



Monitoring makes ____ easy

PHP Application Monitoring Installation Guide

이 문서는 와탭 애플리케이션 모니터링 서비스 에이전트 설치 안내를 위해 작성된 문서입니다.
이 문서는 와탭 랩스의 고유한 자산으로 재배포 또는 사용을 위해서는 와탭랩스(support@whatap.io) 에 연락주시기 바랍니다.

목차

1.1. PHP 애플리케이션 모니터링.....	4
1.1.1. 에이전트 실행 및 모니터링 개요.....	4
1.1.1.1. 에이전트 설치 방식 개요.....	4
1.1.1.2. 구성 파일.....	5
1.1.1.3. 에이전트 이름 식별.....	5
1.1.2. 에이전트 설치/실행/업데이트/중지.....	6
1.1.2.1. 공통.....	6
1.1.2.1.1. 프로젝트 생성.....	6
1.1.2.1.2. 라이선스 발급.....	7
1.1.2.2. 에이전트 설치.....	8
1.1.2.2.1. 패키지 설치.....	8
1.1.2.2.1.1. RedHat/CentOS.....	8
1.1.2.2.1.1.1. 패키지 저장소(Repository) 등록.....	8
1.1.2.2.1.1.2. 패키지 설치.....	8
1.1.2.2.1.2. Debian/Ubuntu.....	8
1.1.2.2.1.2.1. 패키지 저장소(Repository) 등록.....	8
1.1.2.2.1.2.2. 패키지 설치.....	8
1.1.2.2.1.3. Amazon Linux.....	8
1.1.2.2.1.3.1. 패키지 저장소(Repository) 등록.....	8
1.1.2.2.1.3.2. 패키지 설치.....	9
1.1.2.2.1.4. FreeBSD.....	9
1.1.2.2.1.4.1. 패키지 설치.....	9
1.1.2.2.1.4.2. 시스템 UDP 관련 설정.....	9
1.1.2.2.2. PHP 확장 모듈(PHP Extension module) 및 whatap-php 서비스(Service) 등록.....	9
1.1.2.2.2.1. 설정 스크립트 install.sh.....	10
1.1.2.2.2.1.1. PHP 버전 지정 설치.....	11
1.1.2.2.2.1.2. PHP Cli 명령어 지정 설치.....	11
1.1.2.2.3. 애플리케이션 재기동.....	11
1.1.2.2.4. 설치 확인.....	11
1.1.2.2.4.1. PHP 확장 모듈.....	11
1.1.2.2.4.2. PHP 모니터링 연결 확인.....	13
1.1.2.2.5. 구글 앱 엔진 (Goole Cloud App Engine) 설치.....	13
1.1.2.2.5.1. app.yaml 파일 설정.....	13
1.1.2.2.5.2. Dockerfile 생성 및 설정.....	13
1.1.2.2.5.2.1. 방화벽 설정.....	14
1.1.2.2.5.2.2. 배포.....	15
1.1.2.2.5.2.3. 구글 앱 엔진 인스턴스 확인.....	15
1.1.2.3. 실행.....	16
1.1.2.3.1. 애플리케이션 재기동.....	16
1.1.2.3.2. whatap-php 서비스(Service) 재시작.....	16
1.1.2.4. 업데이트.....	16
1.1.2.4.1. Redhat/CentOS.....	17
1.1.2.4.2. Debian/Ubuntu.....	17

목차

1.1.2.4.3. FreeBSD.....	17
1.1.2.5. 중지.....	18
1.1.2.5.1. 일시 중지.....	18
1.1.2.5.1.1. PHP 확장 모듈(PHP Extension module) 중지.....	18
1.1.2.5.1.2. whatap-php 서비스(Service) 중지.....	18
1.1.2.5.2. 에이전트 삭제.....	18
1.1.2.5.2.1. PHP 확장 모듈(PHP Extension module) 및 whatap-php 서비스(Service) 삭제.....	18
1.1.2.5.2.2. 패키지 삭제.....	18
1.1.2.5.2.2.1. /usr/whatap/php 디렉토리 삭제.....	19
1.1.3. 로그.....	20
1.1.3.1. 로그파일 종류.....	20
1.1.3.2. 정상 동작 확인.....	20
1.1.4. 설치 에러 대응.....	21
1.1.4.1. PHP 확장 모듈(PHP Extension module) 및 whatap-php 서비스(Service) 수동 설정.....	21
1.1.4.1.1. whatap.ini 생성.....	21
1.1.4.1.2. PHP 명령어(CLI) 경로 확인.....	22
1.1.4.1.2.1. whatap-php 서비스(Service) 환경 변수 설정.....	22
1.1.4.1.3. PHP API 버전 확인.....	22
1.1.4.1.3.1. PHP ZTS(Zend Thread Safe) 지원 여부 확인.....	22
1.1.4.1.3.1.1. Apache.....	22
1.1.4.1.3.1.2. PHP-FPM.....	22
1.1.4.1.4. PHP 확장 모듈(PHP Extension module) 경로 확인 및 설정.....	22
1.1.4.1.4.1. PHP 확장 모듈(PHP Extension module) 경로 확인.....	22
1.1.4.1.4.2. PHP 확장 모듈(PHP Extension module) 설정.....	23
1.1.4.1.4.3. whatap-php 서비스(Service) 환경 변수 설정.....	23
1.1.4.1.5. whatap.ini 설정.....	23
1.1.4.1.5.1. PHP 추가 ini 설정 경로 확인.....	23
1.1.4.1.5.2. PHP 추가 ini 설정 경로 확인 불가.....	23
1.1.4.1.5.3. whatap-php 서비스(Service) 환경 변수 설정.....	24
1.1.4.1.5.4. 서비스 재시작.....	24
1.1.4.2. Error: Not found PHP API.....	24
1.1.4.3. Error: Not found PHP ini directory.....	24
1.1.4.4. 응답 시간 분포도 (히트맵)에 트랜잭션이 표시되지 않는 경우.....	25
1.1.4.4.1. PHP 확장 모듈(PHP Extension module) 확인.....	25
1.1.5. FAQ.....	25
1.1.6. 제약 사항.....	25
1.1.7. 호환성 목록.....	25
1.1.7.1. 운영체제.....	25
1.1.7.2. PHP.....	25

Agent install Guide

1.1. PHP 애플리케이션 모니터링

1.1.1. 에이전트 실행 및 모니터링 개요

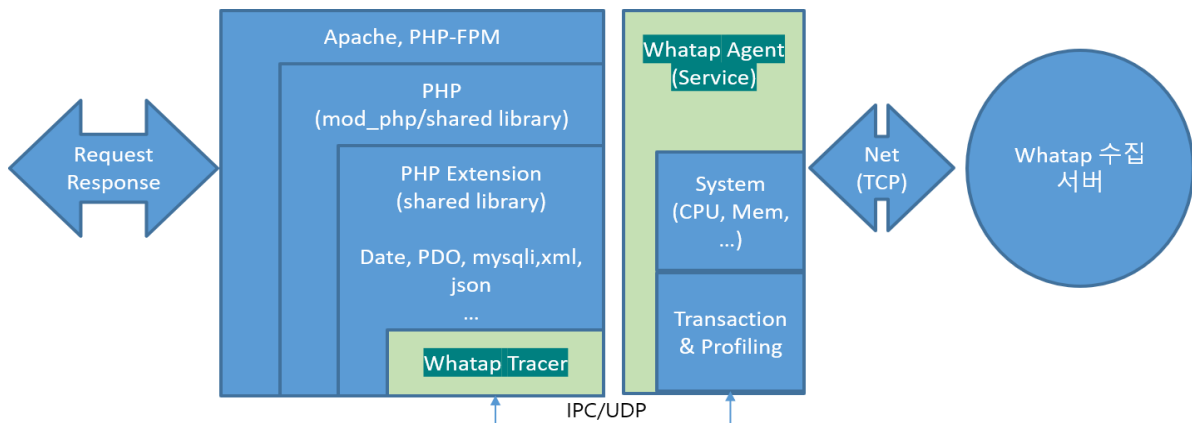
와탭 PHP 애플리케이션 모니터링은 PHP 기반 웹 애플리케이션 서버 모니터링 서비스를 제공합니다.

1.1.1.1. 에이전트 설치 방식 개요

PHP 모니터링 서비스를 사용하기 위해서는 모니터링 대상 애플리케이션에 모니터링 에이전트를 설치해야 합니다. 설치 방식은 리눅스 패키지 설치로 가능합니다.

- 와탭 저장소(Repository)를 설치합니다.
- whatap-php 리눅스 패키지를(yum, apt-get) 설치합니다.
- 설정 스크립트를 실행합니다.
- Apache 또는 PHP-FRPM 을 재시작 합니다.

설정 스크립트를 통해서 트레이서는 PHP 확장 모듈(PHP Extension module)로 등록되고, 에이전트는 whatap-php 서비스(Service)로 실행됩니다.



Agent install Guide

1.1.1.2. 구성 파일

모니터링 정보를 수집하기 위한 트레이서, 수집된 정보를 서버에 전송하기 위한 에이전트, 트레이서와 에이전트를 서버에 동적으로 적용하기 위한 설치 스크립트 파일로 구성됩니다.

PHP 모니터링 서비스를 구성하는 각 파일의 역할은 다음과 같습니다.

파일명	설명
whatap_#_#.so	트레이서 PHP 확장 모듈(PHP Extension module)로 추가 되어 정보를 수집하고 수집된 정보를 에이전트로 전송하는 라이브러리입니다.
whatap-php (whatap_php)	에이전트, 트레이서에서 UDP 로 전달된 정보를 수집서버로 전송하는 프로그램입니다. /etc/init.d/whatap-php 서비스 스크립트 whatap_php 서비스 실행파일
whatap.ini	애플리케이션 서버의 데이터를 수집하는 PHP 확장 모듈(PHP Extension module)의 설정 정보, 수집서버의 주소와 서버의 프로젝트 라이선스 키가 입력되는 파일입니다.
whatap-install-#.log	설치 과정에 대한 로그 파일입니다. (/usr/whata/php/logs)
whatap-boot-#.log	에이전트 로그 파일 입니다. (/usr/whata/php/logs)
install.sh	웹 애플리케이션 서버에 트레이서를 적용하기 위한 쉘 스크립트입니다.
Whatap.php (sample.php)	PHP 소스코드에서 사용할 API 레퍼런스 클래스 (/usr/whatap/php/lib/Whatap.php) 및 예제 소스 파일(sample.php) 입니다.

1.1.1.3. 에이전트 이름 식별

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}-{docker}

Agent install Guide

기본 패턴에 대한 변경은 whatap.ini 에서 설정에서 가능합니다.

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

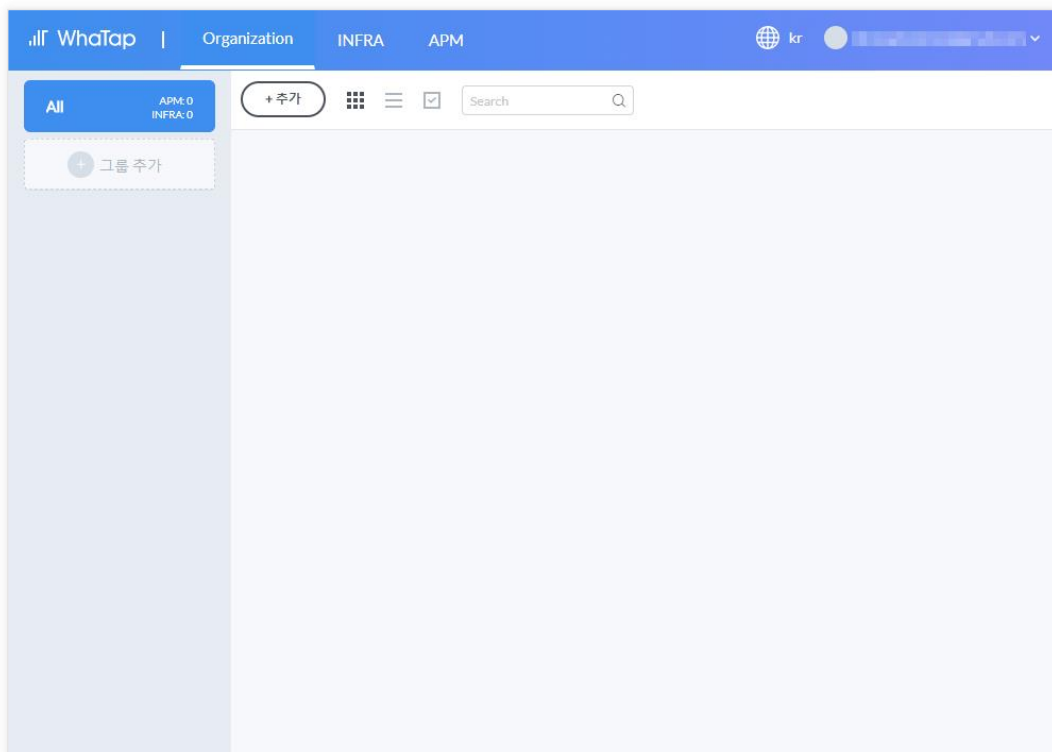
설정	설명
Type	app_name
Ip#	Ip 를 .으로 나누었을 때 #번째 자리(0 부터)
Process	app_process_name
hostname	호스트 명
docker	Docker Container ID

1.1.2. 에이전트 설치/실행/업데이트/중지

1.1.2.1. 공통

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.





이후, 생성된 프로젝트를 클릭하여 관리 화면에 진입합니다.


Agent install Guide


프로젝트 생성
✕


 INFRA
STRUCTURE


 Java


 node


 php


 python

프로젝트 명

데이터 서버 지역

프로젝트 그룹

취소
전송

1.1.2.1.2. 라이선스 발급

**Configuration**

다음과 같이 PHP확장 모듈에서 whatap server host와 license key를 설정 할 수 있습니다.

라이선스 발급받기

 License Key: XXXXXXXXXXXXXXXXXXXX-XXXXXXXXXXXXXXXXXX-XXXXXXXXXXXXXXXXXX

```
sudo /usr/whatap/php/install.sh
Input license key <= [발급된 라이선스 key 입력]
XXXXXXXXXXXXXXXXXX-XXXXXXXXXXXXXXXXXX-XXXXXXXXXXXXXXXXXX

Input whatap.server.host <= [발급된 Server Ip 입력]
52.78.209.94/52.78.224.235
```

Whatap Server (52.78.209.94/52.78.224.235)로 데이터 전송을 위해 6600 포트가 열려 있어야 합니다. (Tcp Outbound)

또한, 내부적으로 Udp통신을 위해 6600 포트가 필요합니다. 포트 충돌이 나는 경우 net_udp_port=xxx 옵션으로 포트를 변경 할 수 있습니다.

프로젝트 관리화면에서는 우선적으로 라이선스를 발급 받습니다. 라이선스 키는 프로젝트별로 귀속되기 때문에, 유출되거나 배포되어서는 안됩니다. 반드시 본인 프로젝트에 서버를 등록할 때에만 이용하시기 바랍니다.

Agent install Guide

1.1.2.2. 에이전트 설치

1.1.2.2.1. 패키지 설치

1.1.2.2.1.1. RedHat/CentOS

1.1.2.2.1.1.1. 패키지 저장소(Repository) 등록

와탭 저장소(Repository)를 등록합니다.

```
$ sudo rpm -Uvh http://repo.whatap.io/centos/5/noarch/whatap-repo-1.0-1.noarch.rpm
```

1.1.2.2.1.1.2. 패키지 설치

아래 명령어를 통해 패키지를 설치합니다.

```
$ sudo yum install whatap-php
```

1.1.2.2.1.2. Debian/Ubuntu

1.1.2.2.1.2.1. 패키지 저장소(Repository) 등록

와탭 저장소(Repository)를 등록합니다.

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

1.1.2.2.1.2.2. 패키지 설치

```
$ sudo apt-get install whatap-php
```

1.1.2.2.1.1. Amazon Linux

1.1.2.2.1.1.1. 패키지 저장소(Repository) 등록

와탭 저장소(Repository)를 등록합니다.

```
$ sudo rpm --import http://repo.whatap.io/centos/release.gpg  
$ echo "[whatap]" | sudo tee /etc/yum.repos.d/whatap.repo > /dev/null  
$ echo "name=whatap packages for enterprise linux" | sudo tee -a /etc/yum.repos.d/whatap.repo >  
/dev/null  
$ echo "baseurl=http://repo.whatap.io/centos/latest/\$basearch" | sudo tee -a  
/etc/yum.repos.d/whatap.repo > /dev/null
```


Agent install Guide

```
$ echo "enabled=1" | sudo tee -a /etc/yum.repos.d/whatap.repo > /dev/null
$ echo "gpgcheck=0" | sudo tee -a /etc/yum.repos.d/whatap.repo > /dev/null
```

1.1.2.2.1.1.2. 패키지 설치

```
$ sudo yum install whatap-php
```

1.1.2.2.1.1. FreeBSD

1.1.2.2.1.1.1. 패키지 설치

```
$ wget https://s3.ap-northeast-2.amazonaws.com/repo.whatap.io/freebsd/10/whatap-php-0.4.4.txz
$ pkg install whatap-php-0.4.4.txz
```

1.1.2.2.1.1.2. 시스템 UDP 관련 설정

UDP Message 크기를 조정합니다.

```
$ sysctl -w net.inet.udp.maxdgram=65536
net.inet.udp.maxdgram: 9216 -> 65536
$ sysctl -w net.inet.udp.recvspace=131072
net.inet.udp.recvspace: 42080 -> 131072
```

- net.inet.udp.maxdgram: Maximum outgoing UDP datagram size
- net.inet.udp.recvspace : Increase UDP "receive" buffer size. Good for 200Mbit without packet drop

영구적으로 설정하기 위해 /etc/sysctl.conf 에도 설정합니다.

```
$ vi /etc/sysctl.conf

net.inet.udp.maxdgram=65536
net.inet.udp.recvspace=131072
```

1.1.2.2.2. PHP 확장 모듈(PHP Extension module) 및 whatap-php 서비스(Service) 등록

PHP 확장 모듈(PHP Extension module) 및 whatap-php 서비스(Service)를 자동으로 설치할 경우에 아래와 같이 적용합니다

```
$ sudo /usr/whatap/php/install.sh
Input license key
xxxxxxxxxxxxxxxxx          # 발급된 라이선스 key 입력

Input whatap.server.host
192.x.x.x                  # 발급된 서버 IP 입력
```

Agent install Guide

PHP 확장 모듈(PHP Extension module) 및 whatap-php 서비스(Service)를 자동으로 인식하지 못하는 경우 아래와 같이 선택 설치를 진행해야 합니다.

주로 Apache 명령어(apachectl, httpd, apache2) 및 PHP 명령어(CLI)가 기본 경로(\$PATH)에 설정 되어 있지 않거나, 여러 개의 PHP가 설치되어 PHP 명령어(CLI)가 여러 개일 경우에 (php5, php70, php-zts, zts-php...) 실제로 적용하고 있는 버전을 선택하여 진행합니다.

```
$ sudo /usr/whatap/php/install.sh manual

Input license key
xxxxxxxxxxxxxxxxx          # 발급된 라이선스 key 입력

Input whatap.server.host
192.x.x.x                  # 발급된 서버 IP 입력

Input : which apache or php-fpm ex) /usr/sbin/httpd, /usr/sbin/apache2, /usr/sbin/php-fpm ...
/usr/sbin/httpd           # apache 및 php-fpm 명령어 위치 입력

Input : which php ex) /usr/bin/php, /usr/bin/php5, /usr/bin/php70 ...
/usr/bin/php5             # php 명령어 위치 입력
```

1.1.2.2.1. 설정 스크립트 install.sh

```
Usage: install.sh
install.sh [ commands manual|remove ]
install.sh -l <license> -s <server> -v <php version> -i <php config file> -p <process_name> [-z ]
install.sh -l <license> -s <server> -e <php command> -p <process_name> [-z ]

Commands
auto(default)      Auto scan web server and php environment
manual             User input web server and php environment
remove            Remove php extension and service(whatap-php)

Options
-v <php version>  PHP version ex) 5.3, 5.4, 5.5, 5.6, 7.0, 7.1, 7.2
                  -v 5.3, --php-version 5.3
-i <php config file> PHP config file
                  -i /etc/php.ini, --php-ini /etc/php.ini
-e <php cli>       PHP CLI(command line interface)
                  -e /usr/bin/php , --php-exe: /usr/bin/php-fpm
-p <process name>  Web server or PHP-FPM process name. for get used memory
                  -p httpd, -p httpd.worker, -process-name: php-fpm
-l <license>       License key
                  -l xxxxxxxxxxxxxxxx, --license: xxxxxxxxxxxxxxxx
-s <server>        Whatap Server Host
                  -s xx.xx.xx.yy.yy.yy, -server xxx.xxx.xxx.xxx
-z                PHP thread safe
                  -z , --php-zts
```

■ license

Agent install Guide

- 발급된 라이선스 key
 - server
 - 발급된 서버 IP
 - php version
 - 서버에 설치된 php 버전 정보 (5.2, 5.3, 5.4, 5.5, 5.6, 7.0, 7.1, 7.2)
 - php config file
 - 서버에 설치된 php.ini 의 전체 경로 (/etc/php.ini)
 - PHP Extension 에 대한 설정을 진행합니다.
 - php cli
 - 커맨드 라인으로 실행가능한 php 명령어의 전체 경로 (/usr/bin/php)
 - 해당 명령어로 PHP 기본 환경을 확인합니다.
 - process name
 - Apache 실행 프로세스명(httpd, apache2, httpd.worker ...) 또는 PHP-FPM 의 실행 프로세스명(PHP-FPM) (php-fpm, php5-fpm)
 - 해당 프로세스 명으로 사용 메모리 정보를 수집합니다.
 - zts
 - Zend Thread Safe 지원여부를 설정합니다.

1.1.2.2.2.1.1. PHP 버전 지정 설치

```
$ /usr/whatap/php/install.sh -l [발급된 라이선스 key] -s [발급된 서버 IP] -v [PHP 버전 x.x] -i [php.ini
파의 전체 경로] -p [프로세스명]
```

실행 예

```
$ /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]
```

1.1.2.2.2.1.2. PHP Cli 명령어 지정 설치

```
$ /usr/whatap/php/install.sh -l [발급된 라이선스 key] -s [발급된 서버 IP] -e [PHP Cli] -p [프로세스명]
```

실행 예

```
$ /usr/whatap/php/install.sh -l xxxxx -s 1.1.1.1/2.2.2.2 -e /opt/php/bin/php -p apache2
```

```
$ /usr/whatap/php/install.sh -l xxxxx -s 1.1.1.1/2.2.2.2 -e /opt/php/bin/php-fpm -p php-fpm
```

1.1.2.2.3. 애플리케이션 재기동

설치가 완료된 후 Apache 또는 PHP-FPM 서비스(Service)를 재시작 하면 설정된 PHP 확장 모듈(PHP Extension module) whatap.so 파일이 로딩됩니다.

1.1.2.2.4. 설치 확인

1.1.2.2.4.1. PHP 확장 모듈

Agent install Guide

PHP 추가 INI 경로 또는 ./usr/whatap/php 경로에 whatap.ini 생성되어 있는지 확인합니다.

```
$ find / | grep whatap.ini
```

PHP 확장 모듈(PHP Extension module) 경로에 whatap.so 파일 생성되어 있는지 확인합니다.
또는 php.ini 하단에 [whatap] 설정이 있는 지 확인합니다.

```
$ find / | grep whatap.so
```

```
$ vi php.ini
```

```
...
```

```
[whatap]  
extension=whatap.so
```

```
...
```

PHP 확장 모듈(PHP Extension module) 실행되고 있는지 확인합니다.

```
$ sudo php -m
```

```
[PHP Modules]
```

```
bz2
```

```
calendar
```

```
Core
```

```
ctype
```

```
curl
```

```
date
```

```
...
```

```
whatap          # whatap 모듈 실행 확인
```

```
...
```

```
[Zend Modules]
```

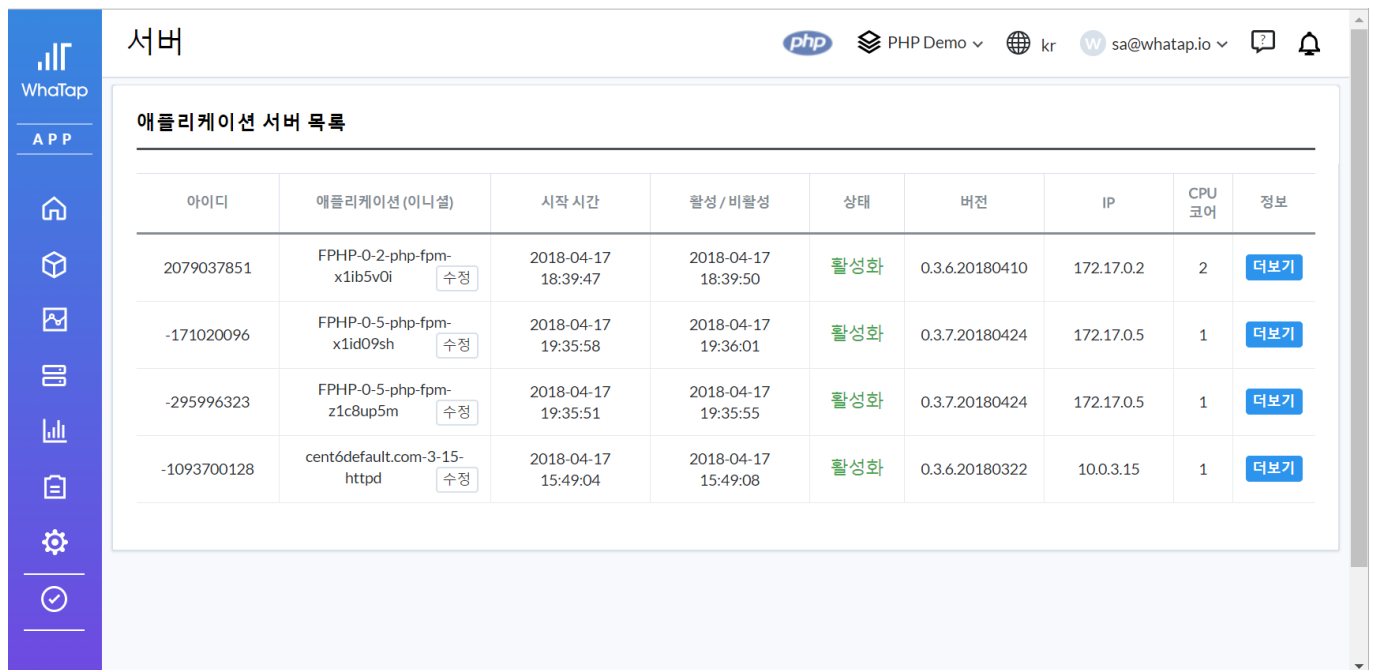
whatap-php 서비스(Service) 상태를 확인합니다.

```
$ service whatap-php status
```

Agent install Guide

1.1.2.2.4.2. PHP 모니터링 연결 확인

애플리케이션 서버 목록에서 설정된 서버가 추가 되었는지 확인합니다.



아이디	애플리케이션 (이니셜)	시작 시간	활성 / 비활성	상태	버전	IP	CPU 코어	정보
2079037851	FPHP-0-2-php-fpm-x1ib5v0i <small>수정</small>	2018-04-17 18:39:47	2018-04-17 18:39:50	활성화	0.3.6.20180410	172.17.0.2	2	<small>더보기</small>
-171020096	FPHP-0-5-php-fpm-x1id09sh <small>수정</small>	2018-04-17 19:35:58	2018-04-17 19:36:01	활성화	0.3.7.20180424	172.17.0.5	1	<small>더보기</small>
-295996323	FPHP-0-5-php-fpm-z1c8up5m <small>수정</small>	2018-04-17 19:35:51	2018-04-17 19:35:55	활성화	0.3.7.20180424	172.17.0.5	1	<small>더보기</small>
-1093700128	centódefault.com-3-15-httpd <small>수정</small>	2018-04-17 15:49:04	2018-04-17 15:49:08	활성화	0.3.6.20180322	10.0.3.15	1	<small>더보기</small>

1.1.2.2.5. 구글 앱 엔진 (Goole Cloud App Engine) 설치

구글 앱 엔진에 PHP 모니터링을 설치하기 위해서는 custom docker 를 사용해야 합니다.

구글 앱 엔진 PHP 의 기본형은 PHP 모니터링의 PHP Extension 및 Service (데몬)의 실행이 불가능하여 Docker Container 를 통하여 설치합니다.

1.1.2.2.5.1. app.yaml 파일 설정

구글 앱 엔진의 설정파일인 app.yaml 파일에 flex 환경, custom 런타임으로 설정합니다.

```
$ vi app.yaml

env: flex
runtime: custom
```

1.1.2.2.5.2. Dockerfile 생성 및 설정

Custom Docker 로 실행하기 위해서는 flex/custom 으로 설정된 app.yaml 파일과 동일한 위치에 Dockerfile 을 생성해야 합니다.

구글 앱 엔진의 기본 php Docker 이미지를 참조합니다.

```
# google-appengine php php56, php70, php71
FROM gcr.io/google-appengine/php56
```

PHP 모니터링 Agent 를 설치합니다.

Agent install Guide

- 라이브러리 설치합니다. (wget)
- php-fpm 실행 명령어 링크를 생성합니다.
- whatap 레포지터리 및 whatap-php 패키지를 설치합니다.
- PHP Extension 및 whatap-php 서비스를 설정합니다.
- supervisor 를 통해 whatap-php 서비스를 실행하도록 설정합니다.

```
# WhaTap. Install Lib
RUN apt-get install wget -y

# WhaTap. Link php-fpm
RUN ln -s /opt/php/sbin/php-fpm /usr/bin/php-fpm

# WhaTap. Install repository and whatap-php
RUN wget http://repo.whatap.io/debian/release.gpg -O - | apt-key add -
RUN wget http://repo.whatap.io/debian/whatap-repo_1.0_all.deb
RUN dpkg -i whatap-repo_1.0_all.deb
RUN apt-get update
RUN apt-get install whatap-php

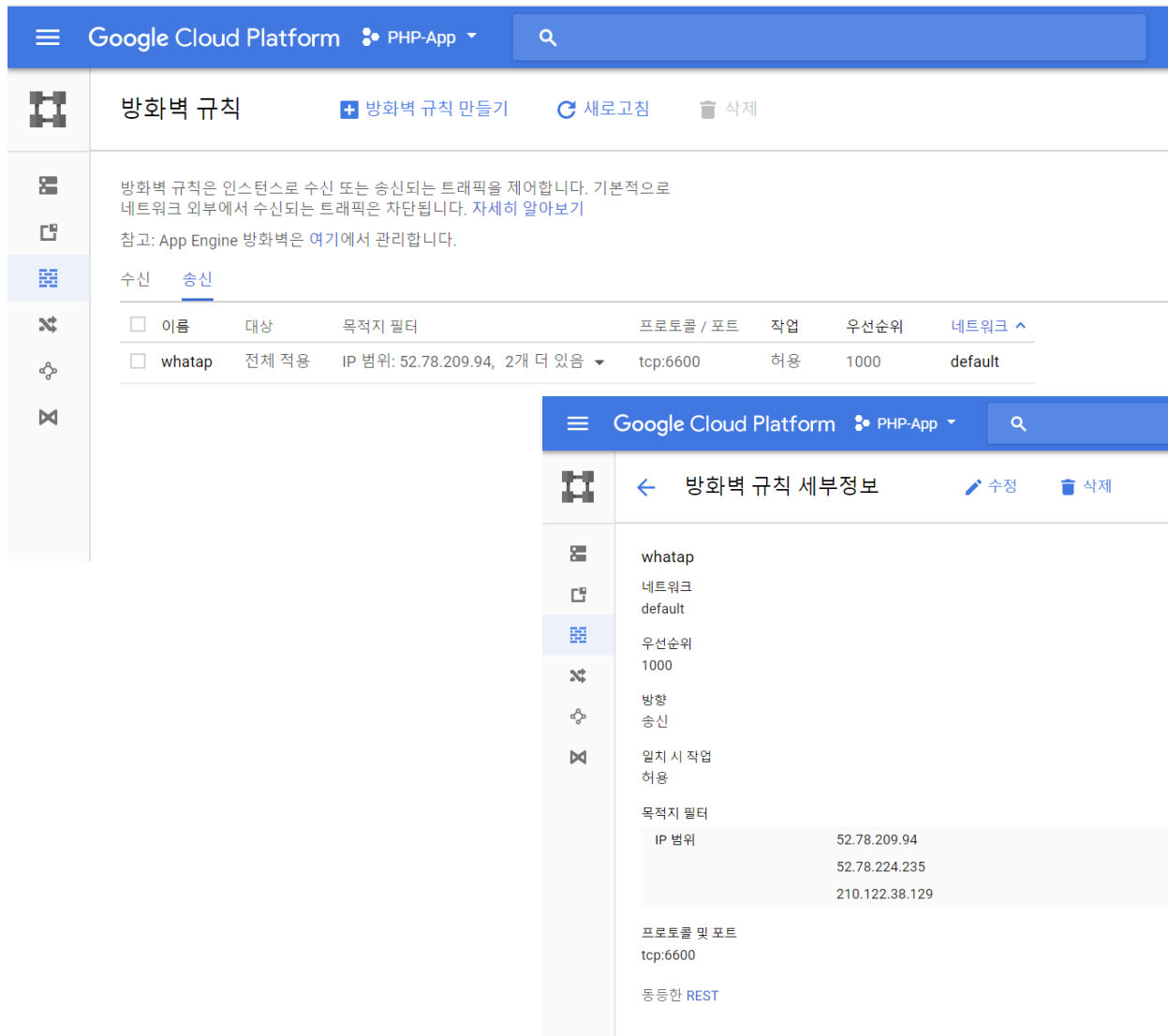
# WhaTap. Configure PHP agent
#RUN (echo "[ 발급된 라이선스 key ]"; echo "[ 발급된 서버 IP ]")|/usr/whatap/php/install.sh
RUN (echo "xxxxxxx"; echo "1.1.1.1/2.2.2.2")|/usr/whatap/php/install.sh

# WhaTap. Add supervisor/conf.d/whatap.conf. Run whatap-php service
RUN echo "[program:whatap-php]" >> /etc/supervisor/conf.d/whatap.conf
RUN echo "command = /etc/init.d/whatap-php start" >> /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
RUN echo "user = root" >> /etc/supervisor/conf.d/whatap.conf
RUN echo "autostart = true" >> /etc/supervisor/conf.d/whatap.conf
RUN echo "autorestart = true" >> /etc/supervisor/conf.d/whatap.conf
RUN echo "priority = 10" >> /etc/supervisor/conf.d/whatap.conf
```

1.1.2.2.5.1. 방화벽 설정

PHP 모니터링 프로젝트 생성 후 라이선스 키와 서버 IP(/ 구분자로 여러 개의 IP)가 발급됩니다.
구글 앱 엔진에서 사용하는 네트워크를 대상으로 방화벽 규칙을 추가합니다.
발급받은 서버 IP 에 대해 6600 포트를 통한 TCP 송신을 허용해야 합니다.

Agent install Guide



방화벽 규칙 + 방화벽 규칙 만들기 새로고침 삭제

방화벽 규칙은 인스턴스로 수신 또는 송신되는 트래픽을 제어합니다. 기본적으로 네트워크 외부에서 수신되는 트래픽은 차단됩니다. 자세히 알아보기

참고: App Engine 방화벽은 [여기에서](#) 관리합니다.

수신 송신

이름	대상	목적지 필터	프로토콜 / 포트	작업	우선순위	네트워크
whatap	전체 적용	IP 범위: 52.78.209.94, 2개 더 있음	tcp:6600	허용	1000	default

방화벽 규칙 세부정보 수정 삭제

whatap

네트워크 default

우선순위 1000

방향 송신

일지 시 작업 허용

목적지 필터

IP 범위	
52.78.209.94	
52.78.224.235	
210.122.38.129	

프로토콜 및 포트 tcp:6600

동등한 REST

1.1.2.2.5.2. 배포

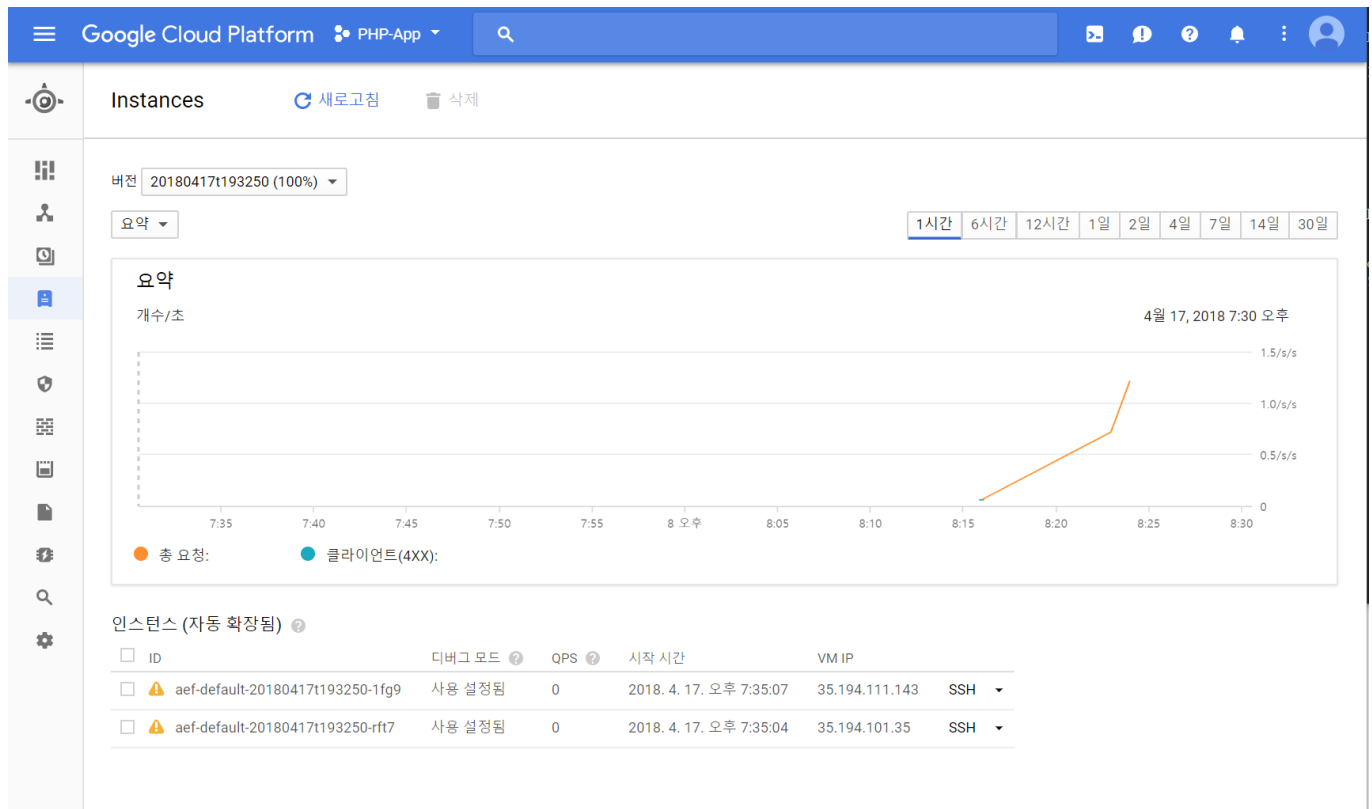
Google Cloud SDK 를 통해서 설정된 app.yaml 파일과 Dockerfile 을 배포합니다.

```
$ gcloud app deploy
```

1.1.2.2.5.1. 구글 앱 엔진 인스턴스 확인

구글 클라우드 콘솔에서 앱 엔진의 생성된 인스턴스를 확인합니다.

Agent install Guide



1.1.2.3. 실행

1.1.2.3.1. 애플리케이션 재기동

설치가 완료된 후 Apache 또는 PHP-FPM 서비스(Service)를 재시작 하면 설정된 PHP 확장 모듈(PHP Extension module) whatap.so 파일이 로딩됩니다.

1.1.2.3.2. whatap-php 서비스(Service) 재시작

whatap-php 서비스(Service)가 실행 중이지 않거나 오류가 발생한 경우 재시작 합니다.

■ Redhat/CentOS

```
$ service whatap-php restart
```

■ Debian/Ubuntu

```
$ sudo service whatap-php restart
```

■ FreeBSD

```
$ service whatap_php restart
```

1.1.2.4. 업데이트

패키지 업데이트는 기존 설정을 유지한 채로 PHP 모니터링 서비스를 업데이트합니다.

0.2.7 이후 버전부터 정상적인 업데이트가 지원됩니다. 이전 버전은 삭제 후 설치를 진행해야 합니다.

Agent install Guide

1.1.2.4.1. Redhat/CentOS

패키지 정보 갱신을 위해 캐시 정보를 삭제합니다.

```
$ yum clean all
```

Apache 또는 PHP-FPM 서비스(Service)를 중지합니다.

whatap-php 패키지를 업데이트 합니다.

```
$ yum update whatap-php
```

Apache 또는 PHP-FPM 서비스(Service)를 시작합니다

1.1.2.4.2. Debian/Ubuntu

패키지 정보 갱신을 위해 캐시 정보를 갱신합니다.

```
$ sudo apt-get update
```

Apache 또는 PHP-FPM 서비스(Service)를 중지합니다.

whatap-php 패키지를 업데이트 합니다.

```
$ sudo apt-get install --only-upgrade whatap-php
```

Apache 또는 PHP-FPM 서비스(Service)를 시작합니다

1.1.2.4.3. FreeBSD

패키지 삭제 후 재설치를 진행합니다.

아래 항목을 참조하여 순서대로 진행합니다.

Apache 또는 PHP-FPM 서비스(Service)를 중지한 상태에서 업데이트를 진행합니다.

1.1.2.5.2. 에이전트 삭제

1.1.2.2.1.1 FreeBSD

1.1.2.2.2. PHP 확장 모듈(PHP Extension module) 및 whatap-php 서비스(Service) 등록

Apache 또는 PHP-FPM 서비스(Service)를 시작합니다

Agent install Guide

1.1.2.5. 중지

1.1.2.5.1. 일시 중지

1.1.2.5.1.1. PHP 확장 모듈(PHP Extension module) 중지

whatap.ini 파일의 'extension=' 구문을 주석처리 합니다.

수동 설정으로 php.ini 에 직접 설정한 경우도 동일하게 'extension=' 구문을 주석처리 합니다. 주석은 세미콜론(;) 으로 설정합니다.

```
$ sudo vi whatap.ini

extension=whatap.so

;주석
;extension=whatap.so
```

1.1.2.5.1.2. whatap-php 서비스(Service) 중지

■ Redhat/CentOS

```
$ service whatap-php stop
```

■ Debian/Ubuntu

```
$ sudo service whatap-php stop
```

■ FreeBSD

```
$ service whatap_php stop
```

1.1.2.5.2. 에이전트 삭제

설치 스크립트를 이용하여 트레이서(PHP Extension module) 및 whatap-php 서비스(Service)를 삭제합니다.

이후 패키지(yum, apt-get) 삭제를 진행하고 필요에 따라서 /usr/whatap/php 아래에 로그 파일 및 기타 파일을 삭제합니다.

1.1.2.5.2.1. PHP 확장 모듈(PHP Extension module) 및 whatap-php 서비스(Service) 삭제

```
$ /usr/whatap/php/install.sh remove
```

1.1.2.5.2.2. 패키지 삭제

■ Redhat/CentOS

```
$ sudo yum remove whatap-php
```

■ Debian/Ubuntu

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

■ FreeBSD

Agent install Guide

```
$ pkg delete whatap-php
```

1.1.2.5.2.2.1. /usr/whatap/php 디렉토리 삭제

Agent install Guide

1.1.3. 로그

1.1.3.1. 로그파일 종류

로그파일은 2 가지로 종류는 다음과 같습니다.

- whatap-boot-[DATE].log : 데이터 수집 로그
- whatap-install-[DATE].log : install.sh 실행 로그

1.1.3.2. 정상 동작 확인

애플리케이션 서버를 기동 또는 재기동 한 뒤, 애플리케이션 서버 로그 및 에이전트 로그를 확인하여 PHP 모니터링 서비스의 정상 기동 여부를 확인합니다.

파일. apache/error_log

```
[Fri Nov 24 11:13:03 2017] [notice] Apache/2.2.15 (Unix) DAV/2 PHP/5.4.45 configured -- resuming normal operations
[Fri Nov 24 15:24:09 2017] [notice] caught SIGTERM, shutting down
[Fri Nov 24 15:24:09 2017] [notice] suEXEC mechanism enabled (wrapper: /usr/sbin/suexec)
[Fri Nov 24 15:24:09 2017] [notice] Digest: generating secret for digest authentication ...
[Fri Nov 24 15:24:09 2017] [notice] Digest: done
WA001 Whatap APM started
```

파일. /usr/whatap/php/logs/whatap-boot-xxx.log

```
2017/11/27 15:49:15 [WA214] Config: /etc/php.d/whatap.ini
2017/11/27 15:49:15
2017/11/27 15:49:15 ## OPEN LOG FILE boot 0.3.0.20170929 20171127 06:49:15.44 ##
2017/11/27 15:49:15
2017/11/27 15:49:15 [WA827] 0xc4204ba460
2017/11/27 15:49:15 [WA214] Config: /etc/php.d/whatap.ini
2017/11/27 15:49:15 [WA171] PCODE=12345693990ID=-11821618710NAME=d54FPHP-3-15-httpd
2017/11/27 15:49:15 [WA174] Net TCP: Connect to121.134.69.246:6600
2017/11/27 15:49:16 [WA632] Net UDP: Try to.... localhost:6600
2017/11/27 15:49:16 [WA635] Net UDP: Listener localhost:6600
```

Agent install Guide

1.1.4. 설치 에러 대응

1.1.4.1. PHP 확장 모듈(PHP Extension module) 및 whatap-php 서비스(Service) 수동 설정

PHP 확장 모듈(PHP Extension module), whatap-php 서비스(Service) 설치 및 선택 설치(install.sh)가 정상적으로 이루어 지지 않을 경우 수동으로 설정하는 방법을 설명합니다.

PHP 컴파일(Compile) 설치 등의 이유로 환경 정보를 확인 할 수 없는 경우 사용합니다.

1.1.4.1.1. whatap.ini 생성

```
$ cp /usr/whatap/php/template.ini /usr/whatap/php/whatap.ini
$ vi /usr/whatap/php/whatap.ini

# 상단에 내용 추가
; Enable whatap extension module
extension=whatap.so
whatap.license=          # 발급된 라이선스 key
whatap.server.host=     # 발급된 서버 IP
whatap.app_name=        # 웹서버 구분 APHP, FPHP (apache : APHP, php-fpm : FPHP)
whatap.app_process_name= # apache, php-fpm 의 프로세스 이름(httpd,php-fpm)
```

설정	설명
whatap.license	프로젝트 > 관리 > 에이전트 설치 페이지에서 발급된 라이선스 키를 확인할 수 있습니다.
whatap.server.host	프로젝트 > 관리 > 에이전트 설치 페이지에서 발급된 서버 IP 를 확인할 수 있습니다.
whatap.app_name	Apache 서버는 'APHP', php-fpm 은 'FPHP'를 사용합니다.
whatap.app_process_name	Apache 또는 php-fpm 의 실행 프로세스 이름 설정으로 정확한 프로세스명 입력하면, 해당 프로세스에 대한 사용 메모리를 수집합니다. 예) httpd, apach2, php-fpm, php-fpm 등.

Agent install Guide

1.1.4.1.2. PHP 명령어(CLI) 경로 확인

```
$ which php
/usr/bin/php
```

1.1.4.1.2.1. whatap-php 서비스(Service) 환경 변수 설정

\$WHATAP_PHP_BIN 환경 변수에 PHP CLI 명령어의 경로를 설정합니다.

```
$ sudo vi /etc/init.d/whatap-php
export WHATAP_PHP_BIN=          # PHP 명령어 위치(/usr/bin/php)
```

1.1.4.1.3. PHP API 버전 확인

\$WHATAP_PHP_BIN 환경 변수에 PHP CLI 명령어의 경로를 설정합니다.

```
$ sudo php -i | grep 'PHP API'
PHP API => 20100412
```

1.1.4.1.3.1. PHP ZTS(Zend Thread Safe) 지원 여부 확인

1.1.4.1.3.1.1. Apache

```
$ sudo apachectl -V | grep MPM
Server MPM: Prefork          # ZTS 지원 안함
Server MPM: Worker          # ZTS 지원
```

1.1.4.1.3.1.2. PHP-FPM

```
$ sudo php-fpm -i | grep Thread
Thread Safety => disabled    # ZTS 지원 안함
Thread Safety => enabled     # ZTS 지원
```

1.1.4.1.4. PHP 확장 모듈(PHP Extension module) 경로 확인 및 설정

1.1.4.1.4.1. PHP 확장 모듈(PHP Extension module) 경로 확인

```
$ sudo php -i | grep extension_dir
extension_dir => /usr/lib64/php/modules => /usr/lib64/php/modules
```

Agent install Guide

1.1.4.1.4.2. PHP 확장 모듈(PHP Extension module) 설정

PHP API 버전, PHP ZTS 지원 여부를 확인하여 환경에 적합한 라이브러리를 선택하여 PHP 확장 모듈(PHP Extension module) 경로에 whatap.so 파일명으로 복사합니다.

- PHP ZTS 를 지원할 경우 - whatap_zts_[PHP API 버전].so
- PHP ZTS 를 지원하지 않을 경우 - whatap_[PHP API 버전].so

```
$ sudo cp /usr/whatap/php/modules/x64/whatap_20100412.so /usr/lib64/php/modules/whatap.so
```

1.1.4.1.4.3. whatap-php 서비스(Service) 환경 변수 설정

\$WHATAP_PHP_EXT_HOME 환경변수에 PHP 확장 모듈 경로를 설정합니다.

\$WHATAP_PHP_EXT_SRC 환경변수에 와탭 라이브러리 전체 파일 경로를 설정합니다.

```
$ sudo vi /etc/init.d/whatap-php

export WHATAP_PHP_EXT_HOME= # PHP Extension 경로(/usr/lib64/php/modules)
export WHATAP_PHP_EXT_SRC= # 와탭 라이브러리 경로 및 파일명
                           # (/usr/whatap/php/modules/x64/whatap_20100412.so)]
```

1.1.4.1.5. whatap.ini 설정

1.1.4.1.5.1. PHP 추가 ini 설정 경로 확인

```
$ sudo php -i | grep '.ini files'

Scan this dir for additional .ini files => /etc/php.d
```

whatap.ini 를 해당 경로에 복사합니다.

```
$ sudo cp /usr/whatap/php/whatap.ini /etc/php.d/whtap.ini
```

1.1.4.1.5.2. PHP 추가 ini 설정 경로 확인 불가

PHP 컴파일(Compile) 설치 옵션 '--with-config-file-scan-dir=PATH'이 설정 안된 경우에 발생합니다.

```
$ sudo php -i | grep '.ini files'

Scan this dir for additional .ini files => (none)
```

whatap.ini 파일 내용을 php.ini 마지막에 추가합니다.

Agent install Guide

```
$ php -i | grep 'php.ini'

Loaded Configuration File => /etc/php.ini

$ sudo vi php.ini

# 파일 마지막에 추가
[whatap]
Enable whatap extension module
extension=whatap.so
whatap.ext.error_enabled=true
whatap.ext.exception_enabled=true
whatap.trace_user_enabled=true
whatap.trace_user_using_ip=false
```

이외 옵션은 /usr/whatap/php/whatap.ini 를 사용합니다.

1.1.4.1.5.3. whatap-php 서비스(Service) 환경 변수 설정

\$WHATAP_CONFIG_HOME 환경변수에 whatap.ini 경로를 설정합니다.

PHP 추가 ini 경로를 확인 할 수 없는 경우 whatap.ini를 생성한 /usr/whatap/php 경로를 설정합니다.

```
$ sudo vi /etc/init.d/whatap-php

export WHATAP_CONFIG_HOME= # whatap.ini 경로(/etc/php.d)
```

1.1.4.1.5.4. 서비스 재시작

Apache 및 PHP-FPM 서비스(Service)를 재시작 합니다.

whatap-php 서비스(Service)를 재시작 합니다.

1.1.4.2. Error: Not found PHP API

PHP 명령어(CLI)를 찾지 못하는 경우 발생합니다.

PHP 명령어(CLI)의 위치를 정확히 찾고, 'PHP 확장 모듈(PHP Extension module) 및 whatap-php 서비스(Service) 선택 설치' 항목을 진행합니다.

PHP API 버전 정보 확인

```
$ sudo php -i | grep 'PHP API'

PHP API => 20100412
```

1.1.4.3. Error: Not found PHP ini directory

PHP 환경 중 'Scan this dir for additional .ini files' 항목의 값을 확인 하지 못하는 경우 발생합니다. PHP 컴파일(Compile) 설치 옵션 '--with-config-file-scan-dir=PATH' 이 설정 안된 경우에 해당 환경정보가 없습니다.

PHP 명령어의(CLI) 경로를 정확히 찾고, 'PHP 확장 모듈(PHP Extension module) 및 whatap-php 서비스(Service) 수동 설정' 항목을 진행합니다.

Agent install Guide

```
$ sudo php -i | grep '.ini files'

Scan this dir for additional .ini files => (none)
```

1.1.4.4. 응답 시간 분포도 (히트맵)에 트랜잭션이 표시되지 않는 경우

CPU, Memory 의 그래프는 표기 되지만 응답 시간 분포도(히트맵)가 표기 되지 않는 현상은 에이전트가 수집서버와 정상적으로 연결 되었지만, 트레이서가 정상적으로 PHP 확장 모듈(PHP Extension module)로 적용되지 않았거나, 설정 후 Apache 및 PHP-FRPM 서비스(Service)를 재시작 하지 않은 경우에 발생합니다.

1.1.4.4.1. PHP 확장 모듈(PHP Extension module) 확인

```
$ sudo php -m

[PHP Modules]
bz2
calendar
Core
ctype
curl
date
...
whatap          # 와탭 모듈 로드 확인
...
[Zend Modules]
```

PHP 확장 모듈(PHP Extension module) 적용을 확인 한 경우 Apache 및 PHP-FRPM 서비스(Service)를 재시작 합니다. 적용이 안된 경우는 정상 설치 되지 않은 것으로 whatap.so 또는 whatap.ini 파일 경로가 PHP 환경과 일치하는 지 확인합니다. 'PHP 명령어(CLI) 경로 확인', 'PHP 확장 모듈(PHP Extension module) 경로 확인', 'PHP 추가 ini 설정 경로 확인' 을 확인합니다.

1.1.5. FAQ

1.1.6. 제약 사항

1.1.7. 호환성 목록

1.1.7.1. 운영체제

- Redhat/CentOS 64bit 6.x 이상
- Ubuntu 64bit 12.04 이상
- FreeBSD 64bit 10.x 이상

1.1.7.2. PHP

- 5.2, 5.3, 5.4, 5.5, 5.6, 7.0, 7.1, 7.2
- Non Tread Safe 방식 지원
- Tread Safe 방식 지원

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