

신규 Zabbix 전체 구성

- 서버 부터 에이전트, 알림 등 전체적인 기본 구성 작업 정리
- Ubuntu 18.04.4 기준

Zabbix 전체 구성 진행

[Zabbix Server 구축]

```
# wget https://repo.zabbix.com/zabbix/6.0/ubuntu/pool/main/z/zabbix-
release/zabbix-release_6.0-4%2Bubuntu18.04_all.deb
( Zabbix repo 패키지 다운로드 )

*ERROR: cannot verify ~~ 에러 발생시 해당 명령어 뒤에 --no-check-certificate 입력

# sudo apt install ./zabbix-release_6.0-4+ubuntu18.04_all.deb
( Zabbix repo 를 Ubuntu 시스템에 추가 )

# sudo apt update

* zabbix repo 업데이트 err발생시
-----
# vi /etc/apt/sources.list.d/zabbix.list
https -> http 로 변경
-----

# sudo apt -y install zabbix-server-mysql zabbix-frontend-php zabbix-agent zabbix-
apache-conf zabbix-sql-scripts

# wget https://repo.mysql.com//mysql-apt-config_0.8.13-1_all.deb
( Zabbix 6.0 버전 이상부터 mysql 8.0 이상 버전만 호환되기 때문에 따로 설치 )

# dpkg -i mysql-apt-config_0.8.13-1_all.deb

# apt update
( 업데이트 후 나오는 키 정보 복사하여 아래 명령어에 삽입 )
# apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys [key 정보]
( 키 정보 등록 후 재 업데이트 )
# apt update

# apt list | grep mysql-server
( 설치 전 버전 정보 확인 )
# apt -y install mysql-server
# mysql -V
( 설치 후 버전 확인 )

# vi /etc/apache2/conf-enabled/zabbix.conf
-----
20 php_value date.timezone Asia/Seoul
```

```

-----
( timezone 주석 풀고 Asia/Seoul로 변경 )

# mysql
( MYSQL 설정 )
-----

> CREATE DATABASE zabbix CHARACTER SET utf8 collate utf8_bin;
> CREATE USER 'zabbix'@'localhost' IDENTIFIED WITH mysql_native_password BY
'zabbix';
> grant all privileges on zabbix.* to zabbix@localhost;
> set global log_bin_trust_function_creators = 1;
> exit;
-----

「SQLSTATE[HY000] [2054] The server requested authentication method unknown to the
client」
-> 웹 초기접속 중 해당 에러 발생 시 mysql_native_password 부분 설정 재 확인

# vi /etc/zabbix/zabbix_server.conf
-----
129 DBPassword=zabbix
-----
( 설정한 DB 패스워드 입력 )

# systemctl stop mysql
( DB 정지 )

# mkdir /data

# vi /etc/mysql/mysql.conf.d/mysqld.cnf
-----

#
# The MySQL database server configuration file.
#
# You can copy this to one of:
# - "/etc/mysql/my.cnf" to set global options,
# - "~/.my.cnf" to set user-specific options.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html

# This will be passed to all mysql clients
# It has been reported that passwords should be enclosed with ticks/quotes
# especially if they contain "#" chars...
# Remember to edit /etc/mysql/debian.cnf when changing the socket location.

```

```
# Here is entries for some specific programs
# The following values assume you have at least 32M ram

[mysqld_safe]
socket          = /var/run/mysqld/mysqld.sock
nice            = 0

[mysqld]
#
# * Basic Settings
#
user            = mysql
pid-file        = /var/run/mysqld/mysqld.pid
socket          = /var/run/mysqld/mysqld.sock
port           = 3306
basedir         = /usr
datadir         = /data
#datadir        = /var/lib/mysql
tmpdir          = /tmp
lc-messages-dir = /usr/share/mysql
skip-host-cache
skip-name-resolve
skip-external-locking
explicit_defaults_for_timestamp=1
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address    = 127.0.0.1
#
# * Fine Tuning
#
key_buffer_size      = 256M
max_allowed_packet   = 16M
thread_stack         = 192K
thread_cache_size    = 8
# This replaces the startup script and checks MyISAM tables if needed
# the first time they are touched
myisam-recover-options = BACKUP
max_connections      = 200
#table_open_cache    = 64
#thread_concurrency  = 10
#
# * Query Cache Configuration
#
#query_cache_limit    = 1M
#query_cache_size     = 16M
#
# * Logging and Replication
#
# Both location gets rotated by the cronjob.
# Be aware that this log type is a performance killer.
# As of 5.1 you can enable the log at runtime!
#general_log_file      = /var/log/mysql/mysql.log
#general_log           = 1
```

```

#
# Error log - should be very few entries.
#
log_error = /var/log/mysql/error.log
#
# Here you can see queries with especially long duration
#slow_query_log            = 1
#slow_query_log_file      = /var/log/mysql/mysql-slow.log
#long_query_time = 2
#log-queries-not-using-indexes
#
# The following can be used as easy to replay backup logs or for replication.
# note: if you are setting up a replication slave, see README.Debian about
#       other settings you may need to change.
#server-id                = 1
#log_bin                  = /var/log/mysql/mysql-bin.log
expire_logs_days          = 10
max_binlog_size           = 100M
#binlog_do_db             = include_database_name
#binlog_ignore_db         = include_database_name
#
# * InnoDB
#
# InnoDB is enabled by default with a 10MB datafile in /var/lib/mysql/.
# Read the manual for more InnoDB related options. There are many!
default_storage_engine    = InnoDB
innodb_buffer_pool_size   = 2048M
innodb_data_home_dir      = /data
innodb_log_buffer_size    = 8M
innodb_file_per_table     = 1
innodb_open_files         = 400
innodb_io_capacity        = 400
innodb_flush_method       = O_DIRECT
#
# * Security Features
#
# Read the manual, too, if you want chroot!
# chroot = /var/lib/mysql/
#
# For generating SSL certificates I recommend the OpenSSL GUI "tinyca".
#
# ssl-ca=/etc/mysql/cacert.pem
# ssl-cert=/etc/mysql/server-cert.pem
# ssl-key=/etc/mysql/server-key.pem
-----
( Datadir 부분 확인 )

# lsof | grep /var/lib/mysql
( 기존경로에서 열린 mysql 관련 파일이 있는지 확인 )

# cd /var/lib/mysql
( mysql 데이터가 쌓인 디렉토리로 이동 )
# ll

```

```

drwx----- 6 mysql mysql 4096 5월 10 14:20 ./
drwxr-xr-x 48 root root 4096 5월 10 14:15 ../
-rw-r----- 1 mysql mysql 56 5월 10 13:10 auto.cnf
-rw----- 1 mysql mysql 1676 5월 10 13:10 ca-key.pem
-rw-r--r-- 1 mysql mysql 1112 5월 10 13:10 ca.pem
-rw-r--r-- 1 mysql mysql 1112 5월 10 13:10 client-cert.pem
-rw----- 1 mysql mysql 1680 5월 10 13:10 client-key.pem
-rw-r--r-- 1 root root 0 5월 10 13:10 debian-5.7.flag
-rw-r----- 1 mysql mysql 7220 5월 10 14:20 ib_buffer_pool
-rw-r----- 1 mysql mysql 50331648 5월 10 14:20 ib_logfile0
-rw-r----- 1 mysql mysql 50331648 5월 10 13:10 ib_logfile1
-rw-r----- 1 mysql mysql 79691776 5월 10 14:20 ibdata1
drwxr-x-- 2 mysql mysql 4096 5월 10 13:10 mysql/
drwxr-x-- 2 mysql mysql 4096 5월 10 13:10 performance_schema/
-rw----- 1 mysql mysql 1676 5월 10 13:10 private_key.pem
-rw-r--r-- 1 mysql mysql 452 5월 10 13:10 public_key.pem
-rw-r--r-- 1 mysql mysql 1112 5월 10 13:10 server-cert.pem
-rw----- 1 mysql mysql 1676 5월 10 13:10 server-key.pem
drwxr-x-- 2 mysql mysql 12288 5월 10 13:10 sys/
drwxr-x-- 2 mysql mysql 20480 5월 10 14:17 zabbix/

# tar cvfz /root/db.tar.gz ./
( 디렉토리에 내용 모두 묶기 )

# cd /root/
# tar tvf db.tar.gz
( 정상적으로 잘 묶였는지 확인 )

# mkdir /data
( 위의 mysql.cnf에 datadir이 /data 이라고 정하고 진행 )
# cd /data

# ls -dl /var/lib/mysql
drwx----- 6 mysql mysql 4096 5월 10 14:20 /var/lib/mysql
( 디렉토리 권한 확인 후 새로 생성한 /data 디렉토리도 동일하게 권한 및 소유자 변경 )

# chown mysql:mysql /data
# chmod 700 /data
( 권한을 바꿀 필요가 있을때만 진행 )

# tar xvfz /root/db.tar.gz -C .
( /data 경로인지 확인 후 해당 경로에 압축 해제 )

* mysql 서비스 실행이 정상적으로 안될때 시도
-----
# vi /etc/apparmor.d/local/usr.sbin.mysql
-----
/data/ r,
/data/** rwk,
-----
( mysql에 2차인증 apparmor가 걸려있어서 추가적으로 설정하여야 mysql 실행가능 )

# systemctl start apparmor.service

```

(해도 안되면 리스타트)

```
-----

# systemctl restart mysql

# zcat /usr/share/zabbix-sql-scripts/mysql/server.sql.gz | mysql -uzabbix -p
zabbix
( Zabbix 서버의 초기 스키마 및 데이터를 포함하는 덤프파일을 가져오기 )
Password : ( 위에서 설정한 패스워드 입력 )
* 패스워드 입력후 따로 출력되는게 없으면 정상

* zcat 후 mysql 재설정
> set global log_bin_trust_function_creators = 1;

# lsof | grep /data
( 정상적으로 mysql 프로세스들이 올라와있는지 확인 )

# systemctl restart zabbix-server zabbix-agent
# systemctl enable zabbix-server zabbix-agent
* Zabbix 서버 정상적으로 안올라오면 Log 확인 및 재부팅

---
# reboot
---

# systemctl restart zabbix-server zabbix-agent
# systemctl enable apache2 mysql
# systemctl restart apache2 mysql
```

여기까지 설정 후 Zabbix 프론트엔드로 접속!

ex) [서버 IP]/zabbix

[프론트엔드 한글 활성화]

```
# sudo locale-gen ko_KR.UTF-8
# vi /etc/default/locale
-----
LANG=ko_KR.UTF-8
LANGUAGE="ko_KR:ko:en_US:en"
-----

# export LANG=ko_KR.utf8
( 서버 전역 한글로 변환 - 필요하면 진행 )
```

```
# vi /usr/share/zabbix/include/locales.inc.php
( /Korean 검색해서 display -> true 인지 확인 )

# systemctl restart apache2
```

< 프론트엔드 초기설정 >

1. 한글 설정 후 Next Step
2. 전부 다 OK 인지 확인 후 Next Step
3. type : MySQL
host : localhost
[name,user,passwd] : zabbix , zabbix , zabbix
4. Name 선택대로 입력 - ex) zabbix_server
5. Next Step - Finish
6. 초기 계정 : Admin , zabbix

<LINE or Telegram 연동>

[Telegram 연동]

-> LINE 연동은 "Zabbix Server LINE 연동.md" 참조

Telegram Bot 생성

- ```

```
1. 검색창에 BotFather 입력해서 시작  
새로운 봇 생성 -> 봇 이름 -> 봇 사용자 이름 - 끝에 bot  
ex) /newbot -> Dwzbx\_bot -> Dwzbx\_bot
  2. Dwzbx 채팅방 들어가서 /start 입력
  3. 웹 들어가서 정상여부 확인  
ex ) [https://api.telegram.org/bot\[API 키\]/getUpdates](https://api.telegram.org/bot[API 키]/getUpdates)  
> "ok":true,"result" 나오면 정상
  4. 그룹방에 생성한 bot 초대  
-> Manager Group -> Members -> Add members
  5. bot 권한 부여  
-> Administrators -> Add Administrator
- ```
-----
```

설치 및 구축 진행

```
# git clone https://github.com/ableev/Zabbix-in-Telegram.git
# mv Zabbix-in-Telegram/* /usr/lib/zabbix/alertscripts
( 텔레그램 템플릿 설치 및 경로 이동 )

# apt -y install python3-pip python3

# pip3 install requests
# pip3 install telegram
# pip3 install python-telegram-bot

# cd /usr/lib/zabbix/alertscripts
# vi chat.py
( 신규 파일 생성 )
-----
import telegram
import requests
my_token = '[api 키]'
bot = telegram.Bot(token = my_token)
updates = bot.getUpdates()
for u in updates :
    print(u.message)
-----
( chat id를 추출하기 위한 스크립트 생성 )

# python3 chat.py
{'chat': {'all_members_are_administrators': True, 'title': 'testdw', 'id':
-656840872, 'type': 'group'}}~~~~~

( 음수로 된 그룹방 id 복사 )

* module ~~ 오류 발생시
1.
# pip uninstall python-telegram-bot telegram
# pip install python-telegram-bot

2.
chat.py 파일 텍스트 오류 있는지 확인 ( print 줄 들여쓰기 , api 키 등등 )

3.
apt -y remove python python3
apt -y install python3-pip python-pip
( 파이썬 삭제후 재설치 하고 1번 다시 진행 )

# cp /usr/lib/zabbix/alertscripts/zbxtdg_settings.example.py
/usr/lib/zabbix/alertscripts/zbxtdg_settings.py
( 샘플파일 복사해서 사용 )
# chown -R zabbix.zabbix /usr/lib/zabbix
# usermod -a -G zabbix zabbix
( 사용자 권한 변경 )
```



```
# vi /usr/lib/zabbix/alertscripts/zbxtg_settings.py
-----
3 tg_key = "[api key]" # telegram bot api key
( api 키 입력 )

5 zbx_tg_prefix = "zbxtg" # variable for separating text from script info
6 zbx_tg_tmp_dir = "/" + zbx_tg_prefix # directory for saving caches, uids,
cookies, etc.
7 zbx_tg_signature = False
( uids.txt 정보가 담긴 경로 설정 )

15 zbx_server = "http://[Server IP]/zabbix/" # zabbix server full url
16 zbx_api_user = "Admin"
17 zbx_api_pass = "zabbix"
( Zabbix 프론트엔드 주소와 계정 입력 )

24 zbx_basic_auth = False
25 zbx_basic_auth_user = "Dwzbx2_bot"
26 zbx_basic_auth_pass = "zabbix"
( 텔레그램 bot 이름과 패스워드 지정 )

42 zbx_tg_daemon_enabled = False
43 zbx_tg_daemon_enabled_ids = [-656840872, ]
44 zbx_tg_daemon_enabled_users = ["Dwzbx2_bot", ]
45 zbx_tg_daemon_enabled_chats = ["Zabbix in Telegram Script", ]
( 추출해놓은 chat id 와 bot 사용자 명 입력 )

47 zbx_db_host = "localhost"
48 zbx_db_database = "zabbix"
49 zbx_db_user = "zabbix"
50 zbx_db_password = "zabbix"
( Zabbix DB 정보 입력 )
-----
```

텔레그램 zabbix 웹 설정

1.

관리 -> 미디어 타입 -> 텔레그램 선택

- 연락방법

```
종류 : 스크립트
스크립트 이름 : zbxtg.py
스크립트 파라미터 : {ALERT.SENDTO}
                   {ALERT.SUBJECT}
                   {ALERT.MESSAGE}
```

2.

관리 -> 유저

admin 클릭 (앞서 선택한 계정)

- 연락방법

추가

종류 : Telegram

수신처 : Testdw_bot [bot 이름]

추가

TEST

1.

python3 zbxtg.py Dwzbx2_bot Test Hihihhi

(/usr/lib/zabbix/alertscripts 경로에서 테스트 진행)

-> 정상적으로 실행되면 설정한 텔레그램 채팅방에 bot이 채팅 전송

* send ~~ 오류 발생 시

cd /zbxtg

vi uids.txt

[bot id];private;[chat id];-----
입력 후 저장

2.

관리 -> 미디어타입

telegram - 테스트

수신처 : 텔레그램 계정명

입력해서 테스트

-> 정상적으로 알림 발송되는지 확인

3.

서버에서 stress로 부하도 테스트

stress -c 2

(stress 서비스 없으면 설치 후 진행)

-> 정상적으로 알림 발송되는지 확인

[LINE 연동]

```
# vi /usr/lib/zabbix/alertscripts/zbxln.sh
( alertscripts 밑에 새로운 쉘 스크립트 생성 )
-----

#!/bin/bash
# LINE Notify Token - Media > "Send to".
TOKEN="$1"
# {ALERT.SUBJECT}
subject="$2"
# {ALERT.MESSAGE}
message="$3"
# Line Notify notice message.
notice="
${subject}
${message}
"

curl -X POST -H "Authorization: Bearer ${TOKEN}" -F "message=${notice}"
https://notify-api.line.me/api/notify
-----

# chmod +x zbxln.sh
( 실행 권한 추가 )

# vi /etc/zabbix/zabbix_server.conf
-----

523 AlertScriptsPath=/usr/lib/zabbix/alertscripts
( 파일 경로와 일치하는지 확인 )
-----

+
-----
그룹이 없다면 -> 그룹 생성 및 토큰 발급
-----

1. 라인 접속 및 그룹 만들기

2. Line Notify를 멤버로 추가 ( 메세지 송신자 )

3. https://notify-bot.line.me/en/ 들어가서 로그인 후 마이페이지 -> 하단 Generate
Token -> 토큰 네임 기입 + 앞서 생성한 그룹 클릭

4. 토큰 생성 완료 ( 따로 복사해서 저장 )
-----
```

Zabbix 홈페이지 설정

1. 미디어타입 가져오기

- xml 파일을 통해 작성

관리 -> 미디어타입

가져오기로 zbx_Line_mediatypes_5.0.xml (미디어 타입 설정 파일) 가져오기

Zabbix Line 연동 파일!

- 문제 발생시 수동 작성

- 연락 방법

이름 : Line Notify_shell
 종류 : 스크립트
 스크립트 이름 : zbxln.sh (전에 만든 스크립트 파일명과 동일)
 스크립트 파라미터 : {ALERT.SENDTO}
 {ALERT.SUBJECT}
 {ALERT.MESSAGE}

- Message templates

장애 , Problem recovery , Problem update , 디스커버리 , Autoregistration 추가

2. 액션 작성

설정 -> 액션 -> 왼쪽 위 Trigger actions -> 액션작성

- 액션

이름 : Line Action (조건 X , 모든 액션에 대해서)

- 오퍼레이션

오퍼레이션 : 사용자에게 메시지를 송신 : Admin (액션 발동 시 어디로 어떤 행동을 취할건지 선택)

복구시, 갱신시 = 필요시에 추가

3. 토큰 연결

관리 -> 유저

- Admin 클릭 -> 추가
- 연락방법

종류 : Line Notify_shell
수신처 : Line 토큰 입력

TEST

1.

관리 -> 미디어타입

Line Notify_shell 테스트 클릭해서 생성 한 그룹으로 메세지 잘 오는지 확인

2.

Zabbix Server에서

```
# cd /usr/lib/zabbix/alertscripts/

# ./zbxln.sh [토큰값] $SUBJECT $MESSAGE
( ./zbxln.sh 라인 토큰값, 제목, 메세지 )
```

-> 그룹으로 메세지 잘 오는지 확인

[Zabbix 그래프 한글패치]

- 먼저 글꼴 하나 다운로드 받기 (ex 나눔글꼴 - NanumGothic.ttf)

```
# cd /usr/share/zabbix/assets/fonts/

# mv graphfont.ttf graphfont.ttf.bak
( 기존 글꼴파일 bak으로 이름 변경 )

# mv NanumGothic.ttf /usr/share/zabbix/assets/fonts/graphfont.ttf
( 다운받은 나눔글꼴 파일을 기존 파일 이름으로 변경하여 저장 )
```

- 여기까지 설정 후 자빅스 웹에서 그래프 들어가서 한글 잘 나오는지 확인

[Zabbix 자동호스트 등록]

[Zabbix 웹]

설정 -> 액션 -> 원 상단 Autoregistration actions

- 액션 작성

이름 : Linux auto ADD

조건 : 호스트 메타데이터 , 포함 , Linux

- 오퍼레이션

종류 : 호스트 추가, 호스트그룹 추가, 템플릿과 링크 작성, 호스트 활성화 등

호스트 추가

호스트그룹을 추가: Linux servers

템플릿과 링크: Template OS Linux by Zabbix agent

호스트를 활성화

[Zabbix Agent 추가]

* Agent 서버로 접속

1. yum / apt 설치

apt -y install zabbix-agent

vi /etc/zabbix/zabbix_agentd.conf

117 Server=[Zabbix Server IP]
158 ServerActive=[Zabbix Server IP]
169 #Hostname=Zabbix server
177 HostnameItem=system.hostname
188 HostMetadata=Ubuntu_Linux_1

(Server에 Zabbix server IP 입력 , Hostname 주석 후 HostnameItem 주석 해제
HostMetadata에 Linux 포함하여 입력)

systemctl restart zabbix-agent

2. 패키지 설치 (rpm / deb 등)

* 패키지 파일 업로드 후 진행

rpm -Uvh zabbix-agent-6.0.7-1.el7.x86_64.rpm

* 해당 패키지 설치 중 아래 의존성 문제 발생 시

error: Failed dependencies:

```
libpcre2-8.so.0()(64bit) is needed by zabbix-agent-6.0.7-1.el7.x86_64  
-> yum -y install pcre2-devel (설치)
```

이후 위와 동일하게 conf 설정 및 서비스 리스타트 진행

TEST

1. 자동호스트 등록으로 Agent 호스트 잘 올라오는지 확인 (ZBX 초록색)

-> 잘 안올라오면 포트 올라와있는지, 서비스 잘 작동중인지, 방화벽 뚫려있는지 등등 확인

★ 정상적으로 작동된다면 신규 Zabbix 전체 구성 완료 ! ★

참조

- <https://nirsa.tistory.com/274>
- <https://jjeongil.tistory.com/1433>
- [https://www.zabbix.com/download?](https://www.zabbix.com/download?zabbix=6.0&os_distribution=ubuntu&os_version=18.04&components=server_frontend_agent&db=mysql&ws=apache)
zabbix=6.0&os_distribution=ubuntu&os_version=18.04&components=server_frontend_agent&db=mysql&ws=apache (Zabbix 공식 설치 홈페이지)