Kafka + ELK 로 글로벌 서비스 쿼리 모니터링 하기

삼성SDS 현충헌

순서

1. 왜 하게 되었는가

2. 어떻게 만들었는가

3. 잘 쓰고 있는가

왜 하게 되었는가

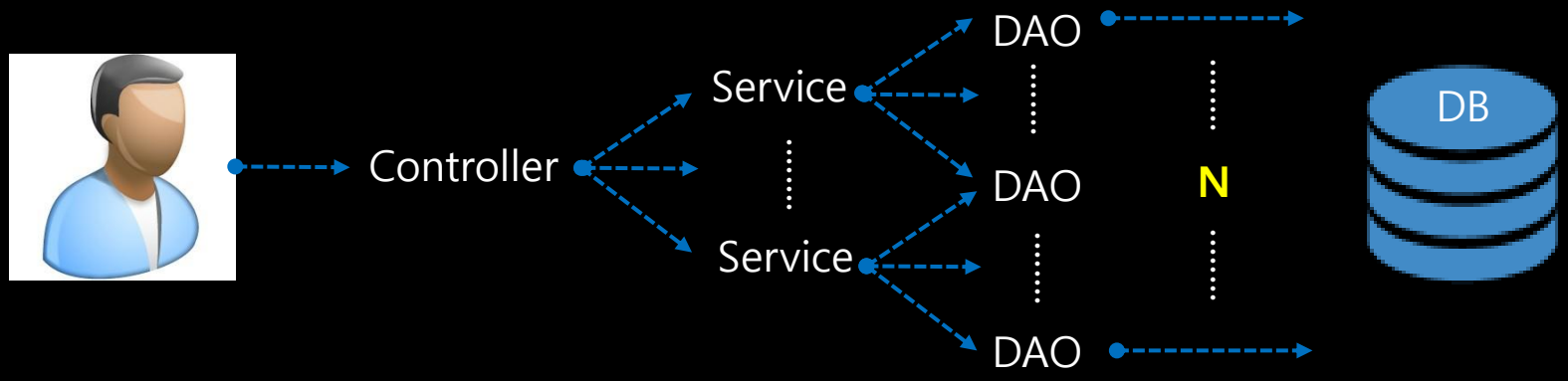
사용자 만족도 조사

시스템이 느리다

대체 어느 부분이 느린가?

Appl. 의 정량적 수치를 확인하자

어떤 수치를 수집할 것 인가?



사용자 → 1개 서비스 호출 → N개 쿼리 수행

느리다 → N개 쿼리의 수행시간 총 합이 크다

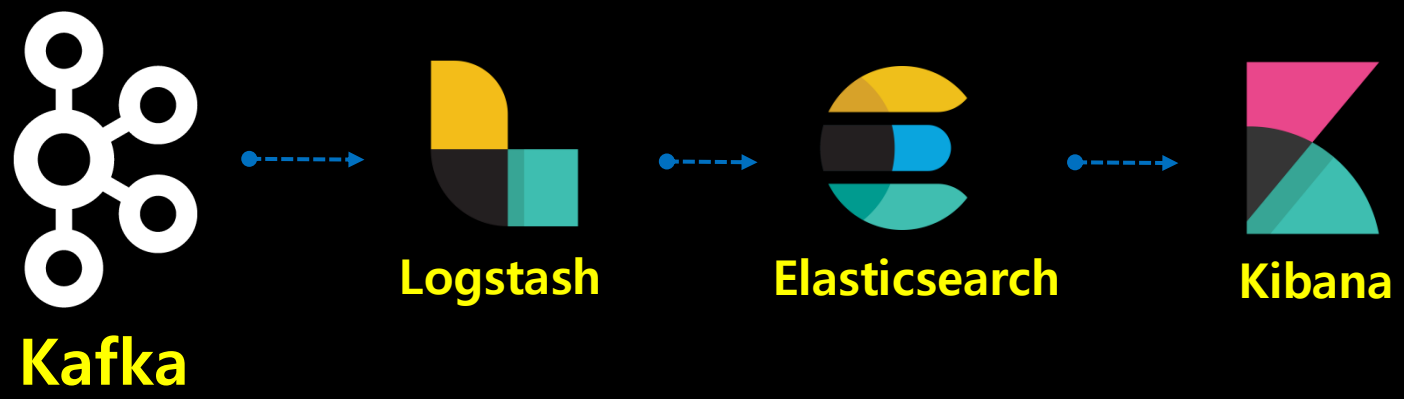
쿼리 수행시간 확인

크기 : 쿼리 + java 로그 = N GB / 시간

검색 : 보고 싶은 것만 보기 어려움

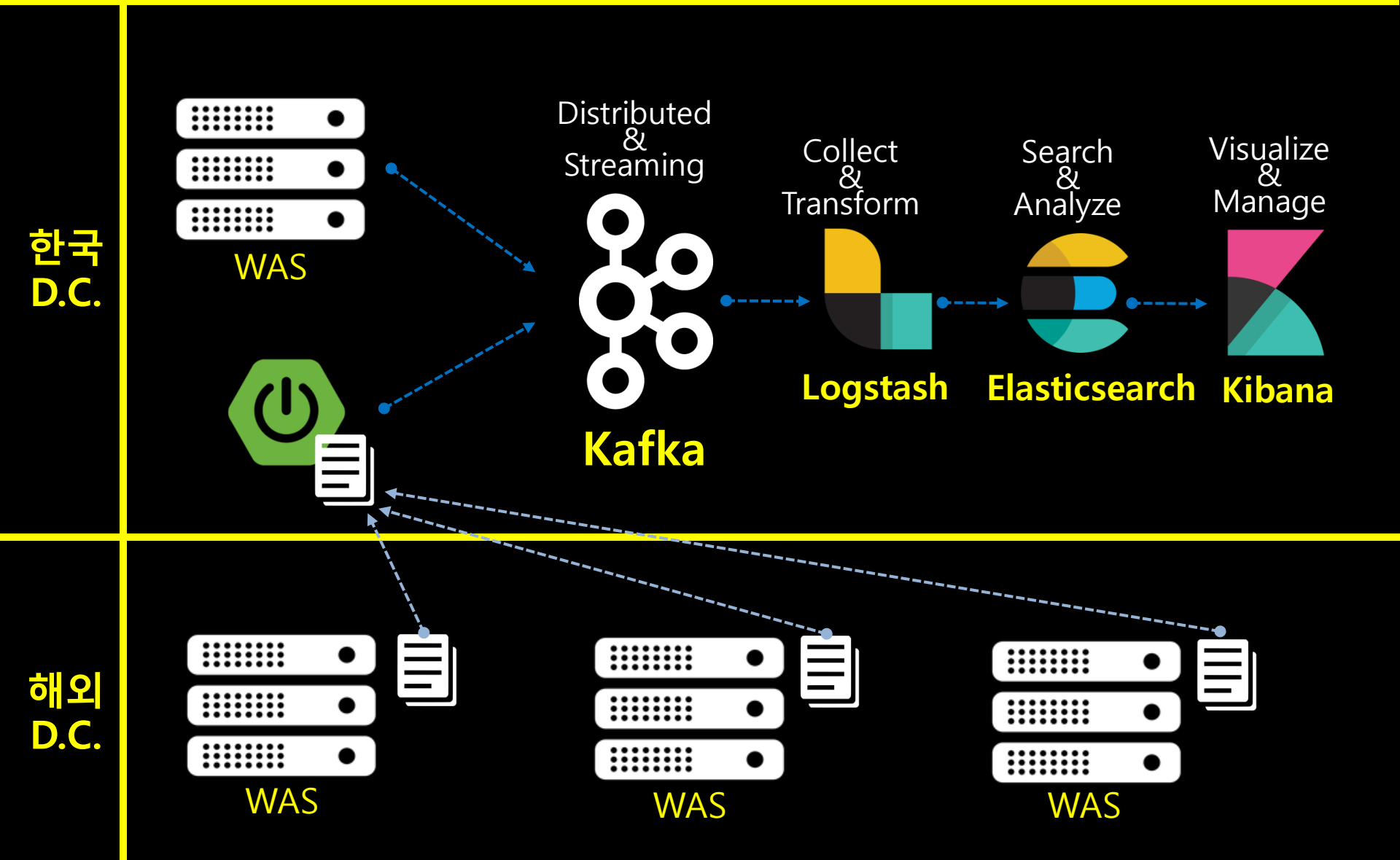
산재 : 다중화

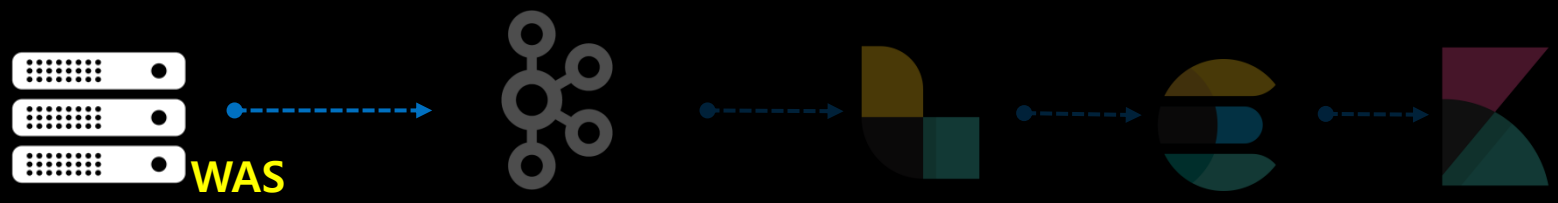
새로운 방법이 필요하다



어떻게 만들었는가

구성


한국
D.C.
WAS
Distributed
&
Streaming
Kafka
Collect
&
Transform
Logstash
Search
&
Analyze
Elasticsearch
Visualize
&
Manage
Kibana
해외
D.C.
WAS
WAS
WAS



쿼리 이벤트 처리


```java
public class CustomSpringEvent extends ApplicationEvent {
    private String message;

    public CustomSpringEvent(Object source, String message) {
        super(source);
        this.message = me
    }
    public String getMess
        return message;
    }
}
```

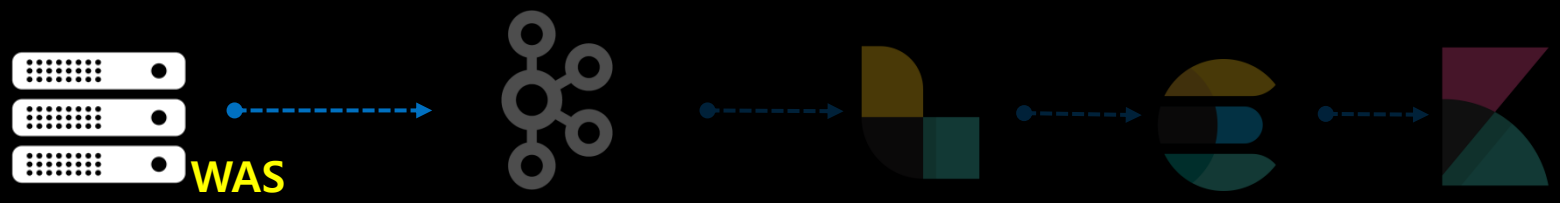
Event

Event Publisher

```java
@Component
public class CustomSpringEventPublisher {
    @Autowired
    private ApplicationEventPublisher applicationEventPublisher;

    public void doStuffAndPublishAnEvent(final String message) {
        System.out.println("Publishing custom event. ");
        CustomSpringEvent customSpringEvent = new CustomSpringEvent(this, message);
        applicationEventPublisher.publishEvent(customSpringEvent);
    }
}
```

Event Listener

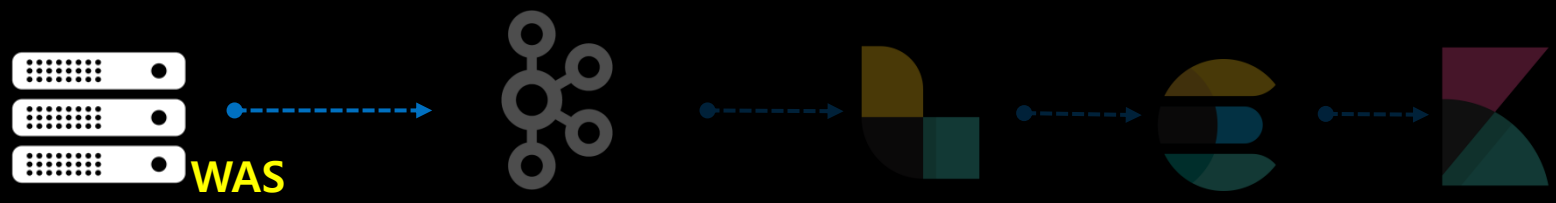
```java
@Component
public class CustomSpringEventListener implements ApplicationListener<CustomSpringEvent> {
    @Override
    public void onApplicationEvent(CustomSpringEvent event) {
        System.out.println("Received spring custom event - " + event.getMessage());
    }
}
```



쿼리 이벤트 처리 – Query Executed Event

```
public class QueryExecutedEventListenerExample implements ApplicationListener<QueryExecutedEvent>
{
    private static Logger log = LoggerFactory.getLogger(QueryExecutedEventListenerExample.class);

    @Override
    public void onApplicationEvent(QueryExecutedEvent event)
    {
        try {
            // loginId 등 필요한 정보들을 추출해서 함께 로깅할 수 있음
            log.debug(" ElapsedTime=[{}] SQL=[{}]", event.getElapsedTime(), event.getSql());

        } catch (Exception e) {
            e.printStackTrace();
        }
    }

}
```

p6spy + custom event + logback



쿼리 이벤트 처리

anyframe

https://github.com/anyframejava

Why GitHub? Enterprise Explore Marketplace Pricing Search / Sign in Sign up

Overview Repositories 44 Projects 0 Stars 0 Followers 33 Following 0

Pinned

anyframe-java-core
다양한 오픈소스를 통합 및
임워크, JDK 1.8, Eclipse Nec
JavaScript 12

anyframe-oden
CI (Continuous Integration)
포넌트 및 각종 설정파일, 웹
리하게 배포할 수 있도록 하
Java 2

anyframe-iam
사용자 인증 및 권한 관리 기능을 제공하는 Identity & Access Management Framework
Java

anyframe-monitoring
Application의 성능을 다양한 형태로 측정, 수집하여 정성적 분석을 가능하게 해주는 툴
Java

Anyframe - Open Source Framework

anyframejava

자바 기반의 어플리케이션을 개발하기 위한 기본 아키텍처, 기술 공통 서비스, 템플릿 등을 제공하는 어플리케이션 프레임워크. (문의: anyframe@samsung.com)

Samsung SDS

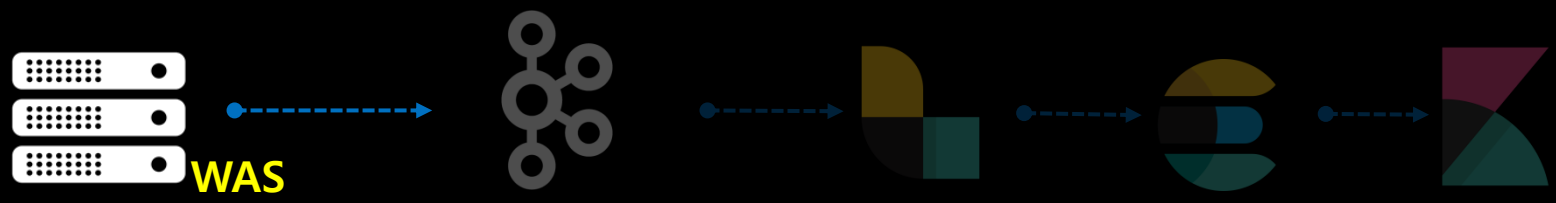
Sign in to view email

Block or report user

anyframe-jdbc-support

오픈소스 p6spy 를 확장하여 SQL Injection 보안 위험을 방어할 수 있는 기능 및 최종 실행 쿼리에 대한 로깅(재처리) 기능을 제공하는 플러그인

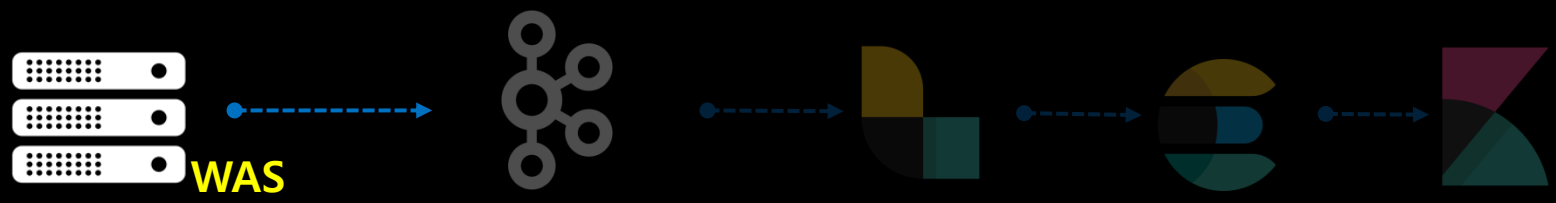
Java 1 1 Updated on 6 Apr 2018



WAS 부하 최소화 – 비동기

https://logback.qos.ch/manual/appenders.html

Property Name	Type	Description
queueSize	int	The maximum capacity of the blocking queue. By default, queueSize is set to 256.
discardingThreshold	int	By default, when the blocking queue has 20% capacity remaining, it will drop events of level TRACE, DEBUG and INFO, keeping only events of level WARN and ERROR. To keep all events, set discardingThreshold to 0.
includeCallerData	boolean	Extracting caller data can be rather expensive. To improve performance, by default, caller data associated with an event is not extracted when the event added to the event queue. By default, only "cheap" data like the thread name and the MDC are copied. You can direct this appender to include caller data by setting the includeCallerData property to true.
maxFlushTime	int	Depending on the queue depth and latency to the referenced appender, the AsyncAppender may take an unacceptable amount of time to fully flush the queue. When the LoggerContext is stopped, the AsyncAppender stop method waits up to this timeout for the worker thread to complete. Use maxFlushTime to specify a maximum queue flush timeout in milliseconds. Events that cannot be processed within this window are discarded. Semantics of this value are identical to that of Thread.join(long).
neverBlock	boolean	If false (the default) the appender will block on appending to a full queue rather than losing the message. Set to true and the appender will just drop the message and will not block your application.



Kafka 에 보내보자

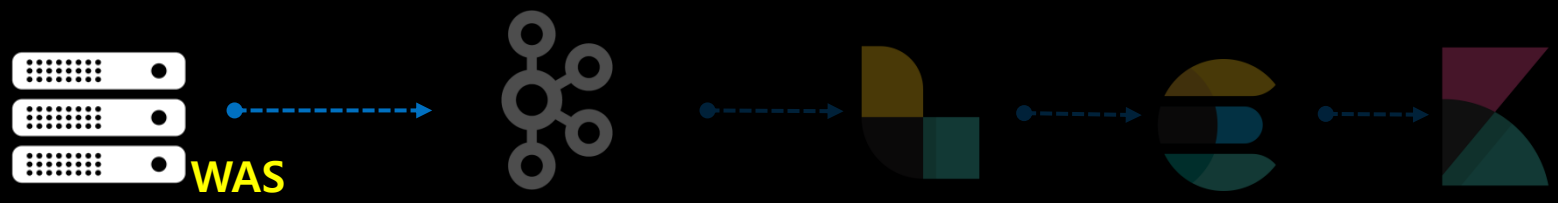
https://github.com/danielwegener/logback-kafka-appender

Note on Broker outages

The `AsynchronousDeliveryStrategy` does not prevent you from being blocked by the Kafka metadata exchange. That means: If all brokers are not reachable when the logging context starts, or all brokers become unreachable for a longer time period (> `metadata.max.age.ms`), your appender will eventually block. This behavior is undesirable in general and can be migitated with kafka-clients 0.9 (see #16). Until then you can wrap the KafkaAppender with logback's own AsyncAppender.

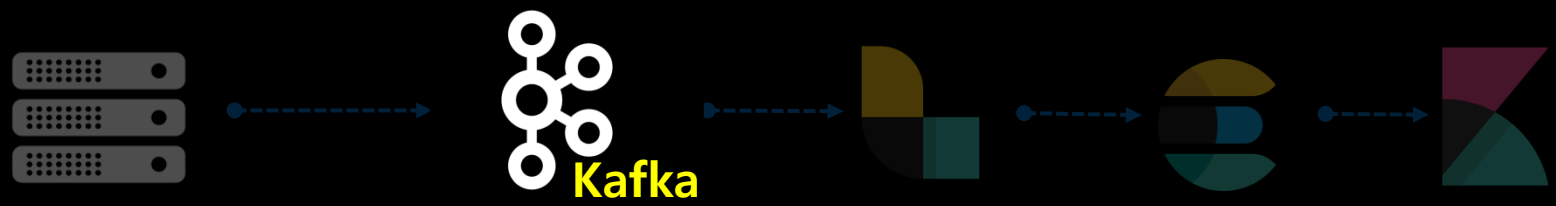
An example configuration could look like this:

```
<configuration>

    <!-- This is the kafkaAppender -->
    <appender name="kafkaAppender" class="com.github.danielwegener.logback.kafka.KafkaAppender">
    <!-- Kafka Appender configuration -->
    </appender>

    <appender name="ASYNC" class="ch.qos.logback.classic.AsyncAppender">
        <appender-ref ref="kafkaAppender" />
    </appender>

    <root level="info">
        <appender-ref ref="ASYNC" />
    </root>
</configuration>
```



비동기로 Kafka 에 보내보자

```
<appender name="asyncKafkaAppender" class="ch.qos.logback.classic.AsyncAppender">
  <queueSize>1024</queueSize>
  <maxFlushTime>3000</maxFlushTime>
  <neverBlock>true</neverBlock>
  <appender name="logstashKafkaAppender" class="com.github.danielwegener.logback.kafka.KafkaAppender">
    <encoder class="com.github.danielwegener.logback.kafka.encoding.PatternLayoutKafkaMessageEncoder">
      <layout class="ch.qos.logback.classic.PatternLayout">
        <pattern>[${ 변수 }|${ 변수 }][%d{ "yyyy-MM-dd'T'HH:mm:ss,SSSZ" }][%-5level][%logger][%method]%X{ MDC 변수 }|%msg%n</pattern>
      </layout>
    </encoder>
    <topic> query-log </topic>
    <keyingStrategy class="com.github.danielwegener.logback.kafka.keying.RoundRobinKeyingStrategy" />
    <deliveryStrategy class="com.github.danielwegener.logback.kafka.delivery.AsynchronousDeliveryStrategy" />
    <producerConfig>bootstrap.servers= localhost:9092 </producerConfig>
    <producerConfig>acks=0</producerConfig>
    <producerConfig>max.in.flight.requests.per.connection=1</producerConfig>
  </appender>
</appender>
```



Kafka 모니터링

ActiveMQ

Home | Queues | Topics | Subscribers | Connections | Network | Scheduled | Send

Queue Name [ ] Create

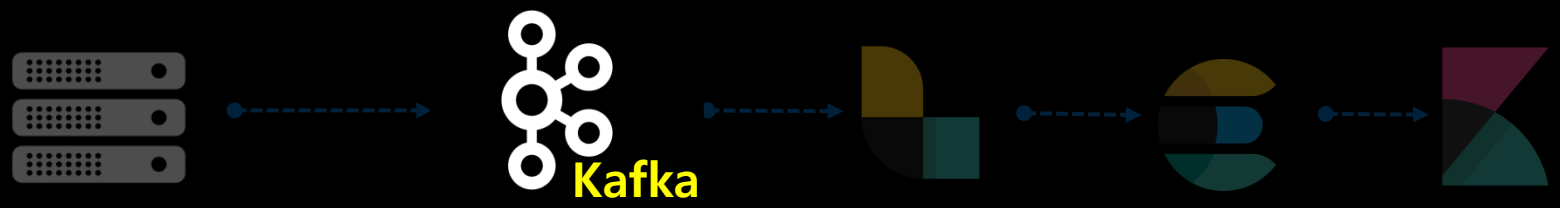
Queues

Name ↑	Number Of Pending Messages	Number Of Consumers	Messages Enqueued	Messages Dequeued	Views	Operations
Module A	0	12	42613	42613	Browse Active Consumers Active Producers atom rss	Send To Purge Delete
Module B	0	40	516157	516157	Browse Active Consumers Active Producers atom rss	Send To Purge Delete
Module C	0	10	2732	2732	Browse Active Consumers Active Producers atom rss	Send To Purge Delete

Kafka 모니터링

./kafka-run-class.sh kafka.admin.ConsumerGroupCommand --group logstash --bootstrap-server localhost:9092 --describe

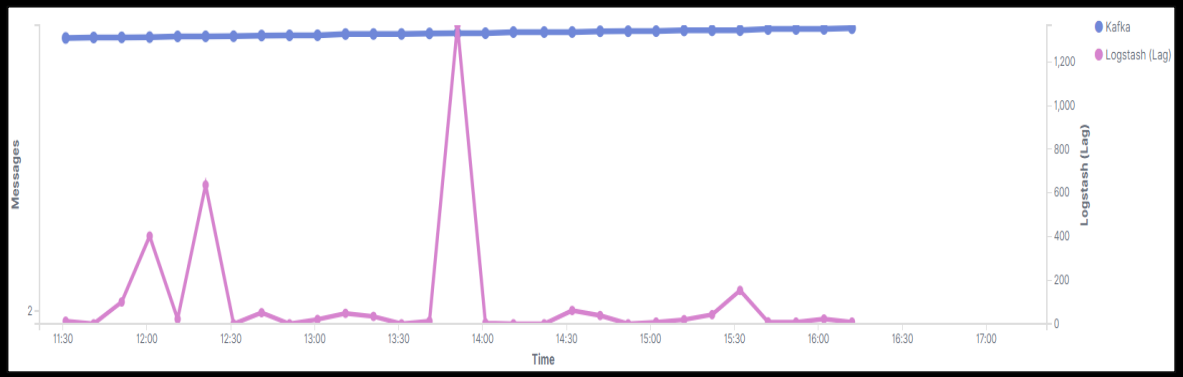
```
TOPIC            PARTITION  CURRENT-OFFSET  LOG-END-OFFSET  LAG             CONSUMER-ID
 HOST            CLIENT-ID
pilot-log-query  0          2464064         2464064         0               logstash-0-19c56d86-7446
                 logstash-0
pliot-log-normal 0          29493           29493           0               logstash-0-19c56d86-7446
                 logstash-0
adm-log-query    0          2135534         2135534         0               logstash-0-19c56d86-7446
                 logstash-0
menu-history     0          1033            1033            0               logstash-0-19c56d86-7446
                 logstash-0
login-history    0          7613            7613            0               logstash-0-19c56d86-7446
                 logstash-0
```



Kafka 모니터링

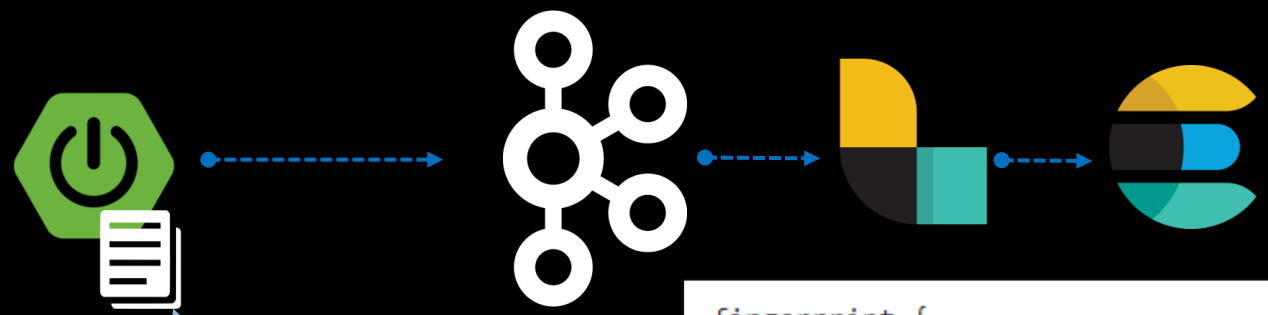
Messages in topic

Time	# of messagens in topic	# of the lag (logstash)
16:12	8,508,222	8
16:02	8,191,432	22
15:52	8,152,665	8
15:42	8,123,570	9
15:32	7,634,155	153
15:22	7,602,602	42
15:12	7,548,620	19
15:02	7,259,110	8
14:52	7,230,933	0
14:42	7,203,062	38



중복 데이터

한국 D.C.



2019090518.log

30분 단위 전송 ① *18:00~18:30*

② *18:00~19:00*

해외 D.C.

2019090518.log

1시간 단위 Rolling

```
fingerprint {
        source => ["logDtm", "globalId", "sqlId"]
    target => "[@metadata][fingerprint]"
    method => "SHA1"
    concatenate_sources => true
  }
```

```
elasticsearch {
    hosts => ["      locahost:6010      "]
    index => "bpo-sql-logs-%{+YYYY.MM.dd}"
    document_id => "%{[@metadata][fingerprint]}"
}
```

글로벌 시간

```
<layout class="ch.qos.logback.classic.PatternLayout">
    <pattern>[${ 변수 }|${ 변수 }][%d{ "yyyy-MM-dd'T'HH:mm:ss,SSSZ" }]
</layout>
```

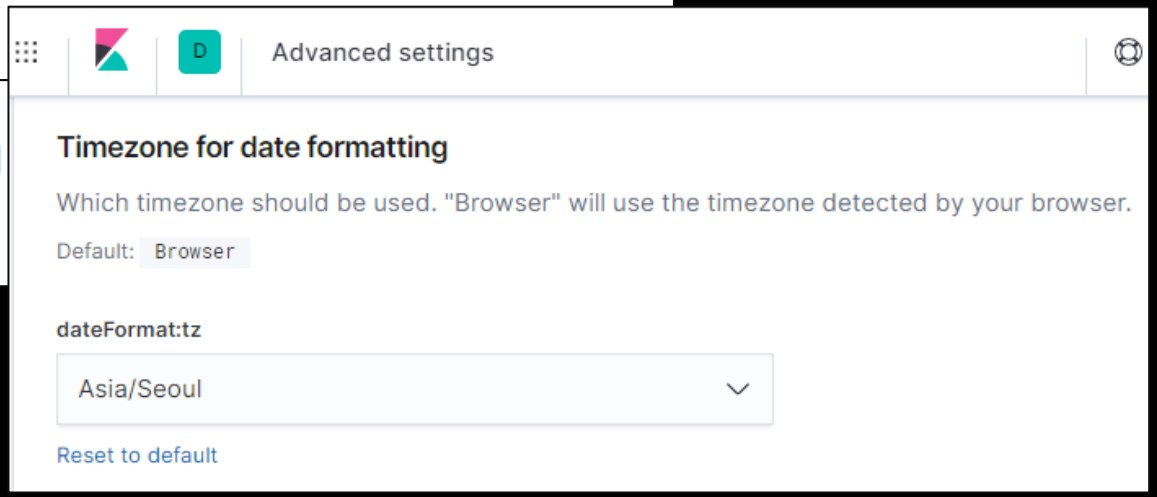
Logback

```
grok {
    match => {"message" => "\[%{DATA:lat}\|%{DATA:lon}\]\[%{TIMESTAMP_ISO8601:logDtm}\]

}
```

Logstash

```
date {
        match => [ "logDtm", "ISO8601" ]
        target => "logDtm"
  }
```

Kibana

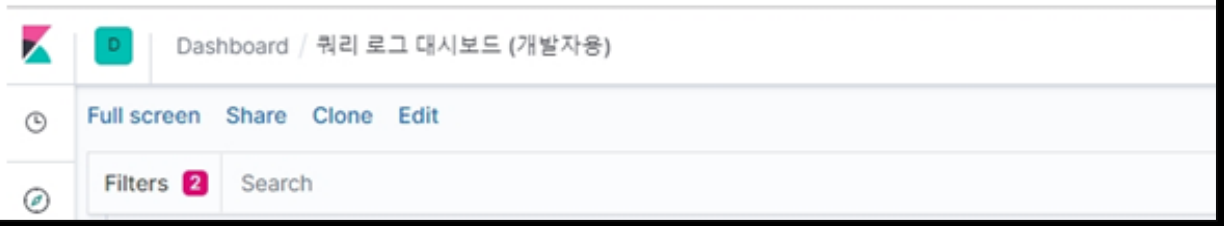


쿼리가 일부만 있다고?

키바나에서 로그를 보려고하는데, 일부 로그가 제대로 나오지 않는것 같아서 확인 부탁드립니다.

- 대상 : 운영 / _007-169-1563512070
- 조회시간 : 7월 19일 13시 54분 전후로 조회

해당 로그를 키바나에서 조회하면.. adm 쿼리만 세개 나오고.. 해당 서비스에서 호출되어야하는 com-cello-sclis-bms-mntr-clr-dao- -selectSubDataExtrcList 는 조회가 되지 않습니다.



SQL 상세현황

Document ID	SQL ID	Execution Time (ms)	Start Time (한국시간)
c33abb20756be443f6ab76d0f2ec6adf3c59e506	com-cello-sclis-adm-plugin-log-dao-	15	Jul 19, 2019 @ 13:54:31.735
7a6d61e2b312e8c2e71c6ac232a43108984d665b	com-cello-sclis-adm-auth-auth-dao-	23	Jul 19, 2019 @ 13:54:30.212
c5b19fdcef3378ef7adbe737a616f9c0d1f85210	com-cello-sclis-adm-auth-resource-d	6	Jul 19, 2019 @ 13:54:30.183

쿼리가 일부만 있다고?

```
if "_grokparsefailure" in [tags] {

        elasticsearch {
            hosts => ["        locahost : 6010        "]
            index => "logstash-grokparsefailure-%{
        }
        stdout { codec => rubydebug }
}
```

Index management

Update your Elasticsearch indices individually or in bulk.

Search

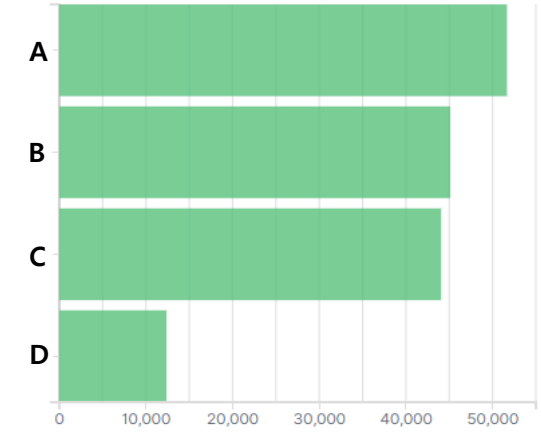
Name ↑	Health
logstash-grokparsefailure-2019.08.21	yellow
logstash-grokparsefailure-2019.08.22	yellow
logstash-grokparsefailure-2019.08.23	yellow
logstash-grokparsefailure-2019.08.24	yellow
logstash-grokparsefailure-2019.08.26	yellow

잘 쓰고 있는가

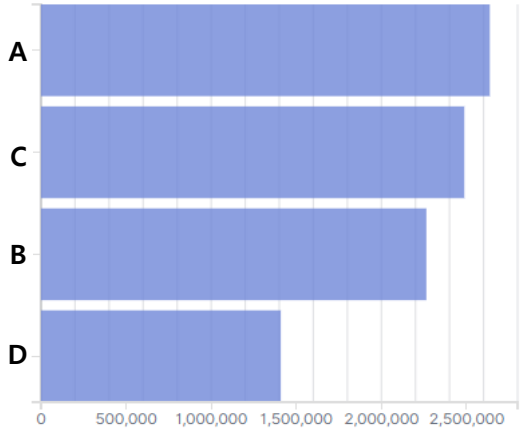
트랜잭션 요약

7,899	242,372	490
발생 트랜잭션	전체 SQL	3초 이상 SQL

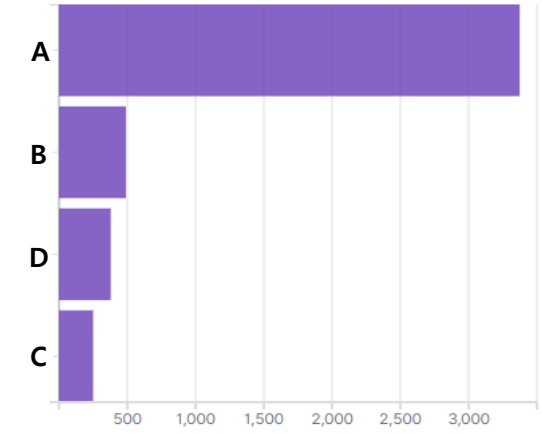
DC별 발생 트랜잭션



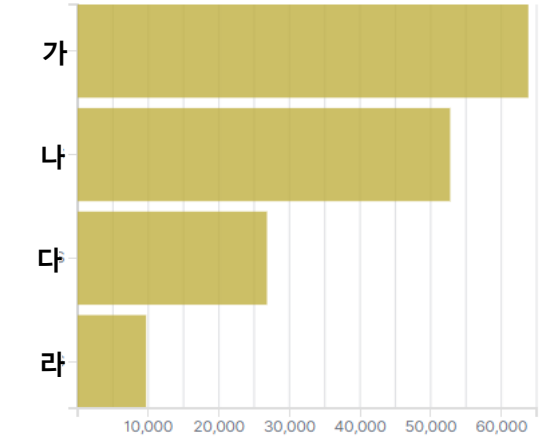
DC별 발생 전체 SQL



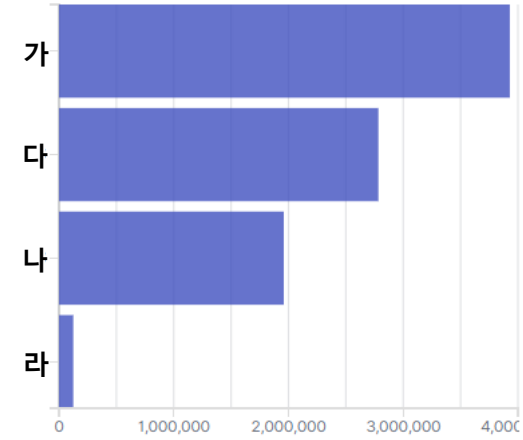
DC별 3초 이상 SQL



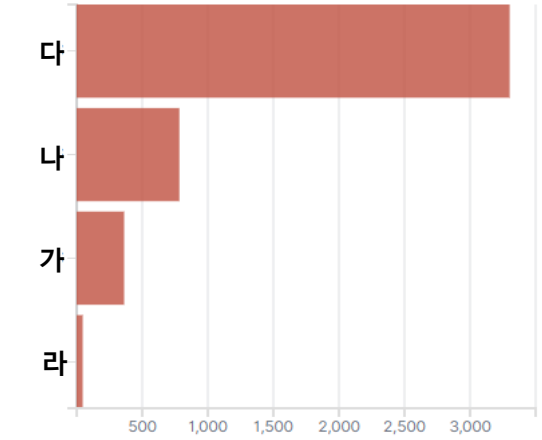
서비스 모듈별 발생 트랜잭션



서비스 모듈별 전체 SQL

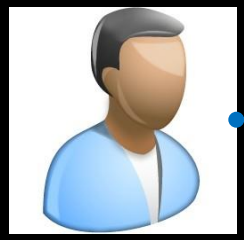


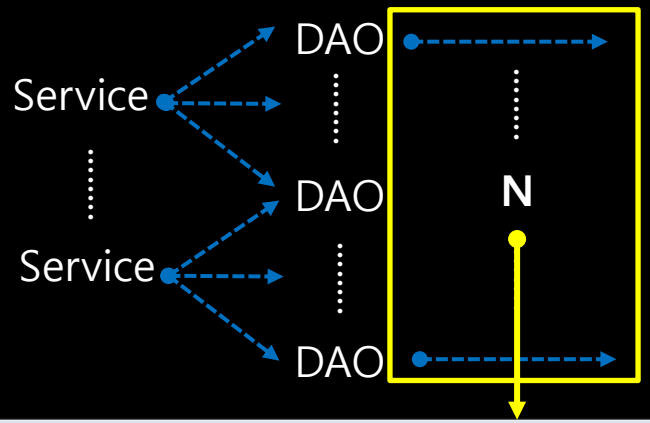
서비스 모듈별 3초 이상 SQL



SQL 기준 TOP 20 응답 지연 서비스

Service	Module	Average (ms)	Max (ms)
download.do	가	5,369.385	25,852
TMS_EXE_11_001	나	1,950.28	23,533
BMS_FI_01_006	다	1,005.4	3,400
TMS_RST_35_001	나	949.95	18,023
FIS_OSR_01_001	가	681.944	51,550





응답지연 TOP 20 SQL

SQL ID		Module	Average (ms)	Max (ms)	Requests
com-cello-sclis-fis-comn-pricec	-listFisPriceCalObjMainExcel	가	14,665.8	25,852	10
com-cello-sclis-tms-clr-clr-dao-	ClrStatusToSendToBms	나	14,106.235	57,677	357
com-cello-sclis-tms-exe-exe-dac	stList	나	9,485	23,533	5
com-cello-sclis-fis-ob-shpngcnt	SrSrchSelect	가	3,944.387	51,550	31
com-cello-sclis-tms-perf-perf-da	crList	나	3,782.4	18,023	10

Transaction 상세 현황

Transaction	Service	Region	Module	LSS	User	#. of SQL	Execution time (ms)	Start Time (한국시간)	Local Time (현지시간)
eucelloiap4_PFIS1-FIS_MIV_01_001-138-1566948568	FIS_MIV_01_001	EU	FIS	T412	yeonjoo.koh	8	2,490	Aug 28, 2019 @ 08:29:28.789	Aug 28, 2019 @ 00:29:28.789

수행된 SQL 문(바인드변수 포함)

1–8 of 8

Time	SQL	elapsedTime	serverNm

Expanded document

View surrounding documents View single document

Table JSON

@timestamp Aug 28, 2019 @ 08:35:38.553

@version 1

SQL

```
/*SQL_ID: com-cello-sclis-adm-auth-resource-dao-AdmRscCommonDao-selectOneRscIdSvcId */
                                    SELECT  RSCM.RSC_ID
                                    FROM    CELLOC.TADM_AH_RSC   RSCM
                                          , ( SELECT  SYS_CD              ,SYS_NM          ,SYS_DESC
,SYS_URL_ADDR          ,SYS_ORDB               ,INS_PERSON_ID          ,UPD_PERSON_ID    FROM  CELLOC.TADM_AH_
SYS   WHERE  DEL_YN = 'N'     AND  SYS_CD IN ( 'ADM','MDM','CMS','OMS','FIS','BMS','WMS','TMS','INS','NOS','I
A','CYM','WOS','LPM','PSM','SVM','VAS','BAS','SQR','IOT','VWS','APS') )   TSYS
                                    WHERE   RSCM.DEL_YN        = 'N'
                                    AND     RSCM.USE_GCD       = 'INT'
                                    AND     RSCM.RSC_CCD       = 'SV'
                                    AND     RSCM.PRNT_RSC_ID   = 'FISS010000021291'
                                    AND     RSCM.SVC_ID        = 'FIS_MIV_01_001'
                                    AND     RSCM.SYS_CD        = TSYS.SYS_CD
```

Aug 28, 2019 @ 08:29:28.802	/*SQL_ID: com-cello-sclis-adm-auth-auth-dao-AdmUserAuthRscCommonDao-listUserRscChildCmpnt AuthExt */ WITH VIEW_INPUT_PARAM AS (SELECT USRG.USER_GRP_ID , USRG.AREA_SVR_CCD	11	elloiap4_P FIS1
Aug 28, 2019 @ 08:29:30.056	INSERT INTO /*SQL_ID: com-cello-sclis-adm-plugin-log-dao-TadmCmLogInfoDao-insertTadmCmLogI nfo */	5	elloiap4_P FIS1

+ Add filter

EDIT FILTER Edit as Query DSL

Field

Select a field

- @timestamp
- @version
- @version.keyword
- SQL
- SQL.keyword
- _id
- _index

Operator

Select an operator

Cancel Save

+ Add filter

EDIT FILTER Edit as Query DSL

Field

loginId

Operator

Select an operator

- is
- is not
- is one of
- is not one of
- exists
- does not exist

Create custom label?

Global Filter

Region	Module	Service	Transaction	User
Select...	Select...	Select...	Select...	Select...

Apply changes Cancel changes Clear form

Thank you !