Telemetry on K8S

임성국 issac.lim 송지현 jenny.ssong
Cloud.Telemetry @ 카카오

if (kakao) dev 2019
01 - 04  Prologue

05 - 09  KOCOON-HERMES
01  Who We Are?
Telemetry?

Telemetry is an automated communications process by which measurements and other data are collected at remote or inaccessible points and transmitted to receiving equipment 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 (Kakao Event Metering and monitoring)
    - KOCOON?

ref: [https://en.wikipedia.org/wiki/Telemetry](https://en.wikipedia.org/wiki/Telemetry)
02 Previous Stage
KEMI Stats Provide ...

Continuously Unused Object into Trash Automatically

KRANE DEV 인스턴스 들은 CUOTA 시스템을 통해 관리 되고 있습니다. 유�두 자원으로 판단 되는 인스턴스를 대상으로 삭제/공급/생산/복제 진행 되며, 인스턴스를 화이트 리스트에 등록한 경우에 관리 대상에서 제외 됩니다.

- 공급으로 관리해야하는 VM 들은 멤버 모두 프로젝트 참여해주사길 부탁드립니다.
- 화이트 리스트 등록 가이드 : https://kakao.agit.in/g/300008675/wall/320944751
- 공급/생산/복제 여러 확인 가이드 : https://kakao.agit.in/g/300008675/wall/320945034

Thx Joanne
KEMI Logs Provide ...

SELECT * FROM kemi_k_public_test.src
WHERE dt = '20180518' AND hr = '09'
LIMIT 1;
03 Is It Enough?
Head: KEMI-*

Longtail: ?

ref: https://mgcabral.wordpress.com/2012/03/04/thelongaileconomics/
Users want

- Own Resource
- To deal with resources in their own way
  realtime(< 1m) & heavy query
- To Customize everything

We think

- Divide resources by services
- Provide self-monitoring & self-healing
- Provide more values (services)
New CaaS (DKOSv3)

DKOSv3
- k8s based container orchestrator
- ifkakao 2019 day1 15:00 ~

카카오톡 적용 사례를 통해 살펴보는 카카오 클라우드의 KaaS
Telemetry on Own Resource

===

Telemetry on k8s
Kakao Container based service monitoring
Metric based Self Monitoring

KOCOON-Prometheus & CUPIDO

App
Ingress
Kube-System

: Pulling  ---: Push
k8s Status View

Cluster Health
- Control Plane UP: UP
- Alerts Firing: 0
- Alerts Pending: 0
- Node Disk Pressure: 0
- Node Memory Pressure: 0
- Node Not Ready: 0
- Nodes Unscheduled: 0
- Crashing Control Plane: 0

Status
- API Servers UP: 100%
- Nodes UP: 100%
- API Server Request Error Rate: 0%

Capacity
- CPU Utilisation: 5.002%
- Memory Utilisation: 39.0%
- Filesystem Usage (AVG): 21.0%
- Pod Utilisation: 7%
k8s DrillDown View
k8s Ingress Controller View
KOCOON-Cupido Alarm

Watchtower

[📸 kocoon cupido 📸]

AlertName: KubeDaemonSetRolloutStuck
Status: 🔥 firing 🔥
StartsAt: 2019-05-28 05:58:31.571433492 +0000 UTC
Message: Only 78.8% of the desired Pods of DaemonSet kube-system/cilium are scheduled and ready.

http://64-prometheus.dev.9rum.cc/graph?
g0.expr=kube_daemonset_status_number_ready%7Bjob%3D%22kube-state-metrics%22%7D+
%2F+kube_daemonset_status_desired_number_scheduled%7Bjob%3D%22kube-state-metrics%22%7D+
%2A+100+%3C+100&g0.tab=1

Watchtower

[📸 kocoon cupido 📸]

AlertName: KubeDaemonSetRolloutStuck
Status: 🌈 resolved 🌈
StartsAt: 2019-05-28 05:58:31.571433492 +0000 UTC
EndsAt: 2019-05-28 05:59:31.571433492 +0000 UTC
Message: Only 78.8% of the desired Pods of DaemonSet kube-system/cilium are scheduled and ready.
Default SLA for KOCOON-Prometheus

- **Resource Request & Limit**
  - CPU 1~4 Core, Memory 4~6 GB
Default SLA for KOCOON-Prometheus

- **Metric 보관 기간**: 3 days
- **Metric 수집 주기**: Every 60s
- **Alarm 주기**: 5~15M 동안 이벤트가 반복되면 1시간에 한 번 알림
- **~ 200 nodes, ~ 1500 pods, ~ 10,000 metrics/sec**
More About for KOCOON-Prometheus

2019-kocoon-kakao-automatic-k8s-monitoring
by templer7 @sildeshare
Log Routing

KOCOON-Hermes

App

Ingress

Kube-System

: Pulling  ➔  : Push

fluentd

kibana

hadoop
05 KOCOON-HERMES
Why Hermes?

Users want
- Own Resource
- To deal with resources in their own way real-time (<1m) & heavy query
- To Customize everything

We think
- Divide resources by services
- Provide self-monitoring & self-healing
- Provide more values (services)

Hermes provide
- Kubernetes Log Customizing
- Sink to own cluster
  - kafka cluster, elasticsearch cluster ...
- Self-monitoring (TOBE)
Dive Into Hermes - Fluentd

Why fluentd?
- Many plugins
- Stable
- CNCF
- fast PR & merge

Compose of
- Input
- Filter
- Output
Why do not using fluent-bit?

Because, fluent-bit does not support tag-rewrite

jihyun.song  Aug 8th at 6:04 PM
Hi, Is there any plugin for rewrite tag name in fluent-bit? like "match record_modifier or rewrite_tag_filter" in fluentd (edited)

2 replies

eduardo  3 days ago
that plugin is currently in "work in process"

👍 1  💯 1

jihyun.song  3 days ago
@eduardo That sounds good! Thank you for the answer!
Dive Into Hermes - KEMI Logs Overview

Application log data flow:
- Viewing
- Alert (watchtower/webhook)
Dive Into Hermes - KEMI Logs Overview

Diagram:
- KEMI Log Aggregator
- Hadoop
- Elasticsearch
- Kibana
- HUE
- KEMI Log Tailer
- Kafka
- Application log data flow
- Viewing
- Alert (watchtower/webhook)
Dive Into Hermes – Hermes goal

- Running on k8s
  - as docker file

- Using helm
  - easy to setup
  - ifkakao 2019 day1 11:00 ~ 카카오에서 컨테이너를 사용하는 방법

- User can customize the logging configuration
  - log customizing
  - routing log to own kafka & elasticsearch cluster
Goals for work
Goals for work – running on k8s

- **use DaemonSet resource type**
  - k8s ensures that all Nodes run a copy of a Pod

- **fluentd-kubernetes-daemonset**
  - slim ruby 2.6 image
  - install fluentd gem and others
  - does not support multiple destination (ex. kafka & elasticsearch)

---

**Debian**
- v1.4.2-debian-elasticsearch-1.1,v1.4-debian-elasticsearch-1
- v1.4.2-debian-loggly-1.1,v1.4-debian-loggly-1
- v1.4.2-debian-logentries-1.1,v1.4-debian-logentries-1
- v1.4.2-debian-cloudwatch-1.1,v1.4-debian-cloudwatch-1
- v1.4.2-debian-stackdriver-1.1,v1.4-debian-stackdriver-1
- v1.4.2-debian-s3-1.1,v1.4-debian-s3-1
- v1.4.2-debian-syslog-1.1,v1.4-debian-syslog-1
- v1.4.2-debian-forward-1.1,v1.4-debian-forward-1
- v1.4.2-debian-gcs-1.1,v1.4-debian-gcs-1
- v1.4.2-debian-graylog-1.1,v1.4-debian-graylog-1
- v1.4.2-debian-papertrail-1.1,v1.4-debian-papertrail-1
- v1.4.2-debian-logio-1.1,v1.4-debian-logio-1
- v1.4.2-debian-kafka-1.1,v1.4-debian-kafka-1
- v1.4.2-debian-kinesis-1.1,v1.4-debian-kinesis-1
Goals for work – fluentd-kubernetes-daemonset

- Makefile
- README.md
- docker-image/
- plugins/
- templates/
  - Dockerfile.erb
  - Gemfile.erb
  - README.md.erb
  - post_push.erb
  - entrypoint.sh.erb
- conf
  - fluent.conf.erb
  - kubernetes.conf.erb
  - prometheus.conf.erb
  - systemd.conf.erb

```
ALL_IMAGES := \
  v1.4/debian-elasticsearch:v1.4.2-debian-elasticsearch-1.1,v1.4-debian-elasticsearch-1 \ 
  v1.4/debian-logzio:v1.4.2-debian-logzio-1.1,v1.4-debian-logzio-1 \ 
  v1.4/debian-kafka:v1.4.2-debian-kafka-1.1,v1.4-debian-kafka-1 \ 
  v1.4/debian-kinesis:v1.4.2-debian-kinesis-1.1,v1.4-debian-kinesis-1

# <Dockerfile>:<version>,<tag1>,<tag2>,...
COPY Gemfile* /fluentd/
RUN buildDeps="sudo make acc a++ libc-dev libffi-dev \ 
  elsif target == "kafka" %> build-essential \ 
  automake libtool pkg-config<% end %> ... \ 
  && apt-get update \ 
  ..."
Goals for work – fluentd-kubernetes-daemonset

- Makefile
- README.md
- docker-image/
- plugins/
- templates/
  - Dockerfile.erb
  - Gemfile.erb
  - README.md.erb
  - post_push.erb
  - entrypoint.sh.erb
  - conf
    - fluent.conf.erb
    - kubernetes.conf.erb
    - prometheus.conf.erb
    - systemd.conf.erb

```ruby
ALL_IMAGES := 
  v1.4/debian-hermes:v1.4.2-debian-hermes-1.0,v1.4-debian-hermes-1
#<Dockerfile>:\<version>,<tag1>,<tag2>,...
```

```erb
<% elsif target == "hermes" %> build-essential autoconf automake libtool pkg-config<% end %>
<% when "hermes" %>
gem "fluent-plugin-kafka", "~> 0.9.0"
gem "zookeeper", "~> 1.4.11"
gem "snappy", "~> 0.8.15"
gem "fluent-plugin-elasticsearch", "~> 3.4.2"
gem "fluent-plugin-record-modifier"
```
Goals for work – using Helm

Helm Chart for In-house

<table>
<thead>
<tr>
<th>k8s resource type</th>
<th>Worker node</th>
<th>Ingress node</th>
</tr>
</thead>
<tbody>
<tr>
<td>daemonset</td>
<td>fluentd</td>
<td>fluentd-ingress</td>
</tr>
<tr>
<td>configmap</td>
<td>fluentd-cm</td>
<td>fluentd-ingress-cm</td>
</tr>
<tr>
<td>clusterrole</td>
<td>fluentd</td>
<td>fluentd-ingress</td>
</tr>
<tr>
<td>clusterrolebindings</td>
<td>fluentd</td>
<td>fluentd-ingress</td>
</tr>
<tr>
<td>serviceaccount</td>
<td>fluentd</td>
<td>fluentd-ingress</td>
</tr>
</tbody>
</table>
Goals for work – daemonset (1/2)

- Use customized fluentd docker
- Resource limit
- Environments

```yaml
apiVersion: extensions/v1beta1
kind: DaemonSet
metadata:

spec:
template:
spec:
  containers:
    image: k8s.gcr.io/kuboard/fluentd/kubernetes-daemonset:...
    imagePullPolicy: IfNotPresent
    name: kocoon-hermes
    resources:
      limits:
        memory: 400Mi
      requests:
        cpu: 100m
        memory: 200Mi
    env:
      - name: K8S_NODE_NAME
        valueFrom:
          fieldRef:
            apiVersion: v1
            fieldPath: spec.nodeName
        name: K8S_NODE_NAME
      - name: KENM_TAG
        value: dkos_v3_jenny
```

if(kakao) dev 2019
Goals for work – daemonset (2/2)

- Mount hostpath
  - timezone, application log path

- Mount configmap
  - fluentd configuration

```yaml
volumeMounts:
  - mountPath: /var/log
    name: varlog
  - mountPath: /var/lib/docker/containers
    name: varlibdockercontainers
    readOnly: true
    name: tz-config
    readOnly: true
  - mountPath: /fluentd/etc/
    name: fluentd-config

volumes:
  - hostPath:
      path: /var/log
      type: """" 
      name: varlog
  - hostPath:
      path: /var/lib/docker/containers
      type: """" 
      name: varlibdockercontainers
  - hostPath:
      path: /etc/ localtime
      type: """" 
      name: tz-config
  - configMap:
      defaultMode: 420
      name: fluentd-config
```
Goals for work – daemonset (2/2)

- Mount hostpath
  - timezone, application log path

- Mount configmap
  - fluentd configuration

```yaml
volumeMounts:
- mountPath: /var/log
  name: varlog
- mountPath: /var/lib/docker/containers
  name: varlibdockercontainers
  readOnly: true
  name: tz-config
  readOnly: true
- mountPath: /fluentd/etc/
  name: fluentd-config
volumes:
- hostPath:
  path: /var/log
  type: ""
  name: varlog
- hostPath:
  path: /var/lib/docker/containers
  type: ""
  name: varlibdockercontainers
- hostPath:
  path: /etc/ localtime
  type: ""
  name: tz-config
- configMap:
  defaultMode: 420
  name: fluentd-config
name: fluentd-config
```
Goals for work – configmap

- Fluentd configmap

- Compose of 3 files
  - input-kubernetes.conf
  - filter-kubernetes.conf
  - fluent.conf
Goals for work

- L10-12
  ∵ too many logs

- L15
  ∵ avoid OOM

- L20, L22, L24
  ∵ docker log driver
  ∵ time zone setup
Goals for work

- L10-12
  - too many logs

- L15
  - avoid OOM

- L20, L22, L24
  - docker log driver
  - time zone setup
Goals for work – filter-kubernetes.conf (1/2)

/var/log/containers$ ls
elasticsearch-data-3_logging_chown-d999f3ebde6806426ca4b87166bc2b4f4cae4769068f32758283e53bce75bc50.log
elasticsearch-master-1_logging_chown-9ae026b63f15021f6d81982d25e54d3893f63fd99ad2667c5fbff0598533622.log
fluentd-c9hc7_kube-system_kocoon-hermes-ea57791cf432f0bb6c6a2e75f026b6a67b58619cc5f254cc5139d50ef78de5f.log

filter-kubernetes.conf:

```bash
<match kubernetes.**>
  @type record_modifier
  @kube_file_regex = 'kubernetes/\./var/\./log/\./containers/\.(?<!pod_namespace)\..*/.*'
  @regexp_compiled = Regexp.compile(@kube_file_regex)
  tag "#\{ENV["KEMI_TAG"]\}"
  <record>
    @timestamp '${Time.at(time).strftime("%Y-%m-%dT%H:%M:%S.%L+09:00")}'
    cluster_name "#\{ENV["CLUSTER_NAME"]\}"
    hostname "#\{ENV["K8S_NODE_NAME"]\}"
    kubernetes ${tag.match(@regexp_compiled).named_captures}
  </record>
</match>
```

- L4-5, L11
  - replace kubernetes-metadata-plugin
  - It uses too many resources on master

=> pod_name, namespace, container_name, docker_id
Goals for work – filter-kubernetes.conf (2/2)

```bash
# log에 해당하는 부분을 파싱하기 위한 부분
# Json Log Example:
#
# "log":{""some"":"log"},
# "stream":"stdout",
# "time":"2016-02-17T00:04:05.931087621Z"
#
# 만약에 @type json이면서, 해당하는 time_key가 존재하지 않는 경우, 이 필터에 도착한 시간으로 event_time이 바뀐다.
<filter >>
    @type parser
    key_name log
    reserve_data true
    emit_invalid_record_to_error false
    remove_key_name_field true
    <parse>
        @type "#{ENV['APP_LOG_FORMAT'] || 'none'}"
        expression "#{ENV['APP_LOG_EXPRESSION'] || '/.*'}"
        types "#{ENV['APP_LOG_FORMATTYPES'] || nil}"
        keep_time_key true
        time_key "#{ENV['APP_LOG_TIME_KEY'] || '__time__log__agent__'}"
        time_format "#{ENV['APP_LOG_TIME_KEY_FORMAT'] || '%Y-%m-%d %H:%M:%S,%NZ'}"
        message_key log
    </parse>
</filter>
```
Goals for work – fluent.conf

- L7,8
  - import configurations

- L10 – 25
  - forward to KEMI Log-aggregator

@type can be replaced with kafka or elasticsearch
Which issue?

case 1. w/ fluentd-metadata-plugin
using memory with K8S node counts & metadata information
⇒ fluentd supervisor (PID 1) OOM killed (docker restart)
⇒ drop pos file

case 2. w/o fluentd-metadata-plugin
⇒ fluentd worker (PID 2) got SIGKILL
⇒ drop data which in output buffer memory (before flushed)
⇒ using same pos file
sensitive with memory
using only worker’s resource

1000 lines / sec is ok (under 4K/msg)
<table>
<thead>
<tr>
<th>차트명</th>
<th>클러스터</th>
<th>Version</th>
<th>AppVersion</th>
<th>KubesprayVersion</th>
<th>K8sVersion</th>
</tr>
</thead>
<tbody>
<tr>
<td>kcooon-hermes</td>
<td></td>
<td>1.0.3</td>
<td>1.4.2</td>
<td>kakao/v2.5.0</td>
<td>v1.9.5</td>
</tr>
<tr>
<td>kcooon-hermes</td>
<td></td>
<td>1.0.3</td>
<td>1.4.2</td>
<td>kakao/v2.5.0</td>
<td>v1.9.5</td>
</tr>
<tr>
<td>kcooon-hermes</td>
<td></td>
<td>1.0.3</td>
<td>1.4.2</td>
<td>kakao/v2.5.0</td>
<td>v1.9.5</td>
</tr>
<tr>
<td>kcooon-hermes</td>
<td></td>
<td>1.0.3</td>
<td>1.4.2</td>
<td>kakao/v2.5.0</td>
<td>v1.9.5</td>
</tr>
<tr>
<td>kcooon-hermes</td>
<td></td>
<td>1.0.3</td>
<td>1.4.2</td>
<td>kakao/v2.5.0</td>
<td>v1.9.5</td>
</tr>
<tr>
<td>kcooon-prometheus</td>
<td></td>
<td>0.0.6</td>
<td>5.7.0</td>
<td>v2.7.3</td>
<td>v1.11.5</td>
</tr>
<tr>
<td>kcooon-prometheus</td>
<td></td>
<td>0.0.6</td>
<td>5.7.0</td>
<td>v2.7.3</td>
<td>v1.11.5</td>
</tr>
<tr>
<td>kcooon-prometheus</td>
<td></td>
<td>0.0.6</td>
<td>5.7.0</td>
<td>v2.7.3</td>
<td>v1.11.5</td>
</tr>
<tr>
<td>kcooon-prometheus</td>
<td></td>
<td>0.0.7</td>
<td>5.7.0</td>
<td>v2.7.3</td>
<td>v1.11.5</td>
</tr>
</tbody>
</table>

*Use Case? … 000+ cluster*
07 KEMI Log-aggregator
HA reference in fluentd official documents

log forwarders

node1
- Log File
- Application
  - tail
  - push

node2
- Log File
- Application
  - tail
  - push

node3
- Log File
- Application
  - tail
  - push

log aggregators

192.168.0.1
- td-agent (active)

192.168.0.2
- td-agent (backup)
HA reference in fluentd official documents

![Diagram showing log forwarding and aggregation between nodes, with arrows indicating data flow.]

- **log forwarders**
  - node1
    - Log File
    - Application
  - node2
    - Log File
    - Application
  - node3
    - Log File
    - Application

- **log aggregators**
  - td-agent
    - 192.168.0.1
    - td-agent (active)
  - td-agent
    - 192.168.0.2
    - td-agent (backup)

Hermes
KEMI Log-aggregator – architecture

App Log
fluentd

Hermes
fluentd

KEMI Log Aggregator
fluentd

CONSUL

5m

hadoop

5s ~ 10s

Apache Kafka
KEMI Log-aggregator – consul?

Health Checks

Pairing service discovery with health checking prevents routing requests to unhealthy hosts and enables services to easily provide circuit breakers.

```
$ dig web-frontend.service.consul. ANY

; <<>> DIG 9.8.3-P1 <<>> web-frontend.service.consul. ANY
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 29981
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL

;; QUESTION SECTION:
;web-frontend.service.consul. IN ANY

;; ANSWER SECTION:
web-frontend.service.consul. 0 IN A 10.0.3.83
web-frontend.service.consul. 0 IN A 10.0.1.109
```

DNS Query Interface

Consul enables service discovery using a built-in DNS server. This allows existing applications to easily integrate, as almost all applications support using DNS to resolve IP addresses. Using DNS instead of a static IP address allows services to scale up/down and route around failures easily.
KEMI Log-aggregator – Health Check

Health check

- CPU
- CLOSE_WAIT (for kafka)

```
<store>
  @type
  @id
  expire_dns_cache 60s
<buffer>
  @type
  flush_interval
  flush_at_shutdown
  forward out_forward
  memory 5s
  true
</buffer>
<server>
  host
</server>
</store>
```
KEMI Log-aggregator – Monitoring

- monitoring with **monitor agent**

Usually alert by …
- **HDFS namenode failover**

```
curl -s :24420/api/plugins
```

![KEMI Log-Agg Health](image)
KEMI Log-aggregator – is stability?

- connection management
- fluentd file buffering is very powerful (v1.0+)
  - reprocess is also good
08 AS-IS …
09 TOBE …
- hermes fluentd self-monitoring with fluent-plugin-prometheus @ kocoon
TOBE …

Phase 1

Inside Kubernetes Cluster

Hermes
Log Aggregator

elasticsearch

kibana

kafka

Outside Kubernetes Cluster

Hive

HUE

Hadoop

kafka

KEMI
Log Tailer

KEMI
DIKE
TOBE … Phase 2
Thank you

also Thanks to cloud.telemetry
Appendix
Too many resources on master

do not use
fluent-plugin-kubernetes-metadata-filter
## compare log monitor

<table>
<thead>
<tr>
<th></th>
<th>O/KAFKA</th>
<th>O/HDFS</th>
<th>O/ES</th>
<th>Service name mgmt</th>
<th>HA</th>
</tr>
</thead>
<tbody>
<tr>
<td>fluentd</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>fluentd</td>
</tr>
<tr>
<td>fluent-bit</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>fluentd</td>
</tr>
<tr>
<td>filebeat</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>Δ</td>
<td>logstash</td>
</tr>
</tbody>
</table>