



Monitoring makes easy

Node.js Application Monitoring Agent Install Guide

이 문서는 와탭 APM 서비스 사용자가 에이전트 설치를 돕기 위해 작성된 문서입니다.
이 문서는 와탭랩스의 고유한 자산으로 재배포 또는 사용을 위해서는
와탭랩스(support@whatap.io) 에 연락주시기 바랍니다.

목차

1.1. Node.js 애플리케이션 모니터링	3
1.1.1. 에이전트 실행 및 모니터링 개요	3
1.1.1.1. 에이전트 설치 방식 개요	3
1.1.1.2. 구성 파일	3
1.1.2. 에이전트 설치/실행/업데이트/중지	5
1.1.2.1. 설치	5
1.1.2.1.1. 프로젝트 생성	5
1.1.2.1.2. 에이전트 설치	6
1.1.2.1.3. 라이선스 발급	6
1.1.2.1.4. 코드 작성	7
1.1.2.2. 실행	7
1.1.2.3. 업데이트	8
1.1.2.4. 중지 (삭제)	8
1.1.3. 설치 에러 대응	8
1.1.3.1. 라이선스 오류	9
1.1.3.2. 네트워크 오류	9
1.1.4. FAQ	10
1.1.5. 제약 사항	10

1.1. Node.js 애플리케이션 모니터링

1.1.1. 에이전트 실행 및 모니터링 개요

와탭 Node.js 애플리케이션 모니터링은 Node.js 기반 웹 애플리케이션 서버 모니터링 서비스를 제공합니다.

1.1.1.1. 에이전트 설치 방식 개요

Node.js 모니터링 서비스를 사용하기 위해서는 모니터링 대상 애플리케이션에 모니터링 에이전트를 설치해야 합니다. 설치는 npm (Node Package Manager) 을 통해 가능 합니다.

- 서비스 중인 Node.js 파일의 폴더에 와탭 에이전트 모듈을 설치 합니다.
- 와탭 라이선스 환경설정 (whatap.conf) 을 합니다.
- Node.js 서비스를 재실행 합니다.

상세한 절차는 1.1.2. 에이전트 설치/실행을 참고해주세요.

1.1.1.2. 구성 파일

모니터링 정보를 수집하기 위한 트레이서, 수집된 정보를 서버에 전송하기 위한 에이전트, npm 설치를 지원하는 파일들로 구성됩니다.

Node.js 모니터링 서비스를 구성하는 각 파일의 역할은 다음과 같습니다.

WhaTap

파일명	설명
whatap.conf	에이전트 설정 파일 샘플 (복사해서 라이선스를 입력)
lib	에이전트, 트레이서 프로그램
README.md	에이전트 설치 매뉴얼
package.json	npm 모듈 환경구성

와탭은 모니터링 정보 수집 대상인 애플리케이션 서버 식별을 위한 정보로 기본적으로 애플리케이션 서버로부터 수집한 정보를 활용합니다. 기본적으로 활용하는 정보는 애플리케이션 서버 종류, 애플리케이션 서버의 IP, 서비스 포트를 조합하여 애플리케이션 서버를 고유 식별자로 사용하게 되며 필요에 따라 사용자가 지정한 명칭을 사용하거나 패턴을 변경하여 사용하는 것도 가능합니다. 이때에는 꼭 고유한 값이어야 합니다. 애플리케이션 서버로부터 추출한 정보를 활용하는 이유는 애플리케이션 서버 정지, 네트워크 단절 또는 에이전트 문제로 인한 수집 서버와 에이전트의 통신 단절 상태가 복구되었을 경우, 재 접속된 에이전트로부터 송신되는 정보가 기존 에이전트로부터 송신된 정보와의 연속성을 유지하기 위해서 입니다. 와탭이 애플리케이션 서버를 식별하기 위해 사용하는 기본 패턴은 다음과 같습니다.

■ default: {type}-{ip2}-{ip3}-{process}

패턴 옵션

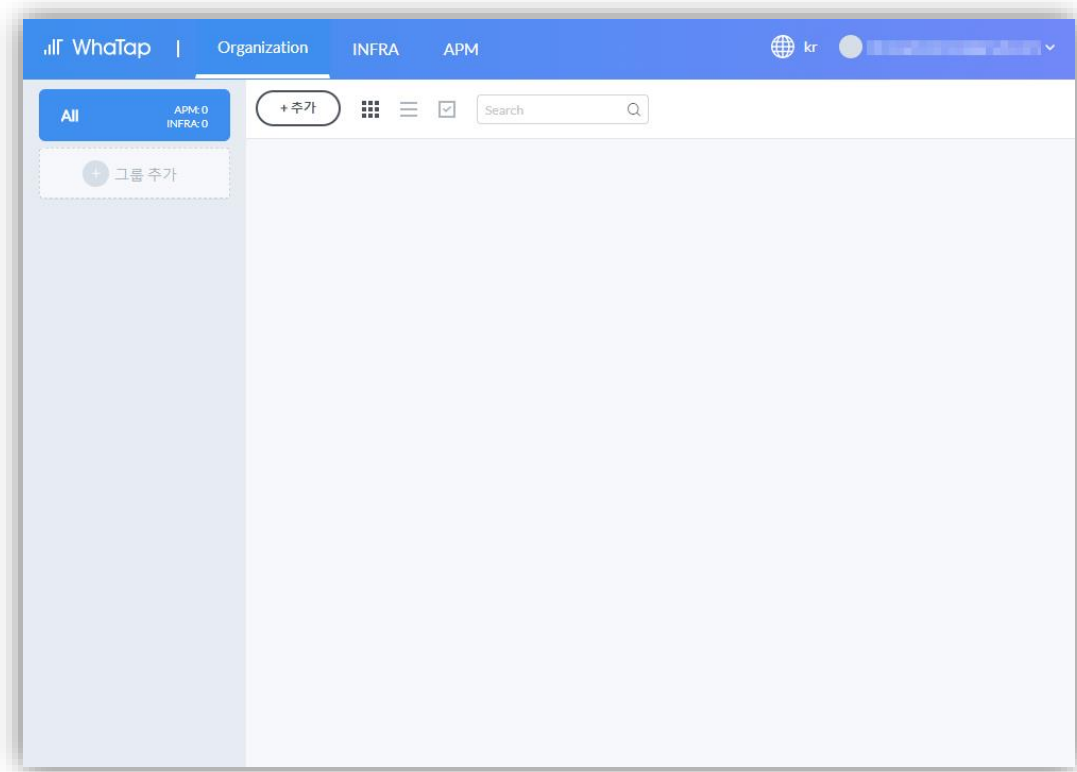
설정	설명
type	app_name
ip#	ip 를 .으로 나누었을 때 #번째 자리(0 부터)
process	app_process_name
hostname	호스트 명

WhaTap

1.1.2. 에이전트 설치/실행/업데이트/중지

1.1.2.1. 설치

1.1.2.1.1. 프로젝트 생성



서버를 등록하기 위해 우선 프로젝트를 생성합니다. 추가 버튼을 선택하면 아래와 같이 프로젝트 생성 창이 나타납니다. PHP 아이콘을 선택한 뒤, 희망하는 프로젝트명과 데이터 서버 지역(Region), 소속하게 될 그룹을 선택한 뒤 프로젝트를 생성합니다.

WhaTap



1.1.2.1.2. 에이전트 설치

npm (Node Package Manager)을 통해서 와탭 에이전트를 설치 합니다.

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

1.1.2.1.3. 라이선스 발급



License Key

node_modules/whatap에 있는 whatap.conf파일을 루트 디렉토리로 복사 후 라이선스 키를 발급받은 후 추가 해 주세요.
whatap.conf 파일은 다음과 같은 내용을 포함합니다.

[라이선스 발급받기](#)

WhaTap

프로젝트 관리화면에서는 우선적으로 라이선스를 발급 받습니다. 라이선스 키는 프로젝트별로 귀속되기 때문에, 유출되거나 배포되어서는 안됩니다.

node_modules/whatap 에 있는 whatap.conf 파일을 루트 디렉토리로 복사 후 라이선스 키를 발급 받은 후 추가해 주세요

1.1.2.1.4. 코드 작성



Usage

루트 디렉토리의 Nodejs Application에 다음과 같은 코드를 추가 해 주세요.

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

copy

루트 디렉토리의 Node.js 애플리케이션에 위와 같이 에이전트 코드를 삽입합니다. (예를 들어 express 의 경우 app.js)

1.1.2.2. 실행

애플리케이션을 실행합니다.

```
$ nodejs app.js
```

WhaTap

1.1.2.3. 업데이트

애플리케이션을 중단하고, npm 을 사용하여 업데이트합니다.

```
$ npm update whatap
```

애플리케이션을 재시작 합니다.

1.1.2.4. 중지 (삭제)

애플리케이션을 중단하고, npm 을 사용하여 모듈을 삭제 합니다.

```
$ npm uninstall whatap
```

관리->프로젝트 관리 페이지에서 프로젝트를 삭제 합니다.

The screenshot shows the WhaTap management interface. On the left is a blue sidebar with icons for home, project, settings, and help. The main content area is titled '관리 프로젝트 관리' (Management Project Management). It includes a '프로젝트 지역' (Project Region) dropdown set to 'Korea-AWS-Seoul' and an 'API토큰' (API Token) input field with a '재발급' (Reissue) button. Below this is a '초대된 멤버' (Invited Members) table with columns for '이메일' (Email), '이름' (Name), '직책 및 직급' (Position and Rank), and '핸드폰' (Phone). The table lists 'WhaTap Trial Support' with the name 'WhaTap' and phone number '02-565-1803'. A '+ 사용자 초대' (+ Invite User) button is below the table. At the bottom, there is a '프로젝트 삭제' (Delete Project) section with a '프로젝트 삭제' (Delete Project) button highlighted by a red box.

1.1.3. 설치 에러 대응

WhaTap

1.1.3.1 라이선스 오류

```
20180220 08:40:39 [WHATAP-100] Start initialize configuration...
20180220 08:40:40 [WHATAP-203] Config file reloaded
20180220 08:40:40 [WHATAP-101] Finish initialize configuration...
20180220 08:40:40 [WHATAP-110] [pcode=5464,SECURE_KEY=bac7d1a838910321b7ea1f635f22494a]
20180220 08:40:40 [WHATAP-170] [WhaTap Agent] now waiting for starting.....
20180220 08:40:45 [WHATAP-180] Try to connect to {"host":"52.78.209.94","port":6600,"agent":false}
20180220 08:40:45 [WHATAP-168] OID: 58376919 ONAME: NODE-0-2-3000 IP: 172.17.0.2
20180220 08:40:45 [WHATAP-183] Disconnected from server, please check your license key..
```

라이선스가 제대로 입력되지 않은 경우입니다. 아래 내용을 점검해주세요.

- whatap.conf 파일이 존재하는지
- whatap.conf 파일에 라이선스 키가 입력되어 있는지
- whatap.conf 에 기록되어 있는 프로젝트가 존재하고 사용가능한 상태인지

1.1.3.2 네트워크 오류

```
20180220 08:48:14 [WHATAP-100] Start initialize configuration...
20180220 08:48:14 [WHATAP-203] Config file reloaded
20180220 08:48:14 [WHATAP-101] Finish initialize configuration...
20180220 08:48:14 [WHATAP-110] [pcode=5464,SECURE_KEY=bac7d1a838910321b7ea1f635f22494a]
20180220 08:48:14 [WHATAP-170] [WhaTap Agent] now waiting for starting.....
20180220 08:48:19 [WHATAP-180] Try to connect to {"host":"52.78.209.94","port":6600,"agent":false}
20180220 08:48:19 [WHATAP-127] Failed to connect to {"host":"52.78.209.94","port":6600,"agent":false}
20180220 08:48:19 [WHTAP-11] TCP SESSION OPEN ERROR
```

와탭 에이전트가 설치되어 있는 서버의 네트워크가 사용가능한 상태인지 확인해 주세요.

와탭은 TCP 6600 포트를 사용하여 데이터를 수집하고 있습니다.

방화벽도 확인해주세요. 일반적으로 Out Bound 포트는 막혀 있지 않습니다.

WhaTap

1.1.4. FAQ

Q: 지원하는 운영체제는 뭔가요?

Windows, mac(osx), linux 등 Node.js 가 실행 가능한 환경이면 모두 지원합니다.

Q: 지원하는 Framework 은 뭔가요?

Express.js, Hapi.js, Mean.js, Koa.js 등 Node.js 기본 모듈을 사용하는 모든 Framework 입니다.

1.1.5. 제약 사항

Node.js 버전 4.3.2 이상을 지원합니다.

Thank you



support@whatap.io

(주)와탭랩스

www.whatap.io

서울특별시 강남구 테헤란로 69 길 5 유기타워 11F (06160)

Tel. 02.565.1803

Fax. 0504.848.1803

Monitoring makes ____ easy