

Оптимизация хранения данных в OpenSearch



Антон Касимов



Поговорим о модели хранения данных и методах оптимизации хранения в OpenSearch

Что вас ждет.



Как хранятся
данные



Как
оптимизировать
хранение



**СПЕЦИАЛИЗИРОВАННЫЙ СИСТЕМНЫЙ ИНТЕГРАТОР
(СИСТЕМЫ МОНИТОРИНГА)**

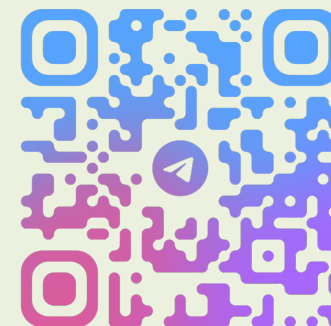
ВНЕДРЕНИЕ И НАСТРОЙКА OPENSEARCH

ТЕХНИЧЕСКАЯ ПОДДЕРЖКА OPENSEARCH

**ОБУЧЕНИЕ НА ТРЕНИНГАХ «БАЗА»
И «ПРОДВИНУТЫЙ»**

ПАРТНЕР Yandex  Cloud

Подписывайтесь на каналы



@ELASTICSTACK_RU



@MONITORIM_IT



Тренинги в 2026 году:

- **OpenSearch База** → 18 – 20 февраля
- **OpenSearch Продвинутый** → 18 – 20 марта

* Полное расписание: <https://gals.software/education/timetable>



OpenSearch База (3 дня)

Теория + практика (40/60)

От основ к глубокому погружению

Установка отказоустойчивого кластера

Загрузка и трансформация данных через Logstash, DataPrepper, Fluent Bit, Vector

Визуализации, ISM, снапшоты, Benchmark



Программа курса

OpenSearch Продвинутый (3 дня)

Теория + практика (40/60)

Глубокая работа с OpenSearch

Ролевая модель и аудит-лог, маппинг и observability

Кросс-кластерная репликация и поиск, продвинутый ISM, оповещения

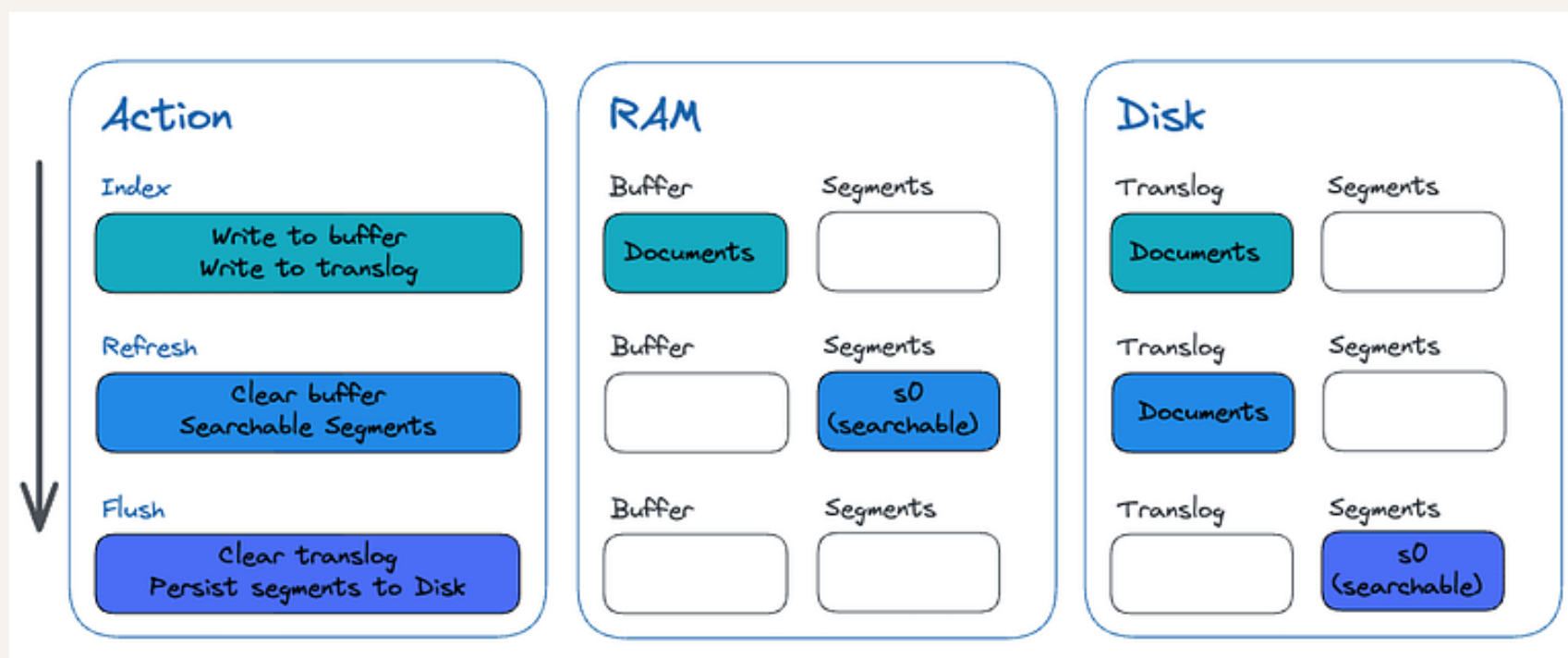
Работа с Vector, Logstash, DataPrepper, OpenTelemetry Collector, Ingest Pipeline и Kafka



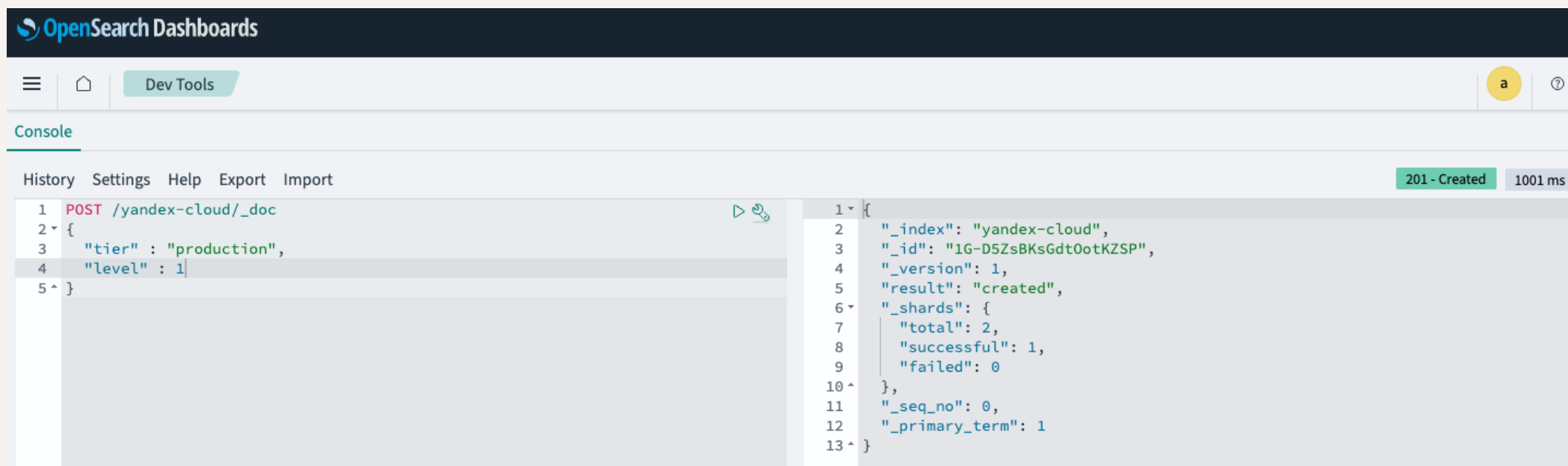
Программа курса

КАК ХРАНЯТСЯ ДАННЫЕ

Как хранятся данные в OpenSearch



Запишем документ в индекс



The screenshot shows the OpenSearch Dashboards interface with the Dev Tools console open. The console displays a successful POST request to the `/yandex-cloud/_doc` endpoint. The request body is a JSON document with `"tier": "production"` and `"level": 1`. The response shows the document was created successfully, with `"result": "created"`, `"_shards": {"total": 2, "successful": 1, "failed": 0}`, `"_seq_no": 0`, and `"_primary_term": 1`. The status bar indicates `201 - Created` and `1001 ms`.

OpenSearch Dashboards

Dev Tools

Console

History Settings Help Export Import

201 - Created 1001 ms

```
1 POST /yandex-cloud/_doc
2 {
3   "tier" : "production",
4   "level" : 1
5 }

1 {
2   "_index": "yandex-cloud",
3   "_id": "1G-D5ZsBKsGdt0otKZSP",
4   "_version": 1,
5   "result": "created",
6   "_shards": {
7     "total": 2,
8     "successful": 1,
9     "failed": 0
10  },
11   "_seq_no": 0,
12   "_primary_term": 1
13 }
```

Почему total = 2, successful = 1?

Посмотрим на шарды и сегменты

OpenSearch Dashboards

Dev Tools

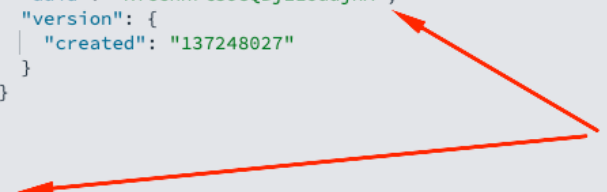
Console

History Settings Help Export Import

200 - OK 1264 ms

```
1 GET yandex-cloud/_settings
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
```

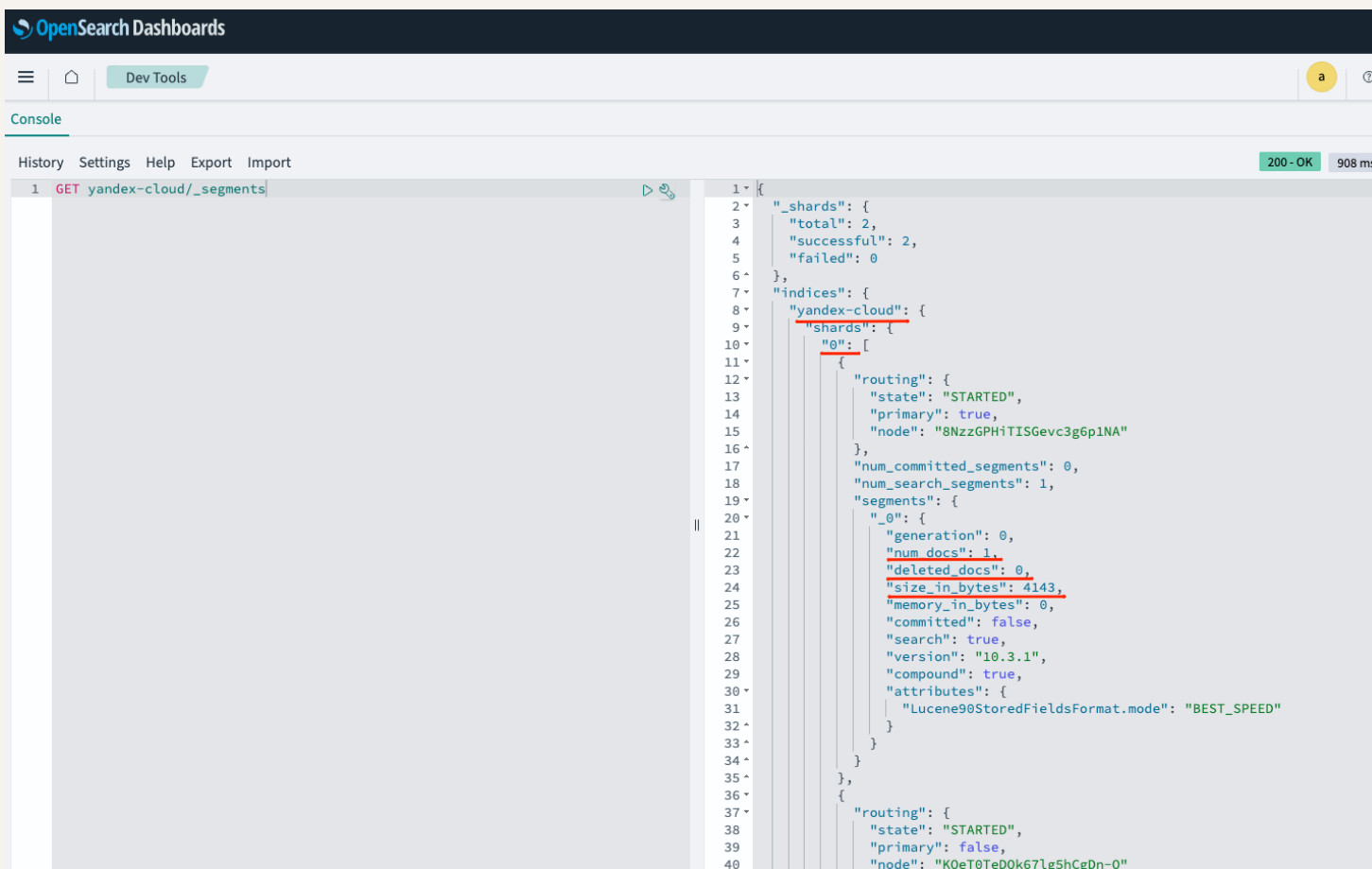
```
1 {
2   "yandex-cloud": {
3     "settings": {
4       "index": {
5         "replication": {
6           "type": "DOCUMENT"
7         },
8         "number_of_shards": "1",
9         "provided_name": "yandex-cloud",
10        "creation_date": "1769083422133",
11        "number_of_replicas": "1",
12        "uuid": "KTCChrtSJeqBjELUuujnA",
13        "version": {
14          "created": "137248027"
15        }
16      }
17    }
18  }
19 }
```



```
root@vm-opensearch01:~# ls -la /var/lib/opensearch/nodes/0/indices/KTCChrtSJeqBjELUuujnA/0/index/
total 24
drwxr-xr-x 2 opensearch opensearch 4096 Jan 22 15:03 .
drwxr-xr-x 5 opensearch opensearch 4096 Jan 22 15:03 ..
-rw-r--r-- 1 opensearch opensearch 517 Jan 22 15:03 _0.cfe
-rw-r--r-- 1 opensearch opensearch 3297 Jan 22 15:03 _0.cfs
-rw-r--r-- 1 opensearch opensearch 329 Jan 22 15:03 _0.si
-rw-r--r-- 1 opensearch opensearch 325 Jan 22 15:03 segments_4
-rw-r--r-- 1 opensearch opensearch 0 Jan 22 15:03 write.lock
```

0 segment

Посмотрим на сегменты



The screenshot shows the OpenSearch Dashboards interface. The top navigation bar includes the OpenSearch logo and the text "OpenSearch Dashboards". Below the navigation bar, there is a "Dev Tools" tab. The "Console" section is active, displaying a list of requests. The first request is a GET request to `yandex-cloud/_segments`. The response is a JSON object showing the status of the segments. The response is highlighted with a green status bar indicating "200 - OK" and a response time of "908 ms".

```
1 GET yandex-cloud/_segments
2 {
3   "_shards": {
4     "total": 2,
5     "successful": 2,
6     "failed": 0
7   },
8   "indices": {
9     "yandex-cloud": {
10      "shards": {
11        "0": [
12          {
13            "routing": {
14              "state": "STARTED",
15              "primary": true,
16              "node": "8NzzGPHiTiISGevc3g6p1NA"
17            },
18            "num_committed_segments": 0,
19            "num_search_segments": 1,
20            "segments": {
21              "_0": {
22                "generation": 0,
23                "num_docs": 1,
24                "deleted_docs": 0,
25                "size_in_bytes": 4143,
26                "memory_in_bytes": 0,
27                "committed": false,
28                "search": true,
29                "version": "10.3.1",
30                "compound": true,
31                "attributes": {
32                  "Lucene90StoredFieldsFormat.mode": "BEST_SPEED"
33                }
34              }
35            }
36          }
37        ],
38        "1": [
39          {
40            "routing": {
41              "state": "STARTED",
42              "primary": false,
43              "node": "KQeT0TeDQk67lg5hCgDn-Q"
```

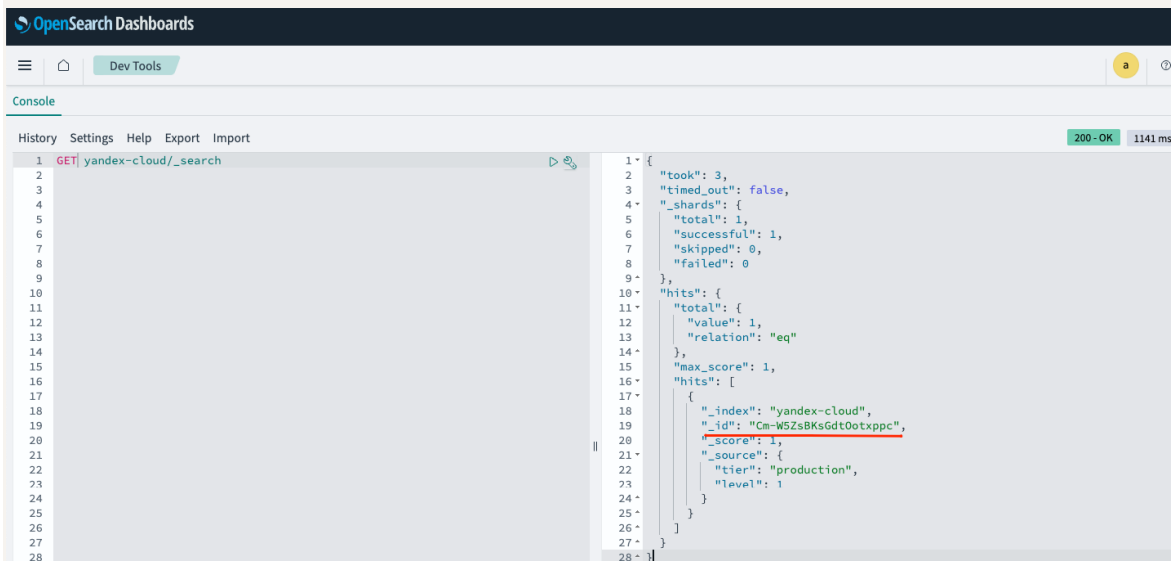
Обратите внимание на размер индекса

Проверим созданный маппинг полей

The screenshot shows the OpenSearch Dashboards interface with the Dev Tools console open. The console displays a successful GET request to `yandex-cloud/_mapping` with a status of 200 - OK and a response time of 685 ms. The response is a JSON object defining the mapping for the `yandex-cloud` index.

```
1 GET yandex-cloud/_mapping
2 {
3   "yandex-cloud": {
4     "mappings": {
5       "properties": {
6         "level": {
7           "type": "long"
8         },
9         "tier": {
10          "type": "text",
11          "fields": {
12            "keyword": {
13              "type": "keyword",
14              "ignore_above": 256
15            }
16          }
17        }
18      }
19    }
20  }
```

Удалим документ



OpenSearch Dashboards

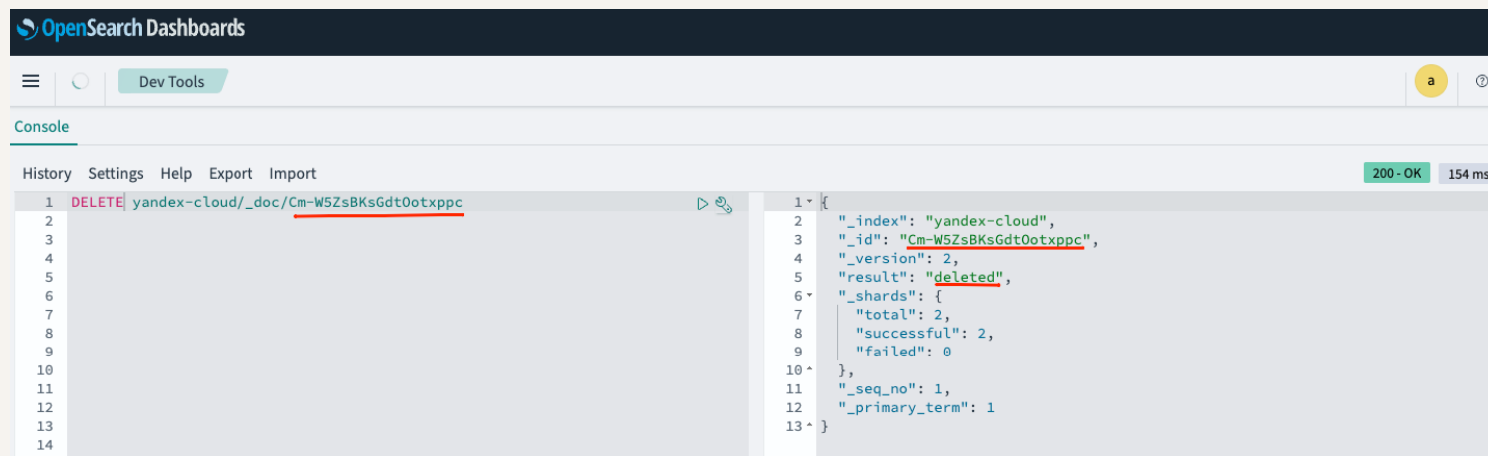
Console

History Settings Help Export Import

1 GET yandex-cloud/_search

200 - OK 1141 ms

```
1 {
2   "took": 3,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": {
12      "value": 1,
13      "relation": "eq"
14    },
15    "max_score": 1,
16    "hits": [
17      {
18        "_index": "yandex-cloud",
19        "_id": "Cm-W5ZsBKsGdt0otxppc",
20        "score": 1,
21        "_source": {
22          "tier": "production",
23          "level": 1
24        }
25      }
26    ]
27  }
28 }
```



OpenSearch Dashboards

Console

History Settings Help Export Import

1 DELETE yandex-cloud/_doc/Cm-W5ZsBKsGdt0otxppc

200 - OK 154 ms

```
1 {
2   "_index": "yandex-cloud",
3   "_id": "Cm-W5ZsBKsGdt0otxppc",
4   "_version": 2,
5   "result": "deleted",
6   "_shards": {
7     "total": 2,
8     "successful": 2,
9     "failed": 0
10  },
11  "_seq_no": 1,
12  "_primary_term": 1
13 }
```

Смотрим размер индекса

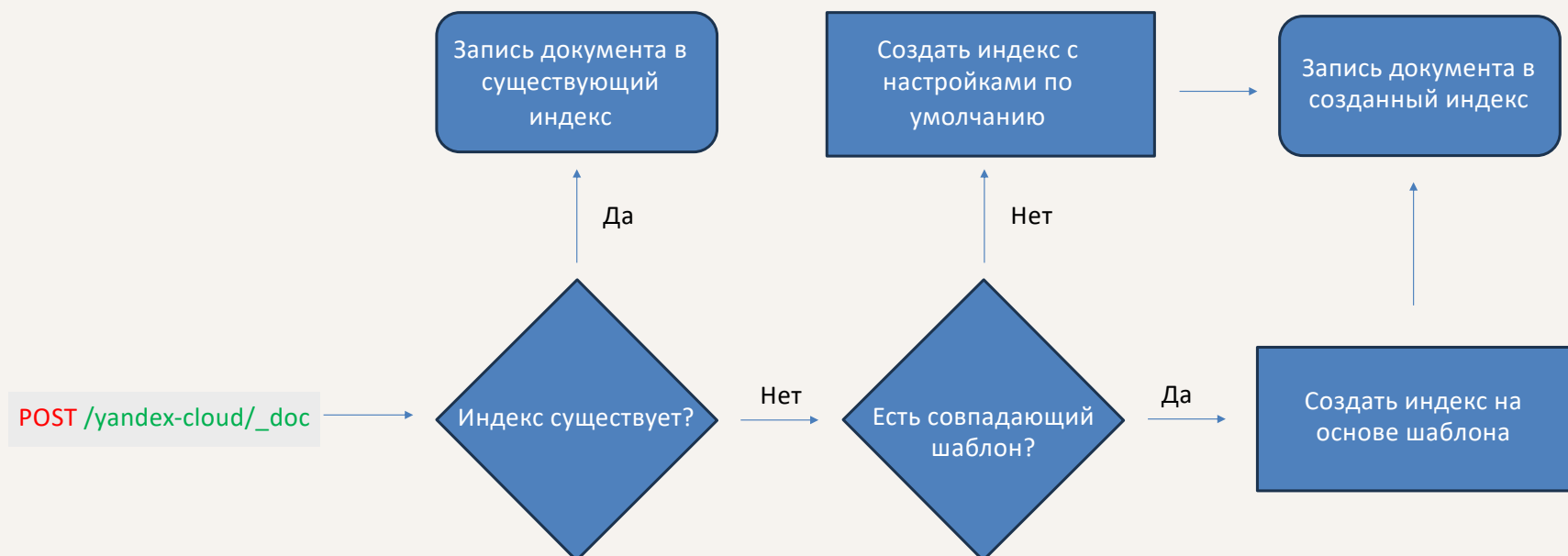
The screenshot shows the OpenSearch Dashboards interface. At the top, there's a dark header with the OpenSearch logo and 'Dashboards' text. Below it, a navigation bar includes a menu icon, a home icon, a 'Dev Tools' tab, and a user profile icon labeled 'a'. The main area is titled 'Console' and contains a 'History' tab with sub-tabs for 'Settings', 'Help', 'Export', and 'Import'. A status bar at the top right of the console shows '200 - OK' and '811 ms'. The console displays a GET request to `yandex-cloud/_stats` and its corresponding JSON response. The response is a JSON object with the following structure:

```
{
  "_shards": {
    "total": 2,
    "successful": 2,
    "failed": 0
  },
  "_all": {
    "primaries": {
      "docs": {
        "count": 1,
        "deleted": 0
      },
      "store": {
        "size_in_bytes": 4457,
        "reserved_in_bytes": 0
      },
      "indexing": {
        "index_total": 1,
        "index_time_in_millis": 0,
        "index_current": 0,
        "index_failed": 0,
        "delete_total": 1,
        "delete_time_in_millis": 6,
        "delete_current": 0,
        "noop_update_total": 0,
        "is_throttled": false,
        "throttle_time_in_millis": 0,

```

Обратите внимание на размер индекса

Как создается индекс



Как еще можно протестировать



<https://docs.opensearch.org/latest/benchmark>

Датасет eventlog

Логи Apache

20 млн документов

~15 Гб сырых данных



GitHub repository: [opensearch-project / opensearch-benchmark-workloads](https://github.com/opensearch-project/opensearch-benchmark-workloads) (Public)

generated from [amazon-archives/_template_Apache-2.0](#)

Code Issues 48 Pull requests 1 Actions Projects Security Insights

Files

- main
- Go to file
- .github
- big5
- clickbench
- common_operations
- eventdata
 - operations
 - test_procedures
 - README.md
 - files.txt
 - index.json
 - workload.json
- geonames
- geopoint
- geopointshape
- geoshape
- http_logs
- nested
- neural_search
- noaa
- noaa_semantic_search

opensearch-benchmark-workloads / eventdata /

OVI3D0 Insert index_name to common operations/cluster health operations in w... ae47393 · 10 months ago History

Name	Last commit message	Last commit date
..		
operations	Change "rally" to "benchmark" in Jinja template (#12)	5 years ago
test_procedures	insert index_name to common operations/cluster health operations in w...	10 months ago
README.md	added query cache to every workload and updated README.md files	3 years ago
files.txt	Added missing files.txt for eventdata and so tracks	8 years ago
index.json	added query cache to every workload and updated README.md files	3 years ago
workload.json	Update workload download endpoints (#122)	3 years ago

README.md

EventData workload

This workload is based on 20 million Apache access log entries generated based on statistics from sample elastic.co access logs using a generator.

The size of the data file is around 15GB, which gives an average JSON record size of 822 bytes. Mappings have been optimized and some of the fields added through `geoip` and `user-agent` enrichment has been removed to achieve a more compact format.

The purpose of this workload is to provide an efficient way to benchmark indexing of this data type as the generator built into the benchmark-eventdata-workload can be CPU intensive.

<https://github.com/opensearch-project/opensearch-benchmark-workloads/tree/main/eventdata>

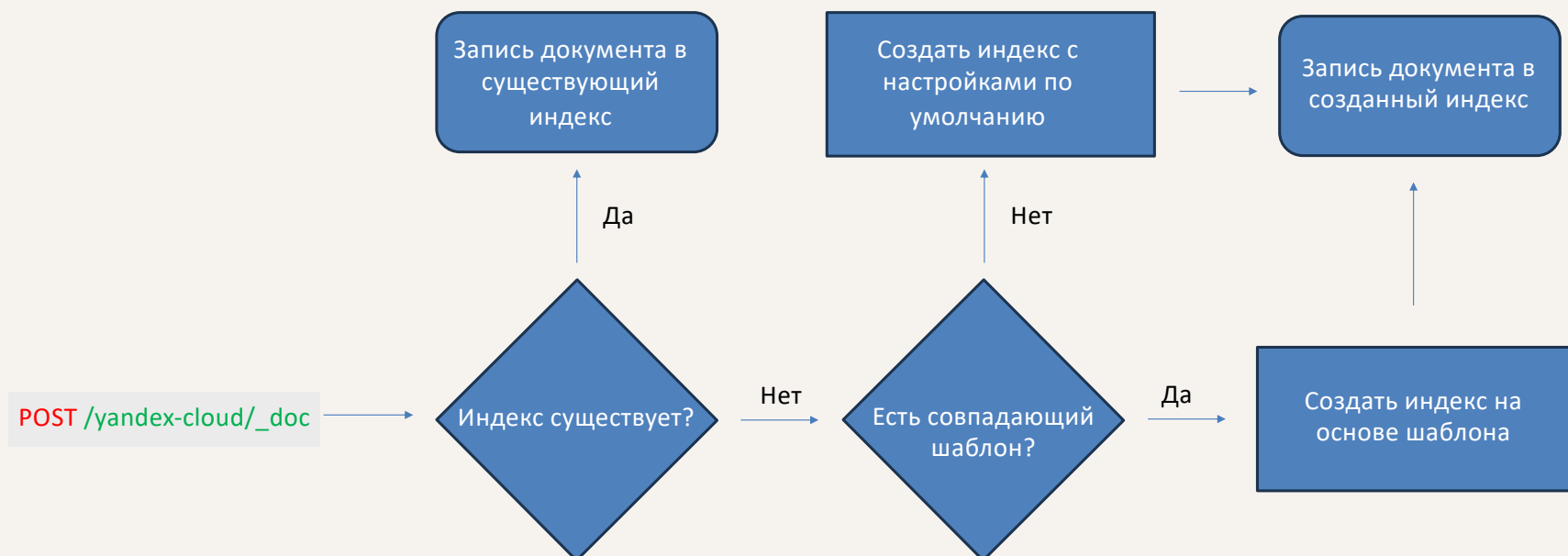
Датасет eventlog

Позже мы его загрузим в кластер и посмотрим...

КАК ОПТИМИЗИРОВАТЬ ХРАНЕНИЕ

Работать с маппингом

Как создается индекс



Динамический маппинг полей (dynamic)

JSON	OPENSEARCH
string	<ul style="list-style-type: none">• text + keyword• date• float или long
integer	long
float	float
boolean	boolean
object	object
array	зависит от первого не пустого значения

Как выглядит маппинг

PostgreSQL

```
CREATE TABLE pages (  
  rating    decimal(40),  
  content   varchar(40),  
  product_id integer,  
  author_first_name varchar(40),  
  author_last_name varchar(40),  
  author_email varchar(40),  
  CONSTRAINT product_id PRIMARY KEY(product_id)  
);
```

OpenSearch

```
PUT pages  
{  
  "mappings": {  
    "properties": {  
      "rating": { "type": "float" },  
      "content": { "type": "text" },  
      "product_id": { "type": "integer" },  
      "author": {  
        "properties": {  
          "first_name": { "type": "text" },  
          "last_name": { "type": "text" },  
          "email": { "type": "keyword" }  
        }  
      }  
    }  
  }  
}
```


Делаем explicit-маппинг

Типы полей

object

long

float

keyword

double

text

short

integer

date

* это далеко не все типы

Типы полей

keyword

text

Анализаторы (standard)

POST /_analyze

```
{  
  "text": "Съешь ещё. .. этих мягких французских булок, да  
выпей 2 литра сока!!!!",  
  "analyzer": "standard"  
}
```

=

POST /_analyze


```
{  
  "text": "Съешь ещё. .. этих мягких французских булок,  
да выпей 2 литра сока!!!!",  
  "char_filter": [],  
  "tokenizer": "standard",  
  "filter": ["lowercase"]  
}
```

Анализатор (standard)

Результат прохождения через анализатор

POST /_analyze

```
{  
  "text": "Съешь ещё. .. этих мягких французских булок, да  
выпей 2 литра сока!!!!",  
  "analyzer": "standard"  
}
```



```
{  
  "tokens": [  
    {  
      "token": "съешь",  
      "start_offset": 0,  
      "end_offset": 5,  
      "type": "<ALPHANUM>",  
      "position": 0  
    },  
    {  
      "token": "ещё",  
      "start_offset": 6,  
      "end_offset": 9,  
      "type": "<ALPHANUM>",  
      "position": 1  
    },  
    {  
      "token": "этих",  
      "start_offset": 15,  
      "end_offset": 19,  
      "type": "<ALPHANUM>",  
      "position": 2  
    },  
    {  
      "token": "мягких",  
      "start_offset": 20,  
      "end_offset": 26,  
      "type": "<ALPHANUM>",  
      "position": 3  
    },  
    {  
      "token": "французских",  
      "start_offset": 27,  
      "end_offset": 38,  
      "type": "<ALPHANUM>",  
      "position": 4  
    }  
  ]  
}
```

Инвертированный индекс

DOCUMENT #1

"text": "Съешь ещё. .. этих мягких французских булок, да выпей 2 литра сока!!!!!"

DOCUMENT #2

"text": "Съешь мягких булок"


TERM	DOCUMENT #1	DOCUMENT #2
съешь	x	x
ещё	x	
этих	x	
мягких	x	x
французских	x	
булок	x	x
да	x	
выпей	x	
2	x	
литра	x	
сока	x	

Анализатор (keyword)


POST /_analyze

```
{  
  "text": "Съешь ещё. .. этих мягких французских булок, да  
выпей 2 литра сока!!!!",  
  "analyzer": "keyword"  
}
```

```
{  
  "tokens": [  
    {  
      "token": "Съешь ещё. .. этих мягких французских булок, да  
выпей 2 литра сока!!!!",  
      "start_offset": 0,  
      "end_offset": 72,  
      "type": "word",  
      "position": 0  
    }  
  ]  
}
```



Создаем маппинг



The screenshot shows the OpenSearch Dashboards interface with the Dev Tools console open. The console displays a successful PUT request to create a mapping for the index 'yandex-cloud'. The request body is a JSON object defining mappings for 'tier' (keyword) and 'level' (integer). The response body shows the mapping was acknowledged successfully.

```
1 PUT yandex-cloud
2 {
3   "mappings": {
4     "properties": {
5       "tier": {
6         "type": "keyword"
7       },
8       "level": {
9         "type": "integer"
10      }
11    }
12  }
13 }
14
```

```
1 {
2   "acknowledged": true,
3   "shards_acknowledged": true,
4   "index": "yandex-cloud"
5 }
```

200 - OK 800 ms

Записываем документ

The screenshot shows the OpenSearch Dashboards interface with the Dev Tools console open. The console displays a successful POST request to the `yandex-cloud/_doc` endpoint. The request body is a JSON object with `"tier": "production"` and `"level": 1`. The response is a JSON object indicating the document was created successfully, including the index name, document ID, version number, and shard information.

OpenSearch Dashboards

Dev Tools

Console

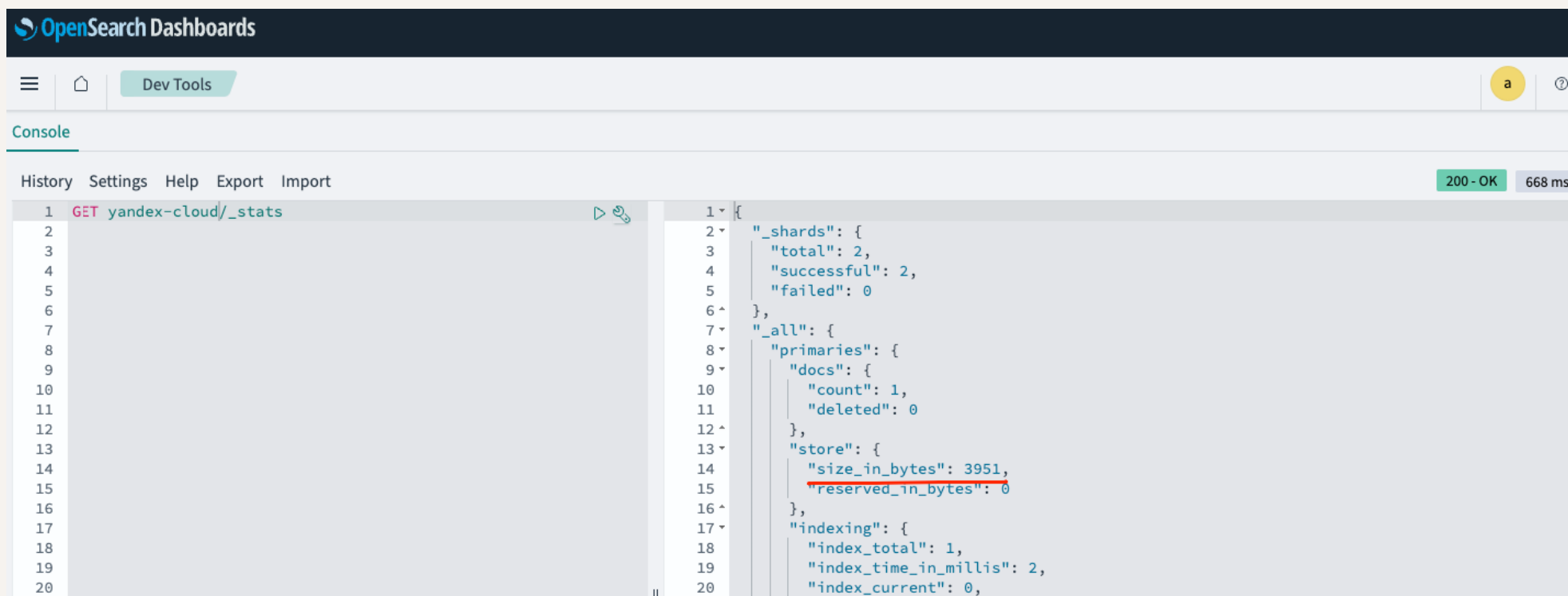
History Settings Help Export Import

201 - Created 164 ms

```
1 POST yandex-cloud/_doc
2 {
3   "tier" : "production",
4   "level" : 1
5 }
6
7
8
9
10
11
12
13
14
```

```
1 {
2   "_index": "yandex-cloud",
3   "_id": "gW_c5ZsBKsGdt0otg6ys",
4   "_version": 1,
5   "result": "created",
6   "_shards": {
7     "total": 2,
8     "successful": 2,
9     "failed": 0
10  },
11   "_seq_no": 0,
12   "_primary_term": 1
13 }
```

Получаем результат



The screenshot shows the OpenSearch Dashboards interface with the DevTools console open. The console displays a successful GET request to `yandex-cloud/_stats` with a status of 200 - OK and a response time of 668 ms. The response is a JSON object containing statistics for shards and indexing.

```
1 GET yandex-cloud/_stats
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
```

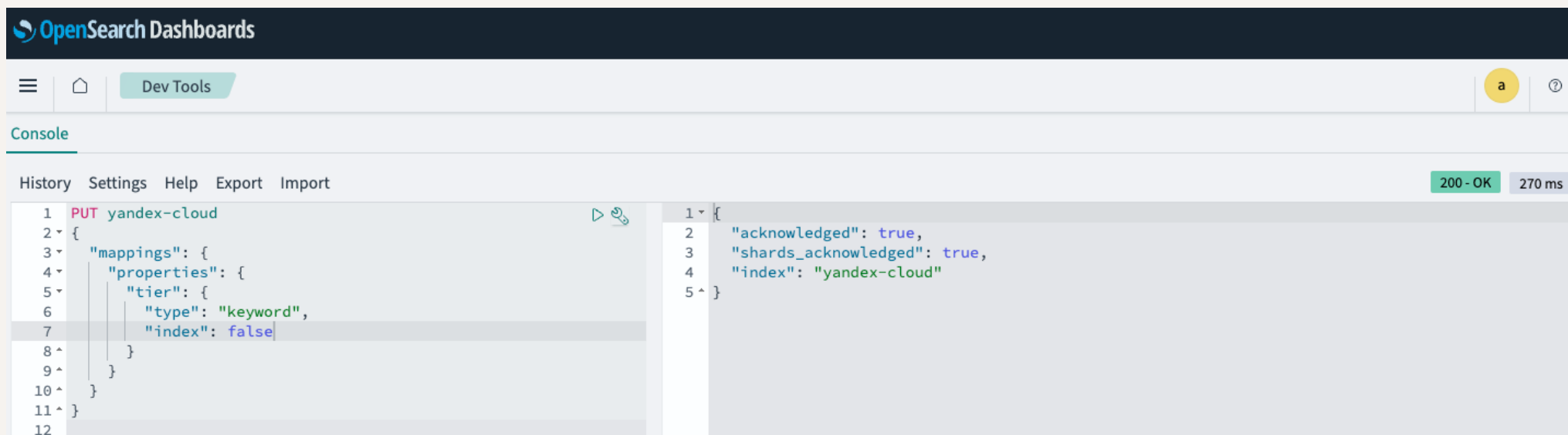
```
1 {
2   "_shards": {
3     "total": 2,
4     "successful": 2,
5     "failed": 0
6   },
7   "_all": {
8     "primaries": {
9       "docs": {
10        "count": 1,
11        "deleted": 0
12      },
13      "store": {
14        "size_in_bytes": 3951,
15        "reserved_in_bytes": 0
16      },
17      "indexing": {
18        "index_total": 1,
19        "index_time_in_millis": 2,
20        "index_current": 0,
```

* было 4143

Параметры маппинга

analyzer, coerce, copy_to, doc_values, dynamic, eager_global_ordinals, enabled, format, index.mapping, ignore_above, ignore_malformed, index, index_options, index_phrases, index_prefixes, meta, fields, normalizer, norms, null_value, position_increment_gap, properties, search_analyzer, similarity, store, subobjects, term_vector

Добавляем параметр index



The screenshot shows the OpenSearch Dashboards interface with the Dev Tools console open. The console displays a PUT request to the `yandex-cloud` index and its successful response.

```
1 PUT yandex-cloud
2 {
3   "mappings": {
4     "properties": {
5       "tier": {
6         "type": "keyword",
7         "index": false
8       }
9     }
10  }
11 }
```

```
1 {
2   "acknowledged": true,
3   "shards_acknowledged": true,
4   "index": "yandex-cloud"
5 }
```

The status bar at the top right of the console indicates a `200 - OK` response with a duration of `270 ms`.

*** отключение записи в инвертированный индекс**

Смотрим размер

OpenSearch Dashboards

Dev Tools

Console

History Settings Help Export Import

200 - OK 691 ms

```
1 GET yandex-cloud/_stats
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
```

```
1 {
2   "_shards": {
3     "total": 2,
4     "successful": 2,
5     "failed": 0
6   },
7   "_all": {
8     "primaries": {
9       "docs": {
10        "count": 1,
11        "deleted": 0
12      },
13      "store": {
14        "size_in_bytes": 3733,
15        "reserved_in_bytes": 0
16      },
17      "indexing": {
18        "index_total": 1,
19        "index_time_in_millis": 1,
20        "index_current": 0,
21        "index_failed": 0,
22        "delete_total": 0,
23        "delete_time_in_millis": 0,
24        "delete_current": 0,
25        "noop_update_total": 0,
26        "is_throttled": false,
```

Отключение скоринга для поля (norms)

```
PUT yandex-cloud
{
  "properties": {
    "title": {
      "type": "text",
      "norms": false
    }
  }
}
```

Принудительная типизация (coerce)

```
PUT yandex-cloud
{
  "settings": {
    "index.mapping.coerce": false
  },
  "mappings": {
    "properties": {
      "number_one": {
        "type": "integer",
        "coerce": true
      },
      "number_two": {
        "type": "integer"
      }
    }
  }
}
```

```
PUT yandex-cloud/_doc/1
{ "number_one": "10" }
```

OK

```
PUT yandex-cloud/_doc/1
{ "number_two": "10" }
```

REJECT

Тип поля `match_only_text`

Экономия на хранении до 25% по сравнению с типом `text`

Отключен скоринг, учет позиций и частоты

```
PUT yandex-cloud
```

```
{  
  "mappings": {  
    "properties": {  
      "tier": {  
        "type": "match_only_text"  
      }  
    }  
  }  
}
```


Отключение динамического маппинга

Отключение динамического маппинга
(дополнительные поля документов не будут проиндексированы, но запишутся в `_source`)

```
PUT yandex-cloud
{
  "mappings": {
    "dynamic": false,
    "properties": {
      "tier": {
        "type": "text"
      }
    }
  }
}
```

Ограничение создания документов
(документ, содержащий дополнительные поля, будет отброшен)

```
PUT yandex-cloud
{
  "mappings": {
    "dynamic": strict,
    "properties": {
      "tier": {
        "type": "text"
      }
    }
  }
}
```

Полу-динамический маппинг

Подсказывает OpenSearch правильный тип данных

```
PUT /dynamic_template_test
{
  "mappings": {
    "dynamic_templates": [
      {
        "integers": {
          "match_mapping_type": "long",
          "mapping": {
            "type": "integer"
          }
        }
      ]
    }
  }
}
```

Полу-динамический маппинг

Типизация по названию полей (match и unmatched)

```
PUT yandex-cloud
{
  "mappings": {
    "dynamic_templates": [
      {
        "strings_only_text": {
          "match_mapping_type": "string",
          "match": "text_*",
          "unmatch": "*_keyword",
          "mapping": {
            "type": "text"
          }
        }
      },
      {
        "strings_only_keyword": {
          "match_mapping_type": "string",
          "match": "*_keyword",
          "mapping": {
            "type": "keyword"
          }
        }
      }
    ]
  }
}
```

Изменить кодек для сжатия

Что такое кодеки

Кодеки индекса определяют способ сжатия и хранения на диске полей, хранящихся в индексе. Кодек индекса управляется статической настройкой `index.codec`, определяющей алгоритм сжатия. Эта настройка влияет на размер сегмента индекса и производительность операций индексирования.

Какие бывают кодеки

default — этот кодек использует алгоритм LZ4 с предустановленным словарем, который отдает приоритет производительности, а не степени сжатия. Он обеспечивает более быструю индексацию и поиск по сравнению с другими кодеками **best_compression**, но может привести к увеличению размера индекса/шарда. Если в настройках индекса не указан кодек, то в качестве алгоритма сжатия по умолчанию используется LZ4.

best_compression — этот кодек использует zlib в качестве базового алгоритма сжатия. Он обеспечивает высокие коэффициенты сжатия, что приводит к уменьшению размеров индексов. Однако это может повлечь за собой дополнительную нагрузку на CPU во время операций индексирования и, следовательно, привести к высоким задержкам индексирования и поиска.

zstd (OpenSearch 2.9 и более поздние) — этот кодек обеспечивает значительное сжатие, сравнимое с кодеком **best_compression**, при разумном использовании CPU и улучшенной производительности индексирования и поиска по сравнению с кодеком **default**.

zstd_no_dict (OpenSearch 2.9 и более поздние) — этот кодек похож на **zstd**, но не включает функцию сжатия словаря. Он обеспечивает более быструю индексацию и операции поиска по сравнению с **zstd**, но за счет немного большего размера индекса.

Какие бывают кодеки

default — этот кодек использует алгоритм LZ4 с предустановленным словарем, который отдает приоритет производительности, а не степени сжатия. Он обеспечивает более быструю индексацию и поиск по сравнению с другими кодеками **best_compression**, но может привести к увеличению размера индекса/шарда. Если в настройках индекса не указан кодек, то в качестве алгоритма сжатия по умолчанию используется LZ4.

best_compression — этот кодек использует zlib в качестве базового алгоритма сжатия. Он обеспечивает высокие коэффициенты сжатия, что приводит к уменьшению размеров индексов. Однако это может повлечь за собой дополнительную нагрузку на CPU во время операций индексирования и, следовательно, привести к высоким задержкам индексирования и поиска.

zstd (OpenSearch 2.9 и более поздние) — этот кодек обеспечивает значительное сжатие, сравнимое с кодеком **best_compression**, при разумном использовании CPU и улучшенной производительности индексирования и поиска по сравнению с кодеком **default**.

zstd_no_dict (OpenSearch 2.9 и более поздние) — этот кодек похож на **zstd**, но не включает функцию сжатия словаря. Он обеспечивает более быструю индексацию и операции поиска по сравнению с **zstd**, но за счет немного большего размера индекса.

Дополнительное сжатие

`zstd` и `zstd_no_dict` дополнительно позволяют указать степень сжатия [1..6]. По умолчанию 3.

Более высокий уровень сжатия приводит к более высокому коэффициенту сжатия (меньшему объему памяти) с ухудшением скорости (более низкие скорости сжатия и распаковки приводят к большей задержке индексирования и поиска).

```
PUT yandex-cloud
{
  "settings": {
    "index.codec": "zstd",
    "index.codec.compression_level": 6
  }
}
```


Датасет eventdata

Логи Apache, 20 млн записей, 15 Гб в сыром виде

```
root@vm-opensearch01:~# ls -la /root/.osb/benchmarks/data/eventdata/eventdata.json
-rw-r--r-- 1 root root 16437108429 Jan 27 14:38 /root/.osb/benchmarks/data/eventdata/eventdata.json
```

Example Document

```
{
  "agent": "\\Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/48.0.2!
  "useragent": {
    "os": "Mac OS X 10.10.2",
    "os_name": "Mac OS X",
    "name": "Chrome"
  },
  "geoip": {
    "country_name": "India",
    "location": [80.2833, 13.083300000000008]
  },
  "clientip": "122.178.238.140",
  "referrer": "\\-\\",
  "request": "/apple-touch-icon-144x144.png",
  "bytes": 0,
  "verb": "GET",
  "response": 304,
  "httpversion": "1.1",
  "@timestamp": "2017-07-03T07:51:49.995Z",
  "message": "122.178.238.140 - - [2017-07-03T07:51:49.995Z] \\\"GET /apple-touch-icon-144x144.png HTTP/1.1\\\" 304 0 "
```

Загрузим eventdata с разной степенью сжатия

default vs zstd 6

Indexes (2)

Refresh

Actions

Create Index

Show data stream indexes

Index ↓	Health	Managed by policy	Status	Total size	Size of primaries	Total documents	Deleted documents	Primaries	Replicas
 eventdata-compressed-6	Green	No	Open	3gb	3gb	20000000	0	5	0
 eventdata	Green	No	Open	4.9gb	4.9gb	20000000	0	5	0

Rows per page: 20

<

1

>

СМОТРИМ НА МАППИНГ

OpenSearch Dashboards

Dev Tools

Console

History Settings Help Export Import

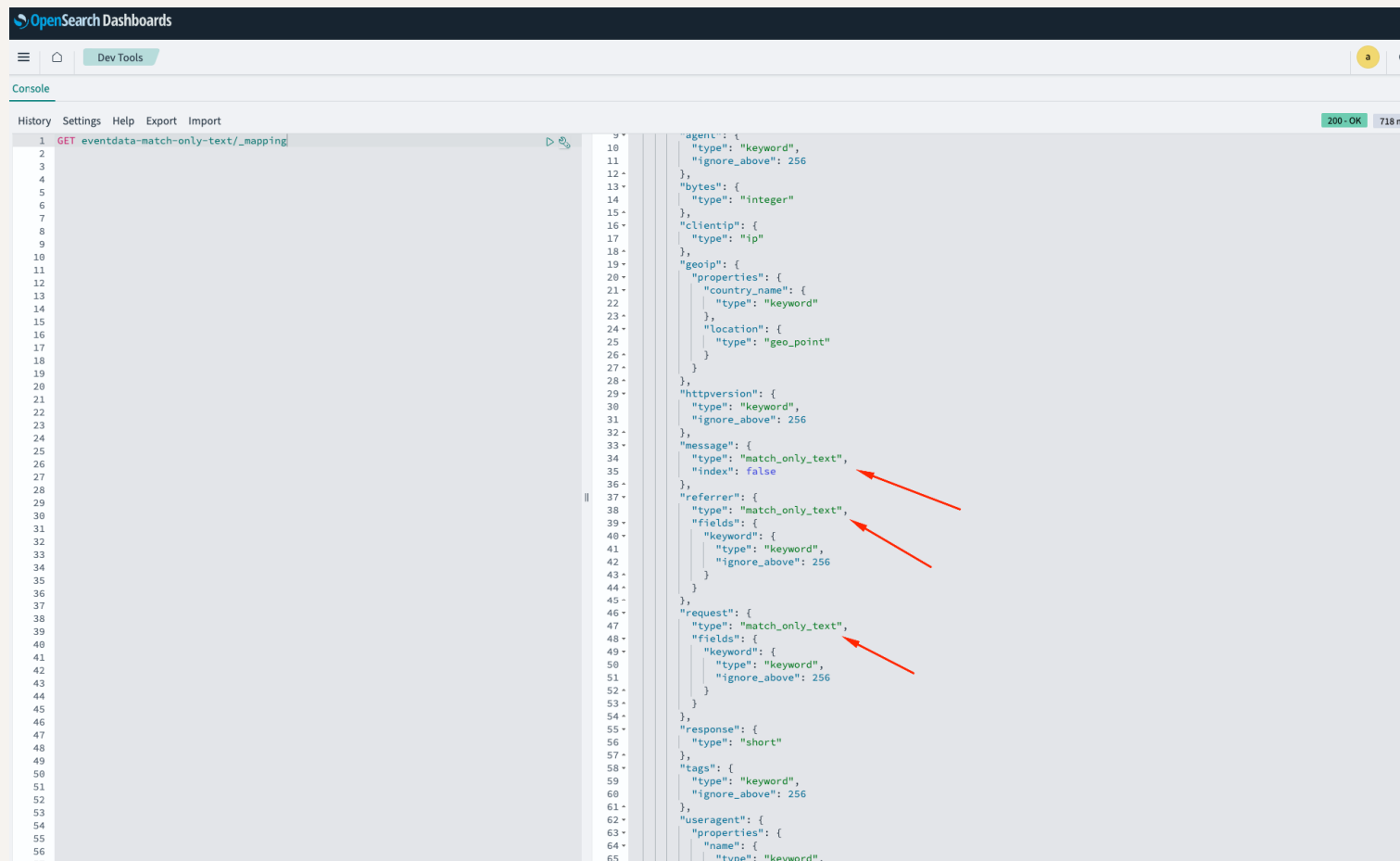
200 - OK 867 ms

```
1 GET eventdata-compressed-6/_mapping
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
```

```
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
```

```
{
  "type": "date"
},
"agent": {
  "type": "keyword",
  "ignore_above": 256
},
"bytes": {
  "type": "integer"
},
"clientip": {
  "type": "ip"
},
"geoip": {
  "properties": {
    "country_name": {
      "type": "keyword"
    },
    "location": {
      "type": "geo_point"
    }
  }
},
"httpversion": {
  "type": "keyword",
  "ignore_above": 256
},
"message": {
  "type": "text",
  "index": false
},
"referrer": {
  "type": "text",
  "fields": {
    "keyword": {
      "type": "keyword",
      "ignore_above": 256
    }
  },
  "norms": false
},
"request": {
  "type": "text",
  "fields": {
    "keyword": {
      "type": "keyword",
      "ignore_above": 256
    }
  },
  "norms": false
},
"response": {
  "type": "short"
},
"tags": {
  "type": "keyword",
  "ignore_above": 256
}
```

Вспоминаем про match_only_text



The screenshot shows the OpenSearch Dashboards interface with the Dev Tools console open. The console displays a GET request to `eventdata-match-only-text/_mapping` and its corresponding JSON response. The response is a mapping definition for the `eventdata-match-only-text` index. The `message`, `referrer`, and `request` fields are all of type `match_only_text`. Red arrows point to these fields in the JSON response.

```
1 GET eventdata-match-only-text/_mapping
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
```

```
{
  "agent": {
    "type": "keyword",
    "ignore_above": 256
  },
  "bytes": {
    "type": "integer"
  },
  "clientip": {
    "type": "ip"
  },
  "geoip": {
    "properties": {
      "country_name": {
        "type": "keyword"
      },
      "location": {
        "type": "geo_point"
      }
    }
  },
  "httpversion": {
    "type": "keyword",
    "ignore_above": 256
  },
  "message": {
    "type": "match_only_text",
    "index": false
  },
  "referrer": {
    "type": "match_only_text",
    "fields": {
      "keyword": {
        "type": "keyword",
        "ignore_above": 256
      }
    }
  },
  "request": {
    "type": "match_only_text",
    "fields": {
      "keyword": {
        "type": "keyword",
        "ignore_above": 256
      }
    }
  },
  "response": {
    "type": "short"
  },
  "tags": {
    "type": "keyword",
    "ignore_above": 256
  },
  "useragent": {
    "properties": {
      "name": {
        "type": "keyword"
      }
    }
  }
}
```

200 - OK 718 ms

Загрузим eventdata снова

text vs match_only_text

Indexes (3)




Refresh

Actions

Create Index

event

Show data stream indexes

<div><input type="checkbox"/> Index</div>	Health	Managed by policy	Status	Total size	Size of primaries	Total documents	Deleted documents	Primaries	Replicas
<div><input type="checkbox"/>  eventdata-match-only-text</div>	<div><div></div>Green</div>	No	Open	2.9gb	2.9gb	20000000	0	5	0
<div><input type="checkbox"/>  eventdata-compressed-6</div>	<div><div></div>Green</div>	No	Open	3gb	3gb	20000000	0	5	0
<div><input type="checkbox"/>  eventdata</div>	<div><div></div>Green</div>	No	Open	4.9gb	4.9gb	20000000	0	5	0

Rows per page: 20

<

1

>

Работать с cluster_state

Определение cluster_state

Внутренняя структура данных, которая отслеживает информацию, необходимую каждому узлу. Распространяется при помощи мастер-ноды. Включает в себя:

- Идентификаторы и атрибуты других узлов в кластере.
- Настройки для всего кластера.
- Метаданные индекса, включая сопоставление и настройки для каждого индекса.
- Местоположение и состояние каждой копии сегмента в кластере.
- Выбранный главный узел гарантирует, что каждый узел в кластере имеет копию одного и того же состояния кластера. API состояния кластера позволяет получить представление этого внутреннего состояния для целей отладки или диагностики.

Определение текущего размера _state

The screenshot shows the Bruno API client interface. On the left, a sidebar lists various API collections, with 'OpenSearch API' expanded and 'cluster state' selected. The main panel displays a GET request to the endpoint `{{opensearchURL}}/_cluster/state`. The response is a JSON object, and the size of the response body is highlighted as 168.26KB.

Request Details:

- Method: GET
- Endpoint: `{{opensearchURL}}/_cluster/state`
- Params: None
- Body: None
- Headers: None
- Auth: None

Response Details:

- Status: 200 OK
- Time: 337ms
- Size: 168.26KB

Response Body (JSON):

```
{
  "cluster_name": "learning-cluster",
  "cluster_uuid": "RDuKCYBfTDSObBDEXJbSw",
  "version": 1796,
  "state_uuid": "nXSqF7xHRqKUpEJqOvvsow",
  "master_node": "rYJwyb6eQyCJUz3A2BHhWw",
  "cluster_manager_node": "rYJwyb6eQyCJUz3A2BHhWw",
  "blocks": {},
  "nodes": {
    "8NzzGPHiTISGevc3g6p1NA": {
      "name": "vm-opensearch02",
      "ephemeral_id": "qRC41grpQxWJkyhKgztDgw",
      "transport_address": "192.168.0.3:9300",
      "attributes": {
        "shard_indexing_pressure_enabled": "true"
      }
    },
    "KQeT0TeDQk67lg5hCgDn-Q": {
      "name": "vm-opensearch01",
      "ephemeral_id": "w9lpRAOhTy2S83TKS4dBMw",
      "transport_address": "192.168.0.2:9300",
      "attributes": {
        "shard_indexing_pressure_enabled": "true"
      }
    }
  }
}
```


Если  OpenSearch , то 

Делаем как для себя.

gals.software

