

MCP(Model Context Protocol)

release date. 2026. 09. 10.

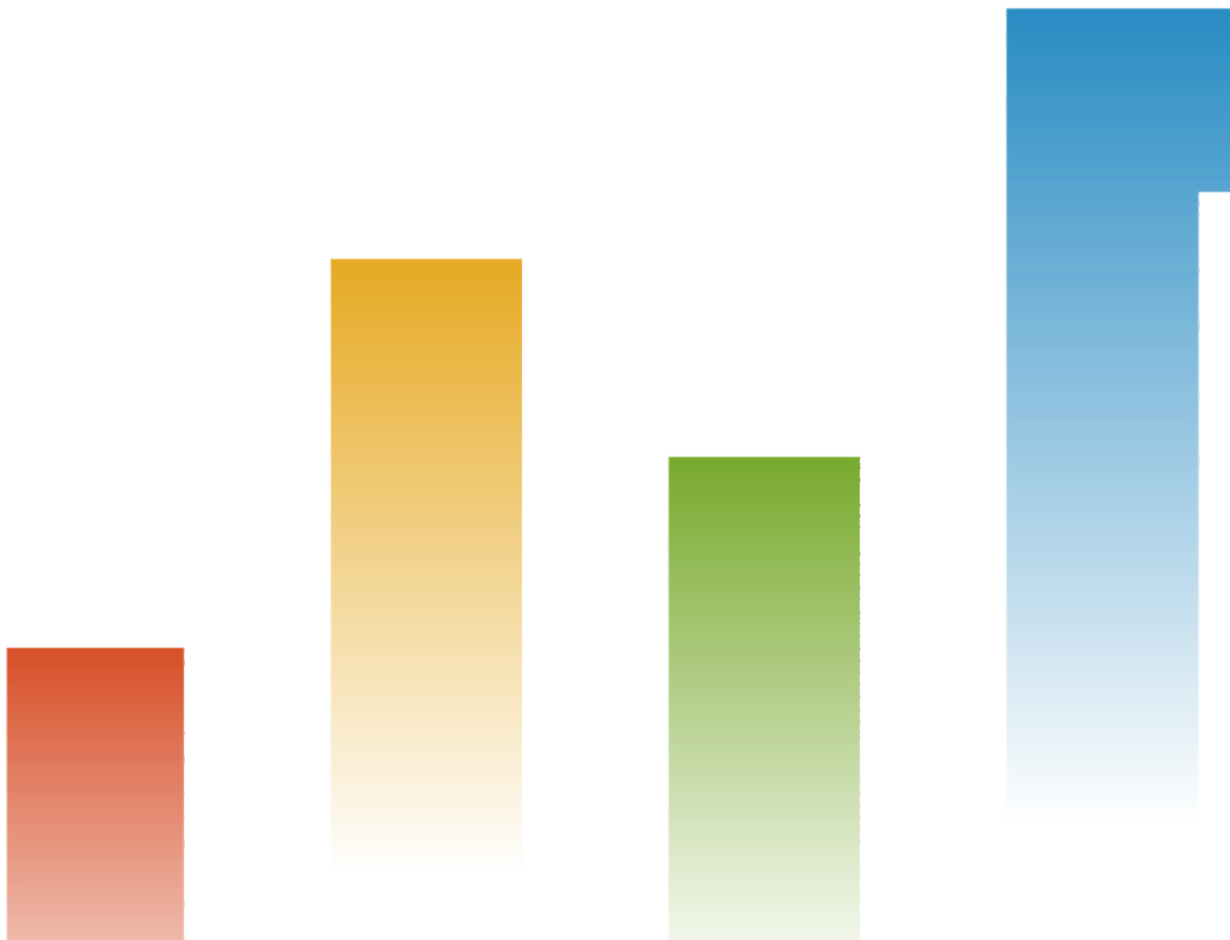


Table of Contents

• MCP	3
◦ 시작하기	4
◦ 설정하기	8
▪ 환경 설정	14
▪ 제거하기	17
◦ 사용하기	19
▪ 활용 예시	23
◦ 문제 해결	28

MCP(Model Context Protocol)

WhaTap MCP는 AI 어시스턴트가 외부 도구를 호출하는 표준 프로토콜인 MCP(Model Context Protocol)를 통해 WhaTap API를 연동하는 서버입니다. 사용자는 자연어 한 줄로 서버 CPU, APM TPS, K8s Pod 상태 등의 모니터링 데이터를 즉시 조회할 수 있습니다.

- 도구: 10개 (프로젝트 관리 3, 데이터 탐색 3, 복합 분석 2, 설치 및 PromQL 2)
- 지원 도메인: Server, APM, Kubernetes, Database, Log
- 현재 버전: v1.2.1
- 오픈소스: github.com/whatap/whatap-open-mcp

❗ 설치된 버전은 `npx -y github:whatap/whatap-open-mcp --version` 으로 확인할 수 있습니다.

WhaTap MCP 주요 특징점

- 복잡한 대시보드 탐색 없이 자연어 질문만으로 모니터링 데이터를 조회할 수 있습니다.
- 서버, APM, 쿠버네티스 등 다양한 WhaTap 모니터링 제품의 데이터를 통합 조회할 수 있습니다.
- 기존 AI 워크플로우에 모니터링 컨텍스트를 자연스럽게 통합할 수 있습니다.

지원 AI 클라이언트

클라이언트	제공사	설정 방식
Claude Desktop	Anthropic	JSON 설정 파일
Claude Code	Anthropic	CLI 한 줄 명령
Codex CLI	OpenAI	CLI 명령 또는 TOML 설정
Gemini CLI	Google	CLI 명령 또는 JSON 설정

시작하기

AI 어시스턴트(Claude, Codex, Gemini 등)에서 WhaTap 모니터링 데이터를 자연어로 조회할 수 있는 MCP 서버입니다. WhaTap MCP는 다음 절차로 시작합니다.

- 1단계. 사전 준비
- 2단계. 회원 가입하기
- 3단계. API 토큰 발급
- 4단계. AI 클라이언트 설정

사전 준비

WhaTap MCP 서버를 사용하기 위해 아래 항목을 미리 준비하세요.

Item	Details
Node.js	18.0.0 이상 (npm, npx 포함)
Git	Windows 환경에서만 필수
WhaTap 계정	whatap.io 무료 또는 유료 계정
AI 클라이언트	Claude Desktop, Claude Code, Codex CLI, Gemini CLI 중 하나

Node.js 설치 확인하기

터미널에서 아래 명령어를 실행하세요. 버전이 낮거나 설치되어 있지 않으면 설치 방법을 참고하세요.

```
node --version # v18.x.x 이상이면 OK
npm --version # 버전이 출력되면 OK
```

Node.js 설치하기

사용 중인 운영체제에 맞는 방법으로 Node.js를 설치하세요.

macOS

```
brew install node
```

❗ Homebrew가 없는 경우 <https://brew.sh> 에서 먼저 설치하거나, <https://nodejs.org> 에서 직접 다운로드하세요.

Windows

```
winget install OpenJS.NodeJS.LTS
```

❗ 또는 <https://nodejs.org> 에서 LTS 버전 설치 파일을 다운로드하세요. 설치 후 터미널을 새로 열어야 합니다.

Ubuntu / Debian

```
curl -fsSL https://deb.nodesource.com/setup_22.x | sudo -E bash -  
sudo apt install -y nodejs
```

❗ apt install nodejs 로 설치하면 구버전(v12~v16)이 설치됩니다. 반드시 NodeSource를 사용하세요.

Fedora / RHEL / Amazon Linux

```
sudo dnf install -y nodejs npm
```

nvm으로 설치(모든 OS)

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.40.3/install.sh | bash
source ~/.bashrc
nvm install --lts
```

Git 설치하기

WhaTap MCP 서버는 GitHub 저장소를 직접 참조할 때 Git을 사용합니다. macOS와 Linux는 Git이 기본으로 들어 있으므로 Windows 환경에서만 설치하세요.

```
winget install --id Git.Git -e --source winget
```

❗ Windows Package Manager(winget)를 사용하지 않는 경우 <https://git-scm.com> 에서 설치 파일을 다운로드하세요. 설치 후 터미널을 새로 열어야 합니다.

설치를 마쳤다면 터미널에서 아래 명령어를 실행하세요.

```
git --version # 버전이 출력되면 OK
```

회원 가입

와탭 모니터링 서비스를 사용하려면 먼저 [회원 가입](#)이 필요합니다. 회원 가입에 관한 자세한 내용은 [계정 관리 > 회원 가입](#) 문서를 참고하세요.

API 토큰 발급

WhaTap MCP 서버가 데이터를 조회할 때 사용할 API 토큰을 발급합니다.

1. [WhaTap Console](#)에 로그인합니다.
2. 우측 상단 **프로필 아이콘**을 클릭합니다.

3. **계정 관리**를 선택합니다.
4. **API 토큰** 항목에서 토큰을 복사합니다.

ⓘ 이 토큰은 **계정 레벨 토큰**입니다. 해당 계정에 속한 모든 프로젝트의 데이터를 **조회**할 수 있습니다. 데이터를 변경하거나 삭제하는 기능은 없으므로 안심하고 사용하세요. 단, 토큰은 외부에 노출되지 않도록 관리해 주세요.

설정하기

사용하는 AI 클라이언트에 맞는 설정 방법입니다. 어떤 클라이언트를 선택하든 동일한 MCP 서버를 사용하며 설정 형식만 다릅니다.

!

모든 클라이언트에서 GitHub 저장소를 직접 참조합니다: `npx -y github:whatap/whatap-open-mcp`

저장소 참조에 Git을 사용하므로 Windows 환경에서는 먼저 설치하세요. 자세한 절차는 [시작하기 > Git 설치하기](#) 문서를 참고하세요.

AI Client	Format	Config file	Command
Claude Desktop	JSON	<code>claude_desktop_config.json</code>	파일 직접 편집
Claude Code	-	-	<code>claude mcp add whatap -e WHATAP_API_TOKEN=... -- npx -y github:whatap/whatap-open-mcp</code>
Codex CLI	TOML	<code>~/.codex/config.toml</code>	<code>codex mcp add whatap --env ... -- npx -y github:whatap/whatap-open-mcp</code>
Gemini CLI	JSON	<code>~/.gemini/settings.json</code>	<code>gemini mcp -s user add whatap -e ... -- npx -y github:whatap/whatap-open-mcp</code>

Claude Desktop

설정 파일에 WhaTap MCP 서버 정보를 추가합니다.

OS	Path
macOS	<code>~/Library/Application Support/Claude/claude_desktop_config.json</code>
Linux	<code>~/.config/Claude/claude_desktop_config.json</code>

OS	Path
Windows	%APPDATA%\Claude\claude_desktop_config.json

설정 추가하기

설정 파일을 열어 아래 내용을 추가합니다.

Claude Desktop 설정 추가

```
{
  "mcpServers": {
    "whatap": {
      "command": "npx",
      "args": ["-y", "github:whatap/whatap-open-mcp"],
      "env": {
        "WHATAP_API_TOKEN": "여기에_토큰_입력"
      }
    }
  }
}
```

ⓘ 이미 다른 MCP 서버가 설정되어 있다면, 기존 "mcpServers" 안에 "whatap": { ... } 블록만 추가하세요.

등록 확인하기

설정 파일을 저장한 후 Claude Desktop을 다시 실행하면 WhaTap MCP 서버가 연결됩니다.

⚠ Claude Desktop은 창 닫기가 아닌 앱 종료 후 다시 실행해야 설정이 적용됩니다.

Claude Code (CLI)

터미널 명령어로 WhaTap MCP 서버를 등록합니다.

적용 범위 선택하기

등록 전에 사용 범위를 먼저 결정합니다.

Scope	Option	Description
전체 프로젝트	<code>-s user</code>	내 계정의 모든 프로젝트에서 사용
현재 프로젝트	<code>-s project</code>	현재 프로젝트에서만 사용 (<code>.mcp.json</code> 에 저장)

명령어로 등록하기

Claude Code (CLI) 등록

```
# 전체 프로젝트 (user scope)
claude mcp add whatap -s user \
-e WHATAP_API_TOKEN=여기에_토큰_입력 \
-- npx -y github:whatap/whatap-open-mcp

# 현재 프로젝트 (project scope)
claude mcp add whatap -s project \
-e WHATAP_API_TOKEN=여기에_토큰_입력 \
-- npx -y github:whatap/whatap-open-mcp
```

등록 확인하기

Claude Code (CLI) 등록 확인

```
claude mcp list
```

Codex CLI (OpenAI)

명령어 또는 설정 파일로 WhaTap MCP 서버를 등록합니다.

등록하기

두 방법 중 하나를 선택해 등록할 수 있습니다.

- **명령어로 등록하기** 설정 파일 직접 편집해 등록하기

Codex CLI (OpenAI) 명령어로 등록

```
codex mcp add whatap \
--env WHATAP_API_TOKEN=여기에_토큰_입력 \
-- npx -y github:whatap/whatap-open-mcp
```

~/codex/config.toml

```
[mcp_servers.whatap]
command = "npx"
args = ["-y", "github:whatap/whatap-open-mcp"]

[mcp_servers.whatap.env]
WHATAP_API_TOKEN = "여기에_토큰_입력"
```

❗ Codex는 JSON이 아닌 TOML 형식을 사용합니다.

등록 확인하기

Codex CLI (OpenAI) 등록 확인

```
codex mcp list
```

Gemini CLI (Google)

명령어 또는 설정 파일로 WhaTap MCP 서버를 등록합니다.

적용 범위 선택하기

등록 전에 사용 범위를 먼저 결정합니다.

Scope	File location	Purpose
전체 (user)	~/.gemini/settings.json	개인 설정, 모든 프로젝트
프로젝트	.gemini/settings.json (프로젝트 루트)	팀 공유용

등록하기

두 방법 중 하나를 선택해 등록할 수 있습니다.

- 명령어로 등록하기 설정 파일 직접 편집해 등록하기

Gemini CLI (Google) 등록

```
gemini mcp -s user add whatap \
-e WHATAP_API_TOKEN=여기에_토큰_입력 \
npx -- -y github:whatap/whatap-open-mcp
```

~/.gemini/settings.json

```
{
  "mcpServers": {
    "whatap": {
      "command": "npx",
      "args": ["-y", "github:whatap/whatap-open-mcp"],
      "env": {
        "WHATAP_API_TOKEN": "여기에_토큰_입력"
      }
    }
  }
}
```

등록 확인하기

Gemini CLI (Google) 등록 확인

```
gemini mcp list
```

연결 확인하기

1. 설정을 완료한 후 AI 클라이언트를 엽니다.
2. 아래와 같이 입력하세요.

```
내 WhaTap 프로젝트 목록 보여줘
```

3. 프로젝트 코드(pcode), 이름, 플랫폼, 제품 유형이 포함된 목록이 나타나면 연결이 정상입니다. 목록이 나타나지 않으면 문제 해결 문서를 참고하세요.

환경 설정

WhaTap MCP 서버는 WHATAP_API_TOKEN 하나만 설정하면 SaaS 환경에서 바로 사용할 수 있습니다. 공공기관이나 On-premises 환경이라면 WHATAP_API_URL 을 추가로 설정해야 합니다.

환경 변수

변수	필수	기본값	설명
WHATAP_API_TOKEN	O	-	WhaTap 계정 API 토큰
WHATAP_API_URL	X	https://api.whatap.io	API 엔드포인트 (SaaS 환경은 변경 불필요)

API 엔드포인트

WhaTap SaaS 환경에서는 모든 리전(서울, 도쿄, 싱가포르, 프랑크푸르트 등)이 동일한 API URL(https://api.whatap.io)을 사용합니다. 리전별 라우팅은 서버에서 자동 처리되므로 별도 설정이 필요 없습니다.

아래의 경우에만 WHATAP_API_URL 을 변경하세요.

환경	URL	비고
SaaS(기본)	https://api.whatap.io	변경 불필요
공공기관 / 정부망	https://api.gov.whatap.io	WHATAP_API_URL 설정 필요
On-premises	고객사 내부 URL	WHATAP_API_URL 설정 필요

공공기관용 설정 예시

Claude Desktop

Claude Desktop

```
{
  "mcpServers": {
    "whatap": {
      "command": "npx",
      "args": ["-y", "github:whatap/whatap-open-mcp"],
      "env": {
        "WHATAP_API_TOKEN": "여기에_토큰_입력",
        "WHATAP_API_URL": "https://api.gov.whatap.io"
      }
    }
  }
}
```

Claude Code

Claude Code

```
claude mcp add whatap \
-e WHATAP_API_TOKEN=여기에_토큰_입력 \
-e WHATAP_API_URL=https://api.gov.whatap.io \
-- npx -y github:whatap/whatap-open-mcp
```

Codex CLI

Codex CLI

```
[mcp_servers.whatap]
command = "npx"
args = ["-y", "github:whatap/whatap-open-mcp"]

[mcp_servers.whatap.env]
WHATAP_API_TOKEN = "여기에_토큰_입력"
```

```
WHATAP_API_URL = "https://api.gov.whatap.io"
```

Gemini CLI

Gemini CLI

```
gemini mcp -s user add whatap \
-e WHATAP_API_TOKEN=여기에_토큰_입력 \
-e WHATAP_API_URL=https://api.gov.whatap.io \
npx -- -y github:whatap/whatap-open-mcp
```

WhaTap 서비스 리전

프로젝트의 리전은 WhaTap Console에서 프로젝트 생성 시 선택합니다. 리전이 달라도 API URL은 동일합니다.

리전	위치	클라우드
서울	대한민국	AWS, Azure, 카카오클라우드
도쿄	일본	AWS, Azure
싱가폴	싱가폴	AWS
자카르타	인도네시아	AWS
뭄바이	인도	AWS
캘리포니아	미국 서부	AWS
버지니아	미국 동부	AWS
프랑크푸르트	독일	AWS

제거하기

WhaTap MCP 서버를 더 이상 사용하지 않거나 설정을 초기화하려는 경우, AI 클라이언트에서 MCP 서버를 제거해야 합니다.

제거 방법

Claude, Codex, Gemini 등 주요 AI 클라이언트에서 WhaTap MCP 서버를 제거하는 방법입니다.

Claude Desktop

1. `claude_desktop_config.json` 에서 `"whatap"` 블록을 삭제합니다.
2. Claude Desktop을 재시작합니다.

Claude Code

```
claude mcp remove whatap
```

Codex CLI

`~/.codex/config.toml` 에서 `[mcp_servers.whatap]` 섹션을 삭제합니다.

Gemini CLI

```
gemini mcp remove whatap
```

또는 `~/.gemini/settings.json` 에서 `"whatap"` 블록을 삭제합니다.

캐시 삭제(선택)

npx 캐시를 삭제하면 실행 시 최신 버전을 받아옵니다.

```
npx clear-npx-cache
```

사용하기

WhaTap MCP 서버에 연결한 후 AI 어시스턴트로 모니터링 데이터를 조회하는 방법을 안내합니다.

동작 방식

AI 어시스턴트에게 자연어로 질문하면 위 과정을 자동으로 처리합니다.

- 프로젝트 목록 조회** — 프로젝트 코드(pcode)를 확인합니다.
- 데이터 카탈로그 탐색** — 조회할 수 있는 데이터 종류를 확인합니다.
- 쿼리 실행** — 원하는 데이터를 조회합니다.

✔ "프로젝트 목록 보여줘"로 시작해서 프로젝트 코드를 먼저 확인하세요. 이후 해당 코드를 사용하면 더 정확한 결과를 얻을 수 있습니다.

제공 도구 목록

MCP 서버는 총 10개의 도구를 AI 어시스턴트에 제공합니다. 사용자가 직접 도구를 호출할 필요는 없으며, 자연어로 질문하면 AI가 적절한 도구를 자동으로 선택합니다.

프로젝트 관리 도구

도구	기능
<code>whatap_list_projects</code>	전체 모니터링 프로젝트 목록 조회 (pcode, 이름, 플랫폼)
<code>whatap_project_info</code>	특정 프로젝트의 상세 정보 조회
<code>whatap_list_agents</code>	프로젝트 내 에이전트(서버/인스턴스) 목록 조회

데이터 탐색 도구

도구	기능
<code>whatap_data_availability</code>	900개 이상 MXQL 쿼리 탐색, 20+ 카테고리 실시간 데이터 감지, OpenMetrics 메트릭 목록, 저장 PromQL 쿼리 표시
<code>whatap_describe_query</code>	MXQL 경로 또는 OpenMetrics 메트릭 상세 조회 (파라미터, 출력 필드, 메트릭 타입, 추천 PromQL)
<code>whatap_query_data</code>	MXQL, PromQL, 저장 쿼리 실행 (단위 변환, 임계값, 분석 가이드 포함)

복합 분석 도구

도구	기능
<code>whatap_apm_anomaly</code>	APM 이상 탐지 — TPS, 응답시간, 에러, Active TX 4개 쿼리를 동시 실행하여 통계 분석
<code>whatap_service_topology</code>	서비스 간 연결 관계 및 네트워크 병목 탐지 (NPM 에이전트 필요)

설치 및 PromQL 도구

도구	기능
<code>whatap_install_agent</code>	29개 플랫폼 에이전트 설치 명령어 생성 (접속 정보 자동 포함, 플랫폼 자동 감지)
<code>whatap_create_promql</code>	PromQL 쿼리 작성, 실시간 검증, 저장 (재사용 가능)

whatap_install_agent 지원 플랫폼

카테고리	플랫폼
인프라 OS	Debian/Ubuntu, Amazon Linux, RHEL/CentOS, SUSE, FreeBSD, Windows 등 9종

카테고리	플랫폼
APM	Java, Node.js, Python, PHP, .NET, Go
데이터베이스	PostgreSQL, Oracle, MySQL, Redis, MongoDB, DB2 등 12종
컨테이너	Kubernetes (Helm)
서버 앱	Kafka, NGINX, Apache 등 7종

도구별 사용 흐름

데이터 조회 워크플로우 (MXQL)

```

whatap_list_projects → projectCode 확인
↓
whatap_data_availability(projectCode) → 사용 가능한 MXQL 경로 확인
↓
whatap_describe_query(path) → 파라미터, 필드, MXQL 원문 확인
↓
whatap_query_data(projectCode, path) → 데이터 조회

```

PromQL 워크플로우 (쿠버네티스/OpenMetrics)

```

whatap_data_availability(projectCode) → OpenMetrics 메트릭 목록 확인
↓
whatap_create_promql(projectCode, name, query) → PromQL 검증 및 저장
↓
whatap_query_data(projectCode, savedQuery="name") → 저장 쿼리 재사용

```

에이전트 설치 워크플로우

```

whatap_list_projects → projectCode + 플랫폼 확인
↓
whatap_install_agent(projectCode) → 접속 정보 + 설치 명령어 출력
↓
whatap_list_agents(projectCode) → 에이전트 등록 확인

```

✓ 이 흐름은 AI가 자동으로 진행합니다. 사용자는 원하는 내용을 자연어로 질문하기만 하면 됩니다.

주요 MXQL 경로 레퍼런스

`whatap_data_availability` 호출 없이 바로 사용할 수 있는 주요 경로 목록입니다.

도메인	경로	설명
Server	<code>v2/sys/server_base</code>	CPU, 메모리, 기본 지표
Server	<code>v2/sys/server_disk</code>	디스크 사용량
Server	<code>v2/sys/server_network</code>	네트워크 I/O
APM	<code>v2/app/tps_pcode</code>	TPS (프로젝트 전체)
APM	<code>v2/app/resp_time_pcode</code>	응답시간 (프로젝트 전체)
APM	<code>v2/app/tx_error_pcode</code>	에러 수 (프로젝트 전체)
K8s	<code>v2/container/kube_pod</code>	Pod 상태
K8s	<code>v2/container/kube_node</code>	노드 상태
K8s	<code>v2/container/kube_event</code>	클러스터 이벤트
DB	<code>v2/db/instance_active_session</code>	활성 세션

활용 예시

자연어 질문 예시와 실제 대화 흐름을 통해 WhaTap MCP 서버 활용 방법을 확인하세요.

자주 사용하는 질문

상황	질문 예시
처음 시작할 때	"내 프로젝트 목록 보여줘"
서버 상태 확인	"최근 5분간 서버 CPU, 메모리 상태 요약해줘"
APM 이상 확인	"이상 징후가 있는 에이전트 찾아줘"
장애 대응	"최근 5분간 에러 현황 요약해줘"
데이터 탐색	"이 프로젝트에서 조회 가능한 데이터가 뭐가 있어?"
K8s 점검	"Pod 상태 확인해줘"
서비스 관계	"서비스 토폴로지 보여줘"
OpenMetrics 탐색	"프로젝트 33194에서 사용 가능한 OpenMetrics 메트릭 보여줘"
PromQL 생성	"Pod별 CPU 사용률 PromQL 쿼리 만들어줘"
저장 쿼리 실행	"저장된 'CPU by Pod' 쿼리 실행해줘"
에이전트 설치	"프로젝트 12345에 모니터링 에이전트 설치 방법 알려줘"

자연어 질문 예시

모니터링 작업에 사용할 자연어 질문 예시입니다. 원하는 작업에 맞는 질문을 선택하거나 참고해 작성하세요.

프로젝트 탐색

내 WhaTap 프로젝트 목록 보여줘
프로젝트 12345의 상세 정보 알려줘
프로젝트 12345에 어떤 에이전트가 있어?

서버 모니터링

최근 1시간 서버 CPU 사용률 보여줘
메모리 사용량이 80% 넘는 서버가 있어?
디스크 사용량 확인해줘
CPU 사용률 높은 순으로 서버 Top 5 보여줘

APM (애플리케이션 성능)

최근 5분간 TPS 추이 보여줘
평균 응답시간이 어떻게 되나?
트랜잭션 에러가 발생했어?
현재 진행 중인 트랜잭션 상태 확인해줘

쿠버네티스

Pod 상태 확인해줘
노드별 CPU, 메모리 사용량 보여줘
최근 클러스터 이벤트 중 Warning이 있어?

데이터베이스

DB 인스턴스 목록 보여줘
현재 활성 세션 확인해줘

이상 탐지

최근 5분간 이상 징후가 있는 에이전트 찾아줘
높은 감도로 1시간 동안 이상 탐지 실행해줘

서비스 토폴로지

서비스 간 연결 관계 보여줘
네트워크 병목 지점이 있어?

OpenMetrics / PromQL

프로젝트 33194에서 사용 가능한 OpenMetrics 메트릭 보여줘
Pod별 CPU 사용률 PromQL 쿼리 만들어줘
저장된 'CPU by Pod' 쿼리 실행해줘

에이전트 설치

프로젝트 12345에 모니터링 에이전트 설치 방법 알려줘
Java APM 에이전트 설치해줘
쿠버네티스 에이전트 설정 방법 보여줘
PostgreSQL DB 모니터링 설치해줘
Debian 서버에 인프라 에이전트 설치해줘

복합 질문

아침 헬스체크 해줘.
전체 프로젝트에서 서버 상태 확인하고,
APM 프로젝트에 이상 징후가 있는지 분석해줘.

대화 예시

아래 대화는 예시이며, 실제 응답은 프로젝트의 모니터링 데이터에 따라 달라집니다.

예시 1. 아침 서버 점검

사용자: 아침 점검 좀 해줘. 서버 CPU, 메모리 상태 요약해줘.

AI: 3대 서버를 확인했습니다.

- web-01: CPU 45%, Memory 62% (정상)
- web-02: CPU 87%, Memory 71% (CPU 주의 — 권장 70% 초과)
- db-01: CPU 23%, Memory 55% (정상)

web-02 서버의 CPU가 87%로 높습니다. 프로세스를 확인해 보시겠습니까?

예시 2. APM 이상 탐지

사용자: Java APM 프로젝트에 이상 징후가 있는지 확인해줘.

AI: 5개 에이전트를 분석했습니다.

이상 징후 발견 (1건):

- api-server-03: 응답시간 급증 (평균 120ms → 최대 2.3초, +4.2σ)

나머지 4개 에이전트는 정상 범위입니다.

api-server-03의 트랜잭션 상세를 확인하시겠습니까?

예시 3. 장애 대응

사용자: 방금 에러 알림이 왔어. 최근 5분간 에러 상황 요약해줘.

AI: 최근 5분간 트랜잭션 에러 현황입니다.

| 에이전트 | TPS | 에러수 | 에러율 |

```
| --- | --- | --- | --- |
| api-01 | 45.2 | 23 | 10.2% |
| api-02 | 42.8 | 2 | 0.9% |
| api-03 | 44.1 | 0 | 0% |
```

api-01에서 에러율 10.2%로 이상 수준입니다.
응답시간 추이와 Active TX도 함께 확인하시겠습니까?

예시 4. 경영진 보고용 데이터

사용자: 이번 주 전체 서비스 응답시간과 에러율 추이를 요약해줘.

AI: 최근 7일간 서비스 현황입니다.

- 평균 응답시간: 142ms (전주 대비 -8%)
- 평균 에러율: 0.3% (전주 대비 변동 없음)
- TPS 평균: 1,240건/초 (전주 대비 +12%)

전반적으로 양호한 상태이며, TPS 증가에 비해 응답시간이
오히려 감소하여 성능이 개선된 것으로 판단됩니다.

문제 해결

MCP 서버 연결 또는 사용 중 문제가 발생하면 아래 항목을 확인하세요.

"WHATAP_API_TOKEN environment variable is required"

Q API 토큰이 설정되지 않았거나 MCP 서버에 전달되지 않습니다.

A 각 클라이언트의 설정 파일에서 다음 항목을 확인하세요.

- Claude Desktop: "env" 블록에 "WHATAP_API_TOKEN" 키가 있는지 확인 ("WHATAP_TOKEN" 아님)
- Codex CLI: [mcp_servers.whatap.env] 섹션이 TOML 파일에 있는지 확인
- Gemini CLI: "env" 블록이 settings.json에 있는지 확인
- Claude Code: `claude mcp add` 명령에 `-env` 플래그가 포함되어 있는지 확인

"No API token found for project XXXXX"

Q 계정 토큰은 유효하지만, 해당 프로젝트 코드가 존재하지 않거나 접근 권한이 없습니다.

A AI 어시스턴트에 "내 프로젝트 목록 보여줘"라고 입력하여 정확한 프로젝트 코드를 확인하세요.

"npx: command not found"

Q Node.js가 설치되어 있지 않거나 PATH에 등록되지 않습니다.

A Node.js 18 이상을 설치하세요.

```
node --version # v18 이상 확인  
npx --version # 버전 출력 확인
```

"spawn git ENOENT" 또는 "An unknown git error occurred"

Q Git이 설치되어 있지 않거나 실행 경로가 PATH에 등록되지 않습니다.

A WhaTap MCP 서버는 GitHub 저장소를 직접 참조하므로 Git이 필요합니다. 이 오류는 주로 Windows 환경에서 발생합니다.

```
winget install --id Git.Git -e --source winget
```

설치 후 터미널을 새로 열고 아래 명령어를 실행하세요.

```
git --version # 버전 출력 확인
```

자세한 절차는 [시작하기](#) > [Git 설치하기](#) 문서를 참고하세요.

MCP 서버가 시작되지 않거나 타임아웃 발생

Q MCP 서버가 시작되지 않거나 타임아웃이 발생합니다.

A 서버를 직접 실행하여 오류 메시지를 확인하세요.

```
WHATAP_API_TOKEN=여기에_토큰_입력 npx -y github:whatap/whatap-open-mcp
```

서버가 정상 실행되면 입력 대기 상태가 됩니다. **Ctrl+C** 로 종료하세요. 오류 메시지가 출력되면 토큰 값 또는 Node.js 버전을 확인하세요.

Claude Desktop에서 도구가 나타나지 않는 경우

Q Claude Desktop에서 MCP 도구가 표시되지 않습니다.

A 다음 항목을 순서대로 확인하세요.

1. JSON 파일이 유효한지 확인합니다(쉼표 누락, 괄호 불일치 등).
2. Claude Desktop을 완전히 종료 후 재시작합니다(창 닫기가 아닌 앱 종료).
3. 로그를 확인합니다. `~/Library/Logs/Claude/mcp*.log` (macOS)

Codex CLI에서 도구가 나타나지 않는 경우

Q Codex CLI에서 MCP 도구가 표시되지 않습니다.

A 다음 항목을 순서대로 확인하세요.

1. `codex mcp list` 로 등록 여부를 확인합니다.
2. `~/.codex/config.toml` 이 올바른 TOML 형식인지 확인합니다(JSON 문법 사용 불가).
3. 삭제 후 재등록합니다. `codex mcp add whatap ...`

Gemini CLI에서 도구가 나타나지 않는 경우

Q Gemini CLI에서 MCP 도구가 표시되지 않습니다.

A 다음 항목을 순서대로 확인하세요.

1. `gemini mcp list` 로 등록 여부를 확인합니다.
2. `~/.gemini/settings.json` 이 올바른 JSON 형식인지 확인합니다.
3. 적용 범위를 확인합니다. `-scope user` (전체) 또는 `-scope project` (프로젝트)

"WhaTap API error (401)" 또는 "(403)"

Q API 토큰이 유효하지 않거나 만료되었습니다.

A WhaTap Console에서 **계정 관리 > API 토큰**으로 이동하여 새 토큰을 발급하세요.

응답이 느린 경우

Q 응답이 느려요.

A 조회 시간 범위를 줄이세요. "5m" (5분)이면 수 초, "1d" (1일)이면 수십 초 걸릴 수 있습니다. 연속 요청 시 WhaTap API 속도 제한에 걸릴 수 있습니다. 잠시 후 다시 시도하세요.

데이터가 조회되지 않는 경우

Q 데이터가 조회되지 않습니다.

A 원인과 확인 방법을 참고하세요.

원인	확인 방법
프로젝트 유형 불일치 (예: Server 프로젝트에 APM 쿼리)	<code>whatap_data_availability(projectCode)</code> 로 활성 카테고리 확인 → "이 프로젝트에서 조회 가능한 데이터 보여줘"
에이전트 미설치 또는 비활성	<code>whatap_list_agents(projectCode)</code> 로 에이전트 상태 확인 → "에이전트 목록 보여줘"
시간 범위가 너무 좁음	<code>timeRange="1h"</code> 등 범위 확대 후 재시도