카프카 기반의 대규모 모니터링 플랫폼 개발 이야기

(카프카 잘못 쓰면 망한다.)

2019.09

@@cloud.telemetry / issac.lim(임성국)

kakao

issac (임성국)

- 2018 ~ Present : KOCOON (telemetry on k8s)
- 2015 ~ Present : KEMI (Monitoring Cloud Service)
- 2014 ~ 2015 : Krane(Kakao IaaS) & CMDB

- 2013 ~ 2014 : legacy service migration
- 2013 : Easily analyze and present current and past issues through news
- 2011 : Openstack-based API Dev Platform
- 2010 ~ 2012 : KT Open API Project

- 2009 ~ 2010 : R&D of SDP for to improve small and medium-sized companies profits in the field of medical device and logistics

Telemetry ?

Telemetry is an **automated communications process** by which measurements and other data are collected at remo te or inaccessible points and transmitted to receiving equi pment for monitoring.

ref: https://en.wikipedia.org/wiki/Telemetry

cloud.telemetry ?

- Remote & Inaccessible points
  - Baremetal, IaaS, CaaS …

- We Develop & Provide automated communications process
  - MaaS (Monitoring as a Service)
  - KEMI Stats, KEMI Logs, KOCOON

카카오의

모든 클라우드 리소스에서

발생하는 데이터를 가져와

모니터링에 필요한 이벤트로

만들어 전달

-> 위에 필요한 모든 것을 자동화

Todays…

KAKAO MaaS 소개

Kafka 사용 중 발생한 이슈

1. 어떻게 문제를 확인하고

2. 어떻게 문제를 해결했는지

KEMI-*

(Kakao Event Metering & monItoring)

KEMI Logs

Why?

Log는 변한다.
-> schemaless 지원이 필요하다.

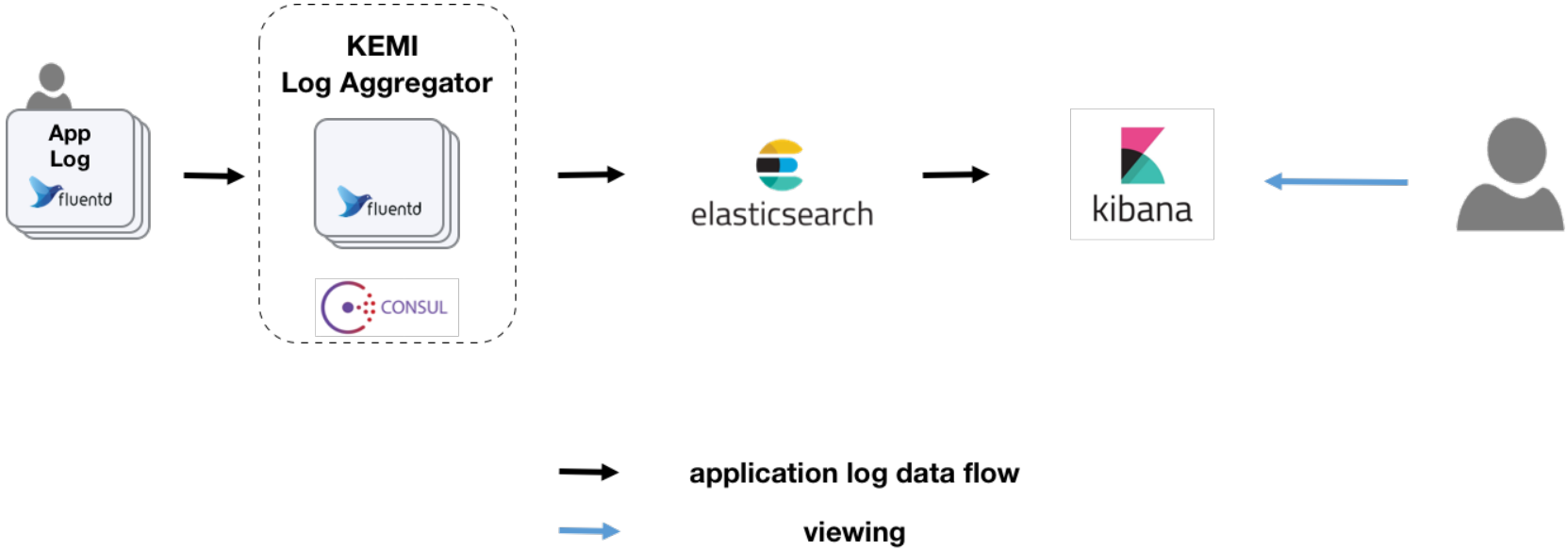
조건에 따라 Log 검색이 필요하다.
-> 쿼리도 지원이 필요하다.

Log에 서비스 상태 정보가 있다.
-> 배치/실시간으로 Log 기반 알람이 필요하다.

Log를 가지고 서비스 디버깅이 되어야 한다.
-> 실시간 로그 조회가 되어야 한다.

한 서비스라도 여러 리소스(서버, container)에서 Log가 나온다.
-> 모아야 한다.

KEMI Log Alpha (구 Crow)

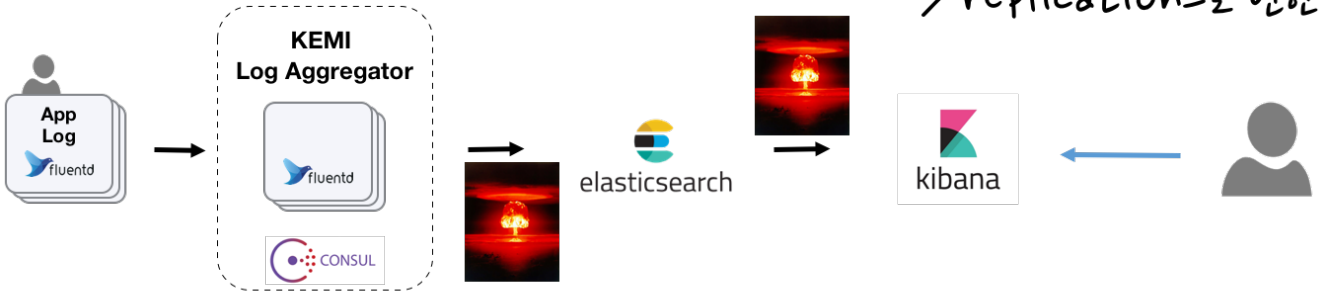


Log 증가에 따라...

heavy query

-> data node OOM

-> replication으로 인한 cluster hang

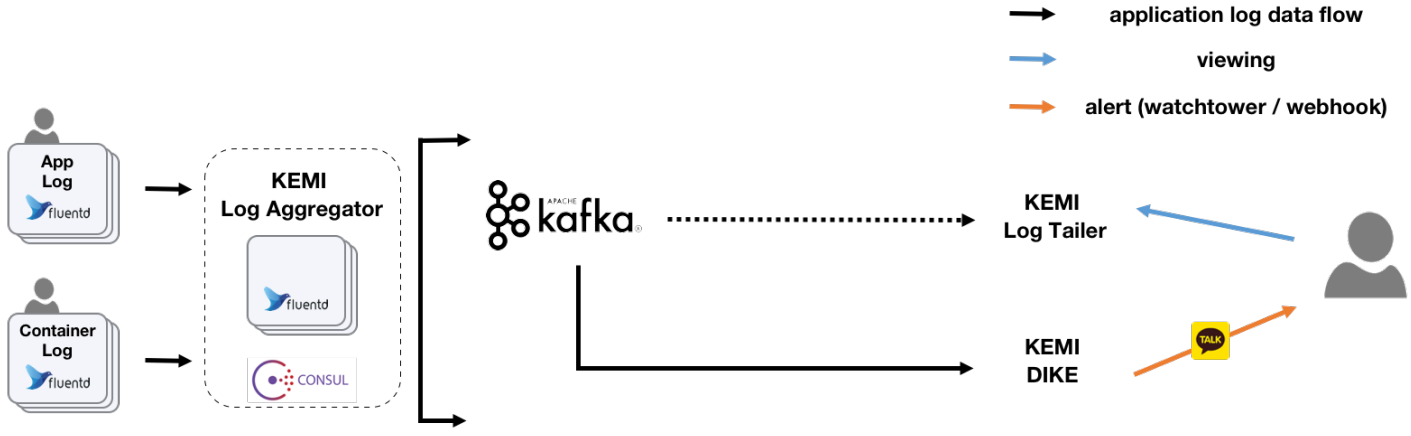


Log Aggregator 수 증가

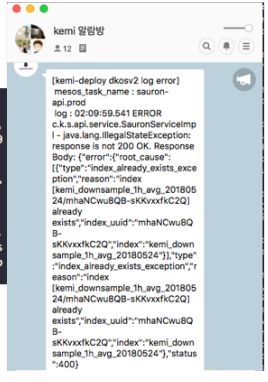
-> elasticsearch connection & 동시 처리 log양 증가

-> bulk indexing 실패

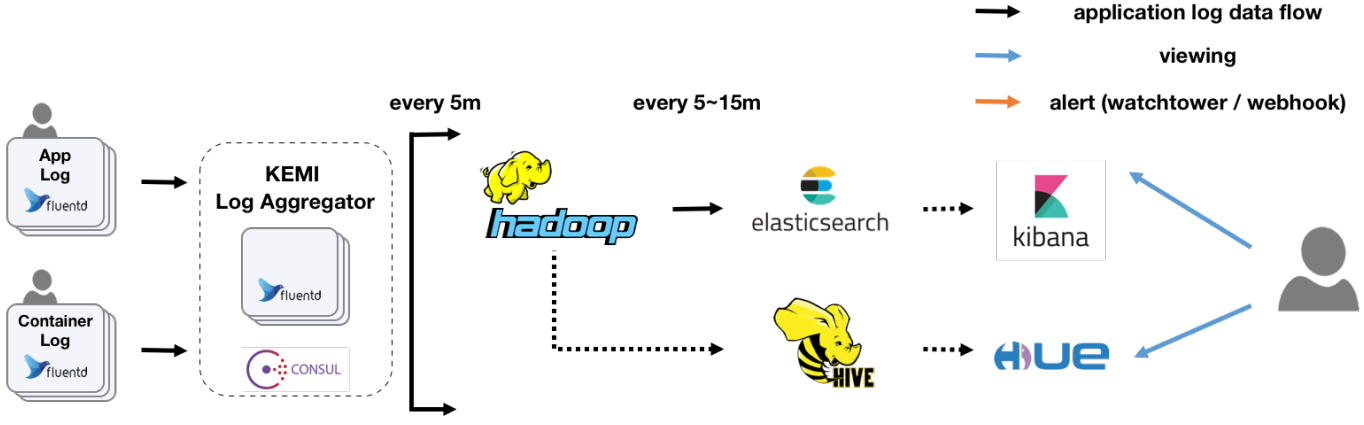
KEMI Log (Realtime)

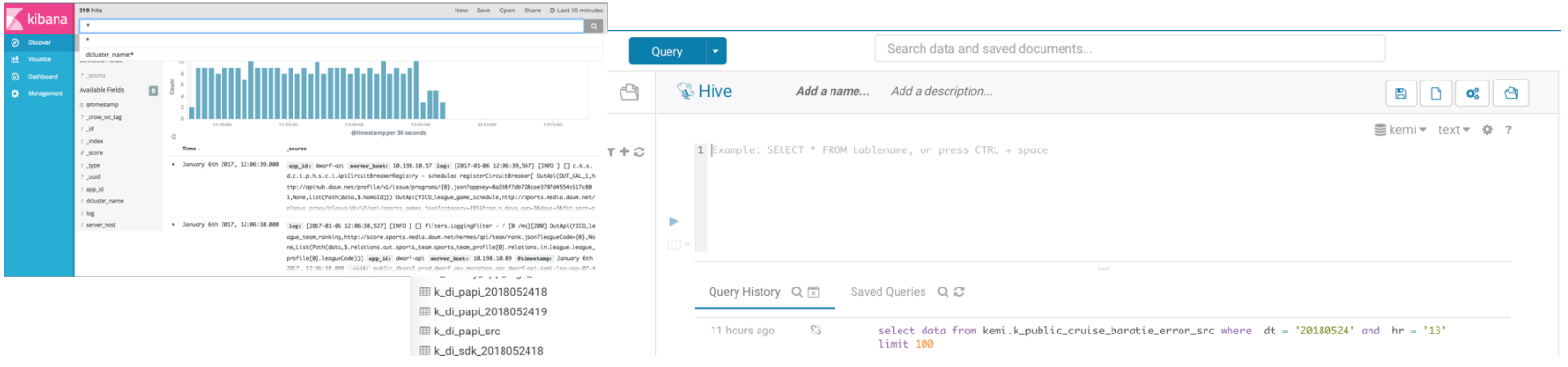



```
❯ ./kemi-log-tailer-mac-1.0.0 -svc_tag public.krane
2018/05/25 11:05:37 Rebalanced: &{Type:rebalance start Claimed:map[] Released:map[] Current:map[]}
{"pid":"5249","severity":"WARNING","ident":"nova.scheduler.client.report","message":"[req-174e79e3-9e63-49ff-805e-b31370ffa50c - - - - -] Unable to refresh my resource provider record",
"hostname":"krane-          ","node":"nova_compute","type":"nova_compute","phase":"pg1","@timestamp":"2018-05-25T11:05:15+09:00","_uuid":"public_krane-          -17e970bb-a471-43e9
-aafa-b402915ac334","_kemi_tag":"public.krane"}
{"pid":"7951","severity":"INFO","ident":"nova.compute.resource_tracker","message":"[req-16d7ace7-1bc5-4d58-bf1b-dbca1f069255 - - - - -] Final resource view: name=krane-
io phys_ram=96657MB used_ram=78336MB phys_disk=870GB used_disk=750GB total_vcpus=24 used_vcpus=76 pci_stats=[]","hostname":"krane-          ","node":"nova_compute","type":"nova_compute"
,"phase":"pg1","@timestamp":"2018-05-25T11:05:27+09:00","_uuid":"public_krane-kemi-log-agg-          :-7843668d-e665-4922-9968-164930793da4","_kemi_tag":"public.krane"}
2018/05/25 11:05:37 Rebalanced: &{Type:rebalance OK Claimed:map[public.krane:[0 1 2 3 4 5 6 7]] Released:map[] Current:map[public.krane:[0 1 2 3 4 5 6 7]]}
{"pid":"25542","severity":"INFO","ident":"neutron.wsgi","message":"[req-320a9441-ebcb-40b6-94fe-528a2ae7e474 1e5d8d3e421a44f6ad2c24f8469f822c 463833717ee64417856ee163280f0bb9 - - -] 10.
61.112.12,10.60.19.248 - - [25/May/2018 11:05:29] \"GET /v2.0/ports.json?network_id=8788e7c1-11fb-41e4-8d5e-23b8149c11e7&device_owner=network%3Adhcp HTTP/1.1\" 200 561470 0.539703","hos
tname":"krane-          ","node":"openstack_control","type":"neutron_server","phase":"pg1","@timestamp":"2018-05-25T11:05:29+09:00","_uuid":"public_krane-brown172-220453d5-4551-4e9b-b6b
0-ef1342175110","_kemi_tag":"public.krane"}
```

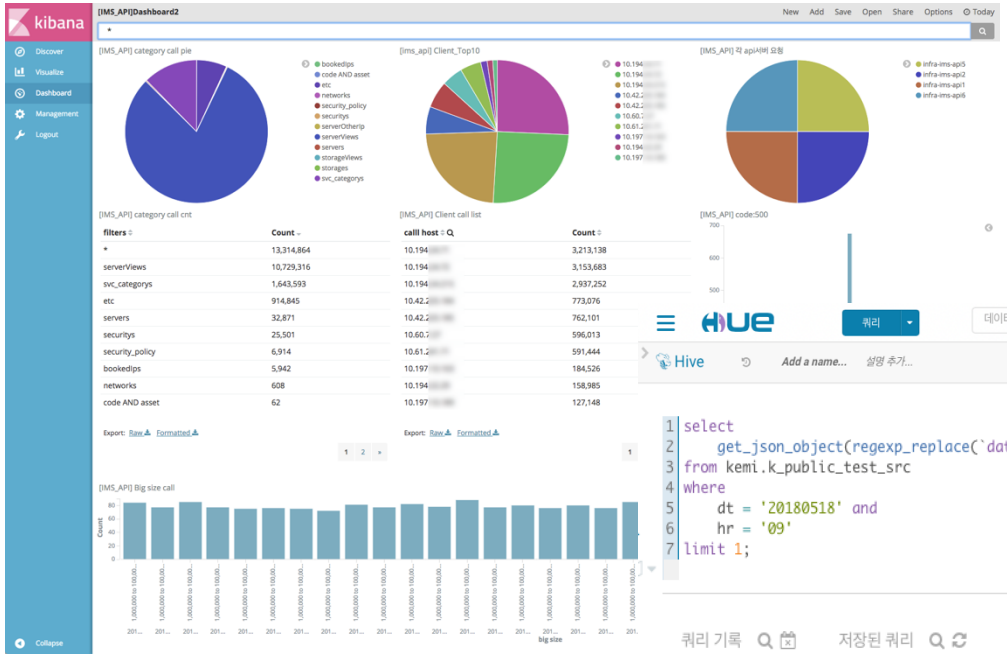


KEMI Log (batch)





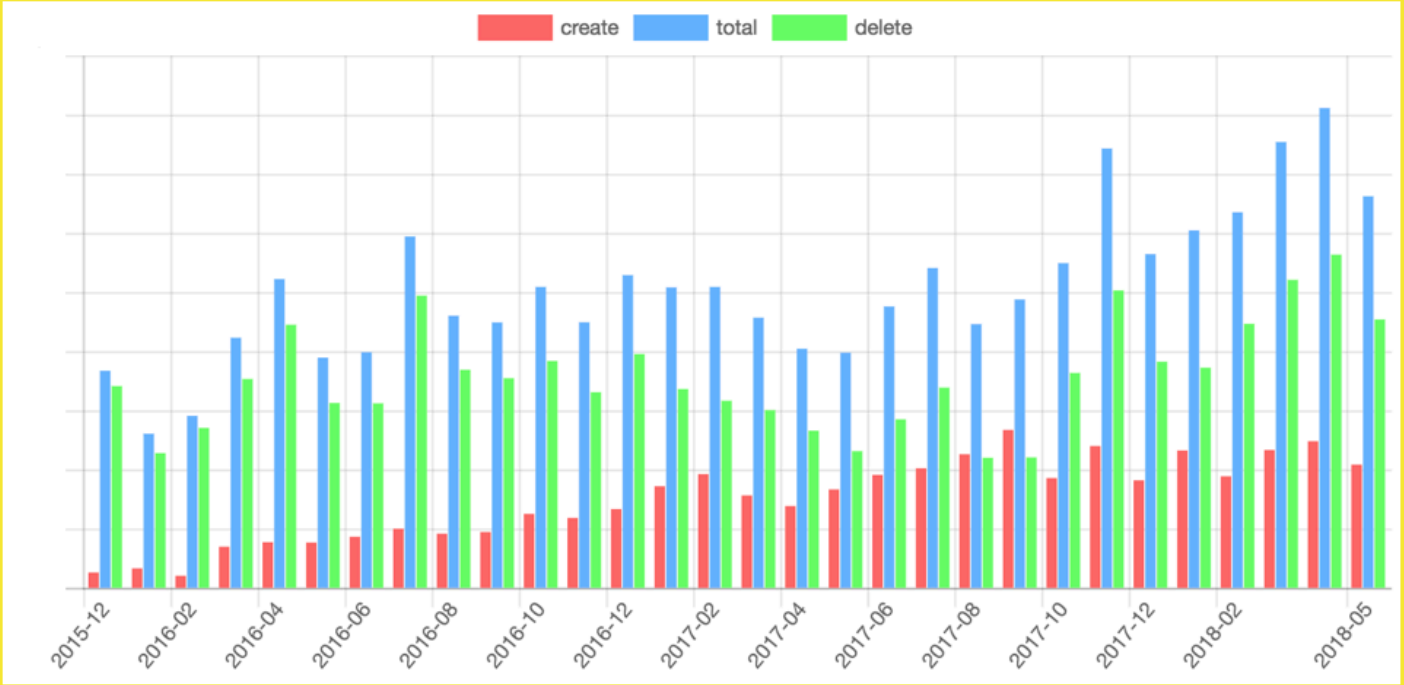
KEMI Logs Provide …



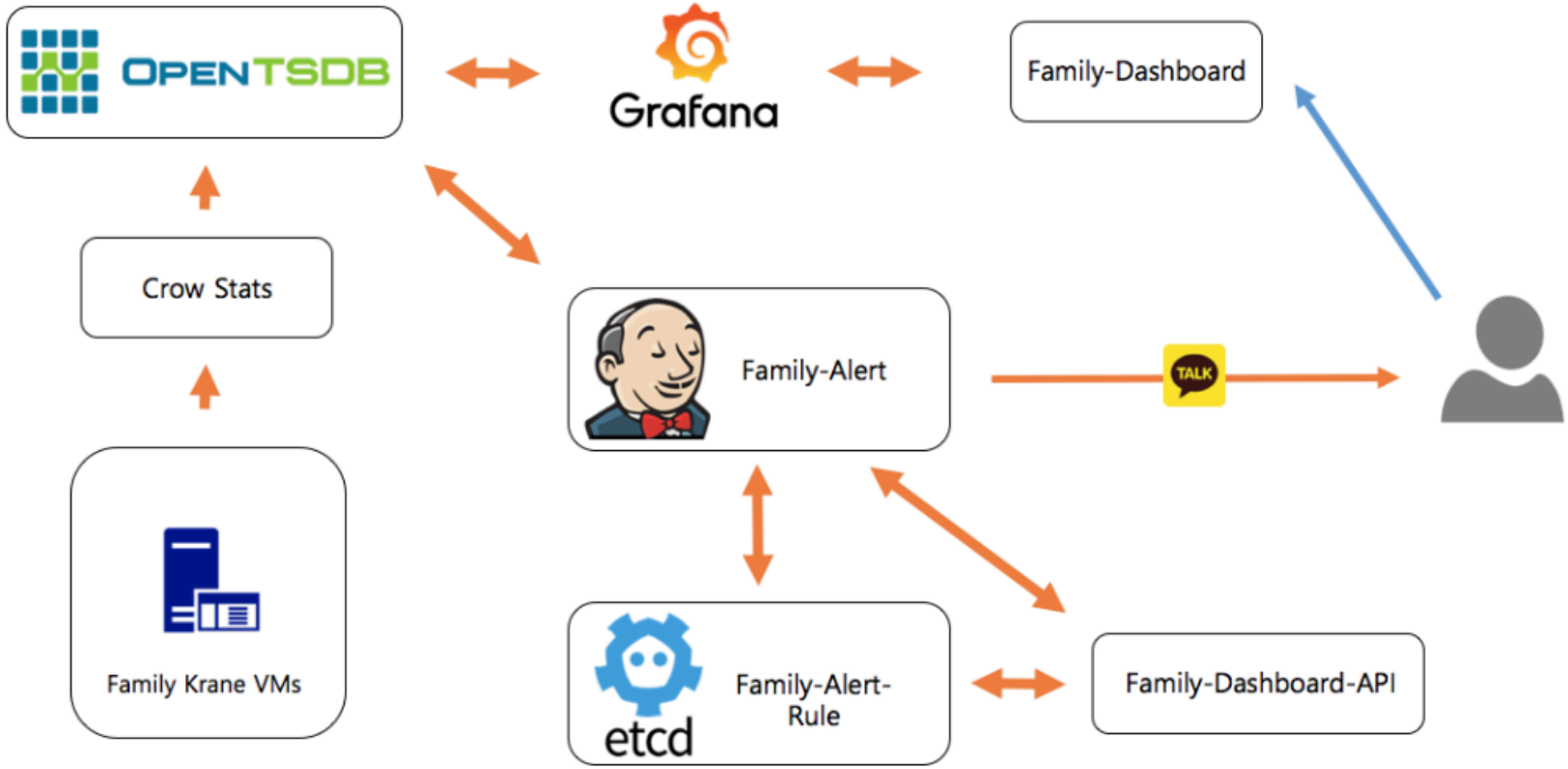
KEMI Stats

Why?

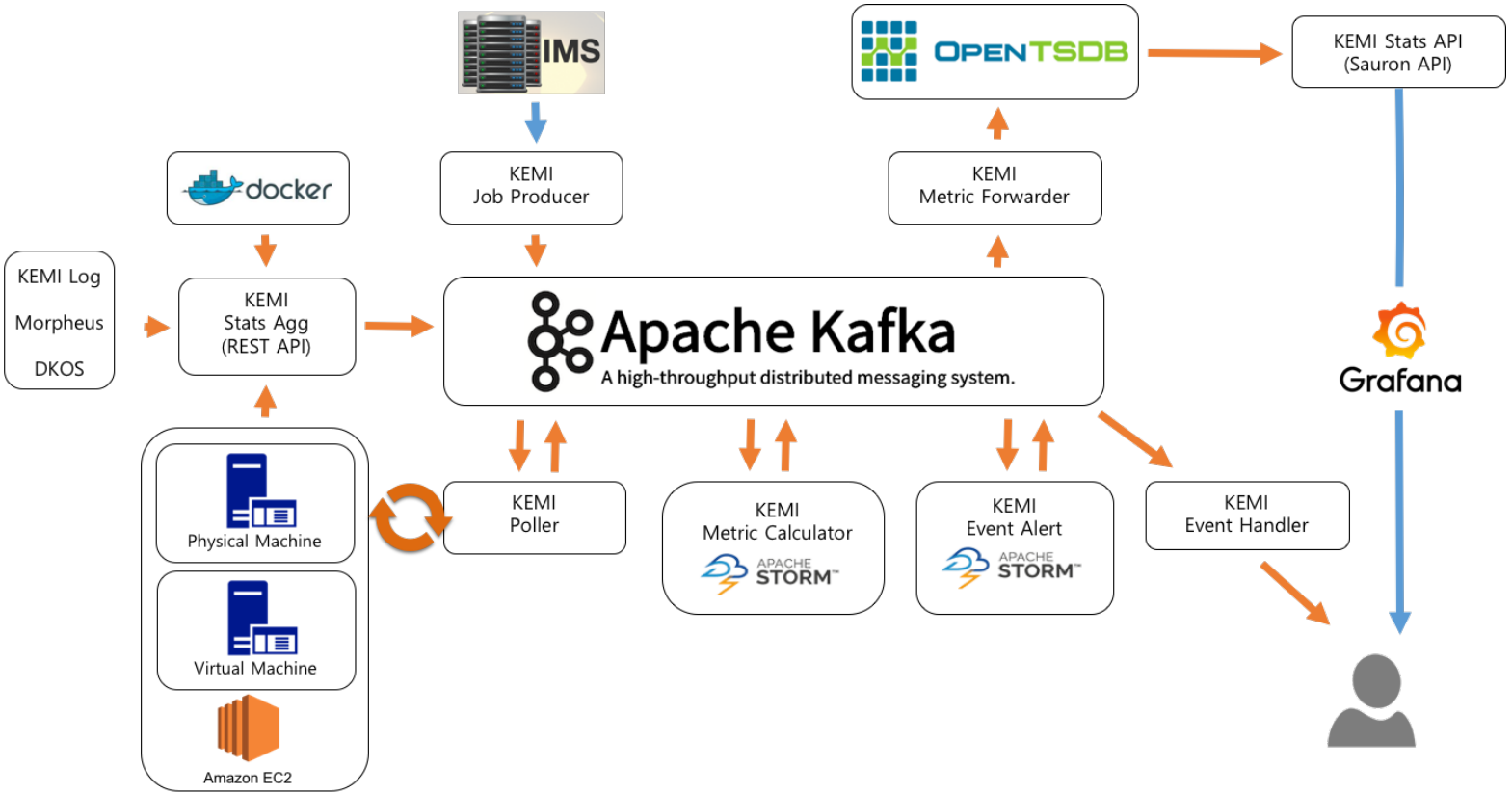
Kakao IaaS (Krane)



KEMI Stats v1

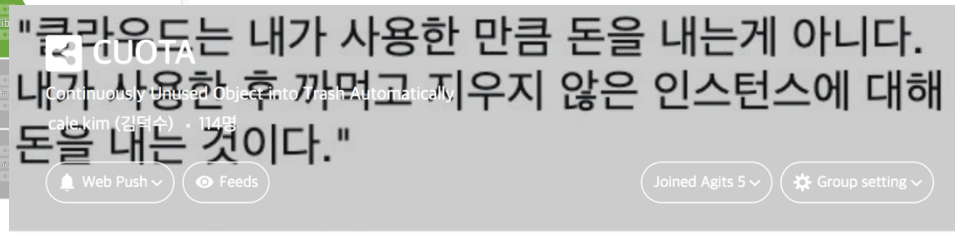


KEMI Stats v2 (Realtime)



KEMI Stats v2

KEMI Stats v2 Provide …



Continuously Unused Object into Trash Automatically

KRANE DEV 인스턴스 들은 CUOTA 시스템을 통해 관리 되고 있습니다. 유휴 자원으로 판단 되는 인스턴스를 대상으로 검색/공지/정지/삭제 진행 되며, 인스턴스를 화이트 리스트에 등록한 경우에 관리 대상에서 제외 됩니다.

- 공용으로 관리해야하는 VM 들은 멤버 모두 프로젝트 참여해주시길 부탁드립니다.
- 화이트 리스트 등록 가이드 : https://kakao.agit.in/g/300008675/wall/320944751
- 공지/정지/삭제 이력 확인 가이드 : https://kakao.agit.in/g/300008675/wall/320945034

Thx Joanne

KAFKA

We use

Version : 0.10.x (from 0.8.x), 0.11.x

KEMI Logs : 00 TB/day, 0000 topics

KEMI Stats : 00 GB/day, 0 topics

Consumer : Use consumer & new consumer API

Producer acks : Leader only

Compression type : snappy

Story 1

Problem

@kemi stats

어느 시점 부터 특정 broker에 원인을 알 수 없는 이슈가 생기면서 별 다른 error msg 없이 consumer, producer들이 제대로 작동하지 않음

Analysis

log에는 warning log정도만…
이슈 생기는 당시에는 별다른 로그가 없음.

[2019-07-14 18:38:56,439] WARN Map failed (kafka.utils.CoreUtils$)[2019-07-14 18:38:56,439] WARN Map failed (kafka.utils.CoreUtils$)java.io.IOException: Map failed at sun.nio.ch.FileChannelImpl.map(FileChannelImpl.java:940) at kafka.log.AbstractIndex$$anonfun$resize$1.apply(AbstractIndex.scala:116) at kafka.log.AbstractIndex$$anonfun$resize$1.apply(AbstractIndex.scala:106) at kafka.utils.CoreUtils$.inLock(CoreUtils.scala:234) at kafka.log.AbstractIndex.resize(AbstractIndex.scala:106) at kafka.log.AbstractIndex$$anonfun$trimToValidSize$1.applymcVsp(Abstr

각 컴포넌트 코드 확인,
ci를 통한 전체 테스트에서는 원인이 발견되지 않음.

서비스에서 사용하는 topic의 사용량이 크게 늘어나지도 않음.

특정 broker를 클러스터에서 제외하면 문제가 해결됨 ?

Analysis

전체 topic 확인

1. __consumer_offsets topic의 특정 파티션이 00GB 정도로 커져있음
2. kafka retention을 조정했으나 별다른 효과 없음
3. __consumer_offsets topic 자체의 config를 확인
   - cleanup.policy : compact
   - segment.bytes: 104857600
   - compression.type: producer

Analysis

__consumer_offsets 관련 이슈들

- https://issues.apache.org/jira/browse/KAFKA-5413
- https://issues.apache.org/jira/browse/KAFKA-3917
- https://grokbase.com/t/kafka/users/1632scdgzx/consumer-offsets-topic-cleanup-policy

compaction 관련 default 설정 변경 이력

- https://kafka.apache.org/documentation/#upgrade_901_notable

We solve the problem

1. Describe topic
   - kafka-configs.sh --zookeeper zookeeper:2181 --entity-type topics --entity-name __consumer_offsets -describe

2. Remove topic config
   - kafka-configs.sh --zookeeper zookeeper:2181 --entity-type topics --entity-name __consumer_offsets --alter --delete-config compression.type
   - kafka-configs.sh --zookeeper zookeeper:2181 --entity-type topics --entity-name __consumer_offsets --alter --delete-config cleanup.policy
   - kafka-configs.sh --zookeeper zookeeper:2181 --entity-type topics --entity-name __consumer_offsets --alter --delete-config segment.bytes

We solve the problem

- Before
  - 000MB ~ 00GB __consumer_offsets-XX
- After
  - 00KB ~ 0MB __consumer_offsets-XX

Story 1

Problem

@kemi logs

We use “auto.create.topics.enable”

Cannot Create Topic

Analysis

kafka는 broker당 적당한 Partition 수 유지가 필수

kafka 1.10 이상에서는 broker당 4000, 전체 클러스터 기준 200,000이 가능

Analysis

#of Partition

- partition count on each broker < 2k
- keep partition size on disk manageable (under 25G per partition)

Cluster Size (no. of brokers)

- how much retention we need
- how much traffic cluster is getting

Cluster Expansion

- Disk usage on the log segments partition should stay under 60%
- Network usage on each broker should stay under 75%

Analysis

Cluster Monitoring

- Keep cluster balanced
- Ensure that partitions of a topic are fairly distributed across brokers
- Ensure that nodes in a cluster are not running out of disk and network

As a rule of thumb, if you care about latency, it's probably a good idea to limit the number of partitions per broker to *100 x b x r*, where *b* is the nu mber of brokers in a Kafka cluster and *r* is the replication factor.

ex: broker 50, replication factor 2 -> 100 * 50 * 2 = 10,000 partition per cluster

ref:

https://www.slideshare.net/HadoopSummit/apache-kafka-best-practices

https://www.confluent.fr/blog/how-choose-number-topics-partitions-kafka-cluster

Analysis

Cannot diff “.” “_”

- public.test.info topic이 있는 경우 public.test_info 생성이 안됨.

We solve the problem

- Service Log 등록 시 “.” “_” 를 동일한 이름으로 체크
- broker 증설
  - 기존 broker * 2
- repartition ???

Story 1

Problem

@kemi logs

0000 개의 서비스 로그가 점점 증가하면서 broker 증설이 필요해짐

기존 broker중 일부는 10MB/sec 정도의 트래픽으로 힘들어함

신규 broker로 TOPIC 별 Repartition이 필요해 짐

kafka manager !

During Repartition : Consumer & Producer … OTL

Analysis

kafka topic : partition 별로 leader & follower

topic이 많은 경우

전체 repartition 시 leader 변경
replication 으로 인한 데이터 복제
… etc …

-> consumer & producer 둘 다 문제

```
#/bin/bash

KAFKABIN="/opt/kafka/bin"
ZK="zookeeper:2181"
BROKERS="1"

for b in $(seq 2 10);
do
    BROKERS="$BROKERS,$b"
done

for t in $($KAFKABIN/kafka-topics.sh --list --zookeeper $ZK);
do
    cp sample-to-move.json topics/$t-to-move.json
    sed -i "s/sample/$t/g" topics/$t-to-move.json

    $KAFKABIN/kafka-reassign-partitions.sh --zookeeper $ZK --topics-to-move-json-file topics/$t-to-move.json --broker-list "$BROKERS" --generate | tail -n 1 > templates/$t-template.json
done
```

```
{
    "topics":
    [{"topic": "sample"}],
    "version":1
  }
```

```
#/bin/bash

KAFKABIN="/opt/kafka/bin"
ZK="zookeeper:2181"

for t in $($KAFKABIN/kafka-topics.sh --list --zookeeper $ZK);
do
    $KAFKABIN/kafka-reassign-partitions.sh --zookeeper $ZK --reassignment-json-file templates/$t-template.json --execute
    $KAFKABIN/kafka-topics.sh --describe --topic $t --zookeeper $ZK
    sleep 10;
    $KAFKABIN/kafka-topics.sh --describe --topic $t --zookeeper $ZK
done
```

We solve the problem

- repartition by topic
- topic 별 retention은 줄이고 작업 해야함
- 단 삭제 중인 topic이 있는 경우 topic 이름을 가져올 때
  - [topic name] - mark for deletion 으로 topic 이름이 넘어옴 (예외처리 필요)

TODO

We Consider

- Move from 0.10.x, 0.11.x to 1.x (kafka)
- Cluster ha 에 대한 고민
- Dynamic Repartition or Cluster by ?
- Lazy retry (poc 진행 중)

Thanks

special thanks to you & @@cloud.telemetry

Q & A