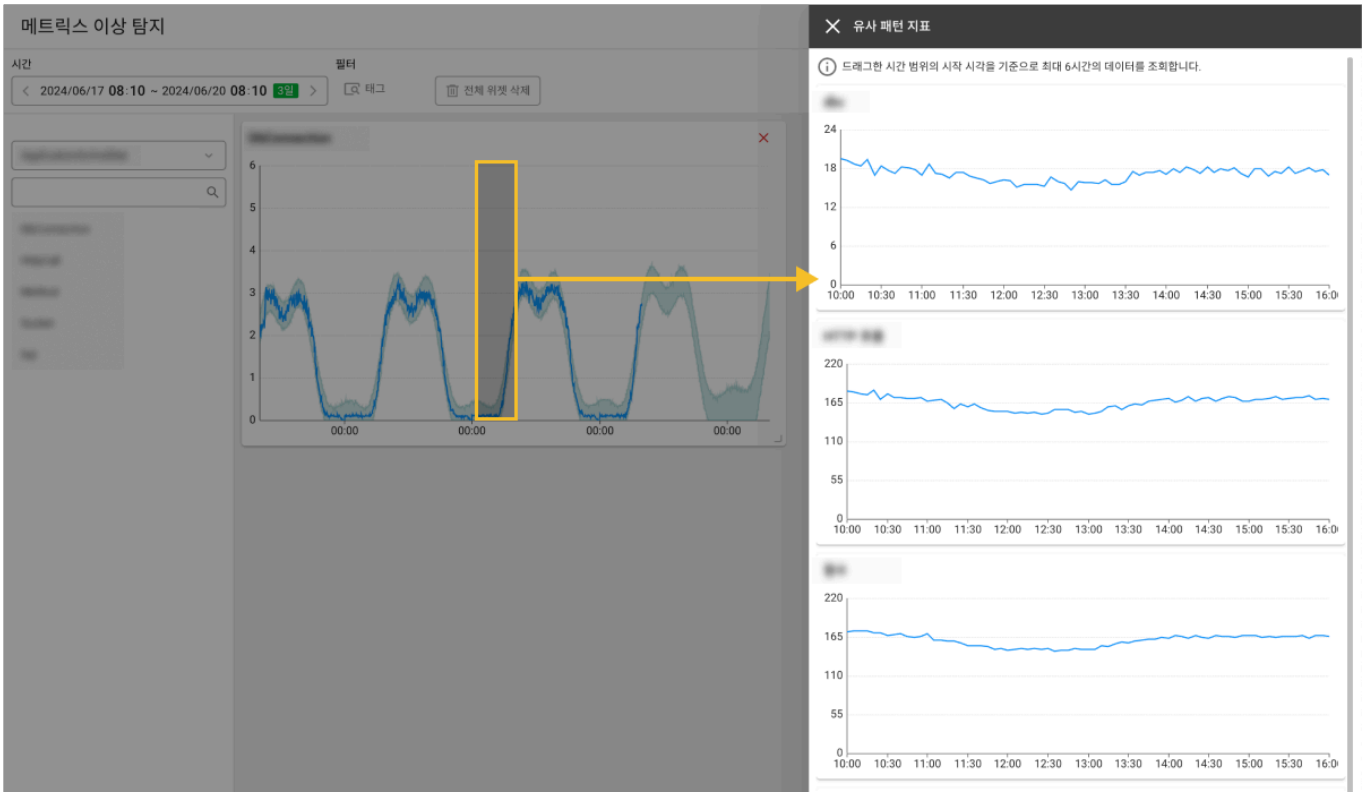
선택한 메트릭을 화면 오른쪽에 배치합니다.

- 모니터링 대상을 선택해 메트릭을 구분해서 확인하려면 2 **태그**를 선택하세요. 화면 오른쪽에 태그 선택 목록이 나타납니다. 원하는 항목을 선택한 다음 위젯을 추가하세요.
- 화면에 배치한 위젯은 **시간** 또는 **태그** 값을 변경해도 차트에 반영되지 않습니다.
- 화면에 배치한 위젯을 삭제하려면 위젯 오른쪽 위에 ✕ 버튼을 선택하세요.
- 위젯의 왼쪽 위를 선택한 상태에서 드래그해 위젯 위치를 변경할 수 있습니다.
- 위젯 오른쪽 아래를 선택한 상태에서 드래그해 위젯 크기를 조절할 수 있습니다.

위젯 삭제하기

화면에 배치한 모든 위젯을 삭제하려면 화면 위에 위치한 **전체 위젯 삭제** 버튼을 선택하세요.

연관 지표 확인하기



위젯에서 차트의 일부 영역을 드래그하세요. 화면 오른쪽에서 **유사 패턴 지표** 창이 나타납니다. 드래그한 시간 범위의 시작 시간을 기준으로 최대 6시간의 연관 지표를 조회할 수 있습니다.

메트릭스 조회

홈 화면 > 프로젝트 선택 > **분석** > **메트릭스 조회**

메트릭스 조회 메뉴에서 태그 기반으로 특정 메트릭스를 조회할 수 있습니다.

시간과 카테고리 선택

메트릭스가 수집된 **시간 선택**과 **최대 개수** 및 **카테고리**를 지정할 수 있습니다. **시간 선택**과 **카테고리**는 반드시 지정해야 합니다.

- **시간 선택**: 메트릭스가 수집된 시간을 지정해 조회할 수 있습니다. 기본값은 `1시간` 입니다. 기본 옵션으로 제공하는 조회 시간 외 사용자가 직접 **시간 선택** 탭을 선택해 날짜와 시간을 지정할 수 있습니다.
- **최대 개수**: **메트릭스 테이블** 목록에 조회할 메트릭스 최대 개수를 지정할 수 있습니다. `10` , `50` , `100` , `200` , `300` , `1000` , `2000` , `3000` 개까지 설정할 수 있습니다.
- **카테고리**: 유관 지표들의 분류 단위입니다. **카테고리** 탭을 선택해 원하는 **카테고리**를 지정할 수 있습니다.
- **카테고리, 태그, 필드 옵션 다시 불러오기**: **카테고리**, **태그** 및 **필드** 옵션을 다시 불러올 수 있습니다.

태그와 필드 선택

태그와 **필드**를 선택합니다. 사용자가 개별적으로 지정하지 않는다면 기본 설정은 전체 선택입니다.

- **태그**: 수집된 대상을 구분할 수 있는 고유 정보 데이터입니다.
- **필드**: 모니터링 대상으로부터 수집된 지표입니다.
- **필터**: **태그값으로 필터링** 버튼을 선택하고 태그값을 설정해 필터링할 수 있습니다.

  예시, `oname` 의 값을 `demo-8101` 로 설정해 필터링한 데이터를 조회할 수 있습니다.
- **검색**: 조건을 설정 후 **검색** 아이콘을 선택하면 **메트릭스 테이블** 영역에서 해당 메트릭스의 원본 데이터를 조회할 수 있습니다.
- CSV : 해당 메트릭스 원본 데이터를 CSV 파일로 다운로드할 수 있습니다.

메트릭스 테이블

수집되는 메트릭스를 사전에 특정할 수 없기에 수집 중인 모든 메트릭스의 원본 데이터를 확인하는 것이 중요합니다. 위의 조건 영역에서 원하는 조건을 설정 후 **메트릭스 테이블** 영역에서 해당 메트릭스의 원본 데이터를 테이블 형식으로 조회할 수 있습니다. 사용자가 **태그**와 **필드** 각 조건을 지정함에 따라 테이블의 컬럼이 변경됩니다.

- 메트릭스 조회 시 **시간 선택**과 **카테고리**는 반드시 지정해야 합니다.
- 메트릭스 조회 시 **태그**와 **필드** 지정은 선택 사항입니다.

로그

쿠버네티스 관련 다양한 로그를 확인할 수 있습니다. 와탭 쿠버네티스에서 제공하는 모니터링 카테고리는 다음과 같습니다.

카테고리	설명
`#K8sEvent`	• 쿠버네티스에서 발생하는 이벤트가 저장된 로그 • 사용자 설정과 무관하게 기본 생성
`#WhatapEvent`	• 와탭 이벤트 설정에 의해 발생한 이벤트가 저장된 로그 • 사용자 설정과 무관하게 기본 생성
`containerStdout`	• 컨테이너 Standard Out 로그 • 사용자 설정 시 생성 • 노드 에이전트에 `logsink_enabled=true` 설정 추가 시
`AppLog`	• 컨테이너 내 애플리케이션 로그 • 사용자 설정 시 생성 • 애플리케이션 에이전트에 `logsink_enabled=true` 설정 추가 시

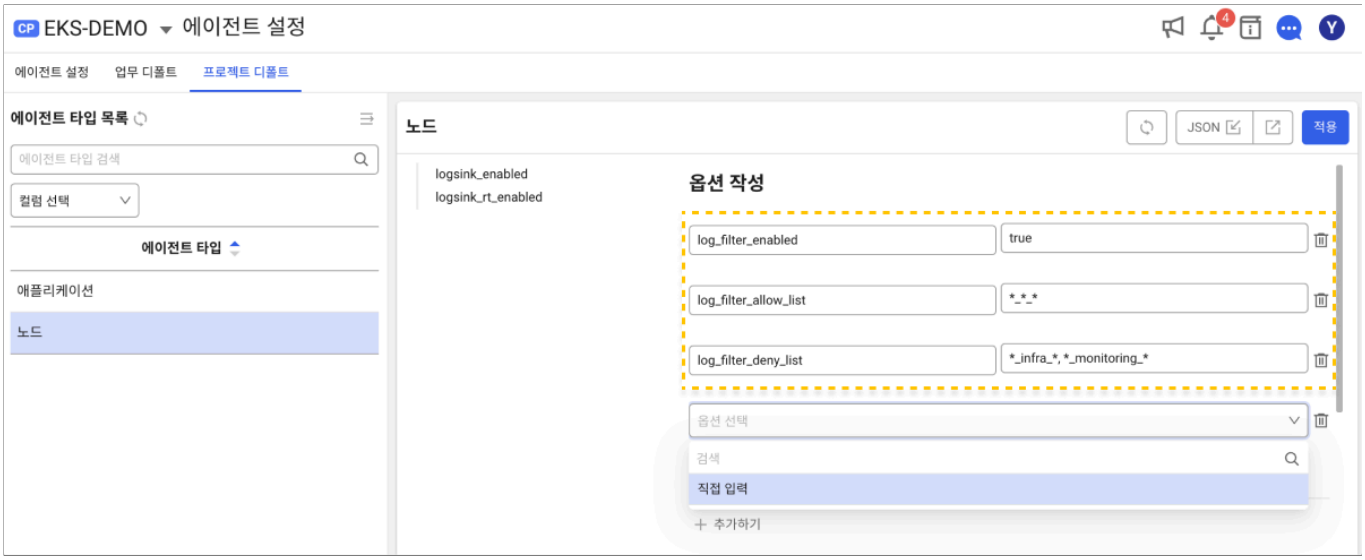
ⓘ 설정에 따라 중복된 로그 내용이 저장될 수 있으니 반드시 중복 여부를 확인하세요.

containerStdOut 로그 필터 설정

홈 화면 > 프로젝트 선택 > **관리** > **에이전트 설정**

쿠버네티스 로그에서 필요한 정보를 선별하기 위한 필터링 옵션을 안내합니다. 필터링을 통해 허용하거나 제외할 로그를 설정할 수 있습니다.

1. **관리** > **에이전트 설정**으로 이동하세요.
2. **프로젝트 디폴트** 탭에서 **옵션 작성** 방식으로 **직접 입력**을 클릭하세요.



로그 네이밍 규칙

쿠버네티스 containerStdOut 로그는 노드의 /var/log/containers/ 경로 하위에 생성됩니다. 다음과 같은 네이밍 규칙을 따르고 있습니다.

```
<podName>_<namespace>_<containerName>-<containerId>.log
```

로그 필터 옵션

- **log_filter_enabled** bool

  기본값 `false`

  로그 필터 사용 여부를 설정합니다.

- **log_filter_allow_list** list

  허용할 로그의 리스트를 설정합니다. 필터링 적용 시 포함할 항목을 의미합니다.

- **log_filter_deny_list** list

  제외할 로그의 리스트를 설정합니다. 필터링 적용 시 제외할 항목을 의미합니다.

로그 필터 동작

`log_filter_enabled` 옵션이 활성화된 경우만 로그 필터가 동작합니다. 해당 옵션의 값이 `true`로 설정된 상태에서 `log_filter_allow_list` 및 `log_filter_deny_list`를 통해 로그를 선택적으로 수집할 수 있습니다.

로그 필터는 블랙리스트 기반으로 작동합니다. 예를 들어 `log_filter_enabled` 활성화 후 `log_filter_allow_list` 옵션에 추가할 항목을 설정하지 않을 경우 기본적으로 모든 컨테이너의 로그 수집이 차단됩니다. 또한 로그 필터는 allow 규칙 보다 deny 규칙의 우선 순위가 높습니다. 동일한 항목이 allow와 deny에 모두 설정된 경우, deny 규칙이 우선 적용되어 해당 로그는 수집되지 않습니다.

로그 필터 예시

단일 네임스페이스 로그 수집

- 허용 **log_filter_allow_list** `*_infra_*`

`infra` 네임스페이스에 존재하는 모든 컨테이너가 수집 대상으로 특정됩니다.

다중 네임스페이스 로그 수집

- 허용 **log_filter_allow_list** `*_infra_*, *_monitoring_*`

`infra`, `monitoring` 네임스페이스에 존재하는 모든 컨테이너가 수집 대상으로 특정됩니다.

특정 단어를 포함하는 Pod 제외

- 허용 **log_filter_allow_list** `*_*_*`
- 제외 **log_filter_deny_list** `*prod*_*_*`

모든 컨테이너가 수집 대상으로 특정되지만, `podName`에 `prod`가 포함된 로그는 제외됩니다.

특정 단어를 포함하는 Pod 로그 수집

- 허용 **log_filter_allow_list** `*prod*_*_*`

`podName`에 `prod` 가 포함된 파드의 로그만 수집됩니다.

다중 네임스페이스 로그 수집 및 특정 단어 포함하는 Pod 제외

- 허용 **log_filter_allow_list** `*_infra_*, *_monitoring_*`
- 제외 **log_filter_deny_list** `*prod*_*_*`

`infra`, `monitoring` 네임스페이스의 로그가 수집되고, `podName`에 `prod`가 포함된 로그는 제외됩니다.

컨테이너 이름 내 특정 단어로 시작하는 컨테이너 로그 제외

- 허용 **log_filter_allow_list** `*_*_db-*`

모든 네임스페이스에서 `containerName`이 `db`로 시작하는 컨테이너의 로그는 제외됩니다.

라이브 테일

홈 화면 > 프로젝트 선택 > **로그** > **라이브 테일**

와탭 쿠버네티스 모니터링 카테고리의 내용을 실시간으로 확인할 수 있습니다. **라이브 테일**에 대한 자세한 내용은 다음 문서를 참조하세요.

로그 트렌드

홈 화면 > 프로젝트 선택 > **로그** > **로그 트렌드**

시간 범위를 설정하여 저장된 로그 추이를 확인할 수 있습니다. **로그 트렌드**에 대한 자세한 내용은 다음 문서를 참조하세요.

로그 검색

홈 화면 > 프로젝트 선택 > **로그** > **로그 검색**

시간 범위를 설정하여 저장된 로그의 내용을 검색할 수 있습니다. 예를 들어, 특정 컨테이너의 Standard Out 로그를 확인하거나, Deployment에 속하는 모든 컨테이너의 Standard Out 로그를 한번에 확인할 수 있습니다. 특정 키워드를 하이라이트하여 표시할 수도 있습니다.

로그 트렌드에 대한 자세한 내용은 다음 문서를 참조하세요.

로그 탐색

홈 화면 > 프로젝트 선택 > **로그** > **로그 탐색**

기능은 로그 데이터를 빠르게 검색, 필터링, 분석할 수 있습니다. **로그 탐색**에 대한 자세한 내용은 다음 문서를 참조하세요.

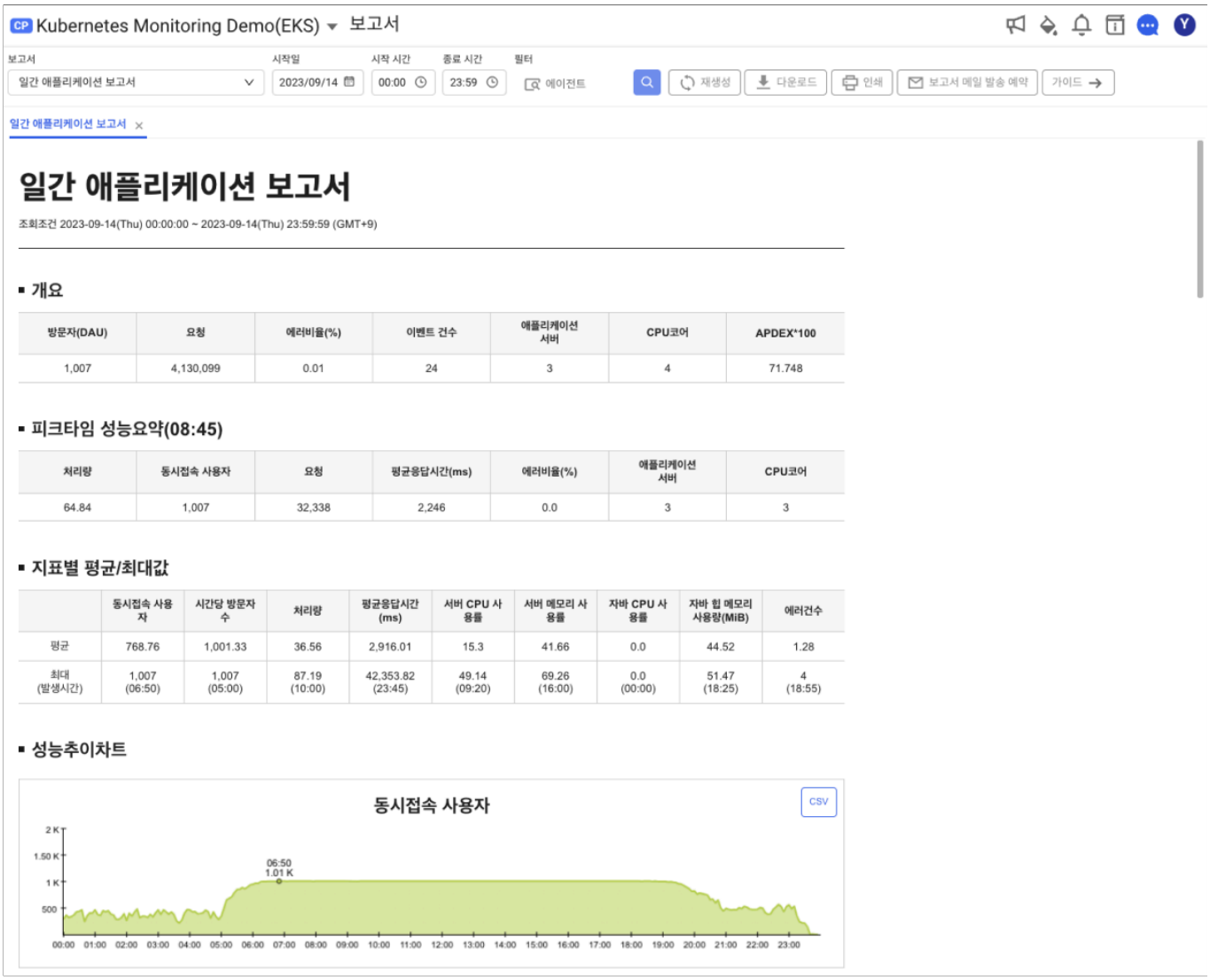
로그 설정하기

홈 화면 > 프로젝트 선택 > **로그** > **설정**

로그 활성화를 포함한 각종 로그 관련 설정을 할 수 있습니다. 로그 설정에 대한 자세한 내용은 다음 문서를 참조하세요.

보고서

화면 위에 **보고서**에서 ⌄ 버튼을 클릭하면 일간, 주간, 월간으로 분류된 각종 분석 보고서를 확인할 수 있습니다. 원하는 보고서 양식을 선택한 다음 조회하길 원하는 날짜와 시간, 필터 옵션을 설정하세요. 🔍 버튼을 선택하면 선택한 보고서를 확인할 수 있습니다.



보고서는 개별 프로젝트에 대해 서비스 이용 현황과 장애 발생 기록을 보고하는 문서입니다. 서비스 모니터링 담당자는 보고서를 통해 관련 부서 담당자들과 현황을 공유합니다. 모니터링 데이터 분석은 서비스의 개선 방향을 정하는 지표가 되기 때문에

중요합니다. 하지만 여러 대시보드의 데이터를 취합해서 문서화하는 일은 번거롭습니다.

와탭의 **보고서** 메뉴는 **보고서 작성 업무 자동화**를 지원합니다. 매주 보고서를 작성해야 하는 일, 정해진 시간에 보고서를 공유하는 일, 여러 가지 서식을 관리하는 일 모두 **보고서** 메뉴에서 할 수 있습니다.

- 보고서를 다운로드하려면 **다운로드** 버튼을 선택하세요. html 파일 형식으로 다운로드 받을 수 있습니다.
- 보고서를 인쇄하려면 **인쇄** 버튼을 선택하세요.
- 보고서를 예약 발송하려면 **보고서 메일 발송 예약** 버튼을 선택하세요.

보고서와 통합 보고서의 차이

보고서는 하나의 프로젝트에 대한 보고서를 생성하는 메뉴입니다. 따라서 프로젝트를 선택 후 왼쪽에 보고서 메뉴를 통해 접근이 가능합니다. 통합 보고서는 여러 프로젝트에 대한 보고서를 생성하는 메뉴입니다. 프로젝트를 선택하지 않고 홈 화면에서 기본 메뉴를 통해 접근이 가능합니다. 통합 보고서에 대한 자세한 내용은 다음 문서를 참조하세요.

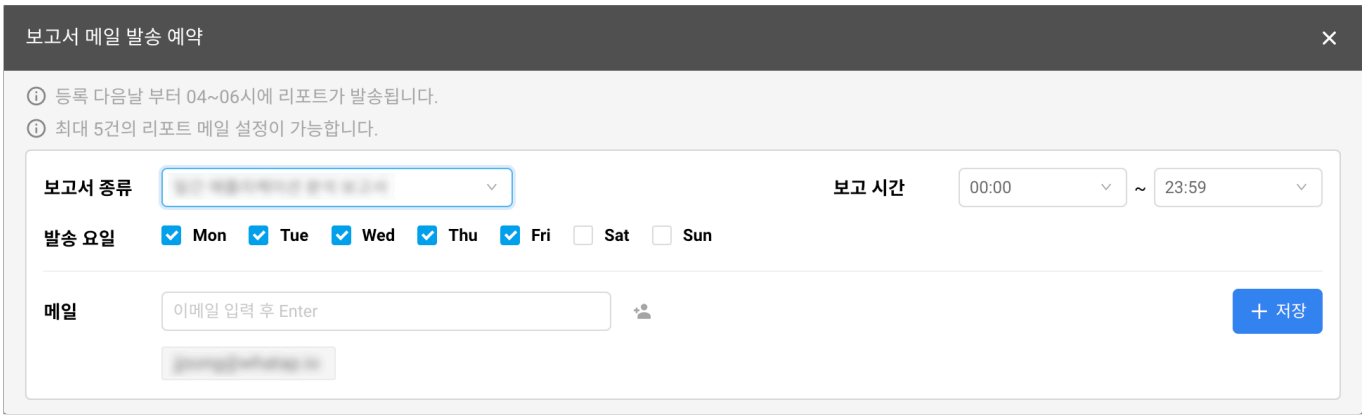
보고서 종류

쿠버네티스 관련 다양한 보고서를 확인할 수 있습니다. 와탭 쿠버네티스에서 제공하는 보고서 유형은 다음과 같습니다.

- **일간 애플리케이션 보고서**
- **일간 애플리케이션 종류별 임계치 초과 트랜잭션 리스트**
- **일간 메트릭 추이현황**
- **일간 애플리케이션 에러통계 보고서**
- **주간 애플리케이션 보고서**
- **월간 애플리케이션 보고서**
- **POD별 일간 Cpu/Memory 보고서**
- **POD별 주간 Cpu/Memory 보고서**
- **일간 서버 보고서**
- **주간 서버 보고서**
- **월간 서버 보고서**

메일 발송 예약

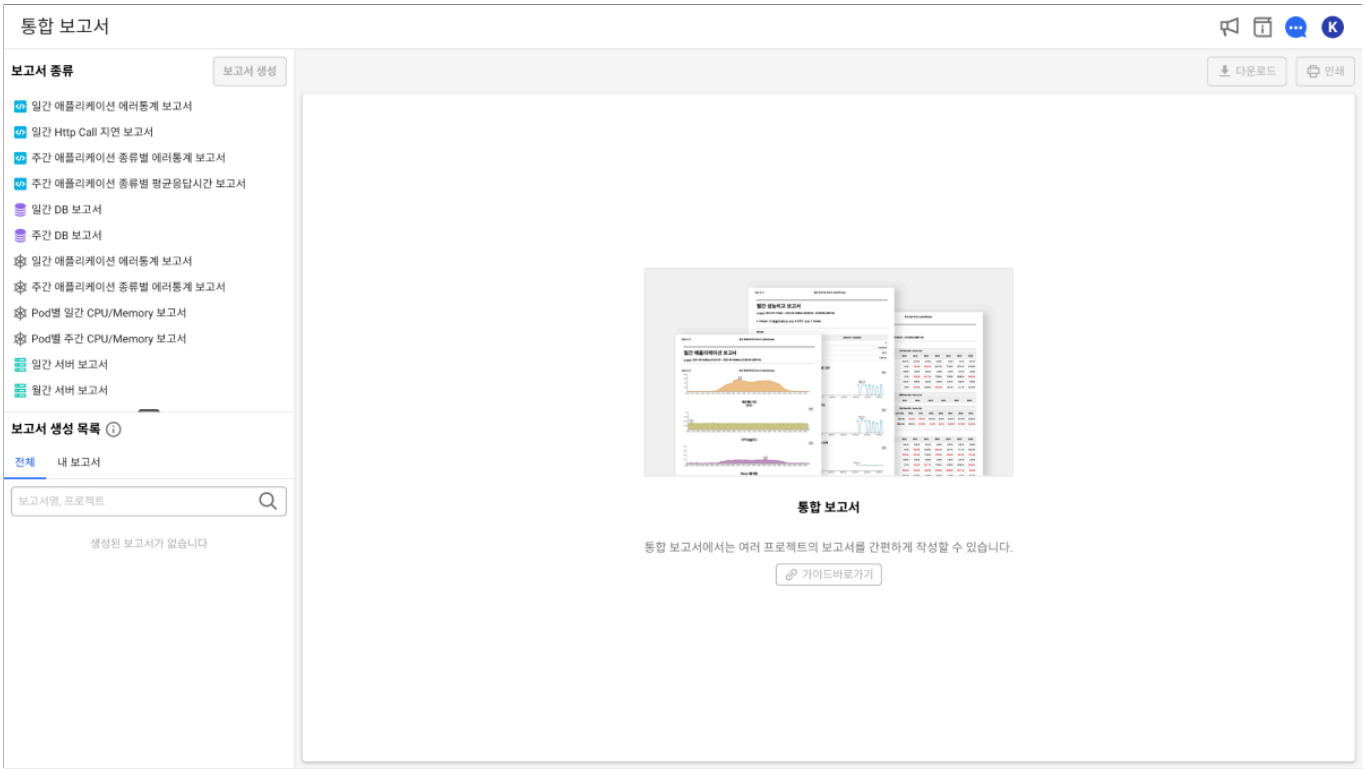
정기적으로 보고서를 email로 받고 싶으시면 **보고서 메일 발송 예약**을 선택하세요. 출근 직후 수행하던 여러 가지 서비스 점검 절차를 메일 확인으로 대체할 수 있습니다.



통합 보고서

홈 화면 > **통합 보고서**

사용자는 여러 프로젝트를 통합한 보고서를 생성하고 공유할 수 있습니다. 보고서 종류와 포함할 프로젝트를 선택해 보고서를 생성할 수 있으며, 생성 목록에서 작업 진행 상태를 확인할 수 있습니다. 특정 프로젝트에서 에러가 발생하면 에러 내용을 확인할 수 있습니다. 생성이 완료된 보고서는 즉시 조회하거나, 다운로드 및 인쇄 기능으로 공유할 수 있습니다.



> **통합 보고서** 메뉴는 **Application**, **Server**, **Kubernetes**, **Database** 제품의 프로젝트에 대한 보고서만 생성할 수 있습니다.

주요 기능 안내

통합 보고서 메뉴에서 제공하는 주요 기능은 다음과 같습니다.

- **멀티 프로젝트 지원**

  여러 프로젝트를 선택해 한 번에 보고서를 생성할 수 있어 편리합니다. 프로젝트에서 이용할 수 있는 **보고서** 메뉴는 해당 프로젝트에 대한 보고서만 생성할 수 있습니다.

- **업무 단위 보고서 생성**

  보고서 생성 기능은 업무 단위로 프로젝트를 선택해 템플릿으로 저장할 수 있습니다. 보통 업무 단위로 프로젝트를 나누어 사용하는 마이크로 서비스 아키텍처(Micro Service Architecture, MSA) 환경에서 유용한 기능입니다.

- **대용량 데이터 최적화**

  보고서 생성 시간이 획기적으로 줄어들었습니다. 그리고 일부 프로젝트 보고서 작성 과정에서 오류가 발생하더라도 보고서 결과를 조회할 수 있습니다.

- **보고서 생성 중 동시 작업 가능**

  보고서를 생성하는 동시에 다른 업무를 진행할 수 있습니다. 프로젝트에서 이용할 수 있는 **보고서** 메뉴는 보고서 생성 작업을 완료할 때까지 기다려야 합니다.

  통합 보고서 메뉴에서 보고서 생성을 시작한 후 다른 페이지로 이동할 수 있습니다. 또한 보고서 결과 목록에서 진행 상태와 완료된 보고서를 조회할 수 있습니다.

- **보고서 작업 공유**

  동일 프로젝트 권한을 가진 사용자라면 누구든 생성한 보고서를 조회할 수 있습니다. 관련 보고서가 이미 생성됐는지 확인할 수 있어 중복으로 작성하지 않을 수 있습니다.

보고서와 통합 보고서의 차이

프로젝트에서 이용할 수 있는 **보고서** 메뉴에서는 해당 프로젝트에 대한 보고서를 생성할 수 있습니다. 초기 화면에서 프로젝트를 선택 후 사이드 메뉴의 **보고서** 또는 **통계/보고서** 메뉴를 통해 접근할 수 있습니다.

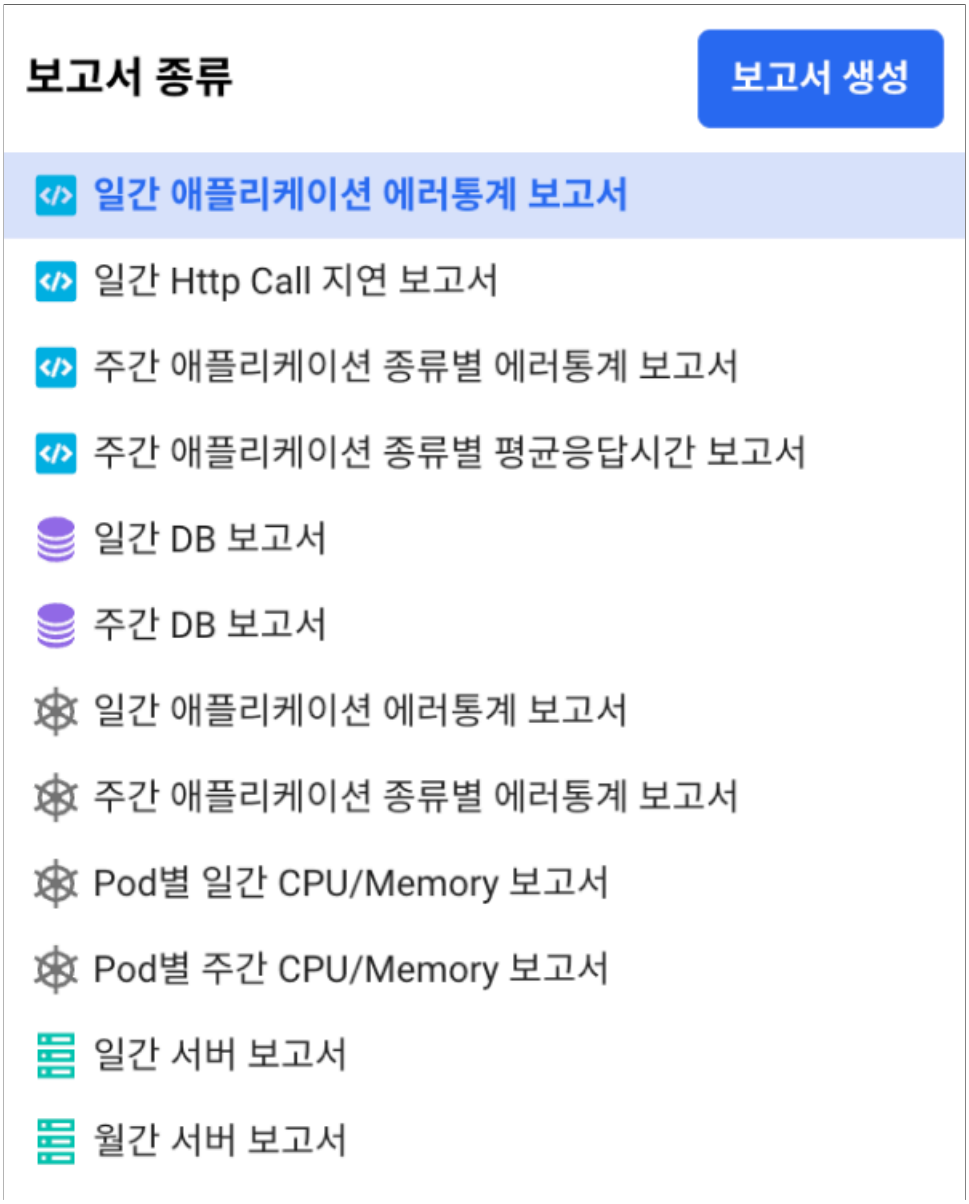
통합 보고서 메뉴에서는 여러 프로젝트에 대한 보고서를 생성할 수 있습니다. 프로젝트를 선택하지 않고 초기 화면에서 접근할 수 있습니다.

보고서 생성하기

보고서 종류 선택하기

보고서 종류 목록에서 원하는 보고서 유형을 선택하면 **보고서 생성** 버튼이 활성화됩니다. 보고서 이름 앞의 **아이콘**은 보고서 생성을 지원하는 프로젝트 플랫폼을 의미합니다.

- **Application** 프로젝트 플랫폼을 나타냅니다.
- **Database** 프로젝트 플랫폼을 나타냅니다.
- **Kubernetes** 프로젝트 플랫폼을 나타냅니다.
- **Server** 프로젝트 플랫폼을 나타냅니다.



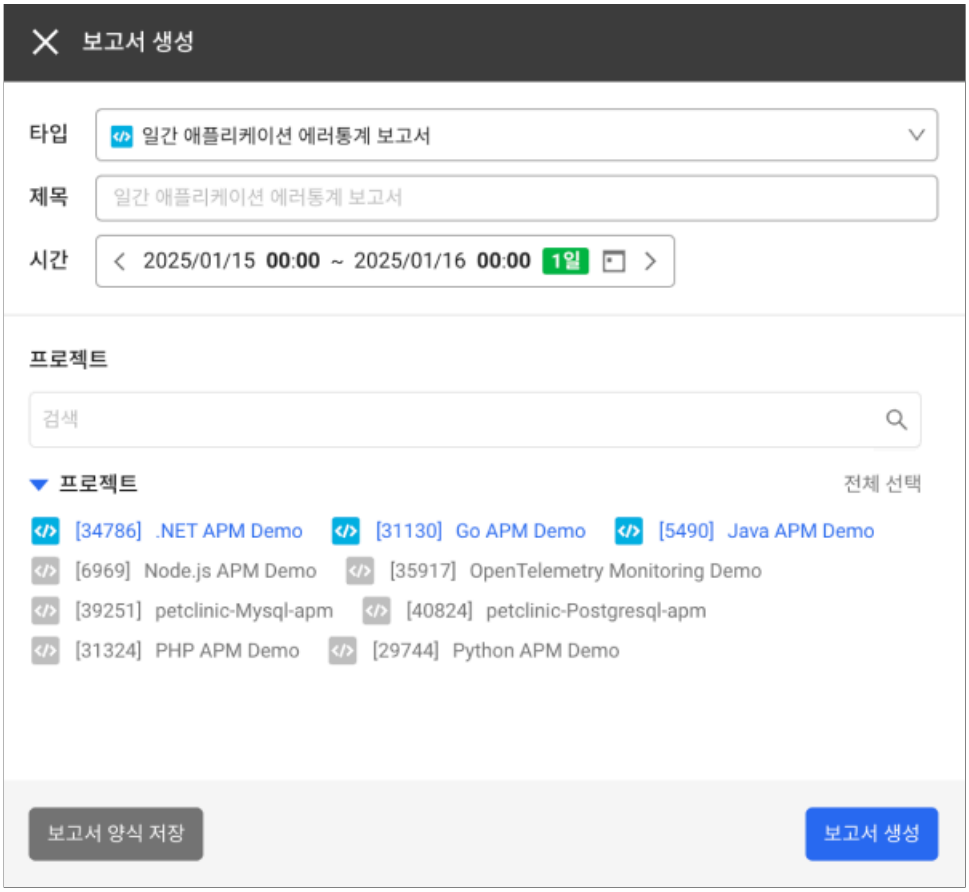
상세 설정하기

보고서 생성 창이 나타나면 각 항목을 설정하세요.

- **제목**: 같은 타입의 보고서가 여러 개인 경우 이름을 지정하면 보고서 결과를 구분하기 쉽습니다.
- **시간**: 보고서에 사용될 데이터의 기간을 선택하세요. 기간은 보고서 타입(일, 주, 월)에 따라 달라집니다.

- **프로젝트**: 보고서를 작성할 프로젝트를 선택하세요. 그룹 및 개별 프로젝트 단위로 선택할 수 있습니다. **검색** 필드를 통해 원하는 프로젝트를 검색할 수 있습니다.
- **보고서 양식 저장**: 현재 입력한 보고서의 설정을 템플릿으로 저장할 수 있습니다. 이후 새로운 보고서 생성 시 **타입** 목록에서 불러올 수 있습니다.

설정을 완료했다면 **보고서 생성** 버튼을 선택하세요.



선택한 보고서의 양식에 따라 조회 기간을 설정하는 방식은 다음과 같습니다.

- **일간 보고서**: 시작과 종료 날짜, 시간을 선택할 수 있습니다.
- **주간 보고서**: 시작 날짜만 선택하면 종료 날짜를 계산해 자동 선택합니다.
- **월간 보고서**: 시작 날짜만 선택하면 종료 날짜를 계산해 자동 선택합니다.

- 시작 날짜가 1일이면 해당 월의 월말까지 자동으로 선택합니다.
- 시작 날짜가 1일이 아니면 다음 달의 전일까지 자동으로 선택합니다.

예시 1, 시작날짜가 **1월 1일**이면, **1월 1일 00시 00분 00초**부터 **1월 31일 23시 59분 59초**를 자동 선택합니다.

예시 2, 시작날짜가 **1월 10일**이면, **1월 10일 00시 00분 00초**부터 **2월 9일 23시 59분 59초**를 자동 선택합니다.

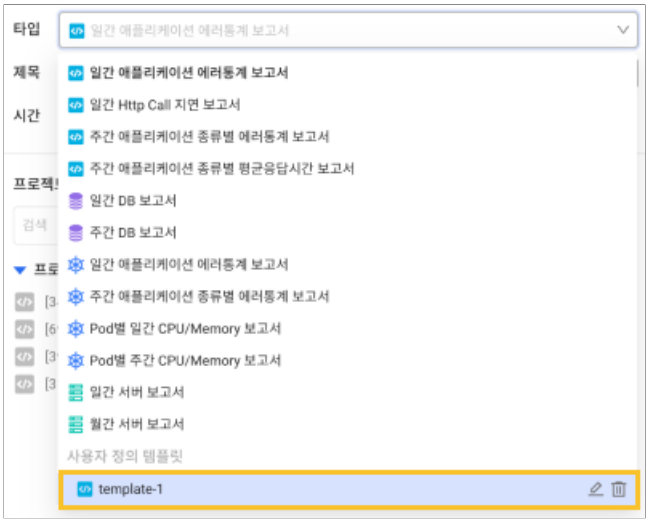
- **시간** 옵션 이용 방법에 대한 자세한 내용은 다음 문서를 참조하세요.
- 저장한 보고서 양식을 불러오는 방법에 대한 자세한 내용은 다음 문서를 참조하세요.

보고서 양식 불러오기

새로운 보고서를 생성할 때 **보고서 양식 저장** 기능을 사용해 저장한 양식은 이후 보고서 생성 시 **타입** 항목에서 확인할 수 있습니다.

업무 단위로 프로젝트를 자주 사용하는 사용자는 유용하게 활용할 수 있습니다.

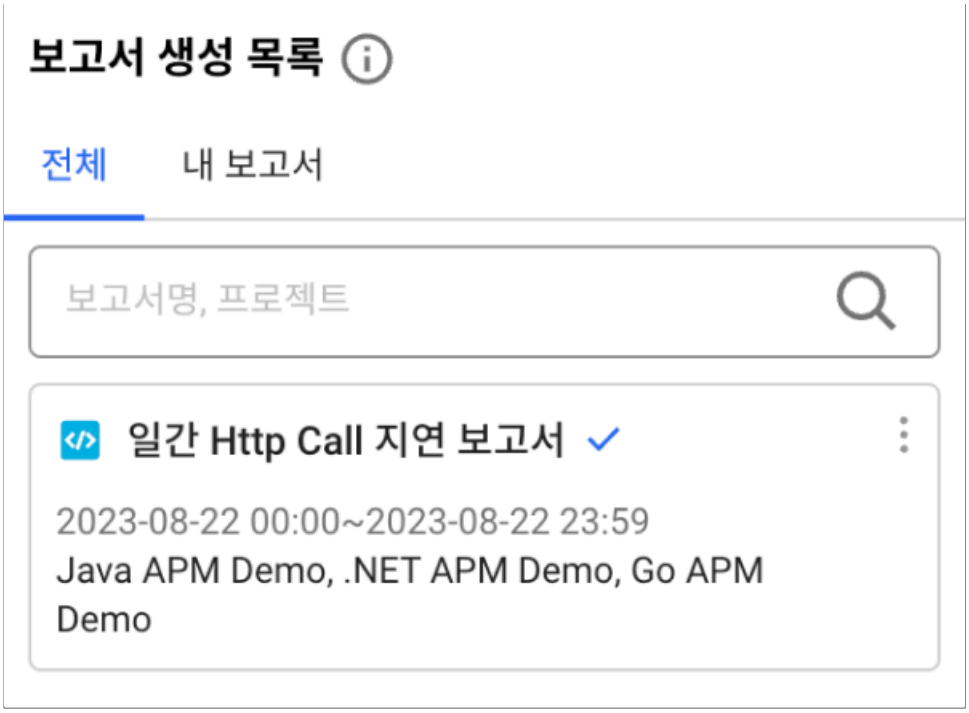
- : 저장한 보고서 양식을 수정할 수 있습니다.
- : 저장한 보고서 양식을 삭제할 수 있습니다.



보고서 목록 확인하기

보고서 생성 목록에서 이전에 만들었던 보고서와 현재 작업 중인 보고서를 확인할 수 있습니다.

- **전체** 탭에서는 사용자 본인이 생성한 보고서 외에도 같은 권한을 가진 사용자가 생성한 보고서를 함께 확인할 수 있습니다.
- **내 보고서** 탭에서는 사용자 본인이 생성한 보고서만 확인할 수 있습니다.
- 보고서 항목의 오른쪽 상단에 ⋮ 버튼을 선택하면, 해당 보고서를 복사하거나 삭제할 수 있는 기능을 이용할 수 있습니다.



- 에러 없이 생성된 보고서는 ✓ 아이콘으로 표시되며, 에러가 포함된 경우는 ⚠ 아이콘이 함께 표시됩니다. ⚠ 아이콘을 선택하면 어떤 프로젝트에서 에러가 발생했는지 확인할 수 있습니다.
- 생성한 보고서는 최대 100개까지 7일간 보관합니다.

조회 및 공유하기

- **화면 조회**

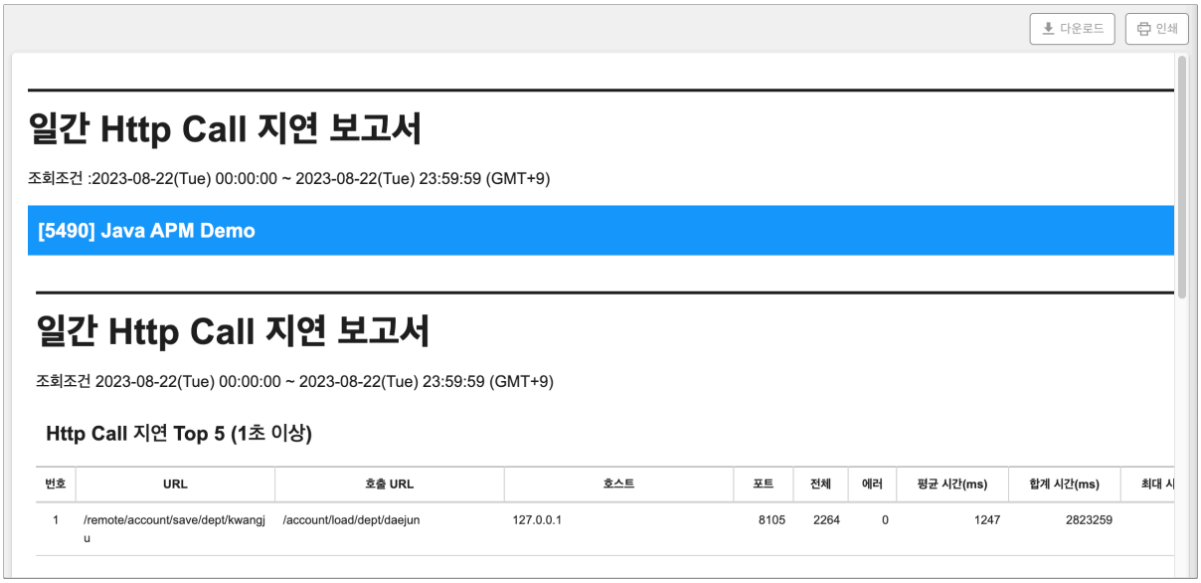
  보고서 생성 목록 목록에서 생성한 보고서를 선택하면 보고서의 상세 내용을 확인할 수 있습니다.

- **다운로드**

  현재 조회 중인 보고서를 html 형식의 파일로 다운로드하려면 **다운로드** 버튼을 선택하세요. 모니터링 관리 권한이 없는 사람에게 보고서를 공유할 수 있습니다.

- **인쇄 및 PDF 저장**

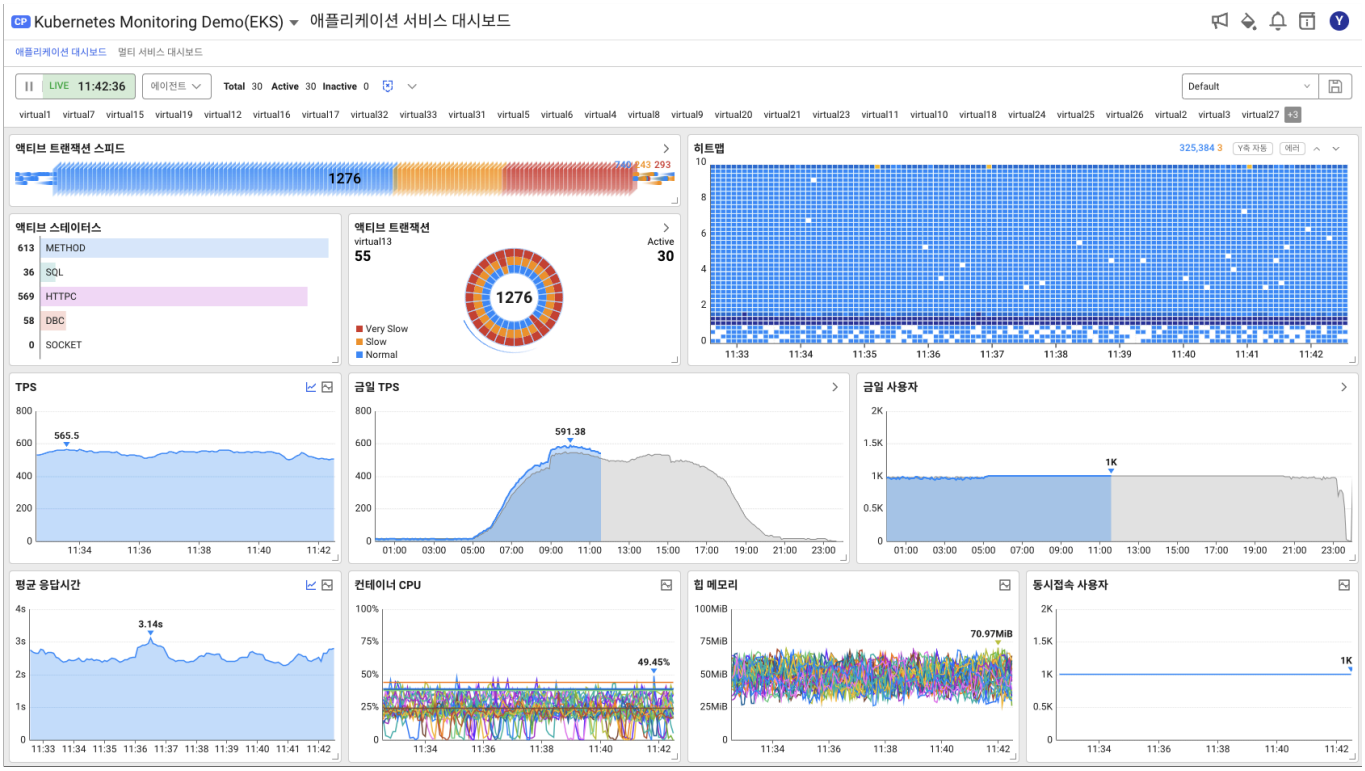
  보고서를 프린터를 통해 인쇄하거나 PDF 형식의 파일로 저장하려면 **인쇄** 버튼을 선택하세요.



애플리케이션 서비스 대시보드

컨테이너 내부에서 실행 중인 애플리케이션 상태를 실시간으로 확인할 수 있는 대시보드입니다. 각 위젯의 정상적인 데이터 출력을 위해서는 쿠버네티스 환경에 노드 에이전트를 설치해야 하고, 컨테이너 내부의 애플리케이션에도 와탭 에이전트를 설치해야 합니다.

- Java, Node.js, Python, PHP로 개발한 애플리케이션에 와탭 에이전트 설치를 지원합니다.
- **대시보드** 메뉴 또는 **애플리케이션** 메뉴 하위의 **대시보드** 메뉴를 통해 접근할 수 있습니다.



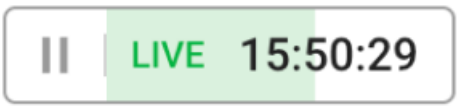
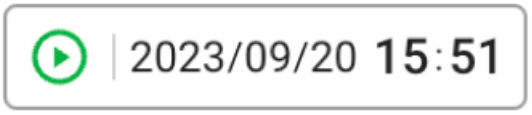
애플리케이션 대시보드에서는 애플리케이션의 주요 지표들을 에이전트를 통해 수집하고 차트를 구성해 실시간으로 모니터링하고 과거의 데이터를 조회할 수 있습니다. 대시보드에 배치된 위젯들을 통해 애플리케이션의 문제를 빠르게 파악하고 분석할 수 있습니다.

CPU, Memory를 제외한 일반적인 차트의 경우 안정적인 데이터는 파란색 계열로 표현되고, 문제로 식별되는 요소들은 붉은색 계열로 표시되어 현황을 쉽게 인지할 수 있습니다.

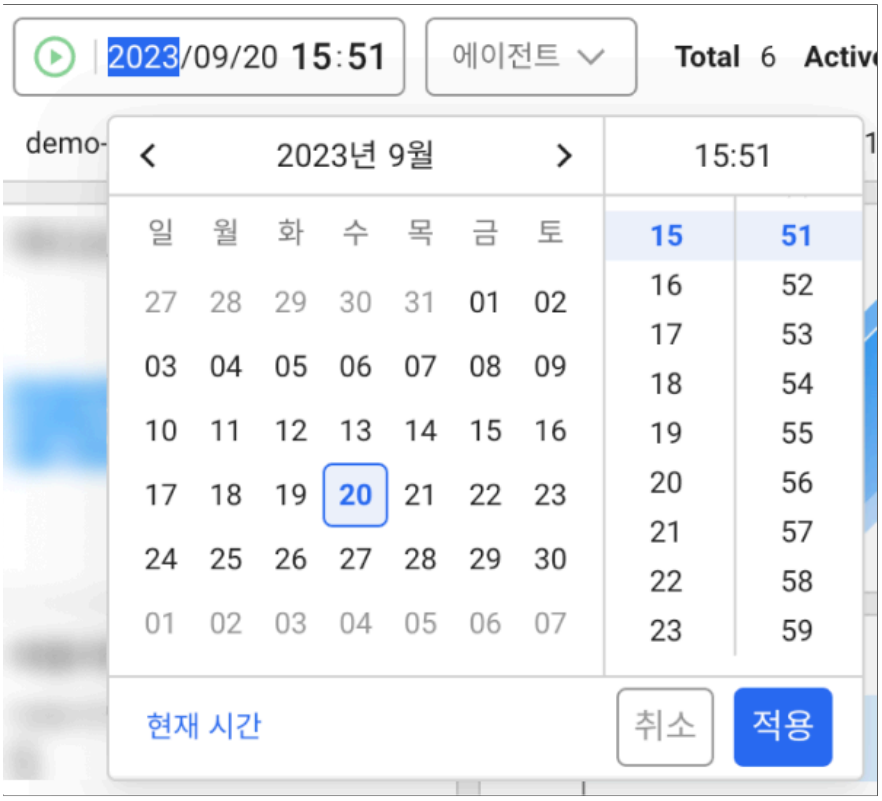
애플리케이션 대시보드

과거 데이터 조회하기

애플리케이션 대시보드는 기본적으로 실시간 모니터링 기능을 제공하지만 과거 시간의 데이터를 조회할 수도 있습니다.

실시간 모드	비실시간 모드
	

1. 화면 왼쪽 위에 시간 선택자에서 ⏸ 버튼을 선택하세요.
2. 날짜 및 시간 영역을 클릭해 원하는 시간대를 설정하세요.

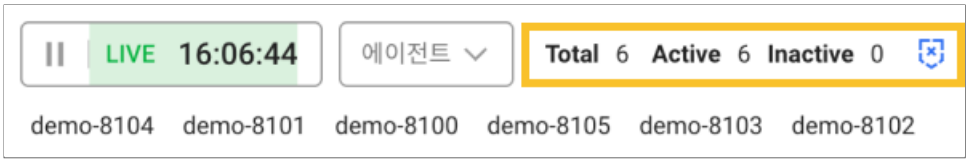


3. **적용** 버튼을 선택하세요.

사용자가 설정한 시간을 기준으로 대시보드에 배치한 위젯의 데이터를 갱신합니다. 실시간 모드로 전환하려면 ▶ 버튼을 선택하세요.

에이전트 확인하기

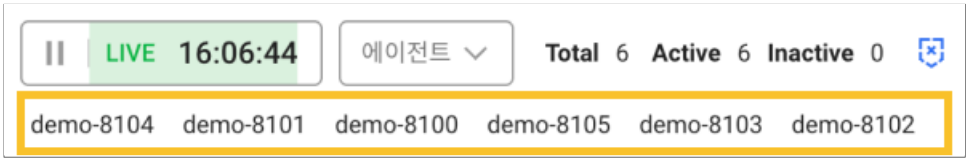
에이전트 연결 상태 확인하기



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

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

에이전트별 모니터링

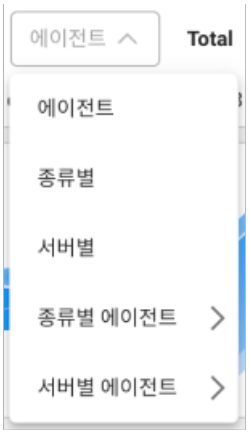


기본적으로 대시보드에는 모든 에이전트로부터 수집한 지표들을 차트에 표시하지만 에이전트별로 데이터를 조회할 수도 있습니다. 시간 선택자 아래에 위치한 에이전트를 하나 또는 둘 이상을 선택하세요. 선택한 에이전트의 지표들로 대시보드에 배치된 위젯의 데이터를 갱신합니다.

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

> 프로젝트에 연결된 에이전트의 수가 많을 경우 에이전트의 이름을 짧게 설정하는 것이 효율적입니다. 에이전트 이름 설정에 대한 자세한 내용은 다음 문서를 참조하세요.

분류별 에이전트 모니터링



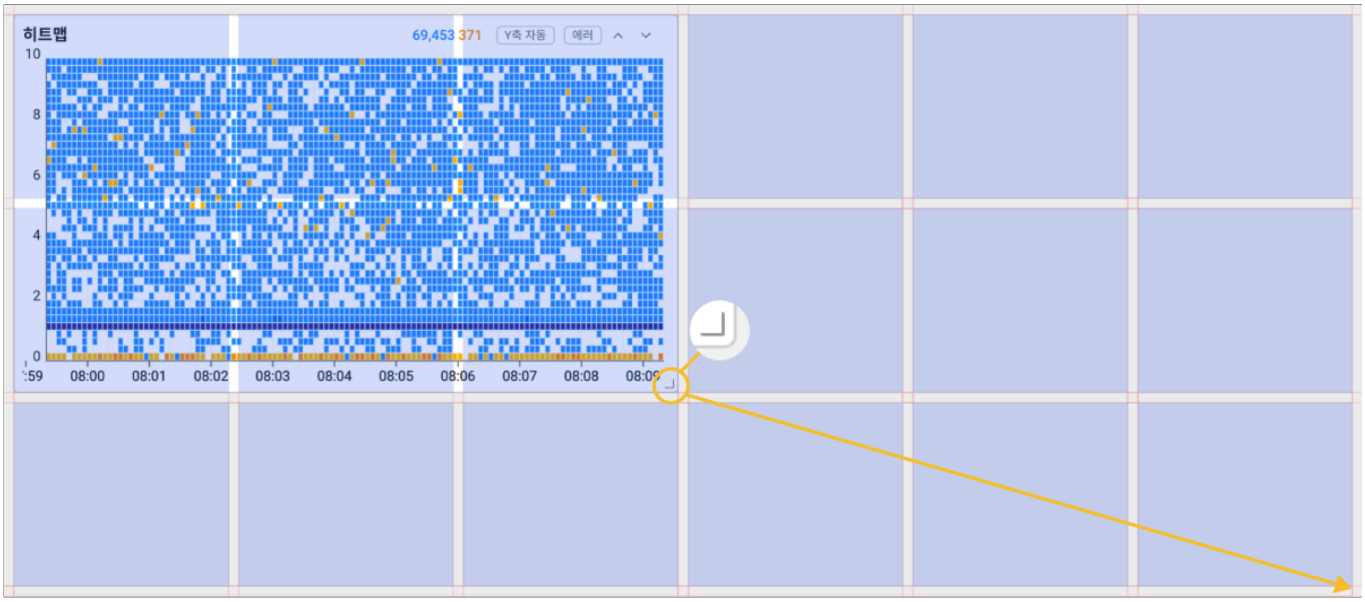
에이전트 설정을 통해 분류한 그룹 단위로 에이전트를 선택해 모니터링할 수 있습니다.

- **에이전트**: 개별 에이전트를 선택하거나 모두 선택할 수 있습니다.
- **종류별**: 에이전트 설정에서 `whatap.okind` 로 분류된 그룹 단위로 에이전트를 모니터링합니다.
- **서버별**: 에이전트 설정에서 `whatap.onode` 로 분류된 그룹 단위로 에이전트를 모니터링합니다.
- **종류별 에이전트**: 에이전트 설정에서 `whatap.okind` 로 분류된 에이전트 목록만을 모니터링합니다.
- **서버별 에이전트**: 에이전트 설정에서 `whatap.onode` 로 분류된 에이전트 목록만을 모니터링합니다.

대시보드 위젯 편집하기

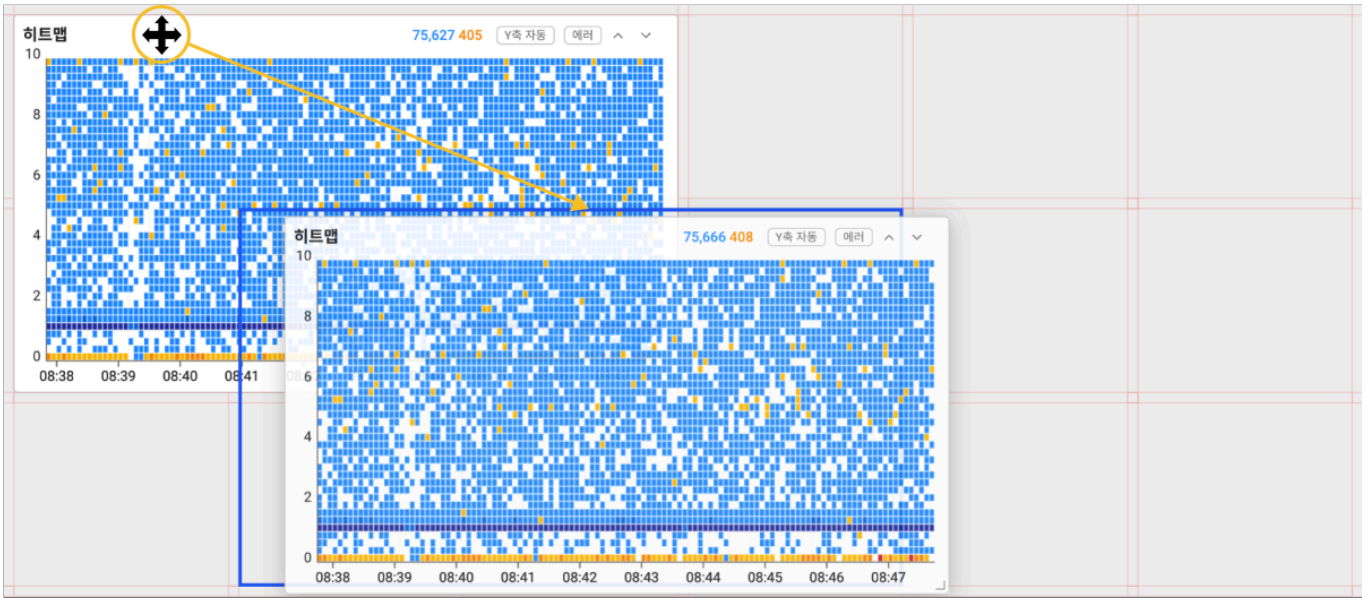
대시보드에 배치한 위젯은 사용자가 원하는 크기로 조절할 수 있고, 원하는 위치에 배치할 수 있습니다. 불필요하다고 생각되는 위젯은 삭제하고 다시 추가할 수도 있습니다.

위젯 크기 조절하기



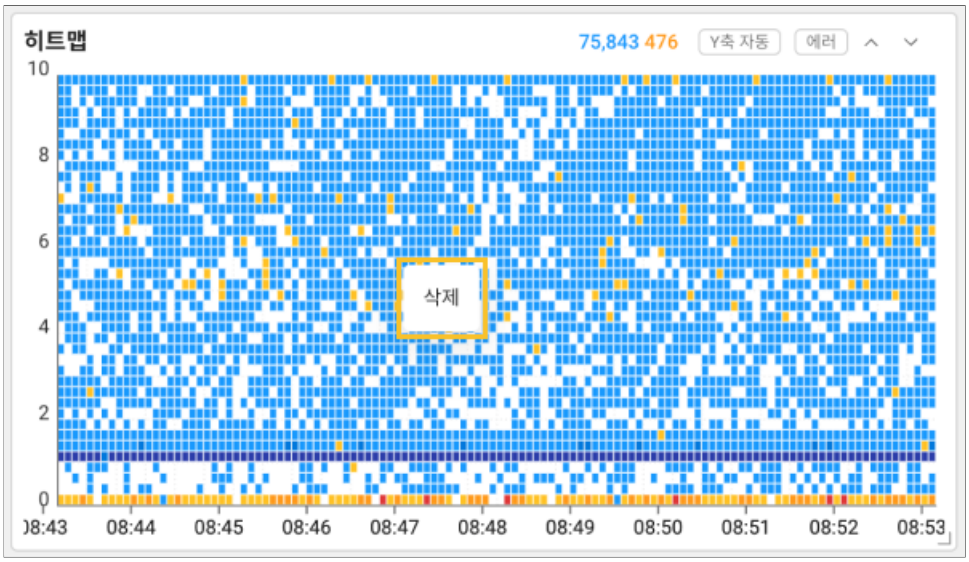
위젯의 오른쪽 아래에 ⌟ 요소를 마우스로 클릭한 상태에서 원하는 크기로 드래그하세요. 균일한 가로, 세로 비율의 격자가 표시되고, 격자 단위로 위젯의 크기를 조절할 수 있습니다.

위젯 이동하기



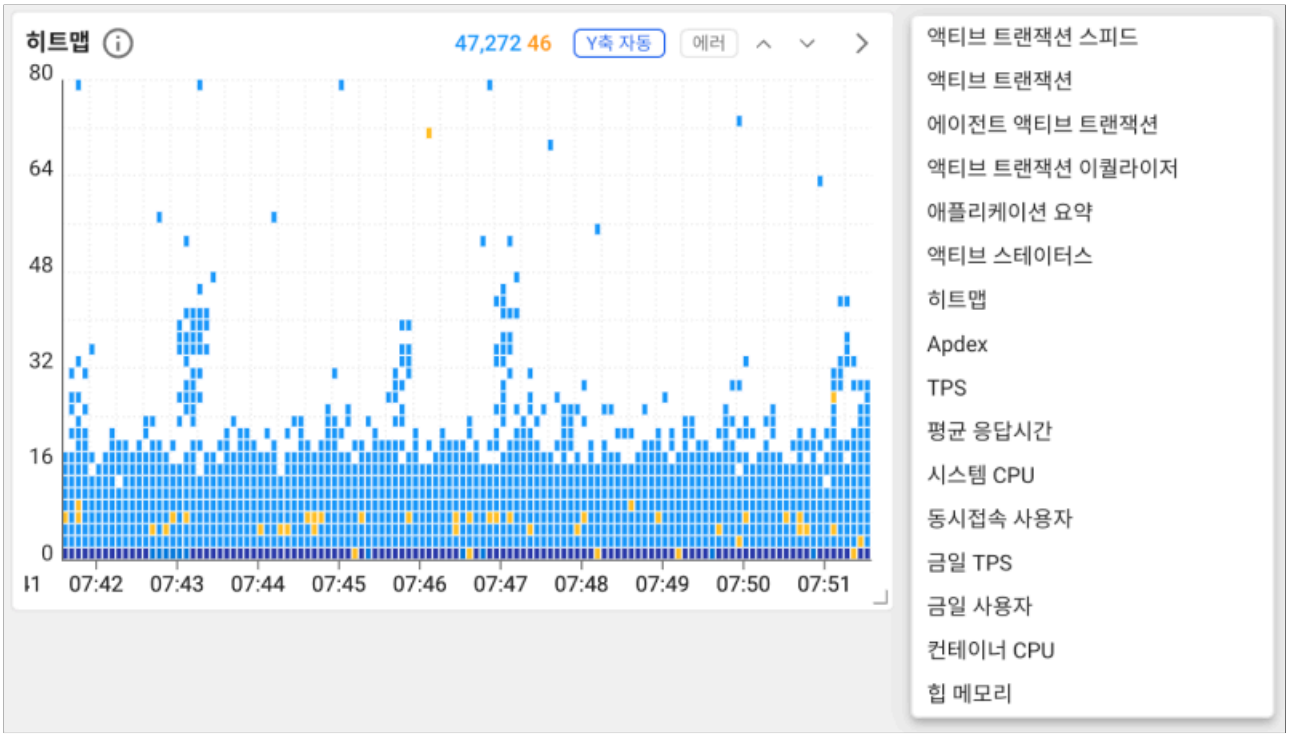
위젯의 윗 부분으로 마우스 커서를 이동하면 커서 모양이 ✛ 모양으로 변경됩니다. 이때 마우스 왼쪽 버튼을 클릭한 상태로 원하는 위치로 드래그하여 위젯을 이동할 수 있습니다.

위젯 삭제하기



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

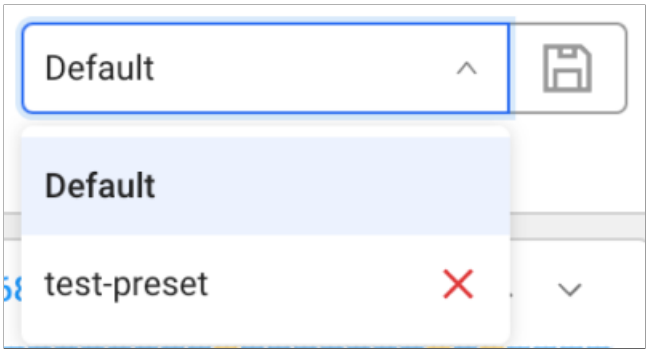
위젯 추가하기



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

> ⓘ 현재 추가할 수 있는 위젯은 고정적이지만 향후 업데이트를 통해 위젯 지원을 늘려갈 계획입니다.

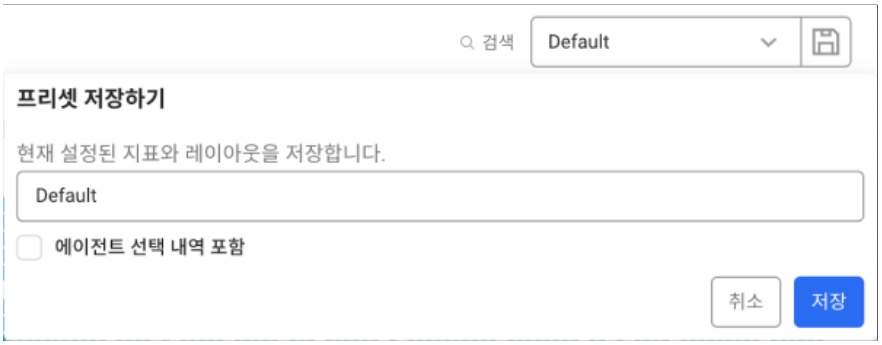
프리셋



대시보드에서 사용자가 설정한 위젯의 설정과 레이아웃 상태를 저장하고 불러올 수 있습니다. 기본 프리셋(**Default**)은 변경할 수 없으며, 원하는 형식으로 위젯의 크기를 조절하고, 배치해 새로운 프리셋을 만들 수 있습니다.

새로운 프리셋 만들기

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



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

4. **저장** 버튼을 선택하세요.

프리셋 목록에서 새로 저장한 프리셋을 확인할 수 있습니다.

- 새로 만든 프리셋에 변경 사항이 생겼다면 다시 프리셋을 저장해야 합니다. 버튼을 선택한 다음 같은 이름으로 프리셋을 저장하세요. 기존의 프리셋에 변경 사항을 덮어쓰기합니다.
- 대시보드의 변경 사항을 저장하지 않고 다른 메뉴로 이동하면 변경 사항은 저장되지 않습니다.
- 프리셋은 사용자 계정으로 저장되며 현재는 다른 사용자와 공유할 수 없습니다. 더 나은 기능을 제공하도록 업데이트 예정입니다.

프리셋 삭제하기

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

위젯 옵션 이용하기

위젯의 위에 표시된 아이콘 버튼의 기능은 다음과 같습니다.

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

위젯에 따라 제공되는 옵션은 다를 수 있습니다.

위젯 알아보기

이 문서에서는 **애플리케이션 대시보드**에 배치된 위젯에 대한 설명을 주로 다룹니다. **애플리케이션 대시보드**를 통해 트랜잭션 및 사용자, 서비스, 리소스 분석 방법에 대한 자세한 내용은 다음 문서를 참조하세요.

- 애플리케이션 대시보드 살펴보기
- 히트맵 트랜잭션
- DB 연결 지연과 커넥션 풀

에이전트 관련 위젯

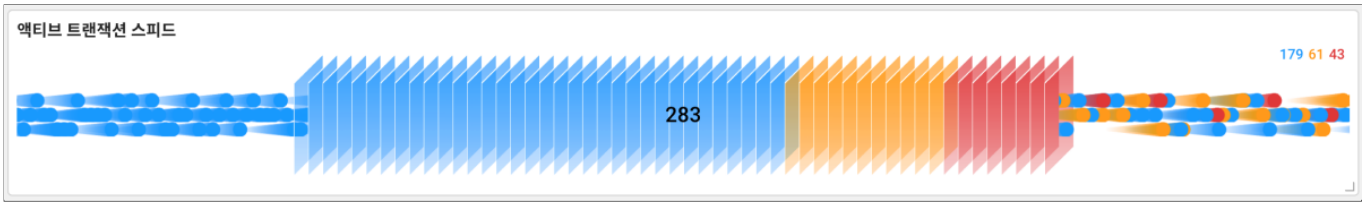
애플리케이션에 설치한 에이전트를 통해서 애플리케이션의 정보를 확인할 수 있습니다.

트랜잭션 관련 위젯

트랜잭션을 통해 확인할 수 있는 장애의 현황은 우선 응답시간을 통해 알 수 있습니다. 또한 진행 중인 트랜잭션이 종료되지 않는다면 이 또한 장애로 인식해야 합니다. 와탭은 진행 중인 상태의 시간에 따라 구간을 나누어 표시합니다. 파랑색(Normal)은 응답 시간이 정상인 트랜잭션, 주황색(Slow)은 응답 시간이 8초 정도의 느린 트랜잭션, 빨간색(Very slow)은 응답 시간이 보통의 2배 이상으로 느린 트랜잭션을 의미합니다. 이를 통해 사용자는 직관적으로 가장 빨리 장애를 인지할 수 있습니다.

> Java 플랫폼에서는 **연계 프로젝트 관리** 메뉴를 통해 데이터베이스 프로젝트를 추가하면 다음 위젯에서 연계한 프로젝트의 성능 지표를 조회할 수 있습니다.
>
> - **액티브 트랜잭션 스피드**, **액티브 트랜잭션**: 현재 진행 중인 트랜잭션과 연계된 데이터베이스의 SQL 성능을 조회할 수 있습니다.
> - **히트맵**: 과거 시점의 트랜잭션과 연계된 데이터베이스의 성능 지표를 조회할 수 있습니다.
>
> 연계 프로젝트 추가 및 분석에 대한 자세한 내용은 다음 문서를 참조하세요.

액티브 트랜잭션 스피드



현재 진행 중인 트랜잭션(가운데 영역)과 종료된 트랜잭션(오른쪽 영역) 현황을 확인할 수 있습니다. 왼쪽에서 오른쪽으로 지나가는 물방울 모양의 아이콘은 트랜잭션을 의미합니다. 차트 영역을 클릭하면 진행 중인 트랜잭션 목록을 확인할 수 있는 팝업 창이 나타납니다. 트랜잭션 목록 중 하나를 선택하면 상세한 트랜잭션 정보와 SQL 쿼리, 콜 스택 정보를 확인할 수 있습니다.

- 왼쪽 영역

  요청 트랜잭션 수를 의미하며, 트랜잭션 수에 비례해 더 많이, 더 빠르게 표시합니다.

- 가운데 영역

액티브 트랜잭션의 상태를 속도로 색상으로 분류하여 우측 상단에 개수로 표시합니다. 차트 안의 숫자는 전체 액티브 트랜잭션의 수입니다. 차트의 색상의 의미는 다음과 같습니다.

- 파란색(Normal): 0초 ~ 3초
- 주황색(Slow): 3초 ~ 8초
- 빨간색(Very Slow): 8초 이상

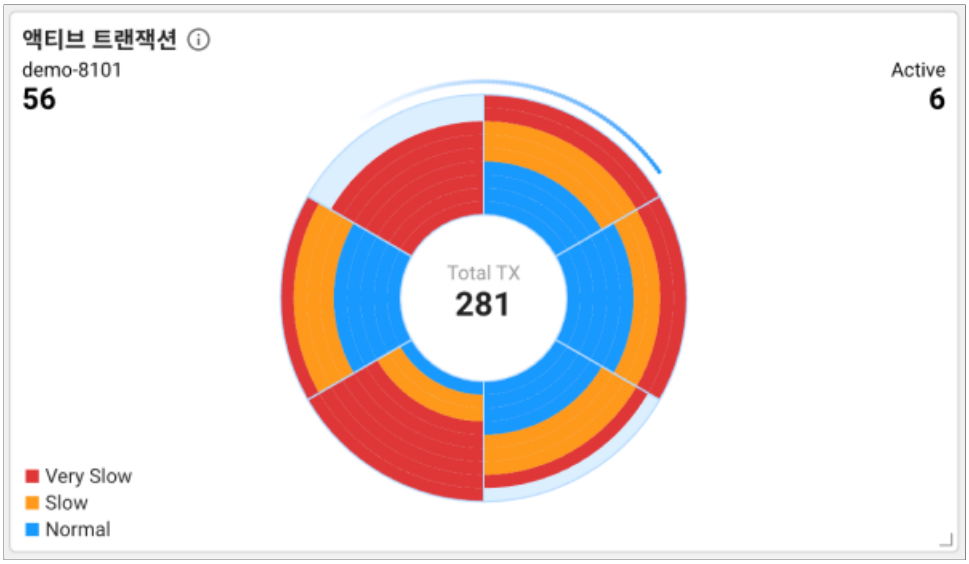
- 오른쪽 영역

애플리케이션 성능 지표(Application Performance Index, Apdex)를 의미합니다. 웹 애플리케이션의 고객 만족도를 측정한 지표로, 0 ~ 1 사이의 값을 갖습니다. 물방울이 날아가는 속도는 Apdex 트랜잭션의 총 수에 비례하며, 물방울의 색상의 의미는 다음과 같습니다.

- 파란색: Apdex 만족(Satisfied) 수
- 주황색: Apdex 허용(Tolerating) 수
- 빨간색: Apdex 불만(Frustrated) 수 = `APDEX 트랜잭션 총 수` - (`Apdex 만족 수` + `Apdex 허용 수`)

> ⓘ 애플리케이션 성능 지표(Application Performance Index, Apdex)에 대한 자세한 내용은 다음 링크를 참조하세요.

액티브 트랜잭션

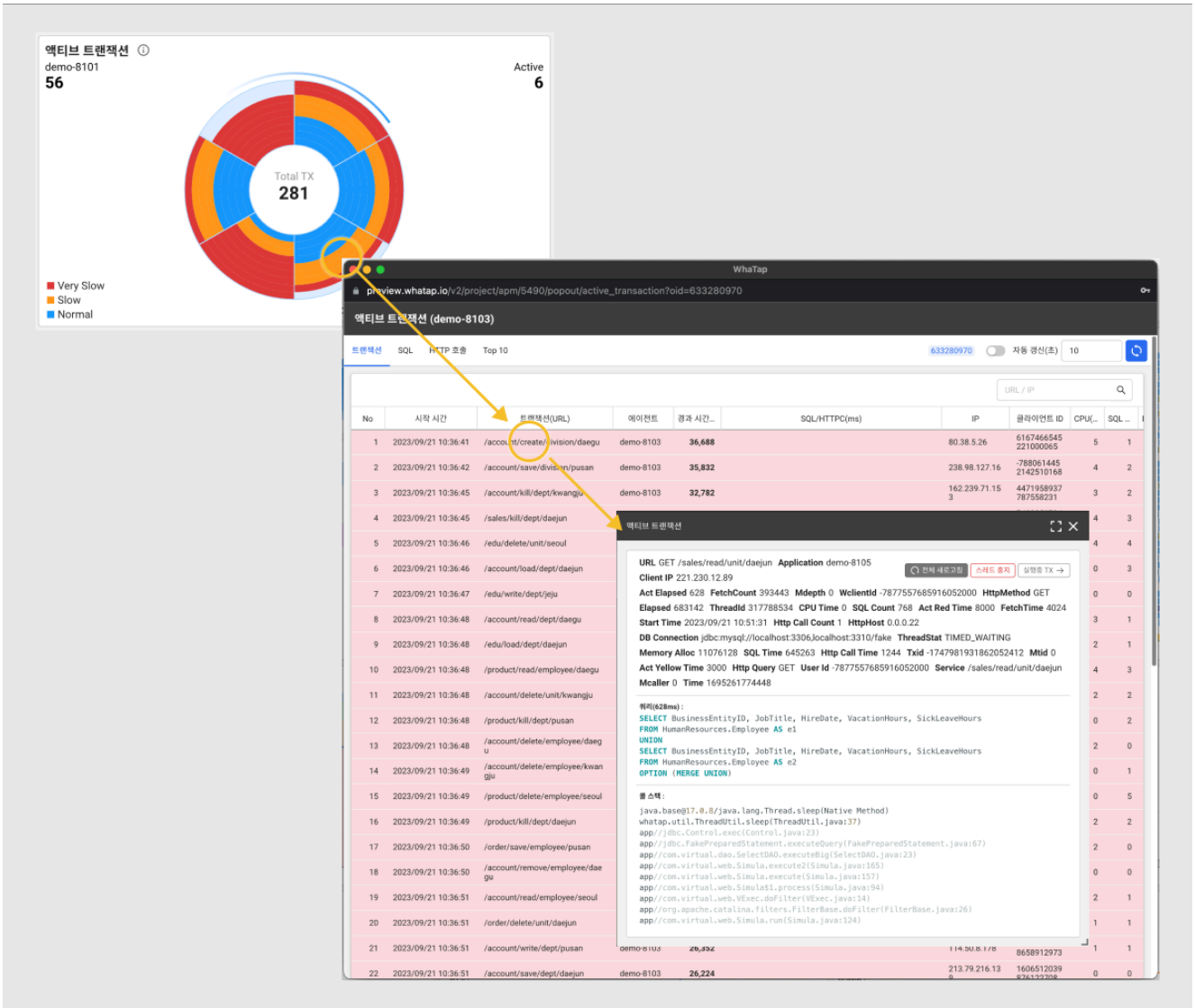


액티브 트랜잭션 위젯에서는 진행 중인 트랜잭션을 **아크 이퀄라이저** 차트로 표현합니다. 지연 발생 현황을 에이전트별로 확인할 수 있습니다. 에이전트 수만큼 아크가 분할됩니다. 5초마다 현재 서버에서 처리 중인 요청의 수를 표현해서 해당 요청이 각각 어느 정도의 시간 동안 처리 중인지 알 수 있습니다. 5초 간격의 시간에 감지된 요청들은 위험 여부를 파악할 수 있도록 다음과 같이 색으로 분류합니다.

- 빨간색(Very Slow): 8초 이상
- 주황색(Slow): 3초 ~ 8초
- 파란색(Normal): 0초 ~ 3초

파란색이 많이 표현되는 상황은 문제가 되지 않습니다. 이 중 일부가 주황색이나 빨간색으로 변경되는지 추이를 지켜봐야 합니다. 장애 상황에는 진행 중 트랜잭션 수가 증가하고 빨간색 비율이 높아집니다. 반면, 응답이 매우 빠른 시스템의 경우 처리하는 트랜잭션 수(TPS)는 높아도 진행 중인 트랜잭션 수는 낮을 수 있습니다.

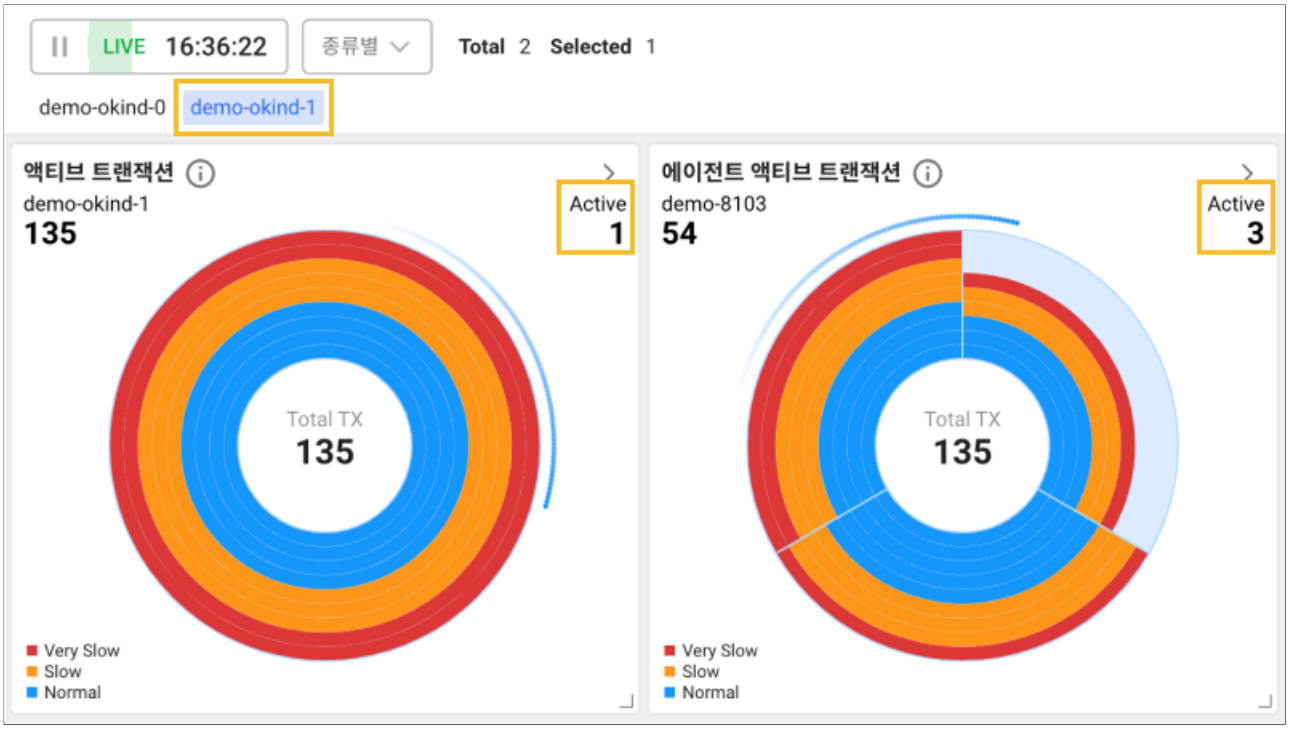
- 왼쪽 위에는 진행 중인 트랜잭션이 가장 많은 에이전트 이름을 표시합니다.
- 가운데 숫자는 전체 에이전트의 액티브 트랜잭션 수의 합계입니다.
- 액티브 트랜잭션 둘레에 두 개의 바는 처리량(TPS)에 따라 3단계 속도로 차트 주변을 회전합니다.
- 분할된 아크 영역을 선택하면 해당 에이전트에서 진행 중인 트랜잭션 목록 팝업창이 나타납니다. 상세한 트랜잭션 정보를 확인하려면 트랜잭션 목록에서 원하는 항목을 하나 선택하세요.



ⓘ ◦ 트랜잭션 목록에서 선택한 항목이 이미 종료된 트랜잭션일 경우 **트랜잭션 보기** 버튼을 선택하세요. **트랜잭션 정보** 창이 나타납니다. **트랜잭션 정보** 창에 대한 자세한 내용은 다음 문서를 참조하세요.

◦ 트랜잭션에 대한 자세한 내용은 다음 문서를 참조하세요.

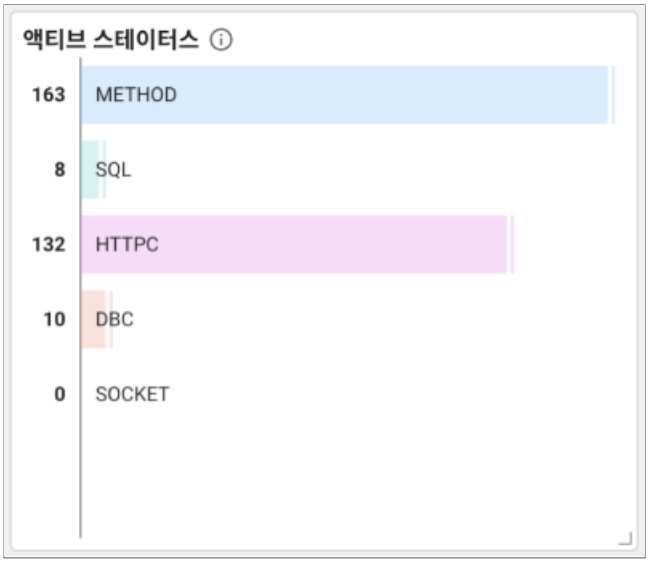
에이전트 액티브 트랜잭션



액티브 트랜잭션 위젯은 에이전트 선택 옵션에서 선택한 기준에 따라 병합된 데이터를 차트에 표시합니다. 반면 **에이전트 액티브 트랜잭션** 위젯은 선택한 에이전트 분류 기준에 포함된 에이전트의 현황을 개별로 확인할 수 있습니다.

대시보드에서 마우스 오른쪽 버튼을 클릭한 다음 위젯 목록에서 **에이전트 액티브 트랜잭션** 위젯을 선택하세요.

액티브 스테이터스



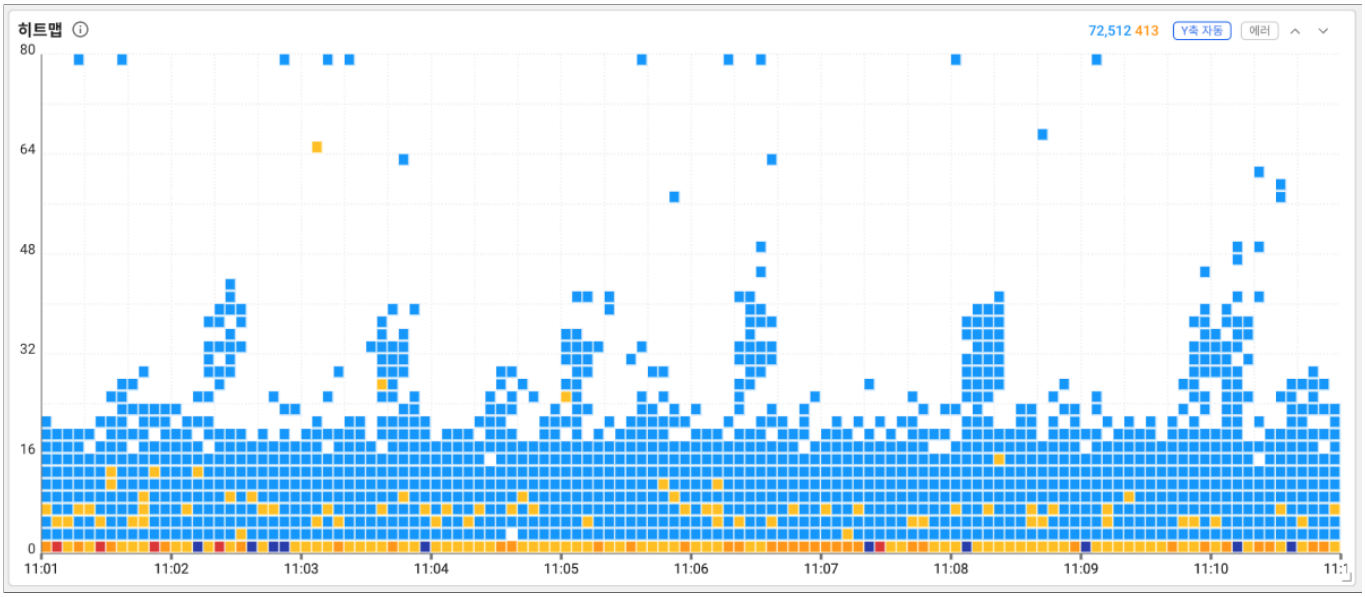
프로젝트 내의 모든 에이전트는 5초마다 액티브 트랜잭션 수를 수집함과 동시에 진행 상태를 수집합니다. 진행 상태는 **METHOD**, **SQL**, **HTTPC**, **DBC**, **SOCKET**으로 분류할 수 있습니다.

- **METHOD**: 메소드를 수행 중인 상태입니다. 일반적인 상황으로 파란색으로 표시됩니다.
- **SQL**: SQL을 수행 중인 상태입니다. **SQL** 수치가 증가한다면 데이터베이스 서버와의 연결에 문제를 확인해 보세요.
- **HTTPC**: 외부 API를 호출 중인 상태입니다. **HTTPC** 수치가 증가한다면 외부와 연결된 서버의 응답이 제대로 이루어지지 않는다고 봐야 합니다.
- **DBC**: 트랜잭션이 Connection Pool로부터 새로운 Connection을 획득(get)하려는 상태입니다. DB Connection Pool의 개수가 부족하면 새로운 연결 요청이 발생할 때마다 지연이 되면서 성능 장애의 원인이 됩니다. 이 경우 **DBC** 수치가 증가합니다.

  > ⓘ DB 연결 지연과 관련한 자세한 내용은 다음 문서를 참조하세요.

- **SOCKET**: 외부 시스템과의 TCP 연결 시도를 의미합니다. **SOCKET** 수치가 지속적으로 증가한다는 것은 외부 시스템과의 연결이 되지 않아 장애가 발생 중일 가능성이 높습니다.

히트맵



히트맵 차트는 종료된 트랜잭션 응답시간을 분포도로 표현합니다. 가로축은 트랜잭션 종료 시간입니다. 세로축은 수행 시간입니다. 수행 시간이 긴 트랜잭션은 분포도 상단에 위치합니다. **히트맵** 내의 차트 영역을 드래그하면 세부 트랜잭션 정보를 확인할 수 있는 **트레이스 분석** 창이 나타납니다. **트레이스 분석**에 대한 자세한 내용은 다음 문서를 참조하세요.

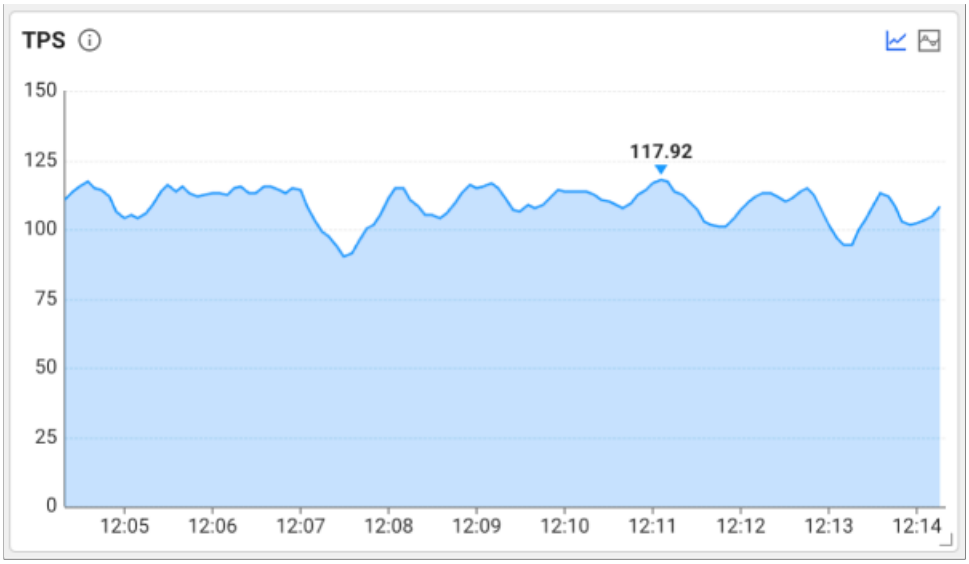
- 정상 트랜잭션은 하늘색 → 파란색 → 남색 순으로 표현됩니다.
- 에러 트랜잭션은 노란색 → 주황색 → 빨간색 순으로 표현됩니다.
- 오른쪽 위에 위치한 숫자는 차트 내 트랜잭션 건 수/에러 건수를 의미합니다.
- **Y축 자동** 버튼을 선택하면 수행 시간을 나타내는 Y축을 자동 조정합니다.
- **에러** 버튼을 선택하면 에러 트랜잭션만 표시합니다.
- ∧ 또는 ∨ 버튼을 선택해 5초 ~ 80초까지 차트를 확대/축소할 수 있습니다.

ⓘ
- 오류가 발생하지 않았고 수행 시간 500ms 이하 트랜잭션의 세부 정보는 url 당 5분에 1건만 수집됩니다.
- TPS나 평균 응답시간 같은 통계정보에는 영향을 미치지 않습니다.
- 에이전트 설정을 통해 정책을 설정할 수 있습니다. 자세한 내용은 다음 문서를 참조하세요.
- 히트맵 위젯을 통해 트랜잭션을 분석하는 과정은 다음 문서를 참조하세요.

서비스 관련 위젯

서비스 관련 위젯을 통해 애플리케이션 성능 튜닝을 위한 지표로 활용할 수 있습니다.

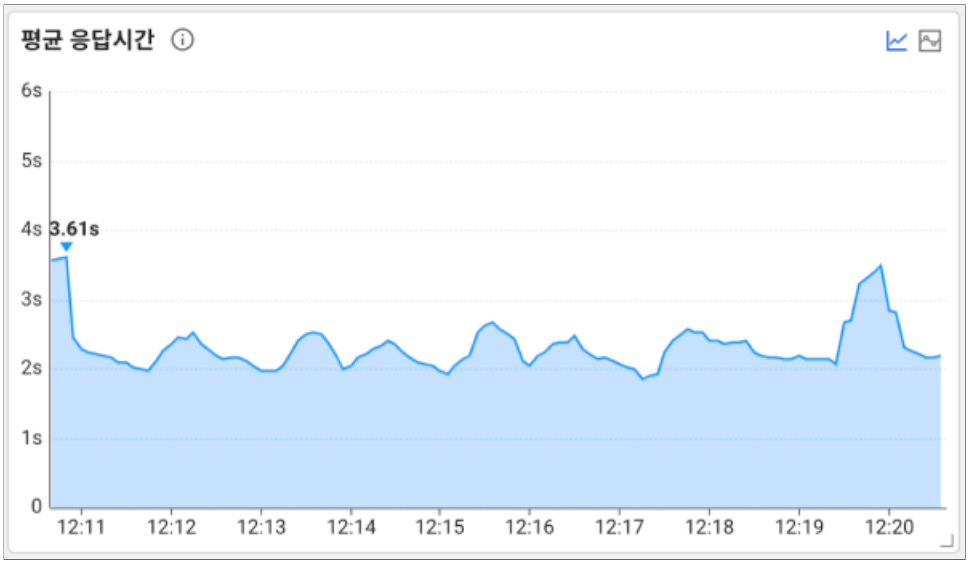
TPS



Tansaction Per Second(TPS)는 초당 처리한 트랜잭션 건수를 의미하며 서비스 성능 지표의 기준이 됩니다. 5초마다 처리한 트랜잭션의 수를 초당 건수로 환산해 차트로 표현합니다. 최근 10분간의 TPS를 보여줍니다.

> TPS, 응답시간, CPU 사용률 분석에 대한 자세한 내용은 다음 문서를 참조하세요.

평균 응답시간

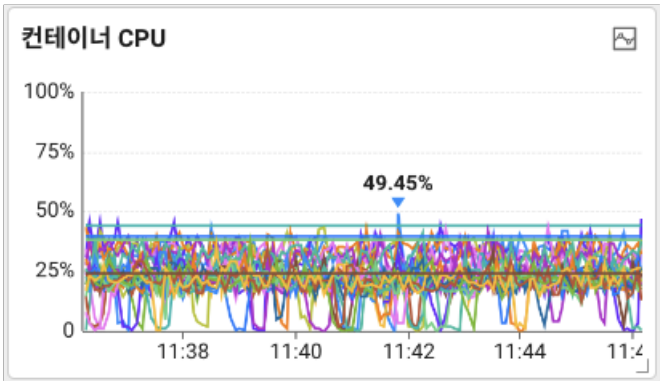


애플리케이션 서버가 사용자에게 요청 결과를 반환하는 데 걸리는 시간을 실시간으로 모니터링합니다. **평균 응답시간**은 튜닝 지표로서 의미를 가집니다. 와탭의 모니터링 서비스는 5초 간격으로 트랜잭션의 평균 응답 시간을 계산합니다.

> TPS, 응답시간, CPU 사용률 분석에 대한 자세한 내용은 다음 문서를 참조하세요.

리소스 관련 위젯

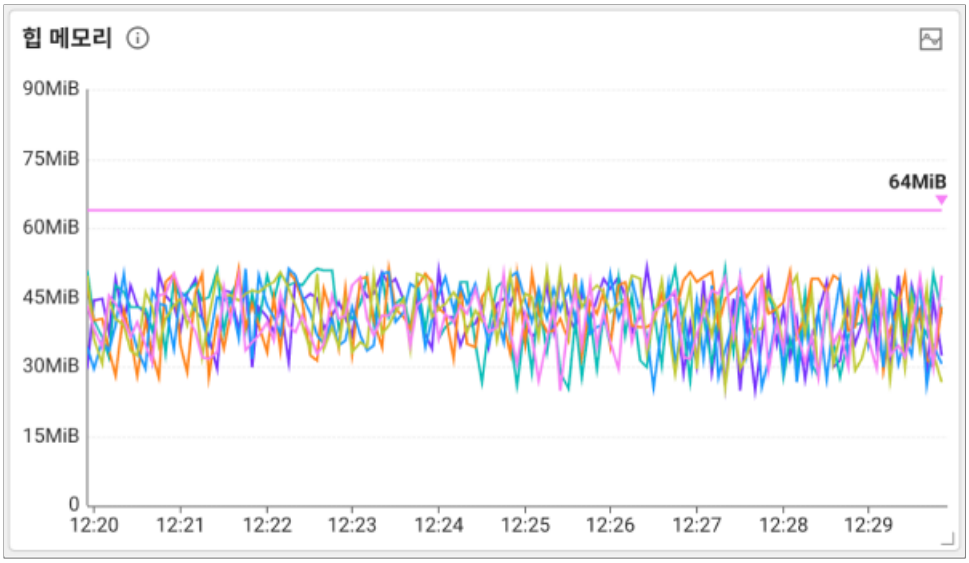
컨테이너 CPU



컨테이너 CPU 사용량입니다. 실시간으로 CPU 사용량 변화 추이를 파악할 수 있습니다.

> TPS, 응답시간, CPU 사용률 분석에 대한 자세한 내용은 다음 문서를 참조하세요.

힙 메모리



각 서버당 사용할 수 있는 최대 메모리와 현재 메모리를 표시해 위험 수치에 있는 서버를 확인할 수 있습니다. 시간에 따른 메모리 사용량 변화를 실시간으로 볼 수 있습니다.

메모리 라인 차트는 보통 계속해서 물결칩니다. 애플리케이션 서버가 요청을 처리하기 위해 메모리를 사용할 때 증가합니다. GC(Garbage Collection)를 통해서 메모리를 정리할 경우에는 감소합니다. **힙 메모리** 지표는 평균값을 사용해 표기합니다.

> - Java와 Node.js는 힙 메모리 총량과 사용량을 보여줍니다.
> - PHP 및 Python, .Net은 프로세스 메모리 사용량을 보여줍니다.

> 힙 메모리 및 힙 메모리 차트 분석에 대한 자세한 내용은 다음 링크를 참조하세요.
>
> - 학습하기: 힙 메모리 메트릭스 및 메트릭스 알림 설정
> - Java 힙 메모리 차트 분석: Ch.1 힙차트 관찰하기
> - JAVA 힙메모리 차트 분석: Ch.2 메모리 릭, 그리고 힙덤프 분석

사용자 관련 위젯

동시접속 사용자

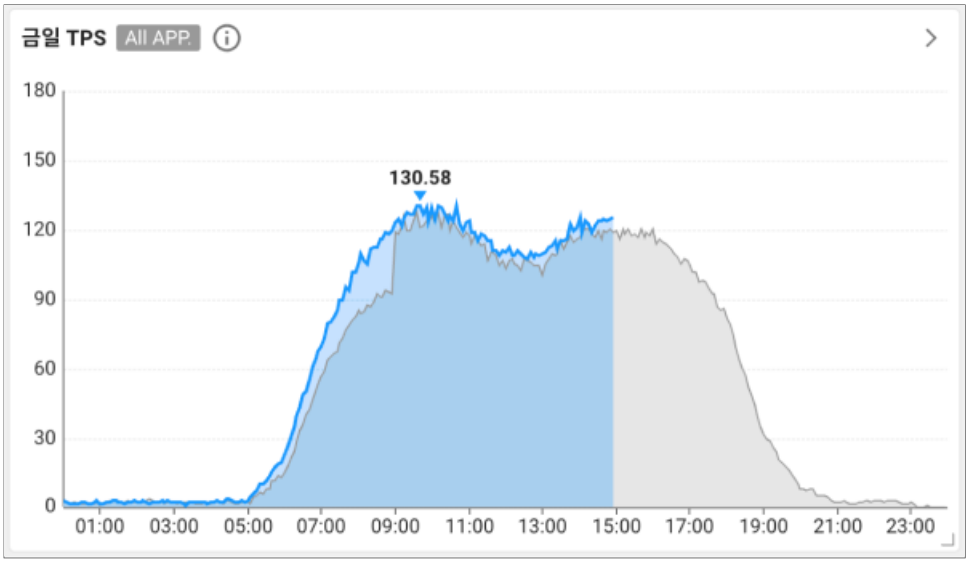
실시간 브라우저 사용자 수를 보여줍니다. 10초마다 최근 5분 이내에 트랜잭션을 일으킨 사용자를 중복없이 카운팅해 표시합니다. 사용자 브라우저의 IP를 기반으로 카운팅합니다.

- 일반적으로 접속 사용자는 현재 네트워크로 접속해 연결된 사용자를 의미합니다. 하지만 웹 시스템은 비 연결 네트워크를 사용하기 때문에 다릅니다. 접속되어 있다는 의미보다는 최근 요청을 보낸 사용자가 측정 대상이 될 수 있습니다.
- 실시간(동시접속) 사용자는 같은 시간대에 시스템을 사용하고 있는 사용자를 의미합니다. 실시간(동시접속) 사용자를 산정하거나 측정하는 방식은 다양합니다.
- 사용자 수는 단순히 합산하는 것이 아니라 **HyperLogLog**로 산출합니다. **HyperLogLog**는 매우 적은 메모리로 집합의 원소 개수를 추정하는 확률적 자료 구조를 의미합니다.
- 사용자 수 집계를 위한 에이전트 옵션 설정 방법은 다음 문서를 참조하세요.

1일 기준 비교 위젯

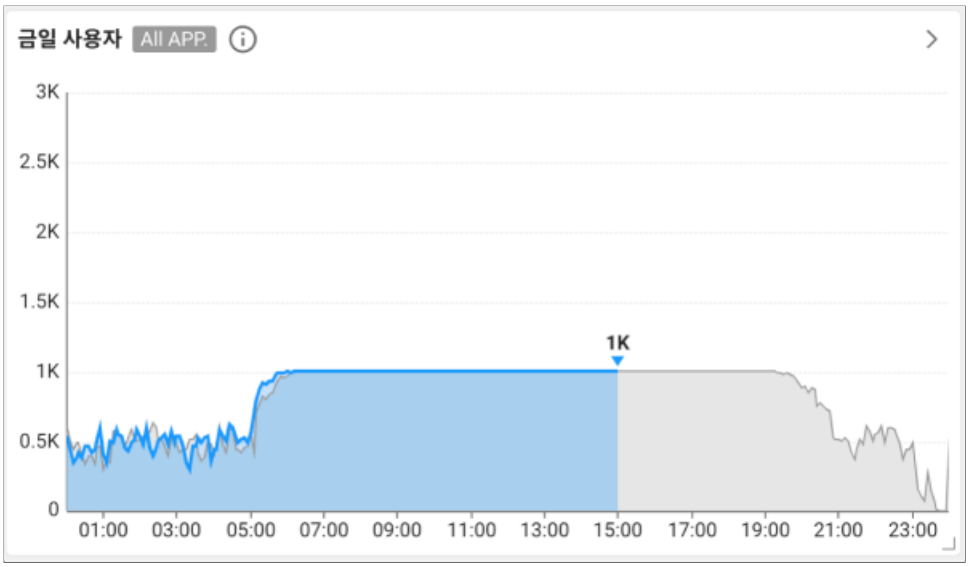
All APPS. 태그가 표시된 위젯은 항상 모든 에이전트에 대한 데이터를 표시합니다.

금일 TPS



금일 하루 동안의 TPS를 시간대별로 표시합니다. 어제 동시간대의 TPS(회색)와 비교해 확인할 수 있습니다. > 버튼을 클릭하면 **통계/보고서** > **일자별 애플리케이션 현황** 메뉴로 이동합니다.

금일 사용자



하루 동안의 **동시접속 사용자** 수를 시간대별로 나타낸 지표입니다. 어제 동시간대의 사용자 수(회색)와 비교해 확인할 수 있습니다. > 버튼을 클릭하면 **통계/보고서** > **일자별 애플리케이션 현황** 메뉴로 이동합니다.

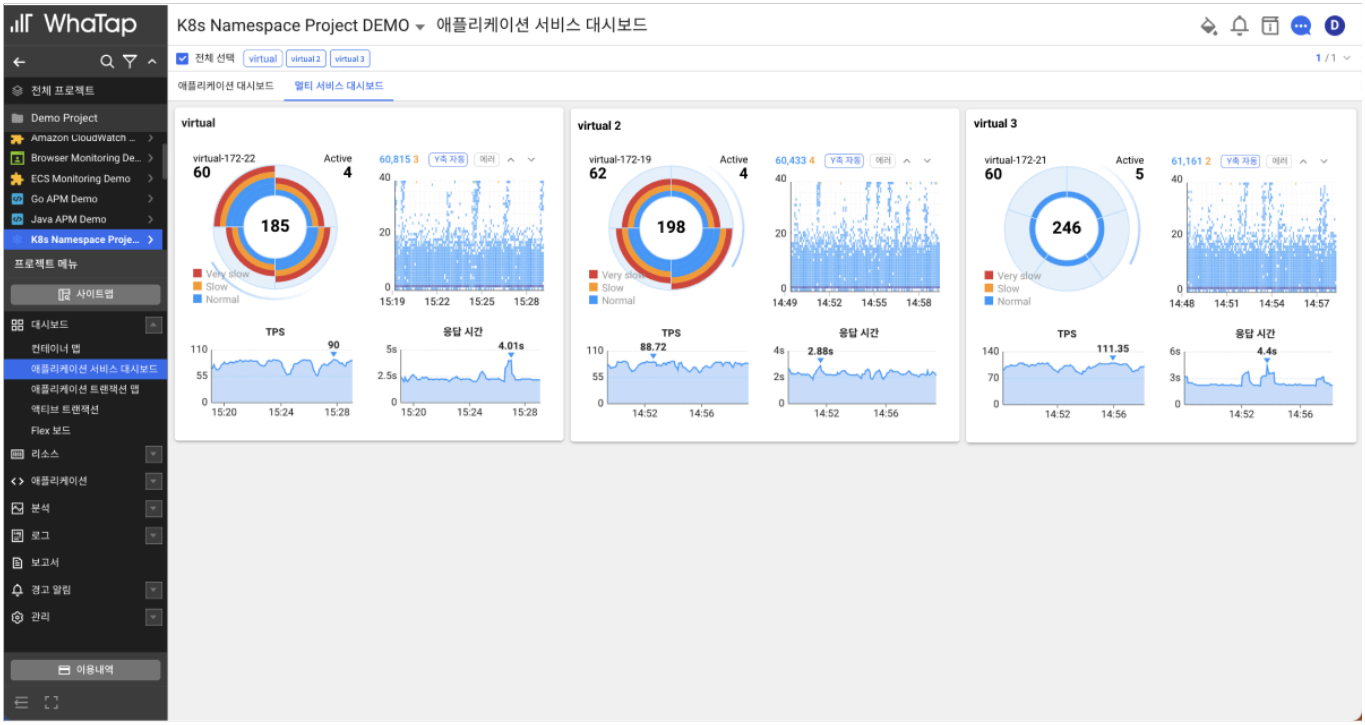
실시간 알림 확인하기

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

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

멀티 서비스 대시보드

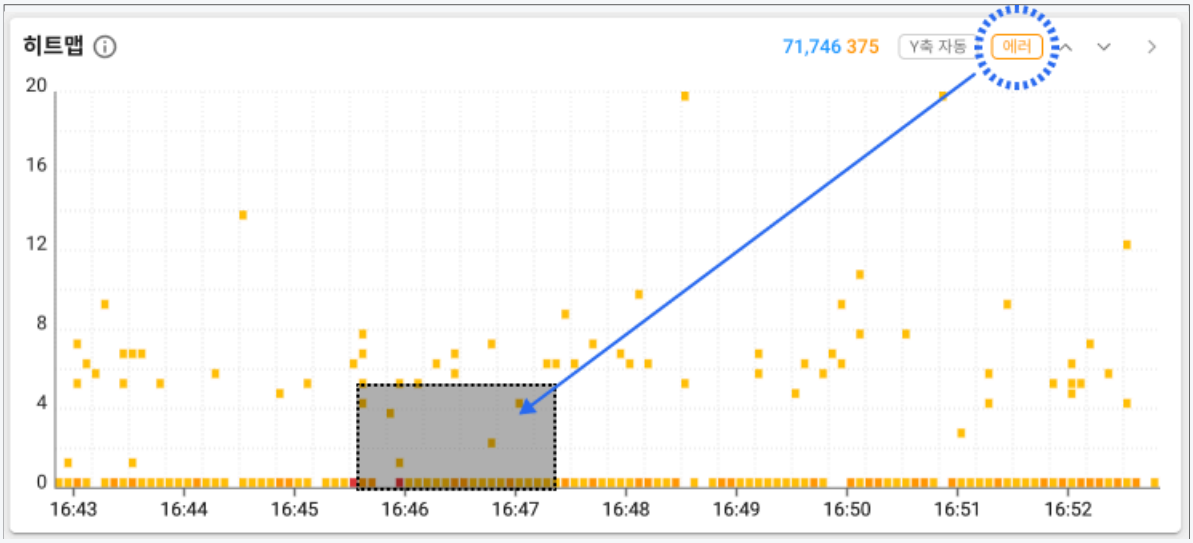
선택한 애플리케이션을 개별 카드 형태로 조회, 비교할 수 있습니다.



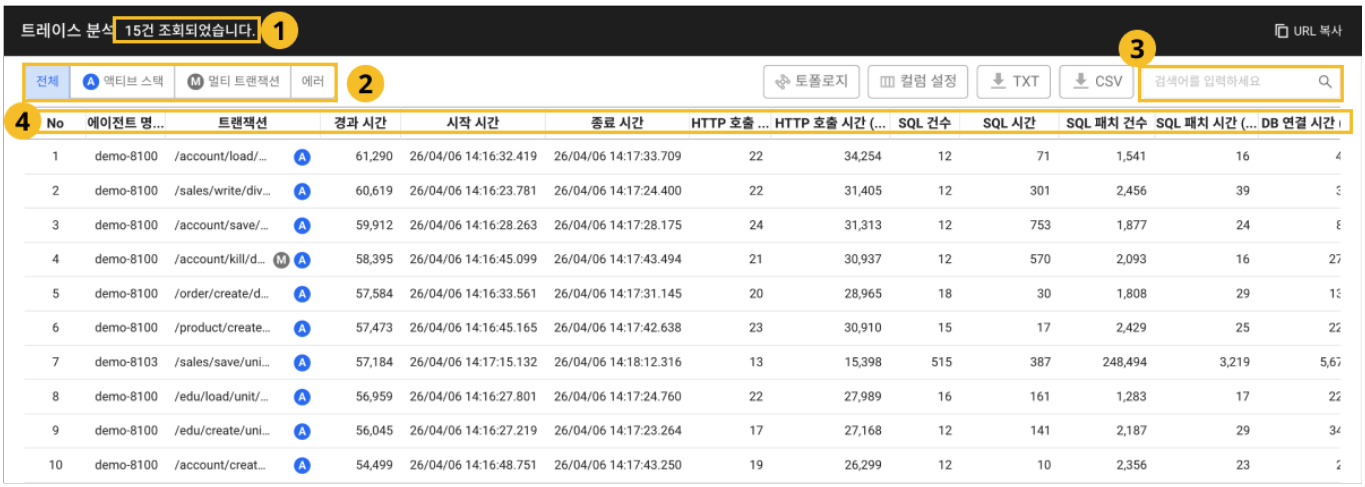
히트맵 - 트레이스 분석

애플리케이션 대시보드의 **히트맵** 위젯을 통해 **트레이스 분석** 창을 열어 트랜잭션 목록과 각 트랜잭션 하위의 스텝 정보를 한 번에 확인할 수 있습니다. 히트맵 위젯에서 원하는 영역을 드래그하세요. 선택한 영역의 트랜잭션 목록과 하위 스텝 정보를 포함하는 **트레이스 분석** 창이 나타납니다.

에러가 발생한 트랜잭션 정보만 분류해 확인하려면 **애플리케이션 대시보드**의 **히트맵** 위젯의 오른쪽 위에 **에러** 버튼을 선택한 다음 차트 영역을 드래그하세요.

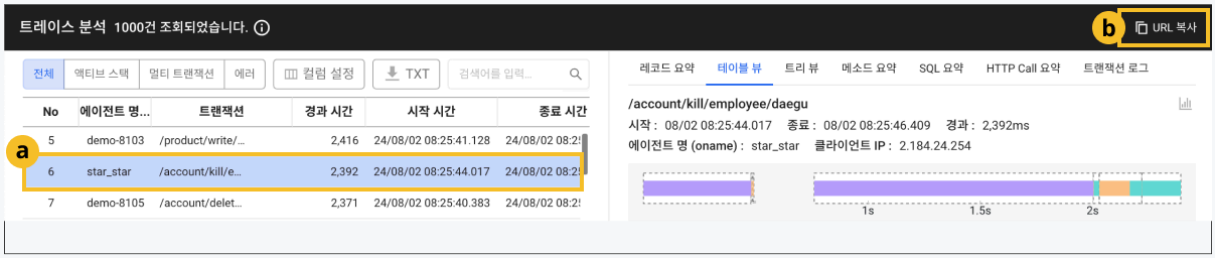


기본 화면 안내



- 1 **트랜잭션 조회 건수**: 히트맵 위젯에서 드래그한 영역의 트랜잭션 목록의 개수를 확인할 수 있습니다. 최대 1000건까지 조회할 수 있습니다.
- 2 **트랜잭션 분류**: A **액티브 스택**과 M **멀티 트랜잭션**, **에러** 항목을 구분하여 목록을 확인할 수 있습니다.
- 3 **트랜잭션 검색**: 트랜잭션 명과 에이전트 명, 에러 메시지를 기준으로 트랜잭션 목록을 검색할 수 있습니다.
- 4 **목록 정렬하기**: 테이블의 헤더 컬럼 항목을 클릭하면 선택한 항목을 기준으로 트랜잭션 목록을 정렬할 수 있습니다.
- 에러가 발생한 트랜잭션 항목은 목록에서 빨간색으로 표시됩니다.

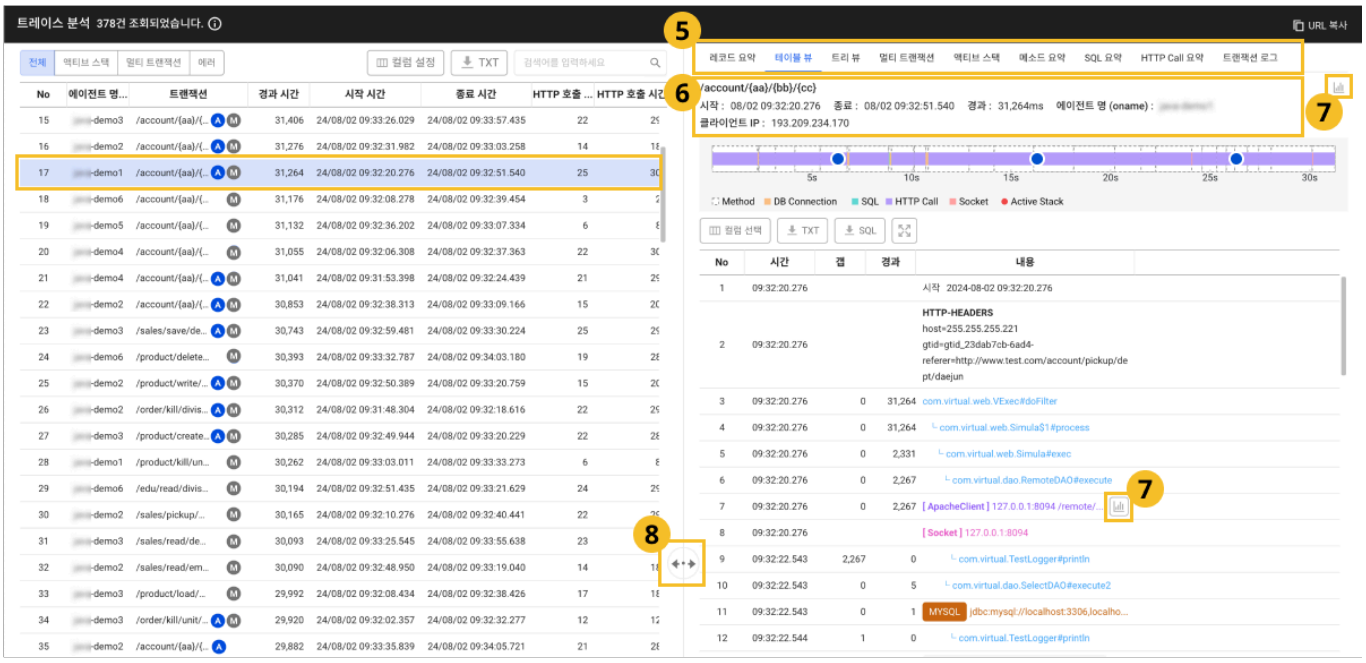
- 테이블에서 자주 보길 원하는 컬럼 항목을 앞으로 배치하거나 추가, 숨기기할 수 있습니다. 자세한 내용은 다음 문서를 참조하세요.
- **히트맵** 위젯에서 다른 영역을 드래그하면 새로운 트레이스 분석 창을 불러올 수 있습니다. 기존에 열린 트레이스 분석 창과 비교해 확인할 수 있습니다.
- 현재 사용자가 보고 있는 트랜잭션 항목을 다른 사용자에게 공유할 수 있습니다. 트랜잭션 항목을 선택한 상태(a)에서 트레이스 분석 창의 b **URL 복사** 버튼을 선택하세요. 복사한 URL 주소를 다른 사용자에게 전달하세요. URL 주소를 전달 받은 사용자는 브라우저 주소 표시줄에 URL 주소를 입력하면 같은 항목을 빠르게 확인할 수 있습니다.



트레이스 분석하기

트랜잭션 목록에서 상세 내용을 확인할 항목을 선택하세요. 화면 오른쪽에 선택한 트랜잭션에 대한 상세 분석 내용을 표시합니다. 이 상태에서 다른 트랜잭션 항목을 선택하면 선택한 트랜잭션 항목에 대한 상세 분석 내용으로 변경됩니다.

와탭은 트랜잭션의 성능을 분석하기 위해 이름과 클라이언트 정보 등의 속성, 트랜잭션의 처리 성능 그리고 각 구간별 상세 수행 이력을 수집하고 보여줍니다.



- 5 영역에서는 스텝 정보를 다양한 형식으로 확인할 수 있습니다. 원하는 형식의 탭을 선택하세요.
- 6 영역에서는 트랜잭션에 대한 기본 정보를 확인할 수 있습니다. 왼쪽 목록의 컬럼 항목과 일치하는 정보입니다.

- 7 : 해당 URL 또는 각 수행 구간의 통계 데이터 창이 나타납니다. 통계 데이터 창의 그래프 차트에서 원하는 시간을 클릭하면 **통계** 또는 **트랜잭션 검색** 메뉴로 이동합니다. 선택한 시간 기준으로 통계 데이터를 조회할 수 있습니다.
- 8 왼쪽 또는 오른쪽으로 드래그해서 화면의 크기를 조절할 수 있습니다. 더 많은 정보를 보길 원할 때 이용하면 유용합니다.

> - **통계** 메뉴에 대한 자세한 내용은 다음 문서를 참조하세요.
> - **트랜잭션 검색** 메뉴에 대한 자세한 내용은 다음 문서를 참조하세요.
> - ◀ 또는 ▶ 버튼을 클릭하면 트랜잭션 목록 또는 상세 분석 화면을 전체 크기로 변경할 수 있습니다.

수집하는 **스텝(Step)**의 종류는 다음과 같습니다. 다이어그램과 텍스트 색상을 구분해 표시합니다.

- **DB Connection** 스텝 START-END

  RDB에 대한 연결에 대한 성능을 포함합니다. 스텝 정보에는 이름, 응답시간, 에러를 포함합니다.

- **SQL** 스텝 START-END

  JDBC SQL에 대한 성능을 포함합니다. 스텝 정보에는 연결 정보, SQL문, 에러가 포함되어 있습니다.

- **HTTP Call** 스텝 START-END

  외부 http 서비스 호출에 대한 성능을 포함합니다. 스텝 정보에는 url, host, port, 응답시간, 에러가 포함됩니다.

- **Message** 스텝 ADD

  트레이스를 수집하는 과정에서 비정형적인 모든 구간에 대한 이력을 수집할 때 메시지 스텝을 사용합니다.

- **Socket** 스텝 ADD

  Socket 오픈을 표현하는 스텝입니다.

- **Method** 스텝 START-END

  메소드 응답시간을 추적합니다.

- **Active Stack** 스텝 START-END

  액티브 스택에 대한 정보를 포함합니다. 별도 스레드가 생성하여 트레이스에 추가하는 방식으로 수집합니다.

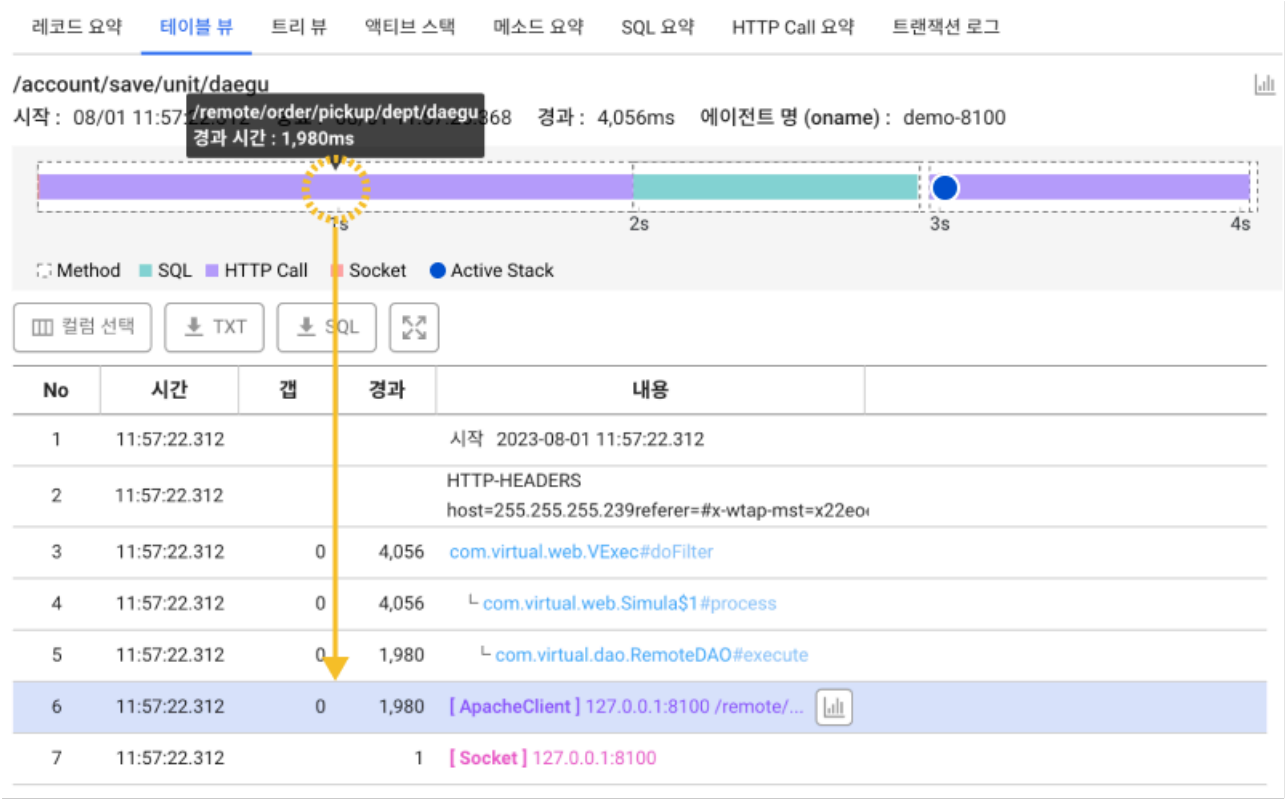
> Java 플랫폼에서는 **연계 프로젝트 관리** 메뉴를 통해 데이터베이스 프로젝트를 추가하면 연계한 프로젝트의 성능 지표를 조회할 수 있습니다. 연계 프로젝트 추가 및 분석에 대한 자세한 내용은 다음 문서를 참조하세요.

- 와탭 모니터링 서비스에서 **스텝(Step)**은 **스팬(Span)**과 같은 뜻으로 사용됩니다.
- 지난 2023년 7월 25일, 릴리스된 **트레이스 분석** 기능의 변경 사항을 확인하려면 다음 문서를 참조하세요.

테이블 뷰

테이블 뷰 탭에서는 트랜잭션의 수행 과정을 시간의 순서대로 확인할 수 있습니다.

- 트랜잭션을 수행 구간별로 분류해 다이어그램을 통해 확인할 수 있습니다. 전체 경과 시간 중 구간별 소요된 시간, 가장 오래 소요된 구간을 빠르게 파악할 수 있습니다.
- 다이어그램의 각 구간을 선택하면 해당 스텝이 위치한 테이블 목록으로 이동합니다.

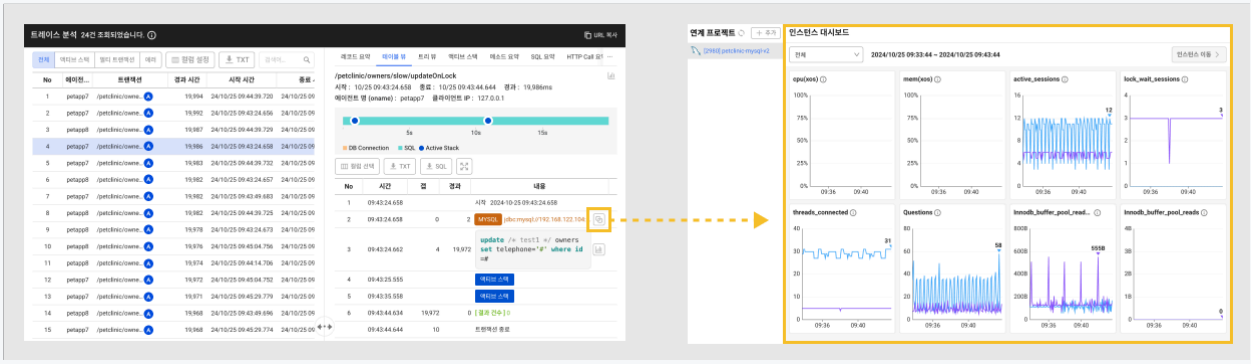


- 빨간색 원(🔵)이 위치한 영역은 액티브 스택이 수집된 순간입니다. 빨간색 원을 선택하면 **액티브 스택** 버튼이 위치한 테이블 목록으로 이동할 수 있습니다. **액티브 스택** 버튼을 선택하면 해당 구간 동안 수행된 스텝 정보를 확인할 수 있습니다.

> ◦ 트랜잭션 목록에서 Ⓐ 아이콘이 표시된 항목에서 확인할 수 있습니다.
> ◦ 액티브 스택에 대한 자세한 내용은 다음 문서를 참조하세요.

- **컬럼 선택**: 테이블 목록에 컬럼으로 메모리 누적 정보 및 CPU 누적 정보를 추가하거나 감출 수 있습니다. 컬럼 정보는 다음과 같습니다.
  - **No**: 스텝의 발생 순서
  - **시간**: 각 스텝의 시작 시각
  - **갭**: 직전 스텝의 시작 시각부터 현재 스텝으로 넘어가기까지 대기 시간, 외부 요인으로 지연될 경우 경과 시간과 차이가 있을 수 있습니다.
  - **경과**: 각 메소드 시작부터 종료까지 총 소요 시간
  - **내용**: 해당 스텝의 세부 수행 내용
- **TXT**: 트랜잭션 기본 정보 및 구간별 수행 정보를 TXT 형식의 파일로 다운로드할 수 있습니다.
- **SQL**: 트랜잭션 기본 정보 및 SQL 수행 정보를 TXT 형식의 파일로 다운로드할 수 있습니다.
  - **치환값 포함 다운로드**: 바인드 변수값을 원래의 값으로 치환하여 SQL 통계 데이터를 다운로드합니다. 보안키(`paramkey`)가 설정되어 있다면 보안키 입력창이 나타납니다. 보안키를 입력를 입력해야 다운로드할 수 있습니다. 보안키와 관련한 에이전트 설정에 대한 자세한 내용은 다음 문서를 참조하세요.
  - **치환값 미포함 다운로드**: 바인드 변수값을 치환하지 않은 상태로 SQL 통계 데이터를 다운로드합니다.
- : 트랜잭션 요약 정보와 다이어그램을 감추고 테이블 목록만 확인할 수 있습니다. 버튼을 선택하면 감춰진 정보를 다시 표시합니다. 테이블 목록이 긴 경우 이 기능을 이용하면 유용합니다.
- SQL 스텝을 선택하면 파라미터를 조회할 수 있는 **SQL** 창이 나타납니다. 자세한 내용은 다음 문서를 참조하세요.

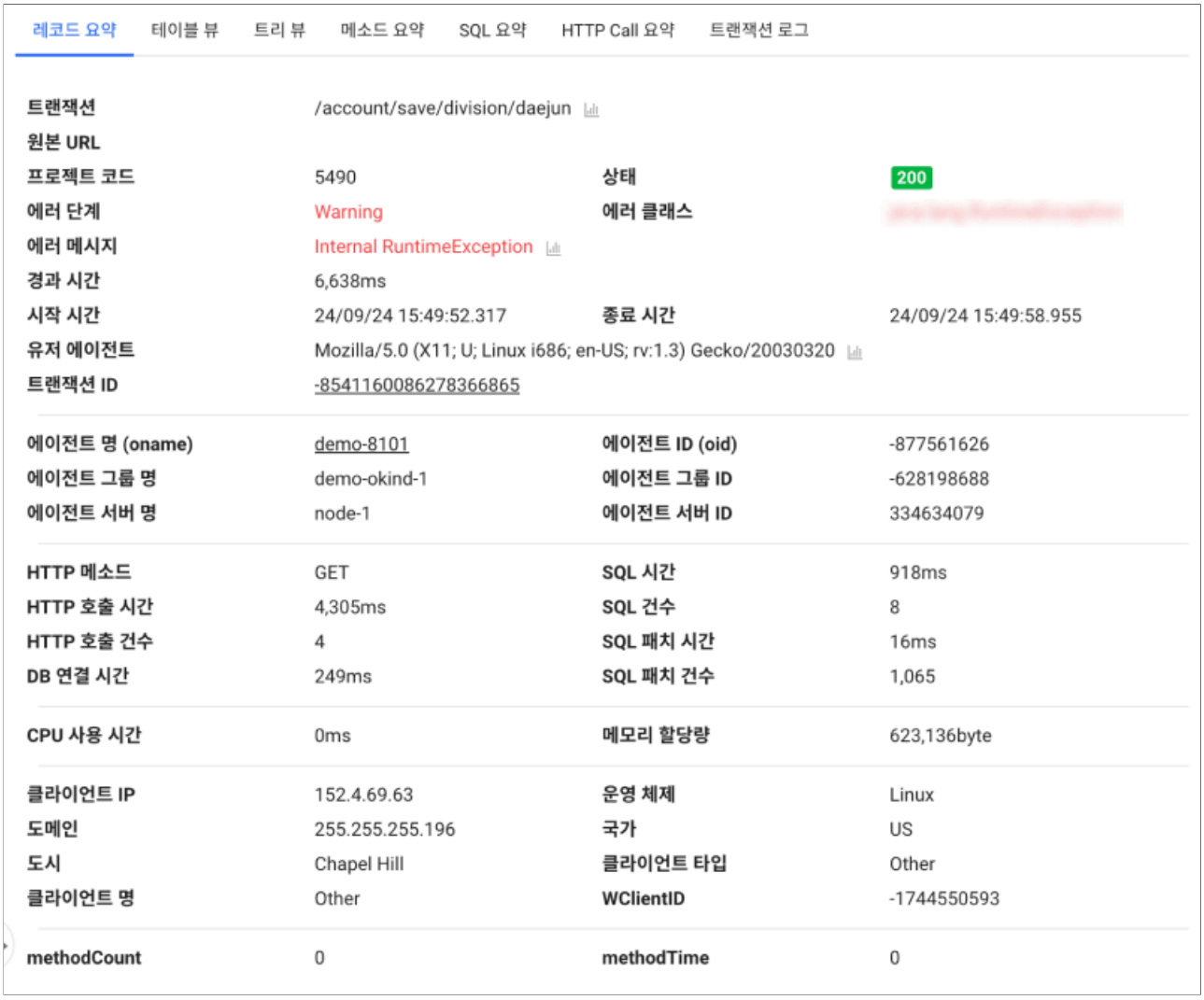
> Java 플랫폼에서는 **연계 프로젝트 관리** 메뉴를 통해 데이터베이스 프로젝트를 추가하면, **DB Connection 스텝**에서 연계한 프로젝트의 성능 지표를 조회할 수 있습니다.



연계 프로젝트 추가 및 분석에 대한 자세한 내용은 다음 문서를 참조하세요.

레코드 요약

트랜잭션의 기본 정보를 확인하려면 **레코드 요약** 탭을 선택하세요. 트랜잭션의 기본 정보 및 해당 트랜잭션이 수행된 에이전트 관련 정보, 메모리 할당 사용량, 클라이언트 관련 정보를 확인할 수 있습니다.



분류	속성	설명
트랜잭션	트랜잭션	애플리케이션에서 실행된 API 또는 URL 호출, 해당 트랜잭션의 경로입니다. 정규 표현식을 사용해 일정한 형식으로 변환합니다.
	원본 URL	실제로 호출된 URL 경로입니다. 변수를 포함하지 않은 구체적인 요청 경로입니다. 원본 URL이 있을 경우 원본 URL 표시, 원본 URL이 없을 경우 표시하지 않습니다.
	프로젝트 코드	와탭 모니터링 서비스에 등록한 프로젝트의 식별 코드입니다.
	상태	해당 트랜잭션의 HTTP 응답 상태 코드입니다.

분류	속성	설명
	에러 단계	해당 트랜잭션에서 발생한 에러 레벨입니다. Warning 또는 Critical로 표시됩니다.
	에러 클래스	해당 트랜잭션에서 발생한 에러 관련 클래스입니다.
	에러 메시지	해당 트랜잭션에서 발생한 에러 메시지입니다.
	경과 시간	트랜잭션 수행 시간입니다.
	시작 시간	트랜잭션 시작 시각입니다.
	종료 시간	트랜잭션 종료 시각입니다.
	유저 에이전트	클라이언트 관련 정보를 추출하는데 이용하는 브라우저 정보입니다.
	Referer	클라이언트가 브라우저에서 이전에 이용한 페이지 주소 또는 유입 경로입니다.
	트랜잭션 ID	해당 트랜잭션의 고유 식별자(ID)입니다.
	멀티 트랜잭션 ID	멀티 트랜잭션의 고유 식별자(ID)입니다. 다른 애플리케이션의 트랜잭션과 병렬로 실행된 경우 표시됩니다.
에이전트	에이전트 명 (oname)	에이전트 이름입니다.
	에이전트 ID (oid)	에이전트의 고유 식별자(ID)입니다.
	에이전트 그룹 명	`okind` 옵션으로 설정된 에이전트 그룹 이름입니다.
	에이전트 그룹 ID	에이전트 그룹의 고유 식별자(ID)입니다.
	에이전트 서버 명	`ondoe` 옵션으로 설정한 에이전트 서버 이름입니다.

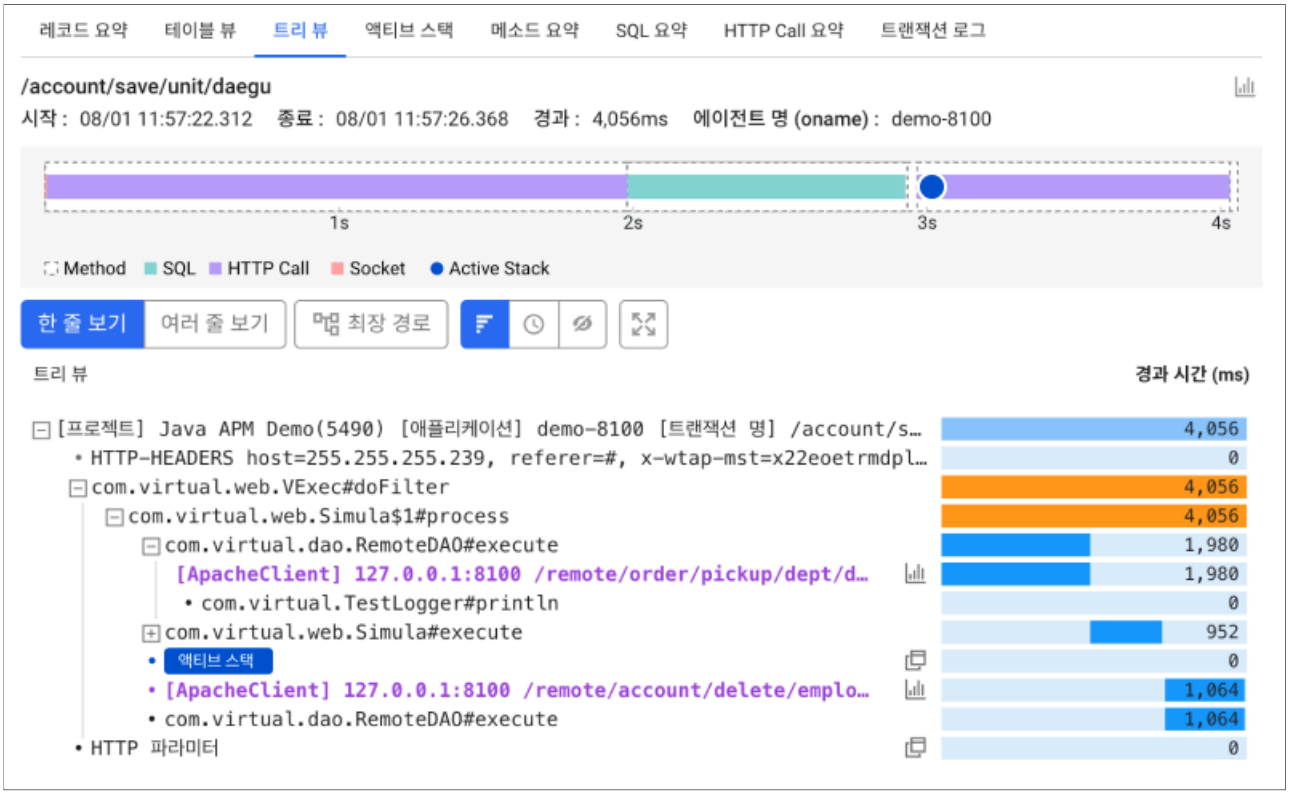
분류	속성	설명
	에이전트 서버 ID	에이전트 서버의 고유 식별자(ID)입니다.
스텝(Step)	HTTP 메소드	HTTP 메소드입니다. (GET, POST, PUT, HEAD 등)
	HTTP 호출 시간	외부 Http Call 시간입니다.
	HTTP 호출 건수	외부 HTTP Call 건수입니다.
	DB 연결 시간	데이터베이스에 연결되기까지 걸린 시간입니다.
	SQL 시간	SQL 수행 시간입니다.
	SQL 건수	SQL 수행 건수입니다.
	SQL 패치 시간	레코드를 조회하는 데 걸린 시간입니다. 중첩이 발생하거나 타 로직이 포함될 수 있습니다.
	SQL 패치 건수	SQL에서 데이터를 조회한 레코드 건수입니다.
자원(Resource)	CPU 사용 시간	트랜잭션 처리에 사용된 CPU 시간입니다.
	메모리 할당량	해당 트랜잭션에서 사용된 메모리 양입니다.
컨테이너	Pod 명	해당 트랜잭션을 처리한 Kubernetes Pod 이름입니다.
	컨테이너 ID	Pod 내에서 실행 중인 특정 컨테이너의 고유 식별자입니다.
클라이언트	클라이언트 IP	클라이언트의 IP 주소입니다.
	운영체제	브라우저가 실행되는 운영 체제 환경입니다.
	도메인	클라이언트가 접속한 IP 주소에 지정된 인터넷 주소입니다.
	국가	클라이언트의 국가 정보입니다.

분류	속성	설명
	도시	클라이언트의 국가 내 도시 정보입니다.
	클라이언트 타입	클라이언트가 이용한 브라우저 종류입니다.
	클라이언트 명	클라이언트가 이용한 기기 이름입니다.
	WClientID	클라이언트의 고유 식별자입니다.
Method	methodCount	트랜잭션에서 호출된 메소드의 수입니다.
	methodTime	메소드 실행에 소요된 시간입니다.

- 애플리케이션 종류나 설정, 스텝의 종류에 따라 수집하는 정보는 달라질 수 있습니다. **컨테이너** 항목은 애플리케이션이 컨테이너 환경에서 실행 중일 경우에만 표시됩니다.
- 에러 관련 항목은 해당 트랜잭션에서 에러가 발생할 경우 표시됩니다.
- 트랜잭션 속성 중 일부 항목은 통계 기능을 제공합니다. 버튼을 선택하면 시계열 차트를 통해 조회 시간 동안 발생한 트랜잭션 현황을 파악할 수 있는 창이 표시됩니다. 차트 내 막대 그래프를 선택하면 **애플리케이션** > **분석** > **트랜잭션 검색** 메뉴로 이동해 상세 조회할 수 있습니다.
- 와탭은 클라이언트와 관련한 정보를 기본 저장합니다. 사용자 데이터 수집과 관련한 에이전트 설정에 대한 자세한 내용은 다음 문서를 참조하세요.

트리 뷰

트랜잭션 수행 과정을 트리 형식으로 확인하려면 **트리 뷰** 탭을 선택하세요. 각 트랜잭션과 그에 속한 트레이스의 세부 정보, 트레이스의 시작 시간 및 소요 시간, 호출 관계를 확인할 수 있습니다. 다이어그램의 각 구간을 선택하면 해당 스텝이 위치한 트리뷰로 이동합니다.



- **한 줄 보기**: 각 구간 별 수행 정보에 표시된 텍스트를 한 줄로 표시해 트리 형식을 간격하게 정리할 수 있습니다.
- **여러 줄 보기**: 각 구간 별 수행 정보에 표시된 텍스트를 줄바꿈해 모두 표시합니다.
- **최장 경로**: 가장 긴 경로로 이동할 수 있습니다.
- **시간바 표시**: 경과 시간을 막대 형식의 차트로 표시합니다.
- **시간 표시**: 각 구간별 타임 스탬프, 갭, 경과 시간을 텍스트 형식으로 표시합니다.
  - 8초 이상: 초과 지연 상태로 빨간색으로 표현합니다.
  - 3초 이상 8초 미만: 지연 상태로 주황색으로 표현합니다.
  - 3초 미만: 정상 상태로 파란색으로 표현합니다.
- **시간 숨기기**: 시간 정보를 숨깁니다.
- : 트랜잭션 요약 정보와 다이어그램을 감추고 트리뷰만 확인할 수 있습니다. 버튼을 선택하면 감춰진 정보를 다시 표시합니다. 트리뷰 목록이 긴 경우 이 기능을 이용하면 유용합니다.
- : SQL 변수와 HTTP 쿼리를 조회할 수 있는 창이 나타납니다. 자세한 내용은 다음 문서를 참조하세요.

> ⓘ 시작 및 소요 시간의 경우 트랜잭션 호출 환경에 따라 발생하는 시차를 상위 트랜잭션 내 트레이스와 매핑을 통해 보정하여 표현하기 때문에 실제 수집된 시간 데이터와 차이가 발생할 수 있습니다.

액티브 스택

액티브 스택 탭을 선택하면 해당 트랜잭션에서 수집한 액티브 스택 목록을 확인할 수 있습니다.

- 왼쪽의 **액티브 스택** 목록에서 수집 시점(No, 시간)을 선택하면 오른쪽에 해당 시점의 스택 정보를 표시합니다.
- **스택**: 액티브 스택 상세 정보를 TXT 파일로 다운로드할 수 있습니다.

> ⓘ
> - 트랜잭션 목록에서 A 아이콘이 표시된 항목에서 액티브 스택을 확인할 수 있습니다.
> - 액티브 스택에 대한 자세한 내용은 다음 문서를 참조하세요.

AI 액티브 스택 분석

Beta

액티브 스택 분석 버튼을 클릭하면 수집한 액티브 스택을 AI가 자동으로 분석합니다. 진행 중인 트랜잭션의 스택 덤프를 바탕으로

성능 문제를 찾고 개선 방법을 안내합니다.

분석 결과는 다음 4개 섹션으로 구성됩니다.

- **전체 요약**: 분석한 스택 수와 전체 실행 흐름을 요약합니다.
- **공통 패턴**: 반복적으로 나타나는 실행 패턴을 표시합니다. 패턴명, 발생 횟수, 설명을 포함합니다.
- **발견된 이슈**: 찾아낸 성능 문제를 심각도(**High**, **Medium**, **Low**)와 함께 표시하며, 판단의 바탕이 된 스택 정보도 제공합니다.

심각도	기준
High	데드락, 완전 블로킹, 동일 블로킹 4회 이상 반복
Medium	I/O 대기, 2~3회 락 경합, 느린 외부 호출
Low	단일 발생 대기, 가벼운 최적화 기회

- **권장 사항**: 우선순위에 따라 구체적인 개선 방법을 안내합니다.

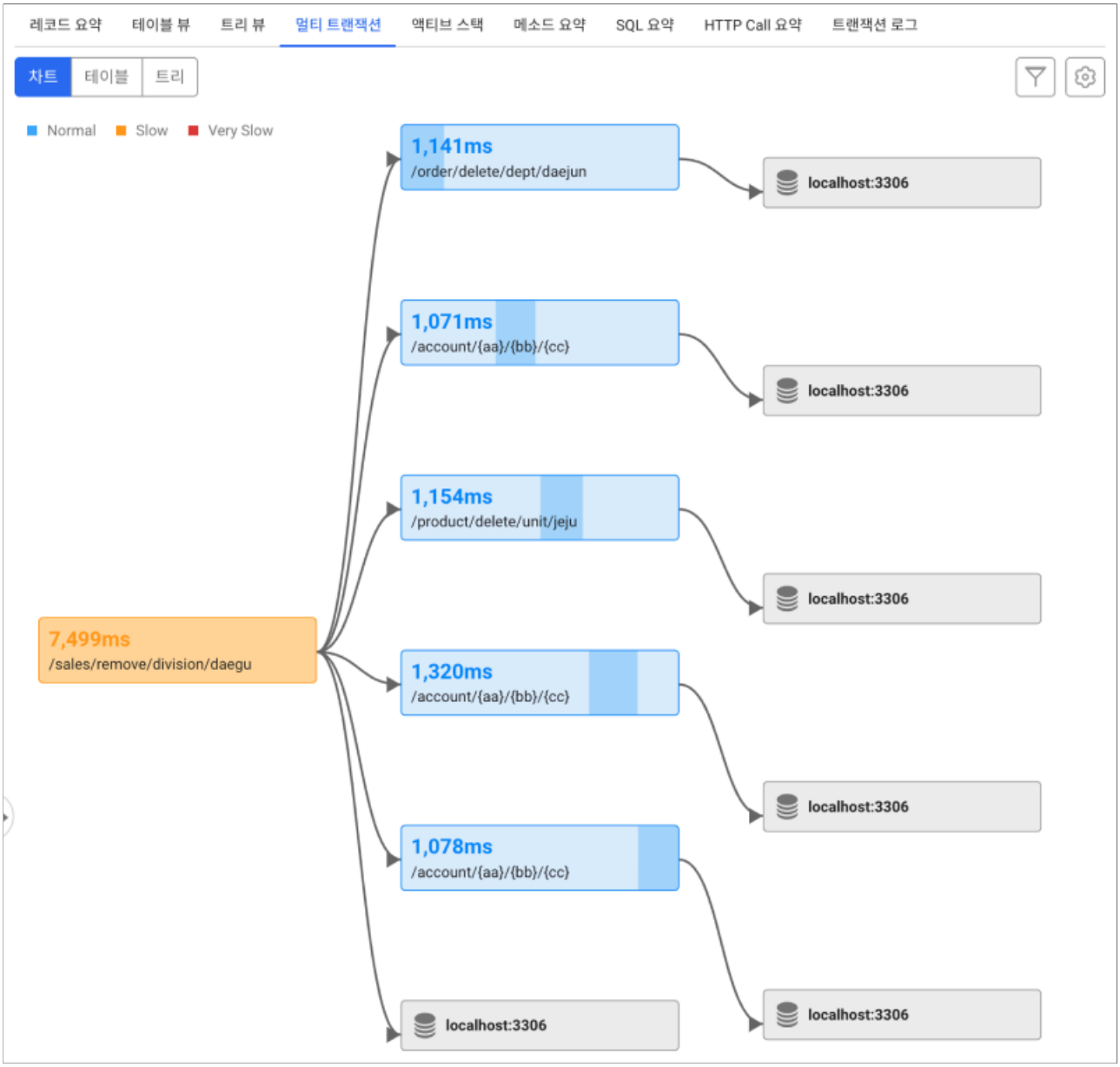
각 분석 항목에는 수집한 스택 번호(`No.N`)가 표시됩니다. 스택 번호를 클릭하면 해당 스택 원본을 확인할 수 있습니다.

> - AI가 만든 분석 결과로, 실제 상황과 다를 수 있습니다. 참고 자료로 활용하세요.
> - 프로젝트 **멤버** 권한 이상이 필요합니다.

멀티 트랜잭션

멀티 트랜잭션은 다른 에이전트나 프로젝트와의 연관된 트랜잭션을 의미합니다. **멀티 트랜잭션** 탭에서는 와탭 모니터링 서비스에 등록한 애플리케이션 간의 호출 관계를 확인할 수 있습니다.

> 멀티 트랜잭션을 추적하려면 **관리** > **에이전트 설정**에서 `mtrace_enabled` 옵션을 `true` 로 설정하세요. 에이전트 설정에 대한 자세한 내용은 다음 문서를 참조하세요.



- **차트**: 각 트랜잭션의 호출 관계를 플로우 차트 형식으로 제공합니다. 트랜잭션 노드를 선택하면 해당 트랜잭션 노드에 대한 트레이스 분석 정보를 확인할 수 있습니다. **차트** 모드에서는 마우스를 이용해 원하는 위치로 이동하거나 스크롤을 통해서 확대, 축소할 수 있습니다.
  - ⚙ **차트 뷰 설정**: 차트에 표시할 요소를 표시하거나 숨길 수 있습니다.
- **테이블**: 테이블 형식으로 멀티 트랜잭션 내에 포함된 각 트랜잭션 별 정보를 확인할 수 있습니다. ▥ 컬럼 선택 아이콘을 선택해 테이블 헤더 컬럼을 편집할 수 있습니다. 각 트랜잭션 항목을 선택하면 트레이스 분석 정보를 확인할 수 있습니다.

- **트리**: 트리 형식으로 트랜잭션 간의 호출 관계를 파악할 수 있습니다. 관련한 부가 기능은 **트리 뷰** 탭의 기능과 같습니다.
- **프로젝트 선택**: 차트에 표시될 프로젝트를 선택하거나 해제할 수 있습니다.

> - 트랜잭션 목록에서 M 아이콘이 표시된 항목에서 확인할 수 있습니다.
> - 멀티 트랜잭션에 대한 자세한 내용은 다음 문서를 참조하세요.

메소드 요약

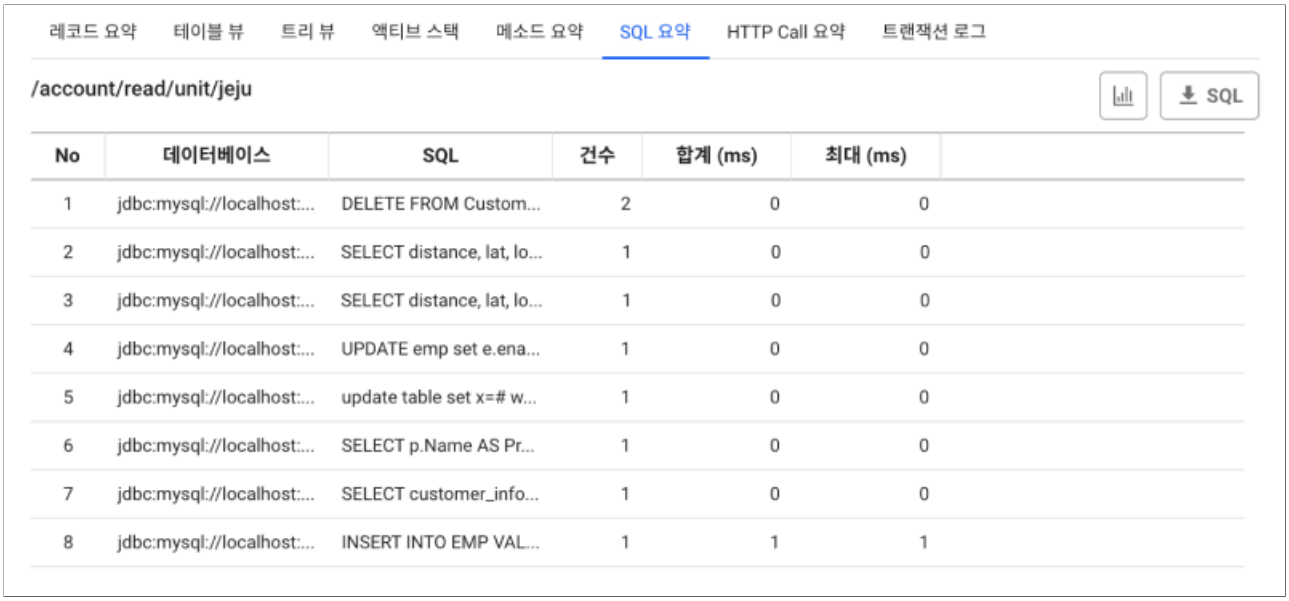
No	클래스	메소드	파라미터/리턴	건수	합계 (ms)	최대 (ms)
1	com.virtual.dao.Remot...	execute	()V	27	39,063	2,776
2	com.virtual.dao.PIITest	exec	(Ljava/sql/PreparedSta...	3	0	0
3	com.virtual.dao.InsertD...	execute2	(Ljava/lang/String;)V	2	4	3
4	com.virtual.dao.Update...	execute2	(Ljava/lang/String;)V	2	18	14
5	com.virtual.dao.Delete...	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

메소드 정보만을 확인하려면 **메소드 요약** 탭을 선택하세요. 에이전트에 추적이 설정된 메소드 이름과 소요 시간을 표시합니다. 불필요한 로직이 반복 실행되는 경우, 낮은 건수에 비해 실행 시간이 오래 걸리는 경우를 파악해 메소드 로직 개선을 위한 분석 정보로 활용할 수 있습니다.

- : **트랜잭션 통계** 정보를 확인할 수 있습니다. 차트의 그래프를 선택하면 **분석** > **트랜잭션 검색** 메뉴로 이동합니다. 선택한 시간대의 트랜잭션 목록을 확인할 수 있습니다.
- **메소드**: 메소드 관련 상세 정보를 TXT 파일로 다운로드할 수 있습니다.

> ⓘ 메소드(method)와 관련한 에이전트 설정 옵션에 대한 자세한 내용은 다음 문서를 참조하세요. 이용 중인 제품에 따라 메소드 추적 지원 여부는 다를 수 있습니다.

SQL 요약



No	데이터베이스	SQL	건수	합계 (ms)	최대 (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

SQL 문에 대한 정보를 확인하려면 **SQL 요약** 탭을 선택하세요. 불필요하게 반복 실행되는 경우, 낮은 건수에 비해 실행 시간이 오래 걸리는 경우를 파악해 쿼리 성능 개선을 위한 분석 정보로 활용할 수 있습니다.

- : **트랜잭션 통계** 정보를 확인할 수 있습니다. 차트의 그래프를 선택하면 **분석** > **트랜잭션 검색** 메뉴로 이동합니다. 선택한 시간대의 트랜잭션 목록을 확인할 수 있습니다.
- **SQL**: SQL 관련 상세 정보를 TXT 파일로 다운로드할 수 있습니다.

HTTP Call 요약

No	원격 서버	URL	건수	합계 (ms)	최대 (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

HTTP 호출의 호출 건수, 합계 시간, 평균 시간 등을 확인하려면 **HTTP Call 요약** 탭을 선택하세요. 불필요한 외부 호출이 반복 실행되는 경우, 낮은 건수에 비해 실행 시간이 오래 걸리는 경우를 파악해 트랜잭션 지연 요인이 외부인지 내부인지를 파악하는 분석 정보로 활용할 수 있습니다.

- : **트랜잭션 통계** 정보를 확인할 수 있습니다. 차트의 그래프를 선택하면 **분석** > **트랜잭션 검색** 메뉴로 이동합니다. 선택한 시간대의 트랜잭션 목록을 확인할 수 있습니다.
- **HTTP**: HTTP Call 관련 상세 정보를 TXT 파일로 다운로드할 수 있습니다.

트랜잭션 로그

트랜잭션과 관련한 로그 정보를 확인하려면 **트랜잭션 로그** 탭을 선택하세요.

- : **트랜잭션 통계** 정보를 확인할 수 있습니다. 차트의 그래프를 선택하면 **분석** > **트랜잭션 검색** 메뉴로 이동합니다. 선택한 시간대의 트랜잭션 목록을 확인할 수 있습니다.
- **로그 검색기**: 로그 검색으로 이동해 원하는 로그 데이터를 쉽게 조회할 수 있습니다.
- **키워드 검색**: 키워드 검색란에 검색하려는 텍스트를 입력한 다음 엔터 키를 입력하거나 버튼을 선택하세요. 입력한 텍스트와 일치하는 키워드를 하이라이트 표시합니다.
- **키워드 하이라이트**: 키워드와 색상을 설정하면 자동으로 로그 목록에서 키워드와 일치하는 텍스트를 하이라이트 표시합니다. 자세한 내용은 다음 문서를 참조하세요.
- **컬럼 설정**: 로그 목록 테이블에 표시되는 컬럼을 추가하거나 순서를 변경할 수 있습니다. 자세한 내용은 다음 문서를 참조하세요.
- **로그 표시 상세 설정**: 로그 목록에 표시되는 내용을 설정할 수 있습니다. 자세한 내용은 다음 문서를 참조하세요.

> - **트랜잭션 로그** 탭을 활성화하려면 로그와 관련한 에이전트 설정 옵션을 적용해야 합니다. 로그 설정에 대한 자세한 내용은 다음 문서를 참조하세요.
> - **트랜잭션 로그** 탭은 **로그 조회** 권한을 가진 멤버만 진입할 수 있습니다. 멤버 권한에 대한 자세한 내용은 다음 문서를 참조하세요.

키워드 하이라이트 설정하기

1. 하이라이트 표시할 키워드를 추가하려면 버튼을 선택하세요.

2. 원하는 키워드를 입력하세요.
3. 색상 아이콘을 선택한 다음 원하는 색상을 선택하세요.
4. 엔터를 입력하세요. 추가한 키워드가 목록에 생성됩니다.

키워드 하이라이트 삭제하기

하이라이트 목록에서 이용하지 않는 항목을 삭제하려면 — 버튼을 선택하세요.

테이블 컬럼 표시하거나 숨기기

1. 테이블 오른쪽 위에 ▥ 버튼을 선택하세요.
2. **컬럼 설정** 창이 나타나면 왼쪽 목록에서 원하는 항목을 선택하거나 선택을 해제하세요.

   오른쪽 목록에서 × 버튼을 선택해 컬럼을 숨길 수도 있습니다.
3. 모든 설정을 완료한 다음 **확인** 버튼을 선택하세요. 설정한 내용이 테이블에 적용됩니다.

테이블 컬럼 순서 바꾸기

1. 테이블 오른쪽 위에 ▥ 버튼을 선택하세요.
2. **컬럼 설정** 창이 나타나면 오른쪽 목록에서 원하는 항목을 드래그해서 순서를 변경하세요.
3. 모든 설정을 완료한 다음 **확인** 버튼을 선택하세요. 설정한 내용이 테이블에 적용됩니다.

테이블 콘텐츠 표시 설정

1. 테이블 오른쪽 위에 ⚙ 버튼을 선택하세요.
2. **로그 표시 상세 설정** 창이 나타나면 원하는 옵션을 설정하세요.
   - **content** 또는 **태그** 옵션을 선택해 로그의 콘텐츠 표시 여부를 선택하세요.
   - **태그 관리**: 태그는 추가한 순서대로 로그에 표시됩니다. 태그의 순서는 드래그로 변경할 수 있습니다. 비활성화된 태그는 로그에서 보이지 않습니다.
3. 모든 설정을 완료한 다음 ✕ 버튼을 클릭하세요. 설정한 내용이 테이블에 적용됩니다.

> ⓘ **로그 표시 상세 설정**에 대한 자세한 내용은 다음 문서를 참조하세요.

부가 기능

HTTP 파라미터 조회

테이블 뷰 탭에서 해당 트랜잭션의 HTTP 파라미터를 조회할 수 있습니다.

1. 페이지 아래로 스크롤해 트랜잭션 수행의 가장 마지막 단계로 이동하세요.
2. 🔒 **HTTP 파라미터** 항목을 선택하세요.
3. **HTTP-PARAMETERS** 창이 나타나면 **비밀번호** 버튼을 선택하세요.
4. 설정한 Param Key를 입력하세요. 가려진 매개 변수를 확인할 수 있습니다.

> ⓘ
> - HTTP 파라미터와 관련한 에이전트 설정은 다음 문서를 참조하세요.
> - **비밀번호**: 복호화된 파라미터 값을 확인할 수 있습니다. 비밀번호는 `WHATAP_HOME` /security.conf (paramkey.txt) 파일 내 6자리 문자열입니다. 다른 문자열로 변경 가능합니다. SQL 변수 조회, HTTP 쿼리 조회, Thread 중지에 필요합니다.
> - 에이전트 버전에 따라 비밀번호 설정 파일이 다를 수 있습니다. security.conf 파일 또는 paramkey.txt 파일 내 문자열을 변경하세요.

SQL 파라미터 조회

테이블 뷰 탭에서 SQL 스텝을 선택하거나 **트리 뷰** 탭에서 버튼을 선택하세요. 파라미터를 조회할 수 있는 **SQL** 창이 나타납니다.

- **비밀번호**: 복호화된 파라미터 값을 확인할 수 있습니다. 비밀번호는 `WHATAP_HOME` /paramkey.txt 파일 내 6자리 문자열입니다. 다른 문자열로 변경 가능합니다.

  > paramkey.txt 내 키는 SQL 변수 조회, HTTP 쿼리 조회, Thread 중지에 필요합니다.

- **SQL 포멧팅**: SQL 문장을 들여쓰기 및 포맷팅하여 가독성을 높일 수 있습니다.
- : SQL 문장을 클립보드에 복사할 수 있습니다.
- **통계**: **통계** > **SQL** 메뉴로 이동합니다.

SQL 변수와 HTTP 쿼리를 조회하려면 다음 옵션을 에이전트 설정에 추가하세요. DB, SQL과 관련한 에이전트 설정에 대한 자세한 내용은 다음 문서를 참조하세요.

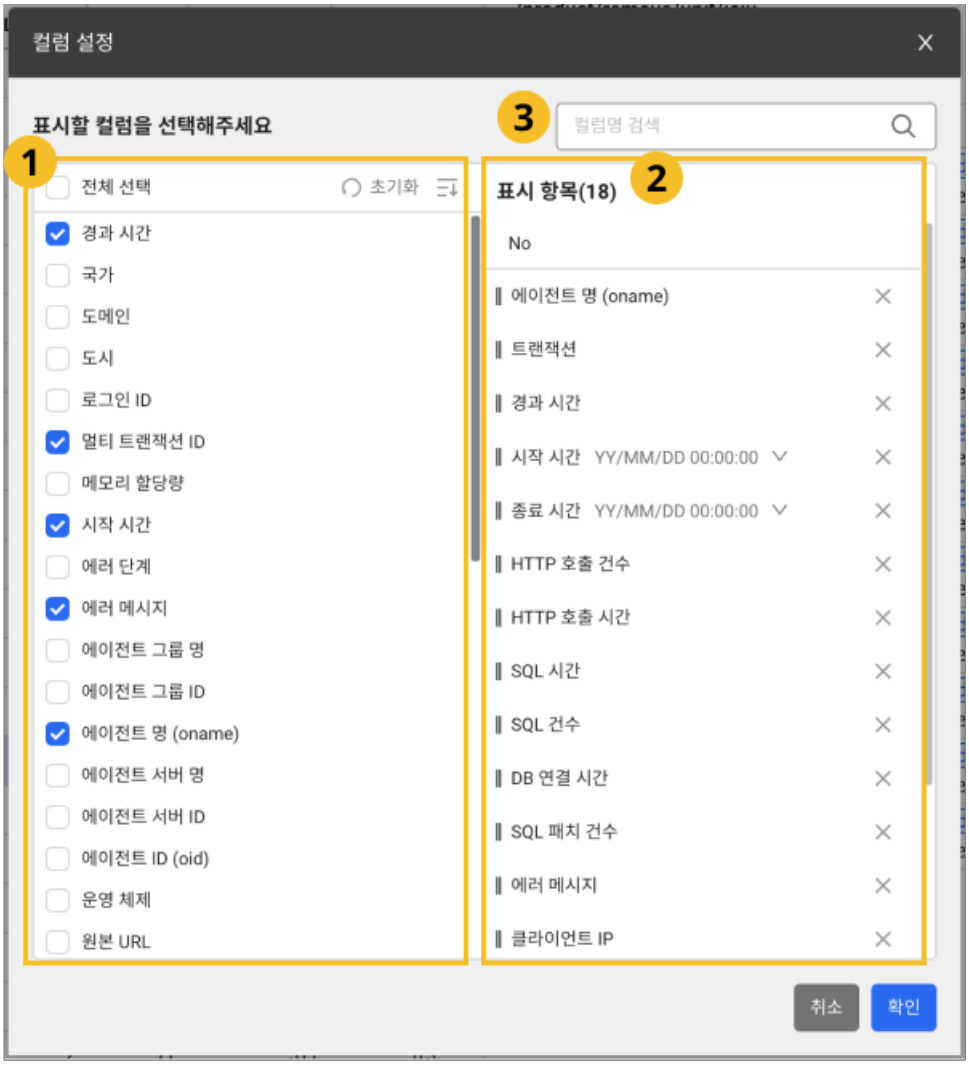
whatap.conf

```
# SQL 파라미터 조회 옵션: 옵션이 적용되면 SQL 파라미터를 암호화하여 수집합니다.
profile_sql_param_enabled=true
```

```
# HTTP 파라미터 조회 옵션: 옵션이 적용되면 HTTP 쿼리 파라미터를 암호화하여 수집합니다.
profile_http_parameter_enabled=true
```

테이블 컬럼 설정하기

트랜잭션 목록에서 테이블 헤더 컬럼을 감추거나 원하는 항목을 추가할 수 있습니다. 컬럼 순서를 변경할 수도 있습니다. **컬럼 설정** 버튼을 선택하세요.



- 설정을 완료한 다음에는 **확인** 버튼을 선택해야 설정 사항이 테이블에 반영됩니다.
- 3 검색란에 텍스트를 입력해 원하는 컬럼 항목을 검색할 수 있습니다. 입력한 텍스트와 매칭되는 컬럼 항목만 표시됩니다.

컬럼 추가하기

1 목록에서 테이블 헤더 컬럼으로 추가할 항목을 선택하세요. 모든 항목을 추가하려면 **전체 선택**을 선택하세요.

컬럼 삭제하기

1 목록에서 삭제할 컬럼 항목의 체크 박스를 선택 해제하세요. 또는 2 목록에서 삭제할 항목의 오른쪽에 ✕ 버튼을 선택하세요.

컬럼 순서 변경하기

2 목록에 순서를 변경할 항목을 드래그해서 원하는 위치로 이동할 수 있습니다.

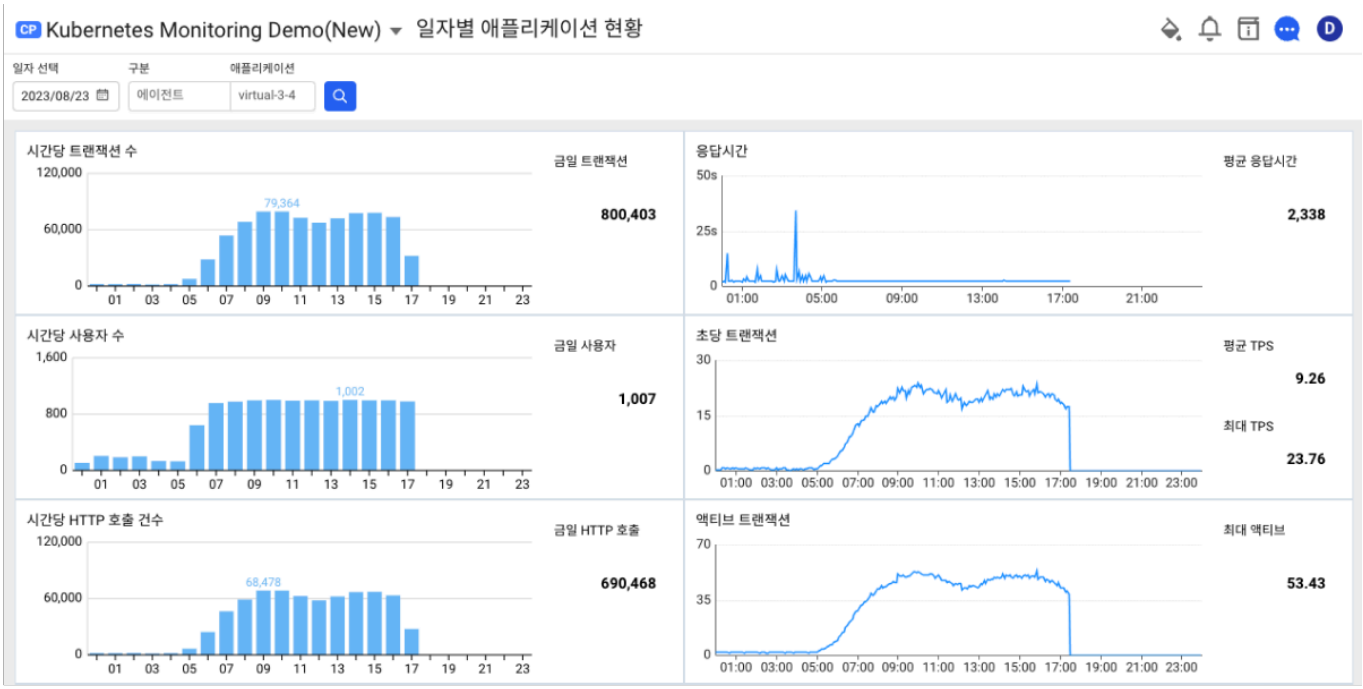
설정 사항 초기화하기

변경 사항은 모두 취소하고 초기화하려면 ↻ **초기화** 버튼을 선택하세요.

조회 데이터 다운로드

테이블에 출력된 데이터를 txt 파일 형식으로 다운로드하려면 ⬇ **TXT** 버튼을 선택하세요.

일자별 애플리케이션 현황



일자별 애플리케이션 상태를 확인할 수 있습니다. 검색 조건은 시작일, 구분, 애플리케이션으로 구분 조건은 다음과 같습니다.

분류	설명
에이전트	컨테이너 내부에 설치된 애플리케이션 에이전트 이름
에이전트 종류	에이전트가 그룹화된 에이전트 종류 (`OKIND`)
에이전트 서버	에이전트가 실행 중인 쿠버네티스 노드 이름

제공하는 차트는 다음과 같습니다.

- **시간당 트랜잭션 수**
- **시간당 사용자 수**
- **시간당 HTTP 호출 건수**

- **시간당 SQL 건수**
- **시간당 SQL 패치 건수**
- **응답시간**
- **초당 트랜잭션**
- **액티브 트랜잭션**
- **5분 합계 에러 건수**
- **에러율**

트랜잭션 검색

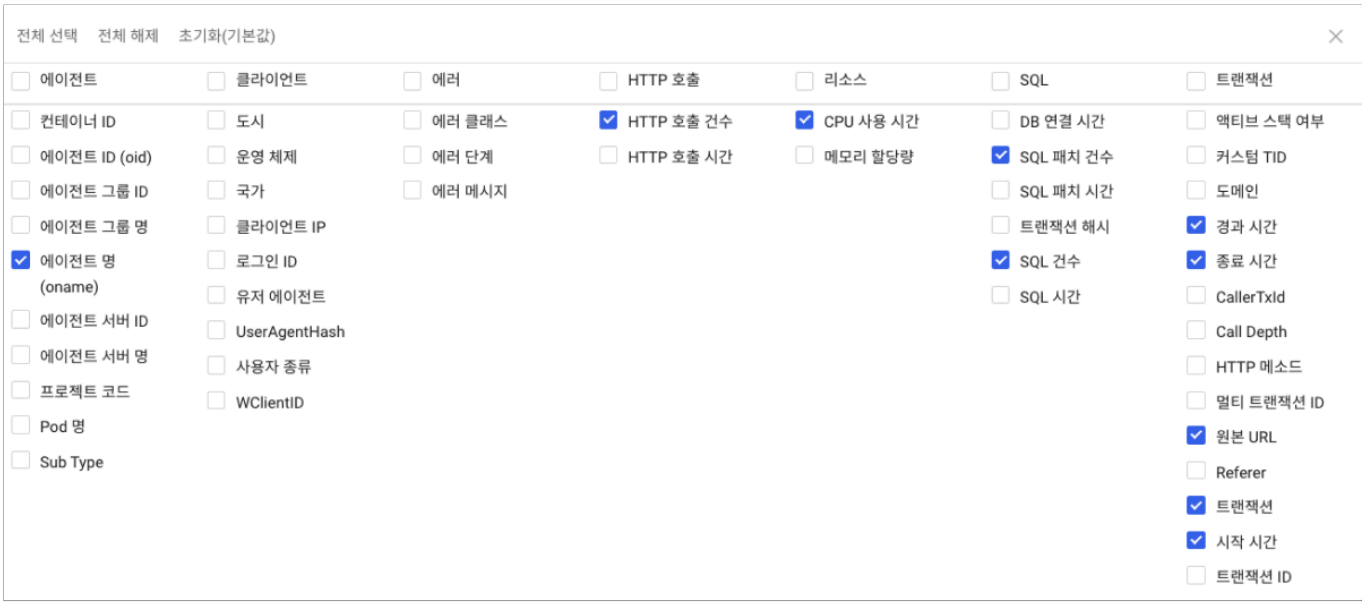
트랜잭션 검색 메뉴를 통해 특정 기간 동안 각 트랜잭션의 성능 속성에 대한 통계와 실행 상태를 검색할 수 있습니다. 모든 트랜잭션을 확인하고 정상 및 에러 상태를 빠르게 확인할 수 있습니다. 제공되는 정보를 통해 어떤 트랜잭션의 사용량이 많고 리소스를 많이 소비하는지 알 수 있습니다. 트랜잭션의 속성에 따라 다양하게 필터링할 수 있어 특정 조건의 트랜잭션을 찾는데 유용합니다.

1 영역에서 조회 시간을 설정 후 **컨테이너** 옵션에서 조회 대상을 선택하세요. 버튼을 선택하면 검색 결과가 테이블에 표시됩니다. 2 영역의 결과 목록에서 원하는 항목을 선택하면 **트랜잭션 정보** 창이 나타납니다. **트랜잭션 정보** 창을 활용한 트랜잭션 트레이스 상세 분석에 관한 자세한 내용은 다음 문서를 참조하세요.

- 에러 상태의 트랜잭션만 검색하려면 **에러만 보기**를 클릭한 다음 버튼을 선택하세요.
- 검색 조건에 따라 응답 지연이 발생할 수 있기 때문에 검색 시간을 가급적 짧게 설정할 것을 권고합니다.

컬럼 선택

2 영역의 **컬럼 선택** 버튼 클릭 시 다음과 같이 컬럼 선택 창이 나타납니다. 원하는 컬럼을 적절하게 선택하세요.

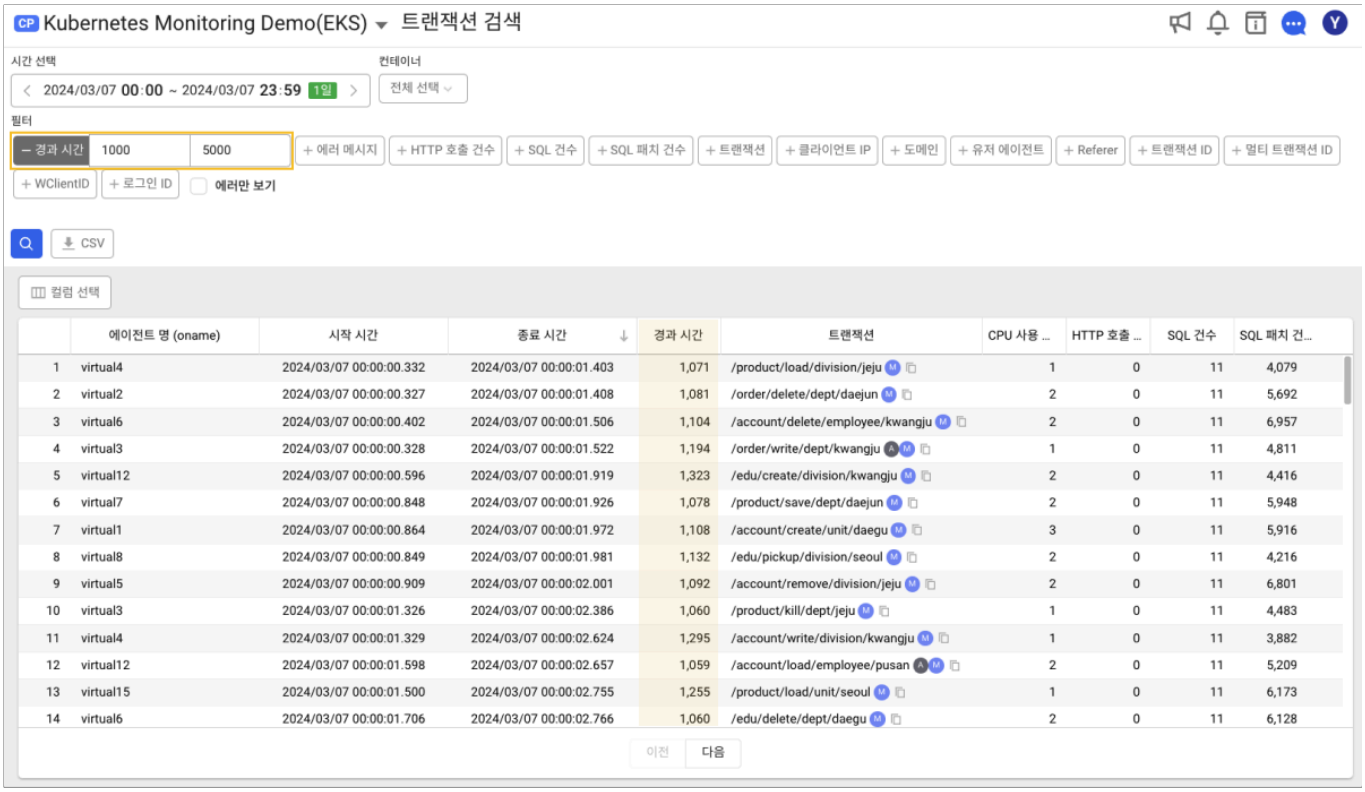


필터 적용

1 영역에서 필터를 활용해 상세 검색이 가능합니다. 다양한 검색 조건에 의한 데이터 탐색 방식을 지원합니다. 필터 항목 역시 여러 개를 적용할 수 있습니다. 원하는 필터를 지정한 다음 버튼을 선택하세요.

- 일부 필터 항목은 제품에 따라 지원하지 않을 수 있습니다.
- 다중 필터 적용 시 **멀티 트랜잭션 ID** 필터를 적용하여 검색할 경우 **트랜잭션 ID** 필터만 추가 적용할 수 있습니다.

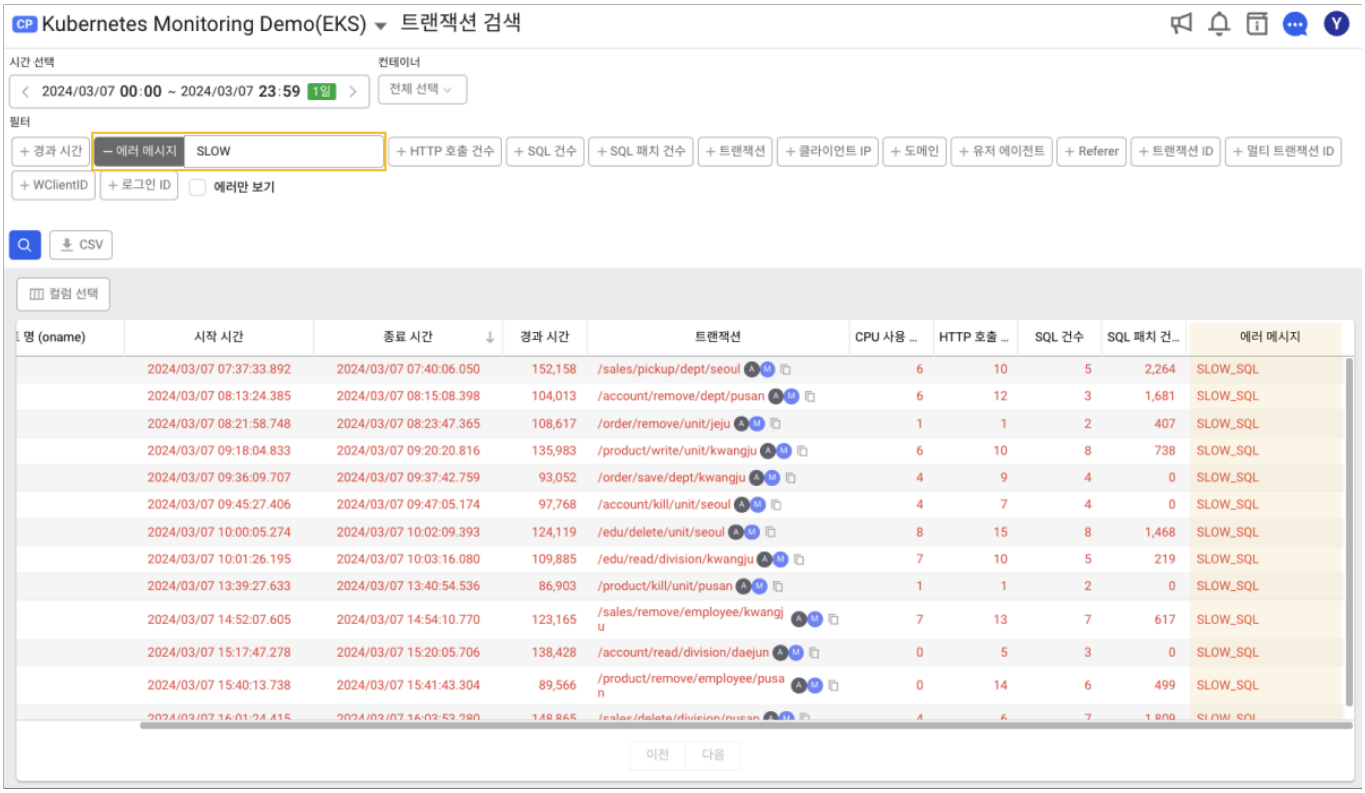
경과 시간



	에이전트 명 (oname)	시작 시간	종료 시간	경과 시간	트랜잭션	CPU 사용 ...	HTTP 호출 ...	SQL 건수	SQL 패치 건...
1	virtual4	2024/03/07 00:00:00.332	2024/03/07 00:00:01.403	1,071	/product/load/division/jeju	1	0	11	4,079
2	virtual2	2024/03/07 00:00:00.327	2024/03/07 00:00:01.408	1,081	/order/delete/dept/daejun	2	0	11	5,692
3	virtual6	2024/03/07 00:00:00.402	2024/03/07 00:00:01.506	1,104	/account/delete/employee/kwangju	2	0	11	6,957
4	virtual3	2024/03/07 00:00:00.328	2024/03/07 00:00:01.522	1,194	/order/write/dept/kwangju	1	0	11	4,811
5	virtual12	2024/03/07 00:00:00.596	2024/03/07 00:00:01.919	1,323	/edu/create/division/kwangju	2	0	11	4,416
6	virtual7	2024/03/07 00:00:00.848	2024/03/07 00:00:01.926	1,078	/product/save/dept/daejun	2	0	11	5,948
7	virtual1	2024/03/07 00:00:00.864	2024/03/07 00:00:01.972	1,108	/account/create/unit/daegu	3	0	11	5,916
8	virtual8	2024/03/07 00:00:00.849	2024/03/07 00:00:01.981	1,132	/edu/pickup/division/seoul	2	0	11	4,216
9	virtual5	2024/03/07 00:00:00.909	2024/03/07 00:00:02.001	1,092	/account/remove/division/jeju	2	0	11	6,801
10	virtual3	2024/03/07 00:00:01.326	2024/03/07 00:00:02.386	1,060	/product/kill/dept/jeju	1	0	11	4,483
11	virtual4	2024/03/07 00:00:01.329	2024/03/07 00:00:02.624	1,295	/account/write/division/kwangju	1	0	11	3,882
12	virtual12	2024/03/07 00:00:01.598	2024/03/07 00:00:02.657	1,059	/account/load/employee/pusan	2	0	11	5,209
13	virtual15	2024/03/07 00:00:01.500	2024/03/07 00:00:02.755	1,255	/product/load/unit/seoul	1	0	11	6,173
14	virtual6	2024/03/07 00:00:01.706	2024/03/07 00:00:02.766	1,060	/edu/delete/dept/daegu	2	0	11	6,128

트랜잭션 응답시간 기준으로 검색합니다. 기준 이상과 미만의 2개 값을 입력해 검색합니다.

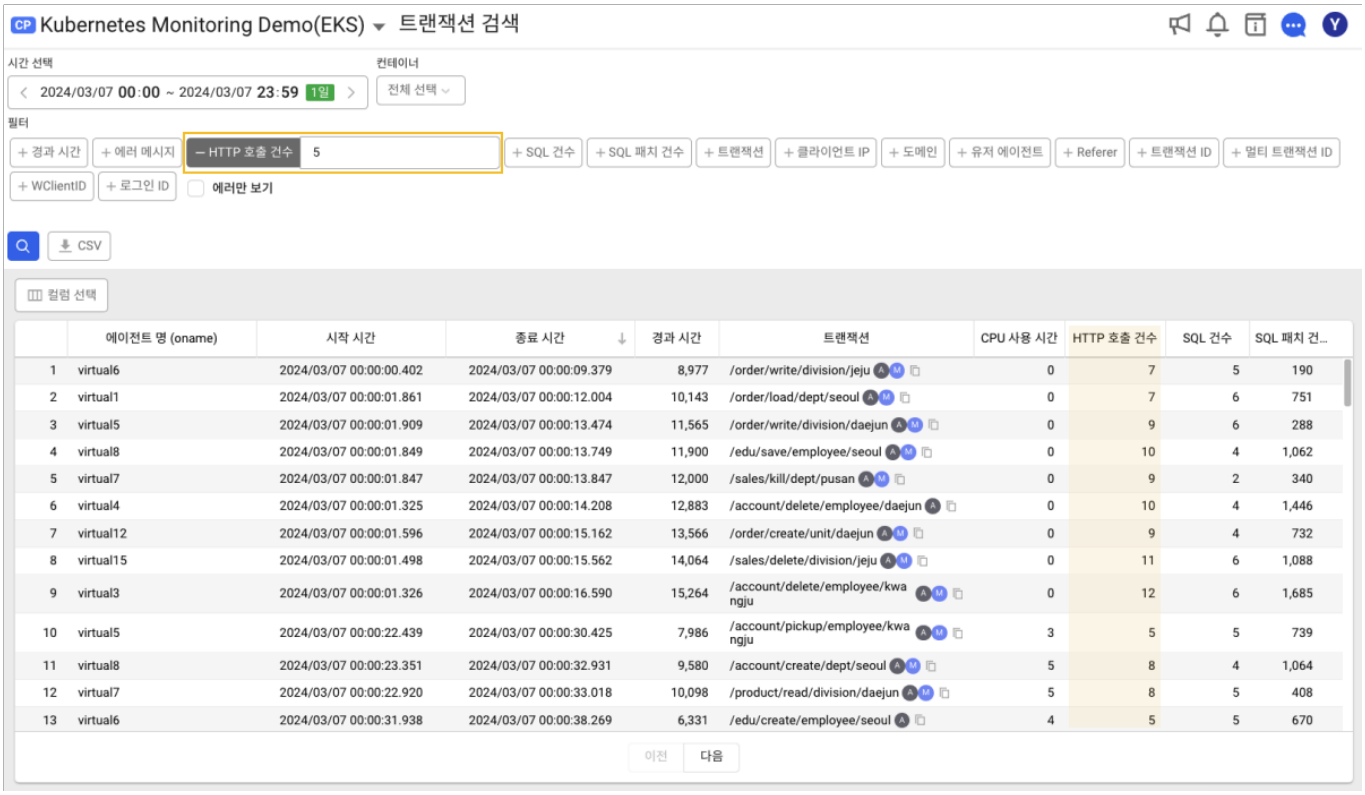
에러 메시지



트랜잭션 에러 메시지를 기반으로 검색합니다. 디폴트 상태에서 에러 관련 컬럼이 보이지 않을 수 있습니다. 컬럼을 선택해 확인하세요.

> ⓘ 에러 상태의 트랜잭션은 빨간색으로 표시됩니다.

HTTP 호출 건수

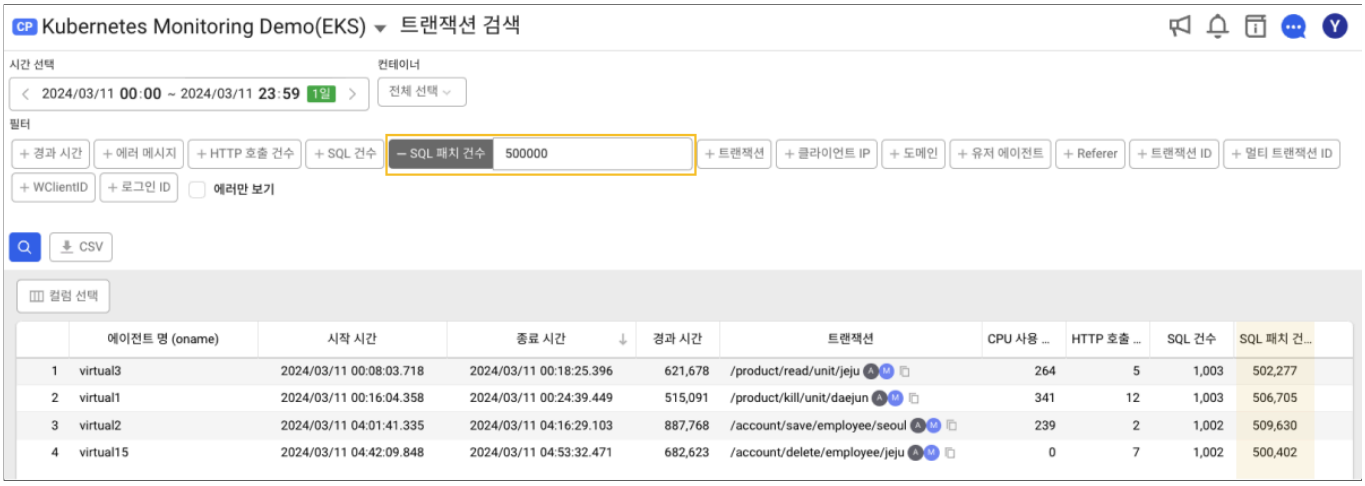


트랜잭션 중 외부 HTTP 서비스를 호출한 건수가 입력값 이상에 해당하는 트랜잭션을 검색합니다. 예시 이미지의 경우 **5**건 이상을 조회합니다.

SQL 건수

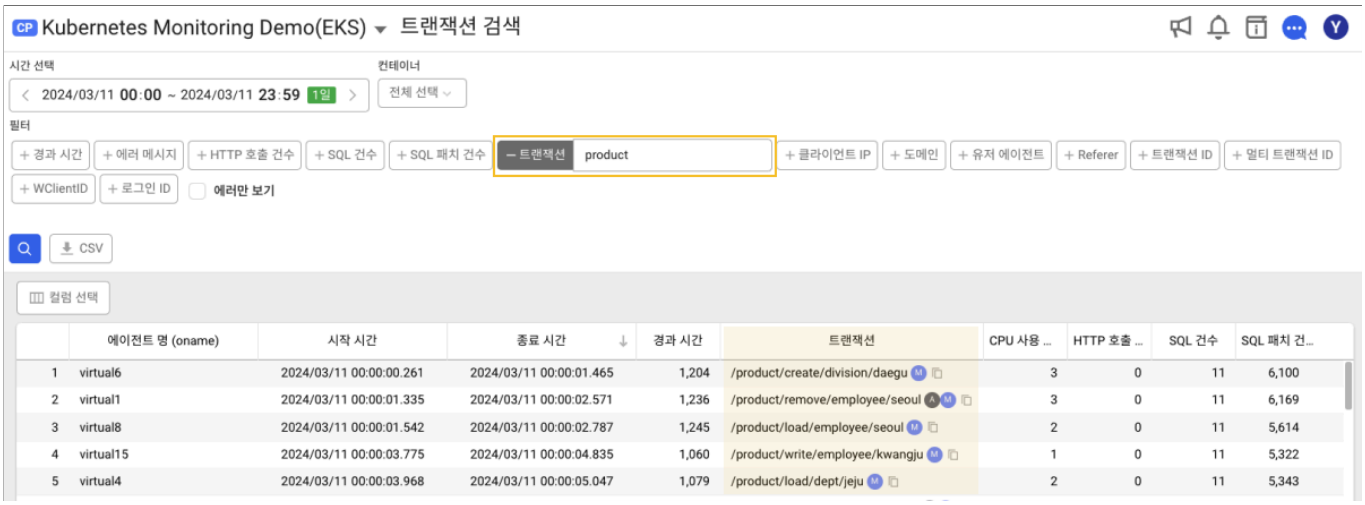
트랜잭션별 SQL 수행 건수가 설정한 입력값 이상에 해당하는 트랜잭션을 검색합니다. 예외적으로 과도한 수의 SQL을 호출하는 트랜잭션을 검색할 때 이용합니다.

SQL 패치 건수



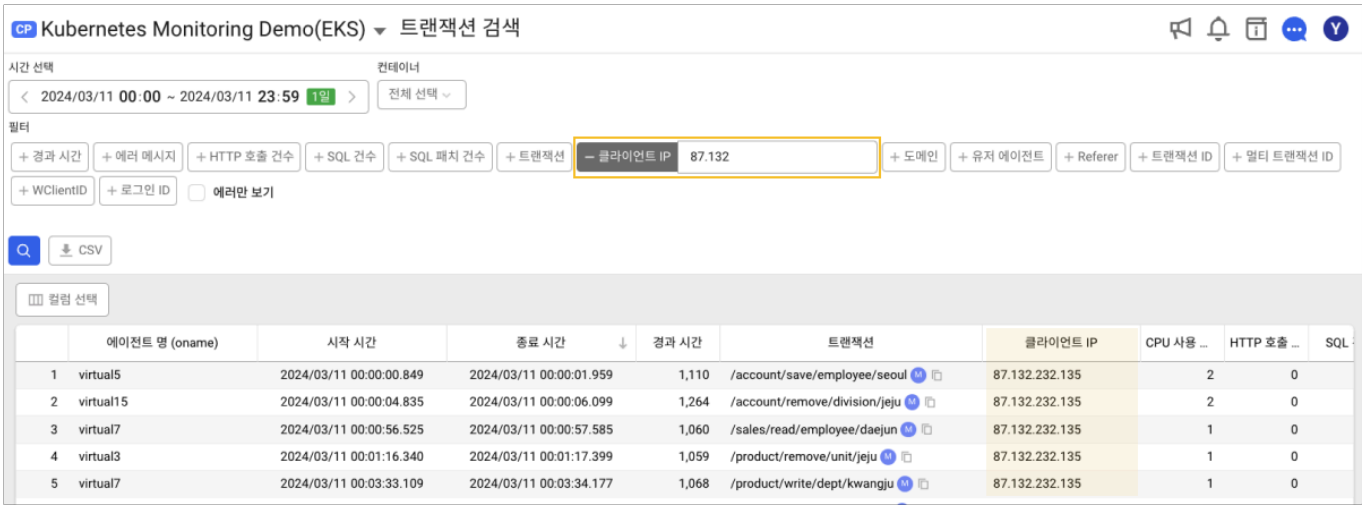
SQL 수행 후 DB 패치 건수를 기준으로 입력값 이상에 해당하는 트랜잭션을 검색합니다. 과도한 데이터 패치를 수행하는 트랜잭션을 검색할 때 이용합니다.

트랜잭션



트랜잭션 기준으로 검색합니다. 입력한 검색 문자열은 포함 여부(부분 일치)로 비교 검색합니다.

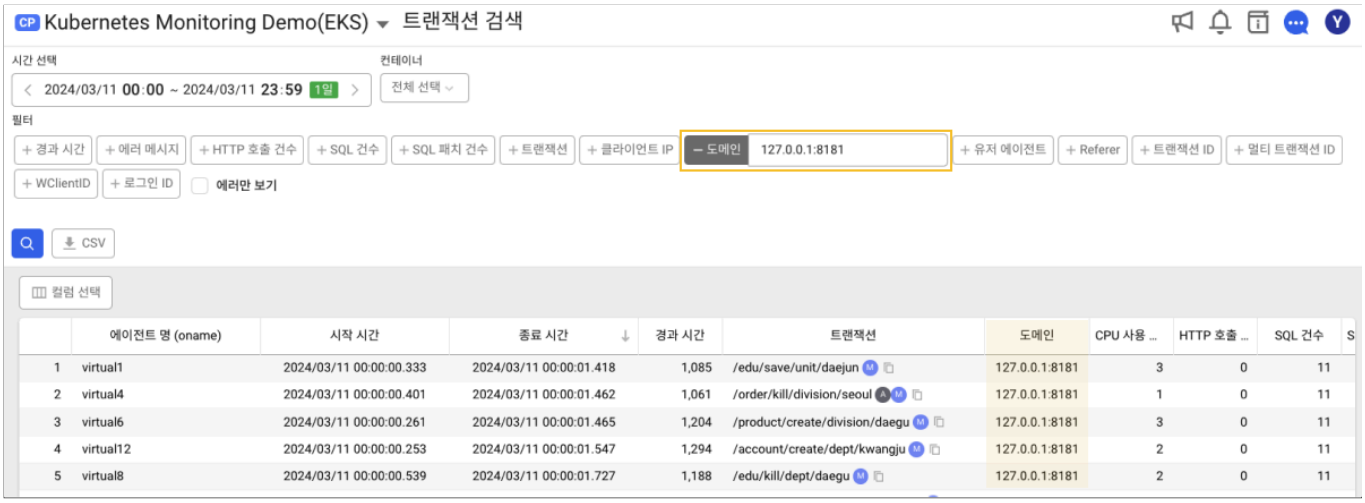
클라이언트 IP



	에이전트 명 (oname)	시작 시간	종료 시간	경과 시간	트랜잭션	클라이언트 IP	CPU 사용 ...	HTTP 호출 ...	SQL
1	virtual5	2024/03/11 00:00:00.849	2024/03/11 00:00:01.959	1,110	/account/save/employee/seoul	87.132.232.135	2	0	
2	virtual15	2024/03/11 00:00:04.835	2024/03/11 00:00:06.099	1,264	/account/remove/division/jeju	87.132.232.135	2	0	
3	virtual7	2024/03/11 00:00:56.525	2024/03/11 00:00:57.585	1,060	/sales/read/employee/daejun	87.132.232.135	1	0	
4	virtual3	2024/03/11 00:01:16.340	2024/03/11 00:01:17.399	1,059	/product/remove/unit/jeju	87.132.232.135	1	0	
5	virtual7	2024/03/11 00:03:33.109	2024/03/11 00:03:34.177	1,068	/product/write/dept/kwangju	87.132.232.135	1	0	

클라이언트 IP 기반으로 트랜잭션을 검색합니다. 단 IP 문자열은 PREFIX 비교(전방 일치)로 검색합니다.

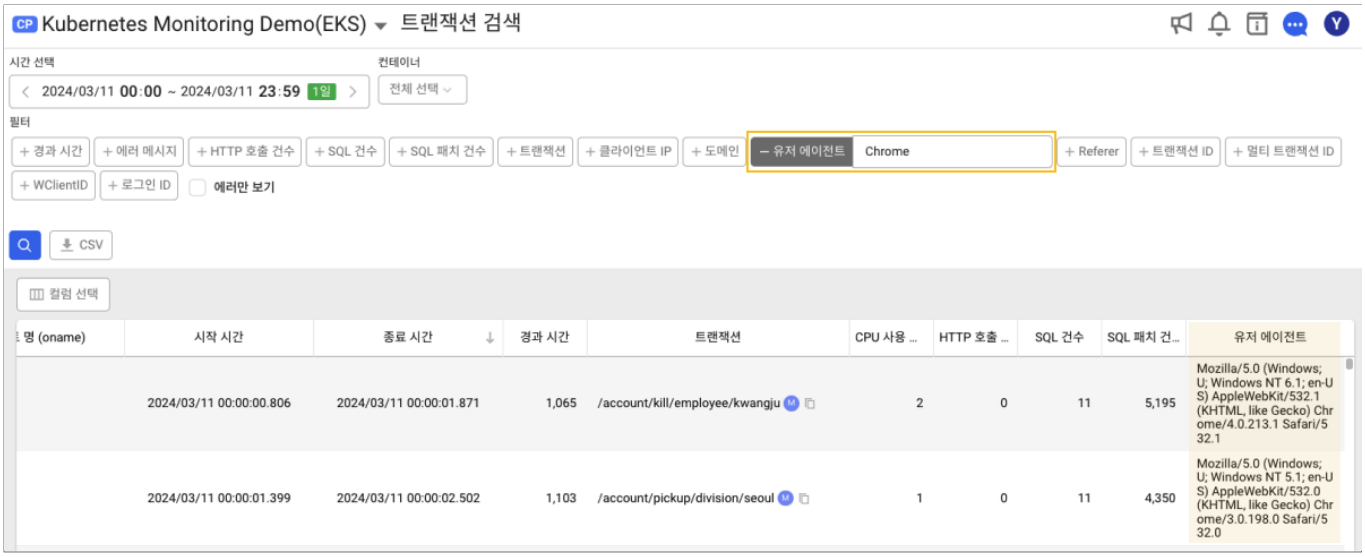
도메인



	에이전트 명 (oname)	시작 시간	종료 시간	경과 시간	트랜잭션	도메인	CPU 사용 ...	HTTP 호출 ...	SQL 건수
1	virtual1	2024/03/11 00:00:00.333	2024/03/11 00:00:01.418	1,085	/edu/save/unit/daejun	127.0.0.1:8181	3	0	11
2	virtual4	2024/03/11 00:00:00.401	2024/03/11 00:00:01.462	1,061	/order/kill/division/seoul	127.0.0.1:8181	1	0	11
3	virtual6	2024/03/11 00:00:00.261	2024/03/11 00:00:01.465	1,204	/product/create/division/daegu	127.0.0.1:8181	3	0	11
4	virtual12	2024/03/11 00:00:00.253	2024/03/11 00:00:01.547	1,294	/account/create/dept/kwangju	127.0.0.1:8181	2	0	11
5	virtual8	2024/03/11 00:00:00.539	2024/03/11 00:00:01.727	1,188	/edu/kill/dept/daegu	127.0.0.1:8181	2	0	11

HTTP 서비스 도메인(host:port) 기반으로 트랜잭션을 검색합니다. 검색 조건은 도메인이 완전히 일치(전체 일치)해야 합니다.

유저 에이전트



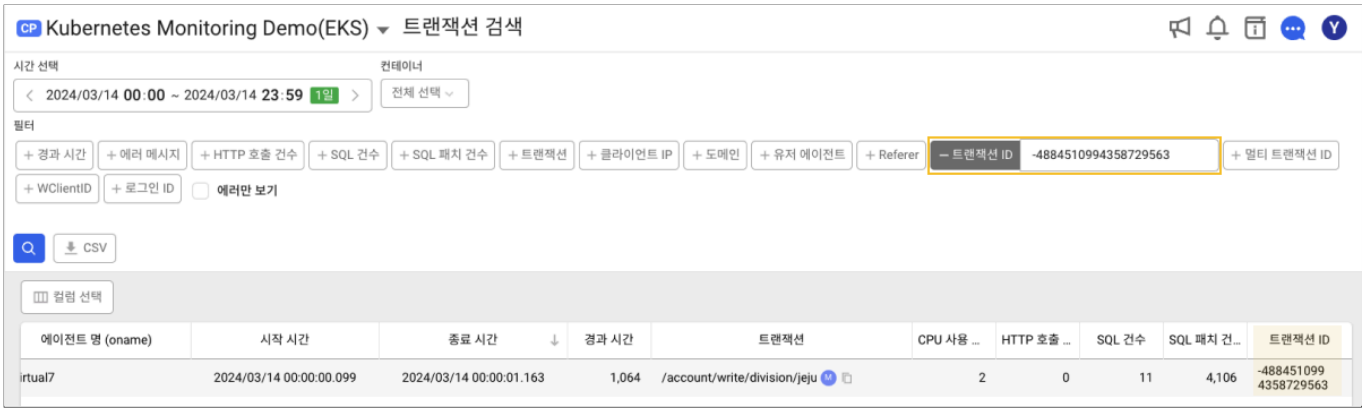
유저 에이전트 기반으로 트랜잭션을 검색(부분 일치)합니다.

Referer

Referer 문자열 기반으로 트랜잭션을 검색합니다.

> ⓘ 기본 설정 상태의 에이전트는 Referer 정보를 수집하지 않습니다. Referer 설정이 필요한 경우 다음 문서를 참조해 관련 설정을 추가하세요.
>
> - Java
> - Node.js
> - Python

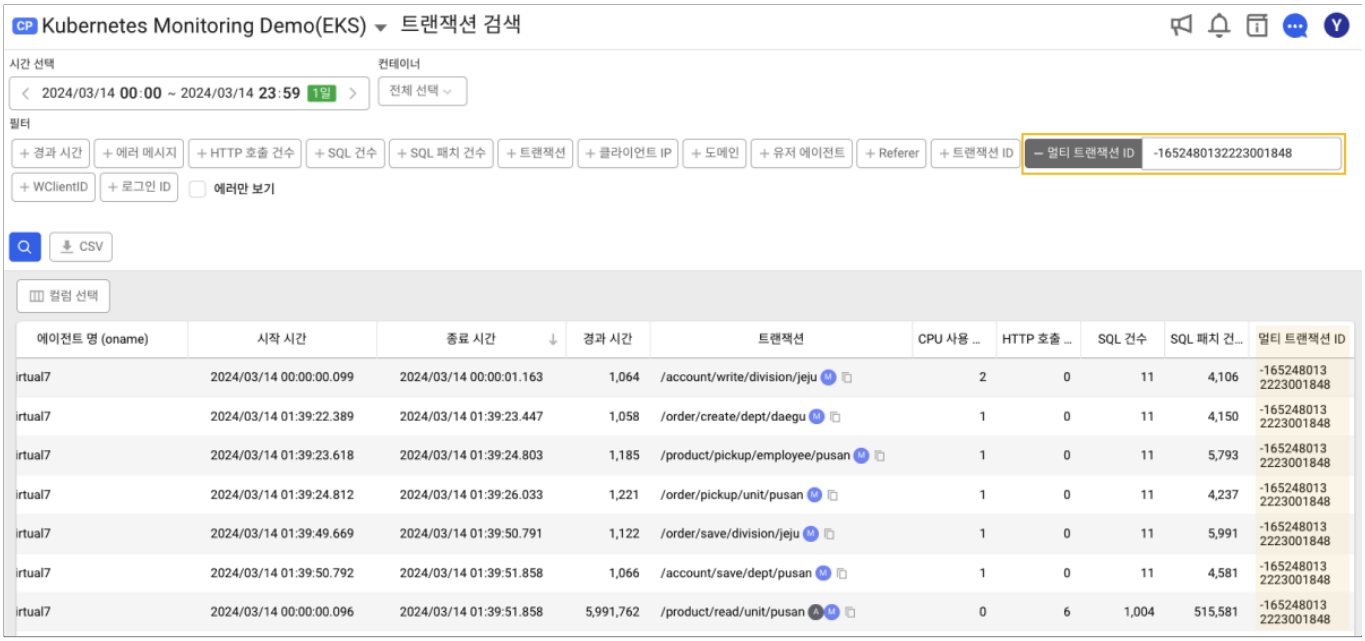
트랜잭션 ID



트랜잭션 ID 기반으로 트랜잭션을 검색(전체 일치)합니다. 트랜잭션 ID의 경우 검색 인덱스가 생성되어 있기 때문에 빠른 속도로 검색할 수 있습니다.

- 트랜잭션 ID는 **트랜잭션 정보** 창의 **레코드 요약** 탭에서 확인할 수 있습니다. 자세한 내용은 다음 문서를 참조하세요.
- 최신 버전의 에이전트가 아닌 경우 인덱스 사용이 아닌 풀스캔이 발생할 수 있습니다.

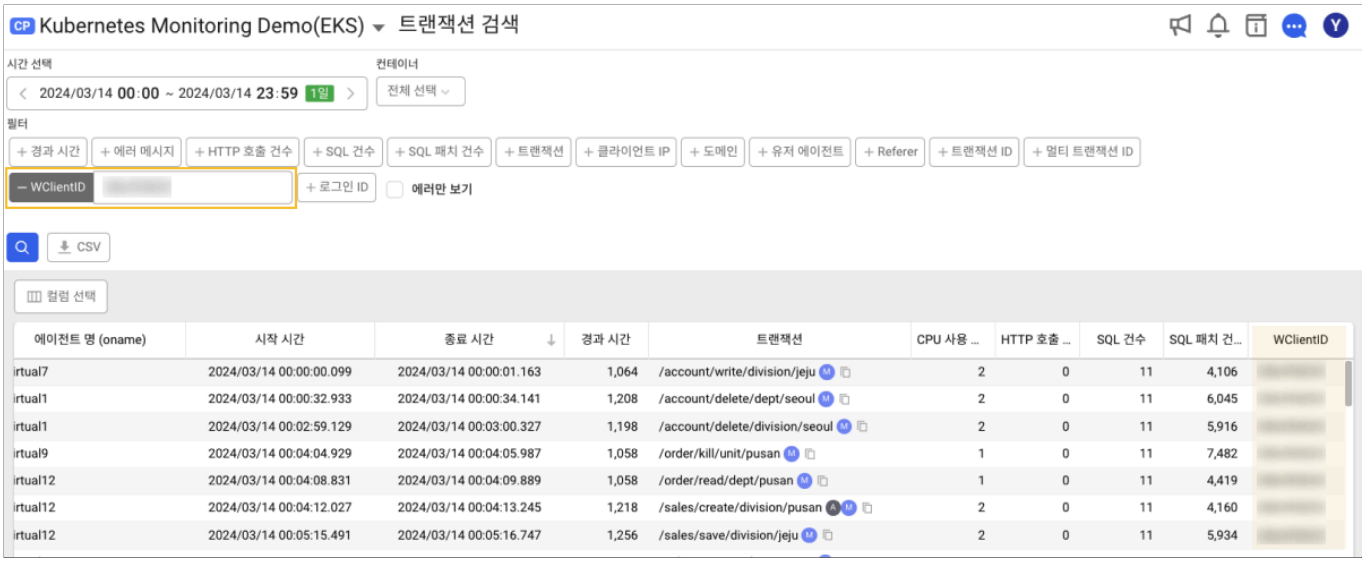
멀티 트랜잭션 ID



멀티 트랜잭션 ID 기반으로 트랜잭션을 검색(전체 일치)합니다.

- 멀티 트랜잭션 ID는 **트랜잭션 정보** 창의 **레코드 요약** 탭에서 확인할 수 있습니다. 자세한 내용은 다음 문서를 참조하세요.
- 최신 버전의 에이전트가 아닌 경우 인덱스 사용이 아닌 풀스캔이 발생할 수 있습니다.
- **멀티 트랜잭션 ID** 필터를 적용하여 검색할 경우 **트랜잭션 ID** 필터만 추가 적용할 수 있습니다.

WClientID



클라이언트 ID(`wClientId`)는 와탭 에이전트가 사용자를 구분하기 위해 부여하는 ID입니다. 이 ID 값 기반으로 트랜잭션을 검색(전체 일치)합니다. 검색을 위한 별도 인덱스는 없습니다.

> ⓘ 와탭은 클라이언트와 관련한 정보를 기본 저장합니다.

로그인 ID

로그인 아이디 기반으로 트랜잭션을 검색합니다.

> ⓘ 기본 설정 상태의 에이전트는 로그인 ID 정보를 수집하지 않습니다. 로그인 ID 설정이 필요한 경우 다음 문서를 참조해 설정을 추가하세요.

검색 결과 다운로드하기

사용자가 설정한 필터 조건에 따라 검색된 결과를 CSV 파일로 다운로드할 수 있습니다.

1. 버튼을 클릭해 검색 결과를 조회하세요.
2. CSV 버튼을 선택하세요.
3. 최대 CSV 라인 수를 입력하세요. 입력한 라인 수만큼 조회된 결과를 저장할 수 있습니다.
4. **다운로드** 버튼을 선택하면 CSV 파일로 다운로드합니다.

> CSV 파일 이름은 profile_ {pcode} _ YYMMDD .csv 형식입니다.

경고 알림

모니터링 대상 시스템 또는 애플리케이션에서 문제가 발생했을 때, 미리 설정한 조건에 따라 사용자에게 알림을 전달하는 기능입니다. 알림은 이메일, SMS, 메신저, 앱 푸시 등 다양한 채널로 전송됩니다.

예를 들어, CPU 사용률, 메모리 사용량, 디스크 I/O, 네트워크 트래픽, 트랜잭션 수, 오류 발생, 응답 시간 지연 등 주요 성능 지표에 알림 기준을 설정해두면, 해당 기준을 초과하거나 이상 징후가 감지될 때 알림이 전송됩니다.

와탭의 AI 기반 알림 기능을 사용하면 복잡한 조건을 직접 설정하지 않아도 됩니다. AI는 애플리케이션의 정상적인 동작 패턴을 학습해 이상 패턴을 자동으로 감지하여 알려줍니다. 태그 카운트 알림 기능을 활용하면 특정 태그 기반으로 집계된 값에 조건을 설정하여 복잡한 상황도 정밀하게 감지할 수 있습니다.

주요 기능

- **이벤트 설정**: 임계치 기반 알림 조건과 수신 채널 설정
- **이벤트 수신 설정**: 사용자별 수신 채널, 시간대 설정
- **이벤트 기록**: 알림 발생 기록 조회

권한

이벤트를 추가하거나 설정하려면 **알림 설정 권한**이 필요합니다.

- **조회**: 메트릭스 이벤트 설정 내용을 확인하려면 **조회 분석 권한**이 필요합니다.
- **추가/수정/삭제**: 설정을 변경하려면 **알림 설정 권한**이 필요합니다.
- **알림 수신**: 알림을 수신하려면 **알림 수신 권한**이 필요합니다.

> **참고 문서**
>
> 자세한 내용은 멤버 권한을 참고하시기 바랍니다.

이벤트 설정

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

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

메트릭스

메트릭스 이벤트는 실시간 데이터를 기반으로 시스템 및 애플리케이션의 성능 지표를 모니터링할 수 있습니다.

복합 메트릭스

복합 메트릭스 기능은 복잡한 규칙을 활용해 이벤트를 생성하고 경고 알림을 보낼 수 있습니다.

히트맵 패턴

히트맵 패턴은 응답 시간과 호출 밀도 등의 분포 변화를 기반으로 이상 징후를 자동 감지해 경고 알림을 보낼 수 있습니다.

이상치 탐지

이상치 탐지 기능은 예상치 못한 패턴을 감지해 경고 알림을 보내도록 설정할 수 있습니다.

실시간 로그

실시간으로 수집한 로그에서 설정한 이벤트 조건이 발견되면 경고 알림을 보냅니다.

복합 로그

로그 내용을 기준으로 경고 알림을 발생시키는 복합 로그 이벤트를 설정합니다.

트랜잭션

트랜잭션 이벤트는 특정 URL, 에러, 상태 코드 등을 기준으로 트랜잭션을 세밀하게 모니터링 할 수 있습니다.

참고

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

공통 기능

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

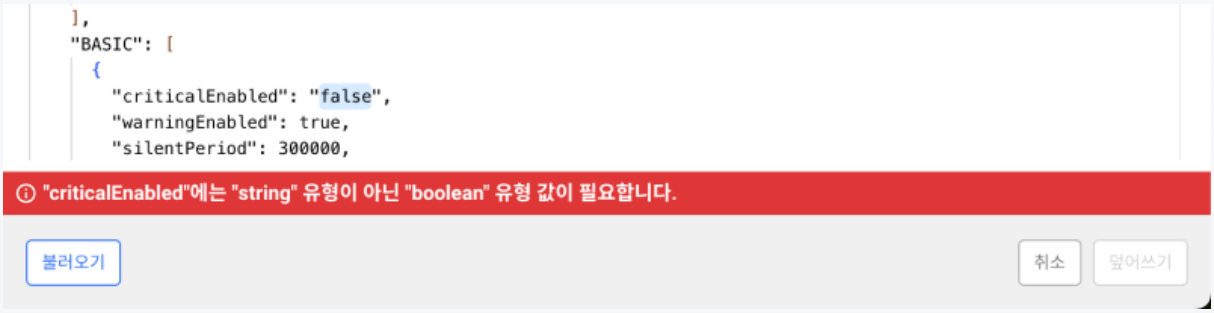
- **JSON 일괄 수정**: 모든 이벤트 조건을 JSON 형식으로 일괄 수정할 수 있습니다.
- **JSON 일괄 다운로드**: 현재 등록된 이벤트 조건을 JSON 파일로 저장할 수 있습니다.
- **Excel 일괄 다운로드**: 이벤트 목록을 Excel 파일로 내려받아 외부 보고나 백업에 활용할 수 있습니다.

JSON 일괄 수정

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

덮어쓰기 후 이전의 이벤트 설정으로 복구할 수 없습니다.

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



JSON 데이터 구조

```
{
 "productType": "{productType}",
 "platform": "{platform}",
 "events": {
  "HITMAP": {
   ...
  },
  "LOG_REALTIME": [
   {...}
  ],
```

```
    "METRICS": [
      {...}
    ],
    "COMPOSITE_LOG": [
      {...}
    ],
    "BASIC": [
      {...}
    ],
    "COMPOSITE_METRICS": [
      {...}
    ],
    "ANOMALY": [
      {...}
    ]
  }
}
```


JSON 필드	설명
productType	제품 타입(예: APM, Infra 등)
platform	플랫폼 정보(예: java, node 등)
events	이벤트 데이터
ㄴ HITMAP	히트맵 패턴
ㄴ LOG_REALTIME	실시간 로그 이벤트
ㄴ METRICS	메트릭스
ㄴ COMPOSITE_LOG	복합 로그 이벤트
ㄴ BASIC	기본 이벤트
ㄴ COMPOSITE_METRICS	복합 메트릭스

JSON 필드	설명
ㄴ ANOMALY	이상치 탐지

JSON 일괄 다운로드

화면 상단의 [JSON 일괄 다운로드] 버튼을 클릭하세요.

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

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

Excel 일괄 다운로드

화면 상단의 [Excel 일괄 다운로드] 버튼을 클릭하세요.

메트릭스

홈 > 프로젝트 선택 > **경고 알림** > **이벤트 설정** > **메트릭스** 탭

프로젝트에 설정된 모든 메트릭스 기반 이벤트 목록을 확인할 수 있습니다. 사용자가 설정한 메트릭스 이벤트를 확인, 수정, 삭제하거나, 새로운 이벤트를 추가할 수 있습니다.

기본 옵션

- **이벤트 레벨**: 이벤트의 심각도를 의미하며 정상(Info), 경고(Warning), 위험(Critical) 세 단계로 구분
- **에이전트 중복 알림**: 설정한 시간(30초, 1분, 3분, 5분, 10분)에 같은 이름의 에이전트가 동시에 2개 이상 동작하는 경우, 선택한 시간 동안 이벤트 발생 안함
- **검색**: 이벤트 이름을 기준으로 검색
- **JSON 다운로드/업로드**: 이벤트 규칙은 JSON 편집기를 통해 바로 수정하거나, 파일로 다운로드하여 수정 후 다시 업로드 가능
- **+ 이벤트 추가**: 새로운 메트릭스 이벤트 추가
- **컬럼 크기 조정**: 각 컬럼의 경계를 드래그하여 너비 조정 가능

표 | 메트릭스 이벤트 목록 구성

항목	설명
No.	이벤트의 순번
	이벤트의 활성화 여부 – 활성화: 조건을 충족하면 이벤트 발생 – 비활성화: 이벤트가 동작하지 않으며, 조건을 검사하지 않음
	이벤트 설정 편집 및 삭제
이벤트 이름	사용자가 지정한 이벤트명 – 최대 255자까지 입력 가능
규칙	이벤트 발생 조건

항목	설명
대상	이벤트가 적용되는 모니터링 대상
발생 횟수	이벤트 발생 기준이 되는 기간과 횟수 – 선택한 시간 동안 설정한 횟수만큼 조건을 충족하면 이벤트 발생
일시 중지	이벤트 알림 발생 후, 같은 이벤트 발생을 일정 시간 동안 멈춤 – 선택 가능 시간: 사용 안 함, 5분, 10분, 15분, 20분, 30분, 1시간, 2시간, 3시간, 6시간, 12시간, 1일 – 해소된 알림이 활성화된 경우, 정상(RECOVERED) 상태 알림을 받은 후 선택한 시간 동안 같은 이벤트가 발생해도 알림이 발생하지 않음
해소된 알림	Critical과 Warning 레벨의 이벤트가 해소되면 RECOVERED(정상) 상태의 알림 수신 여부 – 활성화: 진행 중 → 정상 상태로 전환되면 RECOVERED 알림을 보내고, 이벤트 기록에서 진행 중/해소 상태를 구분해 볼 수 있음 – 비활성화: 임곗값을 초과할 때마다 단발성 이벤트 발생
이벤트 수신	해당 이벤트 알림을 받을 사용자 또는 그룹(수신 태그)

이벤트 설정

메트릭스 이벤트는 모니터링 대상에서 발생하는 이벤트를 조건에 따라 정의하고, 효율적으로 알림을 받을 수 있도록 설정할 수 있는 기능을 제공합니다.

이벤트 추가

메트릭스 이벤트를 추가하려면 **이벤트 설정** > **메트릭스** 탭으로 이동한 후, 오른쪽 상단의 [+ 이벤트 추가] 버튼을 클릭하세요.

메트릭스 이벤트 추가 방법
Step 1. 템플릿 선택: 이벤트의 기본 구조를 선택합니다.
Step 2. 이벤트 조건 정의: 이벤트를 어떤 기준으로 발생시킬지 조건을 설정합니다.

메트릭스 이벤트 추가 방법
Step 3. 이벤트 대상 선택: 이벤트가 발생하는 모니터링 대상을 선택합니다.
Step 4. 기본 정보 및 수신 설정: 이벤트 이름 및 메시지, 수신 대상을 설정합니다.

템플릿 선택

1. 이벤트를 어떻게 구성할지에 따라 화면 왼쪽의 3가지 템플릿 중 하나를 선택하세요.
    - **새로 만들기**: 카테고리와 지표를 처음부터 자유롭게 구성할 수 있으며, 복잡한 조건이나 세밀한 모니터링 규칙이 필요한 고급 사용자에게 적합
    - **빠른 설정**: 사전 정의된 카테고리, 지표, 기본 임계치가 제공되며, 임곗값만 조정하여 빠르게 적용할 수 있어 초급 사용자에게 적합
    - **고급 설정**: 사전 정의된 카테고리와 지표를 기반으로 조건을 추가하거나 자유롭게 변경

> ⓘ 메트릭스 이벤트에서는 다양한 설정 방식을 지원합니다. 모니터링 플랫폼에 따라 제공하는 템플릿은 다를 수 있습니다.

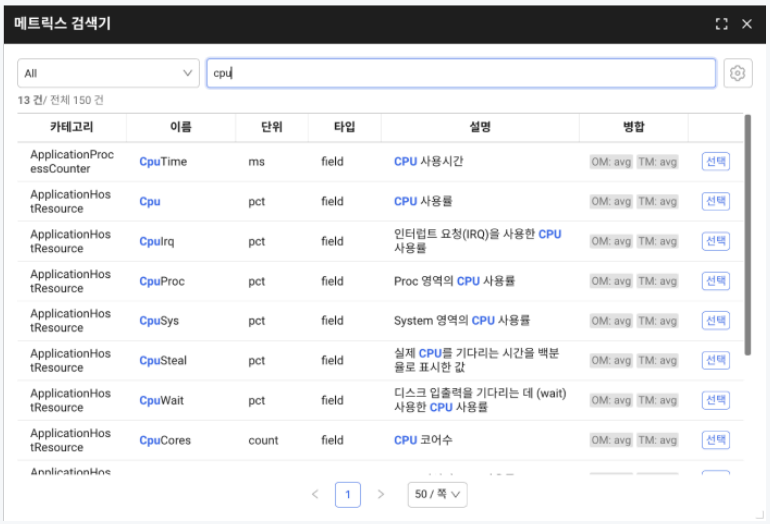
이벤트 조건 정의

이벤트 조건 정의 단계에서 이벤트가 언제 발생할지 결정하는 지표 설정, 발생 횟수, 일시 중지, 해소된 알림, 이벤트 동작 시간을 설정합니다.

지표 설정

1. 검색창에서 카테고리를 입력하거나, 카테고리 목록에서 카테고리(지표)를 선택하세요.
   카테고리를 선택해야 이후 지표 설정을 진행할 수 있습니다.
    - 카테고리 목록은 카테고리 이름, 데이터 수집 주기, 키를 확인할 수 있습니다. 최근 3시간 동안 프로젝트에서 수집된 메트릭스 데이터가 표시됩니다.
2. 목록에 없는 지표를 사용하려면 **직접 입력하기**를 선택하여 카테고리를 직접 입력하세요.

> **카테고리와 지표를 모르는 경우 메트릭스 검색기를 활용하세요.**
>
> 처음 지표를 설정하는 경우 메트릭스 검색기를 활용하면 더 빠르게 지표를 찾을 수 있습니다.
>
> - 메트릭스 검색기에서 원하는 지표를 검색하면 관련 카테고리와 상세 정보를 확인할 수 있습니다.
> - 검색 결과에서 필요한 메트릭스의 [선택] 버튼을 클릭하면, 선택한 값이 이벤트 설정 화면에 자동 반영됩니다.



3. 이벤트 레벨(Critical, Warning, Info)을 선택하세요. 하나의 지표에 여러 레벨을 설정할 수 있습니다.

> **이벤트 레벨 동작**
>
> 여러 조건을 만족하더라도 가장 높은 우선 순위(Critical → Warning → Info)의 레벨만 이벤트가 발생합니다. 레벨 동작 방식은 레벨 동작 가이드를 참고하세요.
>
> - 이벤트 레벨 수준은 위험(Critical)과 경고(Warning), 정상(Info)으로 구분합니다.
> - 같은 지표로 레벨을 설정하고 단계별 임계치를 설정하는 것을 권장합니다.

4. 선택한 이벤트 레벨의 필드, 연산자, 값을 입력하세요.

   a. **+ 추가**를 클릭해 조건을 추가할 수 있습니다.

   - `&&` : **모든 조건을 동시에 만족**해야 이벤트 발생

- `||`: **하나 이상의 조건만 만족**하면 이벤트 발생

b. — 를 클릭하면 해당 조건이 삭제됩니다.

> ⓘ 조건식 작성 방법과 지원되는 연산자는 조건 설정 가이드를 참고하세요.

발생 횟수

조건이 몇 번 충족되었을 때 이벤트를 발생시킬지 설정합니다. 지속적인 문제만 감지하고 싶을 때 유용합니다.

1. 발생 빈도를 선택하세요.
   - **연속**: 설정한 횟수만큼 이벤트가 연속으로 발생하는 경우 이벤트 발생
   - **최근**: 선택한 시간 내에 설정한 횟수만큼 이벤트가 발생하는 경우 이벤트 발생
     - 선택 가능 시간: `5초` , `10초` , `15초` , `30초` , `1분` , `2분` , `3분` , `4분` , `5분` , `10분` , `30분` , `1시간`
2. 발생 횟수를 입력하세요.
   - **Interval** 값은 선택한 카테고리의 데이터 수집 간격입니다. (Interval: 10s)

일시 중지

과도한 이벤트 발생을 방지하기 위해, 이벤트 동작을 일시 중지할 시간을 선택하세요. 이벤트 발생 후 설정한 시간 동안 동일한 이벤트가 발생하지 않습니다.

- 선택 가능 시간: `사용 안 함` , `5분` , `10분` , `15분` , `20분` , `30분` , `1시간` , `2시간` , `3시간` , `6시간` , `12시간` , `1일`

> ⓘ **해소된 알림 활성화 시**
>
> 알림은 정상 상태 알림(**RECOVERED**)을 받은 시점을 기준으로 설정한 시간이 지나기 전까지 동일한 이벤트가 발생해도 알림이 발생하지 않습니다.

해소된 알림

해소된 알림 기능을 활성화하면, 이벤트 발생 이후 상태 변화(진행 중 → 해소됨)를 추적할 수 있습니다. **이벤트 기록** 메뉴에서 진행

중인 이벤트로 표시됩니다.

- 해소된 알림은 Critical과 Warning 레벨에만 적용되며, Info 레벨은 항상 단발성 이벤트로 동작합니다.
- Critical과 Warning 레벨의 이벤트가 해소되면, 정상(RECOVERED) 상태의 알림이 전송됩니다.
- 토글 버튼을 클릭해 **활성화** 또는 **비활성화**하세요.
  - **활성화**: 이벤트는 상태 기반으로 동작하며, 임계치 초과 시 진행 중, 임계치 이하로 복구 시 해소 상태로 전환됨
  - **비활성화**: 임곗값을 초과할 때마다 단발성 이벤트 발생

이벤트 동작 시간

이벤트가 특정 시간대(근무/비근무/점검 시간 등)에 동작하도록 태그를 설정하세요. 태그를 설정하지 않으면 이벤트가 활성화된 상태에서 24시간 상시 동작합니다.

1. **+ 추가**를 클릭하세요.
2. 이벤트 동작 시간에서 **+새로운 태그 생성**을 클릭하세요.
3. **이벤트 동작 시간 태그 생성**에서 **태그 이름, 요일, 시간, 색상**을 선택하고 [적용] 버튼을 클릭하세요.
4. 생성한 태그는 태그 목록에서 확인할 수 있으며, 체크 박스를 선택하면 적용됩니다.
5. 수정 또는 삭제할 태그의 아이콘을 클릭해 **이벤트 동작 시간 태그 수정**에서 태그를 수정하거나 태그 삭제할 수 있습니다.

> 태그 삭제 시 해당 태그가 적용된 모든 사용자의 태그가 삭제됩니다. 단, 이벤트 규칙에서 사용 중인 태그는 삭제할 수 없습니다.

시뮬레이션

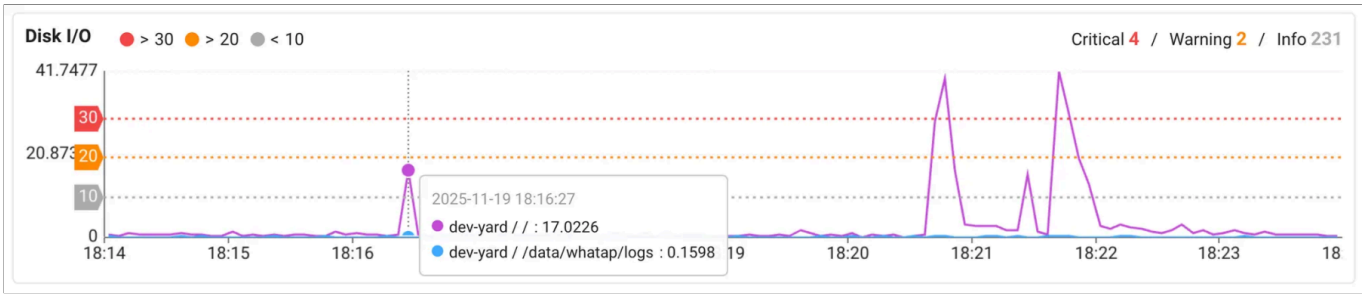
시뮬레이션 기능은 설정한 이벤트 조건을 과거 데이터에 기반으로 사전 테스트 할 수 있습니다. 예상 이벤트 수가 지나치게 많거나 부족하다면 조건을 조정해 최적의 임곗값을 찾을 수 있습니다.

- **규칙 검증**: 새로운 이벤트 규칙을 만들었을 때, 과거 데이터에 적용하면 어떤 결과가 나오는지 확인 가능
- **임곗값 튜닝**: 이벤트가 자주 발생하거나 전혀 발생하지 않을 때 적절한 수준으로 임곗값 조정

시뮬레이션 시작하기

1. 이벤트 조건 설정이 완료되면 [시뮬레이션] 버튼을 클릭하세요.
2. 예상 이벤트 건수와 시각화된 결과를 확인합니다.

시뮬레이션 결과 이해하기



1. **시계열 데이터**

   지표의 시간대별 변화와 각 레벨의 임계선이 그래프로 표시되어 이벤트 발생 시점을 직관적으로 확인할 수 있습니다.

2. **레벨별 발생 현황**

   화면 우측에 Critical, Warning, Info 레벨의 이벤트 발생 횟수가 표시됩니다. 전체 이벤트 분포와 빈도를 파악할 수 있습니다.

3. **대상별 상세 정보**

   차트의 특정 지점에 마우스를 올리면 해당 시점의 정확한 지표 값과 모니터링 대상이 표시됩니다. 여러 대상이 있는 경우 색상으로 구분되며, 대상별 패턴이 다른 경우 필터링 기능을 사용해 개별적으로 분석할 수 있습니다.

4. **피크 구간 식별**

   그래프에서 지표가 급격히 상승하는 시간을 확인하여 문제가 자주 발생하는 시간대나 패턴을 파악할 수 있습니다.

시뮬레이션 활용 팁

- **다양한 시간대 시뮬레이션**

  평일/주말, 피크/비피크 시간대를 각각 시뮬레이션하여 시간대별 최적 임곗값을 설정하세요.

- **실제 장애 기간 포함**

  과거 장애가 발생했던 기간을 포함하여 시뮬레이션을 실행하면 현재 규칙이 장애를 정확히 감지하는지 확인하세요.

- **대상 필터링 추가**

모니터링 대상이 많으면 시뮬레이션 결과가 복잡해집니다. 필터링 기능으로 특정 대상만 선택하여 명확한 결과를 확인할 수 있습니다.

- **임곗값 단계적 조정**

  임곗값을 한 번에 크게 변경하기보다 10% 단위로 점진적으로 조정하면서 최적값을 탐색하는 것이 좋습니다.

이벤트 대상 선택

이벤트 대상을 지정하지 않으면, 프로젝트에 포함된 전체 대상으로 이벤트가 발생되어 많은 이벤트가 발생할 수 있습니다. 따라서 이벤트가 적용될 대상을 명확히 설정하는 것을 권장합니다.

이벤트 대상을 변경한 경우, 이벤트 건수가 달라질 수 있습니다. **시뮬레이션** 버튼을 클릭한 후, 다시 실행해 예상 이벤트 수를 확인하세요.

대상 선택

이벤트가 적용될 대상을 **선택 입력** 또는 **직접 입력** 방식으로 지정할 수 있습니다.

- 선택 입력 **직접 입력**

1. 태그, 연산자, 값을 선택 또는 입력하세요. a. 태그를 찾기 어렵다면, **메트릭스 검색기**에서 태그를 검색할 수 있습니다.
2. **+ 추가**를 클릭해 조건(`&&` , `||`)으로 이벤트 대상을 추가하세요.

모니터링할 대상의 태그, 연산자, 값을 직접 입력하세요.

- 직접 입력을 선택하면 입력란 아래에 사용 가능한 태그 목록(**태그 선택 입력**)이 함께 제공됩니다.

```
ex. endsWith(okindName, 'example_name') && container == 'prod.billing'
ex. ${4xxErrorType} == '401'
```

입력 방법을 변경하면, 작성한 내용이 초기화됩니다.

대상 선택 연산자 목록

선택 입력 방식에서 문자열 태그를 선택하면 연산자 목록에 다음 두 항목이 함께 표시됩니다. 목록에 나타나는 이름은 함수 표기 그대로이며, `startsWith(key, 'match_text')` 와 같은 기존 항목과 형태가 같습니다.

- `equalsIgnoreCase(key, 'match_text')`
- `notEqualsIgnoreCase(key, 'match_text')`

두 연산자는 대소문자를 구분하지 않고 값을 비교합니다. **직접 입력** 방식에서는 같은 함수를 조건식에 직접 작성해 사용할 수 있습니다. 조건식 문법과 지원되는 함수 전체 목록은 조건 설정 가이드를 참고하세요.

> ⓘ 이벤트 발생 조건과 이벤트 대상 선택은 같은 문법을 사용하므로 두 함수를 양쪽 조건식에 모두 작성할 수 있습니다.

기본 정보 및 수신 설정

이벤트의 이름, 메시지, 수신 대상을 설정합니다. 설정한 내용은 이벤트 기록과 실제 알림 메시지에 그대로 표시되므로, 식별하기 쉽고 명확하게 작성하는 것이 중요합니다.

이벤트 활성화

설정한 조건을 충족했을 때 이벤트 발생 여부를 설정합니다.

- **활성화**: 설정한 조건을 충족하면 이벤트가 발생됨
- **비활성화**: 조건을 검사하지 않음

이벤트 이름

이벤트 제목으로 사용될 이벤트 이름을 입력하세요. 설정한 이벤트 이름은 **이벤트 목록** 및 **이벤트 기록** 메뉴에 표시되며, 해당 이름으로 검색할 수 있습니다.

메시지

이벤트 메시지를 입력하세요. 입력한 메시지는 이벤트 기록 및 알림에 사용됩니다. 을 클릭하면 이전에 작성한 메시지 기록을 확인할 수 있습니다. 메시지에 변수를 사용하여 이벤트 발생 시점의 실제 값을 포함할 수 있습니다.

- **변수 사용 규칙**
  - 변수 형식: `${지표명}`

- 설정한 **카테고리와 동일한 카테고리** 내 지표만 변수로 사용할 수 있습니다.
- 여러 변수를 조합해 상세한 메시지를 구성할 수 있습니다.
- 사용 가능한 변수 목록은 **메트릭스 조회** 메뉴에서 확인할 수 있습니다.

- **변수 활용 예시**

	이벤트 설정	이벤트 발생 시
제목	Disk 사용량 증가	Disk 사용량 증가
메시지	`Disk = ${disk}%`	`Disk = 89.9%`

이벤트 수신

이벤트 알림을 받을 멤버를 지정합니다.

- **전체 수신** **태그 선택 수신**

프로젝트에서 이벤트 수신 설정이 활성화된 모든 대상(멤버, 채널)에게 알림을 전송합니다.

해당 태그를 가진 프로젝트 멤버에게 알림을 전송합니다.

1. **태그 선택 수신**을 선택하세요.
2. **수신 태그** 항목에서 [+ 태그 추가] 또는 [+] 버튼을 클릭하세요.
3. **이벤트 수신 태그** 창 아래의 **+ 새 태그 생성**을 클릭하세요.
4. **태그 생성** 창에서 태그 이름을 입력하고, 색상을 선택한 후, [태그 생성] 버튼을 클릭해 태그를 생성하세요.
5. 생성된 태그는 태그 목록의 아이콘을 클릭해 수정 또는 삭제하세요.
6. **이벤트 수신 태그** 창에서 **태그 목록**에서 원하는 태그를 선택하면 적용됩니다.

> **수신 태그**
>
> 수신자 태그는 알림을 받을 사용자 또는 채널을 **그룹 단위로 관리하기 위한 기능**입니다. 이벤트에 수신자 태그를 설정하면, 해당 태그가 포함된 사용자에게만 알림이 전달됩니다. 수신 태그를 설정하지 않으면 프로젝트 내 모든 사용자에게 알림이 전송됩니다.

이벤트 수정/삭제

1. **경고 알림** > **이벤트 설정**의 **메트릭스** 탭으로 이동하세요.
2. 이벤트 목록에서 수정 또는 삭제하려는 이벤트의 아이콘을 클릭하세요.
3. **이벤트 규칙 수정** 창에서 옵션을 수정하고, [저장] 버튼을 클릭하세요.

   a. 선택한 이벤트를 삭제하려면 **이벤트 규칙 수정** 창의 오른쪽 위의 [삭제] 버튼을 클릭하세요.

이벤트 공유

메트릭스 이벤트 설정을 JSON 파일로 저장해 다른 사용자와 설정을 공유하거나 다른 사용자의 설정을 가져올 수 있습니다.

- JSON 파일명: event-rules- `YYYYMMDD` .json

내보내기

1. 화면 오른쪽 위에 [**JSON**] 버튼을 클릭하세요.
2. JSON 편집 창이 나타나면 [내보내기] 버튼을 클릭하세요.
   - 이벤트를 검색 후 내보내기 기능을 이용하면 검색한 목록만 JSON 파일로 다운받을 수 있습니다.
3. JSON 파일이 다운로드되면 공유할 다른 사용자에게 전달하세요.

가져오기

1. 화면 오른쪽 위에 [ ] 버튼을 선택하세요.
2. **내보내기** 기능을 통해 다운로드한 JSON 파일을 선택하세요.
3. JSON 편집 창이 나타나면 [목록에 추가하기] 또는 [덮어쓰기] 버튼을 선택하세요.

> 같은 종류의 제품 간에 이용할 것을 권장합니다. 다른 제품의 프로젝트로부터 이벤트 설정을 가져올 수는 있지만 정상 작동하지 않습니다.

Json 형식으로 수정하기

1. 화면 오른쪽 위에 [**JSON**] 버튼을 클릭하세요.
2. 편집 창이 나타나면 JSON 형식에 맞춰 내용을 수정하세요.
3. 수정을 완료하면 화면 아래 [저장] 버튼을 선택하세요.

> ⓘ 수정한 내용이 JSON 형식에 맞지 않으면 화면 아래에 에러 메시지가 표시되며, 저장할 수 없습니다. 표시되는 에러 메시지는 형식에 따라 다를 수 있습니다.
>
> Expected ',' or '}' after property value in JSON at position 1964 (line 80 column 42)

JSON 데이터 구조

```
{
  "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"
        ]
  }
}
```


표 | JSON 데이터 구조

JSON 필드	타입	설명	비고
version	Integer	이벤트 규칙의 버전	변경 제한
eventId	String	이벤트 규칙의 고유 식별자	
enabled	boolean	이벤트 활성화 여부	
stateful	boolean	상태 기반 이벤트 여부	
title	String	사용자가 설정한 이벤트 규칙 이름	
message	String	사용자가 설정한 이벤트 규칙 메시지	
category	String	데이터 카테고리	
alertLabel	`List<String>`	이벤트의 상태 관리를 위해 카테고리마다 지정되는 기본 식별자(primary key) 값	변경 제한
repeatCount	Integer	반복 횟수	
repeatDuration	Long	반복 기간	

JSON 필드	타입	설명	비고
silent	Integer	일시 중지 시간	
receiver	List<String>	수신자 태그의 키값 목록	
timeTag	List<String>	동작시간 태그의 키값 목록	
selectString	String	대상 선택	
conditions	List<Object>	레벨별 발생 조건 목록	
ㄴ level	byte	레벨 값	Critical: 30 Warning: 20 Info: 10
ㄴ enabled	boolean	레벨 활성화 여부	
ㄴ rule	String	레벨 발생 조건	
createTime	Long	이벤트 규칙 최초 생성 시간	
lastModifiedTime	Long	이벤트 규칙 마지막 수정 시간	
lastModifiedUser	String	이벤트 규칙 마지막 수정 계정	
basic	boolean	빠른 설정으로 생성한 이벤트 여부	변경 제한
metaId	String	빠른 설정 템플릿 고유 식별자	변경 제한: 이 값은 빠른 설정으로 생성한 이벤트에만 존재합니다. (basic=true)
selectCondition	Object	빠른 설정 대상 선택값	이 값은 빠른 설정으로 생성한 이벤트에만 존재합니다. (basic=true)

‘변경 제한’으로 표시된 필드는 이벤트 동작에 영향을 줄 수 있으므로 값을 변경하지 않는 것을 권장합니다.

부록

발생 조건, 대상 선택 가이드

메트릭스 경고 알림의 이벤트 발생 조건과 이벤트 대상 선택은 동일한 문법을 사용합니다. 단, 이벤트 발생 조건은 필드(Field)의 Key를 변수로 사용하고, 이벤트 대상 선택은 태그(Tag)의 Key를 변수로 사용합니다.

레벨 동작 가이드

이벤트 레벨은 위험(Critical)과 경고(Warning), 정상(Info)으로 구분합니다.

1. **우선순위 기반 이벤트 발생**

   여러 레벨의 조건을 동시에 충족할 경우, 가장 높은 우선순위의 이벤트만 발생합니다.

   예시

   ```
   설정:
   - Warning: CPU > 70%
   - Critical: CPU > 90%

   현재 상태: CPU 95%
   → 결과: Critical 이벤트만 발생 (Warning은 억제됨)
   ```

2. **레벨 상승**

   진행중 상태를 갖는 이벤트에서 낮은 레벨 이벤트가 진행 중일 때 높은 레벨 조건을 충족하면, 두 레벨 모두 진행 중 상태가 됩니다.

   예시

   ```
   설정:
   - Warning: CPU > 70%
   - Critical: CPU > 90%

   시나리오:
   1) CPU 80% → Warning 발생 (진행 중)
   ```

```
2) CPU 95% → Warning (진행 중), Critical (진행 중)
```

3. **레벨 하강**

   진행중 상태를 갖는 이벤트에서 높은 레벨에서 낮은 레벨로 전환되면 높은 레벨은 해소되고 낮은 레벨이 유지됩니다.

   예시

```
설정:
- Warning: CPU > 70%
- Critical: CPU > 90%

시나리오:
1) CPU 97% → Critical 발생
2) CPU 85% → Critical 해소, Warning 유지
3) CPU 65% → Warning 해소
```

4. **동작 흐름도**

```
- Warning: CPU > 70%
- Critical: CPU > 90%

메트릭 값: 60%  →  75%  →  92%  →  85%  →  60%
상태:      정상    Warning   Critical  Warning   정상
                   (발생)    (발생)    (유지)    (해소)
                             Warning
                             (유지)
```

> ⓘ 진행 중 상태를 갖는 이벤트는 조건이 충족되는 동안 계속 활성화 상태를 유지하며, 조건이 해제될 때까지 지속됩니다.

조건 설정 가이드

> ⚠ 지표명이 숫자로 시작하거나 특수문자가 포함된 경우 `${지표명}` 형태로 작성해야 합니다.
>
> - 괄호 `()`, `[]`

- 연산자 `+`, `-`, `*`, `/`, `%`
- 구분자 `:`, `@`, `#`, `,`, (공백)
- 기타 특수문자 `!`, `^`, `&`, `|`, `~`, ```, `=`

`${}` 사용 예시:

```
${cpu(xos)} > 50
${mem[0]} >= 100
${cpu-usage} > 80
${namespace:cpu} > 70
${metric name} > 60
${4xx_error} > 10
```

`${}` 없이 사용 가능: 영문자, 언더스코어(`_`), 점(`.`)만 포함된 경우

```
cpu > 80
cpu_usage > 50
CPUUtilization.Average > 0.8
```

1. 비교 연산자

숫자 비교

```
크다:          cpu > 80
크거나 같다:    cpu >= 80
작다:          memory < 1000
작거나 같다:    memory <= 1000
같다:          status == 200
같지 않다:      error != 0
```

문자열 비교

```
status == 'OK'
region == "us-east-1"
```

2. 산술 연산자

- 기본 산술 연산

```
덧셈:   cpu + 10 >= 90
뺄셈:   memory - 100 >= 500
곱셈:   cpu * 2 >= 100
나눗셈:  disk / 1024 >= 100
나머지:  value % 10 == 0
```

- 괄호를 이용한 우선순위 제어

```
(cpu + memory) * 2 >= 200
(disk - used) / total >= 0.2
((cpu + memory) / 2) >= 50
```

- 음수 표현

```
cpu > -100
```

3. 논리 연산자

- AND 연산 (&&)

```
cpu > 80 && memory > 1000
${cpu(xos)} > 50 && ${mem(xos)} > 60
```

- OR 연산 (||)

```
cpu > 90 || memory > 90
disk < 10 || network > 1000
```

- 복합 논리 연산

```
(cpu > 50 && memory > 50) || disk < 20
cpu > 80 && (memory > 1000 || disk > 500)
```

4. 패턴 매칭 연산자

- LIKE 연산자

```
oname like 'prod-*'
url like '*error*'
message like 'WARN%'
```

- NOT LIKE 연산자

```
oname not like 'test-*'
url not like '*debug*'
```

5. 내장 함수

- Null 체크 함수

```
isNull(value)          # null인지 확인
isNotNull(value)       # null이 아닌지 확인
nvl(value, 0)          # null이면 기본값 반환
isEmpty(str)           # 빈 문자열인지 확인
isNotEmpty(str)        # 빈 문자열이 아닌지 확인
```

- 집계 함수

```
sum(cpu, memory, disk)          # 합계
avg(cpu, memory)                # 평균
max(cpu, memory, disk)          # 최대값
min(cpu, memory, disk)          # 최소값
count(value1, value2, value3)   # 개수
```

- 수학 함수

```
round(cpu, 2)          # 반올림 (소수점 2자리)
```

- 문자열 함수

```
length(str)            # 문자열 길이
startsWith(str, 'prefix')  # 접두사 확인
```

```
endsWith(str, 'suffix')    # 접미사 확인
indexOf(str, 'search')     # 문자열 위치
substring(str, 0, 10)      # 부분 문자열
trim(str)                  # 공백 제거
replace(str, 'old', 'new') # 문자열 치환
hasStr(str, 'search')      # 문자열 포함 여부
```

문자열 함수

```
equalsIgnoreCase(str, 'match_text')     # 대소문자를 구분하지 않는 일치 비교
notEqualsIgnoreCase(str, 'match_text')  # 대소문자를 구분하지 않는 불일치 비교
```

`==` 와 `!=` 는 대소문자를 구분하므로 `WebServer` 와 `webserver` 를 다른 값으로 판단합니다. 표기가 일정하지 않은 태그나 필드를 비교할 때 위 두 함수를 사용하면 대소문자 차이를 무시하고 값을 맞출 수 있습니다.

예시

```
equalsIgnoreCase(okindName, 'WebServer')
notEqualsIgnoreCase(container, 'PROD.billing')
```

이벤트 발생 조건과 이벤트 대상 선택은 같은 문법을 사용하므로 두 함수를 양쪽 조건식에 모두 작성할 수 있습니다.

- 조건 함수
  - **if:** 조건식이 참(`true`)이면 `true값` 을 반환하고, 거짓(`false`)이면 `false값` 을 반환하는 조건 함수

    형식

    ```
    if(조건, true값, false값)
    ```

    예시

    ```
    if(value >= 90, 'High', 'Medium') == 'High'
    ```

  - **decode**: 값과 조건을 비교하여 일치하는 조건의 결과를 반환, 없으면 기본값 반환

    형식

```
decode(값, 조건1, 결과1, 조건2, 결과2, ..., 조건N, 결과N, 기본값)
```

예시

```
decode(value, 1, 'Low', 2, 'Medium', 3, 'High', 'Unknown') == 'Unknown'
```

- **in**: 값이 지정한 후보값 중 하나와 일치하면 `true`, 아니면 `false`

형식

```
in(값, 후보값1, 후보값2, ..., 후보값N)
```

예시

```
in(status, 200, 201, 204) == false
```

작성 시 주의사항

- 권장사항
  - 간단한 변수명은 `${}` 없이 사용 (예: cpu, memory)
  - 특수문자 포함 시 반드시 `${}` 사용
  - 복잡한 연산은 괄호로 명확하게 구분
  - 함수명은 정확히 입력 (대소문자 구분)
- 피해야 할 사항
  - 불완전한 표현식 작성 (예: `cpu >`, `memory &&`)
  - 괄호 미스매칭 (예: `(cpu > 80 && cpu > 70)`)
  - 연속된 연산자 사용 (예: `cpu >> 80`, `cpu >< 80`)

복합 메트릭스

홈 > 프로젝트 선택 > **경고 알림** > **이벤트 설정** > **복합 메트릭스** 탭

복잡한 규칙을 활용해 이벤트를 생성하고 경고 알림을 보낼 수 있습니다.

기본 옵션

- **검색**: 이벤트 이름 또는 이벤트 수신 태그를 선택해 원하는 이벤트 검색
- **JSON 다운로드/업로드**: 이벤트 규칙은 JSON 편집기를 통해 바로 수정하거나, 파일로 다운로드하여 수정 후 다시 업로드 가능
- **알림 메시지 설정**: 수신할 메시지의 상세 항목 선택 및 항목 별 내용 편집
- **+이벤트 추가**: 새로운 복합 메트릭스 이벤트 추가

> **권한**
>
> **복합 메트릭스** 이벤트를 설정하려면 **알림 설정** 권한이 있어야 합니다. 알림 설정 권한은 **홈 > 프로젝트 선택 > 관리 > 통합 멤버 관리**에서 권한을 부여할 멤버의 아이콘을 클릭해 설정할 수 있습니다.

표 | 복합 메트릭스 이벤트 목록 구성

항목	설명
이벤트 제목	사용자가 지정한 이벤트명 - 최대 255byte까지 입력 가능
조건	이벤트 발생 조건
이벤트 상태가 해소되면 추가 알림	Critical과 Warning 레벨의 이벤트가 해소되면 RECOVERED(정상) 상태의 알림 수신 여부 – 활성화: 진행 중 → 정상 상태로 전환되면 RECOVERED 알림을 보내고, 이벤트 기록에서 진행 중/해소 상태를 구분해 볼 수 있음 – 비활성화: 임곗값을 초과할 때마다 단발성 이벤트 발생
이벤트 수신 태그	해당 이벤트 알림을 받을 사용자 또는 그룹(수신 태그)

항목	설명
	이벤트 수정 및 삭제
	이벤트의 활성화 여부 – 활성화: 조건을 충족하면 이벤트 발생 – 비활성화: 이벤트가 동작하지 않으며, 조건을 검사하지 않음

이벤트 설정

복합 메트릭스 이벤트는 메트릭스 데이터 질의 언어인 **MXQL**을 기반으로 이벤트 조건을 생성합니다. **차트로 생성하기** 기능은 **MXQL**의 자동완성을 위한 콤보박스 기능을 제공합니다.

이벤트 추가

1. **경고 알림** > **이벤트 설정**에서 **복합 메트릭스** 탭을 클릭하세요.
2. 화면 오른쪽의 [+ 이벤트 추가] 버튼을 클릭하세요.
3. **복합 메트릭스** 창에서 차트로 생성하기를 클릭하세요.

   a. 제공하는 템플릿을 활용할 수 있습니다.
4. **이벤트 설정** 화면에서 먼저 이벤트 데이터를 조회하고, 이벤트 조건을 설정하세요.

이벤트 데이터 조회

이벤트 데이터를 조회하여 차트를 구성합니다. **이벤트 설정** 화면의 오른쪽 **이벤트 데이터 조회**의 **위젯** 또는 **텍스트** 옵션을 선택해 이벤트를 설정하세요.

- **위젯** **텍스트**

시계열 차트를 구성하는 옵션을 통해 이벤트 설정 시 사용할 **MXQL**을 자동 완성할 수 있습니다.

1. 데이터를 조회할 날짜를 선택하세요.
2. 데이터를 조회할 **카테고리**(대상)를 선택하세요.
3. **필터** 항목의 **+ 필터 추가**를 클릭해 이벤트 조건을 생성하세요.

a. 필터 창에서 연산식, 태그, 필터 값을 선택하세요.
b. + **추가**를 클릭해 조건을 추가할 수 있습니다.
c. 조건을 삭제하고 싶다면 조건 옆의 – 를 클릭하세요.

4. **그룹화** 항목에서 그룹화된 메트릭스 데이터를 선택하세요. 다중 선택할 수 있습니다.
5. **타임 유닛** 항목에서 `초`, `분`, `시간` 단위로 그룹화된 데이터를 나눌 시간 기준을 설정하세요.
6. **필드** 항목에서 이벤트 발행 조건에 사용할 필드를 선택하세요. 다중 선택할 수 있습니다.

> ⚠ 텍스트 모드로 입력한 mxql은 위젯 모드와 호환되지 않습니다.

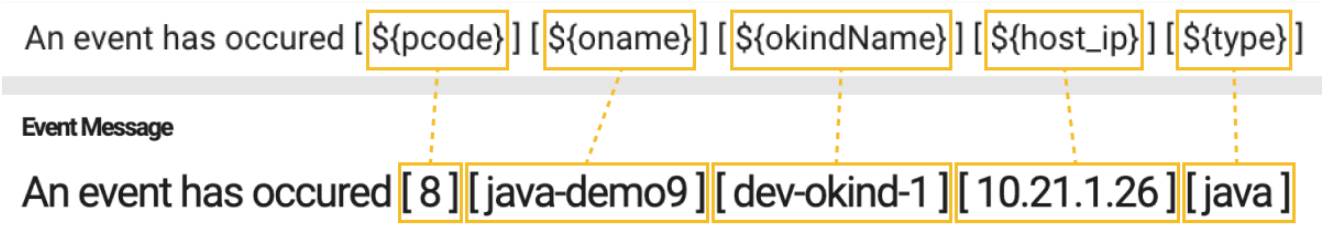
MXQL을 그대로 수정할 수 있는 편집창이 나타납니다.

1. **MXQL**를 직접 수정하고 **조회** 버튼을 클릭하세요.
2. 날짜를 선택하면 해당 내역을 차트 또는 그래프로 확인할 수 있습니다.

알림

경고 알림 설정의 기본 정보를 입력합니다.

- **이벤트 활성화**: 토글 버튼을 활성화하면 이벤트가 동작합니다.
- **레벨**: 이벤트 레벨(Critical, Warning, Info)을 선택하세요.
  - **이벤트 상태가 해소되면 추가 알림**: 토글 버튼을 활성화하면 이벤트가 해소되어 정상(RECOVERED) 상태의 알림이 전송됩니다.
- **제목**: 이벤트 규칙의 제목을 입력하세요.
- **메시지**: 이벤트 발생 시 전달받을 알림 메시지의 내용을 입력하세요. 메시지에 변수를 사용하여 이벤트 발생 시점의 실제 값을 포함할 수 있습니다.



변수 사용 규칙

- 변수 형식: `${지표명}`
- 설정한 **카테고리와 동일한 카테고리** 내 지표만 변수로 사용 가능합니다.
- 여러 변수를 조합해 상세한 메시지를 구성할 수 있습니다.
- 사용 가능한 변수 목록은 **메트릭스 조회**에서 확인할 수 있습니다.

이벤트 발행 조건

- **데이터 조회 범위:** 실시간 데이터 조회 범위를 선택하세요.
  - 조회 가능 범위: `30초` , `1분` , `2분` , `5분` , `10분` , `30분` , `1시간` , `2시간` , `3시간`
- **조건:** 이벤트 발생 조건을 입력하세요. 이벤트 데이터 조회에 포함된 필드만 사용할 수 있습니다.

부가정보

- **인터벌**: 선택한 시간(`1분` , `5분` , `10분` , `30분` , `1시간` , `12시간`) 간격으로 알림 조건을 확인합니다.
- **무음**: 이벤트가 발생한 후, 선택한 시간(`unset` , `1분` , `5분` , `10분` , `30분` , `1시간` , `12시간`) 동안 같은 이벤트가 발생되지 않습니다.
- **이벤트 동작 시간**: 이벤트가 특정 시간대(근무/비근무/점검 시간 등)에 동작하도록 태그를 설정하세요. 태그를 설정하지 않으면 이벤트가 활성화된 상태에서 24시간 상시 동작합니다.
  1. **+ 추가**를 클릭하세요.
  2. **이벤트 동작 시간**에서 **+ 새로운 태그 생성**을 클릭하세요.
  3. **이벤트 동작 시간 태그 생성**에서 **태그 이름, 요일, 시간, 색상**을 선택하고 [적용] 버튼을 클릭하세요.
  4. 생성한 태그는 태그 목록에서 확인할 수 있으며, 체크 박스를 선택하면 적용됩니다.
  5. 수정 또는 삭제할 태그의 아이콘을 클릭해 **이벤트 동작 시간 태그 수정**에서 태그를 수정하거나 태그 삭제할 수 있습니다.

> 태그 삭제 시 해당 태그가 적용된 모든 사용자의 태그가 제거됩니다. 단, 이벤트 규칙에서 사용 중인 태그는 삭제할 수 없습니다.

- **이벤트 수신 태그**: 이벤트에 수신 태그를 설정하면, 해당 태그를 가진 프로젝트 멤버에게 알림을 전송합니다. 수신 태그를

설정하지 않으면 프로젝트 내 모든 사용자에게 알림이 전송됩니다.

1. **+ 태그 추가** 또는 +를 클릭하세요.
2. **이벤트 수신 태그** 창 아래의 **+ 새 태그 생성**을 클릭하세요.
3. **태그 생성** 창에서 태그 이름을 입력하고 색상을 선택한 후, [태그 생성] 버튼을 클릭해 태그를 생성하세요.
4. 생성된 태그는 태그 목록의 아이콘을 클릭해 수정 또는 삭제하세요.
5. **이벤트 수신 태그** 창에서 **태그 목록**에서 원하는 태그를 선택하면 적용됩니다.

> 이벤트 설정 시 이벤트 수신 태그를 선택하여 해당 태그를 가진 프로젝트 멤버와 3rd-party 플러그인에 알림을 전송할 수 있습니다. **경고 알림** > **이벤트 수신 설정**에서 프로젝트 멤버와 3rd-party 플러그인에 각각 태그를 지정할 수 있습니다.

이벤트 규칙 테스트

설정한 조건을 과거 데이터에 기반으로 테스트를 진행해, 몇 건의 이벤트가 발생할지 예측할 수 있습니다.

1. 이벤트 규칙을 테스트할 날짜를 선택하고 [확인] 버튼을 클릭하세요.
2. [실행] 버튼을 클릭해 이벤트 발생 건수를 확인하세요.

이벤트 수정/삭제

1. **경고 알림** > **이벤트 수신 설정**의 **복합 메트릭스** 탭으로 이동하세요.
2. 복합 메트릭스 목록에서 수정 또는 삭제하려는 이벤트의 아이콘을 클릭하세요.
3. **이벤트 수신 설정** 화면에서 옵션을 수정하고, [저장] 버튼을 클릭하세요.

   a. 선택한 이벤트를 삭제하려면 **이벤트 수신 설정** 화면의 오른쪽 위의 [삭제] 버튼을 클릭하세요.

이벤트 공유

메트릭스 이벤트 설정을 JSON 파일로 저장해 다른 사용자와 설정을 공유하거나 다른 사용자의 설정을 가져올 수 있습니다.

- JSON 파일명: event-rules- `YYYYMMDD` .json

내보내기

1. 화면 오른쪽 위에 [**JSON**] 버튼을 클릭하세요.
2. JSON 편집 창이 나타나면 [내보내기] 버튼을 클릭하세요.
   - 이벤트를 검색 후 내보내기 기능을 이용하면 검색한 목록만 JSON 파일로 다운받을 수 있습니다.
3. JSON 파일이 다운로드되면 공유할 다른 사용자에게 전달하세요.

가져오기

1. 화면 오른쪽 위에 [ ] 버튼을 선택하세요.
2. **내보내기** 기능을 통해 다운로드한 JSON 파일을 선택하세요.
3. JSON 편집 창이 나타나면 [목록에 추가하기] 또는 [덮어쓰기] 버튼을 선택하세요.

같은 종류의 제품 간에 이용할 것을 권장합니다. 다른 제품의 프로젝트로부터 이벤트 설정을 가져올 수는 있지만 정상 작동하지 않습니다.

Json 형식으로 수정하기

1. 화면 오른쪽 위에 [**JSON**] 버튼을 클릭하세요.
2. 편집 창이 나타나면 JSON 형식에 맞춰 내용을 수정하세요.
3. 수정을 완료하면 화면 아래 [저장] 버튼을 선택하세요.

수정한 내용이 JSON 형식에 맞지 않으면 화면 아래에 에러 메시지가 표시되며, 저장할 수 없습니다. 표시되는 에러 메시지는 형식에 따라 다를 수 있습니다.

```
Expected ',' or '}' after property value in JSON at position 1964 (line 80 column 42)
```

알림 메시지 설정

수신할 메시지의 상세 항목을 선택하고, 항목별 내용을 편집할 수 있습니다.

부록

MXQL 문법 가이드

복합 메트릭스는 MXQL 쿼리를 기반으로 동작합니다. 자세한 문법은 MXQL 문법 가이드를 참고하시기 바랍니다.

히트맵 패턴

홈 > 프로젝트 선택 > **경고 알림** > **이벤트 설정** > **히트맵 패턴** 탭

히트맵 패턴은 애플리케이션 실행 구간(응답 시간 분포, 호출 밀도 등)의 시간대별 패턴 변화를 기반으로 이상 징후를 자동 탐지하는 기능입니다.

복잡한 조건이나 임계값을 직접 설정하지 않아도, 시스템이 실시간으로 수집한 히트맵에서 정상 패턴과 다른 이상 패턴을 감지하면 자동으로 경고 알림이 발생합니다. 히트맵 패턴 경고 알림은 다음 상황에서 유용합니다.

- 특정 트랜잭션이 평소보다 특정 시간대에만 과도하게 몰리는 경우
- 트래픽 분포나 응답 패턴이 평소와 달라진 경우
- 조건 설정만으로는 포착하기 어려운 패턴 기반 이상 탐지

이벤트 설정

탐지할 이벤트 패턴의 옵션을 설정하세요.

활성화

토글을 켜면 해당 패턴이 감지될 때 경고 알림이 전송됩니다.

- **활성화**: 조건을 충족하면 이벤트 발생
- **비활성화**: 이벤트가 발생하지 않으며, 조건을 검사하지 않음

무음

이벤트 발생 후, 선택한 시간 동안 이벤트가 발생되지 않고 **이벤트 기록**에 남지 않습니다.

이벤트 수신 태그

이벤트에 수신 태그를 설정하면, 해당 태그를 가진 프로젝트 멤버에게 알림을 전송합니다. 수신 태그를 설정하지 않으면 프로젝트 내 모든 사용자에게 알림이 전송됩니다.

1. **+ 태그 추가** 또는 +를 클릭하세요.
2. **이벤트 수신 태그** 창 아래의 **+ 새 태그 생성**을 클릭하세요.
3. **태그 생성** 창에서 태그 이름을 입력하고, 색상을 선택한 후, [태그 생성] 버튼을 클릭해 태그를 생성하세요.
4. 생성된 태그는 태그 목록의 아이콘을 클릭해 수정 또는 삭제하세요.
5. **이벤트 수신 태그** 창에서 **태그 목록**에서 원하는 태그를 선택하면 적용됩니다.

이벤트 동작 시간

이벤트가 특정 시간대(근무/비근무/점검 시간 등)에 동작하도록 태그를 설정하세요. 태그를 설정하지 않으면 이벤트가 활성화된 상태에서 24시간 상시 동작합니다.

1. **+ 추가**를 클릭하세요.
2. **이벤트 동작 시간**에서 **+새로운 태그 생성**을 클릭하세요.
3. **이벤트 동작 시간 태그 생성**에서 **태그 이름, 요일, 시간, 색상**을 선택하고 [적용] 버튼을 클릭하세요.
4. 생성한 태그는 태그 목록에서 확인할 수 있으며, 체크 박스를 선택하면 적용됩니다.
5. 수정 또는 삭제할 태그의 아이콘을 클릭해 **이벤트 동작 시간 태그 수정**에서 태그를 수정하거나 태그 삭제할 수 있습니다.

> 태그 삭제 시 해당 태그가 적용된 모든 사용자의 태그가 제거됩니다. 단, 이벤트 규칙에서 사용 중인 태그는 삭제할 수 없습니다.

히트맵 패턴 유형

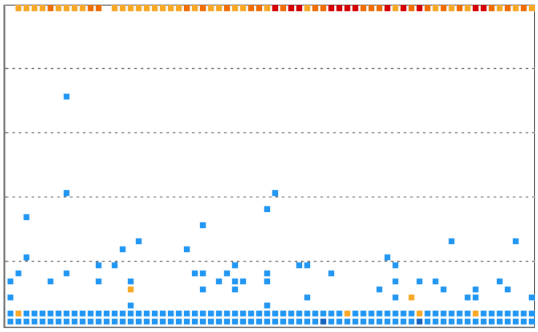
각 패턴의 토글을 켜면 해당 패턴이 감지되었을 때 이벤트가 발생됩니다.

히트맵 가로라인 패턴

여러 트랜잭션이 비슷한 응답 시간 구간에서 일괄적으로 종료되는 패턴입니다. 특정 응답 시간대에 몰림 현상이 발생하며, 서비스의 지연 증가 또는 외부 자원 응답 불안정을 의미합니다.

- 외부 API/DB/HTTPS 호출에서 Timeout 또는 지연 발생

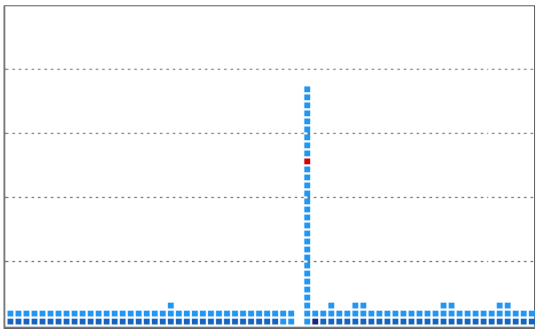
- 특정 리소스를 사용하려던 요청이 일정 시간 대기 후 한번에 처리되는 경우
- 공통 I/O 또는 네트워크 구간에서 반복적인 지연 발생



히트맵 세로라인 패턴

트랜잭션의 시작 시점은 다르지만 같은 시점에 동시에 종료되는 패턴입니다. 여러 요청이 특정 순간에 한 번에 처리되며 공통 자원의 병목(Bottleneck)을 의미합니다.

- DB Lock, File Lock 등 공유 자원에서 대기 시간 증가
- Queue 또는 Thread Pool에서 동시 처리 지연
- GC Pause 등 시스템 정지로 인해 요청이 잠시 멈췄다가 동시에 재개

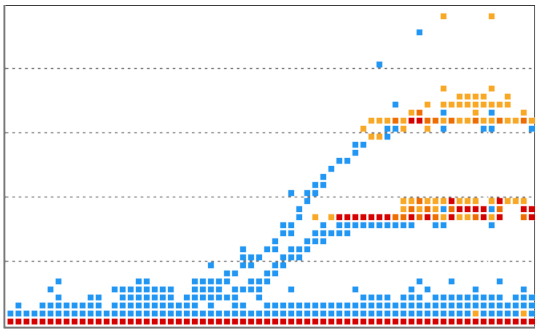


히트맵 세로라인 패턴은 락 경쟁 또는 병목 자원의 상태 점검이 필요한 신호입니다.

히트맵 플라잉 패턴

특정 리소스나 로그와 같은 공통 자원 부족으로 응답 시간이 일정 간격을 두고 파도처럼 오르내리는 패턴입니다.

- GC, 커넥션 풀, 파일 핸들 등 공통 리소스 부족
- 로그 flush, 스토리지 I/O 등 주기적 지연 요소 존재
- 배치성 작업 또는 백그라운드 job이 주기적으로 리소스 점유

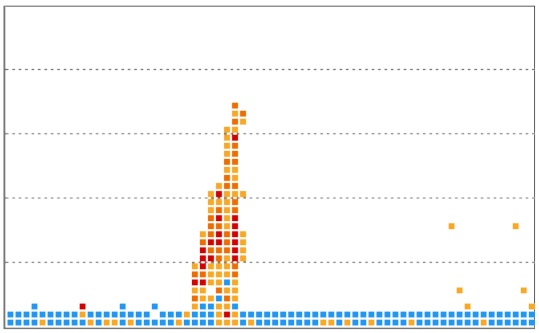


> 히트맵 플라잉 패턴은 주기적 병목 또는 리소스 부족의 가능성이 높습니다.

히트맵 과부하 패턴

특정 응답 시간대에 트랜잭션이 비정상적으로 밀집되는 패턴입니다. 일시적인 부하 또는 장애 전조로 인해 요청이 정상적으로 분산되지 못하고 특정 구간에 집중될 때 나타납니다.

- 일시적 장애 또는 지연으로 인해 트랜잭션이 대기열에 쌓였다가 일괄 처리
- 백엔드 서비스 응답 지연
- 특정 시간대 트래픽 집중으로 처리 여유 부족

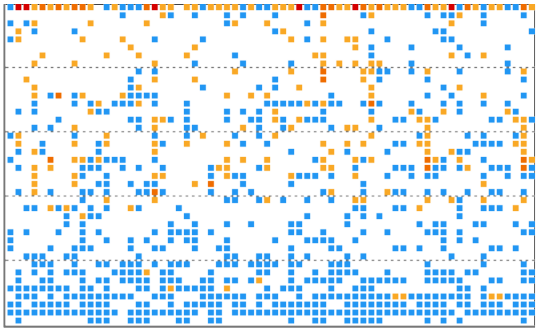


> 과부하 패턴은 일시적 서비스 품질 저하 또는 부하 분산 실패를 의미합니다.

히트맵 폭주 패턴

과도한 요청 또는 부하로 인해 전체 응답 시간이 전반적으로 증가하는 패턴입니다. 서비스 전반의 성능 저하가 발생하고 있음을 의미합니다.

- 사용자 트래픽 또는 자동화 트래픽 급증
- CPU, 메모리, GC 등 시스템 자원 부족
- 백엔드 병목으로 전체 응답 품질이 악화되는 경우



폭주 패턴은 즉각적인 점검이 필요합니다.

이상치 탐지

홈 > 프로젝트 선택 > **경고 알림** > **이벤트 설정** > **이상치 탐지** 탭

이상치 탐지는 메트릭스 데이터에서 비정상적인 상승 또는 하락 패턴을 식별해, 평소와 다른 움직임이 감지되면 이벤트를 발생시키는 기능입니다.

기본 옵션

- **검색**: 이벤트 이름 또는 이벤트 수신 태그를 선택해 원하는 이벤트 검색
- **+ 이벤트 규칙 추가**: 새로운 이상치 탐지 이벤트 추가

표 | 이상치 탐지 이벤트 목록 구성

항목	설명
이벤트 메시지	이벤트 발생 시 전송되는 메시지
카테고리	메트릭스 데이터를 구분하는 단위로 메트릭스 이벤트 설정 기준이 되는 카테고리
필드	이상치를 감지할 지표(메트릭스 필드)
필터	이상치 탐지 대상이 될 데이터 범위를 제한하는 조건
상승/하락 패턴	상승 또는 하락 패턴의 민감도
이벤트 수신 태그	해당 이벤트 알림을 받을 사용자 또는 그룹(수신 태그)
	이벤트 설정 수정
	이벤트 삭제
	이벤트의 활성화 여부 - 활성화: 조건을 충족하면 이벤트 발생 - 비활성화: 이벤트가 동작하지 않으며, 조건을 검사하지 않음

이벤트 설정

메트릭스의 데이터를 기반으로 값의 움직임이 평소와 다르게 상승하거나 하락할 때 경고 알림을 보내도록 설정할 수 있습니다.

이벤트 규칙 추가

1. 화면 오른쪽의 [+ 이벤트 규칙 추가] 버튼을 클릭하세요.
2. **이벤트 기본 정보, 메트릭스, 이상치 탐지, 부가정보**를 차례로 설정하세요.

이벤트 기본 정보

1. 이벤트 레벨(Critical, Warning, Info)을 선택하세요.
   - 이벤트의 중요도와 알림 레벨이 결정됩니다.
2. 이벤트 발생 시 전달받을 알림 메시지를 입력하세요.
   - 메시지에 변수를 사용하여 이벤트 발생 시점의 실제 값을 포함할 수 있습니다.

> ⓘ **사용 가능 변수**
>
> - `$pcode` : 프로젝트 코드
> - `$category` : 카테고리
> - `$field` : 필드
> - `$value` : 발생값
> - `$oname` : 이벤트 대상

메트릭스

1. **카테고리**를 선택하세요.
   - 이상치를 감지할 메트릭스 데이터의 기준이 되는 카테고리입니다.
2. 이벤트 발생 조건에 사용할 **필드**를 선택하세요. 다중 선택할 수 있습니다.

3. 이벤트 대상을 필터링할 이벤트 조건 대상인 필터를 선택하세요. 다중 선택할 수 있습니다.

이상치 탐지

1. **상승 태턴**과 **하락 패턴**의 민감도를 선택하세요.
2. 각 패턴의 토글 버튼을 클릭해 켜거나 끌 수 있습니다.
   - 상승 패턴만 감지하고 싶은 경우: 상승 패턴 **ON**, 하락 패턴 **OFF**
   - 하락 패턴만 감지하고 싶은 경우: 하락 패턴 **ON**, 상승 패턴 **OFF**

부가 정보

이벤트 동작 시간

이벤트가 특정 시간대(근무/비근무/점검 시간 등)에 동작하도록 태그를 설정하세요. 태그를 설정하지 않으면 이벤트가 활성화된 상태에서 24시간 상시 동작합니다.

1. **+ 추가**를 클릭하세요.
2. **이벤트 동작 시간**에서 **+새로운 태그 생성**을 클릭하세요.
3. **이벤트 동작 시간 태그 생성**에서 **태그 이름, 요일, 시간, 색상**을 선택하고 [적용] 버튼을 클릭하세요.
4. 생성한 태그는 태그 목록에서 확인할 수 있으며, 체크 박스를 선택하면 적용됩니다.
5. 수정 또는 삭제할 태그의 ✎ 아이콘을 클릭해 **이벤트 동작 시간 태그 수정**에서 태그를 수정하거나 삭제할 수 있습니다.

> ⚠ 태그 삭제 시 해당 태그가 적용된 모든 사용자의 태그가 삭제됩니다. 단, 이벤트 규칙에서 사용 중인 태그는 삭제할 수 없습니다.

이벤트 수신 태그

이벤트 수신 태그를 설정하면, 해당 태그를 가진 프로젝트 멤버에게 알림을 전송합니다. 수신 태그를 설정하지 않으면 프로젝트 내 모든 사용자에게 알림이 전송됩니다.

1. **+ 태그 추가** 또는 +를 클릭하세요.
2. **이벤트 수신 태그** 창 아래의 **+ 새 태그 생성**을 클릭하세요.

3. **태그 생성** 창에서 태그 이름을 입력하고, 색상을 선택한 후, [태그 생성] 버튼을 클릭해 태그를 생성하세요.
4. 생성된 태그는 태그 목록의 아이콘을 클릭해 수정 또는 삭제하세요.
5. **이벤트 수신 태그** 창의 **태그 목록**에서 원하는 태그를 선택하면 적용됩니다.

> 이벤트 설정 시 이벤트 수신 태그를 선택하여 해당 태그를 가진 프로젝트 멤버와 3rd-party 플러그인에 알림을 전송할 수 있습니다. **경고 알림** > **이벤트 수신 설정**에서 프로젝트 멤버와 3rd-party 플러그인에 각각 태그를 지정할 수 있습니다.

이벤트 수정

1. **경고 알림** > **이벤트 설정**의 **이상치 탐지** 탭으로 이동하세요.
2. **이상치 탐지** 목록에서 수정할 이벤트의 오른쪽 아이콘 클릭하세요.
3. **이상치 탐지** 창에서 옵션을 수정하고, [저장] 버튼을 클릭하세요.

이벤트 삭제

1. **경고 알림** > **이벤트 설정**의 **이상치 탐지** 탭으로 이동하세요.
2. **이상치 탐지** 목록에서 수정할 이벤트의 오른쪽 아이콘 클릭하세요.
3. 삭제 확인 창에서 [삭제] 버튼을 클릭하세요.

실시간 로그 이벤트

홈 > 프로젝트 선택 > 경고 알림 > **이벤트 설정** > **실시간 로그** 탭

수집한 로그 데이터를 조건에 맞춰 필터링하려 경고 알림을 설정할 수 있습니다. 로그 이벤트를 사용하려면 로그 모니터링이 활성화되어야 합니다. **로그 모니터링 활성화**에 대한 자세한 내용은 각 제품 로그 문서를 참고하세요.

- **실시간 로그 이벤트**: 실시간으로 수집한 로그에서 설정한 이벤트 조건이 충족되면 이벤트가 발생합니다.

기본 옵션

- **검색**: 이벤트 이름 또는 이벤트 수신 태그를 선택해 원하는 이벤트 검색
- **JSON 다운로드/업로드**: 이벤트 규칙은 JSON 편집기를 통해 바로 수정하거나, 파일로 다운로드하여 수정 후 다시 업로드 가능
- **알림 메시지 설정**: 수신할 메시지의 상세 항목 선택 및 항목 별 내용 편집
- **+ 이벤트 추가**: 새로운 로그 이벤트 추가

표 | 실시간 로그 이벤트 목록 구성

항목	설명
이벤트 이름	알림 제목으로 사용되는 이벤트 이름 - 최대 255byte까지 입력 가능
카테고리	로그 구분 명칭(로그 폴더명)
검색 키	로그 데이터에서 찾고 싶은 항목 - 예: HTTP 응답 상태 코드를 나타내는 값에 접근하고자 할 경우 검색 키 `status`
검색 값	검색 키에 해당하는 실제 데이터로 로그에서 입력한 단어를 포함할 경우 경고 알림을 보냄 - 예: 검색 키 `status` 검색 값 `200`을 설정한 경우 HTTP 응답 상태 코드 200을 포함하는 로그 데이터 수집 시 경고 알림 발생
이벤트 발생 일시 중지	이벤트 알림 발생 후, 같은 이벤트 발생을 일정 시간 동안 멈춤 - 선택 가능 시간: 사용 안 함, 5분, 10분, 15분, 20분, 30분, 1시간, 2시간, 3시간, 6시간, 12시간, 1일 - 해소된 알림이 활성화된 경우, 정상(RECOVERED) 상태 알림을 받은 후 선택한 시간 동안 같은 이벤트가

항목	설명
	발생해도 알림이 발생하지 않음
이벤트 대상 필터링	이벤트가 적용될 대상의 조건 - 대상 조건을 설정하지 않으면 수집되는 모든 데이터에 대해 알림 여부 판단 - 선택 입력: 태그, 연산자, 값을 선택해 조건 구성 - 직접 입력: 조건을 직접 입력해 대상 지정
이벤트 수신 태그	해당 이벤트 알림을 받을 사용자 또는 그룹(수신 태그)
	이벤트 수정 및 삭제
	이벤트의 활성화 여부 - 활성화: 조건을 충족하면 이벤트 발생 - 비활성화: 이벤트가 동작하지 않으며, 조건을 검사하지 않음

이벤트 설정

로그 이벤트는 실시간 로그 또는 일정 조건을 만족하는 로그가 누적될 때 경고 알림을 설정할 수 있는 기능입니다.

실시간 로그 이벤트 추가

1. **경고 알림** > **이벤트 설정**의 **실시간 로그** 탭을 클릭하세요.
2. 화면 오른쪽의 [+ 이벤트 추가] 버튼을 클릭하세요.
3. **이벤트 규칙 추가** 창에서 로그 이벤트 조건을 설정하세요.
    - **이벤트 이름**: 이벤트 제목으로 사용할 이름을 입력하세요.
    - **이벤트 활성화**: 토글 버튼 활성화 시, 토글 버튼 활성화 시, 조건을 충족하면 이벤트 발생합니다.
    - **레벨**: 이벤트 레벨(Critical, Warning, Info)을 선택하세요.
    - **메시지**: 이벤트 발생 시 전달받을 알림 메시지를 입력하세요. 메시지에 변수를 사용하여 이벤트 발생 시점의 실제 값을 포함할 수 있습니다.

> **변수 사용 규칙**
>
> - 변수 형식: `${지표명}`
> - 설정한 **카테고리와 동일한 카테고리** 내 지표만 변수로 사용할 수 있습니다.
> - 여러 변수를 조합해 상세한 메시지를 구성할 수 있습니다.
> - 사용 가능한 변수 목록은 **메트릭스 조회**에서 확인할 수 있습니다.

- **카테고리**: 로그 구분 명칭(로그 폴더명)을 목록에서 선택하거나 직접 입력하세요. 카테고리를 선택해야 검색 키와 검색 값을 입력할 수 있습니다.
- **검색 키**: 로그 데이터에서 찾고 싶은 항목을 입력하세요.
  - 예: HTTP 응답 상태 코드를 나타내는 값에 접근하고자 할 경우 **검색 키** `status`
- **검색 값**: **검색 키**에 해당하는 실제 데이터를 입력하세요. 로그에서 입력한 단어를 포함할 경우 경고 알림을 보냅니다.
  - 예: **검색 키** `status` **검색 값** `200` 을 설정한 경우 HTTP 응답 상태 코드 200을 포함하는 로그 데이터 수집 시 경고 알림 발생
- **이벤트 대상 필터링**: 이벤트가 적용될 대상의 조건을 입력하세요. 입력값이 없으면, 수집되는 모든 데이터에 대해 알림 여부 판단합니다.
  - **선택 입력**: 태그, 연산자, 값을 선택해 대상 조건을 설정하세요.

    > 조건식 작성 방법과 지원되는 연산자는 조건 설정 가이드를 참고하세요.

  - **직접 입력**: 조건을 직접 입력하세요.
- **이벤트 발생 일시 중지**: 알림을 일시 중지할 시간을 선택하세요. 이벤트 발생 후 설정한 시간 동안 동일한 이벤트가 발생하지 않습니다.
  - 선택 가능 시간: `사용 안 함` , `1분` , `2분` , `3분` , `5분` , `10분` , `20분` , `30분` , `1시간` , `3시간` , `6시간` , `12시간`
- **이벤트 동작 시간**: 이벤트가 특정 시간대(근무/비근무/점검 시간 등)에 동작하도록 태그를 설정하세요. 태그를 설정하지 않으면 이벤트가 활성화된 상태에서 24시간 상시 동작합니다.

1. **+ 추가**를 클릭하세요.
2. **이벤트 동작 시간**에서 **+새로운 태그 생성**을 클릭하세요.
3. **이벤트 동작 시간 태그 생성**에서 **태그 이름, 요일, 시간, 색상**을 선택하고 [적용] 버튼을 클릭하세요.
4. 생성한 태그는 태그 목록에서 확인할 수 있으며, 체크 박스를 선택하면 적용됩니다.
5. 수정 또는 삭제할 태그의 아이콘을 클릭해 **이벤트 동작 시간 태그 수정**에서 태그를 수정하거나 태그 삭제할 수 있습니다.

> 태그 삭제 시 해당 태그가 적용된 모든 사용자의 태그가 제거됩니다. 단, 이벤트 규칙에서 사용 중인 태그는 삭제할 수 없습니다.

- **이벤트 수신 태그**: 이벤트에 수신 태그를 설정하면, 해당 태그를 가진 프로젝트 멤버에게 알림을 전송합니다. 수신자 태그를 설정하지 않으면 프로젝트 내 모든 사용자에게 알림이 전송됩니다.

  1. **+ 태그 추가** 또는 **+**를 클릭하세요.
  2. **이벤트 수신 태그** 창 아래의 **+ 새 태그 생성**을 클릭하세요.
  3. **태그 생성** 창에서 태그 이름을 입력하고, 색상을 선택한 후, [태그 생성] 버튼을 클릭해 태그를 생성하세요.
  4. 생성된 태그는 태그 목록의 아이콘을 클릭해 수정 또는 삭제하세요.
  5. **이벤트 수신 태그** 창의 **태그 목록**에서 원하는 태그를 선택하면 적용됩니다.

  > 이벤트 설정 시 이벤트 수신 태그를 선택하여 해당 태그를 가진 프로젝트 멤버와 3rd-party 플러그인에 알림을 전송할 수 있습니다. **경고 알림 > 이벤트 수신 설정**에서 프로젝트 멤버와 3rd-party 플러그인에 각각 태그를 지정할 수 있습니다.

4. **이벤트 규칙 추가** 창에서 설정이 완료되면 [저장] 버튼을 클릭하세요.

이벤트 수정/삭제

1. **경고 알림** > **이벤트 설정**의 **실시간 로그** 탭으로 이동하세요.

2. 로그 이벤트 목록에서 수정 또는 삭제하려는 이벤트의 아이콘을 클릭하세요.
3. **이벤트 설정** 화면에서 옵션을 수정하고, [저장] 버튼을 클릭하세요.

   a. 선택한 이벤트를 삭제하려면 **이벤트 설정** 화면의 오른쪽 위의 [삭제] 버튼을 클릭하세요.

이벤트 공유

메트릭스 이벤트 설정을 JSON 파일로 저장해 다른 사용자와 설정을 공유하거나 다른 사용자의 설정을 가져올 수 있습니다.

- JSON 파일명: event-rules-`YYYYMMDD`.json

내보내기

1. 화면 오른쪽 위에 [**JSON**] 버튼을 클릭하세요.
2. JSON 편집 창이 나타나면 [내보내기] 버튼을 클릭하세요.
   - 이벤트를 검색 후 내보내기 기능을 이용하면 검색한 목록만 JSON 파일로 다운받을 수 있습니다.
3. JSON 파일이 다운로드되면 공유할 다른 사용자에게 전달하세요.

가져오기

1. 화면 오른쪽 위에 [ ] 버튼을 선택하세요.
2. **내보내기** 기능을 통해 다운로드한 JSON 파일을 선택하세요.
3. JSON 편집 창이 나타나면 [목록에 추가하기] 또는 [덮어쓰기] 버튼을 선택하세요.

> 같은 종류의 제품 간에 이용할 것을 권장합니다. 다른 제품의 프로젝트로부터 이벤트 설정을 가져올 수는 있지만 정상 작동하지 않습니다.

Json 형식으로 수정하기

1. 화면 오른쪽 위에 [**JSON**] 버튼을 클릭하세요.
2. 편집 창이 나타나면 JSON 형식에 맞춰 내용을 수정하세요.

3. 수정을 완료하면 화면 아래 [저장] 버튼을 선택하세요.

> ⓘ 수정한 내용이 JSON 형식에 맞지 않으면 화면 아래에 에러 메시지가 표시되며, 저장할 수 없습니다. 표시되는 에러 메시지는 형식에 따라 다를 수 있습니다.
>
> ```
> Expected ',' or '}' after property value in JSON at position 1964 (line 80 column 42)
> ```

부록

발생 조건, 대상 선택 가이드

메트릭스 경고 알림의 이벤트 발생 조건과 이벤트 대상 선택은 동일한 문법을 사용합니다. 단, 이벤트 발생 조건은 필드(Field)의 Key를 변수로 사용하고, 이벤트 대상 선택은 태그(Tag)의 Key를 변수로 사용합니다.

레벨 동작 가이드

이벤트 레벨은 위험(Critical)과 경고(Warning), 정상(Info)으로 구분합니다.

1. **우선순위 기반 이벤트 발생**

   여러 레벨의 조건을 동시에 충족할 경우, 가장 높은 우선순위의 이벤트만 발생합니다.

   예시

   ```
   설정:
   - Warning: CPU > 70%
   - Critical: CPU > 90%

   현재 상태: CPU 95%
   → 결과: Critical 이벤트만 발생 (Warning은 억제됨)
   ```

2. **레벨 상승**

   진행중 상태를 갖는 이벤트에서 낮은 레벨 이벤트가 진행 중일 때 높은 레벨 조건을 충족하면, 두 레벨 모두 진행 중

상태가 됩니다.

예시

```
설정:
- Warning: CPU > 70%
- Critical: CPU > 90%

시나리오:
1) CPU 80% → Warning 발생 (진행 중)
2) CPU 95% → Warning (진행 중), Critical (진행 중)
```

3. 레벨 하강

진행중 상태를 갖는 이벤트에서 높은 레벨에서 낮은 레벨로 전환되면 높은 레벨은 해소되고 낮은 레벨이 유지됩니다.

예시

```
설정:
- Warning: CPU > 70%
- Critical: CPU > 90%

시나리오:
1) CPU 97% → Critical 발생
2) CPU 85% → Critical 해소, Warning 유지
3) CPU 65% → Warning 해소
```

4. 동작 흐름도

```
- Warning: CPU > 70%
- Critical: CPU > 90%

메트릭 값: 60%  →  75%  →  92%  →  85%  →  60%
상태:      정상     Warning   Critical   Warning   정상
                    (발생)    (발생)     (유지)    (해소)
                              Warning
                              (유지)
```

진행 중 상태를 갖는 이벤트는 조건이 충족되는 동안 계속 활성화 상태를 유지하며, 조건이 해제될 때까지 지속됩니다.

조건 설정 가이드

지표명이 숫자로 시작하거나 특수문자가 포함된 경우 `${지표명}` 형태로 작성해야 합니다.

- 괄호 `()`, `[]`
- 연산자 `+`, `-`, `*`, `/`, `%`
- 구분자 `:`, `@`, `#`, `,`, (공백)
- 기타 특수문자 `!`, `^`, `&`, `|`, `~`, ```, `=`

`${}` 사용 예시:

```
${cpu(xos)} > 50
${mem[0]} >= 100
${cpu-usage} > 80
${namespace:cpu} > 70
${metric name} > 60
${4xx_error} > 10
```

`${}` 없이 사용 가능: 영문자, 언더스코어(`_`), 점(`.`)만 포함된 경우

```
cpu > 80
cpu_usage > 50
CPUUtilization.Average > 0.8
```

1. **비교 연산자**

   숫자 비교

```
크다:        cpu > 80
크거나 같다:    cpu >= 80
작다:        memory < 1000
작거나 같다:    memory <= 1000
같다:        status == 200
```

```
같지 않다:      error != 0
```

문자열 비교

```
status == 'OK'
region == "us-east-1"
```

2. 산술 연산자

- 기본 산술 연산

```
덧셈:   cpu + 10 >= 90
뺄셈:   memory - 100 >= 500
곱셈:   cpu * 2 >= 100
나눗셈:  disk / 1024 >= 100
나머지:  value % 10 == 0
```

- 괄호를 이용한 우선순위 제어

```
(cpu + memory) * 2 >= 200
(disk - used) / total >= 0.2
((cpu + memory) / 2) >= 50
```

- 음수 표현

```
cpu > -100
```

3. 논리 연산자

- AND 연산 (&&)

```
cpu > 80 && memory > 1000
${cpu(xos)} > 50 && ${mem(xos)} > 60
```

- OR 연산 (||)

```
cpu > 90 || memory > 90
disk < 10 || network > 1000
```

- 복합 논리 연산

```
(cpu > 50 && memory > 50) || disk < 20
cpu > 80 && (memory > 1000 || disk > 500)
```

4. 패턴 매칭 연산자

- LIKE 연산자

```
oname like 'prod-*'
url like '*error*'
message like 'WARN%'
```

- NOT LIKE 연산자

```
oname not like 'test-*'
url not like '*debug*'
```

5. 내장 함수

- Null 체크 함수

```
isNull(value)          # null인지 확인
isNotNull(value)       # null이 아닌지 확인
nvl(value, 0)          # null이면 기본값 반환
isEmpty(str)           # 빈 문자열인지 확인
isNotEmpty(str)        # 빈 문자열이 아닌지 확인
```

- 집계 함수

```
sum(cpu, memory, disk)          # 합계
avg(cpu, memory)                # 평균
max(cpu, memory, disk)          # 최대값
```

```
min(cpu, memory, disk)            # 최소값
count(value1, value2, value3)     # 개수
```

- 수학 함수

```
round(cpu, 2)          # 반올림 (소수점 2자리)
```

- 문자열 함수

```
length(str)              # 문자열 길이
startsWith(str, 'prefix')  # 접두사 확인
endsWith(str, 'suffix')    # 접미사 확인
indexOf(str, 'search')     # 문자열 위치
substring(str, 0, 10)      # 부분 문자열
trim(str)                # 공백 제거
replace(str, 'old', 'new') # 문자열 치환
hasStr(str, 'search')      # 문자열 포함 여부
```

문자열 함수

```
equalsIgnoreCase(str, 'match_text')    # 대소문자를 구분하지 않는 일치 비교
notEqualsIgnoreCase(str, 'match_text')  # 대소문자를 구분하지 않는 불일치 비교
```

`==` 와 `!=` 는 대소문자를 구분하므로 `WebServer` 와 `webserver` 를 다른 값으로 판단합니다. 표기가 일정하지 않은 태그나 필드를 비교할 때 위 두 함수를 사용하면 대소문자 차이를 무시하고 값을 맞출 수 있습니다.

예시

```
equalsIgnoreCase(okindName, 'WebServer')
notEqualsIgnoreCase(container, 'PROD.billing')
```

이벤트 발생 조건과 이벤트 대상 선택은 같은 문법을 사용하므로 두 함수를 양쪽 조건식에 모두 작성할 수 있습니다.

- 조건 함수
  - **if:** 조건식이 참(`true`)이면 `true값` 을 반환하고, 거짓(`false`)이면 `false값` 을 반환하는 조건 함수

형식

```
if(조건, true값, false값)
```

예시

```
if(value >= 90, 'High', 'Medium') == 'High'
```

- **decode**: 값과 조건을 비교하여 일치하는 조건의 결과를 반환, 없으면 기본값 반환

형식

```
decode(값, 조건1, 결과1, 조건2, 결과2, ..., 조건N, 결과N, 기본값)
```

예시

```
decode(value, 1, 'Low', 2, 'Medium', 3, 'High', 'Unknown') == 'Unknown'
```

- **in**: 값이 지정한 후보값 중 하나와 일치하면 `true`, 아니면 `false`

형식

```
in(값, 후보값1, 후보값2, ..., 후보값N)
```

예시

```
in(status, 200, 201, 204) == false
```

작성 시 주의사항

- 권장사항
  - 간단한 변수명은 `${}` 없이 사용 (예: cpu, memory)
  - 특수문자 포함 시 반드시 `${}` 사용
  - 복잡한 연산은 괄호로 명확하게 구분
  - 함수명은 정확히 입력 (대소문자 구분)

- 피해야 할 사항
  - 불완전한 표현식 작성 (예: `cpu >` , `memory &&`)
  - 괄호 미스매칭 (예: `(cpu > 80 && cpu > 70)`)
  - 연속된 연산자 사용 (예: `cpu >> 80` , `cpu >< 80`)

복합 로그 이벤트 설정

홈 > 프로젝트 선택 > **경고 알림** > **이벤트 설정** > **복합 로그** 탭

수집한 로그 데이터를 조건에 맞춰 필터링하려 경고 알림을 설정할 수 있습니다. 로그 이벤트를 사용하려면 로그 모니터링이 활성화되어야 합니다. **로그 모니터링 활성화**에 대한 자세한 내용은 (각 제품 로그 문서 링크)를 참고하세요.

- **복합 로그 이벤트**: 설정한 **데이터 조회 범위** 내 로그를 기준으로, 로그 검색 조건을 충족하는 로그 수가 임계 개수 이상일 경우 이벤트가 발생합니다.

> **권한**
>
> 복합 메트릭스 이벤트를 설정하려면 **알림 설정** 권한이 필요합니다. 알림 설정 권한은 **홈 화면 > 프로젝트 선택 > 관리 > 통합 멤버 관리**에서 권한을 부여할 멤버의 아이콘을 클릭해 설정할 수 있습니다.

기본 옵션

- **검색**: 이벤트 이름 또는 이벤트 수신 태그를 선택해 원하는 이벤트 검색
- **JSON 다운로드/업로드**: 이벤트 규칙은 JSON 편집기를 통해 바로 수정하거나, 파일로 다운로드하여 수정 후 다시 업로드 가능
- **알림 메시지 설정**: 수신할 메시지의 상세 항목 선택 및 항목 별 내용 편집
- **+ 이벤트 추가**: 새로운 로그 이벤트 추가

표 | 복합 로그 이벤트 목록 구성

항목	설명
이벤트 이름	알림 제목으로 사용될 이벤트 이름
템플릿	복합 로그 템플릿
카테고리	로그 구분 명칭(로그 폴더명)
규칙	이벤트 발생하는 조건

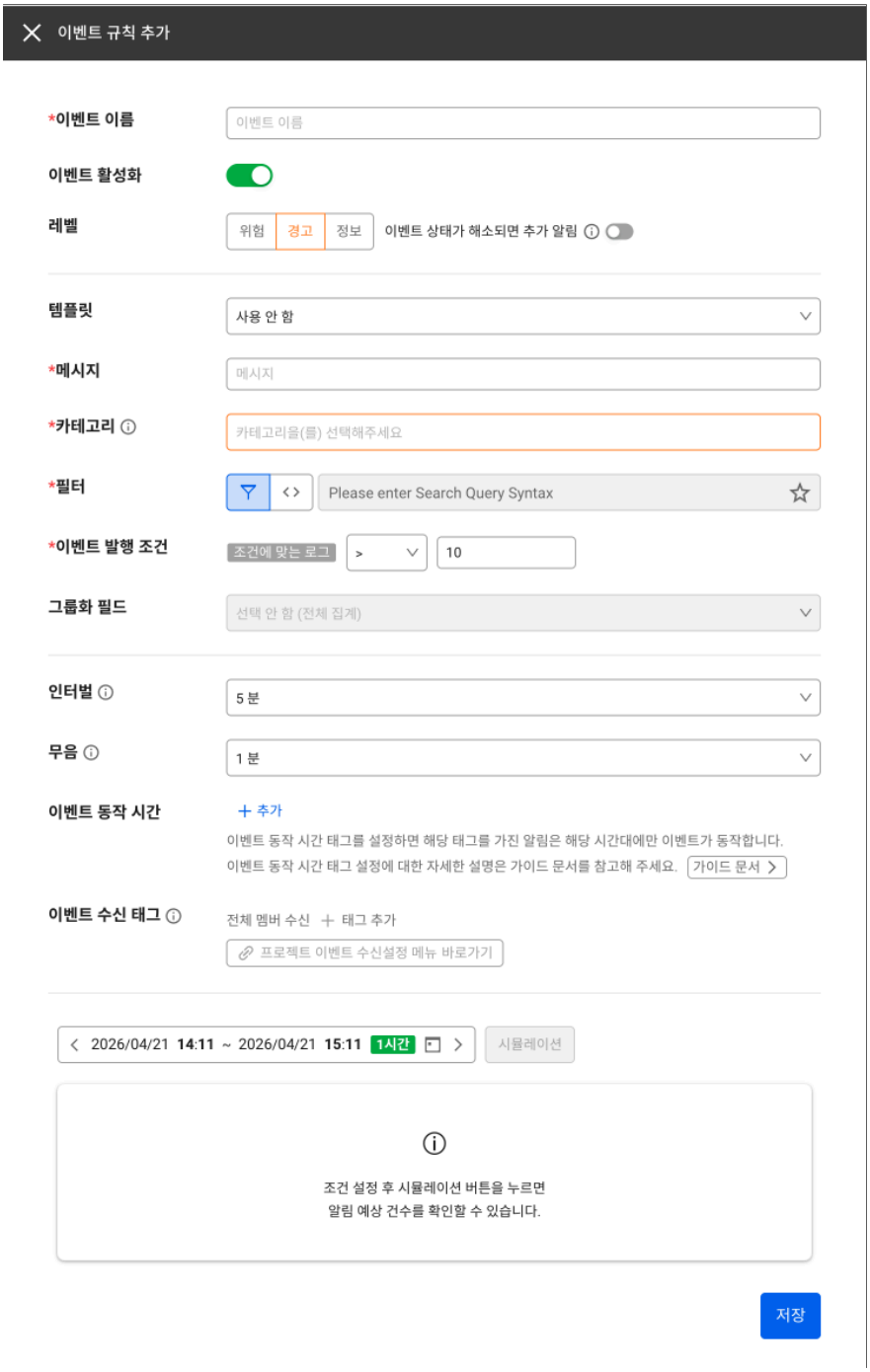
항목	설명
이벤트 상태가 해소되면 추가 알림	Critical과 Warning 레벨의 이벤트가 해소되면 정상(RECOVERED) 상태의 알림 수신 여부 - 활성화: 진행 중 → 정상 상태로 전환될 때 RECOVERED 알림을 보내며, 이벤트 기록에서 진행 중/해소 상태를 구분해 볼 수 있음 - 비활성화: 임계값을 초과할 때 마다 단발성 이벤트 발생
이벤트 발생 일시 중지	이벤트 알림 발생 후, 같은 이벤트 발생을 일정 시간 동안 멈춤 - 선택 가능 시간: 사용 안 함, 5분, 10분, 15분, 20분, 30분, 1시간, 2시간, 3시간, 6시간, 12시간, 1일 - 해소된 알림이 활성화된 경우, 정상(RECOVERED) 상태 알림을 받은 후 선택한 시간 동안 같은 이벤트가 발생해도 알림이 발생하지 않음
인터벌	로그 데이터를 조회할 시간 범위
이벤트 수신 태그	해당 이벤트 알림을 받을 사용자 또는 그룹(수신 태그)
	이벤트 수정 및 삭제
	이벤트의 활성화 여부 - 활성화: 조건을 충족하면 이벤트 발생 - 비활성화: 이벤트가 동작하지 않으며, 조건을 검사하지 않음

이벤트 설정

복합 로그 이벤트는 설정한 **데이터 조회 범위** 내 로그를 기준으로, 로그 검색 조건을 충족하는 로그 수가 임계 개수 이상일 경우 경고 알림을 설정할 수 있는 기능입니다.

복합 로그 이벤트 추가

1. **경고 알림** > **이벤트 설정**의 **복합 로그** 탭을 클릭하세요.
2. 화면 오른쪽의 [+ 이벤트 추가] 버튼을 클릭하세요.
3. **이벤트 규칙 추가** 창에서 로그 이벤트 조건을 설정하세요. 자세한 내용은 기본 설정, 상세 설정, 추가 설정를 참고하세요.



4. **이벤트 규칙 추가** 창에서 설정이 완료되면 [저장] 버튼을 클릭하세요.

기본 설정

- **이벤트 이름**: 이벤트 제목으로 사용할 이름을 입력하세요.
- **이벤트 활성화**: 토글 버튼 활성화 시, 조건을 충족하면 이벤트 발생합니다.
- **레벨**: 이벤트 레벨(Critical, Warning, Info)을 선택하세요.

상세 설정

템플릿(프리셋), 메시지, 카테고리, **필터**, 이벤트 발생 조건 항목을 설정합니다.

템플릿(프리셋)

미리 정의된 프리셋을 선택하면 **필터, 쿼리 타입, 이벤트 발행 조건**이 자동으로 설정됩니다. 프리셋을 선택한 후 필요에 맞게 수정할 수 있습니다.

복합 로그 템플릿 유형

프리셋	쿼리 타입	기본 규칙
사용 안 함	none	`rows > 10`
2xx 상태코드 건수	count	`count > 10`
3xx 상태코드 건수	count	`count > 10`
4xx 상태코드 건수	count	`count > 10`
5xx 상태코드 건수	count	`count > 10`
정상 상태코드(2xx, 3xx) 건수	count_v2	`include_minus_exclude_count > 10`
에러 상태코드(4xx, 5xx) 건수	count_v2	`include_minus_exclude_count > 10`
에러 수신 건수	count	`count > 10`
예외 수신 건수	count	`count > 10`

프리셋	쿼리 타입	기본 규칙
초당 로그 발생건수	rps	`rps > 100`
필드 집계 (Aggregation)	aggr	`avg > 1000`
비율 (Rate)	rate	`rate > 10`
고유값 수 (Cardinality)	cardinality	`cardinality > 100`
백분위 (Percentile)	percentile	`p99 > 3000`

메시지

이벤트 발생 시 전달받을 알림 메시지를 입력하세요. 메시지에 변수를 사용하여 동적 메시지를 구성할 수 있습니다.

- 예시: 메시지 `${host}`에서 에러 `${count}`건 발생 → 알림 메시지 web-01에서 에러 15건 발생

표 | 메시지 변수

변수	설명	사용 조건
`${count}`	조건에 맞는 로그 건수	항상 사용 가능
`${groupField}`	그룹화 필드 이름	그룹화 알림만
`${groupValue}`	그룹화 필드 값	그룹화 알림만
`${필드명}`	그룹화 필드 값 (필드명으로 직접 참조)	그룹화 알림만

`${count}` 변수는 그룹화 여부와 관계없이 모든 알림에서 사용할 수 있습니다. `${groupField}`, `${groupValue}`, `${필드명}` 변수는 그룹화 필드를 설정한 경우에만 치환됩니다.

카테고리

로그 구분 명칭(로그 폴더명)을 목록에서 선택하거나 직접 입력하세요. 카테고리를 선택해야 검색 키와 검색 값을 입력할 수 있습니다.

필터

Lucene 쿼리 문법으로 로그 검색 조건을 입력합니다. 두 가지 입력 모드를 지원합니다.

- **필터 모드** : 필드와 값을 선택하여 쿼리를 구성합니다.
- **코드 모드** <>: Lucene 쿼리를 직접 입력합니다.

표 | Lucene 쿼리 예시

쿼리	설명
`level:ERROR`	level 필드가 ERROR인 로그
`status_nxx:status_4xx OR status_nxx:status_5xx`	4xx 또는 5xx 상태코드
`appender:accessLog`	accessLog appender의 로그
`response_time.n:*`	response_time.n 필드가 존재하는 로그
`host:web-01 AND level:ERROR`	web-01 호스트의 에러 로그

이벤트 발생 조건

쿼리 타입별로 설정합니다. 선택한 템플릿(프리셋) 또는 쿼리 타입에 따라 추가 설정 항목이 표시됩니다.

표 | 이벤트 발행 조건 키 의미

키	의미
`rows` / `count`	필터 조건에 맞는 로그 건수
`rps`	초당 로그 발생건수 (건수 / 인터벌 초)

키	의미
avg	대상 필드 값의 평균
sum	대상 필드 값의 합계
min	대상 필드 값의 최솟값
max	대상 필드 값의 최댓값
rate	분자 건수 / 분모 건수 × 100 (%)
cardinality	대상 필드의 고유한 값 개수
p99 / p95 / p90 / p75 / p50	대상 필드의 백분위 값
include_minus_exclude_count	포함 필터 건수 - 제외 필터 건수

- **건수 (count / none)**: 설정한 필터 조건에 맞는 로그의 건수를 카운트합니다.

표 | 건수 설정 항목

설정 항목	설명
필터	로그 검색 조건 (Lucene 쿼리)
이벤트 발행 조건	건수 또는 조건에 맞는 로그 {연산자} {임계값} (예: count > 10)

- **초당 로그 발생건수 (rps)**: 설정한 필터 조건에 맞는 로그의 초당 발생건수(Requests Per Second)를 계산합니다. 인터벌 시간 동안의 총 건수를 인터벌 초로 나누어 산출합니다.

표 | 초당 로그 발생건수 설정 항목

설정 항목	설명
필터	로그 검색 조건 (Lucene 쿼리)
이벤트 발행 조건	RPS {연산자} {임계값} (예: rps > 0.5)

- **필드 집계 (aggr)**: 지정한 숫자 필드의 집계 값(평균, 합계, 최솟값, 최댓값)을 계산합니다. 이벤트 발행 조건에서 집계 방식을 드롭다운으로 선택할 수 있습니다.

표 | 필드 집계 설정 항목

설정 항목	설명
필터	로그 검색 조건 (Lucene 쿼리)
대상 필드	집계 대상 숫자 필드 (예: `response_time.n`)
이벤트 발행 조건	`평균` / `합계` / `최솟값` / `최댓값` `{연산자}` `{임계값}` (예: `avg > 1000`)

- **비율 (rate)**: 분자 필터와 분모 필터의 건수 비율(%)을 계산합니다. 전체 로그 대비 특정 조건의 비율을 모니터링할 때 유용합니다.

표 | 비율 설정 항목

설정 항목	설명
필터 (분자)	비율 계산의 분자가 되는 로그 조건 (예: `level:ERROR`)
분모 필터	비율 계산의 분모가 되는 로그 조건 (예: `level:*`)
이벤트 발행 조건	`비율 (%)` `{연산자}` `{임계값}` (예: `rate > 10`)

  - **예시**: 필터 `level:ERROR`, 분모 필터 `level:*`, rule `rate > 10` → 전체 로그 중 에러 비율이 10%를 초과하면 알림 발생

- **고유값 수 (cardinality)**: 지정한 필드의 고유한 값 개수를 계산합니다. 비정상적으로 많은 종류의 값이 발생하는 경우를 감지할 때 유용합니다.

표 | 고유값 수 설정 항목

설정 항목	설명
필터	로그 검색 조건 (Lucene 쿼리)
대상 필드	고유값을 계산할 필드 (예: `host`)

설정 항목	설명
이벤트 발행 조건	`고유값 수` `{연산자}` `{임계값}` (예: `cardinality > 100`)

- **백분위 (percentile)**: 지정한 숫자 필드의 백분위(Percentile) 값을 계산합니다. 응답 시간의 tail latency를 모니터링할 때 유용합니다. Percentile 선택을 변경하면 이벤트 발행 조건의 키가 자동으로 업데이트됩니다.

표 | 백분위 설정 항목

설정 항목	설명
필터	로그 검색 조건 (Lucene 쿼리)
대상 필드	백분위를 계산할 숫자 필드 (예: `response_time.n`)
Percentile	P99, P95, P90, P75, P50 중 선택
이벤트 발행 조건	`P99` / `P95` /... `{연산자}` `{임계값}` (예: `p99 > 3000`)

표 | 백분위 의미

백분위	의미
P50	중앙값. 전체 데이터의 50%가 이 값 이하
P75	전체 데이터의 75%가 이 값 이하
P90	전체 데이터의 90%가 이 값 이하
P95	전체 데이터의 95%가 이 값 이하
P99	전체 데이터의 99%가 이 값 이하

- **포함/제외 건수 (count_v2)**: 포함 필터의 건수에서 제외 필터의 건수를 뺀 순수 건수를 계산합니다.

표 | 포함/제외 건수 설정 항목

설정 항목	설명
필터 (포함)	포함할 로그 조건 (예: `status_nxx:status_4xx OR status_nxx:status_5xx`)
필터 제외	제외할 로그 조건 (예: `status_nxx:status_4xx`)
분모 필터	비율 계산용 분모 (선택사항)
이벤트 발행 조건	`포함-제외 건수` `{연산자}` `{임계값}` (예: `include_minus_exclude_count > 10`)

- **예시**: 필터 `status_nxx:status_4xx OR status_nxx:status_5xx`, 필터 제외 `status_nxx:status_4xx` → 4xx+5xx에서 4xx를 제외한 순수 5xx 건수가 10을 초과하면 알림 발생

그룹화 필드(Group By)

그룹화 필드를 설정하면 해당 필드의 값별로 독립적으로 알림 조건을 판정합니다. 드롭다운에서 그룹화할 필드를 선택합니다. 드롭다운을 클릭하면 현재 카테고리에서 사용 가능한 필드 목록이 자동완성됩니다.

- **예시:** 필터 `level:ERROR`, 그룹화 필드 `host`, rule `rows > 3`

표 | 그룹화 설정 비교

항목	그룹화 없음	그룹화 사용
판정 단위	전체 로그를 하나로 집계	그룹 값별 독립 집계
알림 발생	1건	조건 충족 그룹 수만큼
알림 제목	`이벤트 제목`	`[그룹 값] 이벤트 제목`
Stateful	전체 기준 상태 전이	그룹별 독립 상태 전이

- `[web-01] 에러 알림` — web-01에서 에러 8건 → 알림 발생
- `[web-02] 에러 알림` — web-02에서 에러 5건 → 알림 발생
- `[api-01] 에러 알림` — api-01에서 에러 2건 → 조건 미충족, 알림 미발생

> **이벤트 상태가 해소되면 추가 알림**을 활성화한 경우, 그룹별로 독립적으로 ABNORMAL → RECOVERED 상태가 전이됩니다. 예를 들어 `web-01` 에서 에러가 해소되면 `[web-01] RECOVERED` 알림이 발생하며, 다른 호스트의 상태에는 영향을 미치지 않습니다.

추가 설정

- **인터벌**: 선택한 시간(`unset` , `1분` , `5분` , `10분` , `30분` , `1시간` , `12시간`) 간격으로 알림 조건을 확인합니다.
- **무음**: 이벤트 발생 후, 선택한 시간(`unset` , `1분` , `5분` , `10분` , `30분` , `1시간` , `12시간`) 동안 같은 이벤트가 발생되지 않습니다.
- **이벤트 동작 시간**: 이벤트가 특정 시간대(근무/비근무/점검 시간 등)에 동작하도록 태그를 설정하세요. 태그를 설정하지 않으면 이벤트가 활성화된 상태에서 24시간 상시 동작합니다.
    1. **+ 추가**를 클릭하세요.
    2. **이벤트 동작 시간**에서 **+새로운 태그 생성**을 클릭하세요.
    3. **이벤트 동작 시간 태그 생성**에서 **태그 이름, 요일, 시간, 색상**을 선택하고 [적용] 버튼을 클릭하세요.
    4. 생성한 태그는 태그 목록에서 확인할 수 있으며, 체크 박스를 선택하면 적용됩니다.
    5. 수정 또는 삭제할 태그의 ✎ 아이콘을 클릭해 **이벤트 동작 시간 태그 수정**에서 태그를 수정하거나 태그 삭제할 수 있습니다.

> 태그 삭제 시 해당 태그가 적용된 모든 사용자의 태그가 제거됩니다. 단, 이벤트 규칙에서 사용 중인 태그는 삭제할 수 없습니다.

- **이벤트 수신 태그**: 이벤트에 수신 태그를 설정하면, 해당 태그를 가진 프로젝트 멤버에게 알림을 전송합니다. 수신자 태그를 설정하지 않으면 프로젝트 내 모든 사용자에게 알림이 전송됩니다.
    1. **+ 태그 추가** 또는 **+**를 클릭하세요.
    2. **이벤트 수신 태그** 창 아래의 **+ 새 태그 생성**을 클릭하세요.
    3. **태그 생성** 창에서 태그 이름을 입력하고, 색상을 선택한 후, [태그 생성] 버튼을 클릭해 태그를 생성하세요.
    4. 생성된 태그는 태그 목록의 ✎ 아이콘을 클릭해 수정 또는 삭제하세요.
    5. **이벤트 수신 태그** 창의 **태그 목록**에서 원하는 태그를 선택하면 적용됩니다.

> 이벤트 설정 시 이벤트 수신 태그를 선택하여 해당 태그를 가진 프로젝트 멤버와 3rd-party 플러그인에 알림을 전송할 수 있습니다. **경고 알림 > 이벤트 수신 설정**에서 프로젝트 멤버와 3rd-party 플러그인에 각각 태그를 지정할 수 있습니다.

시뮬레이션

알림을 저장하기 전에 과거 데이터로 시뮬레이션하여 알림 발생 예상 건수를 확인할 수 있습니다.

1. 알림 설정을 완료한 후 하단의 시간 범위를 선택하세요.
2. [시뮬레이션] 버튼을 클릭하세요.
3. 차트에 시간대별 데이터와 알림 발생 횟수가 표시됩니다.

   > 시뮬레이션 시간 범위를 조절하여 다양한 시간대의 알림 발생 패턴을 확인할 수 있습니다. 최근 1~2분의 데이터는 인덱스 빌드 지연으로 조회되지 않을 수 있습니다.

4. **이벤트 규칙 추가** 창에서 설정이 완료되면 [저장] 버튼을 클릭하세요.

로그 탐색기에서 알림 생성

로그 탐색기에서 직접 알림을 생성할 수 있습니다.

1. **로그** > **로그 탐색** 메뉴로 이동하세요.
2. 카테고리와 검색 쿼리를 설정하세요.
3. 화면 오른쪽 위의 [이벤트 추가] 버튼을 클릭하세요.
4. Drawer에서 알림 설정 폼이 열리며, 현재 카테고리와 검색 쿼리가 자동으로 채워집니다.
5. 추가 설정을 완료하고 [저장] 버튼을 클릭하세요.
6. 저장 완료 후 [이동] 버튼을 클릭하면 알림 목록으로 이동합니다.

이벤트 수정/삭제

1. **경고 알림** > **이벤트 설정**의 **복합 로그** 탭으로 이동하세요.
2. 로그 이벤트 목록에서 수정 또는 삭제하려는 이벤트의 아이콘을 클릭하세요.
3. **이벤트 설정** 화면에서 옵션을 수정하고, [저장] 버튼을 클릭하세요.

   a. 선택한 이벤트를 삭제하려면 **이벤트 설정** 화면의 오른쪽 위의 [삭제] 버튼을 클릭하세요.

이벤트 공유

메트릭스 이벤트 설정을 JSON 파일로 저장해 다른 사용자와 설정을 공유하거나 다른 사용자의 설정을 가져올 수 있습니다.

- JSON 파일명: event-rules-`YYYYMMDD`.json

내보내기

1. 화면 오른쪽 위에 [**JSON**] 버튼을 클릭하세요.
2. JSON 편집 창이 나타나면 [내보내기] 버튼을 클릭하세요.
   - 이벤트를 검색 후 내보내기 기능을 이용하면 검색한 목록만 JSON 파일로 다운받을 수 있습니다.
3. JSON 파일이 다운로드되면 공유할 다른 사용자에게 전달하세요.

가져오기

1. 화면 오른쪽 위에 [ ] 버튼을 선택하세요.
2. **내보내기** 기능을 통해 다운로드한 JSON 파일을 선택하세요.
3. JSON 편집 창이 나타나면 [목록에 추가하기] 또는 [덮어쓰기] 버튼을 선택하세요.

> 같은 종류의 제품 간에 이용할 것을 권장합니다. 다른 제품의 프로젝트로부터 이벤트 설정을 가져올 수는 있지만 정상 작동하지 않습니다.

Json 형식으로 수정하기

1. 화면 오른쪽 위에 [**JSON**] 버튼을 클릭하세요.
2. 편집 창이 나타나면 JSON 형식에 맞춰 내용을 수정하세요.
3. 수정을 완료하면 화면 아래 [저장] 버튼을 선택하세요.

> 수정한 내용이 JSON 형식에 맞지 않으면 화면 아래에 에러 메시지가 표시되며, 저장할 수 없습니다. 표시되는 에러 메시지는 형식에 따라 다를 수 있습니다.
>
> ```
> Expected ',' or '}' after property value in JSON at position 1964 (line 80 column 42)
> ```

트랜잭션

홈 > 프로젝트 선택 > **경고 알림** > **이벤트 설정** > **트랜잭션** 탭

트랜잭션 이벤트는 특정 URL, 에러, 상태 코드 등을 기준으로 트랜잭션을 세밀하게 모니터링하는 기능입니다. 설정한 조건을 만족하는 트랜잭션이 감지되면 이벤트가 발생합니다. 트랜잭션 데이터를 1분 단위로 분석하며, 조건을 충족하는 트랜잭션 수가 임계값을 초과할 경우 알림을 전송합니다.

최신 에이전트 버전

트랜잭션 이벤트를 사용하려면 최소 지원 버전 이상의 에이전트가 필요합니다.

언어	최소 지원 버전
Java	2.2.50 버전 이상
PHP	버전 미정 (최신 버전 이상 사용 권장, 별도 문의 필요)
Node.js	0.5.21 버전 이상
Python	1.7.8 버전 이상
.NET	2.3.5 버전 이상
Go	버전 미정 (최신 버전 이상 사용 권장, 별도 문의 필요)

기본 옵션

- **이벤트 레벨**: 이벤트의 심각도를 의미하며 Info, Warning, Critical 세 단계로 구분
- **검색**: 이벤트 이름 또는 이벤트 수신 태그를 선택해 원하는 이벤트 검색
- **+ 이벤트 추가**: 새로운 트랜잭션 이벤트 추가
- **컬럼 크기 조정**: 각 컬럼의 경계를 드래그하여 너비 조정 가능

표 | 트랜잭션 이벤트 목록 구성

항목	설명
No.	이벤트의 순번
	이벤트의 활성화 여부 - 활성화: 조건을 충족하면 이벤트 발생 - 비활성화: 이벤트가 동작하지 않으며, 조건을 검사하지 않음
	이벤트 설정 수정 및 삭제
이벤트 이름	사용자가 지정한 이벤트명 - 최대 255byte까지 입력 가능
조건 유형	이벤트 발생 조건 유형(경과 시간, 에러 발생, 패턴만)
집계 기준	트랜잭션 수를 집계하는 기준
대상 패턴	이벤트가 적용되는 포함/제외 패턴 (URL, 상태 코드 등)
발생 횟수	이벤트 발생 기준이 되는 횟수 - 집계 주기: 1분 고정(매 정시 00~59초 기준, 기간 선택 불가) - 해당 1분 동안 조건을 충족한 트랜잭션이 설정한 횟수 이상 발생하면 이벤트 발생
일시 중지	이벤트 알림 발생 후, 같은 이벤트 발생을 일정 시간 동안 멈춤 - 선택 가능 시간: 사용 안 함, 5분, 10분, 15분, 20분, 30분, 1시간, 2시간, 3시간, 6시간, 12시간, 1일 - 해소된 알림이 활성화된 경우, 정상(RECOVERED) 상태 알림을 받은 후 선택한 시간 동안 같은 이벤트가 발생해도 알림이 발생하지 않음
이벤트 수신	해당 이벤트 알림을 받을 사용자 또는 그룹(수신 태그)

이벤트 설정

트랜잭션 이벤트는 **이벤트 조건, 대상 패턴, 기본 정보 및 수신 설정** 단계를 통해 구성됩니다. 이벤트 설정을 통해 특정 트랜잭션 패턴을 감지하고 알림을 받을 수 있습니다.

이벤트 추가

트랜잭션 이벤트를 추가하려면 **이벤트 설정** > **트랜잭션** 탭으로 이동한 후, 화면 오른쪽 상단의 [+ 이벤트 추가] 버튼을 클릭하세요.

트랜잭션 이벤트 추가 방법
Step 1. 이벤트 조건 정의: 이벤트의 기본 구조를 선택합니다.
Step 2. 이벤트 대상 선택: 이벤트가 발생하는 모니터링 대상을 선택합니다.
Step 3. 기본 정보 및 수신 설정: 이벤트 이름 및 메시지, 수신 대상을 설정합니다.

이벤트 조건 정의

트랜잭션 이벤트 발생 조건을 설정합니다.

조건 유형

이벤트를 발생시킬 기준을 선택하세요.

- **경과 시간:** 지정한 시간(ms) 이상 소요된 트랜잭션
  - **경과 시간 임계값**: 조건 유형이 경과 시간인 경우, ms 단위로 입력하세요. 설정한 시간(ms) 이상 소요된 트랜잭션이 발생할 경우 이벤트가 발생합니다.
- **에러 발생:** 에러가 발생한 트랜잭션
- **패턴만:** 특정 패턴(히트맵 패턴 등)에 해당하는 트랜잭션

집계 기준

알림 조건을 어떤 단위로 집계할지 선택하세요.

- 집계 기준: 에이전트별, 각 에이전트의 URL별, 각 에이전트의 에러 클래스별, 각 에이전트의 업무별, URL별, 에러 클래스별, 업무별, 전체

발생 횟수(필수)

조건을 충족한 트랜잭션이 몇 번 발생했을 때 알림을 보낼지 설정합니다.

예: 발생 횟수 **1회** → 1분 내 조건 충족 트랜잭션이 1건이라도 있으면 즉시 알림 / 발생 횟수 **3회** → 같은 1분 동안 조건 충족 트랜잭션이 3건 이상 누적되어야 알림

일시 중지

과도한 이벤트 알림을 방지하기 위해, 알림을 일시 중지할 시간을 선택하세요.

- 이벤트 발생 후 설정한 시간 동안 동일한 이벤트가 발생하지 않습니다.
- 선택 가능 시간: `1분` , `5분` , `10분` , `15분` , `30분` , `1시간` , `2시간` , `3시간` , `6시간` , `12시간` , `1일`

이벤트 동작 시간

이벤트가 어떤 시간대에 동작할지 설정합니다. **비근무 시간, 점검 시간대 등의 알림 발생을 제한**할 수 있습니다. 태그를 설정하지 않으면 이벤트가 활성화된 상태에서 24시간 상시 동작합니다.

1. **+ 추가**를 클릭하세요.
2. 이벤트 동작 시간에서 **+새로운 태그 생성**을 클릭하세요.
3. **이벤트 동작 시간 태그 생성**에서 **태그 이름, 요일, 시간, 색상**을 선택하고 [적용] 버튼을 클릭하세요.
4. 생성한 태그는 태그 목록에서 확인할 수 있으며, 체크 박스를 선택하면 적용됩니다.
5. 수정 또는 삭제할 태그의 ✎ 아이콘을 클릭해 **이벤트 동작 시간 태그 수정**에서 태그를 수정하거나 태그 삭제할 수 있습니다.

> ⚠ 태그 삭제 시 해당 태그가 적용된 모든 알림에서 삭제됩니다. 단, 이벤트 설정에서 사용 중인 태그는 삭제할 수 없습니다.

이벤트 대상 선택

트랜잭션 필터링을 위한 패턴을 설정합니다.

패턴 설정 방법

- `Enter`로 구분하여 여러 패턴을 입력할 수 있습니다.
- 별도로 설정하지 않으면, 프로젝트 내 모든 트랜잭션을 대상으로 이벤트가 발생합니다.
- 같은 유형의 패턴은 **OR** 조건으로 적용됩니다.
- 다른 유형의 패턴은 **AND** 조건으로 적용됩니다.
- 와일드카드(`*`)를 사용하여 패턴을 설정할 수 있습니다. 와일드카드 위치는 최대 2곳까지 허용됩니다.
- 와일드카드(`*`)와 슬래시(`/`) 외에 다른 문자가 함께 포함되어야 합니다.

> **슬래시(`/`)와 와일드카드(`*`) 조합에 주의하세요**
>
> - `/api/*` : `/api` 는 매칭되지 않음, `/api/` 뒤에 경로가 있어야 매칭 (예: ✅ /api/users, ❌ /api)
> - `/api*` : `/api` 자체도 매칭됨 (예: ✅ /api, ✅ /api/users, ✅ /apitest)

일치 패턴

지정한 항목과 일치하는 데이터를 필터링합니다. 일치 패턴은 적용 후 파란색 태그로 표시됩니다.
각 항목의 패턴은 최대 10개까지 입력할 수 있습니다.

항목	설명
업무명	업무명 패턴
URL	트랜잭션 URL 패턴 - 예: `/api/*`, `*/users/*`
에러 클래스	에러 클래스명 패턴 - 예: `*Exception`, `*Error`
에러 메시지	에러 메시지 내용 패턴
상태 코드	HTTP 상태 코드 패턴 - 예: `4**`, `5**`

제외 패턴

지정한 항목과 일치하는 데이터를 필터링 대상에서 제외합니다. 제외 패턴은 적용 후 빨간색 태그로 표시되고, 일치 패턴과 동일한 형식을 사용합니다. 각 항목의 패턴은 최대 10개까지 입력할 수 있습니다.

패턴 예시

- **완전 매칭**

```
패턴: "hello"
"hello"    → 일치
"Hello"    → 불일치 (대소문자 구분)
```

- **접두사 매칭**

```
패턴: "hello*"
"hello"      → 일치
"helloworld" → 일치
"hi"         → 불일치
```

- **접미사 매칭**

```
패턴: "*world"
"world"      → 일치
"helloworld" → 일치
"hello"      → 불일치
```

- **부분 문자열 매칭**

```
패턴: "*ell*"
"hello"      → 일치 (h'ell'o)
"excellent"  → 일치 (exc'ell'ent)
"welcome"    → 불일치 (ell이 포함되지 않음)
```

- **복합 패턴**

```
패턴: "hello*world"
"hellobeautifulworld"  → 일치

패턴: "hello*test*"
```

```
"hellomytestcase"    → 일치

패턴: "*test*world"
"mytestcaseworld"    → 일치
```

표 | 실제 사용 예시

패턴 유형	예시
URL 패턴	- `/api/*` : /api/로 시작하는 모든 URL - `/users/*` : users가 포함된 모든 URL - `/api/v1/users` : 정확히 일치하는 URL만
에러 클래스 패턴	- `Exception` : Exception으로 끝나는 모든 에러 - `java.lang.*` : java.lang으로 시작하는 모든 클래스 - `Timeout*` : Timeout이 포함된 모든 에러
상태 코드 패턴	- `4**` : 400번대 모든 상태 코드 - `404` : 404 상태 코드만 - `5**` : 500번대 모든 상태 코드

패턴 제한사항

- **필수 문자열**: 패턴에는 와일드카드(`*`) 외에 최소 1개 이상의 문자가 포함되어야 합니다. 와일드카드만으로는 패턴을 설정할 수 없습니다.
  - ❌ 불가: `**` , `*`
  - ✅ 가능: `a*` , `a` , `a*` , `/api*`
- **빈 문자열**: 빈 문자열이나 null 값은 매칭되지 않습니다.
- **연속된 와일드카드**: 연속된 `*` 는 하나의 와 동일하게 처리됩니다.
- **복잡한 패턴**: 3개 이상 포함된 패턴은 지원하지 않습니다.
- **정규식 미지원**: 문자 클래스(`[a-z]`), 수량자(`+` , `?`), 그룹화(`()`) 등은 지원하지 않습니다.
- **대소문자 구분**: 모든 패턴 매칭은 대소문자를 구분합니다.
- **이스케이프 불가**: 문자 자체를 매칭할 방법은 없습니다.

기본 정보 및 수신 설정

이벤트의 기본 정보와 알림 수신 방식을 설정합니다.

이벤트 활성화

이벤트가 발생했을 때 알림 전송 여부를 설정합니다.

- **활성화**: 설정한 조건을 충족하면 이벤트의 알림이 전송됨
- **비활성화**: 설정한 조건을 충족하더라도 알림이 전송되지 않음

이벤트 이름(필수)

알림 제목으로 사용될 이벤트 이름을 입력하세요.

이벤트 레벨(필수)

이벤트 레벨(Info, Warning, Critical)을 선택하세요.

이벤트 수신

이벤트 알림을 받을 멤버를 지정합니다.

- **전체 수신** **태그 선택 수신**

프로젝트에서 이벤트 수신 설정이 활성화된 모든 대상(멤버, 채널)에게 알림을 전송합니다.

해당 태그를 가진 프로젝트 멤버에게 알림을 전송합니다.

1. **태그 선택 수신**을 선택하세요.
2. **수신 태그** 항목에서 [+ 태그 추가] 또는 [+] 버튼을 클릭하세요.
3. **이벤트 수신 태그** 창 아래의 **+ 새 태그 생성**을 클릭하세요.
4. **태그 생성** 창에서 태그 이름을 입력하고, 색상을 선택한 후, [태그 생성] 버튼을 클릭해 태그를 생성하세요.
5. 생성된 태그는 태그 목록의 아이콘을 클릭해 수정 또는 삭제하세요.
6. **이벤트 수신 태그** 창에서 **태그 목록**에서 원하는 태그를 선택하면 적용됩니다.

> **수신 태그**
>
> 수신자 태그는 알림을 받을 사용자 또는 채널을 **그룹 단위로 관리하기 위한 기능**입니다. 이벤트에 수신자 태그를 설정하면, 해당 태그가 포함된 사용자에게만 알림이 전달됩니다. 수신 태그를 설정하지 않으면 프로젝트 내 모든 사용자에게 알림이 전송됩니다.

참고. 이벤트 동작 예시

트랜잭션 이벤트가 실제로 발생한 예시입니다.

> **설정 조건**
>
> - 집계 기준: 에이전트별
> - 발생 횟수: 2회
> - 일시 중지: 5분
> - URL 패턴: `/api/posts/**` , `/delete/*` , `/*test*/`
> - 에러 클래스 패턴: `*TimeoutException*` , `*NullPointerException`
> - 상태 코드 패턴: `4**`
> - 제외 에러 클래스: `*IOException`
> - 제외 상태 코드: `5**`

발생한 알림 메시지 예시

이벤트 메시지의 트랜잭션 정보는 조건에 부합한 마지막 트랜잭션의 값을 표시합니다. 이벤트 메시지는 설정한 조건과 실제 발생한 트랜잭션의 정보를 모두 포함하여, 어떤 조건에 의해 알림이 발생했는지 명확히 확인할 수 있습니다.

```
(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
```

이벤트 메시지 구성 설명

- **(ignored count: n)**: 일시 중지 시간 동안 조건에 부합했지만 알림이 발생하지 않은 트랜잭션 수입니다. 일시 중지 시간 동안 무시된 건이 없으면 이 항목은 표시되지 않습니다.
- **집계 정보**: 설정한 집계 방식에 따라 다르게 표시됩니다.
  - **에이전트별**: `[Per Agent] limit: n, count: n`
  - **URL별**: `[Per Url] limit: n` 후 각 URL과 발생 건수 나열
  - **에러 클래스별**: `[Per Error Class] limit: n` 후 각 에러 클래스와 발생 건수 나열
  - **컨텍스트별**: `[Per Context] limit: n` 후 각 컨텍스트와 발생 건수 나열
  - **각 에이전트의 URL별**: `[Per Agent & Url] limit: n` 후 각 URL과 발생 건수 나열
  - **각 에이전트의 에러 클래스별**: `[Per Agent & Error Class] limit: n` 후 각 에러 클래스와 발생 건수 나열
  - **에이전트별 컨텍스트별**: `[Per Agent & Context] limit: n` 후 각 컨텍스트와 발생 건수 나열
  - **전체**: `[Total] limit: n, count: n`
- **[Condition]**: 이벤트 발생 시점에 설정되어 있던 조건들을 표시합니다. 설정된 패턴만 표시됩니다.
  - **context patterns**: 컨텍스트 일치 패턴
  - **url patterns**: URL 일치 패턴
  - **error class patterns**: 에러 클래스 일치 패턴
  - **error message patterns**: 에러 메시지 일치 패턴
  - **status patterns**: 상태 코드 일치 패턴
  - **exclude context patterns**: 컨텍스트 제외 패턴
  - **exclude url patterns**: URL 제외 패턴
  - **exclude error class patterns**: 에러 클래스 제외 패턴
  - **exclude error message patterns**: 에러 메시지 제외 패턴

  - **exclude status patterns**: 상태 코드 제외 패턴

- **[Transaction]**: 조건에 부합한 마지막 트랜잭션의 상세 정보입니다.
  - **url**: 트랜잭션 URL
  - **txid**: 트랜잭션 고유 식별자
  - **status**: HTTP 응답 상태 코드 (HTTP 트랜잭션이 아닌 경우 미표시)
  - **error class**: 에러 클래스 (에러 클래스가 없는 경우 미표시)
  - **error message**: 에러 메시지 (에러 메시지가 없는 경우 미표시)

> **참고**
>
> - 트랜잭션 이벤트는 1분 단위로 집계되어 처리됩니다.
> - 조건 유형이 '패턴만'인 경우, 반드시 하나 이상의 일치 또는 제외 패턴을 설정해야 합니다.
> - 패턴 매칭은 대소문자를 구분합니다.
> - 알림 메시지는 조건에 부합한 마지막 트랜잭션의 상세 정보가 포함됩니다.

컨테이너

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

컨테이너의 자원 사용량과 진행 중 트랜잭션, 에러 상황의 이벤트 조건을 통해 경고 알림을 보냅니다.

> 이벤트 설정 *classic* 은 기존 방식의 이벤트 설정을 계속 사용할 수 있도록 분리된 메뉴입니다.
> 더 다양한 기능은 이벤트 설정에서 이용하실 수 있습니다.

기본 옵션

에이전트 비활성 경고

설정한 시간 동안 에이전트로부터 데이터를 수신하지 못한 경우 이벤트가 발생합니다. 토글 버튼을 클릭해 기능을 켜거나 끌 수 있습니다.

- **활성화**: 일정 기간 데이터 미수신 시 에이전트 비활성 이벤트 발생
- **비활성화**: 데이터 미수신 여부과 관계없이 이벤트 발생하지 않음

> 기존의 에이전트 비활성 확인 간격 설정은 **관리 > 에이전트 목록**에서 설정할 수 있습니다.

이벤트 상태가 해소되면 추가 알림

조건이 충족되어 이벤트가 발생한 후, 상태가 정상으로 복구되면 **해소(RECOVERED) 알림**을 보낼지 여부를 선택할 수 있습니다. 토글 버튼을 클릭해 기능을 켜거나 끌 수 있습니다.

- **활성화**: 이벤트는 상태 기반으로 동작하며, 임계치 초과 시 진행중, 임계치 이하로 복구 시 해소 상태로 전환됨
- **비활성화**: 임계값을 초과할 때 마다 단발성 이벤트 발생

에이전트 중복 알림

동일한 이름의 에이전트가 동시에 2개 이상 기동될 경우 정상적인 모니터링이 어려울 수 있어 이벤트가 필수적으로 발생합니다.
단, 설정한 시간 동안은 중복 에이전트 이벤트가 발생하지 않습니다.

- 에이전트 이름을 중복으로 설정하지 않았는지 확인하려면 **이벤트 설정** 메뉴에서 관련 옵션을 검토하세요.
- 선택 가능 시간: 30초 , 1분 , 3분 , 5분 , 10분

조건

항목별로 이벤트 발생 조건을 설정할 수 있으며, 항목에 따라 입력 형식이 다릅니다.

- **서버 CPU/서버 메모리/서버 디스크** **액티브 트랜잭션/트랜잭션 에러** **트랜잭션 응답시간**

조건(%)

Warning과 Critical 두 단계의 임계값을 설정하세요.
설정값을 초과 시 경고 알림이 발생하며, Critical이 우선 적용됩니다.

- 왼쪽 값: 경고(Warning) 임계값
- 오른쪽 값: 위험(Critical) 임계값

조건(건수)

이벤트 발생 기준이 되는 건수를 입력하세요. 지정된 지속 시간 동안 설정한 건수를 초과하는 상태가 유지되면 이벤트가 발생합니다.

조건(건수)

조건을 설정하세요. 설정한 응답시간을 초과한 트랜잭션이 지정 개수 이상 발생하면 이벤트 조건 충족합니다.

조건(응답시간)

기준이 되는 응답시간을 선택하세요.

- 선택 가능 시간: 1초 , 2초 , 3초 , 5초 , 10초 , 15초 , 30초 , 1분 , 2분 , 3분 , 5분 , 10분 , 30분 , 1시간
- 예시: 건수 기준이 **10건,** 응답시간 기준이 **1초,** 지속 시간이 **5초**인 경우 → 1초를 넘는 트랜잭션이 10건 이상 발생하고 그 상태가 5초 지속되면 이벤트가 발생합니다.

지속

선택한 시간 동안 이벤트 조건이 지속될 경우 이벤트가 발생합니다.

- 선택 가능 시간: `5초` , `10초` , `15초` , `30초` , `1분` , `2분` , `3분` , `5분` , `10분` , `30분` , `1시간`

무음

이벤트 발생 후, 선택한 시간 동안 동일한 조건의 이벤트가 발생하지 않습니다. 적절한 무음시간 설정으로 과도한 이벤트 발생을 방지할 수 있습니다.

- 선택 가능 시간: `1분` , `2분` , `3분` , `5분` , `10분` , `30분` , `1시간` , `2시간` , `3시간` , `6시간` , `12시간` , `1일`

이벤트 활성화

토글 버튼을 클릭해 이벤트의 동작을 제어할 수 있습니다.

- **활성화**: 설정한 조건을 충족하면 이벤트가 발생됨
- **비활성화**: 조건을 검사하지 않음

이벤트 설정

다음 항목에 이벤트를 설정하여 시스템 상태를 실시간으로 모니터링 할 수 있습니다. 각 항목을 설정한 후 화면 오른쪽 상단의 [저장] 버튼을 클릭하세요.

서버 CPU / 서버 메모리 /서버 디스크

서버 자원 사용량이 설정한 임계값(%)을 초과할 경우 이벤트가 발생합니다. Warning과 Critical 단계별로 다른 임계치를 설정할 수 있습니다.

- 예: CPU 사용률이 Warning: 80%, Critical: 90%로 설정되었을 경우, 90%를 초과하면 Critical 이벤트가 발생합니다.

액티브 트랜잭션

현재 처리 중인 트랜잭션의 수를 모니터링합니다. 설정한 기준 건수를 초과하고, 이 상태가 **지속 시간** 이상 유지되면 이벤트가

발생합니다. 대량의 트랜잭션이 동시에 처리되고 있는 상황을 감지하는 데 유용합니다.

- 발생 건수 기본값: 100개 기준

트랜잭션 에러

정상적으로 종료되지 않은 트랜잭션을 에러로 판단합니다. 에러 건수가 설정한 기준 건수를 초과하고, 해당 상태가 지속되면 이벤트가 발생합니다.

- 수집 주기: 5초

트랜잭션 응답시간

각 트랜잭션이 완료되는 데 걸린 시간(응답 시간)을 기준으로 이벤트를 설정합니다. 사용자가 기대하는 응답시간을 기준으로 느려진 서비스 감지를 설정할 수 있습니다.

- 예: 응답시간 기준을 10초, 건수 기준을 10건으로 설정한 경우
  “응답시간 10초를 넘긴 트랜잭션이 10건 이상 발생하고, 이 상태가 5초간 지속되면 이벤트 발생”

이벤트 수신 설정

사용자별 이벤트 수신 설정

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

수신 수단 설정

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

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

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

이메일 알림

이메일 알림은 회원 가입 시 입력한 이메일 주소로 알림을 보냅니다.

SMS 알림

SMS로 알림을 수신할 수 있습니다.

> SMS 알림 기능은 유료 프로젝트에 한정됩니다.

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

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

WhatsApp 알림

WhatsApp을 통해 알림을 수신할 수 있습니다.

> WhatsApp 알림 기능은 유료 프로젝트에 한정됩니다.

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

모바일 알림

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

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

반복 알림(에스컬레이션) 설정

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

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

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

이벤트 수신 태그

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

이벤트 수신 태그 추가/수정/삭제하기

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

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

이벤트 수신 태그 해제하기

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

수신 태그 미설정 알림 받기

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

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

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

이벤트 수신 모드 선택하기

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

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

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

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

선택한 태그만 받기

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

> ⓘ **선택한 태그만 받기** 모드에서 태그를 하나도 선택하지 않았을 때 나타나는 완전 차단 경고는 **선택한 태그만 제외하기** 모드에는 표시되지 않습니다. 제외 태그가 없는 상태는 알림을 전부 차단하는 상태가 아니라 전부 수신하는 상태이기 때문입니다.

선택한 태그만 제외하기

경고 알림을 대부분 받되 특정 분류만 걸러 내고 싶을 때 사용합니다. 받을 태그를 하나씩 고르는 **선택한 태그만 받기**와 방향이 반대입니다.

1. **사용자별 이벤트 수신 설정** 섹션에서 설정할 멤버 또는 3rd 파티 채널 행의 **이벤트 수신 태그** 열을 클릭하세요.
2. **선택한 태그만 제외하기** 라디오 버튼을 선택하세요.
3. 아래 태그 선택 영역에서 알림을 받지 않을 태그를 하나 이상 선택하세요.
    - 이벤트에 붙은 태그와 제외 태그가 하나라도 겹치면 해당 경고 알림은 전송되지 않습니다. 겹치는 태그가 없는 알림은 모두 수신합니다.
    - 제외 태그는 최소 한 개 이상 선택해야 합니다. 제외 태그가 하나도 없는 상태는 **모든 알림 받기**와 결과가 같아지므로, 마지막 태그는 제거할 수 없고 제외 태그가 비어 있는 동안 **최소한 하나의 태그를 선택해 주세요.** 안내가 표시됩니다.
    - 제외 태그 목록은 **선택한 태그만 받기** 모드의 수신 태그 목록과 별도로 저장됩니다. 두 모드를 번갈아 선택해도 각 모드에서 고른 태그가 그대로 남습니다.
    - **미분류** 가상 태그도 제외 대상으로 선택할 수 있으며, 태그 한 개로 계산됩니다.

알림 수신 언어 선택

사용자의 알림 수신 언어를 선택할 수 있습니다.

1. **경고 알림** > **이벤트 수신 설정**으로 이동하세요.
2. **사용자별 이벤트 수신 설정** 섹션의 사용자 목록에서 언어를 설정할 사용자의 가장 왼쪽에 ▶ 버튼을 클릭하세요.
3. **알림 수신 언어**를 선택하세요.

요일 및 시간별 알림 설정

사용자의 요일별, 시간별 알림 수신 여부를 선택할 수 있습니다.

1. **경고 알림** > **이벤트 수신 설정**으로 이동하세요.
2. **사용자별 이벤트 수신 설정** 섹션의 사용자 목록에서 요일 및 시간별 알림을 설정할 사용자의 가장 왼쪽에 ▶ 버튼을 클릭하세요.
3. 경고 알림 수신을 원하는 요일을 선택하거나 시간을 입력하세요.
    - 알림 수신 수단별로 설정할 수 있습니다.

3rd 파티 플러그인

외부 애플리케이션을 통해 경고 알림을 받을 수 있습니다.

- 지원 애플리케이션: Slack, Telegram, Teams, Teams Workflows, Jandi, AlertNow, Webhook, Webhook JSON, Opsgenie, PagerDuty, LINE, ilert

> ⓘ 와탭랩스의 지원 범위에 포함하지 않는 사내 메신저는 표준 Webhook, Webhook json을 통해 연동할 수 있습니다.

표 | 3rd 파티 플러그인 목록 구성

항목	설명
플러그인 이름	플러그인의 이름

항목	설명
인증 키	인증 키
인증 값	인증값
수신 레벨	경고, 전체, 위험으로 구분
반복 알림(에스컬레이션)	알림 발생 시간으로부터 알림 발생 상황이 해소되지 않았을 경우, 최초 알림 발생 시각으로부터의 알림 재발행 주기 - 위험(Critical) 단계의 알림에만 적용되는 기능
Smart 메시지(AI)	Telegram 3rd 파티 플러그인 추가 시 AI 알림 기능 제공 - 지원 언어: 한국어, 영어, 일본어
이벤트 수신 태그	프로젝트의 멤버 중 특정 멤버 또는 팀을 대상으로 알림 수신 여부
삭제	3rd 파티 플러그인 삭제

3rd 파티 플러그인 추가

1. **경고 알림** > **이벤트 수신 설정**으로 이동하세요.
2. **3rd 파티 플러그인** 섹션의 [추가하기] 버튼을 클릭하세요.
3. **서드 파티 플러그인 추가** 창에서 원하는 서비스 탭으로 선택하세요.
4. 선택한 서비스의 안내(3rd 파티 플러그인 알림 추가 방법)에 따라 설정을 진행하세요.
5. [수신 테스트] 버튼을 클릭해 설정이 잘 되었는지 확인하세요.
6. 모든 과정을 완료하면 [추가하기] 또는 [등록] 버튼을 클릭하세요.

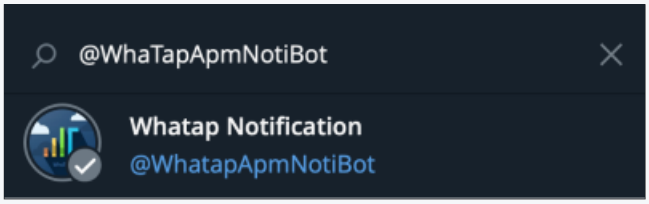
3rd 파티 플러그인 알림 추가 방법

Slack

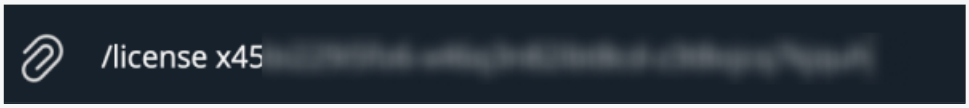
> **서드 파티 플러그인 추가**의 **Slack** 탭 화면의 [**슬랙에 추가하기**] 버튼을 클릭하면 Slack의 팀 및 채널에 **WhaTapNotiBot**을 등록할 수 있습니다.

Telegram

1. **Telegram**에서 **@WhaTapApmNotiBot**을 검색하세요.



2. Bot을 선택하고 [시작] 버튼을 클릭하세요.
3. 채팅창에 `/license` 명령어와 알림을 받을 프로젝트의 액세스 키를 채팅창에 입력하세요.

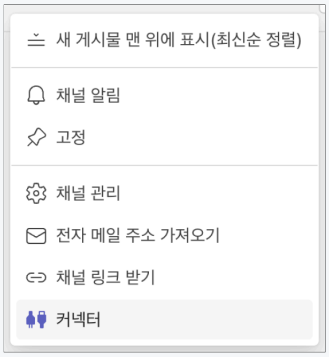


사용할 수 있는 명령어 목록

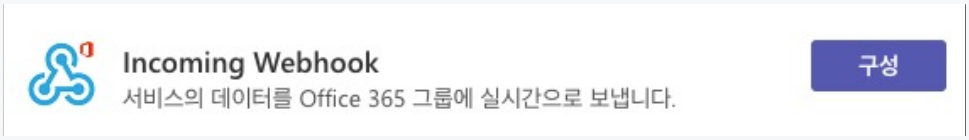
- `/help` : 명령어 목록 보기
- `/license` : 알림 받을 프로젝트의 액세스 키 입력
- `/remove` : 알림을 받지 않을 프로젝트 코드 입력
- `/list` : 현재 알림을 받고 있는 프로젝트 코드 보기

Teams

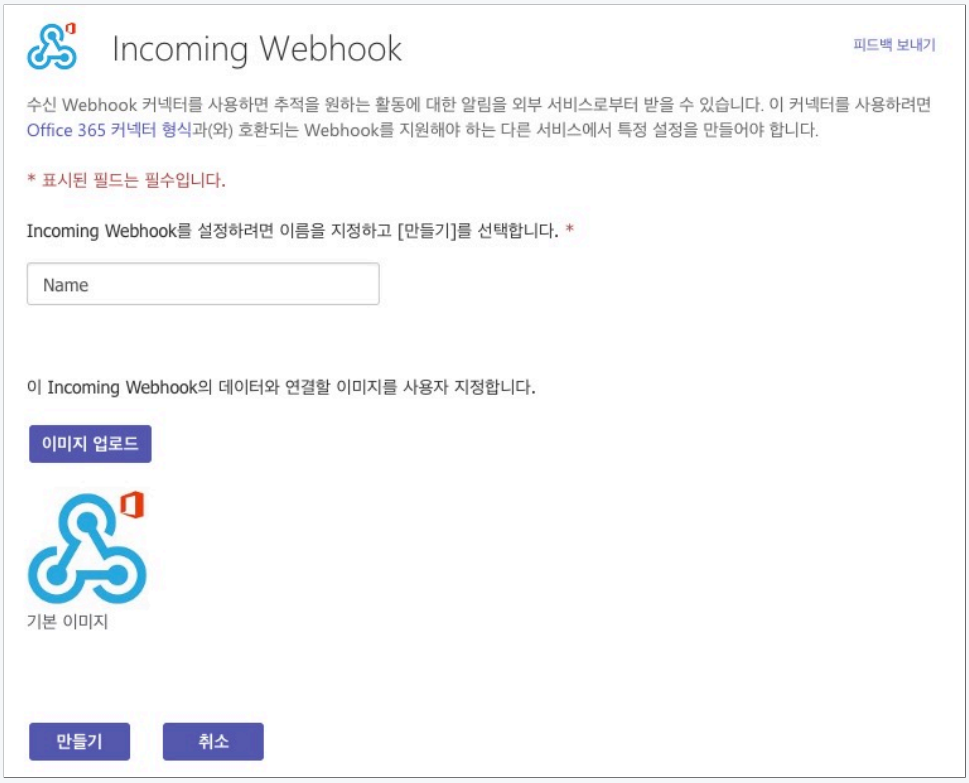
1. 알림을 받을 채널의 **커넥터**로 이동하세요.



2. **Webhook**을 추가하고 [구성] 버튼을 클릭하세요.



3. Webhook 이름을 입력하고 [만들기] 버튼을 클릭하세요.



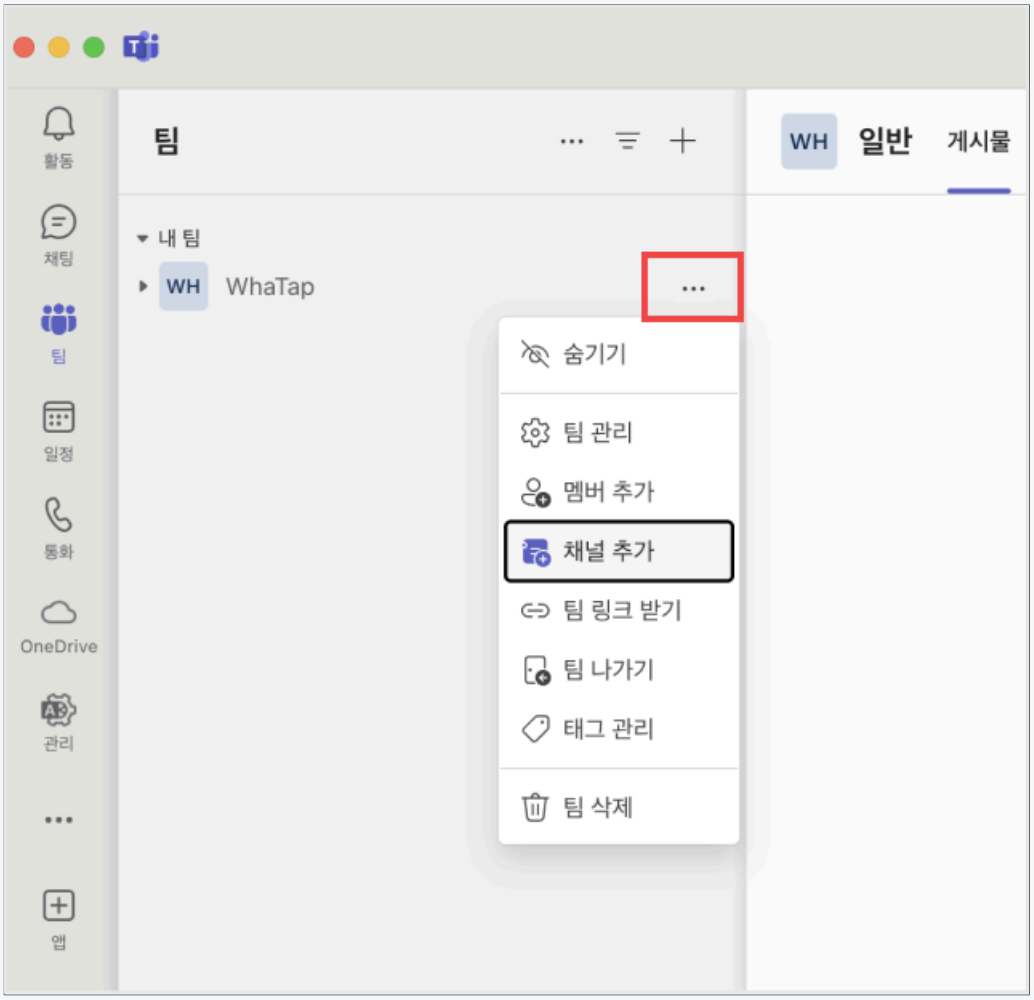
4. 생성된 Webhhok URL을 복사하세요.

5. 와탭 모니터링 서비스의 **서드 파티 플러그인 추가** 창으로 돌아와 **Teams** 탭 하단의 **Channel Name**과 **Webhook URL** 항목을 입력하세요.
6. [**등록**]버튼을 클릭하세요.

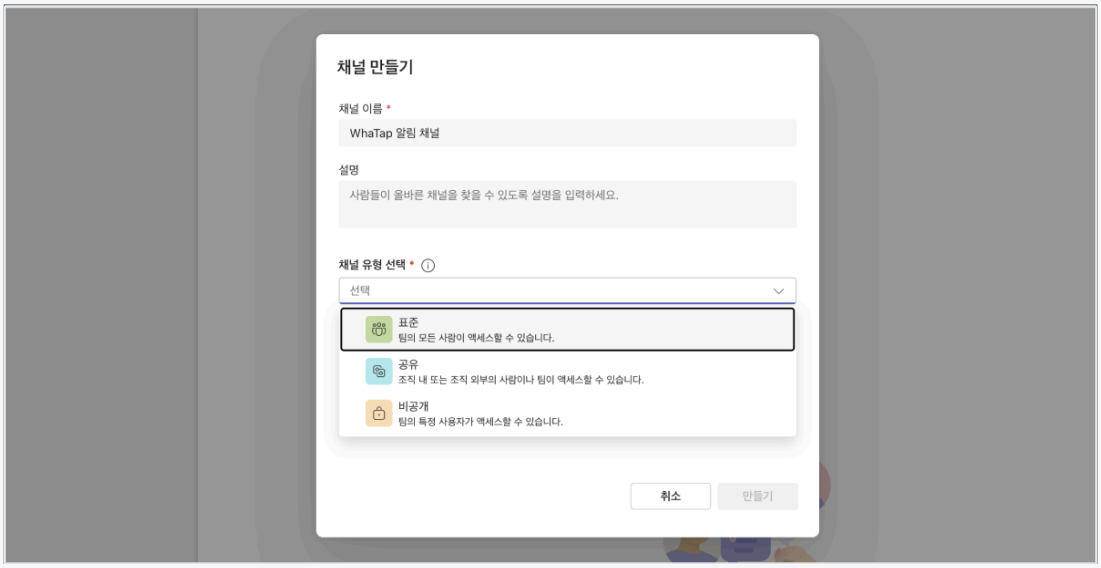
Teams Workflows

워크플로를 사용하려면 **Microsoft Teams**에 액세스할 수 있는 계정과 비즈니스 플랜 구독이 필요합니다.

1. **팀** 메뉴의 더보기(...) > **채널 추가**를 클릭하세요.

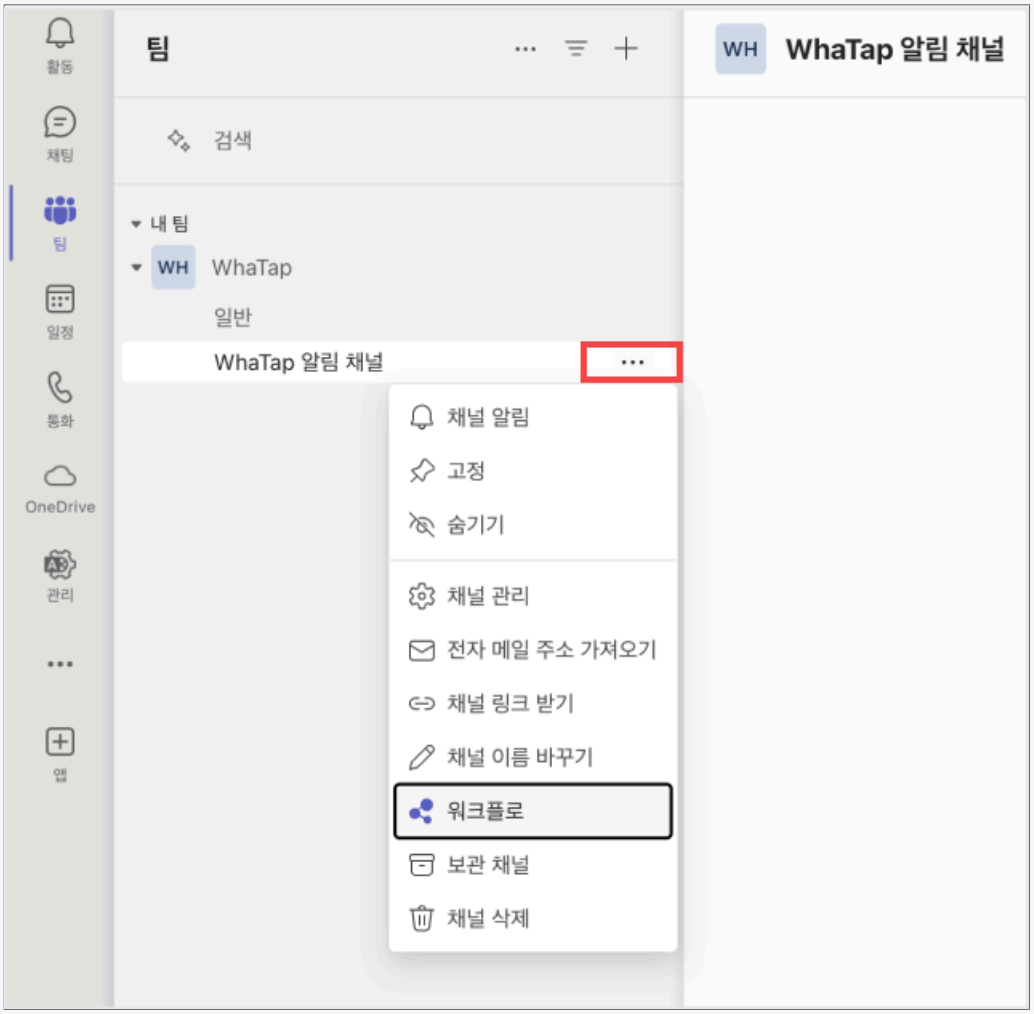


2. **채널 만들기** 창에서 **채널 이름**과 **설명**을 입력하고, **채널 유형 선택** 항목은 **표준**으로 선택한 후, [만들기] 버튼을 클릭하세요.



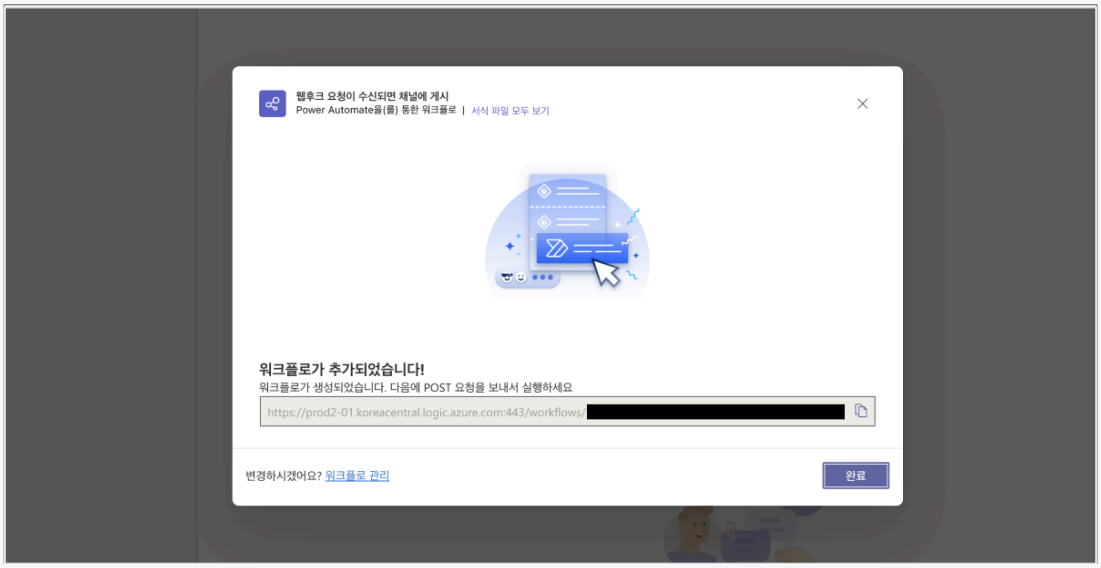
> 채널 유형을 **공유**로 선택하면 워크플로를 사용할 수 없으며, **비공개**를 선택하면 워크플로를 추가할 수 있지만 알림을 수신할 수 없습니다.

3. 생성한 채널의 더보기(...) > **워크플로**를 클릭하세요.



4. **워크플로** 대화상자의 **워크플로**의 목록에서 **웹후크 요청이 수신되면 채널에 게시**를 선택 후, [추가 워크플로] 버튼을 클릭하세요.

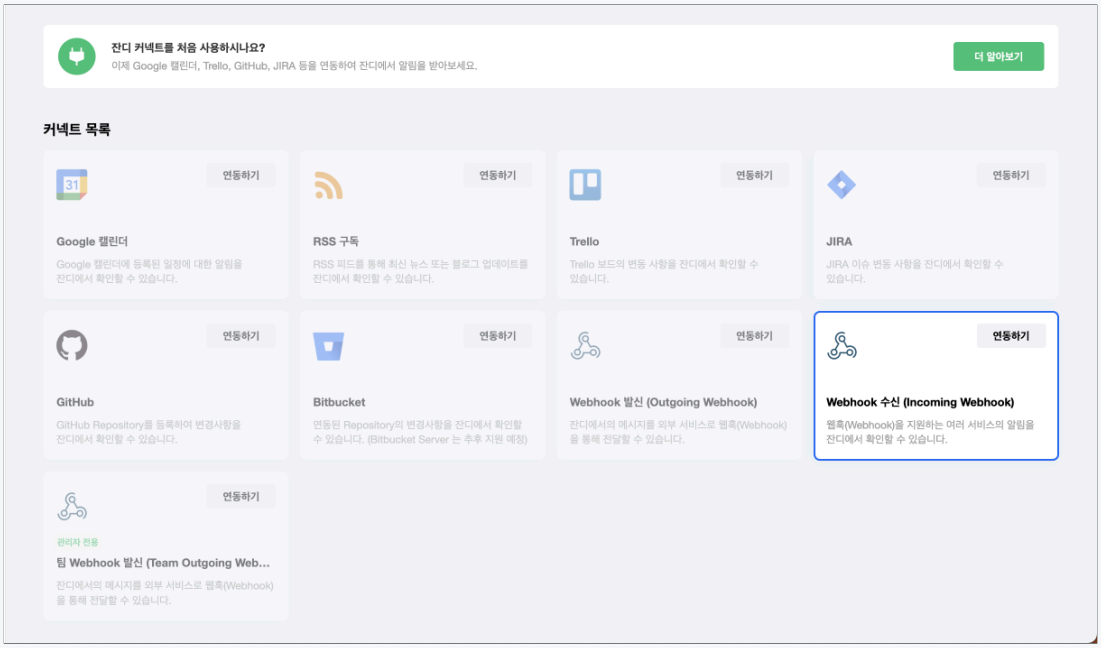
5. **이름**과 **연결** 상태를 확인한 후, [다음] 버튼을 클릭하세요.
6. **세부 정보**를 확인한 후, [워크플로 추가] 버튼을 클릭하세요.
7. 워크플로가 추가되었습니다. 생성된 주소를 복사하세요.



8. 와탭 모니터링 서비스의 **서드 파티 플러그인 추가** 창으로 돌아와 **Teams Workflows** 탭 안내 하단의 **Channel Name**과 **Webhook URL** 항목을 입력하세요.
9. [**등록**] 버튼을 선택하세요.

잔디(Jandi)

1. 알림을 받을 토픽 또는 채팅에서 잔디 커넥트 설정으로 이동하세요.
2. **커넥트 목록**에서 **Webhook 수신 (Incoming Webhook)** 항목에서 [연동하기] 버튼을 클릭하세요.

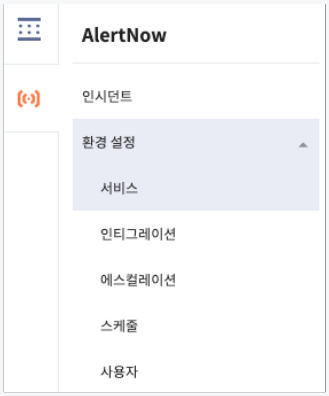


3. [Webhook 수신 추가] 버튼을 클릭하세요.
4. **Webhook URL 등록** 항목에 Webhook URL을 복사하세요.

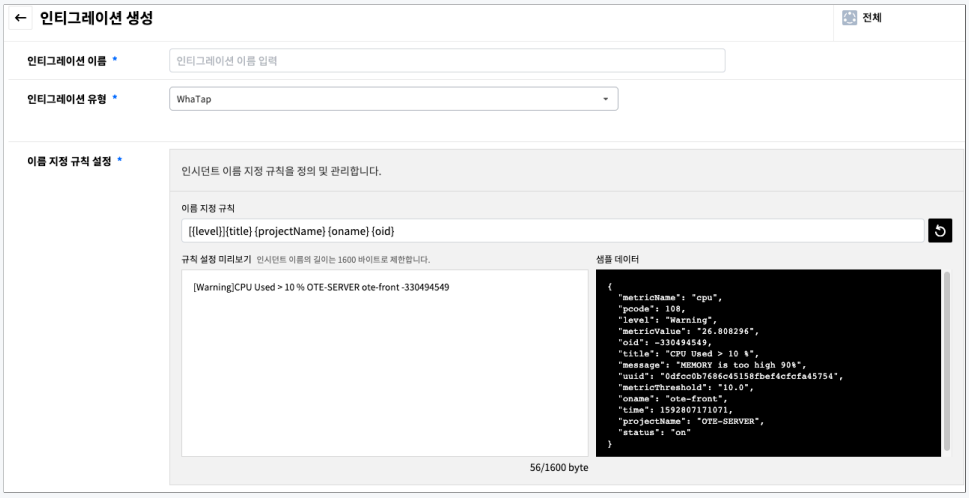
5. 와탭 모니터링 서비스의 **서드 파티 플러그인 추가** 창으로 돌아와 **Jandi** 탭 안내 하단의 **Channel Name**과 **Webhook URL** 항목을 입력하세요.
6. [**등록**] 버튼을 클릭하세요.

AlertNow

1. **AlerNow** 서비스를 접속한 후, **환경 설정** > **Integration** 메뉴로 이동하세요.



2. **인티그레이션** 화면에서 [인티그레이션 생성] 버튼을 클릭하세요.
3. Integration 유형 중 **WhaTap**을 선택하세요.
4. **인티그레이션 생성** 화면에서 이름 및 이름 지정 규칙을 설정하세요.



5. [확인] 버튼을 클릭하면 Integration이 생성됩니다.
6. 생성된 인티그레이션 목록을 확인하고 생성된 Integration 항목을 선택하세요.

7. 상세 화면에서 URL 오른쪽의 복사 아이콘을 클릭해 URL을 복사하세요.

8. 와탭 모니터링 서비스의 **서드 파티 플러그인 추가** 창으로 돌아와 **AlertNow** 탭 안내 하단의 **Channel Name**과 **Webhook URL** 항목을 입력하세요.
9. [**등록**] 버튼을 클릭하세요.

Webhook

1. 알림을 받을 Webhook을 등록하세요. 등록한 URL에 POST Method로 알림의 `title` , `text` 가 전송됩니다.

```
{
    "title":"Alert Title",
    "text": "Sample warning message"
}
```

1. **Webhook Name**과 **Webhook URL** 항목을 입력하세요.
2. [**등록**] 버튼을 클릭하세요.

Webhook JSON

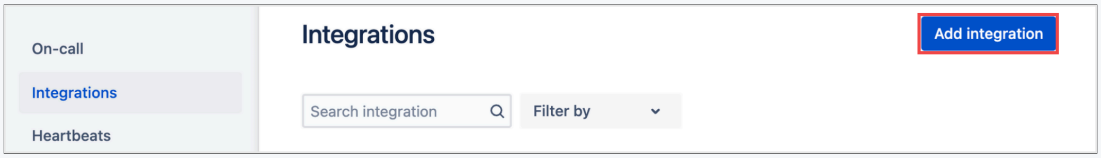
1. 알림을 받을 Webhook을 등록하세요. 등록한 URL에 POST Method로 알림 데이터가 JSON 형태로 전송됩니다. JSON 내에 포함 시킬 수 있는 대체 변수에 대해선 **사용 가능 변수** 부분을 참고하세요.
   - `${something}` 은 와탭에서 제공하는 대체될 값입니다. 지원하는 전체 목록과 의미는 사용 가능 변수를 참고하세요.
   - 이벤트 상태(`status`)는 이벤트 발생(`on`), 이벤트 해소(`off`)로 구분할 수 있습니다.
   - `level` 은 `None` , `Info` , `Warning` , `Critical` 로 구분합니다.

```
{
  "pcode": "${pcode}",
  "projectName": "${projectName}",
  "time": "${time}",
  "oid": "${oid}",
  "oname": "${oname}",
  "title": "${title}",
  "message": "${message}",
  "level": "${level}",
  "status": "${status}",
  "metricName": "${metricName}",
  "metricValue": "${metricValue}",
  "metricThreshold": "${metricThreshold}",
  "uuid": "${uuid}",
  "okind": "${okind}",
  "onode": "${onode}"
}
```

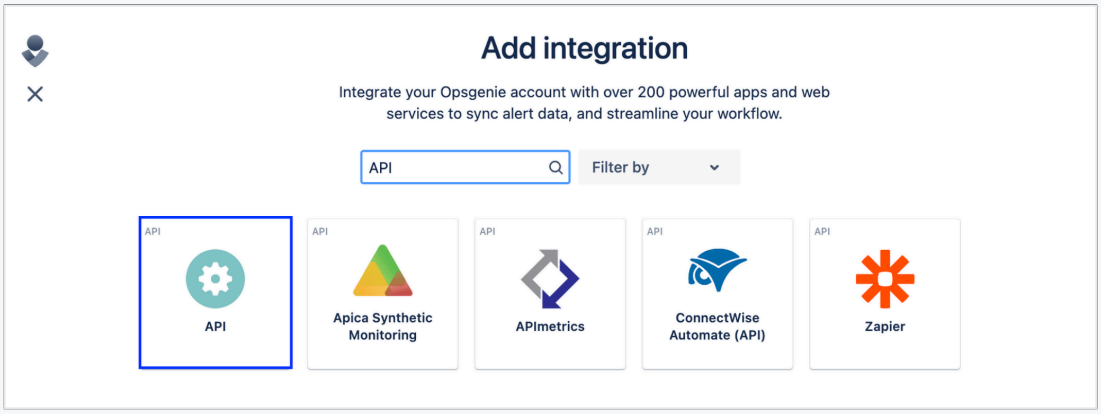
2. **Webhook Name**과 **Webhook URL** 항목을 입력하세요.
3. Header 값을 추가하려면 **사용자 Header 값 추가**의 **+ 추가**를 클릭해 key, value을 추가하세요.
4. [**등록**] 버튼을 클릭하세요.

Opsgenie

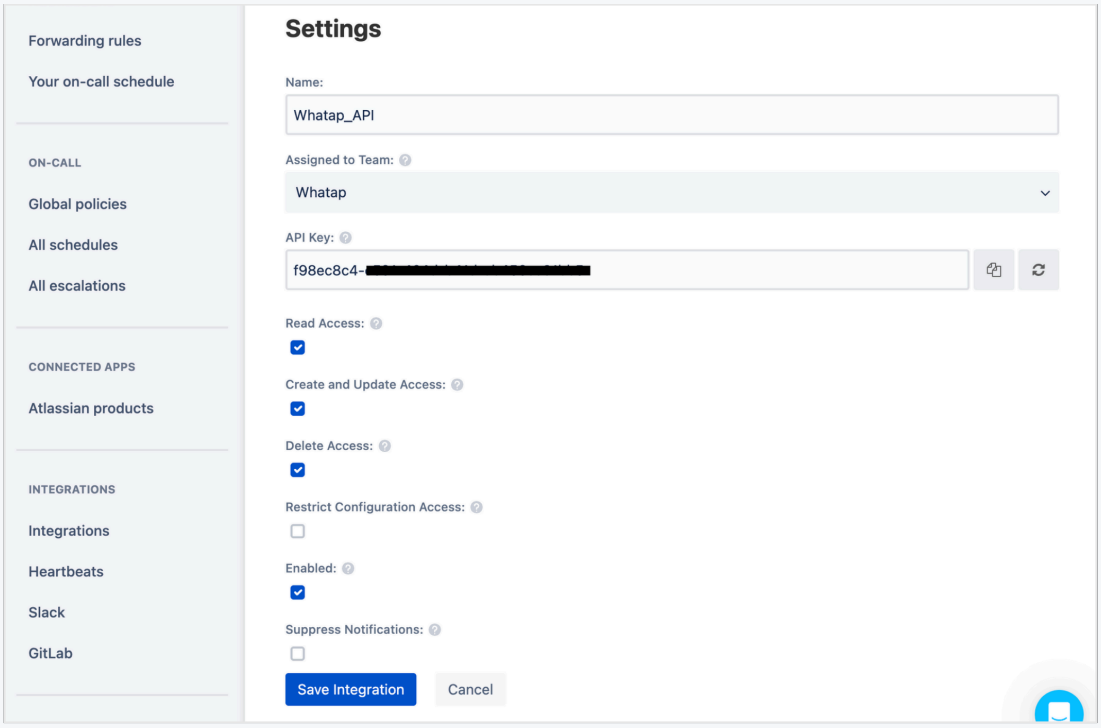
1. **Teams** 메뉴에서 알림을 수신할 팀을 선택하세요.
2. **Integration** 메뉴로 이동해 [Add Integration] 버튼을 클릭하세요.



3. Integration 목록에서 'API' 검색 후 선택하세요.



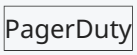
4. API 명과 Team을 할당하고 각 권한을 설정한 후, [Save Integration] 버튼을 클릭하세요.



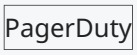
5. 와탭 모니터링 서비스의 **서드 파티 플러그인 추가** 창으로 돌아와, **Opsgenie** 탭 안내 하단의 **Name**과 **API Key** 항목을 입력하세요.
6. [**등록**] 버튼을 선택하세요.

PagerDuty

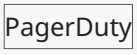
1. PagerDuty 계정에 등록하고 로그인하세요.
2. PagerDuty에서 **Services** > **Service Directory** 메뉴에서 [+ New Service] 버튼을 클릭하세요.

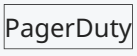
3. Service 정보(**Name**)를 입력한 후 [Next] 버튼을 클릭하세요.

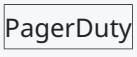
4. Escalation Policy를 설정 후 [Next] 버튼을 클릭하세요.

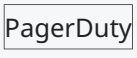
5. **Alert Grouping**과 **Transient Alerts** 항목을 설정한 후 [Next] 버튼을 클릭하세요.

6. **Integrations**에서 **Event API V2**를 선택한 후 [Create Service] 버튼을 클릭하세요.

7. 생성한 Service 화면에서 **Integration Name**과 **Integration Key** 값을 복사하세요.

8. 와탭 모니터링 서비스의 **서드 파티 플러그인 추가** 화면으로 돌아와 **Integration Name**과 **Integration Key** 항목을 입력하세요.
9. [등록] 버튼을 클릭하세요.

> 메시지를 수신하지 못했거나 플러그인이 등록되지 않았다면 **Integration Name** 또는 **Integration Key**값이 올바른지 확인하세요.

LINE

1. LINE 앱을 설치합니다.
2. 아래 QR 코드를 스캔하여 **WhaTapLabs** 공식 채널을 친구로 추가합니다.

3. 다음 명령어를 사용하여 프로젝트를 라인과 연동하고 알림을 설정하세요.

명령어

```
/help # 사용 가능한 명령어 안내
/list # 연동된 프로젝트 목록 조회
/license [액세스키] # 새 프로젝트 연동 추가
/remove [프로젝트코드] # 프로젝트 연동 해제
/language [ko|ja|en] # 알림 언어 설정 (한국어/일본어/영어)
```

- 프로젝트 액세스 키는 **관리 > 프로젝트 관리**에서 확인하실 수 있습니다.
- 라인 내에서 `/language` 명령어를 사용하는 것은 라인 내 채팅방에만 적용됩니다. 다른 3rd 플러그인이나 프로젝트의 알림 수신 언어가 변경되는 것은 아닙니다.

사용 예시

```
/license XXXXXXXXXX # 프로젝트 연동
/language ko # 한국어로 언어 설정
```

```
/list # 연동된 프로젝트 확인
```

- 프로젝트 액세스 키: WhaTap 프로젝트 내 **관리 > 프로젝트 관리**에서 확인할 수 있음
- 언어 설정: `/language` 명령어는 LINE 채팅방 내에서만 적용되며, 다른 알림 채널의 언어는 변경되지 않음

ilert

1. ilert 콘솔에서 **Alert sources** 화면의 오른쪽 위의 [+ Create alert source]를 클릭하세요.
2. **Integration type 단계**의 하단에 있는 **Filter integrations 검색창**에 `WhaTap` 을 입력해 검색합니다.
3. 검색 결과에서 **WhaTap 타일**을 선택한 후 [Next] 버튼을 클릭합니다.
4. 알림 소스 이름을 입력하고 팀을 지정한 후 [Next] 버튼을 클릭합니다.
   - **에스컬레이션 정책은** 새 정책을 생성하거나 기존 정책을 선택할 수 있습니다.
5. 알림 그룹화 방식을 선택한 후 [Continue setup] 버튼을 클릭합니다.
   - 우선 선택한 **Do not group alerts는** 이후 변경할 수 있습니다.
6. 다음 화면에서 **알림 템플릿, 우선순위 등 추가 설정**을 할 수 있습니다. 우선 [Finish setup] 버튼을 클릭해 설정을 완료합니다.
7. 설정이 완료되면 완료 화면에서 **API 키** 또는 **Webhook URL**이 생성됩니다.
8. **Webhook JSON** 탭에서 생성된 **Webhook URL**을 복사해 아래의 **Webhook URL 입력란**에 붙여넣습니다.
9. [등록] 버튼을 클릭하세요.

대량 알림 발생 방지

알림이 대량으로 발생하면 지정한 시간 동안 알림이 일시적으로 중지됩니다. 예를 들어, 5분 동안 20회의 이벤트가 발생하면 5분

동안 경고 알림을 중지합니다. 설정한 **정지 시간** 시간이 지나면 대량 알림 발생 방지 기능은 해제됩니다.

1. **경고 알림** > **이벤트 수신 설정**의 **대량 알림 발생 방지** 섹션으로 이동하세요.
2. 토글 버튼을 클릭해 **활성화** 또는 **비활성화** 할 수 있습니다.
   - **활성화**: 대량 알림 일시적 중지
   - **비활성화**: 대량 알림 재개
3. **탐지 시간, 탐지 횟수, 정지 시간**을 설정하세요.
   - **탐지 시간** 동안 **탐지 횟수** 이상의 이벤트가 발생하면 **정지 시간** 동안 경고 알림을 중지합니다.

> ⓘ 대량 알림으로 차단되는 경우 차단 안내 박스가 표시됩니다. 대량 알림 차단 기능을 해제하려면 박스의 [대량 알림 발생 중지] 버튼을 클릭하세요.

이벤트 수신 포맷

공용 알림 템플릿

공통 템플릿은 대부분의 경우 적용 가능합니다. 다양한 환경에서 동일한 포멧으로 알림을 받을 수 있습니다.

- Product Type: 애플리케이션, 데이터베이스, 쿠버네티스
- Event Type: 애플리케이션 알림, 데이터베이스 알림, 메트릭스 알림
- Event Channel: sms, mobile, 3rd party plugin, plugin

이벤트 제목

Event title format

```
[Level][Platform][ProjectName][ApplicationName][EventTitle]
```

Event title example

```
[Info][JAVA][애플리케이션 프로젝트][TC-0-1-8081][CRITICAL_HIGH_MEMORY]
```

ⓘ ApplicationName을 설정하지 않은 경우 이벤트 제목에서 생략합니다.

- Platform은 다음 중 한 가지로 표시합니다.
  - JAVA
  - NODEJS
  - PYTHON
  - PHP
  - DOTNET
  - GO
  - POSTGRESQL
  - ORACLE

- MYSQL
- MSSQL
- BSM_JAVA
- CLOUDWATCH
- TIBERO
- KUBERNETES
- KUBE_NS
- URLCHECK
- URLCHECK_ADMIN
- CUBRID
- ALTIBASE
- CLUSTER
- REDIS
- MONGODB
- VR
- RUM

이벤트 메시지

이벤트 메시지에 포함할 수 있는 정보입니다. `optional` 이 `false` 인 경우 항상 메시지에 포함합니다. `optional` 이 `true` 인 경우 해당 데이터를 확인할 수 있으면 표시합니다.

Event message example

```
Project Name : 애플리케이션 프로젝트
Project Code : 3
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
```

표 | 이벤트 메시지 구성 요소

En	Ko	지원되는 알림 타입	설명
Project Name	프로젝트 이름	전체	-
Project Code	프로젝트 Code	전체	-
Agent Name	에이전트 이름	전체(optional)	oname
Message	이벤트 메시지	전체	-
Alert Type	이벤트 종류	전체	아래의 AlertType 종류 표 참고
Event Time	이벤트 발생 시간	전체	2022-04-13 10:40:49 +0900에서 +0900는 GMT를 의미합니다.
Event Off Time	이벤트 해제 시간	전체(optional)	2022-04-13 10:40:49 +0900에서 +0900는 GMT를 의미합니다.
Metric Name	메트릭스 이름	전체(optional)	이벤트 조건 판단에서 사용하는 메트릭스의 이름
Metric Value	메트릭스 값	전체(optional)	메트릭스 값이 메트릭스 임계치를 넘으면 이벤트 발생 조건이 만족한 경우입니다.
Metric Threshold	메트릭스 임계치	전체(optional)	메트릭스 값이 메트릭스 임계치를 넘으면 이벤트 발생 조건이 만족한 경우입니다.
Stateful	해소된 이벤트 알림	전체(optional)	해소된 이벤트 알림 기능 사용 중이면 true, 아니면 false
Event Rule	이벤트 발생 조건	메트릭스 알림	-

En	Ko	지원되는 알림 타입	설명
Target Filter	이벤트 대상 선택	메트릭스 알림	특정 대상에서 수집된 메트릭스에 대해서만 이벤트 조건을 확인합니다.
Repeat Count	이벤트 반복 횟수	메트릭스 알림	이벤트 조건이 이벤트 반복 시간동안 이벤트 반복 횟수만큼 만족해야 이벤트가 발생됩니다.
Repeat Duration	이벤트 반복 시간	메트릭스 알림	이벤트 조건이 이벤트 반복 시간동안 이벤트 반복 횟수만큼 만족해야 이벤트가 발생됩니다.
Receiver	수신자	메트릭스 알림	-
Query	MXQL 쿼리	복합 메트릭스 알림	-
Rule	이벤트 발생 조건	복합 메트릭스 알림	-
Query Period	쿼리 기간	복합 메트릭스 알림	-
Query Interval	쿼리 간격	복합 메트릭스 알림	-
Silent Time	무음 시간	복합 메트릭스 알림	-
Query	URL	Exception 알림	Exception을 발생시킨 요청의 URL
TXID	트랜잭션 ID	Exception 알림	-
Class	에러 클래스 이름	Exception 알림	-

En	Ko	지원되는 알림 타입	설명
Log Message	로그 메시지	서버 - 파일 로그 알림	-
Log File	로그 파일 경로	서버 - 파일 로그 알림	-
IP	IP	서버 알림 전체	-
CPU	CPU	서버 알림 전체	이벤트 발생 당시의 Snapshot
CPU_load1	CPU_load1	서버 알림 전체	이벤트 발생 당시의 Snapshot
CPU_loadPerCore	CPU_loadPerCore	서버 알림 전체	이벤트 발생 당시의 Snapshot
Memory	Memory	서버 알림 전체	이벤트 발생 당시의 Snapshot
Swap	Swap	서버 알림 전체	이벤트 발생 당시의 Snapshot
Disk Name | Used Percent | Free Size | IO Percent	디스크 퍼포먼스	서버 알림 전체	이벤트 발생 당시의 Snapshot
Name | Bps | Pps	트래픽 퍼포먼스	서버 알림 전체	이벤트 발생 당시의 Snapshot
Message | Time | Name	처리내역 메시지	서버 알림 전체	-

ⓘ 해당 이벤트에서 제공할 수 있는 최대한 많은 정보를 보여줍니다.

- AlertType은 다음 중 한 가지로 표시합니다.

AlertType	설명
APPLICATION_CPU	애플리케이션 CPU 알림
APPLICATION_MEMORY	애플리케이션 MEMORY 알림

AlertType	설명
APPLICATION_DISK	애플리케이션 DISK 알림
APPLICATION_ACTIVE_TRANSACTION	애플리케이션 액티브 트랜잭션 알림
APPLICATION_ERROR_TRANSACTION	애플리케이션 에러 트랜잭션 알림
APPLICATION_SLOW_TRANSACTION	애플리케이션 트랜잭션 응답시간 알림
METRICS	메트릭스 알림
COMPOSITE_METRICS	복합 메트릭스 알림
ANOMALY	이상치 탐지 알림
LOG_REALTIME	로그 실시간 알림
COMPOSITE_LOG	복합 로그 알림
SERVER_REBOOT	서버 - 재시작 알림
SERVER_NO_DATA	서버 - 미수신 알림
SERVER_PORT	서버 - 포트 알림
SERVER_NETWORK_IOPS	서버 - 네트워크 IOPS 알림
SERVER_NETWORK_BPS	서버 - 네트워크 BPS 알림
SERVER_DISK_IO	서버 - 디스크 I/O 알림
SERVER_DISK_QUOTA	서버 - 디스크 사용량 알림
SERVER_DISK_INODE	서버 - inode 알림

AlertType	설명
SERVER_CPU	서버 - CPU 알림
SERVER_MEMORY	서버 - 메모리 알림
SERVER_CPU_STEAL	서버 - steal 알림
SERVER_MEMORY_SWAP	서버 - 스왑 알림
SERVER_LOG_FILE	서버 - 로그 파일 알림
SERVER_WINDOW_EVENT	서버 - 윈도우 이벤트 알림
SERVER_OFF	서버 - 알림 OFF 알림
SERVER_ACKNOWLEDGE	서버 - 처리내역 알림
SERVER_PROCESS_COUNT	서버 - 프로세스 수 알림
SERVER_PROCESS_CPU	서버 - 프로세스 CPU 알림
SERVER_PROCESS_MEMORY	서버 - 프로세스 메모리 알림
SERVER_PROCESS_OFF	서버 - 프로세스 알림 OFF 알림
AGENT_ACTIVE	에이전트 활성화 알림
AGENT_INACTIVE	에이전트 비활성화 알림
AGENT_REACTIVATED	에이전트 재활성화 알림
URL	URL 알림
TOO_MANY_EVENT	너무 많은 이벤트 발생 알림

AlertType	설명
CLOUD_WATCH	Cloud Watch 알림
EXCEPTION	Exception 알림

이벤트 기록 및 모니터링

홈 > 프로젝트 선택 > **경고 알림** > **이벤트 기록**

이벤트 기록은 프로젝트에서 발생한 여러 이벤트를 조회하고 상세 정보를 분석할 수 있습니다. Elasticsearch DSL 쿼리를 지원하며, 원하는 조건에 맞는 이벤트를 정확하고 빠르게 찾을 수 있습니다.

이벤트 기록 조회하기

검색 결과에서 **이벤트 제목을 클릭**하면 해당 이벤트가 발생한 시간대의 상세 데이터를 분석할 수 있는 화면으로 이동합니다. 이를 통해 이벤트 발생 전후 상황을 종합적으로 파악할 수 있습니다.

1. **경고 알림** > **이벤트 기록**으로 이동하세요.
2. **이벤트 기록** 화면의 **시간 선택**에서 이벤트 기록을 조회할 기간를 설정하세요.
3. 필터 입력 창에 쿼리을 입력하고 [돋보기] 버튼을 클릭하세요.
    - 쿼리를 입력하지 않으면 전체 이벤트가 조회됩니다.

컬럼 설정

원하는 컬럼을 표시하거나 숨길 수 있습니다.

1. **이벤트 기록** 화면의 왼쪽 위 [컬럼 설정] 버튼을 클릭하세요.
2. **컬럼 설정** 창에서 표시하거나 숨길 컬럼을 선택하세요.
    - **표시 항목**에서 컬럼의 순서를 변경할 수 있습니다.
    - **컬럼명 검색**창에 원하는 컬럼 항목을 검색할 수 있습니다.

컬럼 항목	설명
이벤트 이름	이벤트의 제목
메시지	이벤트의 메시지 또는 스냅샷 정보

컬럼 항목	설명
이벤트 상태	이벤트의 현재 상태
에이전트 명	이벤트가 발생한 대상 에이전트의 이름(Oname) - 특정 에이전트에 대해 발생한 이벤트가 아닌 경우 빈칸으로 표시됨
에이전트 그룹	이벤트가 발생한 대상 에이전트의 종류(OkindName) - 특정 에이전트에 대해 발생한 이벤트가 아닌 경우 빈칸으로 표시됨
에이전트 노드	이벤트가 발생한 대상 에이전트의 서버(OnodeName) - 특정 에이전트에 대해 발생한 이벤트가 아닌 경우 빈칸으로 표시됨
이벤트 발생 시각	이벤트가 발생한 시간
이벤트 해소 시각	이벤트가 해소된 시간 상태가 없는 이벤트의 경우 `-`로 표시됨

3. 설정을 완료한 후, [확인] 버튼을 클릭해 컬럼 설정을 반영합니다.

CSV 다운로드

조회 결과를 CSV 파일 형식으로 다운로드할 수 있습니다.

1. **경고 알림** > **이벤트 기록**에서 [CSV] 버튼을 클릭하세요.
2. 현재 필터 조건에 맞는 이벤트 목록이 CSV 파일로 저장됩니다.

진행 중인 이벤트만 보기

현재 진행 중인 상태의 이벤트만 조회할 수 있습니다.

1. **경고 알림** > **이벤트 기록**에서 [진행 중인 이벤트만 보기] 버튼을 클릭하세요.
2. 필터 쿼리에 다음 조건이 자동으로 추가되어 검색됩니다.

```
Stateful: true and Status : "ON"
```

이벤트 검색

- **정확한 검색**: 원하는 조건을 정확히 지정하여 필요한 이벤트만 조회
- **복합 조건**: 여러 필드를 동시에 검색하여 복잡한 조건도 한 번에 처리
- **유연한 패턴**: 와일드카드, 부분 검색 등 다양한 검색 방식 지원
- **빠른 성능**: 인덱싱된 데이터를 활용한 고속 검색

검색 가능한 필드

필드와 값에 대해 대소문자를 구분하지 않습니다.

- **필드명 태그**: 필드가 지원되는 이벤트 타입
- **필드 타입**: 필드의 데이터의 타입
  - `#`: Number
  - `T`: String
  - : Boolean
  - : Date

이벤트 정보 필드

필드명	타입	설명	예시 값
Title	String	이벤트 제목	"Database Connection Error"
OffTitle	String	이벤트 해소 제목	"RECOVERED: Database Connection Error"
Message	String	이벤트 메시지	"Connection timeout occurred"
OffMessage	String	이벤트 해소 메시지	"RECOVERED: Connection timeout occurred"
Level	String	현재 이벤트 레벨	`Critical`, `Warning`, `Info`

필드명	타입	설명	예시 값
OriginLevel	String	원본 이벤트 레벨	`Critical`, `Warning`, `Info`
Status	String	이벤트 상태	`ON`, `OFF`, `CANCEL`, `ACKNOWLEDGE`, `MAINTENANCE`, `DISABLED`
ActCount	Number	처리내역 개수	2
MetricName	String	지표 이름	"memory"
MetricValue	String	지표 값	"85.5"
OffValue	String	해소 값	"72.8"
MetricThreshold	String	임계치	"80"
alertType	String	이벤트 종류	"BASIC", "METRICS", "TRANSACTION" 등
alertId	String	이벤트 규칙의 고유식별자	"zf3uojer0fv4v7"

이벤트 타입별 지원 필드

> ⓘ 특정 이벤트 타입에서만 지원되는 필드입니다. 해당 이벤트 타입이 아닌 경우 값이 없을 수 있습니다.

필드명	타입	이벤트 종류	설명	예시 값
eventRule	String	기본, 메트릭스	이벤트 발생 규칙	"memory ≥ 80"
field	String	실시간 로그	로그 검색 키	"content"

필드명	타입	이벤트 종류	설명	예시 값
keyword	String	실시간 로그	로그 검색 값	"Error"
logCategory	String	실시간 로그	로그 카테고리	"AppLog"
logContent	String	실시간 로그	로그 내용	"00:00000:00009:2025/04/06 23:03:52.55 server Error: 1601, Severity: 17, State: 3\n00:00000:00009:2025/04/06 23:03:52.55 server There are not enough 'user connections' available to start a new process."

대상 필드

필드명	타입	설명	예시 값
Oid	Number	에이전트 고유식별자	-98765432
Oname	String	에이전트 이름	"web-server-01"
Okind	Number	에이전트 종류 고유식별자	867318026
OkindName	String	에이전트 종류명	"web-server"
Onode	Number	에이전트 서버 고유식별자	334634079
OnodeName	String	에이전트 서버명	"production-node-1"

이벤트 고유값 필드

필드명	타입	설명	예시 값
Id	Number	이벤트의 고유식별자	5768121

필드명	타입	설명	예시 값
UUID	String	이벤트 고유식별자	"550e8400-e29b-41d4-a716"

상태/플래그 필드

필드명	타입	설명	예시 값
Stateful	Boolean	상태 기반 이벤트 여부	true, false
Disabled	Boolean	종료된 이벤트 여부	true, false
Escalation	Boolean	에스컬레이션 여부	true, false
SystemEvent	Boolean	시스템 이벤트 여부	true, false

검색 유형별 문법

> ⓘ 검색 기본 문법은 WhaTap log search query 문법 문서를 참고하시기 바랍니다.

1. 키워드 검색

Database Connection 문자열이 포함된 이벤트

```
title: "Database Connection"
```

2. 여러 값 검색

여러 옵션 중 하나라도 포함되는 이벤트를 조회합니다.

```
title: Database Connection
```

현재 이벤트 레벨이 Info이거나 Warning인 이벤트

```
level: info warning
```

3. 패턴 검색

와일드카드를 사용하여 패턴을 검색할 수 있습니다.

이름 패턴으로 특정 에이전트에서 발생된 이벤트 찾기

```
oname: web-*-prod
```

제목이 Connection으로 끝나는 이벤트 찾기

```
title: *Connection
```

4. 복합 조건 검색

여러 조건을 조합하여 정확히 검색할 수 있습니다.

Critical 레벨이면서 제목에 Database가 포함된 이벤트

```
level: critical and title: Database
```

특정 에이전트의 Warning 또는 Critical 이벤트

```
oname: web-server-01 and level: Warning Critical
```

5. OR 조건 검색

여러 조건 중 하나라도 만족하는 이벤트를 조회합니다.

제목 또는 메시지에 특정 키워드가 포함된 이벤트

```
title: Connection or message: Connection
```

6. 제외 조건

특정 조건을 제외해 검색할 수 있습니다.

Info 레벨을 제외한 모든 이벤트

```
not level: info
```

실시간 알림

실시간 알림 기능은 **이벤트 설정**을 통해 설정한 이벤트가 발생할 때 팝업 형태 표시됩니다. 이벤트 발생 즉시 알림을 받아 문제를 빠르게 인지하고 대응할 수 있습니다. 이미 발생했거나 놓친 알림 내역을 확인하여 실시간 대응이 어려웠던 상황을 파악하고 해결할 수 있습니다.

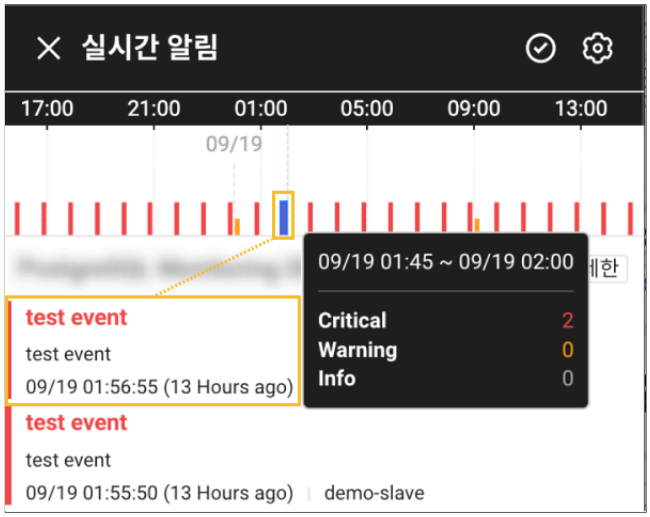
실시간 알림 확인하기

실시간 알림은 화면 오른쪽 위 팝업 형태로 표시됩니다. 알림이 발생한 시간, 제목, 메시지를 확인할 수 있습니다.

- 개별 알림을 닫으려면 메시지 오른쪽 상단의 ✕ 버튼을 클릭하세요.
- 화면에 표시된 모든 팝업 알림을 닫으려면 **전체 알림 닫기**을 클릭하세요.

알림 내역 확인하기

1. 프로젝트를 선택해 진입한 화면의 가장 위에 🔔 아이콘을 클릭하세요.
2. **실시간 알림** 창이 나타납니다.
    - 현재 프로젝트의 최근 24시간 동안의 알림을 최대 100개까지 확인할 수 있습니다.
    - 개별 이벤트 항목을 선택하면 관련 메뉴로 이동합니다.
    - **실시간 알림** 창의 시간별 차트에서 원하는 시간대를 선택하면 해당 이벤트 항목으로 이동할 수 있습니다.



- 전체 알림 내역을 확인하려면 **전체 알림 기록** 버튼을 선택하세요. **경고 알림** > **이벤트 기록** 메뉴로 이동합니다.

> **실시간 알림** 목록이 비어 있을 경우, [전체 알림 기록] 버튼이 표시됩니다. **수신 태그 미설정 알림 받기** 옵션을 해제하세요.

이벤트 일괄 해소하기

실시간 알림 목록에서 진행 중인 모든 이벤트를 일괄 해소합니다. 이벤트 해소 작업 완료까지 약 30초에서 1분 정도 소요됩니다.

- **이벤트 일괄 처리** 기능을 사용하면 현재 진행 중인 이벤트에 대한 해소 알림이 다량 발생할 수 있습니다.
- **이벤트 일괄 처리** 기능을 사용한 후에도 이벤트 규칙이 활성화되어 있고, 데이터가 임계치를 초과했다면 이벤트는 다시 발생할 수 있습니다.
- 이벤트 해소 알림은 **경고 알림** > **이벤트 설정** 메뉴에서 이벤트 수정을 통해 비활성화할 수 있습니다. **이벤트 상태가 해소되면 추가 알림** 또는 **해소된 알림** 옵션을 비활성화하세요.

사용 방법:

1. 프로젝트를 선택해 진입한 화면 상단의 아이콘을 클릭하세요.
2. **실시간 알림** 창이 나타나면 오른쪽 상단의 아이콘을 클릭하세요.

3. `실시간 알림 목록에서 진행중인 이벤트를 일괄 해소합니다.` 메시지 창이 나타나면 [해소] 버튼을 클릭하세요.

실시간 알림 설정하기

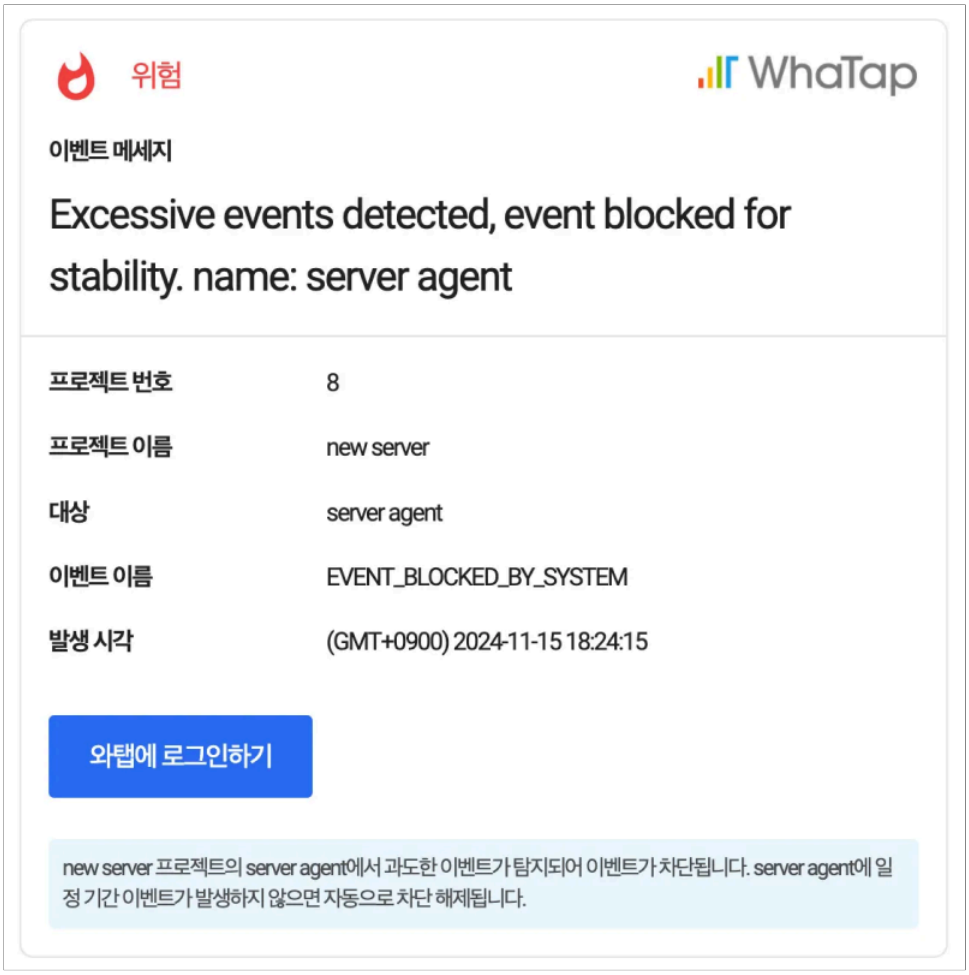
1. 프로젝트를 선택해 진입한 화면 상단의 🔔 아이콘을 클릭하세요.
2. **실시간 알림** 창이 나타나면 오른쪽 상단의 ⚙ 아이콘을 클릭하세요.
3. 실시간 알림 설정 창에서 각 항목을 설정하세요.
   - **팝업 알림**: **실시간 알림** 창이 닫힌 상태에서도 알림을 팝업 형태로 제공합니다.
   - **해소된 이벤트 보기**: **실시간 알림** 목록에서 해소된 이벤트를 포함해 확인할 수 있습니다.
   - **알림음 미리 듣기**: 브라우저에서 발생하는 알림음을 미리 들을 수 있습니다. Critical 또는 Warning 옆의 📢 버튼을 클릭하세요. 알림음이 들리지 않으면 브라우저 설정을 확인하시기 바랍니다.

시스템 이벤트

와탭 모니터링은 플랫폼 전반에서 자동으로 탐지 및 생성되는 공용 이벤트를 제공합니다. 이 문서는 시스템에 의해 발생하는 이벤트의 유형 및 발생 조건, 대응 방법에 대해 안내합니다.

EVENT_BLOCKED_BY_SYSTEM

프로젝트에서 짧은 시간에 과도한 이벤트가 발생하면, 와탭 시스템이 이를 탐지하여 이벤트를 차단하고 다음 메시지를 사용자에게 전달합니다. 이벤트가 차단되었다면 사용자의 이벤트 설정을 점검하세요.



차단 기준

이벤트를 차단하는 기준은 다음과 같습니다.

1. **에이전트 단위 차단**

   한 개의 에이전트에서 **1분** 내에 **10개**의 이벤트가 발생하면 해당 에이전트에서 발생하는 이벤트를 차단합니다.

2. **다중 에이전트 차단 시 전체 차단**

   1의 경우로 이벤트가 차단된 에이전트가 **1분** 내에 **3개** 이상이 되는 경우 프로젝트 전체에서 발생하는 이벤트를 차단합니다.

3. **프로젝트 단위 차단**

   한 개의 프로젝트에서 **1분** 내에 **100개**의 이벤트가 발생하면 프로젝트 전체에서 발생하는 이벤트를 차단합니다.

해제 방법

시스템에 의해 이벤트가 차단된 경우, 1분 동안 대상 에이전트 또는 프로젝트에서 이벤트가 발생하지 않아야 합니다.

- **경고 알림** > **이벤트 설정** 메뉴에서 설정한 이벤트를 모두 비활성화하세요. **이벤트 설정을 점검하여 과도한 이벤트가 발생하지 않도록 임계치를 조정해야 합니다**.
- 이벤트를 설정하기 전 **시뮬레이션** 기능을 활용해 이벤트 발생 횟수를 예측해볼 수 있습니다.

실험실

사용자에게 새로운 기능 또는 실험적인 기능을 제공합니다. 사이드 메뉴의 **실험실** 또는 **사이트맵**을 통해 접근할 수 있습니다. 제품에 따라 제공하는 기능이 다를 수 있습니다.

사이트맵으로 실험실 메뉴 접근하기

사이드 메뉴에서 **실험실**가 보이지 않는다면 **사이트맵**으로 접근할 수 있습니다.

1. 전체 화면 왼쪽 아래에 **사이트맵** 아이콘을 클릭하세요.
2. **사이트맵** 창의 오른쪽 아래에 **실험실** 섹션에서 원하는 메뉴를 선택하세요.

주의

실험실의 기능은 현재 개발 중인 베타 버전으로 예기치 않은 오류가 발생할 수 있습니다.
해당 기능 사용 시 주의가 필요합니다. 중요한 데이터나 운영 환경에서 사용을 권장하지 않습니다. 피드백이나 문제점이 발생하면 지원팀(support@whatap.io)으로 문의하시기 바랍니다.

MXQL 데이터 조회

MXQL은 와탭의 성능 데이터(메트릭스)를 유연하게 조회하기 위한 쿼리 언어입니다. 하나의 프로젝트에 포함된 여러 에이전트에서 수집한 메트릭스들을 종합적으로 조회하고 활용하기 위해서 사용합니다. **MXQL**에 대한 자세한 내용은 다음 문서를 참고하세요.

Pending Pod 현황

Pod의 상태와 오류 메시지를 확인하고 클러스터의 리소스 상태, 네트워크 정책, 스케줄러 로그 등을 검토하는데 유용한 기능을 제공합니다.

이 기능은 쿠버네티스 마스터 에이전트 1.7.0 버전 이상에서 지원합니다.

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.