

Custom DevOps Monitoring System in MelOn

(InfluxDB + Telegraf+ Grafana)

발표자 소개

유승민 (Seungmin yu)

Neowiz Games (2006 ~ 2012)

Publishing Game DBA



발표자 소개

유승민 (Seungmin yu)

Loen Entertainment (2013 ~)

MelOn Business division IT Infra Team

DBA / DA

smyoo@iloen.com





Smyoo0316@gmail.com

MelOn



Custom DevOps Monitoring System?

InfluxData 社 에서 정의한 Use Cases
(with InfluxDB + Telegraf + Chronograf + kapacitor)

IoT and Sensor Data	Custom DevOps Monitoring	Real-Time Analytics	Cloud Monitoring
			
<p>InfluxData is for IoT deployments requiring support for thousands of sensors. Collect, store, visualize and alert on time-series data emitted from ARM, Arduino, Raspberry Pi and more. Learn More...</p>	<p>InfluxData is the alerting, visualization and backend of choice for building custom monitoring solutions for servers, sensors, storage appliances, network infrastructure, apps, logs and more. Learn More...</p>	<p>InfluxData enables you to deliver real-time analytics about users, sensor states, business metrics, plus the utilization of apps, infrastructure and networks with tools that are integrated from end-to-end. Learn More...</p>	<p>Service providers and Enterprises looking for a time-series data management stack to monitor their virtualization, container, OpenStack and other IaaS deployments rely on InfluxData. Learn More...</p>

출처 : <https://www.influxdata.com/>

What is DevOps?

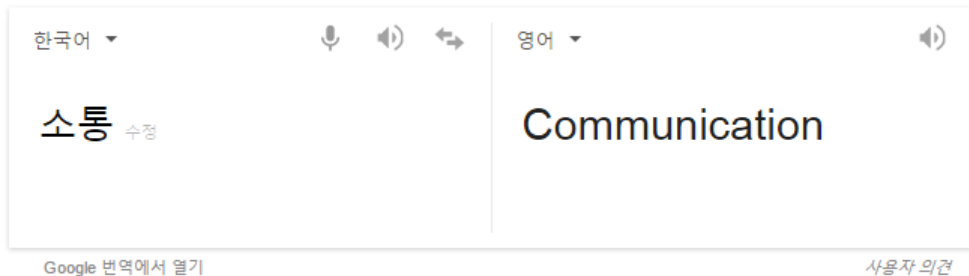
Development

Operations

What is DevOps?

DevOps is a culture, movement or practice that emphasizes the **collaboration** and **communication** of both **software developers** and **other IT professionals** while automating the process of software delivery and infrastructure changes.

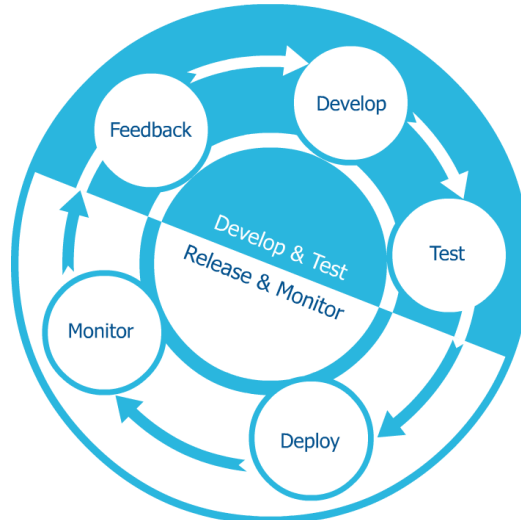
<From Wikipedia>



What is DevOps?

It aims at establishing a culture and environment where building, testing, and releasing software can happen **rapidly**, **frequently**, and more **reliably**

<From Wikipedia>



What is DevOps?

Dev : Developer (변화)

Ops : SE, DBA (안정성)

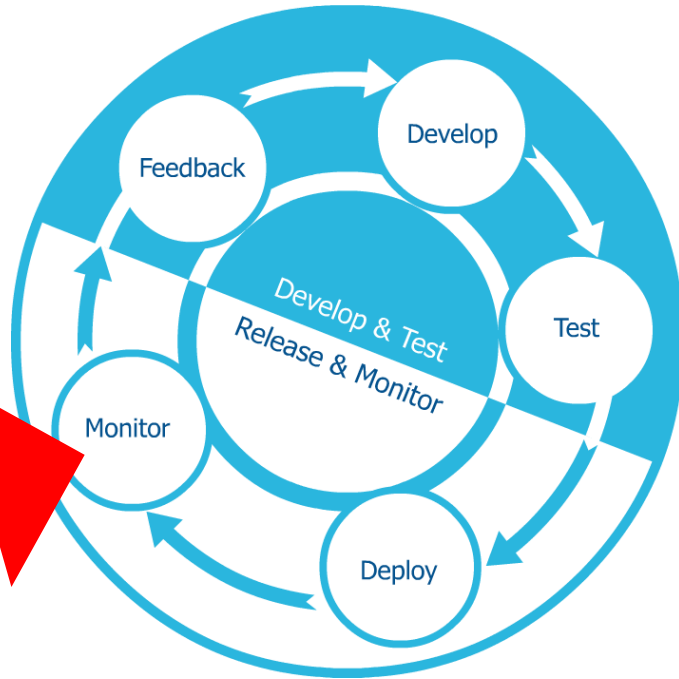
“동료의식, 소통, 공유, 그리고 Tools”

DevOps 스텝다 (주관적 정의)

- ✓ DevOps스러운 사람 : 소통과 협력이 잘되는 사람
- ✓ DevOps스러운 툴 : 소통과 협력하는데 유용한 툴

“DevOps스러운 Monitoring System은 무엇인가.”

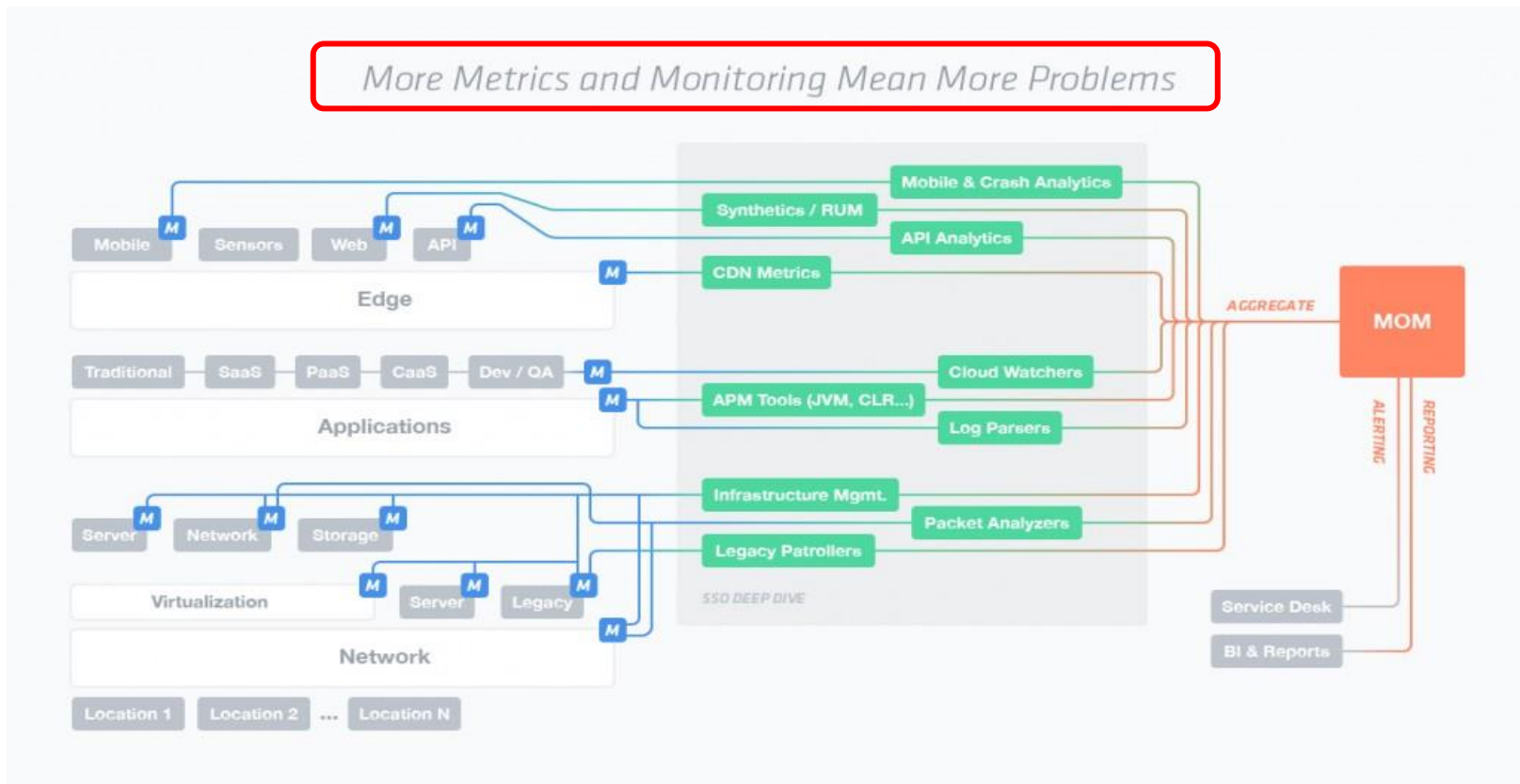
DevOps스러운 Monitoring System



Rapidly, Frequently, Reliably
하게 서비스함에 있어서 유용한가.

신속한 정확한 모니터링 → 빠른 개선점 도출 및 안정성 확보

보고자 하는 것들. 그리고 현실적인 문제.



현실적인 문제 (Rapidly, Frequently, Reliably 하기 위해 해결해야 할 문제들)

저장

'보고 싶은 METRIC 수 X 서버 수' 데이터가 많다.
고성능 Write & Query, 저장공간 효율이 중요하다.

수집

수집 대상이 다양하다.
쉽게 수집 가능해야 한다.

시각화

쉬운 사용성 및 Customizability가 중요하다.
보고 싶은 View만 효율적으로 볼 수 있어야 한다.

현실적인 문제 - 세상이 좋아 졌다

Custom DevOps Monitoring System in MelOn

저장



수집

Telegraf

시각화



여기서 잠깐, 비슷하지만 용도가 다른 것 (ELK Stack)

#METRIC



Telegraf



VS

#LOG



InfluxDB (저장) + Telegraf(수집) + Grafana (시각화)

Easy
Powerful
Beautiful

Easy Installation

InfluxDB Installation

```
$ sudo yum install https://s3.amazonaws.com/influxdb/influxdb-1.0.1.x86\_64.rpm
```

```
$ sudo service influxdb start
```

Telegraf Installation

```
$ sudo yum install https://dl.influxdata.com/telegraf/releases/telegraf-1.0.1.x86\_64.rpm
```

```
$ .. 약간의 설정 ..
```

```
$ sudo service telegraf start
```

Grafana Installation

```
$ sudo yum install https://grafanarel.s3.amazonaws.com/builds/grafana-3.1.1-1470047149.x86\_64.rpm
```

```
$ sudo service grafana-server start
```


InfluxDB - 주요 특징

High Performance write and query metric

Written entirely in Go (no external dependencies)

SQL-like query language

Retention policies

Continuous queries (downsampling)

High availability

Web admin

InfluxDB – Simple Data Insert

DATABASE

MEASUREMENT → table

TAG → Indexed column

FIELD → Not indexed column

```
$ curl -i -XPOST 'http://localhost:8086/write?db= DATABASE_NAME' --data-binary  
'MEASUREMENT_NAME,TAG_NAME1=TAG_VALUE1,TAG_NAME2=TAG_VALUE2 (공백)  
FIELD_NAME1=FIELD_VALUE1,FIELD_NAME2=FIELD_VALUE2'
```

입력 예시

```
curl -i -XPOST 'http://localhost:8086/write?db=testdb' --data-binary 'aztalk,chnl_name=아이유  
₩ 생중계₩ 채널 connection_cnt=213444,heart_cnt=123,msg_cnt=84759'
```

InfluxDB – SQL-like query language (InfluxQL)

```
[root@testdb ~]$ influx
Visit https://enterprise.influxdata.com to register for updates, InfluxDB server management, and monitoring.
Connected to http://localhost:8086 version 1.0.0
InfluxDB shell version: 1.0.0
>
> show databases
name: databases
-----
name
_internal
testdb

> use testdb
> show measurements
name: measurements
-----
name
aztalk

>
> select * from aztalk;
name: aztalk
-----
time                chnl_name                connect ion_cnt          hear t_cnt              msg_cnt
1462256721282477897  아이유 생중계 채널       12341                    123                     84759
1462256726539652308  아이유 생중계 채널       12341                    123                     84759
1462256727816872582  아이유 생중계 채널       12341                    123                     84759
1462256728879818970  아이유 생중계 채널       12341                    123                     84759
1462256729993665031  아이유 생중계 채널       12341                    123                     84759
1462256731026852953  아이유 생중계 채널       12341                    123                     84759
1462257207899082445  아이유 생중계 채널       123463                   123                     84759
```

InfluxDB – 데이터 보관 주기 (Retention Policy)

```
> SHOW DATABASES
name: databases
-----
name
_internal
mon
oracle
mysql
memcached

> CREATE RETENTION POLICY realtime ON oracle DURATION 30d REPLICATION 1 DEFAULT
> CREATE RETENTION POLICY Retention_1y_ds10s ON oracle DURATION 365d REPLICATION 1
> CREATE RETENTION POLICY Retention_5y_ds1min ON oracle DURATION 1825d REPLICATION 1
> DROP RETENTION POLICY autogen ON oracle
>
> SHOW RETENTION POLICIES ON oracle
name                duration          shardGroupDuration  replicaN  default
realtime            720h0m0s         24h0m0s             1         true
Retention_1y_ds10s 8760h0m0s        168h0m0s            1         false
Retention_5y_ds1min 43800h0m0s       168h0m0s            1         false
```

InfluxDB – Continuous Query (CQ)

```
CREATE CONTINUOUS QUERY cq_1min_mysql
ON test_database
BEGIN
  select
    mean(commands_delete) as commands_delete,
    mean(commands_insert) as commands_insert,
    mean(commands_update) as commands_update,
    mean(commands_select) as commands_select
  INTO Retention_5y_ds1min.mysql /* Retention_5y_ds1min 은 미리 정의된 Retention Policy 이름 */
  FROM realtime.mysql /* realtime 도 미리 정의된 Retention Policy 이름 */
  GROUP BY time(1m),host,"server" /* 다운샘플링 1분, TAG 는 GROUP BY 절에 이어서 붙여 줌. 예약어(server)는 쌍 따옴표로 감싸줌 */
END

# 생성된 CONTINUOUS QUERY 확인
SHOW CONTINUOUS QUERIES
```

1초 마다 체크하여 다운샘플링 결과를 따로 저장.

Background 로 자동 수행되며 체크 주기는 설정 가능. (/etc/influxdb/influxdb.conf)

Oracle database 의 MVIEW 와 유사한 개념.

InfluxDB - High Performance (for ONLY time-series data)

InfluxDB v1.0.0 VS Cassandra v3.7

5.3x write throughput

9.3x less disk space

up to 168x faster response times

InfluxDB v1.0.0 VS Elasticsearch v5.0.0-alpha5

8x write throughput

4x less disk space

4x to 10x faster response times

InfluxDB v1.0.0 VS MongoDB v3.3.11

27x write throughput

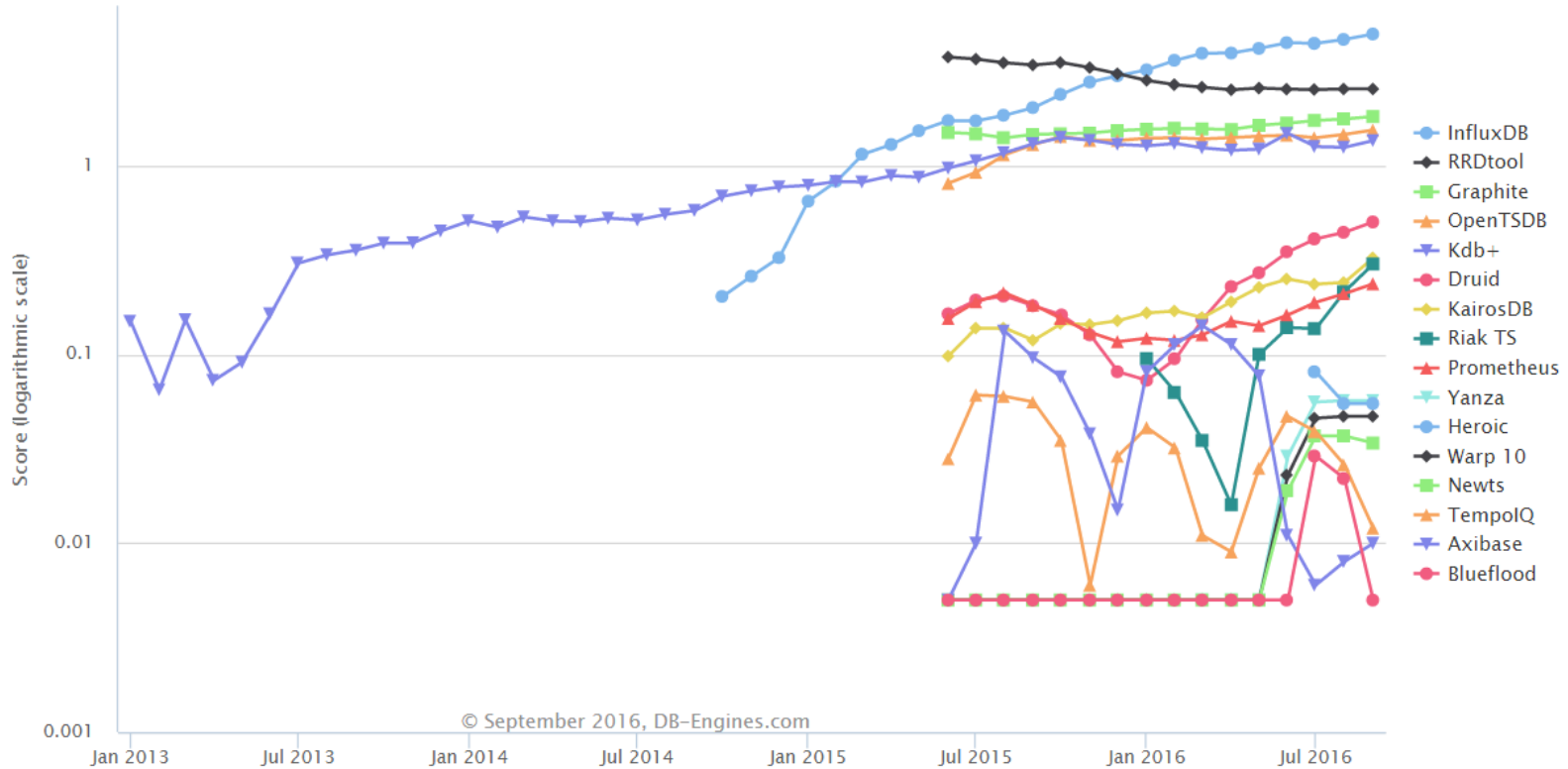
84x less disk space

same

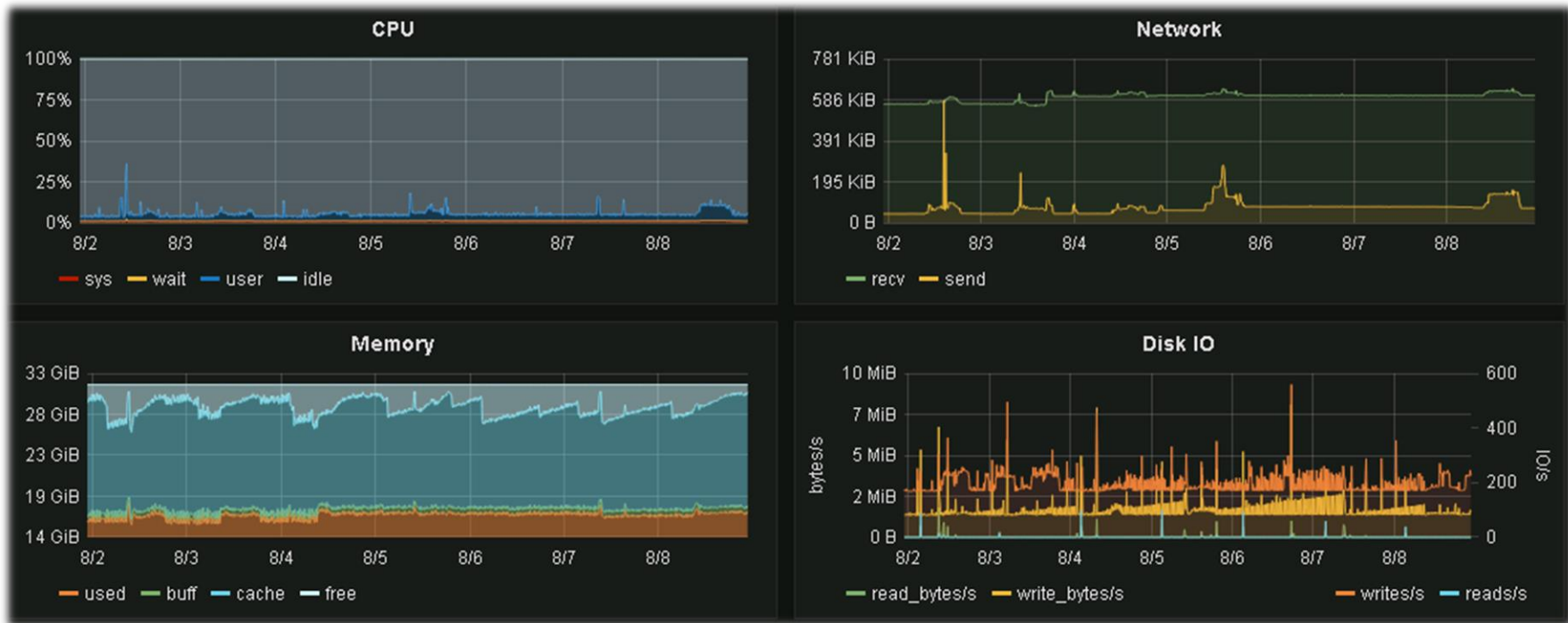
LSM Tree → TSM Tree

InfluxDB – 인기 1위 (시계열DB계의 멜론차트)

DB-Engines Ranking of Time Series DBMS



InfluxDB - 실제 운영 모니터링



<모니터링 서버(InfluxDB서버) SPEC>

- ✓ Model : HP DL360p G8
- ✓ CPU : E5-2630 @ 2.30GHz * 2 (6 core)
- ✓ MEMORY : 32G
- ✓ DISK : 300GB X 4 , 900GB X 2

Telegraf – 기본구성

```
$ cd /etc/telegraf  
$ telegraf -sample-config -input-filter cpu:disk:mem:net:mysql -output-filter influxdb >  
telegraf.conf  
$ vi /etc/telegraf/telegraf.conf
```



Telegraf – 수집 가능한 것들 (input plugins)

Linux system

Apache

Cloudwatch

Docker

Jolokia(JMX)

Kafka

Rabbitmq

Zookeeper

Mongodb

Cassandra

Couchbase

Elasticsearch

Memcached

Redis

Influxdb

Mysql

Postgresql

SQL server

- 하고 있는 것들
- 앞으로 할 것들

Telegraf – 가능한 저장소 (output plugins)

InfluxDB

Graphite

OpenTSDB

Prometheus

Elasticsearch

Kafka

File

AMQP

AWS Kinesis

AWS CloudWatch

Datadog

Graylog

Librato

MQTT

Telegraf – 처리 성능

```
2016/10/03 22:37:17 Output [influxdb] wrote batch of 72 metrics in 12.855776ms
2016/10/03 22:37:18 Output [influxdb] buffer fullness: 72 / 10000 metrics. Total gathered metrics: 3503. Total dropped metrics: 0.
2016/10/03 22:37:18 Output [influxdb] wrote batch of 72 metrics in 9.954726ms
2016/10/03 22:37:19 Output [influxdb] buffer fullness: 72 / 10000 metrics. Total gathered metrics: 3575. Total dropped metrics: 0.
2016/10/03 22:37:19 Output [influxdb] wrote batch of 72 metrics in 12.886334ms
2016/10/03 22:37:20 Output [influxdb] buffer fullness: 72 / 10000 metrics. Total gathered metrics: 3647. Total dropped metrics: 0.
2016/10/03 22:37:20 Output [influxdb] wrote batch of 72 metrics in 11.791857ms
2016/10/03 22:37:21 Output [influxdb] buffer fullness: 72 / 10000 metrics. Total gathered metrics: 3719. Total dropped metrics: 0.
2016/10/03 22:37:21 Output [influxdb] wrote batch of 72 metrics in 13.667747ms
2016/10/03 22:37:22 Output [influxdb] buffer fullness: 72 / 10000 metrics. Total gathered metrics: 3791. Total dropped metrics: 0.
2016/10/03 22:37:22 Output [influxdb] wrote batch of 72 metrics in 12.174984ms
2016/10/03 22:37:23 Output [influxdb] buffer fullness: 72 / 10000 metrics. Total gathered metrics: 3863. Total dropped metrics: 0.
2016/10/03 22:37:23 Output [influxdb] wrote batch of 72 metrics in 13.066251ms
2016/10/03 22:37:24 Output [influxdb] buffer fullness: 72 / 10000 metrics. Total gathered metrics: 3935. Total dropped metrics: 0.
2016/10/03 22:37:24 Output [influxdb] wrote batch of 72 metrics in 13.484553ms
```

- ✓ 1초 마다 수집 및 InfluxDB로 전송
- ✓ 수집 대상 : cpu, disk, kernel, mem, net, netstat, system, mysql(all global status, slave status)

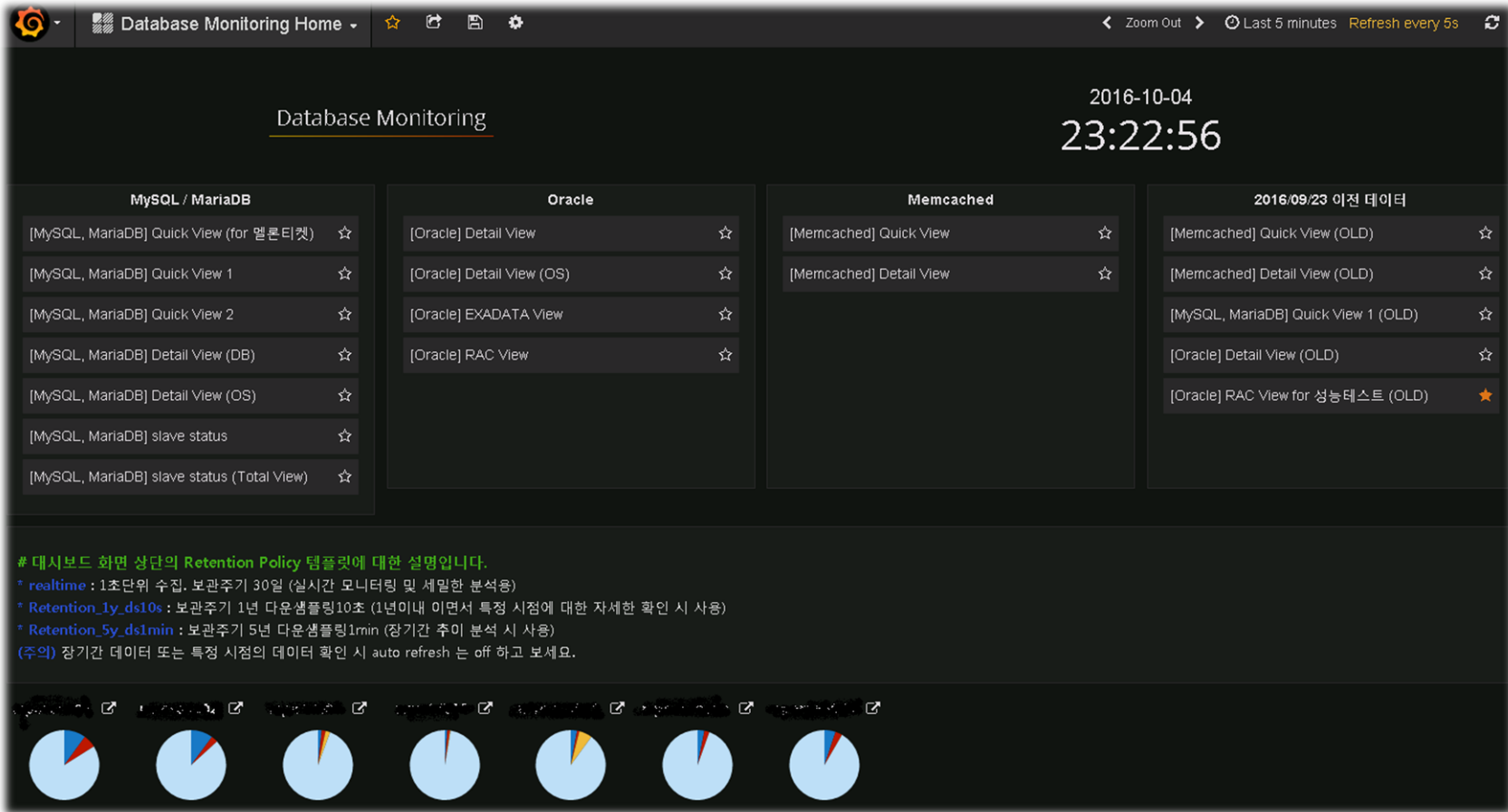
수집 - 처리 성능



✓ 1초 마다 수집 및 InfluxDB로 전송

✓ 수집 대상 : cpu, disk, kernel, mem, net, netstat, system, mysql(all global status, slave status)

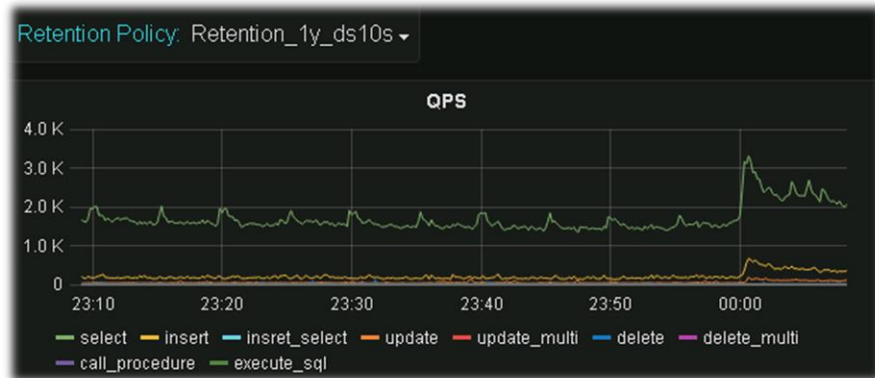
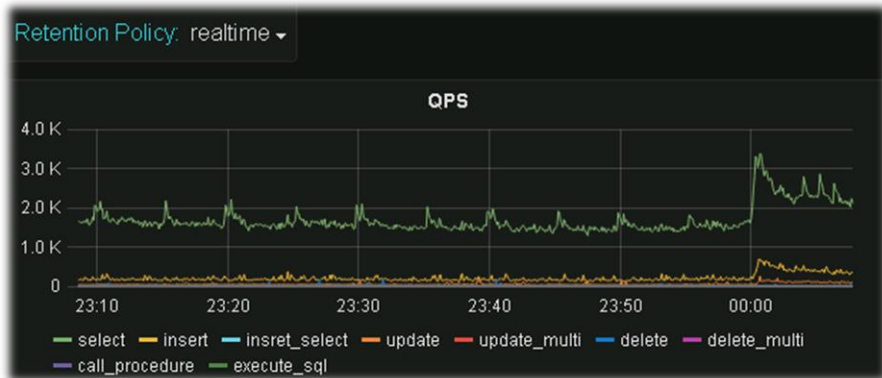
Grafana – DB Monitoring Main in MelOn



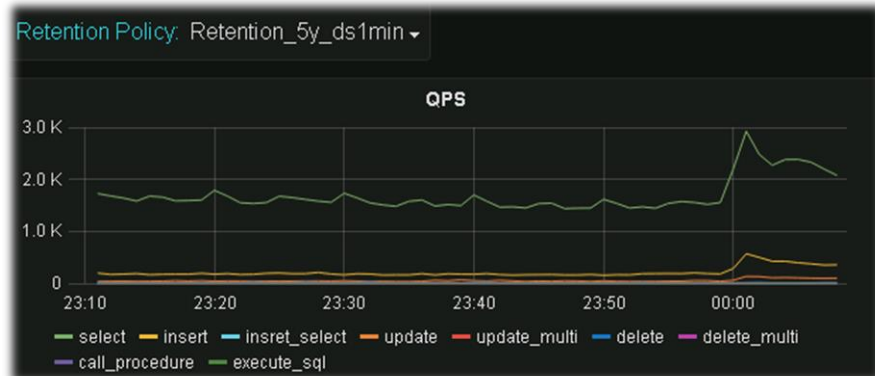
Grafana – 다양한 Panel들



Grafana — Template 기능과 InfluxDB의 RP & CQ 의 조화

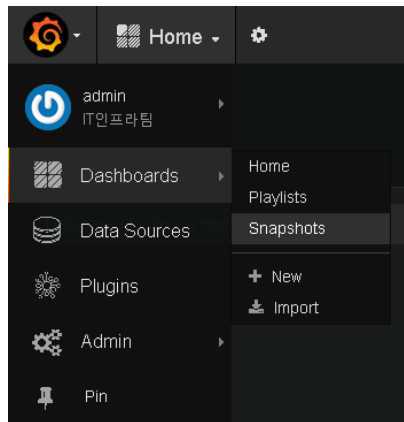


- ✓ **realtime** : 1초 간격 수집
- ✓ **Retention_1y_ds10s** : realtime으로부터 downsampling 10초
- ✓ **Retention_5y_ds1min** : realtime으로부터 downsampling 1분









Grafana – Snapshot

특정 시점 또는 성능테스트 측정 결과를 스냅샷으로 저장 후 필요한 시점에 참고.



Available snapshots

Name	Snapshot url		
2016/09/06 멜론티켓 박효신 팬클럽 선예매 12,000명 (네티퍼넬 5,000)	dashboard/snapshot/sHrp63rmPzEvBQAhicqY0hudjBZ4TZ1q		
2016/09/07 멜론티켓 Asia Song Festival (네티퍼넬 5,000)	dashboard/snapshot/ScnO5EQw3Y5JdEp1Sz0HM8vPUhe75hlf		
2016/09/08 멜론티켓 박효신 잔여석 일반예매 (네티퍼넬 5,000)	dashboard/snapshot/BsJ6e3DkTyUAJ0v9dTkVM0AW9hoINDcj		

Use Case in MelOn (mysql)



궁금한 것이 있으면 일단 그려 본다.

서버마다 slave 지연 정도가 다른 이유를 innodb buffer pool 과의 상관관계를 통해 확인해보는 예시.

Use Case in MelOn (memcached)



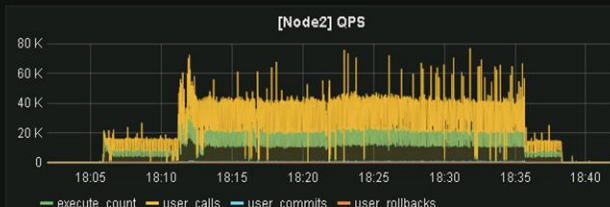
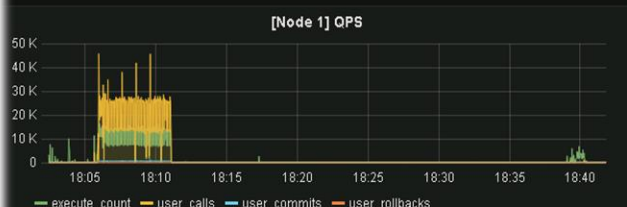
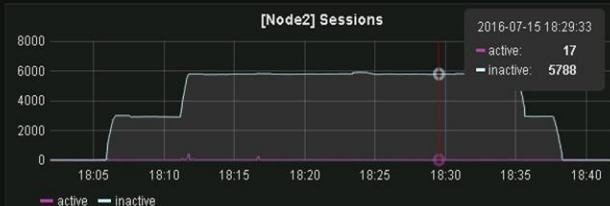
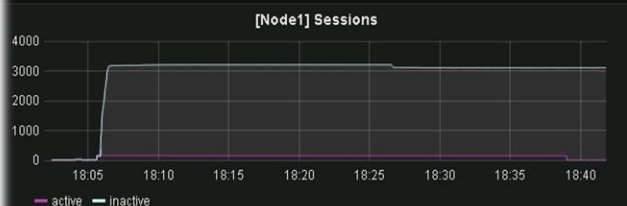
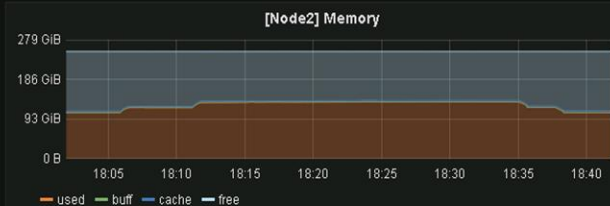
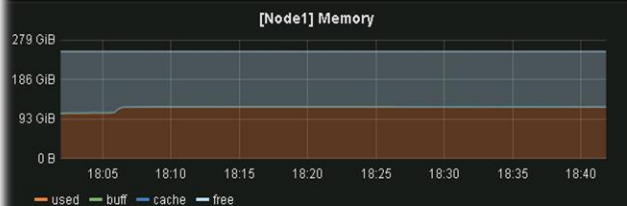
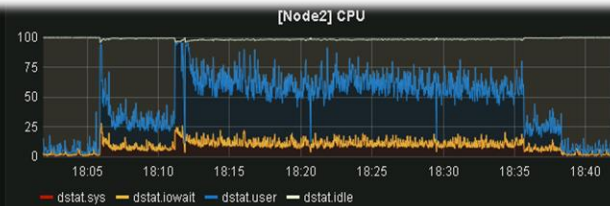
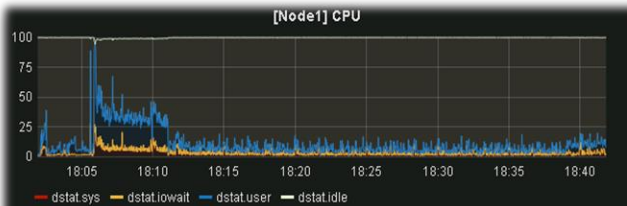
잘 모르는 것도 일단 수집하고 그려본다.

트래픽이 집중되는 특정 시점에 memcached의 문제가 의심되는 상황.

Telegraf 에 memcached plugin 추가하여 일단 모니터링 데이터 수집하고 대시보드 생성. 다음 재현 상황에서 원인 발견되어 조치. (conn_yields 튜닝)

Memcached의 주요 모니터링 지표에 대해서 공부하게 됨.

Use Case in MelOn (Oracle, Exadata)



성능테스트

DB Replay로 실서비스 환경의 DB트래픽 캡처 후 신규서버 2대(RAC)에 재생하여 결과 측정.

양쪽 노드에 부하 재생하다가 1번 노드 shutdown 했을 시 fail-over 확인.

세션 수, SQL트래픽이 실서비스와 동일한지 확인.

Fail-over시 유실되는 DB트래픽이 없는지 확인.

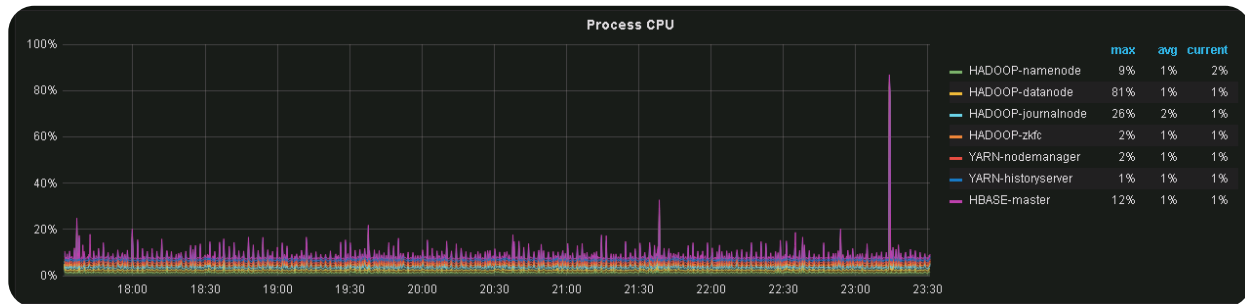
실서비스와 SQL트래픽이 동일하지 않을 경우 원인 파악하여 해결. (네트워크이슈, DB이슈 등등)

SQL트래픽이 실서비스와 거의 동일하게 문제없이 실행되는 상황에서 리소스 사용량 측정 및 비교.

(참고) Telegraf에서 oracle database plugin은 지원하지 않습니다. 자체적으로 수집 agent를 제작하였습니다.

(v\$sysstat 및 일부 정보 초단위로 수집)

Use Case in MelOn (Bigdata)

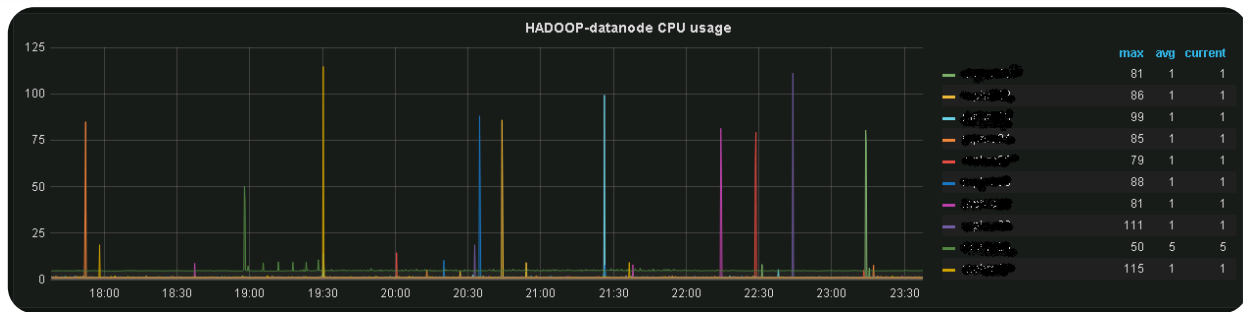


적용 중

Telegraf 의 procstat plugin으로 프로세스별 서버 리소스 사용량 메트릭 수집.

테스트 중

jolokia plugin 으로 GC모니터링 예정.
zookeeper, kafka plugin도 활용 예정.



Use Case in MelOn (그외 기타)

그 외 활용하고 있는 것

어플리케이션 성능 모니터링 (Jennifer) (by. Dev)

서비스 보안 모니터링 (by. Dev, Ops, 보안담당자)

사내 보안 시스템 관제 (by. 보안담당자)

실시간 생중계 동시접속자 (by. Dev)

앞으로 할 것들

Bigdata Infra 모니터링

Jolokia (JMX)

Kafka

Elasticsearch

Zookeeper

어플리케이션 성능 모니터링(Scouter)

등등...

그래서 좋은점

수치에 기반한 명확한 커뮤니케이션으로 신속한 대응

직관에 의존한 경솔한 의심(시스템, 사람) → 시간 낭비 발생 (덤으로 동료의 기분 손상 그로 인한 팀웍 손상)

"측정할 수 없으면 관리할 수 없다. (by 피터드러커)" → 가능한 수집하고 언제든지 모니터링(측정) 할 수 있는 상태를 유지

개발자, DBA, SE 때로는 기획자까지 모두 하나의 대시보드로..

서비스 채널 별로 동접자 수, 채팅 수 그에 따른 관련 시스템 자원 사용량 [같이 보고 싶어요.](#)

PUSH발송 시 처리량, 응답률, 랜딩 페이지와 관련된 WAS 및 DB 사용량을 [실시간으로 보고 싶어요.](#)

비정상 사용자 모니터링을 위해 개발자, 보안담당자, DBA, SE 모두의 데이터가 필요한데 [하나의 VIEW로 보고 싶어요](#)

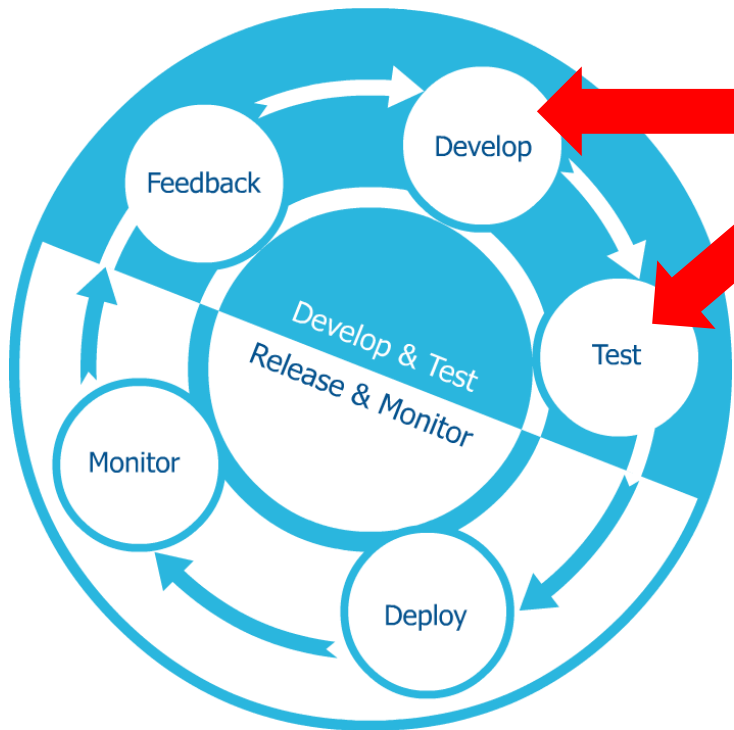
DevOps in MelOn?

모든 노력에는 성공의 가능성을 극적으로 높여주는 제1원칙이 존재한다. (마셜골드스미스 저, 트리거(trigger) 중에서)

- ✓ **목수일** : 두 번 **측정하고** 한 번에 잘라라.
- ✓ **항해** : 바람이 어디서 불어오는지를 **알아라**.
- ✓ **여성패션** : 검은색 드레스를 **사라**.
- ✓ **멜론IT본부** : 최고가 되기 위해 서로 **돕는다**. (오글 주의)



그리고 DevOps스러운 또 다른 툴 (검토 중)



TadpoleHub
Anywhere Database Management System

What is DevOps?

Dev : Developer (변화)

Ops : SE, DBA (안정성)

“동료의식, 소통, 공유, 그리고 Tools”

참고자료

InfluxDB+Telegraf+Grafana를 이용한 Monitoring System 만들기

<http://www.popit.kr/author/smyoo/>

<https://www.facebook.com/popitkr/> (Popit 페이스북 페이지)

DevOps

<https://en.wikipedia.org/wiki/DevOps>

<http://www.slideshare.net/ssuser2f0173/devops-41731364>

InfluxData, Grafana

<https://www.influxdata.com/technical-papers/>

<https://docs.influxdata.com/influxdb/v1.0/>

<https://docs.influxdata.com/telegraf/v1.0/>

<http://grafana.org/>

감사합니다