

가장 많은 고객이 선택한 클라우드 모니터링 서비스 1위, 와탭
당신의, IT 서비스의 성능을 극대화합니다.



Docs

On-premise Installation Guide

WhaTap Support

Version 1.0.4

Table of Contents

수집서버 설치 가이드	1
1. 안내사항	2
1.1. 와탭 수집서버 모듈	2
1.2. 최소형과 확장형	2
1.3. 라이선스	3
1.4. 라이선스 만기	3
2. 수집 서버 준비	4
2.1. 공통 항목	4
2.2. 최소형 수집 서버 사양	4
2.3. 확장형 수집 서버 사양	4
2.4. 수집서버 Disk 산정	4
2.5. 방화벽 오픈	5
2.6. 파일 디스크립터 설정 사항	5
2.7. 수집서버 시간 동기화	6
3. 설치 작업 절차	7
3.1. 설치 파일 및 라이선스 업로드	7
3.2. JDK 설치	7
3.3. 실행 파일 편집	7
3.4. 설정편집	7
3.5. 로그 경로 변경	10
3.6. 실행	10
3.7. 실행 확인	13
3.8. 점검	15
4. 최소형 설치 예제	16
4.1. 환경	16
4.2. 서버 설치	16
4.3. 기동	17
5. 확장형 설치 예제	19
5.1. 환경	19
5.2. 설치 절차	19
5.3. 기동	25
5.4. 설치 확인	26
6. 알람연동	28
6.1. SMTP 연동	28
6.2. 텔레그램 연동	28
6.3. Slack 연동	32
7. FAQ	37
7.1. 재설치	37
7.2. 사이트 관리자 패스워드	37
7.3. 포트 변경	37
7.4. 설치 파일 다운로드	37
7.5. MySQL DB 사용	37
7.6. Windows 환경 수집서버의 응답지연	39
7.7. 이중화 Proxy 간 리밸런싱	39
7.8. WindowsServices 등록	40
7.9. 최적의 환경	41
7.10. 문의처	42

수집서버 설치 가이드

제목 : On-premise Installation Guide

작성자 : WhaTap Support

이메일 : support@whatap.io

날짜 : 2019-08-14

버전 : 1.0.4

설명 : 본 문서는 WhaTap 설치형 수집서버 설치에 대해 설명합니다. 문서에 관한 문의는 support@whatap.io 로 연락 주시기 바랍니다.

Chapter 1. 안내사항



SaaS형 서비스 사용시 와탭의 수집서버를 사용하므로 별도 수집서버 설치는 불필요 합니다.

1.1. 와탭 수집서버 모듈

와탭 수집서버 모듈은 역할별로 나누어져 있습니다. 하나의 모듈은 JVM Process 하나와 대응합니다.

Table 1. 수집서버 모듈

모듈명	역할	비고
front	UI 프론트	
account	계정 관리	최소형의 경우 front 와 통합
yard	수집데이터 저장, 가공 및 제공	
keeper	데몬의 정보를 관리	
proxy	에이전트에서 수집한 데이터를 송신하는 Proxy	
gateway	확장형에서 front 와 yard 간 gateway 역할	최소형의 경우 불필요
eureka	각 데몬의 서비스 정보를 찾는 DNS 역할	최소형의 경우 불필요
notihub	smtp, sms 등과 연계하여 알람발송	알람 연계시 구성

1.2. 최소형과 확장형

와탭 수집서버는 최소형과 확장형 중 선택해 설치 가능 합니다.

확장형을 적용해야 하는 경우

- 수집서버 다중화가 필요한 경우
- 향후 대규모 확장 가능성 있는 경우
- 10,000TPS 이상의 애플리케이션 모니터링
- 1000대 이상의 서버 모니터링

위 요건에 해당하지 않는다면 최소형 설치로도 충분합니다.

최소형 설치

- ▣ 1대의 수집서버 내에서 모든 서비스 모듈이 구동됨
- ▣ 단순한 설치와 구성

확장형 설치

- ▣ 2대 이상 수집서버가 역할을 분담하는 구성으로 확장이 손쉬움
- ▣ 다량의 에이전트, 프로젝트, 리전을 수용 할 수 있음

1.3. 라이선스

설치형 서버는 별도의 수집서버 라이선스를 필요로 합니다.

일반 고객에게 직접 발급 하지 않으며 기술지원 파트너에게 발급 합니다.

아래 정보들을 포함해 작성 후 license@whatap.io 으로 이메일 요청 합니다.

설치 환경 정보

고객사와 사업명 정보를 기입합니다.

고객 담당자

최종 사용자의 이름, 전화번호, 이메일 정보를 기입합니다.

라이선스 시작/종료일

라이선스 시작/종료일을 기입합니다. 영구 발행인 경우 영구로 기입합니다.

서버 모니터링

대상 코어 수/에이전트 수

어플리케이션 모니터링

대상 코어 수/에이전트 수

DB 모니터링

대상 코어 수/에이전트 수

라이선스 신청 예

```
mailto: license@whatap.io
mailfrom: whatap_engineer@partner.com
Title: XXX주식회사 차세대시스템에 적용할 애플리케이션 모니터링 라이선스 요청 드립니다.
contents:

XXX주식회사 차세대시스템에 적용할 애플리케이션 모니터링 라이선스 요청 드리지 검토후 발급 부탁 드립니다.
정보는 다음과 같습니다.

설치 환경 정보: XXX주식회사 차세대시스템
고객 담당자: 홍길동 과장 / 000-0000-0000 / gd.hong@xxxcompany.co.kr
라이선스 시작/종료일: 2019-08-15 / 영구
서버 모니터링: 0/0
애플리케이션 모니터링: 128 / 32
DB 모니터링: 0/0

감사합니다.
```

1.4. 라이선스 만기

수집서버는 매 시 정각 라이선스 사용기간과 사용량을 체크합니다.

기간 만료나 수량을 초과한 경우 에이전트 데이터 수집은 정상적으로 이루어지나 모니터링 UI로 접속이 제한됩니다.

라이선스 연장 요청은 최초 발급 과정과 동일하며, 연장 임을 명시하여 요청합니다.

Chapter 2. 수집 서버 준비

2.1. 공통 항목

- CPU : X86 (Unix 비권장)
- OS : 안정화 버전 Linux 배포판 (Windows 설치 가능하나 비권장)
- JDK : JDK 1.8 사전 설치
- 높은 File I/O로 인한 운영체제 file descriptor 설정 조정
- 방화벽 사전 오픈
- 서버 NTP 동기화 적용

2.2. 최소형 수집 서버 사양

- 서버(VM) : 1 대
- CPU : 4Core 이상 권장 (최소 2Core)
- Memory : 16G 이상 권장 (최소 8G)
- Disk : 수집서버 Disk 산정 기준에 따름 (OS 영역 별도, SSD 또는 고성능 스토리지 필요)

2.3. 확장형 수집 서버 사양

- 서버(VM) : 2 대 이상
- CPU
 - Front : 2Core 이상 권장 (최소 2Core)
 - Yard : 4Core 이상 권장 (최소 2Core)
- Memory
 - Front : 8G 이상 권장 (최소 4G)
 - Yard : 16G 이상 권장 (최소 8G)
- Disk
 - Front : 2G 이상 (OS 영역 별도)
 - Yard : 수집서버 Disk 산정 기준에 따름 (OS 영역 별도, SSD 또는 고성능 스토리지 필요)

2.4. 수집서버 Disk 산정

Table 2. Disk 사용량 산출 상세

구분	데이터 보존기간	Disk 용량	산출 근거
Infra Monitoring	1 년	8.2G / 연	(1EA * Raw Data 100M * 30일) (Summary Data 14M * 365일) 1 Agent 기준
Application Monitoring	30 일	30G / 월	(1Gb * 30일) 100 TPS 기준

2.5. 방화벽 오픈

2.5.1. 기본

출발지	목적지	프로토콜	포트	용도
에이전트	Yard 서버	TCP	6600	모니터링 데이터 전송
모니터링 PC	Front 서버	TCP	8080	모니터링 UI 사용

2.5.2. 확장형

출발지	목적지	프로토콜	포트	용도
Front	Yard	TCP	6500	사용자 초대
Front	Yard	TCP	8800	모니터링 API 조회
Yard	Front	TCP	6761	서버 정보 등록
Yard	Front	TCP	18080	이벤트 알림 대상 조회

Yard를 다중화 한 경우라면 Yard끼리 서로 통신 가능 해야 합니다.

Table 3. Yard를 다중화 한 경우

출발지	목적지	프로토콜	포트
Yard	Yard	TCP/UDP	6789
Yard	Yard	TCP	7710
Yard	Yard	TCP	6610
Yard	Yard	TCP	6620
Yard	Yard	TCP	6500
Yard	Yard	TCP	8800

2.6. 파일 디스크립터 설정 상황

시스템의 파일 디스크립터 설정을 상황 조정합니다.

/etc/sysctl.conf

```
$ sudo vi /etc/sysctl.conf
fs.file-max = 999999 ①
$ sysctl -p ②
$ sysctl -a | grep fs.file-max = 999999 ③
```

① sysctl.conf 파일 내용에 fs.file-max = 999999 설정을 추가합니다.

② 추가한 설정을 적용합니다.

③ 추가한 설정을 확인 합니다.

설치형 서버를 실행할 사용자에게 대해 파일 디스크립터 설정을 상향 합니다.

/etc/security/limits.conf

```
$ sudo vi /etc/security/limits.conf ①
{사용자 계정 또는 @그룹명} soft nofile 999999
{사용자계정또는 @그룹명} hard nofile 999999
```

① /etc/security/limits.conf 파일에 사용자 또는 그룹 이름으로 soft, hard 설정을 상향 합니다.
사용자 계정은 그대로 사용하고, 그룹은 앞에 @를 붙여서 설정합니다.

2.7. 수집서버 시간 동기화

수집서버의 시간을 동기화 합니다.

ntpq 또는 rdate

```
$ntpq -p #또는
$rdate -s time.bora.net
```



확장형 구성에서 수집서버간 시간이 다른 경우 데이터 누락, 통계 왜곡이 발생하므로 반드시 올바른 시간이 설정되어 있어야 합니다.

Chapter 3. 설치 작업 절차

3.1. 설치 파일 및 라이선스 업로드

단일 서버에 구성하는 경우, 최소형 패키지를 활용하며 향후 확장성을 고려할 때 확장형 패키지를 활용하여 설치를 진행합니다. JDK 및 와탭 모니터링 패키지를 설치 대상 서버에 업로드 하고 압축을 해제합니다.

- 최소형 패키지: whatap_single-*.*.*.tar
- 확장형 패키지: whatap_multi-*.*.*.tar
- JDK 1.8 이상
- 라이선스 파일: 텍스트 파일로 별도 준비

와탭 모니터링 패키지를 압축 해제하면 whatap_package 라는 디렉토리로 압축이 해제됩니다.
본 문서에서 이후 해당 경로를 \$WHATAP_PACKAGE로 기술합니다.

3.2. JDK 설치

대상 서버에 JDK가 설치되어 있는 경우 이를 활용합니다. 설치되어 있지 않은 경우 JDK8 이상을 사전 설치합니다.

본 문서에서 이후 JDK 설치 경로를 \$JDK로 기술합니다.

3.3. 실행 파일 편집

\$WHATAP_PACKAGE/bin 하위에는 쉘 스크립트가 존재합니다. 스크립트에 실행 권한을 부여합니다.

\$WHATAP_PACKAGE/bin

```
$ cd $WHATAP_PACKAGE/bin
$ chmod +x *.sh
```

이후 쉘 스크립트를 편집하여 JAVA_HOME 경로를 지정합니다.

예) front.sh

```
#!/usr/bin/env bash
SERVICE_NAME=front.apm
JAVA_HOME=$JDK
SERVER_HOME=cd ../pwd LIB_HOME=$SERVER_HOME/lib CONF_PATH=$SERVER_HOME/conf
EXE_JAR=ls $LIB_HOME/*.{SERVICE_NAME}.boot* | sort | tail -1 $JAVA_HOME/bin/java -Djava.security.egd=/dev/./urandom
-Dwhatap.log.path=. -Xmx512m -jar $EXE_JAR
```

최소형 설치의 경우 front.sh, keeper.sh, yard.sh, proxy.sh 가 사용되고,
확장형 설치의 경우 eureka.sh, account.sh, front.sh / keeper.sh, yard.sh, proxy.sh, gateway.sh 가 사용됩니다.
부가적으로 이벤트 알림 설정을 추가할 경우 notihub.sh 가 사용됩니다.

3.4. 설정편집

\$WHATAP_PACKAGE/lib 하위에는 어플리케이션 라이브러리(jar), \$WHATAP_PACKAGE/conf 하위에는 설정 파일(conf)이 존재합니다.



실행 파일 편집시 개행코드로 인한 문제 발생 시에는 vi에디터에서 :set ff=unix 로 지정하고 편집합니다.

각 설정 파일에서 필수로 설정해야 하는 정보를 수정합니다.

Table 4. 설치 유형별 수정 대상 설정

구분	파일명	항목명	설정
최소형	account.conf	owner	ifconfig/ipconfig로 식별 가능한 IP
		license	발급받은 서버 라이선스
	front.conf	region.proxy.address	에이전트가 데이터를 전송하게 될 수집 서버의 IP
		admin.password	사이트 관리자 계정 패스워드 (초기값 : admin)
	notihub.conf	mail.host	SMTP 연계를 통한 초대 메일 및 알람발송 등 적용 시
		mail.port	
		mail.username	
		mail.password	
		mail.sender	
		mail.smtp.auth	
		mail.smtp.ssl.enable	
		mail.smtp.starttls.enable	
		mail.smtp.starttls.required	
		smsformat	
		smssender	이벤트 알림 중 문자 발송 시 고객사 커스터마이징 설정을 지정
		smsformat	
확장형	account.conf	owner	ifconfig/ipconfig로 확인 가능한 account 서버 IP
		license	발급받은 서버 라이선스
		eureka.addr	eureka 서버 접근 정보
		eureka.hostname	eureka에 등록할 명칭(복수의 서버가 동일 명칭을 가질 수 있음)
		eureka_client_ip_address	gateway에서 접근 가능한 account 서버 IP
		region.id	첫 번째 리전의 ID
		region.name	
		region.proxy.address	첫 번째 리전의 proxy IP(복수 지정 가능)

구분	파일명	항목명	설정
		mail.host	SMTP 연계를 통한 초대 메일 적용 시
		mail.port	
		mail.username	
		mail.password	
		mail.sender	
		mail.smtp.auth	
		mail.smtp.ssl.enable	
		mail.smtp.starttls.enable	
		mail.smtp.starttls.required	
	front.conf	eureka.addr	eureka 서버 접근 정보
		eureka.hostname	eureka에 등록할 명칭(복수의 서버가 동일 명칭을 가질 수 있음)
		eureka_client_ip_address	front 서버 IP
		admin.password	사이트 관리자 계정 패스워드 (초기값 : admin)
	yard.conf	keeper	yard에서 접근 가능한 keeper 서버 IP:Port
		server.name	keeper에 등록할 이름(서버 단위)
		net_noti_ip	yard에서 접근 가능한 noti 서버 IP
	proxy.conf	keeper	proxy에서 접근 가능한 keeper 서버 IP:Port
		server.name	keeper에 등록할 이름(서버 단위)
	gateway.conf	eureka.addr	eureka 서버 접근 정보
		eureka.hostname	eureka에 등록할 명칭(복수의 서버가 동일 명칭을 가질 수 있음)
		eureka_client_ip_address	account/front 에서 접근 가능한 gateway 서버 IP
		keeper	gateway에서 접근 가능한 keeper 서버 IP:Port
		region.name	region 명 첫 번째 region명은 account.conf에서 지정한 region.name과 일치해야 함 두 번째 이후 region명은 사이트 관리자 페이지에서 지정한 region명과 일치해야 함
	notihub.conf	eureka.addr	eureka 서버 접근 정보

구분	파일명	항목명	설정
		eureka.hostname	eureka에 등록할 명칭(복수의 서버가 동일 명칭을 가질 수 있음)
		eureka_client_ip_address	noti 서버 IP
		keeper	noti에서 접근 가능한 keeper 서버 IP:Port
		mail.host	이벤트 알람 중 SMTP 통한 메일발송 기 적용 시
		mail.port	
		mail.username	
		mail.password	
		mail.sender	
		mail.smtp.auth	
		mail.smtp.ssl.enable	
		mail.smtp.starttls.enable	
		mail.smtp.starttls.required	
		smsformat	
		smssender	이벤트 알람 중 문자 발송 시 고객사 커스터마이징 설정을 지정
		smsformat	

3.5. 로그 경로 변경

실행 시 로그는 \$WHATAP_PACKAGE /logs/{server명}.log 로 출력됩니다. 로그를 외부 경로에 출력할 경우 다음과 같이 지정합니다.

로그 경로 변경

```
$ cd $WHATAP_PACKAGE
$ rmdir logs
$ ln -s {외부경로} logs
$ cd javam/server
$ ln -s {외부경로} logs
```

3.6. 실행

서버 실행은 \$WHATAP_PACKAGE /bin/control.sh를 통해 실행하게 됩니다. 최소형 설치본의 경우 \$WHATAP_PACKAGE /bin/start.sh, \$WHATAP_PACKAGE /bin/stop.sh 파일을 통해서도 실행/정지가 가능합니다.

3.6.1. control.sh 를 통한 실행

최소형 서버 실행 시에는 front를 구성한 서버에서 다음의 명령을 실행합니다.

control.sh - 최소형

```
./control.sh keeper start
./control.sh front start
./control.sh yard start
./control.sh proxy start
./control.sh notihub start (이벤트 알림 설정 시)
```

확장형 서버 실행 시에는 front를 구성한 서버에서 다음의 명령을 실행합니다.

control.sh - 확장형 Front

```
./control.sh eureka start
./control.sh account start
./control.sh front start
./control.sh notihub start
```

yard 구성 서버에서 다음의 명령을 실행합니다.

control.sh - 확장형 Yard

```
./control.sh keeper start
./control.sh yard start
./control.sh proxy start
./control.sh gateway start
```

control.sh 실행 시 다음과 같이 메뉴를 선택하여 작업을 수행할 수도 있습니다.

control.sh - Usage

```
$ cd $WHATAP_PACKAGE/bin
$ ./control.sh

-----
[ Usage ] ./control.sh [ service_name [ command ] ]
./control.sh menu
./control.sh all (stop/status)
./control.sh single (status/start/stop/restart)
./control.sh (multi-front/mf) (status/start/stop/restart)
./control.sh (multi-yard/my) (status/start/stop/restart)
./control.sh front (status/start/stop/restart)
./control.sh proxy (status/start/stop/restart)
./control.sh yard (status/start/stop/restart)
./control.sh keeper (status/start/stop/restart)
./control.sh notihub (status/start/stop/restart)
./control.sh eureka (status/start/stop/restart)
./control.sh gateway (status/start/stop/restart)
./control.sh account (status/start/stop/restart)
/apps/whatap/bin/admin_console.sh
-----
```

```
./control.sh menu
=====#
Select type
=====#
1. service start/stop
2. start-single
3. restart-single
4. start-multi
5. restart-multi
6. status-all
7. stop-all

99. Exit
=====#
number> {원하는 메뉴 번호를 입력함}

=====#
Select command
=====#
1. front start
2. front stop
3. front restart

4. proxy start
5. proxy stop
6. proxy restart

7. yard start
8. yard stop
9. yard restart

10. keeper start
11. keeper stop
12. keeper restart

13. notihub start
14. notihub stop
15. notihub restart

16. billing start
17. billing stop
18. billing restart

19. eureka start
20. eureka stop
21. eureka restart

22. gateway start
23. gateway stop
24. gateway restart

25. account start
26. account stop
27. acdount restart

0. menu
99. Exit
=====#
number> {원하는 메뉴 번호를 입력함}
```

3.7. 실행 확인

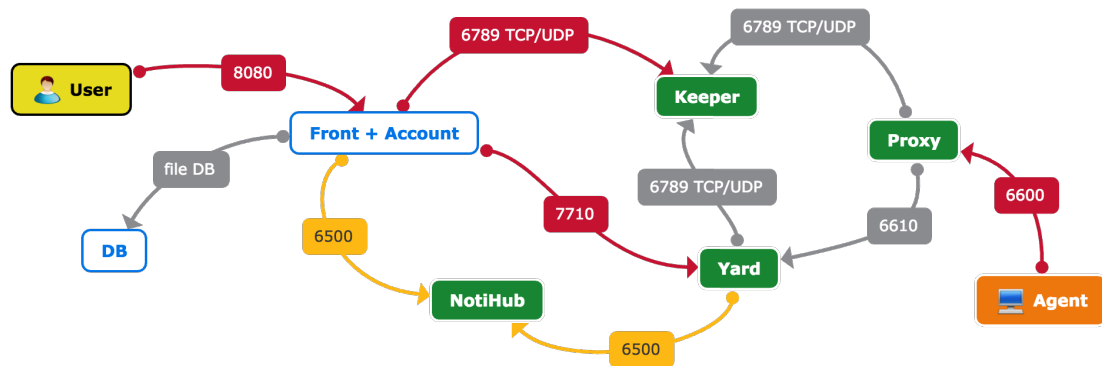
기동 완료 여부는 서버의 포트 리스닝 여부와 로그를 확인합니다.

3.7.1. 포트 리스닝 확인

구성 환경에 따른 서버의 포트 리스닝 여부를 체크합니다.

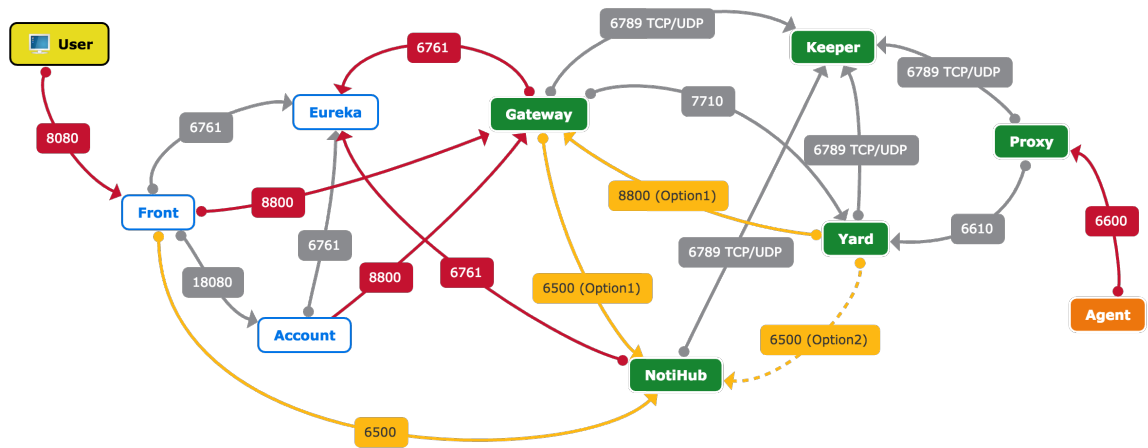
```
$ netstat -na | grep {체크대상포트} | grep LISTEN
```

최소형



어플리케이션 서버	포트	프로토콜	용도
front (+account)	8080	HTTP	모니터링 사이트
keeper	6789	TCP/UDP	서버 정보 수신
yard	7710	TCP	모니터링 데이터 조회
	6610	TCP	모니터링 데이터 저장
proxy	6600	TCP	에이전트 데이터 수신
notihub	6500	TCP/UDP	이벤트 알림

확장형



어플리케이션 서버	포트	프로토콜	용도
eureka	6761	TCP	서버 위치 관리
account	18080	HTTP	계정 관리
front	8080	HTTP	모니터링 사이트
gateway	8800	TCP	리전 요청 수신
keeper	6789	TCP/UDP	서버 정보 수신
yard	7710	TCP	모니터링 데이터 조회
	6610	TCP	모니터링 데이터 저장
proxy	6600	TCP	에이전트 데이터 수신
notihub	6500	TCP/UDP	이벤트 알람

3.7.2. 로그 확인

실행 시 로그는 \$WHATAP_PACKAGE /logs 하위에 출력되므로, 본 경로의 로그를 확인하여 이상 로그 출력 여부를 점검합니다. 서버 초기 기동 시의 상호 접속 실패로 인한 로그는 이상 로그로 간주하지 않습니다.

설치 형태 구분	어플리케이션 로그
최소형/확장형	front.log
	keeper.log
	yard.log
	proxy.log
	notihub.log

설치 형태 구분	어플리케이션 로그
확장형	eureka.log
	gateway.log

3.7.3. 실행 화면 확인

front 서버의 기동이 정상 완료되면, admin@whatap.io / admin 계정으로 로그인 합니다. 본 계정은 와탭 수집 서버 관리용으로 사용되는 계정으로, 수집 서버 관리용으로만 사용됩니다. 본 계정으로 프로젝트 생성하지 않습니다.

화면 우상단의 계정메뉴(이메일 클릭시 노출)에서 사이트 관리를 선택하여 각 화면에 표시되는 내용을 확인합니다. 하기의 메뉴가 노출됩니다.

- ▣ SERVERS: 서버 관리 현황
- ▣ ACCOUNTS: 계정 관리
- ▣ PROJECTS: 프로젝트 생성 현황
- ▣ NOTICE: 공지사항 등록 관리 (설치형에서는 사용불가)
- ▣ REGIONS: 수집 서버 관리 리전 정보
- ▣ MAIL: 최소형 설치 시 메일 설정 변경
- ▣ SITELOGO: 설치형 제품의 로고 변경 옵션

3.8. 점검

이후 점검은 신규 계정을 생성하여 진행합니다. 최소 점검 사항은 다음과 같습니다.

- ▣ 서버 라이선스 점검 : admin@whatap.io 계정으로 접속 후 <http://수집서버주소/v2/admin/check> URL로 접속하여 라이선스 정상 등록 여부를 확인 합니다.
- ▣ 계정 생성 : 로그인 페이지 하단의 계정 생성 링크 또는 사이트 관리자 메뉴의 ACCOUNTS를 통해 계정을 생성합니다. 사이트 관리자 메뉴를 통해 계정을 생성하는 경우, 패스워드에 대한 제약이 존재하지 않습니다.
- ▣ 프로젝트 생성 : 새로 생성한 계정으로 프로젝트를 생성하여 프로젝트 생성 시 이상 현상이 발생하지 않는지 확인합니다.
- ▣ 프로젝트 라이선스 발급 : 프로젝트 카드 클릭 시의 설치 안내 페이지에서 라이선스 발급 버튼을 클릭하고, 이상 현상 발생 여부를 확인합니다.
- ▣ 에이전트를 적용하고 기능 체크를 수행합니다.

Chapter 4. 최소형 설치 예제

최소형 서버를 실제 운영환경에 구성해야 하는 경우 구성에 앞서 예제를 통해 연습 해 볼 수 있습니다.

4.1. 환경

예제에 사용 되는 환경은 다음과 같습니다.

호스트명	항목	내용	비고
vmsvr03	용도	front 서버	
vmsvr03	OS	X86 Linux	
vmsvr03	IP	192.168.122.23	
vmsvr03	구동 서비스	Keeper Proxy Yard Front Notihub	

4.2. 서버 설치

예제 환경의 설치 절차는 다음과 같습니다. 방화벽, JDK 와 같은 제반 환경은 사전 구성되어 있어야 합니다.

파일 업로드 및 환경 구성

```
whatap@vmsvr03:/apps$ ls -alrt ①
합계 519544
drwxr-xr-x 24 root root 4096 7월 6 11:48 ..
-rw-rw-r-- 1 whatap whatap 531998720 12월 17 12:09 whatap_single-1.2.6.2240.tar
drwxrwxr-x 2 whatap whatap 4096 12월 17 12:32
whatap@vmsvr03:/apps$ tar -xvf whatap_single-1.2.6.2240.tar ②
whatap@vmsvr03:/apps$ cd whatap/bin
whatap@vmsvr03:/apps/whatap/bin$ chmod +x *sh ③
whatap@vmsvr03:/apps/whatap/bin$ rm -f *.bat *.exe *solaris.sh ④
```

① 최소형 서버의 설치 파일명은 whatap_single 로 시작합니다.

② 설치 파일 압축을 해제 합니다.

③ 실행 파일 권한 추가 합니다.

④ Linux 환경에서 불필요 파일 제거 합니다. - (선택사항)

Account 설정 변경

/apps/whatap/conf/account.conf 파일을 편집하여 { } 내 항목을 아래값으로 변경합니다.

[illegible]

② {server license} → 발급된 라이선스

/apps/whatap/conf/front.conf파일을편집하여 {}내항목을아래값으로변경합니다.

```
standalone=true
domain=service.whatap.io
keeper=127.0.0.1:6789

region.name=LOCAL
region.id=LOCAL
#region.proxy.address={IP_ADDRESS_accessible_from_AGENT}
region.proxy.address=192.168.122.23 ①

admin.email=admin@whatap.io
admin.password=admin

sessionTimeoutSecs=-1

login_footer_visible=false

# NotiHub
notihub.enable=true
notihub=127.0.0.1:6500
```

① {IP_ADDRESS_accessible_from_AGENT} → 192.168.122.23

Keeper, Proxy, Yard, front 기동

-
- ① control 명령을 사용하여 각각의 데몬을 기동 할 수 있습니다.
 - ② start 명령을 사용하여 일괄 기동 할 수 있습니다.

Chapter 5. 확장형 설치 예제

확장형 서버를 실제 운영환경에 구성해야 하는 경우 구성에 앞서 예제를 통해 연습 해 볼 수 있습니다.

5.1. 환경

예제에 사용 되는 환경은 UI용도의 서버 1기, 수집데이터 저장 용도의 서버 1기 이며 세부 정부는 다음과 같습니다.

5.1.1. UI 용도 Front #1

호스트명	항목	내용	비고
vmsvr01	용도	front 서버	
vmsvr01	OS	X86 Linux	
vmsvr01	IP	192.168.122.21	
vmsvr01	구동 서비스	Eureka Front Account Notihub	

5.1.2. 수집데이터 저장용도 Yard #1

호스트명	항목	내용	비고
vmsvr02	용도	yard 서버	
vmsvr02	OS	X86 Linux	
vmsvr02	IP	192.168.122.22	
vmsvr02	구동 서비스	Keeper Gateway Proxy Yard	

5.2. 설치 절차

예제 환경의 설치 절차는 다음과 같습니다. 방화벽, JDK 와 같은 제반 환경은 사전 구성되어 있어야 합니다.

5.2.1. Front 서버 설치

설치 파일을 업로드 하고 실행 환경을 구성 합니다.

```

whatap@vmsvr01:/apps$ ls -alrt ①
합계 769828
drwxr-xr-x 25 root root 4096 8월 14 17:15 ..
drwxr-xr-x 2 whatap whatap 4096 9월 27 12:08 .
-rw-r--r-- 1 whatap whatap 788295680 9월 27 12:08 whatap_multi-1.1.4.2092.tar
whatap@vmsvr01:/apps$ tar -xvf whatap_multi-1.1.4.2092.tar ②
whatap@vmsvr01:/apps$ cd whatap/bin
whatap@vmsvr01:/apps/whatap/bin$ chmod +x *sh ③
whatap@vmsvr01:/apps/whatap/bin$ rm -f *.bat *.exe *solaris.sh ④
whatap@vmsvr01:/apps/whatap/bin$ rm -f gateway.sh keeper.sh *yard* proxy.sh billing.sh billing.sh adaptor.sh start.sh stop.sh noti.sh
⑤

```

① 확장형 서버의 설치 파일명은 whatap_multi 로 시작합니다.

② 설치 파일 압축을 해제 합니다.

③ 실행 파일 권한 추가 합니다.

④ Linux 환경에서 불필요 파일 제거 합니다. - (선택사항)

⑤ Yard 관련 모듈 제거 - (선택사항)

운영 과정에서 발생 할 수 있는 혼란 방지를 위해 Front 에서 사용 되지 않는 실행 파일을 삭제합니다.

Account 설정 변경

/apps/whatap/conf/account.conf 파일을 편집하여 { } 내 항목을 아래값으로 변경합니다.

/apps/whatap/conf/front.conf

```
# Eureka
#eureka.addr=http://whatap:1qaz$RFV@{EUREKA_IP_accessible_from_FRONT}:6761/eureka/
#eureka.hostname={FRONT}
#eureka_client_ip_address={FRONT_IP}
eureka.addr=http://whatap:1qaz$RFV@192.168.122.21:6761/eureka/ ①
eureka.hostname=FRONT ②
eureka_client_ip_address=192.168.122.21 ③
ribbon_protocol=http
prefer_ip_address=true

# Front Control
standalone=false
domain=service.whatap.io
whatap.admin=admin@whatap.io
loginable=true
sessionTimeoutSecs=-1
login_footer_visible=false

# NotiHub
notihub.enable=true
```

① {EUREKA_IP_accessible_from_FRONT} → 192.168.122.21

② {FRONT} → FRONT

③ {FRONT_IP} → 192.168.122.21

NotiHub 설정

/apps/whatap/conf/notihub.conf 파일을 편집하여 {} 내항목을 아래값으로 변경합니다.


```
# Eureka
eureka.enable=true
#eureka.addr=http://whatap:1qaz$RFV@{EUREKA_IP_accessible_from_NOTI}:6761/eureka/
#eureka.hostname={NOTIHUB}
#eureka_client_ip_address={NOTIHUB_ADDRESS}
eureka.addr=http://whatap:1qaz$RFV@192.168.122.21:6761/eureka/ ①
eureka.hostname=NOTIHUB ②
eureka_client_ip_address=192.168.122.21 ③
ribbon_protocol=http
prefer_ip_address=true

# H2 DB
# Relative path from Home or Absolute path
h2.file.path=./db
h2_auto_server=true
db_driver_event=org.h2.Driver
h2_bind_address=0.0.0.0

jpa_show_sql=false
jpa_generate_ddl=true
jpa_ddl_auto=update

# Mail
mail.sender=no-reply@whatap.io
mail.smtp.debug=false
mail.host=
mail.username=
mail.password=
mail.smtp.protocol=smtp
mail.port=25
mail.smtp.auth=true
mail.smtp.ssl.enable=false
mail.smtp.starttls.enable=true
mail.smtp.starttls.required=true
```

① {EUREKA_IP_accessible_from_FRONT} → 192.168.122.21

② {NOTIHUB} → NOTIHUB

③ {NOTIHUB_IP} → 192.168.122.21

5.2.2. Yard 서버 설치

설치 파일을 업로드 하고 실행 환경을 구성 합니다.

```
whatap@vmsvr02:/apps$ ls -alrt ①
합계 769828
drwxr-xr-x 25 root root 4096 8월 14 17:15 ..
drwxr-xr-x 2 whatap whatap 4096 9월 27 12:08 .
-rw-r--r-- 1 whatap whatap 788295680 9월 27 12:08 whatap_multi-1.1.4.2092.tar
whatap@vmsvr02:/apps$ tar -xvf whatap_multi-1.1.4.2092.tar ②
whatap@vmsvr02:/apps$ cd whatap/bin
whatap@vmsvr02:/apps/whatap/bin$ chmod +x *sh ③
whatap@vmsvr02:/apps/whatap/bin$ rm -rf *.bat *.exe *solaris.sh ④
whatap@vmsvr02:/apps/whatap/bin$ rm -f account.sh eureka.sh adaptor.sh billing.sh noti* start.sh *front.sh ⑤
```

① 확장형 서버의 설치 파일명은 whatap_multi 로 시작합니다.

② 설치 파일 압축을 해제 합니다.

③ 실행 파일 권한 추가 합니다.

④ Linux 환경에서 불필요 파일 제거 합니다. - (선택사항)

⑤ Front 관련 항목 제거 - (선택사항)

운영 과정에서 발생 할 수 있는 혼란 방지를 위해 Yard 에서 사용 되지 않는 실행 파일을 삭제합니다.

Gateway 설정

/apps/whatap/conf/gateway.conf 파일을 편집하여 {} 내항목을 아래값으로 변경합니다.

/apps/whatap/conf/gateway.conf

```
# Eureka
#eureka.addr=http://whatap:1qaz$RFV@{EUREKA_IP_accessible_from_GATEWAY}:6761/eureka/
#eureka.hostname={GATEWAY}
#eureka_client_ip_address={GATEWAY_IP_accessible_from_ACCOUNT}
eureka.addr=http://whatap:1qaz$RFV@192.168.122.21:6761/eureka/ ①
eureka.hostname=GATEWAY ②
eureka_client_ip_address=192.168.122.22 ③
prefer_ip_address=true

# Keeper
#keeper={KEEPER_IP_accessible_from_GATEWAY}:6789
keeper=192.168.122.22:6789 ④

# Account 의 Region Value
#region.name={REGION_NAME_displayed_by_SiteAdminPage}
region.name=REGION-01 ⑤
```

① {EUREKA_IP_accessible_from_GATEWAY} → 192.168.122.21

② {GATEWAY} → GATEWAY

③ {GATEWAY_IP_accessible_from_ACCOUNT} → 192.168.122.22

④ {KEEPER_IP_accessible_from_GATEWAY} → 192.168.122.22

⑤ {REGION_NAME_displayed_by_SiteAdminPage} → REGION-01

Proxy 설정

/apps/whatap/conf/proxy.conf 파일을 편집하여 {} 내항목을 아래값으로 변경합니다.

/apps/whatap/conf/proxy.conf

```
# Keeper
#keeper={KEEPER_IP_accessible_from_PROXY}:6789
#server.name={PROXY-01}
keeper=192.168.122.22:6789 ①
server.name=PROXY-01 ②
secure_file_name=secure.conf

# Self Monitoring
selfmon.enabled=false
```

① {KEEPER_IP_accessible_from_PROXY} → 192.168.122.22

② {PROXY-01} → PROXY-01

Yard 설정

/apps/whatap/conf/yard.conf 파일을 편집하여 {} 내항목을 아래값으로 변경합니다.

/apps/whatap/conf/yard.conf

```
# Keeper
#keeper={KEEPER_IP_accessible_from_YARD}:6789
#server.name={YARD-01}
keeper=192.168.122.22:6789 ①
server.name=YARD-01 ②

# Port
web.port=7710
data.port=6610

# Self Monitoring
selfmon.enabled=false

# Noti
#net_noti_ip={NOTI_IP_accessible_from_YARD}
#net_noti_port=6500

# NotiHub
notihub_enabled=true
#notihub=127.0.0.1:6500
notihub=192.168.122.21:6500 ③
```

① {KEEPER_IP_accessible_from_YARD} → 192.168.122.22

② {YARD-01} → YARD-01

③ notihub=127.0.0.1:6500 → notihub=192.168.122.21:6500

5.3. 기동

5.3.1. Front 서버 기동

Eureka, Front, Account 기동

```
①
whatap@vmsvr01:/apps/whatap/bin$ ./control.sh eureka start
whatap@vmsvr01:/apps/whatap/bin$ ./control.sh front start
whatap@vmsvr01:/apps/whatap/bin$ ./control.sh account start

# 또는
②
whatap@vmsvr01:/apps/whatap/bin$ ./start_front.sh
```

① control 명령을 사용하여 각각의 데몬을 기동 할 수 있습니다.

② start_front 명령을 사용하여 일괄 기동 할 수 있습니다.

Notihub 기동

```
whatap@vmsvr01:/apps/whatap/bin$ ./control.sh notihub start
```

5.3.2. Yard 서버 기동

Keeper, Gateway, Proxy, Yard 기동

```

①
whatap@vmsvr02:/apps/whatap/bin$ ./control.sh keeper start
whatap@vmsvr02:/apps/whatap/bin$ ./control.sh gateway start
whatap@vmsvr02:/apps/whatap/bin$ ./control.sh proxy start
whatap@vmsvr02:/apps/whatap/bin$ ./control.sh yard start

# 또는
②
whatap@vmsvr02:/apps/whatap/bin$ ./start_yard.sh

```

① control 명령을 사용하여 각각의 데몬을 기동 할 수 있습니다.

② start_yard 명령을 사용하여 일괄 기동 할 수 있습니다.

5.4. 설치 확인

브라우저에서 <http://192.168.122.21:8080> 에 admin@whatap.io 로 접속 후 확인합니다.

1. 우측 상단의 계정명 클릭해 “사이트 관리” 메뉴로 이동



1. 등록항목 확인 및 REGION-01 정보 확인

NOTIHUB, REGION-01, FRONT, ACCOUNT 항목이 보여지는지 확인 후 REGION-01의 화살표 버튼을 클릭해 이동



1. REGION-01 세부 정보 확인

KEEPER, PROXY, YARD 정보가 확인되면 정상 입니다

Region: REGION-01

K8senv

서버명	주소	정보
k8senv	192.168.122.22:4789	

Proxy

서버명	주소	해이컨트 수	정보
PROXY-01	192.168.122.22:7700	0	

Yard

서버명	주소	해이컨트 수	설정	정보
YARD-01	192.168.122.22:7710	0	<input checked="" type="checkbox"/>	<button>Check DBIO</button>

Project

Master: ALL Status: 전체

Project	이름	코어 수	해이컨트 수	리소스	지정	상태	삭제	정보
---------	----	------	--------	-----	----	----	----	----

1. 라이선스 정보 확인

<http://서버접속정보/v2/admin/check> URL로 접속하여 등록된 라이선스 정보가 올바른지 확인 합니다.

← → ↻ ⓘ 주의 요함 | demo.whatapkvm.io:8080/v2/admin/check

Whatap 관리자

License: [REDACTED] 51k8: Send

Start Time : 2018-08-17 End Time : 2019-12-31

Type	Used	License
Application	Agent	0
	Core	9999
Infra	Agent	0
	Core	9999
Database	Agent	0
	Core	9999
Cloud	Agent	0
	Core	999
BSM	Agent	0
	Core	0

Start Metering

Chapter 6. 알람연동

6.1. SMTP 연동

SaaS 환경에서는 별도 설정 없이 이메일 계정을 통해 알람 및 초대 발송이 가능하나 설치형 환경에서 이메일 발송을 위해 대상 환경의 SMTP 정보를 등록 해 주어야 합니다.

SMTP 연동 설정은 \${WHATAP_HOME}/conf/notihub.conf 에서 MAIL 항목을 다음과 같이 변경합니다.

/apps/whatap/conf/notihub.conf

```
...  
  
# Mail  
mail.smtp.debug=false ①  
mail.host= ②  
mail.username= ③  
mail.password= ④  
mail.smtp.protocol=smtp  
mail.port=25 ⑤  
mail.smtp.auth=true ⑥  
mail.smtp.ssl.enable=false ⑦  
mail.smtp.starttls.enable=true ⑧  
mail.smtp.starttls.required=true ⑧  
  
...
```

- ① 상세 로그를 \${WHATAP_HOME}/logs/notihub.log 에 출력할 지 여부를 설정합니다.
- ② SMTP 서버 호스트 정보를 입력합니다.
- ③ SMTP 계정 정보를 입력합니다.
- ④ SMTP 계정에 대한 비밀번호를 입력합니다.
- ⑤ SMTP 포트 정보를 입력합니다.
- ⑥ SMTP 인증 여부를 설정합니다. 계정으로 인증하지 않는 경우 false 로 설정합니다.
- ⑦ SSL을 사용하는 경우 true 로 설정합니다.
- ⑧ TLS를 사용하는 경우 true 로 설정합니다.

notihub를 재기동 합니다.

```
$ ./control.sh notihub restart
```

6.2. 텔레그램 연동

와탭의 알람은 텔레그램 메시지를 통해서도 수신 할 수 있습니다.

SaaS 서비스와 달리 설치형 서비스에서는 텔레그램 연동을 위해서는 토큰정보를 등록하고 봇을 생성하는 과정이 필요합니다.

텔레그램 클라이언트에서 @botfather 사용자를 채팅 목록에 추가합니다.

텔레그램클라이언트에서 @BotFather를검색하거나웹브라우저에서 <https://telegram.me/botfather>주소로접속해 @BotFather사용자를 추가할수있습니다.

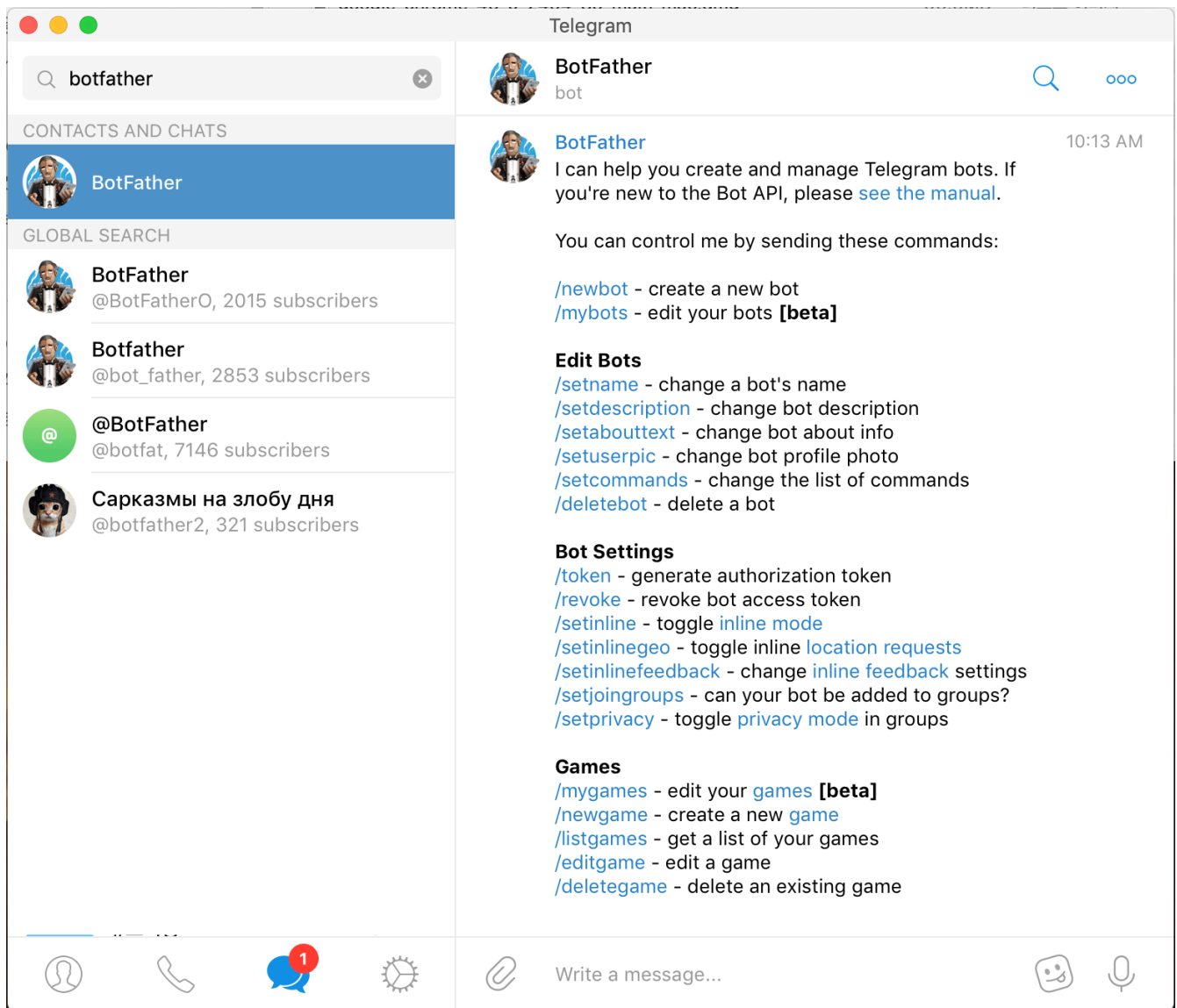


Figure 1. botfather 를 추가한 화면

새로운 봇을 생성하고 토큰을 발급 받습니다.

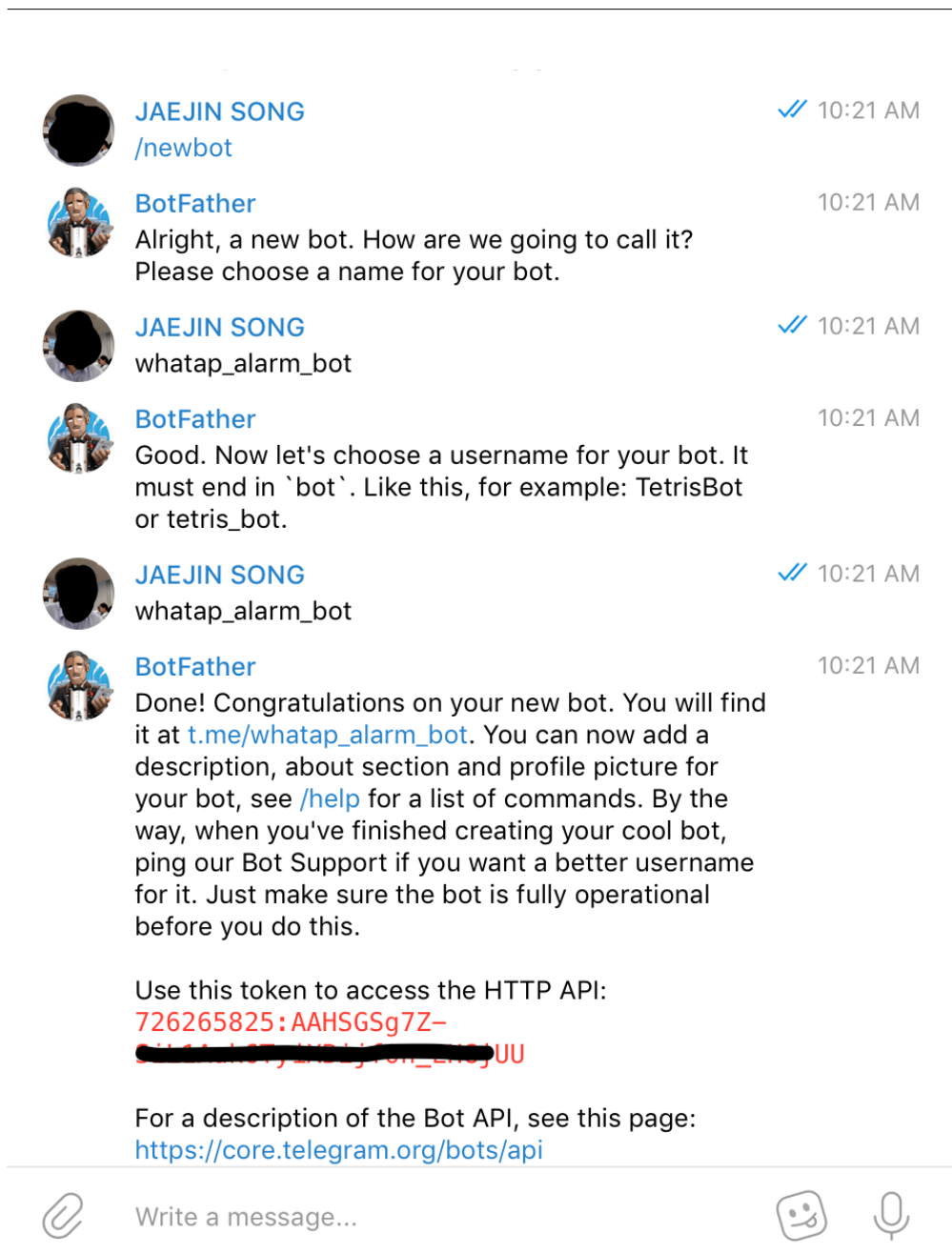


Figure 2. 봇 생성

- ① /newbot 을 입력합니다.
- ② 이름을 입력 합니다. 예) 와탭알람테스트
- ③ 계정정보를 입력합니다. 계정은 영문이며 bot으로 끝나야 합니다. 예) whatap_alarm_bot
- ④ 봇 생성이 완료되면 토큰이 발행됩니다.

생성한 계정으로 봇을 검색하고 대화방에 생성한 봇을 초대 합니다.

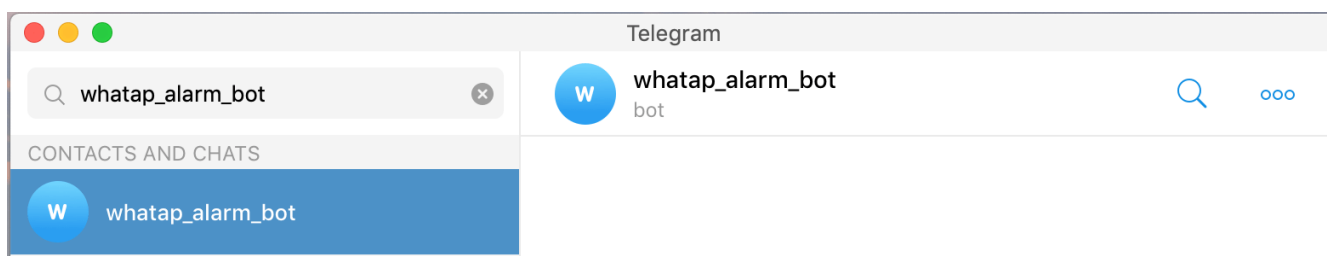


Figure 3. 생성한 계정으로 검색 후 추가

발급받은 텔레그램 토큰은 \${WHATAP_HOME}/conf/notihub.conf 에 추가 합니다.

/apps/whatap/conf/notihub.conf

```
telegram.token= ①
```

① telegram.token 속성으로 토큰을 등록 합니다.

notihub를 재기동 합니다.

```
$ ./control.sh notihub restart
```

관리 - 이벤트 설정 메뉴로 이동해 텔레그램 박스를 선택합니다.

The screenshot shows the '관리' (Management) section of the WhatAp interface, specifically the '이벤트 설정' (Event Settings) menu. The interface is in Korean. On the left is a blue sidebar with various icons. The main area has a header with '관리' and '이벤트 설정'. Below this, there are tabs for '애플리케이션 비활성성 경고' (Application Inactivity Warning) and '채널' (Channel). Under '채널', there are checkboxes for '텔레그램' (Telegram) and '슬랙' (Slack), with an 'Add to Slack' button. Below these are four panels for different metrics: CPU, 디스크 (Disk), 메모리 (Memory), and 히트맵 세로 라인 (Hitmap Vertical Line). Each panel has a slider with markers at 70% and 90% (except for 히트맵 세로 라인 which has markers at 30% and 50%), and buttons for '반복' (Repeat) and '무음' (Mute) with time intervals like '30 sec' and '5 min'. There are also toggle switches for each panel.

생성된 봇과의 대화창에서 /help 를 입력하면 사용법이 안내됩니다.

The screenshot shows a Telegram chat interface. At the top, a message from 'JAEJIN SONG' with the text '/help' is shown, timestamped '11:43 AM'. Below it, a response from the bot 'whatap_alarm_bot' is shown, timestamped '11:43 AM'. The bot's message lists several commands: '/list : Show registered project code list', '/license : Register project notification', '/license [Apm project license Key]', '/remove : Remove registered project notification', and '/remove [Apm project code]'. At the bottom of the chat window is a text input field with the placeholder 'Write a message...' and icons for attachments, emojis, and voice recording.

알람을 받고자 하는 프로젝트의 라이선스를 봇에 등록하면 알람을 수신 할 수 있습니다.

프로젝트의 라이선스는 에이전트 설정(whatap.conf)에 사용된 것과 동일하며 에이전트 설치 메뉴에서 확인 가능합니다.

/license <프로젝트 라이선스> ⓘ

ⓘ 봇과의 대화창에 대상 프로젝트 라이선스를 등록합니다.

6.3. Slack 연동

설치형 서비스에서 Slack 을 연동하기 위해서는 Custom App을 생성하고 해당 인증을 사용합니다.

Slack 에 로그인 후 <https://api.slack.com/apps> 로 이동합니다.

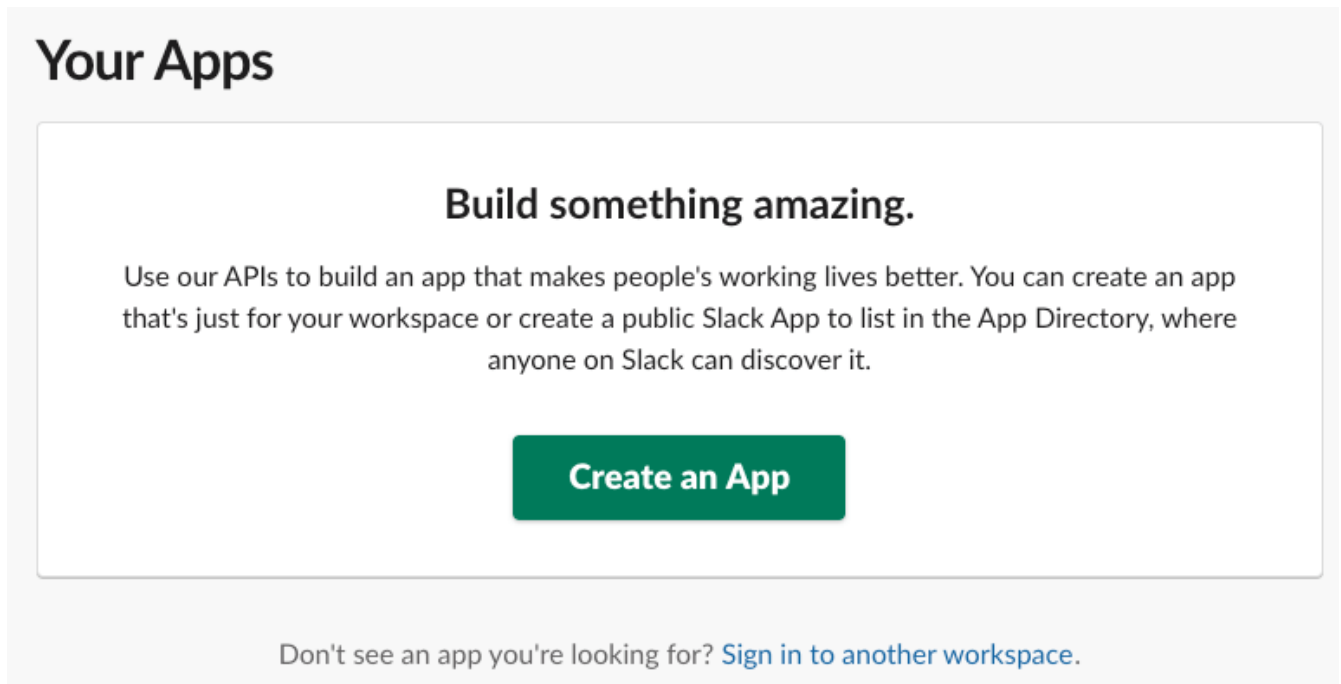


Figure 4. 만들기 버튼

Create an App 버튼을 클릭해 적당한 이름의 앱을 생성합니다.


Create a Slack App

App Name

Slack_Example_FAQ

Don't worry; you'll be able to change this later.

Development Slack Workspace

 jjsongtest

▼

Your app belongs to this workspace—leaving this workspace will remove your ability to manage this app. Unfortunately, this can't be changed later.

By creating a Web API Application, you agree to the [Slack API Terms of Service](#).

Cancel

Create App

Figure 5. App 생성메뉴

App Credentials 에서 인증 정보를 확인합니다.

App Credentials

These credentials allow your app to access the Slack API. They are secret. Please don't share your app credentials with anyone, include them in public code repositories, or store them in insecure ways.

App ID

AFKRZKSPJ

Date of App Creation

January 22, 2019

Client ID

508472741975.529883672800

Client Secret

.....

Show

Regenerate

You'll need to send this secret along with your client ID when making your [oauth.access](#) request.

Signing Secret

.....

Show

Regenerate

Slack signs the requests we send you using this secret. Confirm that each request comes from Slack by verifying its unique signature.

Verification Token

AXvDhdDdzlBy0UW9uh9i5vlc

Regenerate

This deprecated Verification Token can still be used to verify that requests come from Slack, but we strongly recommend using the above, more secure, signing secret instead.

Figure 6. 인증정보

OAuth & Permissions 메뉴를 선택해 Redirect URLs에 수집서버 URL을 등록합니다.

{수집서버_URL}/slack/auth주소를추가합니다.

Slack_Example_FAQ ▾

Settings

Basic Information

Collaborators

Install App

Manage Distribution

Features

Incoming Webhooks

Interactive Components

Slash Commands

OAuth & Permissions

Event Subscriptions

Bot Users

User ID Translation

Slack ♥

Help

Contact

Policies

Our Blog

OAuth & Permissions

OAuth Tokens & Redirect URLs

These [OAuth Tokens](#) will be automatically generated when you finish connecting the app to your workspace. You'll use these tokens to authenticate your app.

Install App to Workspace

Redirect URLs

You will need to configure redirect URLs in order to automatically generate the Add to Slack button or to distribute your app. If you pass a URL in an OAuth request, it must (partially) match one of the URLs you enter here. [Learn more.](#)

Redirect URLs

http://demo.whatapkv.io:8080/slack/auth

CancelDone

Add New Redirect URL

Save URLs

수접서버 {WHATAP_HOME}/conf/front.conf 에 확인한 인증정보를 추가 합니다.

```
slack.clientId=508472741975.529883672800
slack.clientSecret=32634xxxxxxxxxxxxxxxxxxxxxxxxb1f
```

이벤트 알림을 수신하고자 하는 프로젝트의 [관리 - 이벤트 설정](#) 메뉴로 이동해 [Add to Slack](#) 버튼을 클릭합니다.

35

애플리케이션 비활성 경고

채널

후 30초 ☐ 활성화

☒ 텔레그램 ☐ 슬랙 [Add to Slack](#)

경고 치명적

CPU

70 %

90 %

반복 30 sec

무음 5 min

디스크

70 %

90 %

반복 30 sec

무음 5 min

메모리

70 %

90 %

반복 30 sec

무음 5 min

히트맵 세로 라인

30 %

50 %

반복 5 sec

무음 5 min

Chapter 7. FAQ

7.1. 재설치

설정 오류 혹은 설정 변경을 통해 문제를 해결할 수 없는 경우, 재설치 하는 과정을 안내합니다. 재설치 시, 기존 모니터링 정보는 폐기합니다.

로그 삭제

```
$ cd $WHATAP_PACKAGE $ rm -rf logs/*
```

모니터링 데이터 삭제

```
$ cd $WHATAP_PACKAGE $ rm -rf yardbase/*
```

DB 삭제

```
$ cd $WHATAP_PACKAGE $ rm -rf db/*.db
```

이후 재설치 과정은 신규 설치 과정과 동일합니다.

7.2. 사이트 관리자 패스워드

사이트 관리자 계정(admin@whatap.io) 은 변경 불가능합니다. 사이트 관리자 계정의 패스워드는 front.conf의 admin.password 를 통해 설정 가능합니다.

7.3. 포트 변경

어플리케이션 서버는 디폴트로 지정된 포트가 사전 점유되어 있는 경우, 포트 번호를 +1씩 증가시켜 가용한 포트를 활용합니다. 임의로 포트를 지정하여 사용해야 할 경우 다음의 옵션을 적용합니다.

파일	설정	용도	디폴트
front.conf	web.port	사이트 접근	8080
proxy.conf	data.port	모니터링 데이터 수신 (서버측)	6600
whatap.conf	whatap.server.port	모니터링 데이터 전송(에이전트)	6600

7.4. 설치 파일 다운로드

문의로 요청시 Google Drive 및 AWS S3를 통해 제공 (라이선스 요청 시 별도 제공)

7.5. MySQL DB 사용

수집된 모니터링 데이터는 RDB 를 사용하지 않고 와탭 만의 독자적인 저장구조를 가지지만 이외 계정 정보, 구성 정보, 알림 발송 이력은 내장 DB인 H2를 사용합니다. 이를 MySQL DB로 변경하는 방법 입니다.

계정 정보 및 구성 정보 DB로 MySQL 사용은 \${WHATAP_HOME}/conf/account.conf 을 변경합니다.



MySQL에서 account 명의 DB와 CRUD 권한이 있는 {whatap_dbuser} 가 존재 해야 합니다.

/apps/whatap/conf/account.conf

```
...

# DB
# Relative path from Home or Absolute path
#h2.file.path=./db ①
#db_driver=org.h2.Driver ①
#h2_auto_server=true ①
#h2_bind_address=127.0.0.1 ①
#jpa_ddl_auto=update ①
#db_name=account ①
#db_name_project=account ①

# DB for MySQL
db_driver=com.mysql.jdbc.Driver ②
jdbc_url=jdbc:mysql://127.0.0.1:3306 ②
db_username={whatap_dbuser} ②
db_password={whatap_dbpassword} ②
db_name=account ②
db_name_project=account ②

...
```

① 기존 설정은 주석처리 하거나 삭제 합니다.

② 사용 할 Mysql DB 정보를 입력 합니다.

알림 이력의 MySQL 사용은 \${WHATAP_HOME}/conf/notihub.conf 에서 DB 항목을 다음과 같이 변경합니다.

/apps/whatap/conf/notihub.conf

```
...

# H2 DB
# Relative path from Home or Absolute path
#h2.file.path=./db ①
#h2_auto_server=true ①
#db_driver_event=org.h2.Driver ①
#h2_bind_address=0.0.0.0 ①

db_driver_event=com.mysql.jdbc.Driver ②
jdbc_url_event=jdbc:mysql://127.0.0.1:3306 ②
db_username_event=whatap_dbuser ②
db_password_event=whatap_dbpassword ②
db_name_event=account ②

...
```

① 기존 설정은 주석처리 하거나 삭제 합니다.

② 사용 할 Mysql DB 정보를 입력 합니다.

관련 서비스를 재기동 합니다.

```
$ ./control.sh account restart
$ ./control.sh notihub restart
```

자동생성되는 DDL 문제로 MySQL 사용시 컬럼정보 수정이 필요합니다.

```
ALTER TABLE EventPolicy MODIFY Rules TEXT;
```

7.6. Windows 환경 수집서버의 응답지연

Windows 2012 이상에서 ECN Capability 로 인해 수집서버의 반응 지연이 발생할 수 있습니다. 이 경우 ECN Capability 기능을 Disable 합니다.

CMD 관리자권한

```
netsh int tcp set global ecncapability=disabled
```



네트워크가 느린 시절 과도한 데이터 트래픽으로 인한 네트워크 붕괴 현상이 문제가 되었고 이를 막기 위해 TCP 혼잡 제어(TCP Congestion Control) 알고리즘이 제안/적용되었습니다. 기본적으로 TCP 혼잡 제어는 TCP 데이터 송신자 측에서 패킷 유실 발생(혹은 그에 준하는 상태)을 감지한 이후 송신자의 데이터 전송률을 낮추는 방식으로 동작합니다.

7.7. 이중화 Proxy 간 리밸런싱

Proxy 서버가 이중화 된 경우 서비스 재가동시 나머지 Proxy로 에이전트가 Fail-Over 되어 집중되는 현상이 발생 합니다. 이 경우 아래 스크립트 참고하여 Proxy 서비스를 리밸런싱 할 필요가 있습니다.

```
#!/bin/bash

CNT1=0;
CNT2=0;
PROXY1=http://{Proxy1-IP}:{Proxy-Port};
PROXY2=http://{Proxy2-IP}:{Proxy-Port};
RESULT=0;

function chkCount() {
    RESULT=`curl $1/tcp/agent/count |awk -F"="| " '{print $2}'`;
}

function rbCount() {
    RESULT=`curl $1/tcp/agent/reset?cut=$2`;
}

# Check Count
chkCount $PROXY1;
CNT1=$RESULT;
chkCount $PROXY2;
CNT2=$RESULT;
echo "[Before-Proxy1] : $CNT1";
echo "[Before-Proxy2] : $CNT2";

# Rebalance
if [ $CNT1 -gt $CNT2 ]; then
    rbCount $PROXY1 $(((CNT1-CNT2)/2));
    sleep 10;
elif [ $CNT1 -eq $CNT2 ]; then
    echo "Equal";
else
    rbCount $PROXY2 $(((CNT2-CNT1)/2));
    sleep 10;
fi;

# ReCheck Count
chkCount $PROXY1;
CNT1=$RESULT;
chkCount $PROXY2;
CNT2=$RESULT;
echo "[After-Proxy1] : $CNT1";
echo "[After-Proxy2] : $CNT2";
```



리밸런싱 작업을 상시로 할 필요는 없습니다.
조치가 없어도 시간이 지나면 해소 됩니다.

7.8. WindowsServices 등록

Windows OS에 수집서버 설치시 WindowsServices 에 등록이 원활하지 않은 경우 아래와 같이 조치가 가능합니다.

7.8.1. 추가 파일

다운로드 링크

https://s3.ap-northeast-2.amazonaws.com/repo.whatap.io/package/install_WindowsServer.zip

파일목록

```
install_WindowsServer.bat
instsrv.exe
srvany.exe
```

7.8.2. 사용법

WhaTap 수집서버의 MicroServer를 WindowsServices에 등록하고 삭제하는 기능을 제공합니다.

등록대상 MicroServer

```
keeper / yard / proxy / front / notihub
```

등록방법

```
> install_WindowsServer.bat create <등록대상>

ex) install_WindowsServer.bat create keeper
```

삭제방법

```
> install_WindowsServer.bat delete <삭제대상>

ex) install_WindowsServer.bat delete keeper
```

WindowsServices 등록명

```
1) keeper -> WhaTapKeeper
2) yard -> WhaTapYard
3) proxy -> WhaTapProxy
4) front -> WhaTapFront
5) notihub -> WhaTapNotihub
```

패치 등 활용법

```
# 레지스트리편집기를 통한 실행커맨드 수정
1) 레지스트리편집기 열기 (regedit)
2) 위치 이동 : HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\{WhaTap등록명}\Parameters
3) 'Application' 키 : java 실행커맨드 내용 수정
```

7.9. 최적의 환경

최적의 사용 환경을 위해서 다음 사항을 확인 해 주세요.

1. 권장 사양의 서버로 구성합니다.
 - X86 4Core, 16Gb Memory, Linux OS
2. 3000IOPS 이상, latency 10ms 이하 SSD로 저장소 구성합니다.
 - AWS: gp2, Azure: Premium SSD, GCP: Zonal SSD persistent disks
3. Java Heap Min/Max는 동일한 값으로 여유있게 설정합니다.
4. Java8 최신 Fix 를 유지하고 G1GC, StringDedup 적용합니다.
"-XX:+UseG1GC -XX:+UseStringDeduplication"

이들 사항을 준수하면 1000대 이상의 서버 또는 20,000TPS 애플리케이션을 충분히 수용할 수 있습니다.

7.10. 문의처

Tel : 02-565-1803

Mail : support@whatap.io