日志平台Elasticsearch升级之路

华泰证券-信息技术部　葛宝磊

目录 CONTENTS

项目背景介绍

01

项目背景介绍

2016

日志平台项目在2016下半年上线第一版，第一个接入的系统是OA办公，日志量每天不到1G，Elasticsearch选用的版本是2.4.0

2016-2018

随着接入用户的增多，数据量和用户的需求也成指数级的增加，目前已接入的应用系统有近100个，每天的日志量1T+，高峰的Elasticsearch入库速率20W+

2018-2019

ES性能瓶颈明显，通过横向扩展无法解决，需要升级Elasticsearch版本

项目背景介绍

Indexing Rate

Indexing 速率明显跟不上采集端的采集速率，特别是在9:30和下午1:00，非常明显查询日志无法实时的展现。

GC问题

分片数达到一定数量后，频繁的full GC，导致集群不响应

索引创建失败导致堆内存无法释放，需要定时重启

CPU告警

业务高峰期个别节点CPU使用率超过90%，影响整个集群的性能

数据恢复问题

当某个节点或者集群需要重启时，耗时时间长

cluster.routing 不支持节点级别的配置

磁盘告警

机器中部分服务器采用的裸盘的方式挂载，经常会出现部分磁盘占用率超过70%情况，而其他盘利用率低于50%

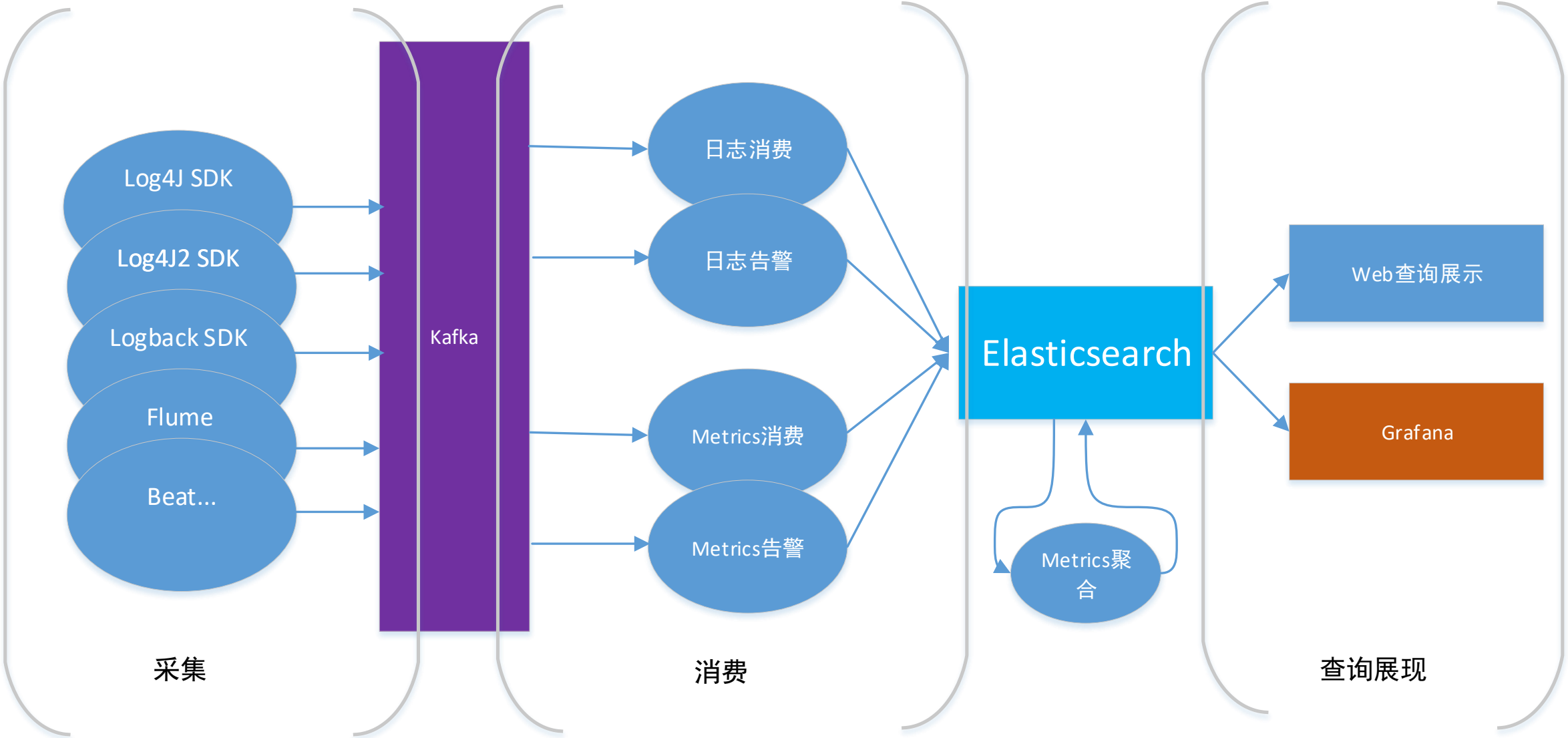
Query内存占用

ES使用方查询海量数据时，容易导致OOM

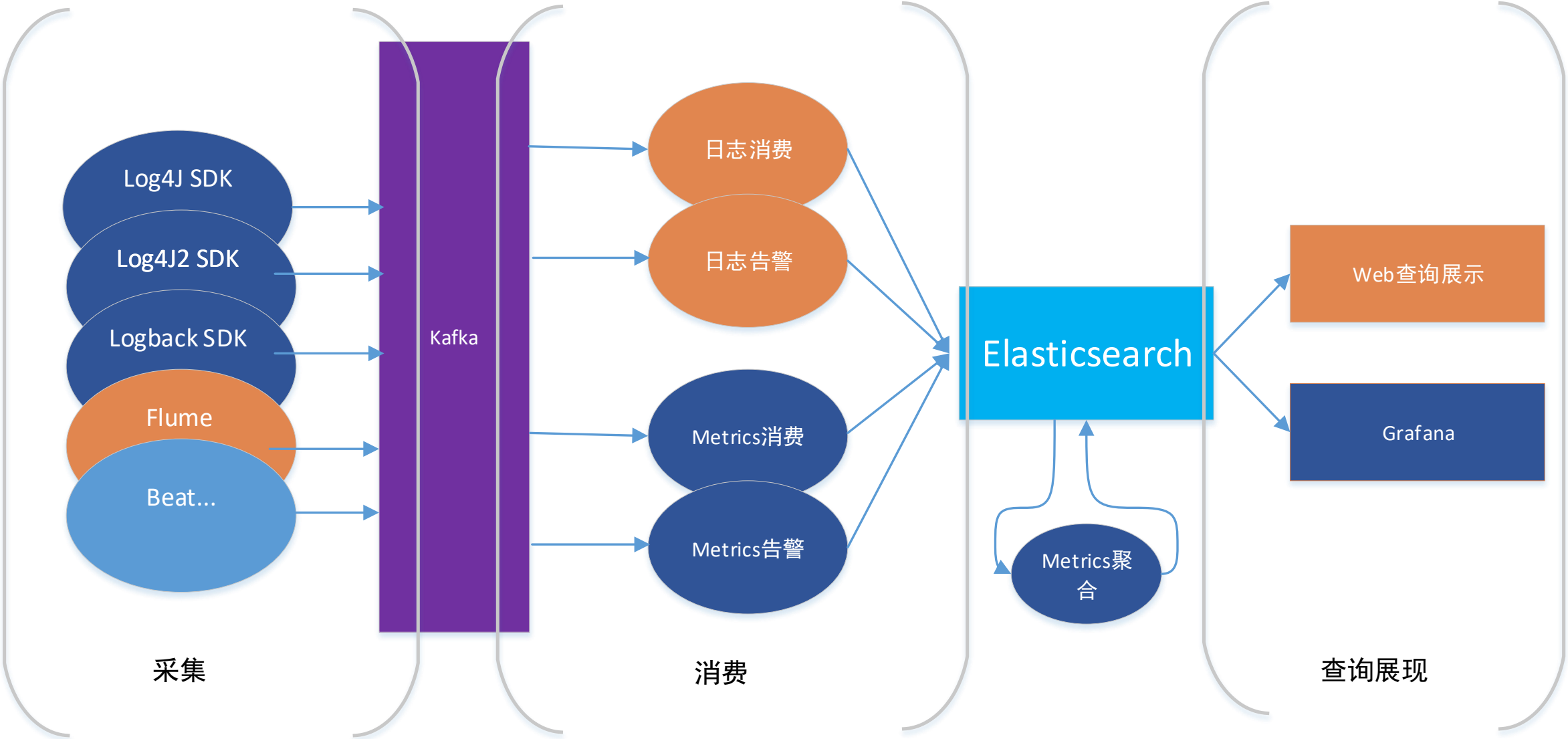
关键系统设计

02

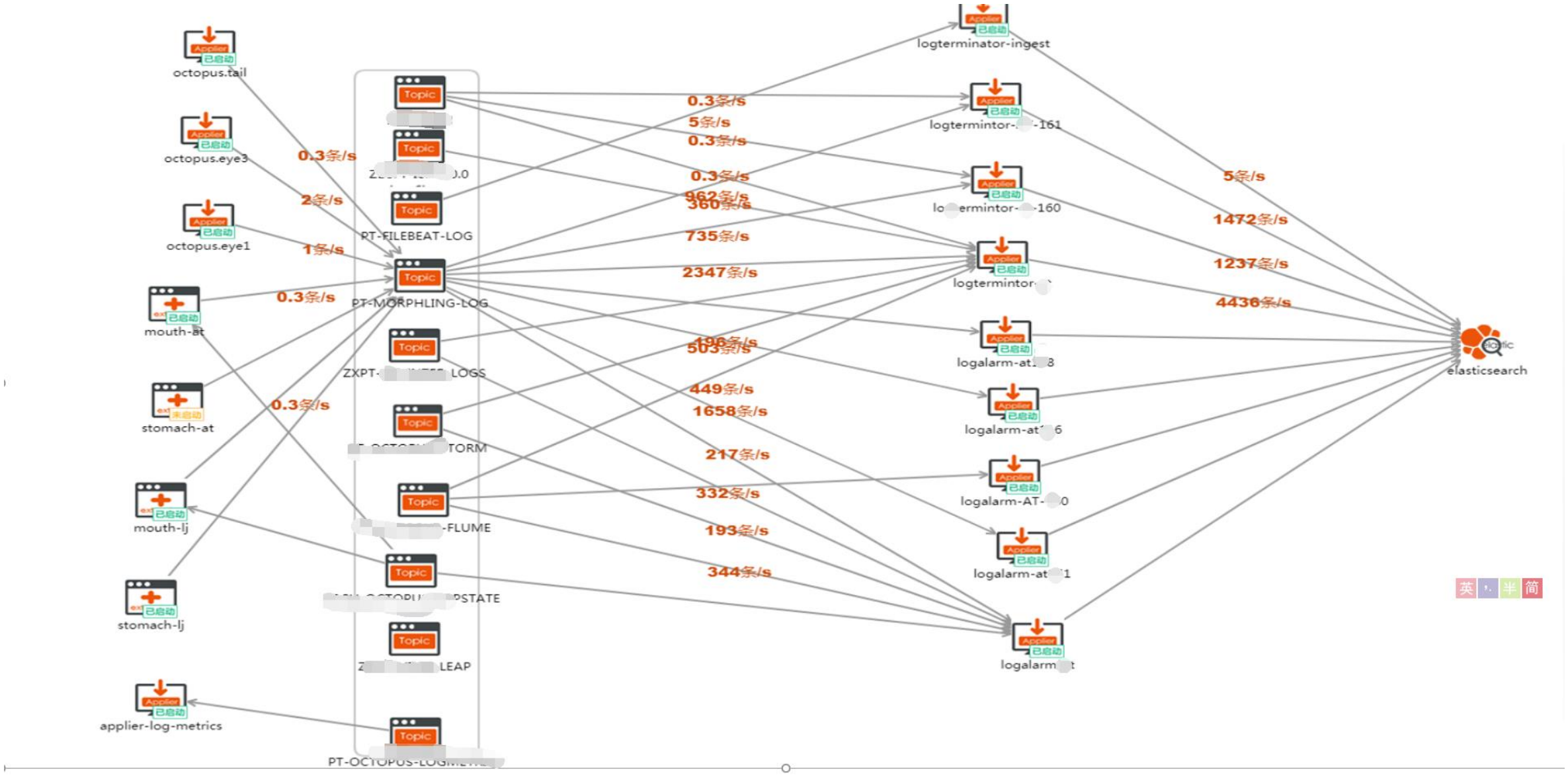
关键系统设计 – 系统架构


Log4J SDK
Log4J2 SDK
Logback SDK
Flume
Beat...
Kafka
日志消费
日志告警
Metrics消费
Metrics告警
Elasticsearch
Metrics聚合
Web查询展示
Grafana
采集
消费
查询展现

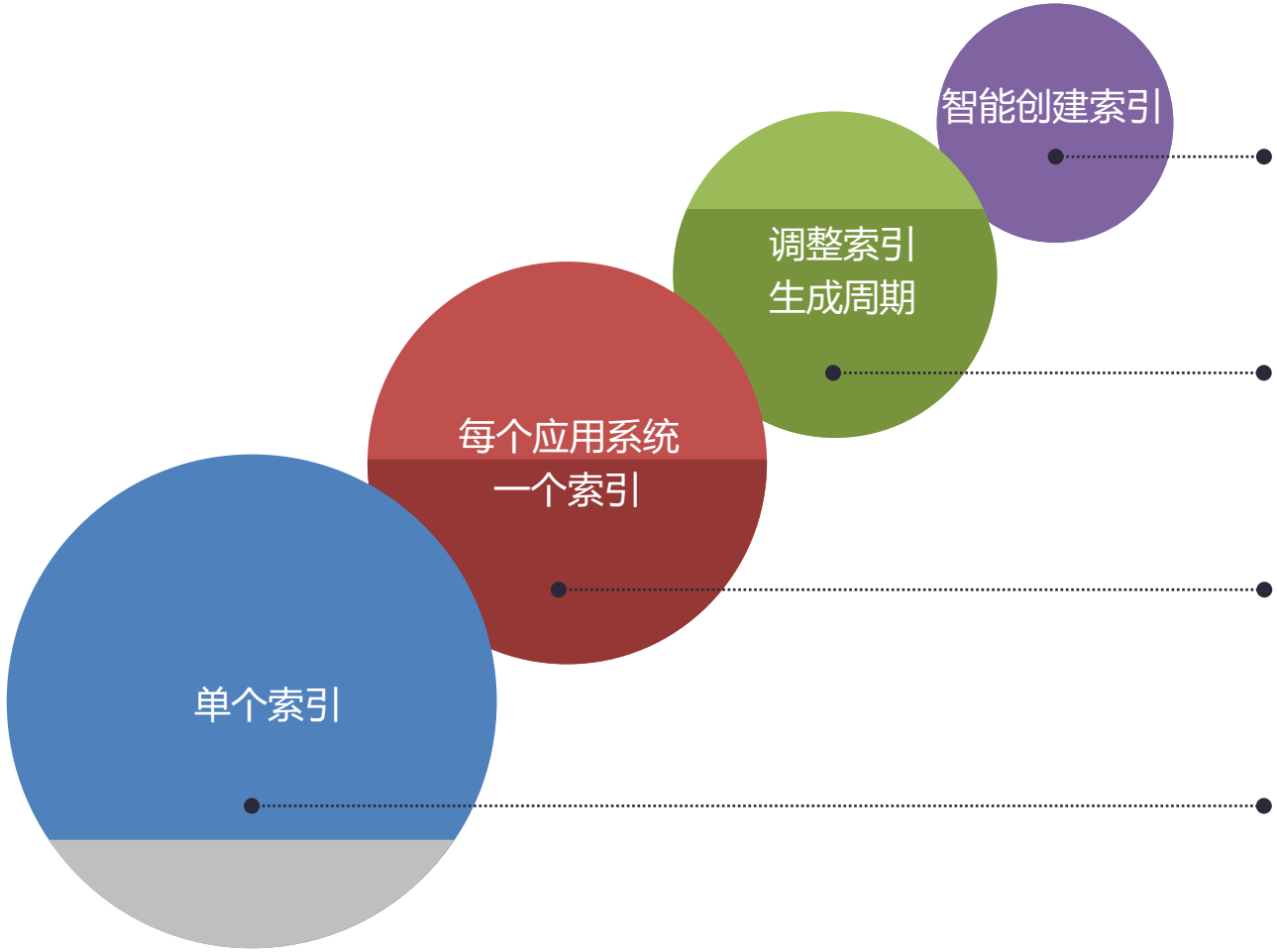
关键系统设计 – 系统架构


Log4J SDK
Log4J2 SDK
Logback SDK
Flume
Beat...
Kafka
日志消费
日志告警
Metrics消费
Metrics告警
Elasticsearch
Metrics聚合
Web查询展示
Grafana
采集
消费
查询展现

关键系统设计 – 系统架构


octopus.tail
octopus.eye3
octopus.eye1
mouth-at
stomach-at
mouth-lj
stomach-lj
applier-log-metrics
0.3条/s
2条/s
1条/s
0.3条/s
0.3条/s
Topic
PT-FILEBEAT-LOG
PT-MORPHLING-LOG
ZXPT-
LOGS
TORM
-FLUME
PSTATE
LEAP
PT-OCTOPUS-LOGMETRICS
0.3条/s
5条/s
0.3条/s
0.3条/s
962条/s
360条/s
735条/s
2347条/s
503条/s
449条/s
1658条/s
217条/s
332条/s
193条/s
344条/s
logterminator-ingest
logtermintor-
-161
-160
logtermintor-
logalarm-at
8
logalarm-at
6
logalarm-AT-
0
logalarm-at
1
logalarm
t
5条/s
1472条/s
1237条/s
4436条/s
elasticsearch

关键系统设计 – 索引设计



智能创建索引

每周统计应用系统的日志量，根据上周的日志量大小决定下周写入索引生成周期。

调整索引生成周期

根据应用的预估日志量按不同的时间间隔创建索引，支持按天，按周，按月，按年创建索引

优点： 索引量适中，分片数适中，查询效率高

缺点：对于突发情况或者应用日志量发生较大变更无法自适应。

每个应用系统每天创建一个索引

每个应用系统每天创建一个索引

优点：索引管理比较方便，查询效率较高

缺点：索引数量多，分片数太多， jvm使用率高，恢复慢

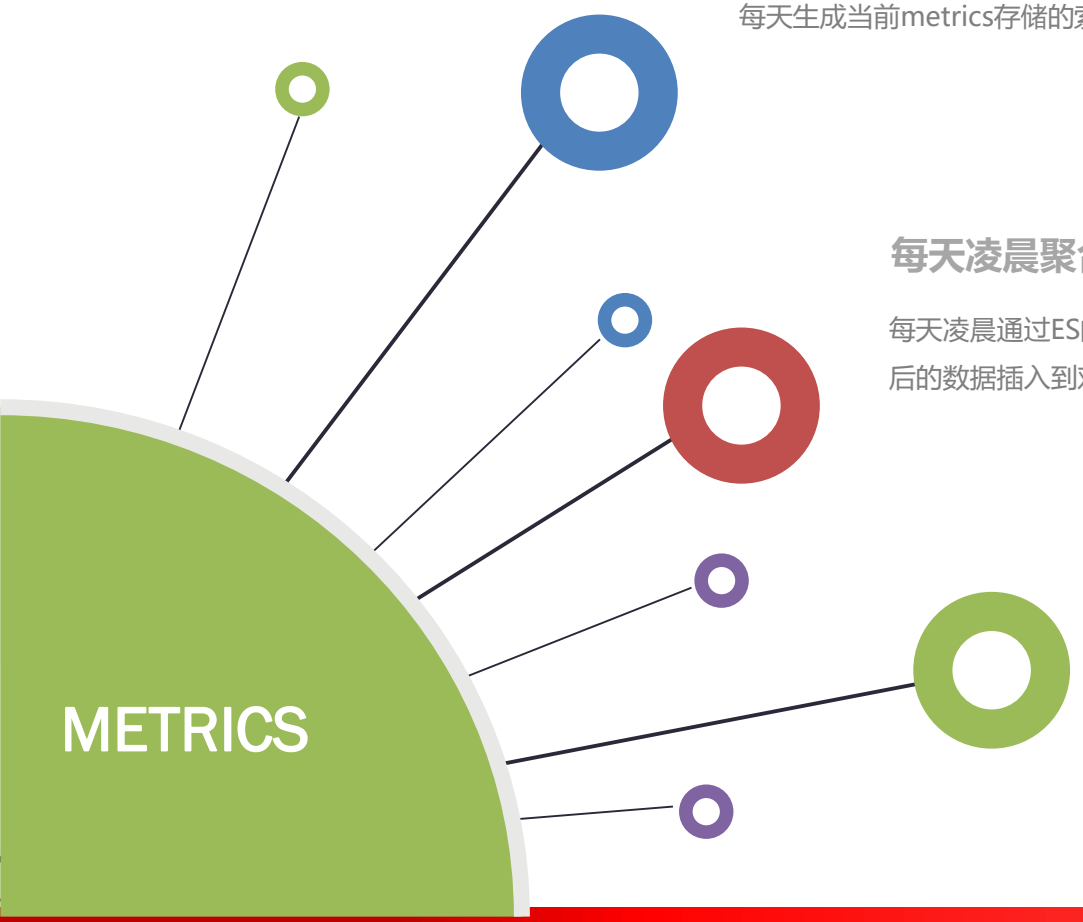
每天创建一个索引

按天创建索引，接入的应用系统通过Type来区分

优点： 索引量小，管理方便

缺点： 单索引数据量太大，查询效率低，各应用系统日志的留存时间必须保持一致。

关键系统设计 – 索引设计



每天生成一个Metrics索引

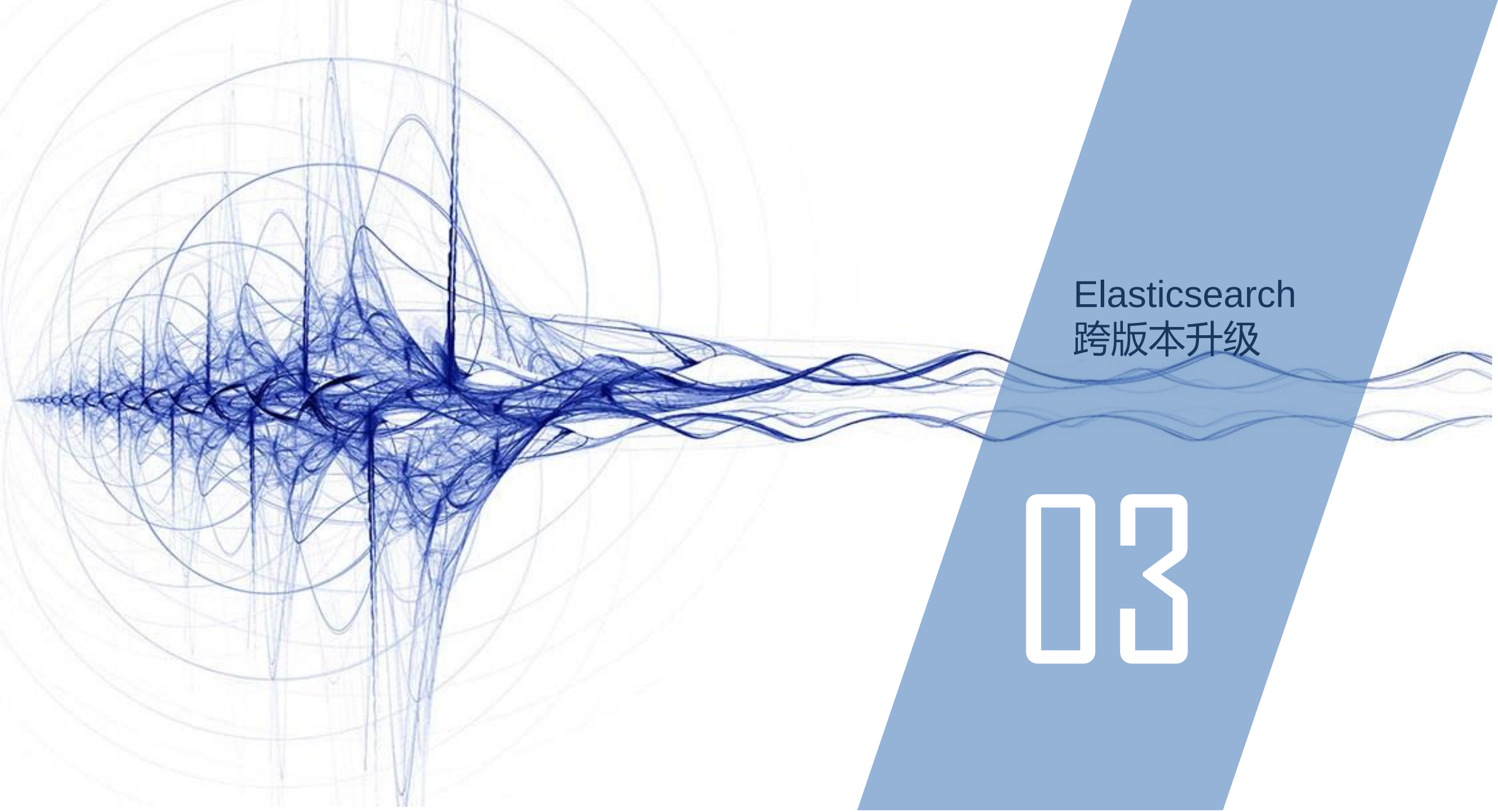
每天生成当前metrics存储的索引

每天凌晨聚合昨天的索引

每天凌晨通过ES的聚合操作，将前一天的Metrics数据按照1m，10m，1h等时间间隔进行聚合，将聚合后的数据插入到对应的索引上，1m、10m的聚合索引按月创建，1h的索引按照年创建

查询自动感知对应的索引

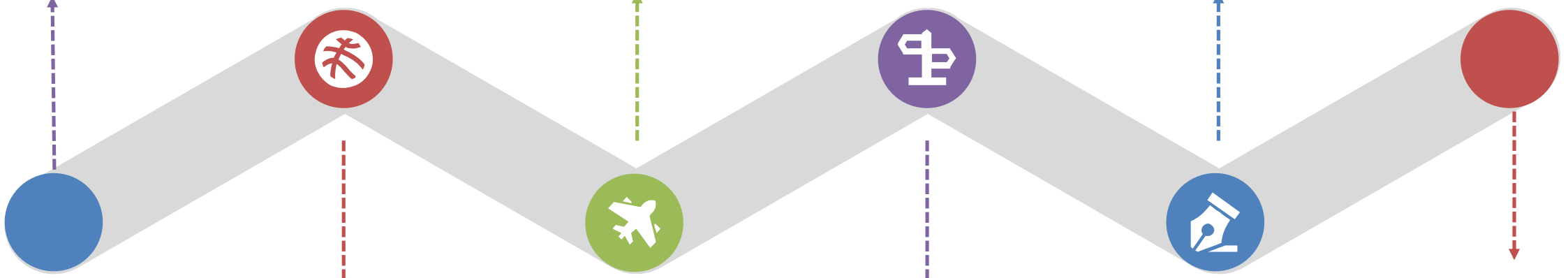
指标展现会智能查询对应时间段内最小粒度的Metrics数据，保证数据更准确



Elasticsearch 跨版本升级

03

跨版本升级 – 升级步骤



调研

集群搭建

确认服务器数量
确认节点的数量
确认节点的类型

索引模板升级

确认原来字段类型是否支持
确认分词类型

代码编写

使用官方推荐的RestHighLevelClient
确定各模块开发顺序

数据迁移

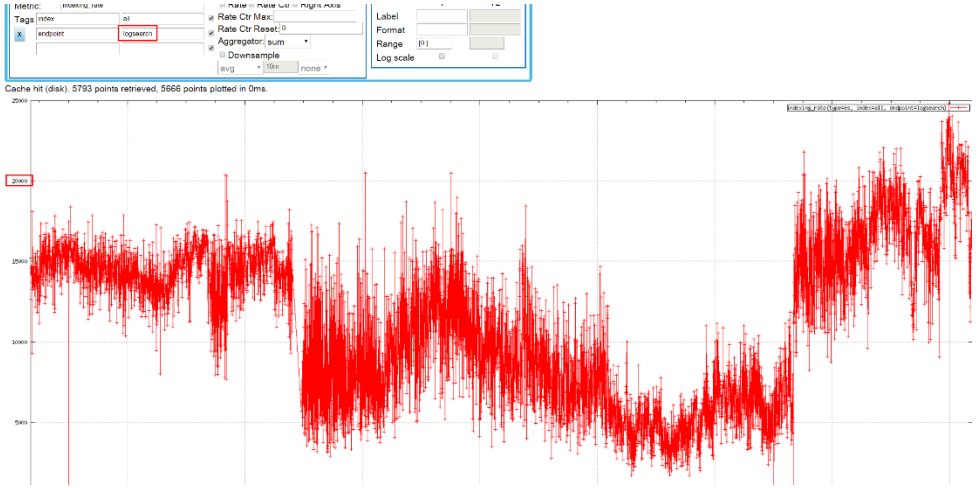
哪些数据需要迁移
不需要迁移的数据怎么处理
需要迁移的数据怎么处理
如何快速有效的完成迁移

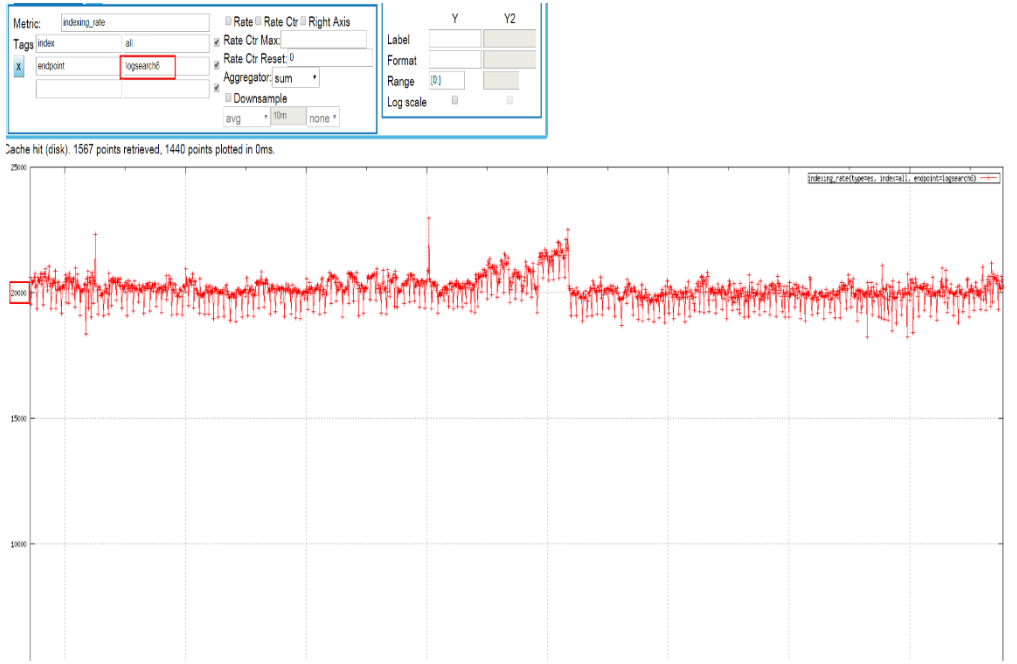
切换

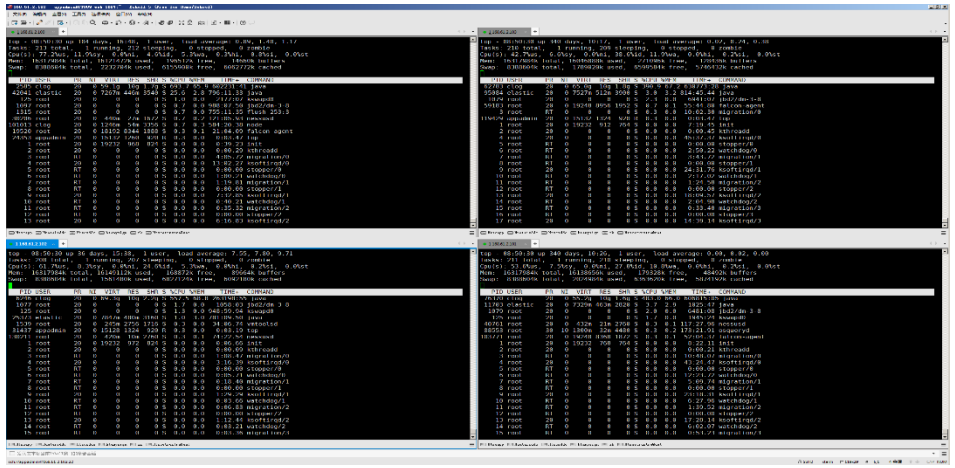
系统的无缝切换

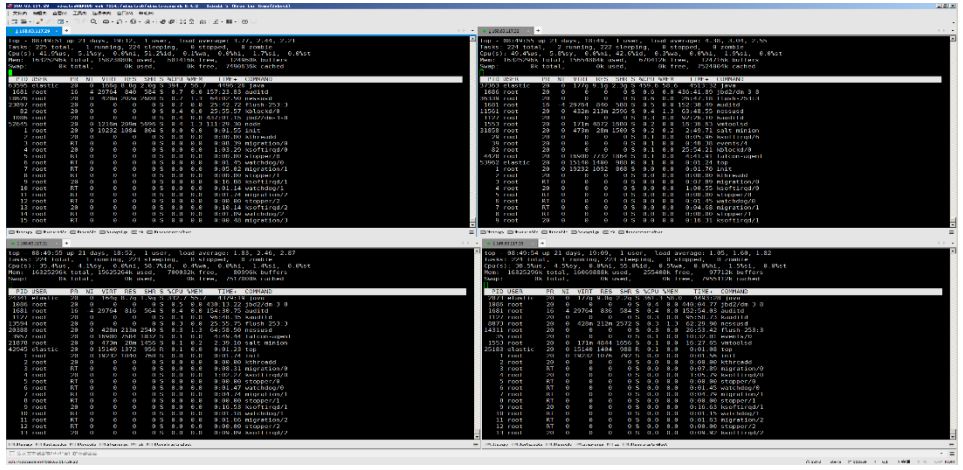
跨版本升级 – 集群搭建

性能对比



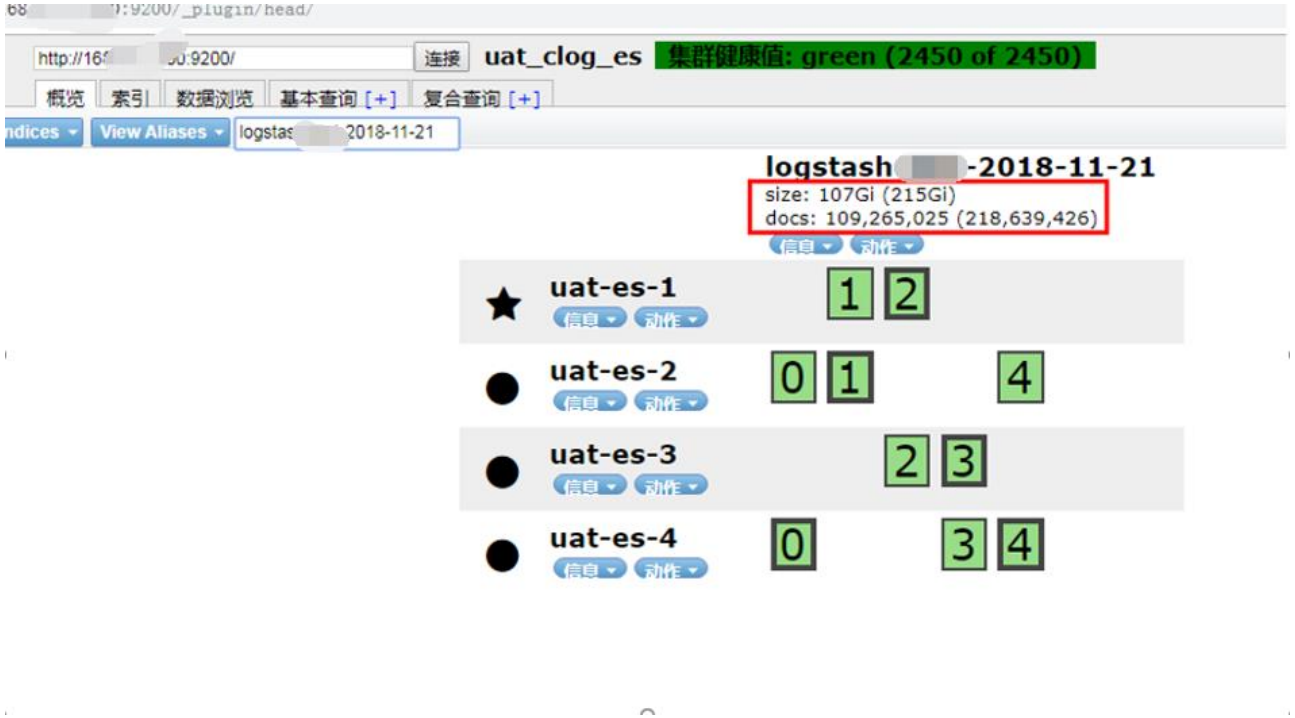




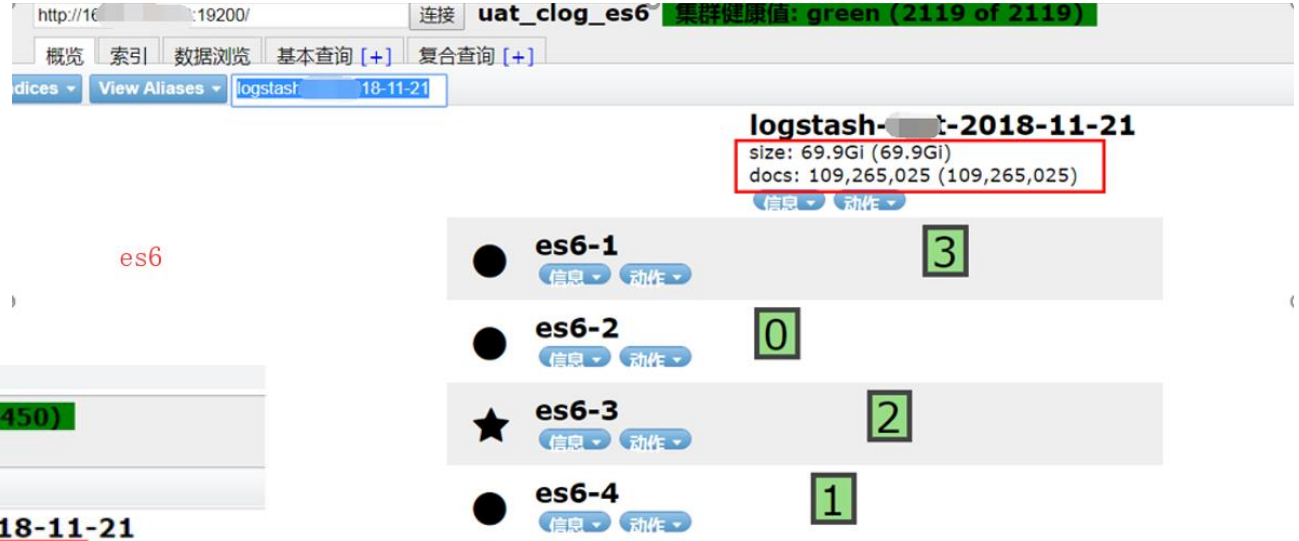


跨版本升级 – 集群搭建

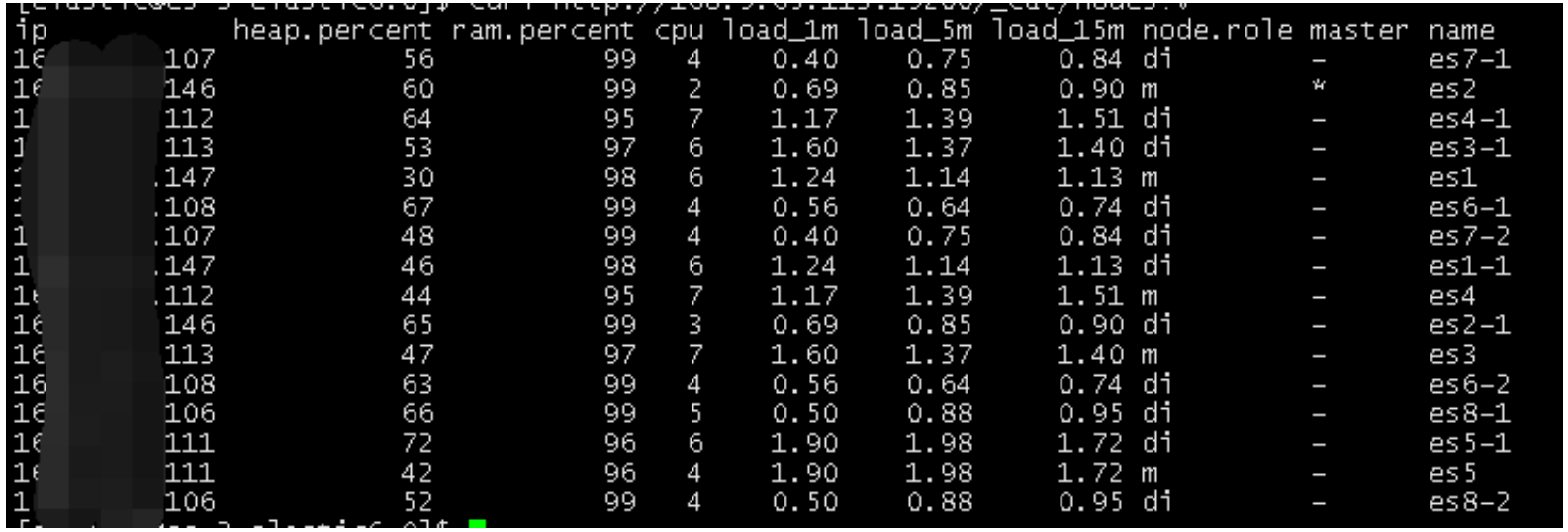
磁盘占用对比



es6



跨版本升级 – 集群搭建

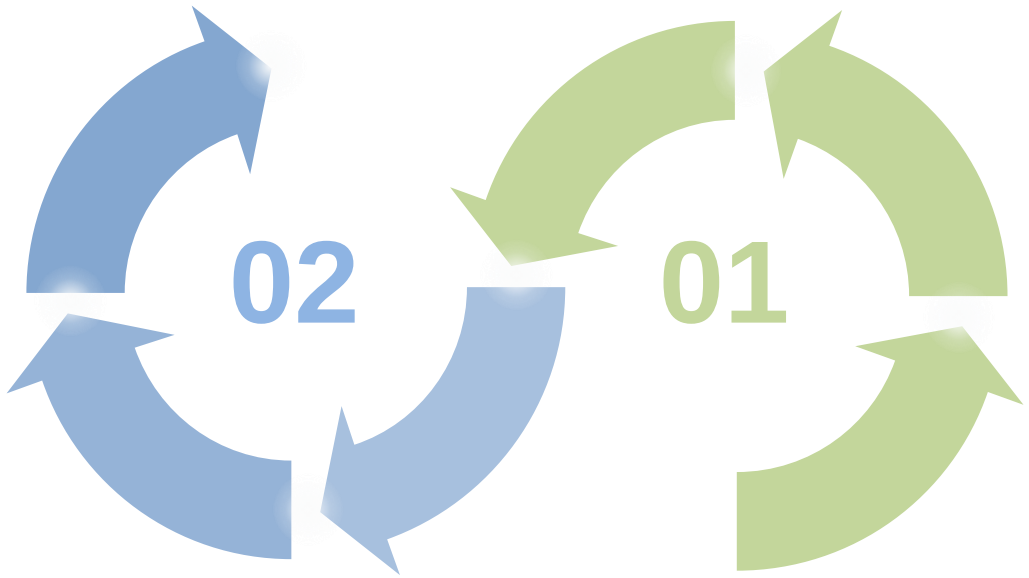


ip	heap.percent	ram.percent	cpu	load_1m	load_5m	load_15m	node.role	master	name
16 107	56	99	4	0.40	0.75	0.84	di	-	es7-1
16 146	60	99	2	0.69	0.85	0.90	m	*	es2
1 112	64	95	7	1.17	1.39	1.51	di	-	es4-1
1 113	53	97	6	1.60	1.37	1.40	di	-	es3-1
1 147	30	98	6	1.24	1.14	1.13	m	-	es1
1 108	67	99	4	0.56	0.64	0.74	di	-	es6-1
1 107	48	99	4	0.40	0.75	0.84	di	-	es7-2
1 147	46	98	6	1.24	1.14	1.13	di	-	es1-1
1 112	44	95	7	1.17	1.39	1.51	m	-	es4
16 146	65	99	3	0.69	0.85	0.90	di	-	es2-1
16 113	47	97	7	1.60	1.37	1.40	m	-	es3
16 108	63	99	4	0.56	0.64	0.74	di	-	es6-2
16 106	66	99	5	0.50	0.88	0.95	di	-	es8-1
16 111	72	96	6	1.90	1.98	1.72	di	-	es5-1
16 111	42	96	4	1.90	1.98	1.72	m	-	es5
1 106	52	99	4	0.50	0.88	0.95	di	-	es8-2

原来： 6台服务器，3个master节点，18的data节点， 70T存储空间
升级： 5台服务器，5个master节点， 5(11)个data节点， 60T存储空间

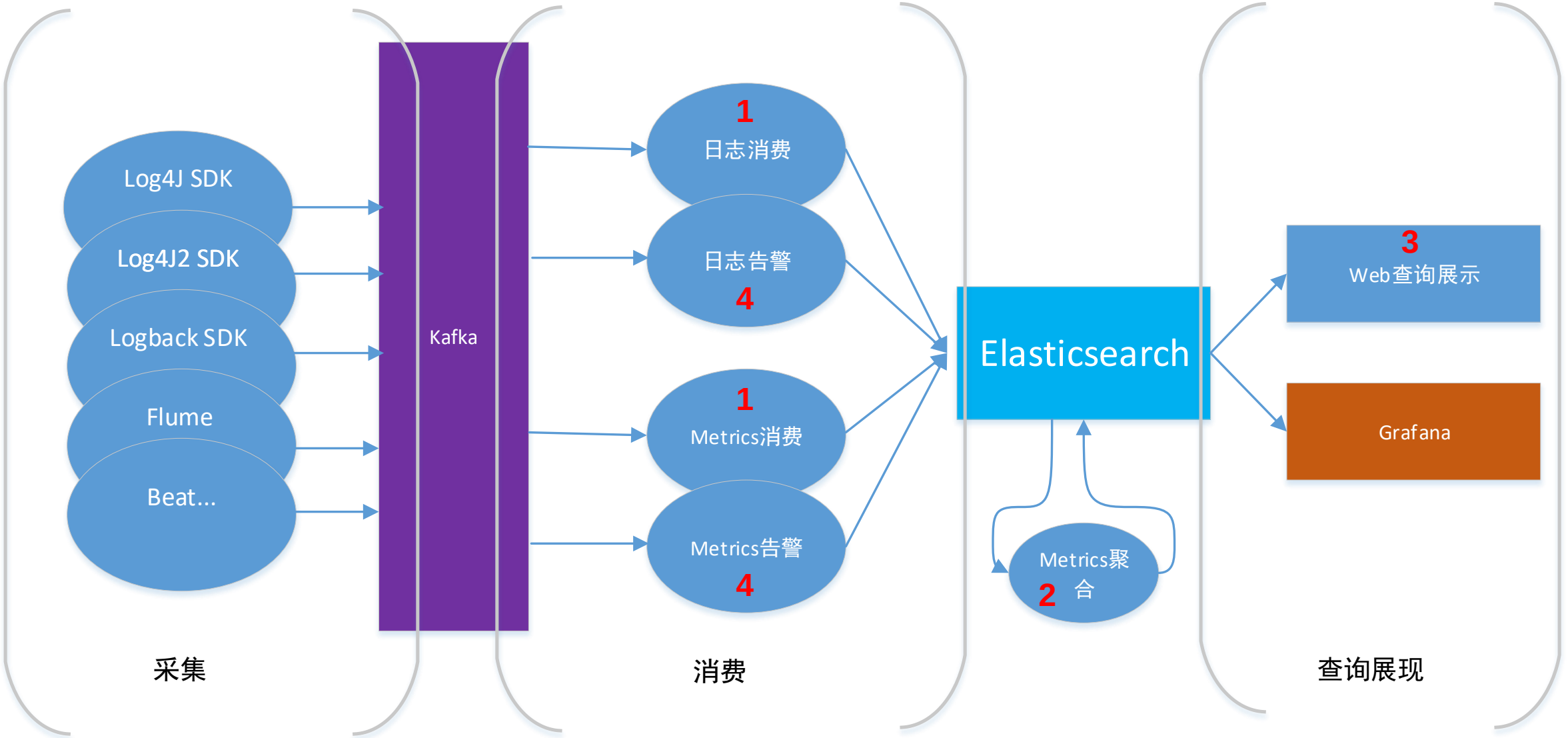
跨版本升级 – Template升级

字段类型

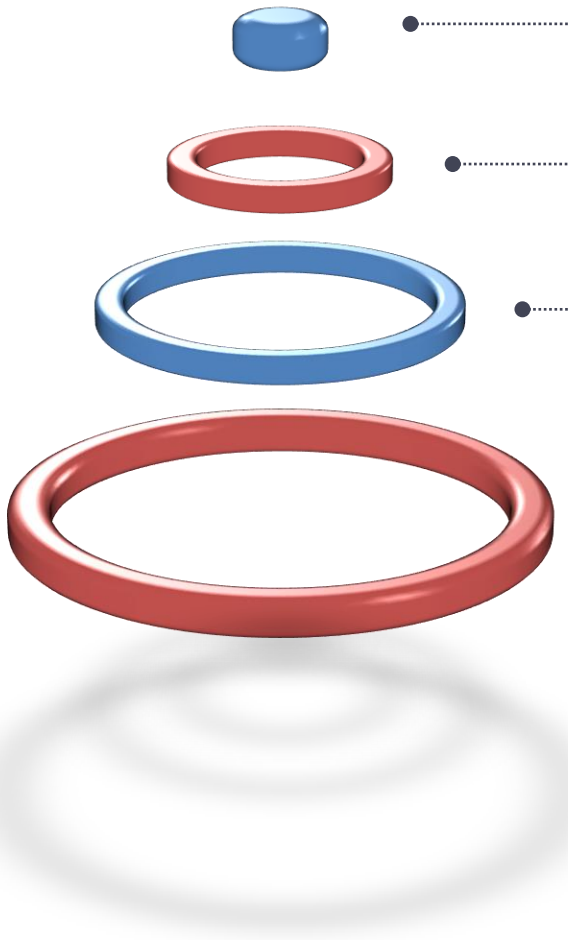


索引Type

跨版本升级 – 代码改造


Log4J SDK
Log4J2 SDK
Logback SDK
Flume
Beat...
Kafka
1
日志消费
日志告警
4
1
Metrics消费
Metrics告警
4
Elasticsearch
Metrics聚合
2
3
Web查询展示
Grafana
采集
消费
查询展现

跨版本升级 – 数据迁移



哪些数据需要迁移

大部分日志数据和指标数据都是有时效性的

不需要迁移部分如何处理

在升级过程中会过期的数据是需要迁移，让自然消亡

需要迁移数据如何处理

要有快速有效的迁移方案，自己造轮子还是使用已有的轮子?

如何快速有效完成迁移

靠人工来一个一个索引进行迁移是几乎不可能完成的事情，工作量大，也容易犯错

跨版本升级 – 数据迁移

Reindex API

edit

IMPORTANT

Reindex requires `_source` to be enabled for all documents in the source index.

IMPORTANT

Reindex does not attempt to set up the destination index. It does not copy the settings of the source index. You should set up the destination index prior to running a `_reindex` action, including setting up mappings, shard counts, replicas, etc.

The most basic form of `_reindex` just copies documents from one index to another. This will copy documents from the `twitter` index into the `new_twitter` index:

```
POST _reindex
{
  "source": {
    "index": "twitter"
  },
  "dest": {
    "index": "new_twitter"
  }
}
```

COPY AS CURL VIEW IN CONSOLE

Reindex from Remote

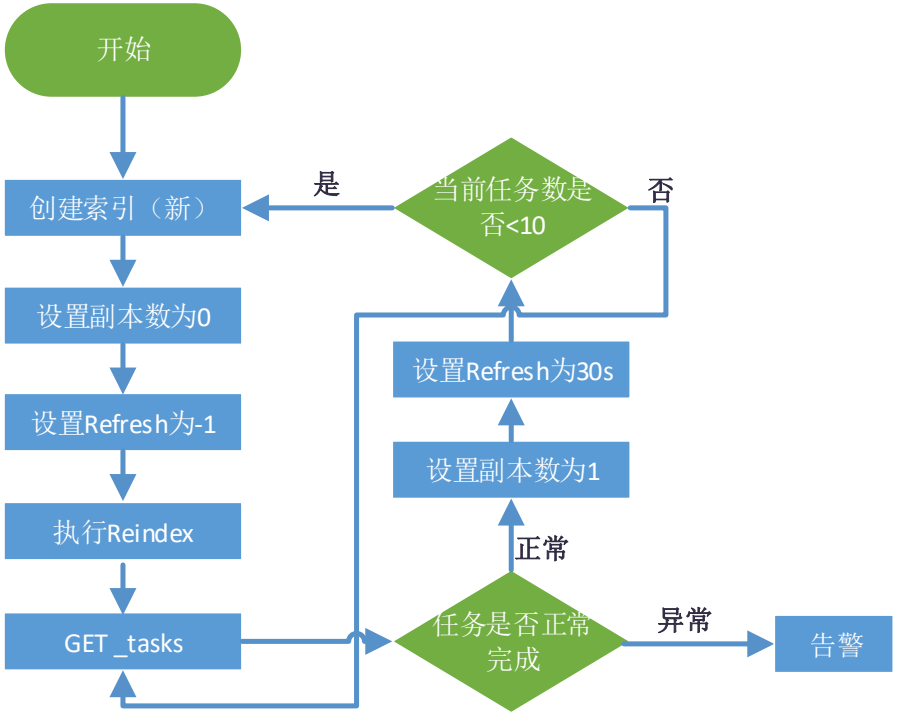
edit

Reindex supports reindexing from a remote Elasticsearch cluster:

```
POST _reindex
{
  "source": {
    "remote": {
      "host": "http://otherhost:9200",
      "username": "user",
      "password": "pass"
    },
    "index": "source",
    "query": {
      "match": {
        "test": "data"
      }
    }
  },
  "dest": {
    "index": "dest"
  }
}
```

COPY AS CURL VIEW IN CONSOLE

跨版本升级 – 数据迁移



1. 创建索引

2. 设置副本数为0
PUT xxxx_index/_settings { "number_of_replicas": 0 }

3. 设置refresh时间间隔为不刷新
PUTxxxx_index/_settings { "refresh_interval": -1 }

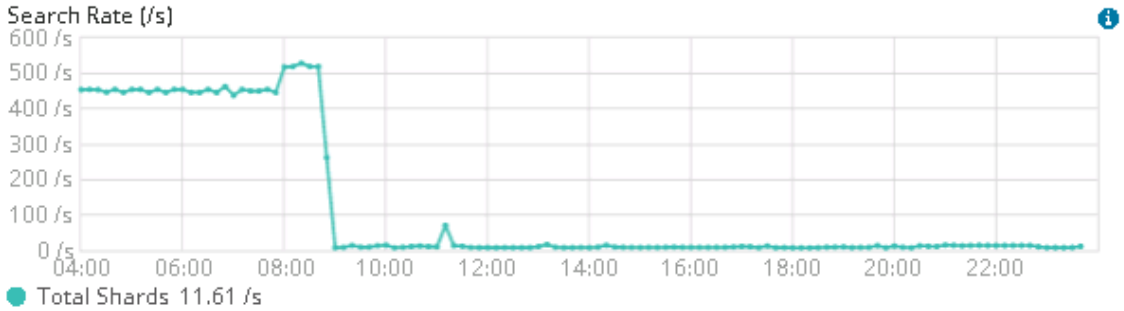
4. 执行reindex操作

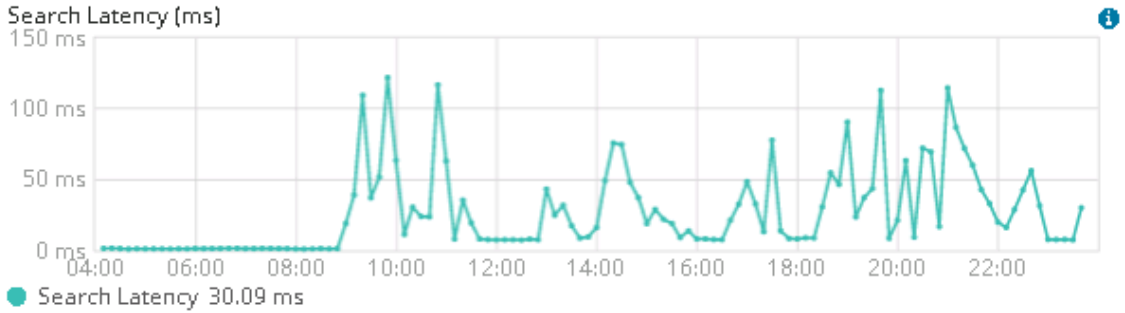
5. 定时通过task api 记录正在迁移的任务
GET _tasks?detailed=true&actions=*reindex

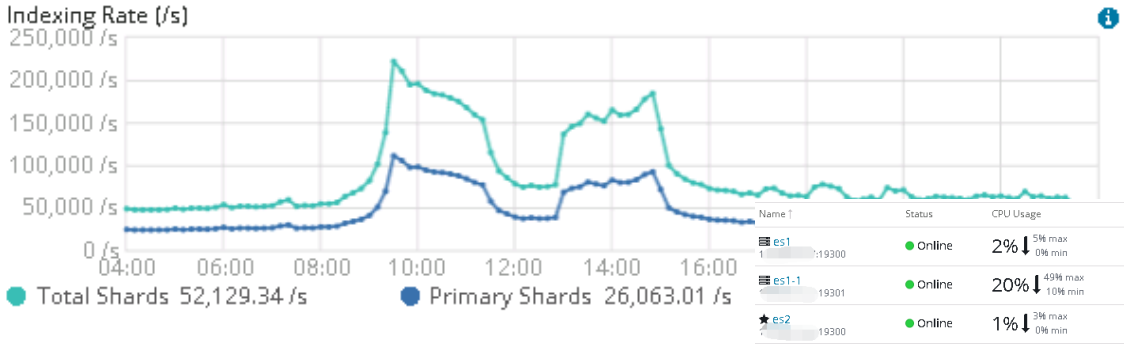
6. 判断已经完成的任务所对应的索引 日志条数是否一致。一致，设置副本数为1， refresh间隔为30s，否则告警。

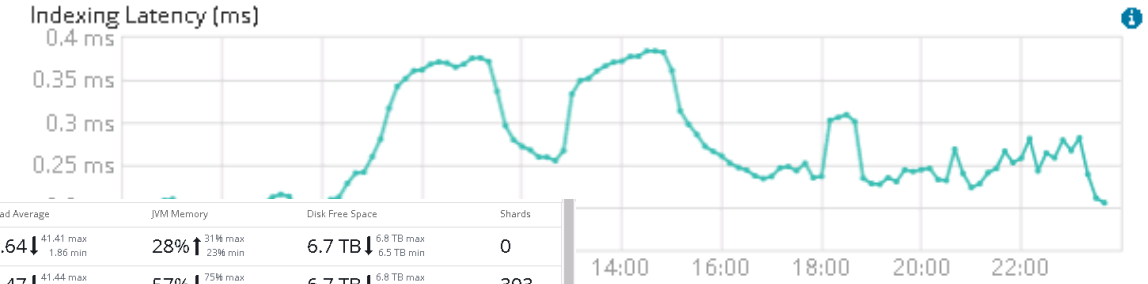
7.任务数小于10，并且存在需要迁移的索引，继续进行1操作

跨版本升级 – 现有集群









Name	Status	CPU Usage	Load Average	JVM Memory	Disk Free Space	Shards
es1	Online	2%	4.64	28%	6.7 TB	0
es1-1	Online	20%	4.47	57%	6.7 TB	393
es2	Online	1%	3.93	74%	6.8 TB	0
es2-1	Online	14%	3.77	74%	6.8 TB	392
es3	Online	2%	6.41	61%	9.3 TB	0
es3-1	Online	18%	6.79	60%	9.3 TB	393
es4	Online	2%	5.64	56%	9.3 TB	0
es4-1	Online	21%	5.64	54%	9.3 TB	393
es5	Online	2%	3.05	45%	9.1 TB	0
es5-1	Online	20%	2.88	50%	9.1 TB	393
es6-1	Online	5%	1.75	52%	2.5 TB	484
es6-2	Online	7%	1.75	59%	2.5 TB	484
es7-1	Online	5%	3.42	54%	2.5 TB	485
es7-2	Online	4%	3.8	49%	2.5 TB	484
es8-1	Online	6%	2.57	53%	2.3 TB	483
es8-2	Online	6%	2.57	64%	2.5 TB	484

坑和运维

04

坑

A

GC问题

内存的设置和ES2不通，
导致采用的默认内存大小

B

Bulk操作报错

ES写入压力过大时，ES
会报错写入操作

C

Jvm使用率

随着分片数的增多，jvm
的使用率偏高，GC的越来
越频繁

D

磁盘占用

裸盘的节点还是会出现使
用率差距很大的问题导致
告警

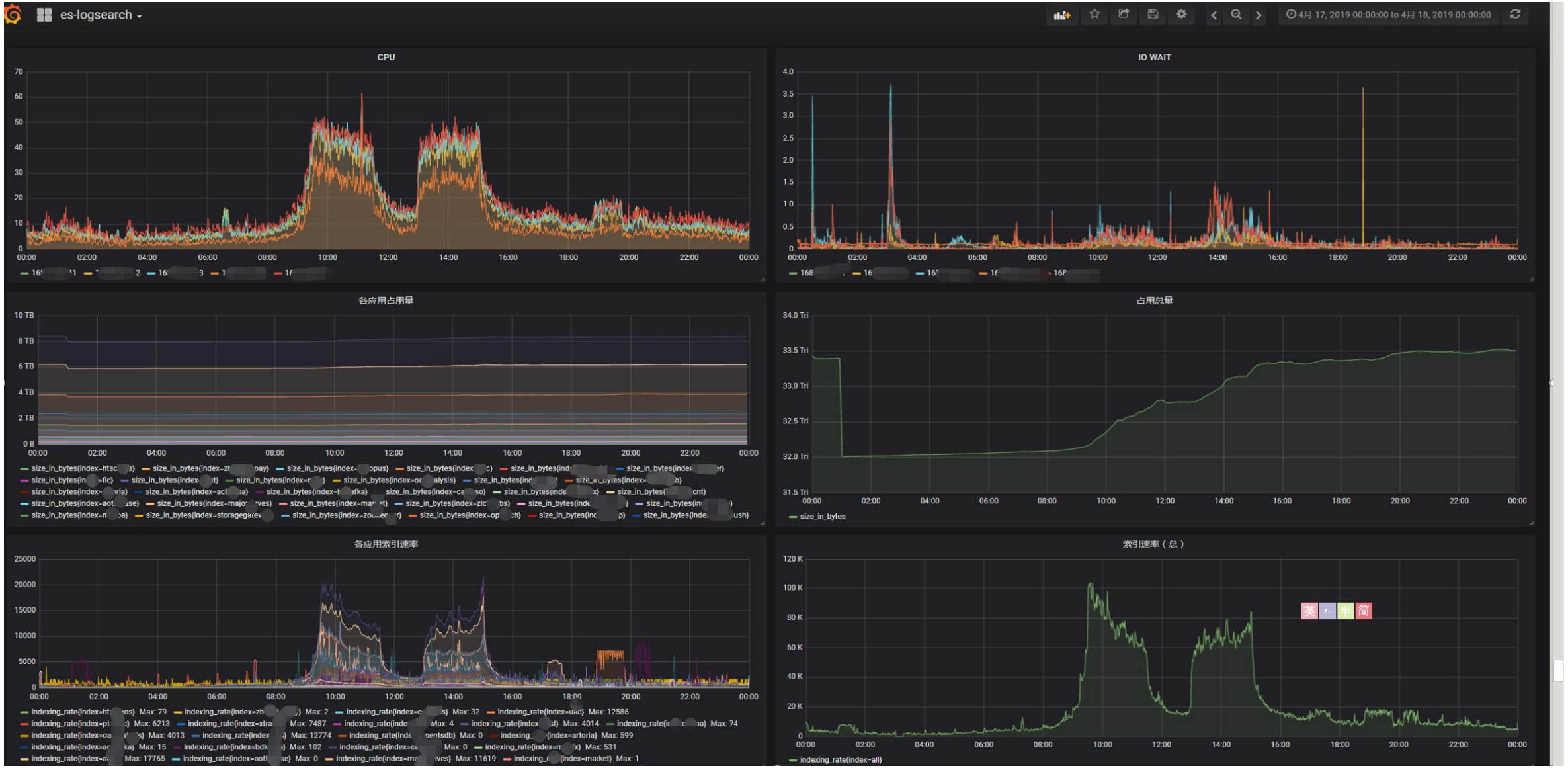
```
# Xms represents the initial size of total heap space
# Xmx represents the maximum size of total heap space

-Xms30g
-Xmx30g
```

```
indices.fielddata.cache.size: 1gb
indices.queries.cache.size: 1gb
```

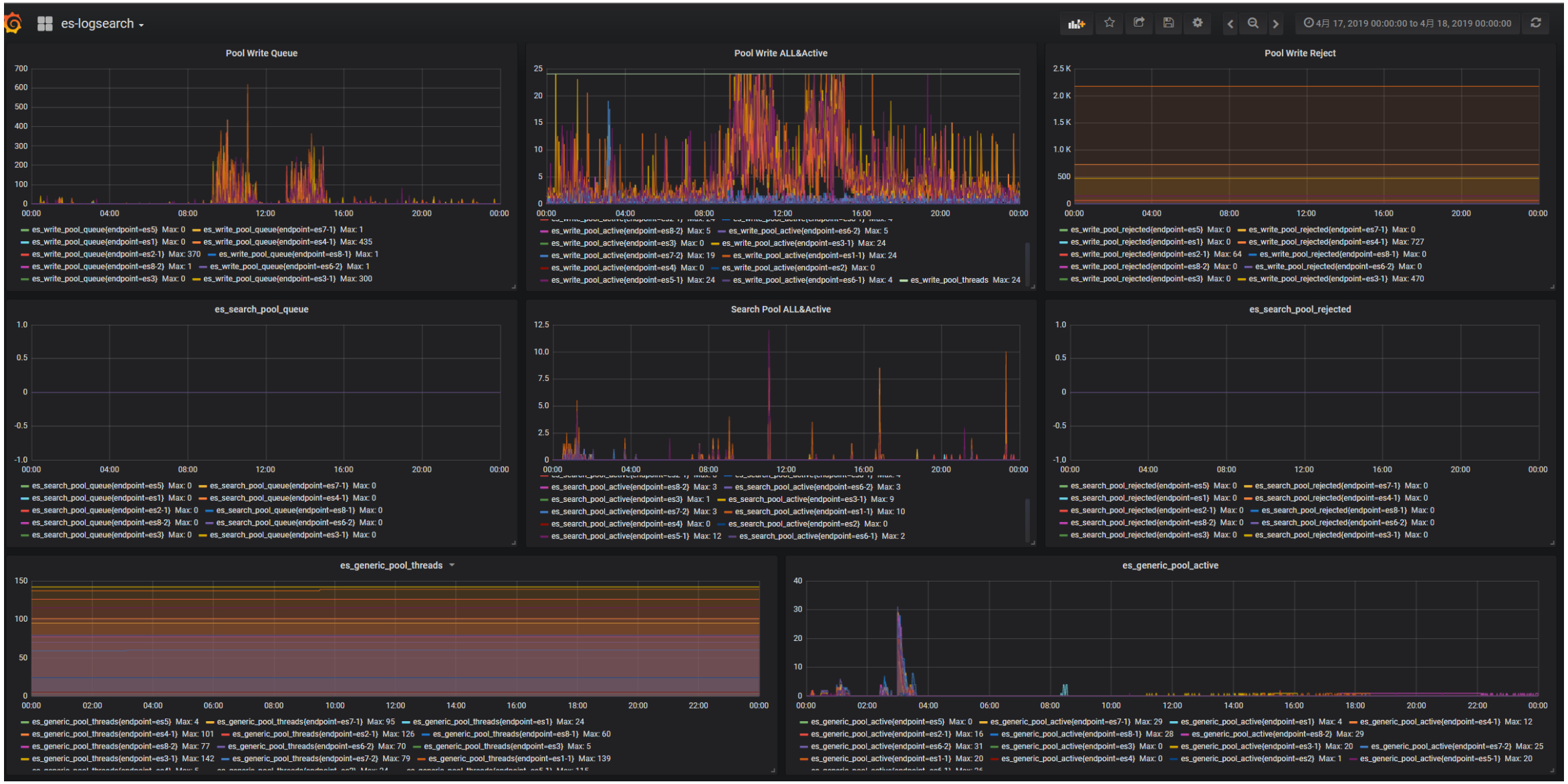
```
thread_pool:
    write:
        size: 24
        queue_size: 2000
```

```
"cluster": {
  "routing": {
    "allocation": {
      "disk": {
        "watermark": {
          "low": "75%",
          "high": "80%"
        }
      }
    }
```

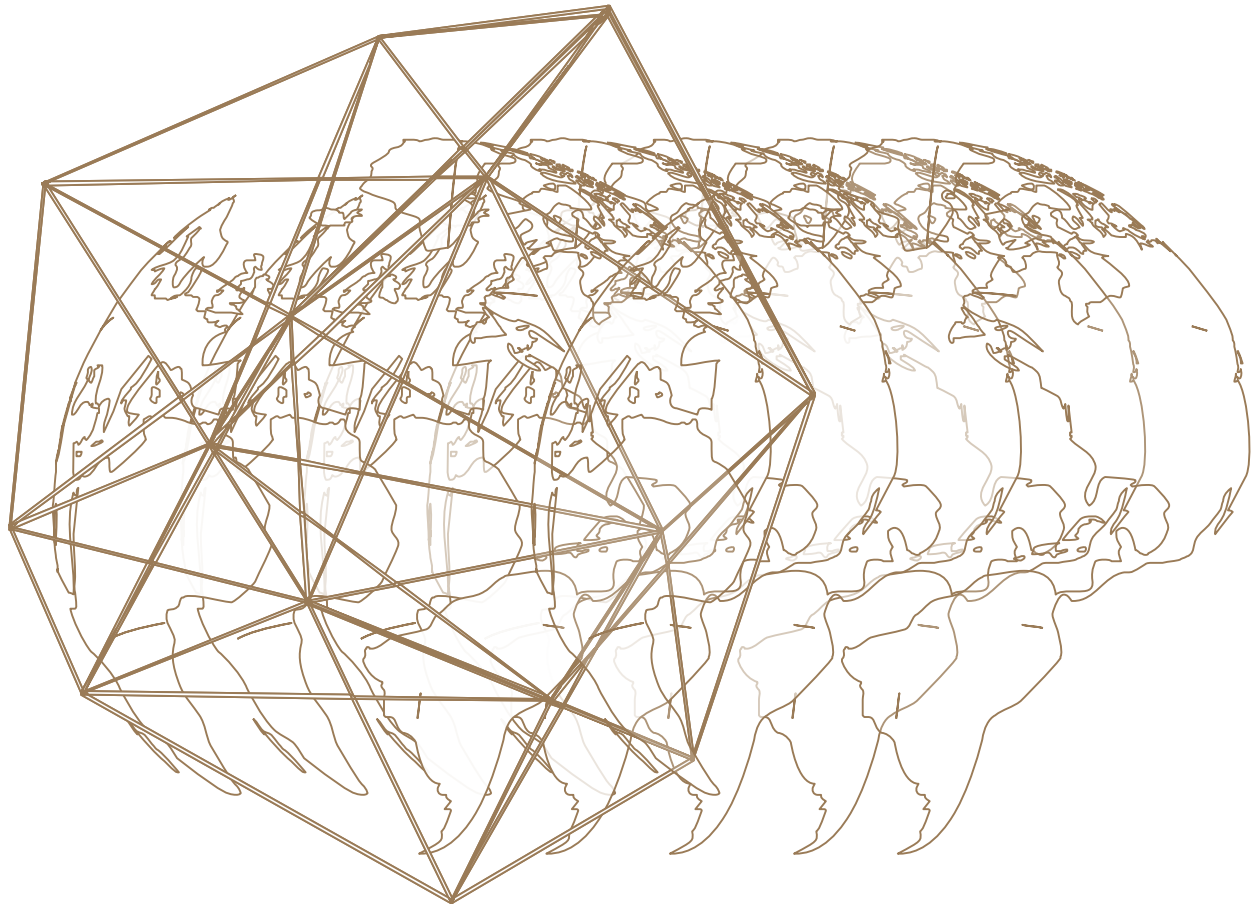
运维


es-logsearch
4月 17, 2019 00:00:00 to 4月 18, 2019 00:00:00
CPU
IO WAIT
各应用占用量
占用总量
size_in_bytes
各应用索引速率
索引速率（总）
indexing_rate{index=all}

运维


es-logsearch
4月 17, 2019 00:00:00 to 4月 18, 2019 00:00:00
Pool Write Queue
es_write_pool_queue{endpoint=es5} Max: 0
es_write_pool_queue{endpoint=es7-1} Max: 1
es_write_pool_queue{endpoint=es1} Max: 0
es_write_pool_queue{endpoint=es4-1} Max: 435
es_write_pool_queue{endpoint=es2-1} Max: 370
es_write_pool_queue{endpoint=es8-1} Max: 1
es_write_pool_queue{endpoint=es8-2} Max: 1
es_write_pool_queue{endpoint=es6-2} Max: 1
es_write_pool_queue{endpoint=es3} Max: 0
es_write_pool_queue{endpoint=es3-1} Max: 300
Pool Write ALL&Active
es_write_pool_active{endpoint=es8-2} Max: 5
es_write_pool_active{endpoint=es6-2} Max: 5
es_write_pool_active{endpoint=es3} Max: 0
es_write_pool_active{endpoint=es3-1} Max: 24
es_write_pool_active{endpoint=es7-2} Max: 19
es_write_pool_active{endpoint=es1-1} Max: 24
es_write_pool_active{endpoint=es4} Max: 0
es_write_pool_active{endpoint=es2} Max: 0
es_write_pool_active{endpoint=es5-1} Max: 24
es_write_pool_active{endpoint=es6-1} Max: 4
es_write_pool_threads Max: 24
Pool Write Reject
es_write_pool_rejected{endpoint=es5} Max: 0
es_write_pool_rejected{endpoint=es7-1} Max: 0
es_write_pool_rejected{endpoint=es1} Max: 0
es_write_pool_rejected{endpoint=es4-1} Max: 727
es_write_pool_rejected{endpoint=es2-1} Max: 64
es_write_pool_rejected{endpoint=es8-1} Max: 0
es_write_pool_rejected{endpoint=es8-2} Max: 0
es_write_pool_rejected{endpoint=es6-2} Max: 0
es_write_pool_rejected{endpoint=es3} Max: 0
es_write_pool_rejected{endpoint=es3-1} Max: 470
es_search_pool_queue
es_search_pool_queue{endpoint=es5} Max: 0
es_search_pool_queue{endpoint=es7-1} Max: 0
es_search_pool_queue{endpoint=es1} Max: 0
es_search_pool_queue{endpoint=es4-1} Max: 0
es_search_pool_queue{endpoint=es2-1} Max: 0
es_search_pool_queue{endpoint=es8-1} Max: 0
es_search_pool_queue{endpoint=es8-2} Max: 0
es_search_pool_queue{endpoint=es6-2} Max: 0
es_search_pool_queue{endpoint=es3} Max: 0
es_search_pool_queue{endpoint=es3-1} Max: 0
Search Pool ALL&Active
es_search_pool_active{endpoint=es8-2} Max: 3
es_search_pool_active{endpoint=es6-2} Max: 3
es_search_pool_active{endpoint=es3} Max: 1
es_search_pool_active{endpoint=es3-1} Max: 9
es_search_pool_active{endpoint=es7-2} Max: 3
es_search_pool_active{endpoint=es1-1} Max: 10
es_search_pool_active{endpoint=es4} Max: 0
es_search_pool_active{endpoint=es2} Max: 0
es_search_pool_active{endpoint=es5-1} Max: 12
es_search_pool_active{endpoint=es6-1} Max: 2
es_search_pool_rejected
es_search_pool_rejected{endpoint=es5} Max: 0
es_search_pool_rejected{endpoint=es7-1} Max: 0
es_search_pool_rejected{endpoint=es1} Max: 0
es_search_pool_rejected{endpoint=es4-1} Max: 0
es_search_pool_rejected{endpoint=es2-1} Max: 0
es_search_pool_rejected{endpoint=es8-1} Max: 0
es_search_pool_rejected{endpoint=es8-2} Max: 0
es_search_pool_rejected{endpoint=es6-2} Max: 0
es_search_pool_rejected{endpoint=es3} Max: 0
es_search_pool_rejected{endpoint=es3-1} Max: 0
es_generic_pool_threads
es_generic_pool_threads{endpoint=es5} Max: 4
es_generic_pool_threads{endpoint=es7-1} Max: 95
es_generic_pool_threads{endpoint=es1} Max: 24
es_generic_pool_threads{endpoint=es4-1} Max: 101
es_generic_pool_threads{endpoint=es2-1} Max: 126
es_generic_pool_threads{endpoint=es8-1} Max: 60
es_generic_pool_threads{endpoint=es8-2} Max: 77
es_generic_pool_threads{endpoint=es6-2} Max: 70
es_generic_pool_threads{endpoint=es3} Max: 5
es_generic_pool_threads{endpoint=es3-1} Max: 142
es_generic_pool_threads{endpoint=es7-2} Max: 79
es_generic_pool_threads{endpoint=es1-1} Max: 139
es_generic_pool_active
es_generic_pool_active{endpoint=es5} Max: 0
es_generic_pool_active{endpoint=es7-1} Max: 29
es_generic_pool_active{endpoint=es1} Max: 4
es_generic_pool_active{endpoint=es4-1} Max: 12
es_generic_pool_active{endpoint=es2-1} Max: 16
es_generic_pool_active{endpoint=es8-1} Max: 28
es_generic_pool_active{endpoint=es8-2} Max: 29
es_generic_pool_active{endpoint=es6-2} Max: 31
es_generic_pool_active{endpoint=es3} Max: 0
es_generic_pool_active{endpoint=es3-1} Max: 20
es_generic_pool_active{endpoint=es7-2} Max: 25
es_generic_pool_active{endpoint=es1-1} Max: 20
es_generic_pool_active{endpoint=es4} Max: 0
es_generic_pool_active{endpoint=es2} Max: 1
es_generic_pool_active{endpoint=es5-1} Max: 20

感谢聆听



专业、垂直、纯粹的 Elastic 开源技术交流社区

https://elasticsearch.cn/