Python Application
Monitoring
Agent install Guide

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

목차

목차

1.1. Python 애플리케이션 모니터링

1.1.1. 에이전트 실행 및 모니터링 개요

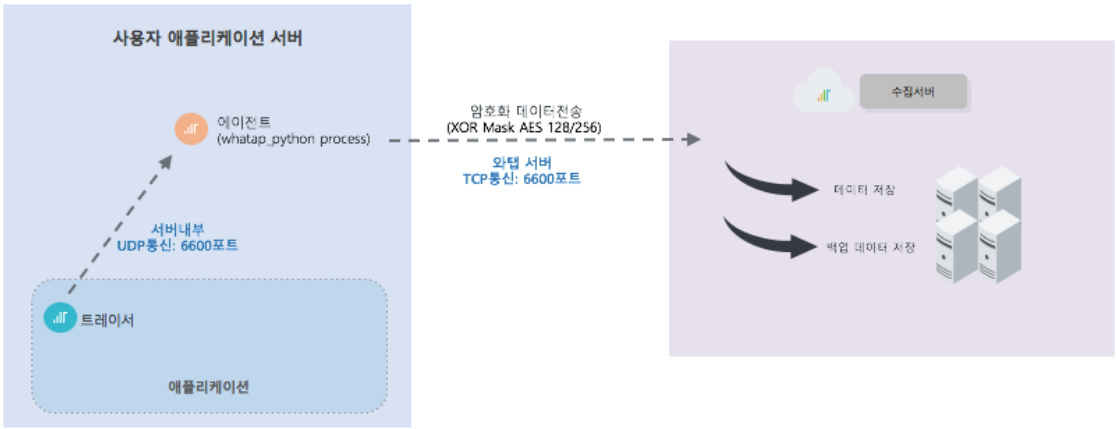
와탭 Python 애플리케이션 모니터링은 Python 기반 웹 애플리케이션 서버 모니터링 서비스를 제공합니다.

1.1.1.1. 에이전트 구성

에이전트의 구성은 수집 서버, 에이전트, 그리고 수집서버로 이루어집니다.

- 수집서버
  - 에이전트가 수집한 애플리케이션의 성능 데이터를 수집, 저장 및 통계 정보 추출하고 이를 사용자에 효율적인 방법으로 제공합니다. 수집서버는 지역(Region)별로 설정이 가능합니다. 지역(Region)별로 수집서버의 주소가 다르게 할당 되므로 사용자가 선택한 지역(Region)에 따라 수집서버 주소는 다를 수 있습니다. 지역(Region) 선택은 프로젝트 생성시에 함께 설정합니다.
- 에이전트
  - 애플리케이션 서버에 설치되어, 애플리케이션 성능 데이터를 수집하여 서버로 전송합니다.
- 트레이서
  - 애플리케이션 코드에서 프로파일링 데이터를 추적합니다.
- 네트워크
  - 와탭 모니터링 에이전트는 모니터링 정보를 수집하여 서버에 데이터 전송하기 위하여 외부통신(TCP)을 위한 6600 포트내부통신(UDP)을 위한

6600 포트를 사용합니다. 내부 포트가 충돌이 나는 경우,
net_udp_port=xxx 옵션으로 포트를 변경 할 수 있습니다.



1.1.1.2. 에이전트 구성 파일

에이전트 구성 파일의 종류는 다음과 같습니다.

파일명	설명
whatap.conf	에이전트 설정 파일
plugin.json	플러그인 옵션에서 참조하는 파일
paramkey.txt	보안키를 필요로 하는 옵션에서 참조하는 파일

1.1.1.2.1. 설정 파일

- 파일명: whatap.conf

에이전트 설정 기본 필수 파일입니다. 에이전트와 관련된 옵션은 모두 whatap.conf 에서 설정이 가능합니다.

1.1.1.2.2. 플러그인 파일

- 파일명: plugin.json

기본적으로 제공하는 트랜잭션 추적 모듈 이외에 사용자가 메뉴얼하게 모듈을 추가 할때 참고 하는 파일입니다. 형식에 맞게 내용을 추가하면 특정 모듈의 데이터를 추적할 수 있습니다. 옵션은 whatap.conf 에서 설정이 가능합니다. 관련 옵션은 다음과 같습니다.

- plugin default: false

1.1.1.2.3. 보안키 파일

- 파일명: paramkey.txt

추적한 트랜잭션의 프로파일정보로 수집한 HTTP 와 SQL 데이터의 파라미터 정보를 확인 하는데 사용합니다. 보안키를 파일에 저장하고 실제 수집된 데이터를 브라우저에서 확인 하고자 할 때 파일에 저장해 둔 보안키를 입력해야 조회 할 수 있습니다. 파일의 내용을 변경하여 보안키 수정이 가능하다. 옵션은 whatap.conf 에서 설정이 가능합니다. 관련 옵션은 다음과 같습니다.

- profile_http_parameter_enabled default: false

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 값이어야만 합니다. 애플리케이션 서버로부터 추출한 정보를 활용하는 이유는 애플리케이션 서버 정지, 네트워크 단절 또는 에이전트 문제로 인한 수집 서버와 에이전트의 통신 단절 상태가 복구되었을 경우, 재접속된 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}

기본 패턴에 대한 변경은 whatap.conf 에서 설정에서 가능합니다.
object_name default: {type}-{ip2}-{ip3}-{process}

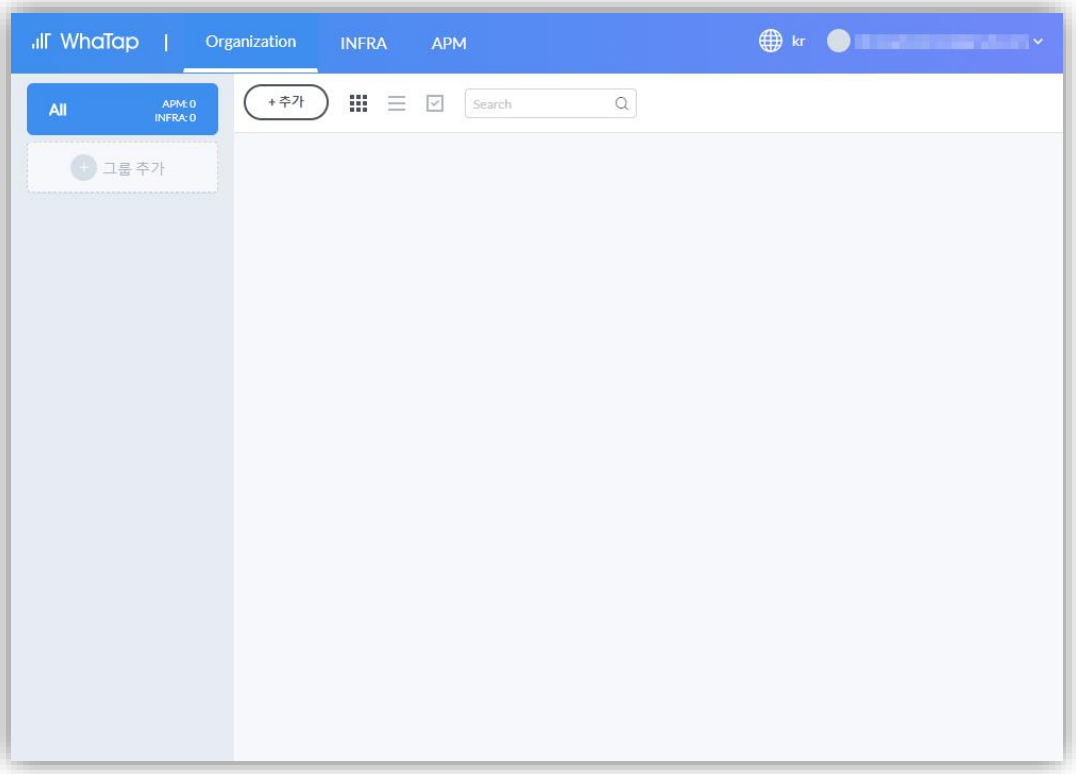
패턴 옵션

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

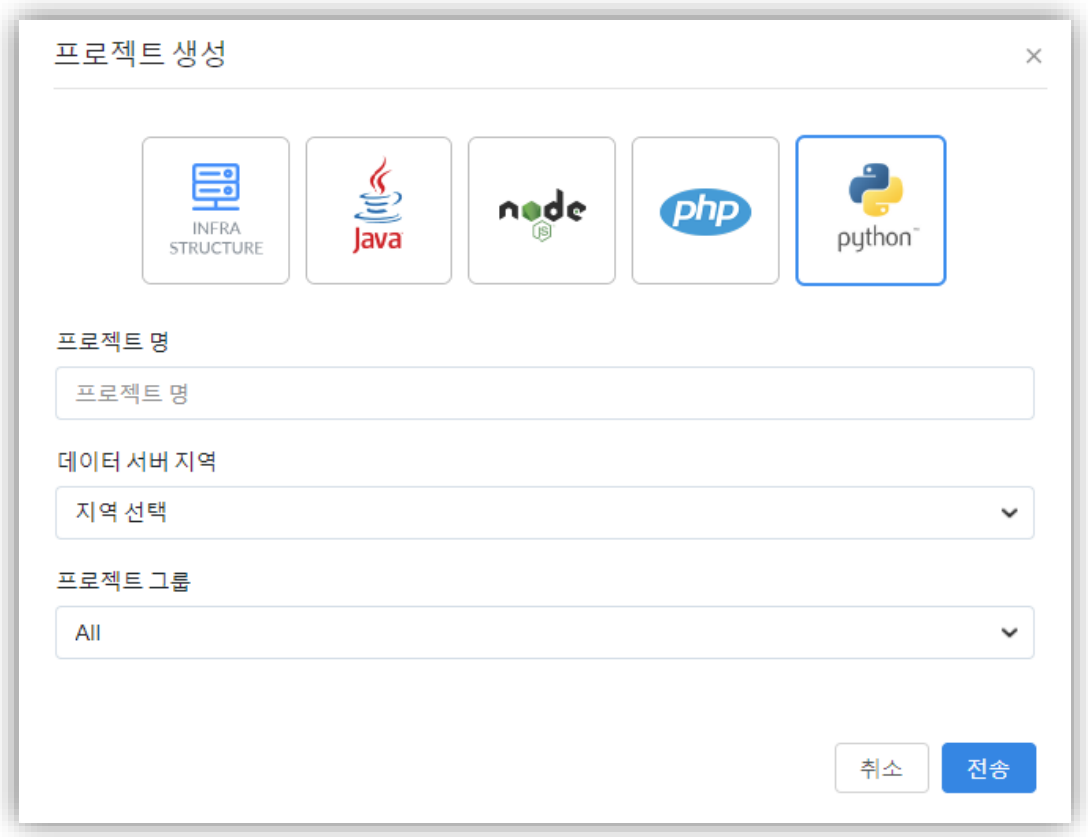
1.1.2. 에이전트 설치 및 패키지 업데이트/제거

1.1.2.1. 설치

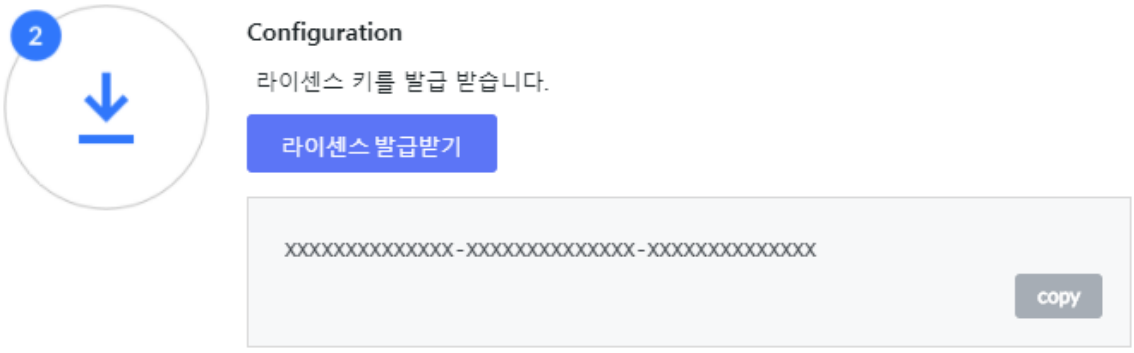
1.1.2.1.1. 프로젝트 생성



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.



1.1.2.1.2. 라이선스 발급



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

1.1.2.1.3. WHATAP_HOME 기본 경로 설정

로그와 설정파일 경로를 위한 $WHATAP_HOME 경로를 지정해 주세요.
`whatap` 디렉토리를 새로 생성하는 것을 권장합니다.

```
$ export WHATAP_HOME=[PATH]
```

1.1.2.1.4. 라이선스 키 및 수집서버 설정

다음 명령어를 실행하면 $WHATAP_HOME 에 지정한 경로에 바로 whatap.conf 파일이 생성 및 설정 됩니다.

```
$ whatap-setting-config
 --host [HOST_ADDR]
 --license [LICENSE_KEY]
 --app_name [APPLICATION_NAME]
 --app_process_name [APP_PROCESS_NAME ex]uwsgi, gunicorn..]
```

1.1.2.1.5. 설정 확인

다음과 같이 설정파일이 잘 생성되어 있는지 확인 해 주세요.

```
$ cat $WHATAP_HOME/whatap.conf
```

1.1.2.1.6. 명령어 실행

WHATAP_AGENT 시작 커맨드와 함께 애플리케이션 서버를 재시작해주세요.

```
$ whatap-start-agent [YOUR_APPLICATION_START_COMMAND]
```

1.1.2.1.7. 애플리케이션 재 시작

애플리케이션 서버가 실행되면 애플리케이션의 모니터링 정보를 수집하기 시작합니다.

1.1.2.1.8. 에이전트 프로세스 확인

다음과 같은 명령어를 통하여 동작중인 와탭 Python 에이전트 프로세스를 확인 할 수 있습니다.

```
$ ps -ef | grep whatap_python
```

1.1.2.1.9. 에이전트 삭제

에이전트 삭제를 원하는 경우, 다음 절차를 진행 해 주세요..

1) 애플리케이션 서버 재시작 시 함께 추가한 WHATAP_AGENT 시작 커맨드를 삭제 합니다.
2) whatap-stop-agent 명령어로 와탭 Python 에이전트 프로세스를 종료합니다.

```
$ whatap-stop-agent
```

2-1) 혹시 되지 않으시면 다음과 같이 프로세스를 종료시켜 주세요.

```
$ killall whatap-python
```

3) 프로세스가 정삭적으로 죽었는지에 대한 확인은 다음 명령어로 확인 가능합니다.

```
$ $ ps -ef|grep whatap_python
```

1.1.2.2. 패키지 설치 확인/업데이트/제거

1.1.2.2.1. 패키지 설치 확인

다음과 같은 명령어를 통하여 설치된 와탭 Python 에이전트 패키지를 확인할 수 있습니다.

```
$ pip list | grep whatap-python
```

1.1.2.2.2. 패키지 업데이트

다음과 같은 명령어를 통하여 설치된 와탭 Python 에이전트 패키지를 업데이트 할 수 있습니다.

```
$ pip install -U whatap-python
```

또는

```
$ pip install -U whatap-python==[특정 버전]
```

1.1.2.2.3. 패키지 제거

다음과 같은 명령어를 통하여 설치된 와탭 Python 에이전트 패키지를 제거할 수 있습니다.

```
$ pip uninstall whatap-python
```

1.1.2.3. 사용 예

1.1.2.3.1. 기본

예제.

```
$ whatap-start-agent python manage.py runserver
```

1.1.2.3.2. uWSGI

예제 1.

```
$ whatap-start-agent uwsgi --ini myapp.ini
```

예제 2.

```
description "uWSGI application server handling myapp"

start on runlevel [2345]
stop on runlevel [!2345]

exec whatap-start-agent [YOUR_APPLICATION_START_COMMAND]
또는
exec env WHATAP_HOME=[PATH] [절대 경로]/whatap-start-agent
[YOUR_APPLICATION_START_COMMAND]
```

예제 3.

```
$ whatap-start-agent gunicorn myapp.wsgi
```

1.1.2.3.3. Gunicorn

예제 1.

```
NAME="uwsgi"
PATH=/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
DAEMON=/usr/local/bin/uwsgi

########## WHATAP_AGENT_CONF ##########
WHATAP_HOME=[PATH]
WHATAP_AGENT=[절대 경로]/whatap-start-agent
...
do_start(){
    env WHATAP_HOME=$WHATAP_HOME $WHATAP_AGENT [YOUR_APPLICATION_START_COMMAND]
}
```

예제 2.

```
description "Gunicorn application server handling myapp"

start on runlevel [2345]
stop on runlevel [!2345]

exec whatap-start-agent [YOUR_APPLICATION_START_COMMAND]
또는
exec env WHATAP_HOME=[PATH] [절대 경로]/whatap-start-agent
[YOUR_APPLICATION_START_COMMAND]
```

예제 3.

```
NAME="gunicorn"
PATH=/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
DAEMON=/usr/local/bin/gunicorn

########## WHATAP_AGENT_CONF ##########
WHATAP_HOME=[PATH]
WHATAP_AGENT=[절대 경로]/whatap-start-agent
...
do_start(){
    env WHATAP_HOME=$WHATAP_HOME $WHATAP_AGENT [YOUR_APPLICATION_START_COMMAND]
}
```

1.1.2.3.4. Supervisor

예제 4.

```
[program:app-uwsgi]
environment = WHATAP_HOME=[PATH]
command = [절대 경로]/whatap-start-agent /usr/local/bin/uwsgi --ini
/home/blog/backend/config/uwsgi.ini

[program:nginx-app]
command = /usr/sbin/nginx
```

1.1.2.3.5. WSGI 애플리케이션 직접 구현

예제.

```
import whatap

@whatap.register_app
def simple_app(environ, start_response):
    """Simplest possible application object"""
    status = '200 OK'
    response_headers = [('Content-type', 'text/plain')]
    start_response(status, response_headers)
    return ['Hello world!\n']
```

1.1.3. 에이전트 로그

애플리케이션 서버가 실행되면 애플리케이션의 모니터링 정보를 수집하기 시작합니다. 로그의 위치는 $WHATAP_HOME 의 logs 디렉토리에 다음과 같은 파일로 생성됩니다.

1.1.3.1. 로그파일 종류

로그 파일은 2 가지로 종류는 다음과 같습니다.

- whatap-boot-[DATE].log : 데이터 수집 로그
- whatap-hook.log : 데이터 수집 로그

1.1.3.2. 정상 동작 확인

다음은 정상 동작하는 경우의 로그 예 입니다.

```
_      ____        ______Python-AGENT
| | /| / / /  ___ /_  __/__ ____
| |/ |/ / _ \/ _ `// / / _ `/ _ \
|__/|__/_//_/\_,_//_/  \_,_/ .__/
                          /_/
Just Tap, Always Monitoring
WhaTap Python Agent version 0.1.33, 20171017

WHATAP_HOME: /Users/kimjihye/workspace/whatap/python-sample/whatap
Config: /Users/kimjihye/workspace/whatap/python-sample/whatap/whatap.conf
Logs: /Users/kimjihye/workspace/whatap/python-sample/whatap/logs
```

1.1.4. 설치 에러 대응

1.1.4.1. Permission denied 에러 발생 시

와탭 파이썬 모니터링을 사용하기 위해서는 다음과 같은 권한 필요 합니다.

1) 와탭 설정을 위한 $WHATPA_HOME/whatap.conf 파일의 읽기 및 쓰기 권한
2) 와탭 로그를 위한 $WHATPA_HOME/logs 디렉토리와 하위 파일의 읽기 및 쓰기 권한

권한 문제가 발생하는 경우(Permission denied error), 다음과 같이 $WHATPA_HOME 에 권한을 부여합니다.

```
$ echo `sudo chmod -R 777 $WHATAP_HOME`
```

1.1.4.2. 프로젝트에 에이전트가 등록되지 않는 경우 모니터링 데이터 수집이 이루어지지 않는 경우

로그 파일($WHATAP_HOME/logs/)을 확인 한 후 각각의 문제에 대하여 다음과 같이 문제를 해결 할 수 있습니다.

- whatap-hook.log
  - CONF FILE ERROR: 설정파일 생성 권한이 없습니다. 파일을 만들어 주세요.
  - CONF READ ERROR: 설정파일은 있으나 리드권한이 없습니다. 권한을 주어야 합니다.
  - LOG FILE ERROR: 로그 디렉토리 생성 권한이 없이 없습니다. 디렉토리를 만들어 주세요.
  - LOGGING ERROR: 로그 디렉토리는 있으나, 쓰기 권한 없습니다. 권한을 주어야 합니다.

3) whatap-boot-[DATE].log

- license or whatp.server.host error: 라이선스키 또는 수집서버 주소가 잘못되었습니다

1.1.4.2.1. 포트 충돌이 발생하는 경우

내부 통신을 하는 에이전트는 기본으로 UDP 6600 포트를 사용합니다. 내부 포트가 충돌이 나는 경우, net_udp_port=xxx 옵션으로 포트를 변경 할 수 있습니다

1.1.4.3. $WHATAP_HOME 환경 변수가 설정되지 않는 경우

1.1.4.3.1. Apache HTTPD

아파치로 웹 서버 구동하는 경우 환경변수 설정을 다음과 같이 해 주어야 합니다.

```
<VirtualHost *:80>
    #ServerName
    #DocumentRoot

        SetEnv WHATAP_HOME "application path"
    # Directory
</VirtualHost>
```

1.1.4.3.2. 수동으로 환경변수 설정 하는 경우

필요에 따라서는, 다음과 같이 수동으로 환경 변수를 설정 해 주어야 합니다.

```
import os
os.environ.setdefault("WHATAP_HOME", [application path]")
import whatap
```

1.1.5. FAQ

1.1.6. 제약 사항

1.1.7. 호환성 목록

1.1.7.1. 운영체제

- CentOS 64bit 6 이상
- Ubuntu 64bit 14 이상

1.1.7.2. Python

- Python 2.7 이상 & Python 3.3 이상

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