elastic stack

Search and Beyond

Philipp Krenn @xeraa

elastic

elastic

Developer 🥑

This is not a Training

https://training.elastic.co

Agenda

09:00 - 10:40: Intro & Architecture & Search
10:40 - 11:00: Coffee break
11:00 - 12:20: More Search
12:20 - 13:05: Lunch
13:05 - 15:00: Monitoring
15:00 - 15:20: Coffee break
15:20 - 17:00: More Monitoring & Q&A

Elastic Stack Architecture

elasticsearch.

```
$ curl http://localhost:9200
{
  "name" : "elasticsearch-hot",
  "cluster_name" : "metrics-cluster",
  "cluster_uuid" : "O6nHPLLgTrmZEpYli6JW5w",
  "version" : {
    "number" : "6.5.0",
    "build_flavor" : "default",
    "build_type" : "tar",
    "build_hash" : "c53b7d3",
    "build_date" : "2018-11-08T21:28:50.577384Z",
    "build_snapshot" : false,
    "lucene_version" : "7.5.0",
    "minimum_wire_compatibility_version" : "5.6.0",
    "minimum_index_compatibility_version" : "5.0.0"
  },
  "tagline" : "You Know, for Search"
}
```

elastic

346 systems in ranking, October 2018

Rank			DBMS	Database Model	Score		
Oct 2018	Sep 2018	Oct 2017			Oct 2018	Sep 2018	Oct 2017
1.	1.	1.	Oracle	Relational DBMS	1319.27	+10.15	-29.54
2.	2.	2.	MySQL	Relational DBMS	1178.12	-2.36	-120.71
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1058.33	+7.05	-151.99
4.	4.	4.	PostgreSQL	Relational DBMS	419.39	+12.97	+46.12
5.	5.	5.	MongoDB	Document store	381.19	+4.39	+33.79
6.	6.	6.	DB2	Relational DBMS	179.69	-1.38	-14.90
7.	↑ 8.	↑ 9.	Redis	Key-value store	145.29	+4.35	+23.24
8.	↓ 7.	↑ 10.	Elasticsearch	Search engine	142.33	-0.28	+22.09
9.	9.	↓ 7.	Microsoft Access	Relational DBMS	136.80	+3.41	+7.35
10.	10.	↓ 8.	Cassandra	Wide column store	123.39	+3.83	-1.40
11.	11.	11.	SQLite	Relational DBMS	116.74	+1.29	+4.76
12.	12.	12.	Teradata	Relational DBMS	78.63	+1.25	-1.45
13.	13.	↑ 16.	Splunk	Search engine	76.90	+2.87	+12.54
14.	14.	↑ 18.	MariaDB	Relational DBMS	73.13	+2.49	+16.73
15.	15.	↓ 13.	Solr	Search engine	61.31	+1.11	-9.82

https://db-engines.com/en/ranking

346 systems in ranking, October 2018

Rank			DBMS	Database Model	Score		
Oct 2018	Sep 2018	Oct 2017			Oct 2018	Sep 2018	Oct 2017
1.	1.	1.	Oracle	Relational DBMS	1319.27	+10.15	-29.54
2.	2.	2.	MySQL	Relational DBMS	1178.12	-2.36	-120.71
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1058.33	+7.05	-151.99
4.	4.	4.	PostgreSQL	Relational DBMS	419.39	+12.97	+46.12
5.	5.	5.	MongoDB	Document store	363.19	+4.39	+33.79
6.	6.	6.	DB2	Relational DBMS	179.69	-1.38	-14.90
7.	↑ 8.	↑ 9.	Redis	Key-value store	145.29	+4.35	+23.24
8.	↓ 7.	↑ 10.	Elasticsearch	Search engine	142.33	-0.28	+22.09
9.	9.	↓ 7.	Microsoft Access	Relational DBMS	136.80	+3.41	+7.35
10.	10.	↓ 8.	Cassandra	Wide column store	123.39	+3.83	-1.40
11.	11.	11.	SQLite	Relational DBMS	116.74	+1.29	+4.76
12.	12.	12.	Teradata	Relational DBMS	78.63	+1.25	-1.45
13.	13.	↑ 16.	Splunk	Search engine	76.90	+2.87	+12.54
14.	14.	↑ 18.	MariaDB	Relational DBMS	73.13	+2.49	+16.73
15.	15.	↓ 13.	Solr	Search engine	61.31	+1.11	-9.82

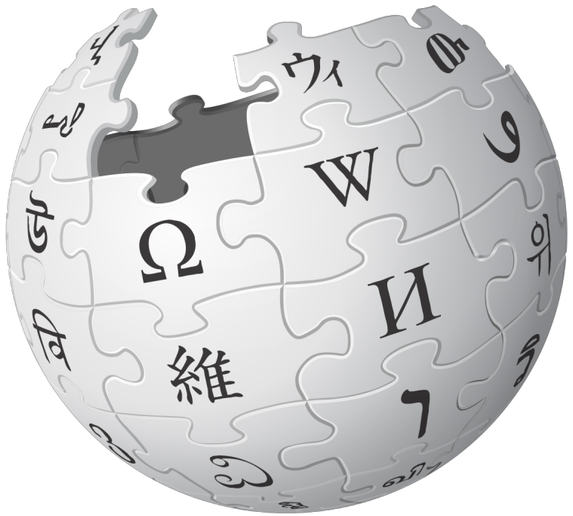
9. Redis

0. Elasticsearch

7. Microsoft Access

8. Cassandra

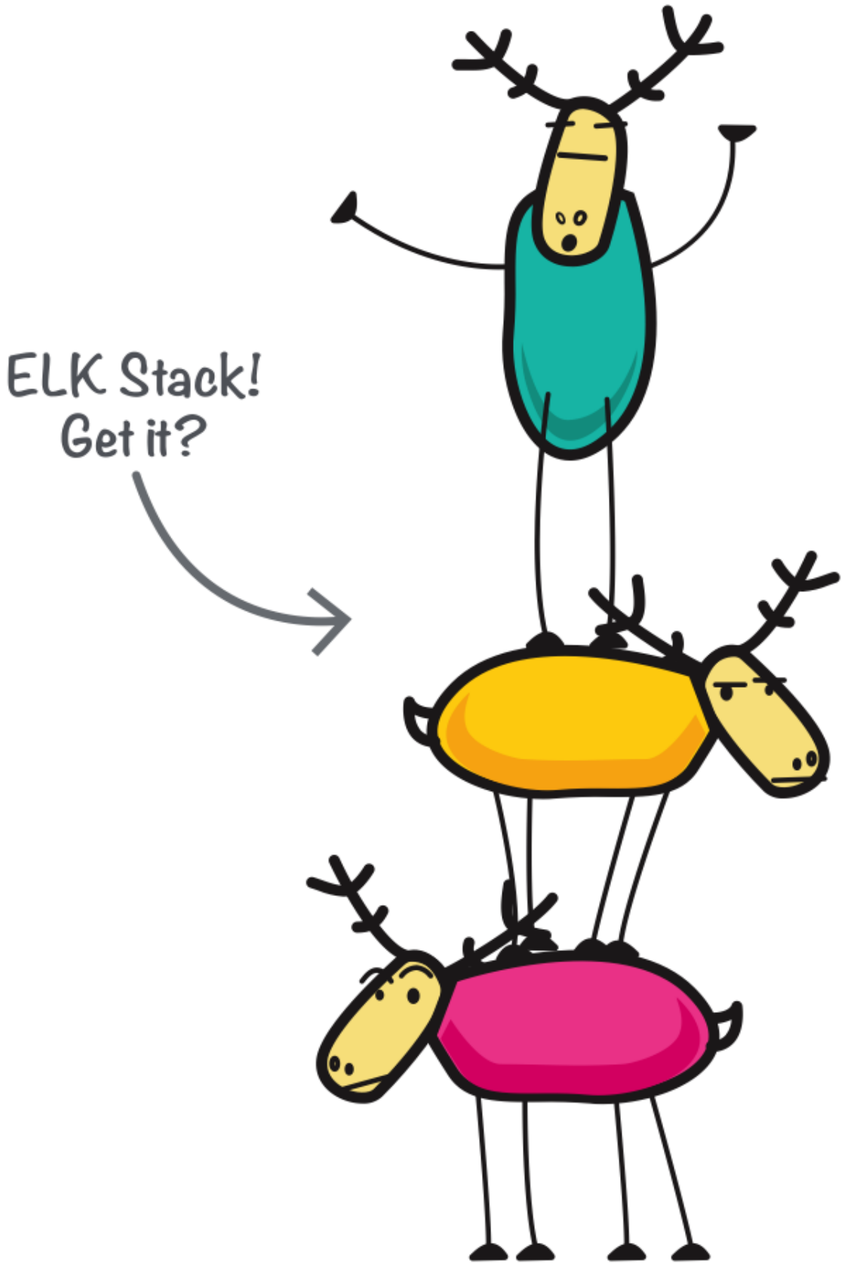
1. SQLite



kibana

logstash

elastic



E Elasticsearch

L Logstash

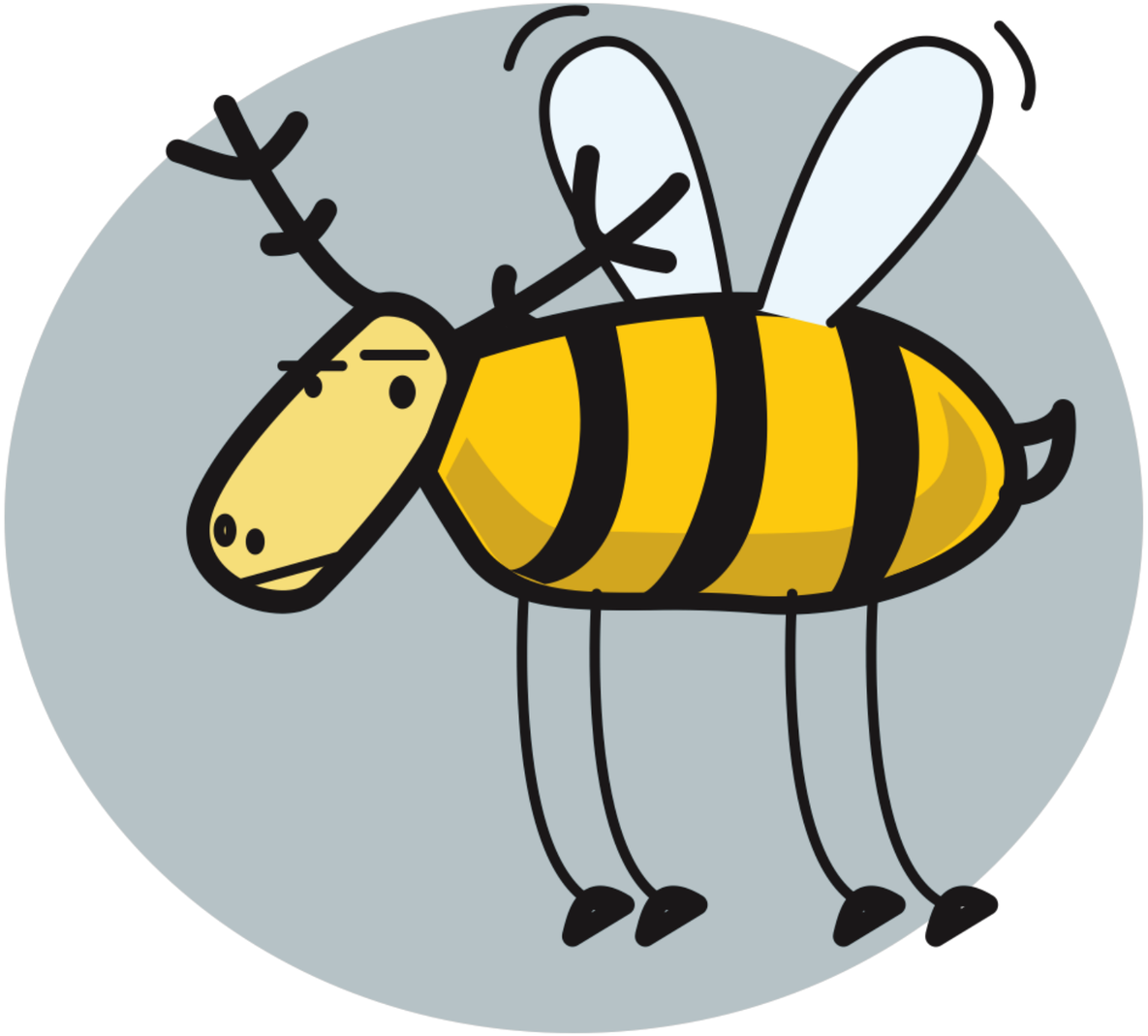
K Kibana

lyft

slack

fitbit

Apparently, I'm an ELKB personality.



elastic

elastic stack

BLIZZARD
ENTERTAINMENT

CAR
2GO

ebay

Only accept features that scale.

— **https://github.com/elastic/engineering/blob/master/development_constitution.md**

Horizontal Scaling

Shards
Replication
Writes & Reads

elastic

Exhibit A: A JSON Document

```
{
  "name": "Elasticsearch",
  "author": "Shay Banon",
  "stable_version": "6.5.0",
  "preview_version": "7.0.0-alpha1"
}
```

Exhibit B: A cURL Command

```
$ curl -XPOST -i localhost:9200/databases/nosql -d '
{
  "name": "Elasticsearch",
  "author": "Shay Banon",
  "stable_version": "6.5.0",
  "preview_version": "7.0.0-alpha1"
}'
```

Exhibit B: A cURL Command

```
HTTP/1.1 201 Created
Location: /databases/nosql/AVfD8XQaeuK3k1LGtT8-
content-type: application/json; charset=UTF-8
content-length: 162

{
  "_index":"databases",
  "_type":"nosql",
  "_id":"AVfD8XQaeuK3k1LGtT8-",
  "_version":1,
  "result":"created",
  "_shards": {
    "total":2,
    "successful":1,
    "failed":0
  },
  "created":true
}
```

elastic

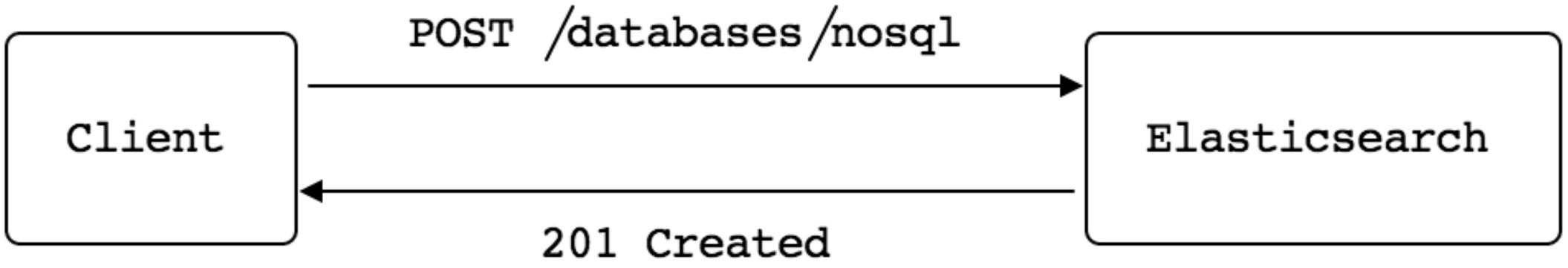
Exhibit C: A Console Command

```
POST /databases/nosql
{
  "name": "Elasticsearch",
  "author": "Shay Banon",
  "stable_version": "6.4.2",
  "preview_version": "7.0.0-alpha2"
}
```

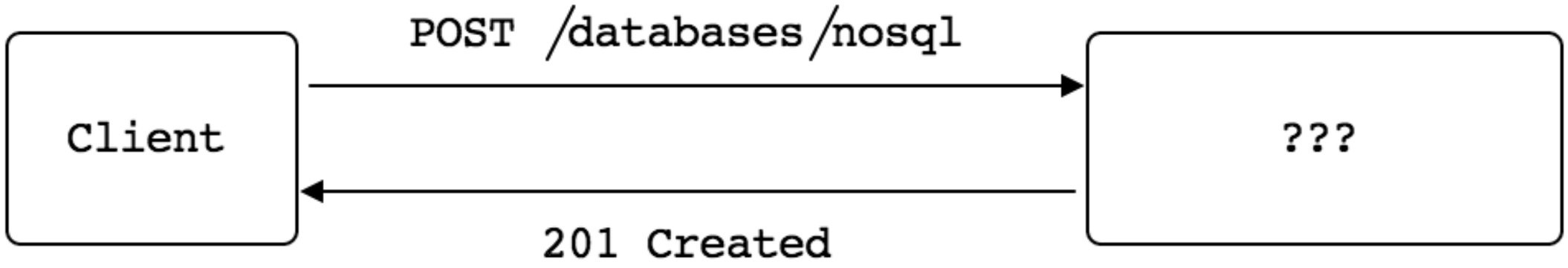
Exhibit C: A Console Command


```
{
  "_index": "databases",
  "_type": "nosql",
  "_id": "AVfD6ukyeuK3k1LGtSwT",
  "_version": 1,
  "result": "created",
  "_shards": {
    "total": 2,
    "successful": 1,
    "failed": 0
  },
  "created": true
}
```


elastic

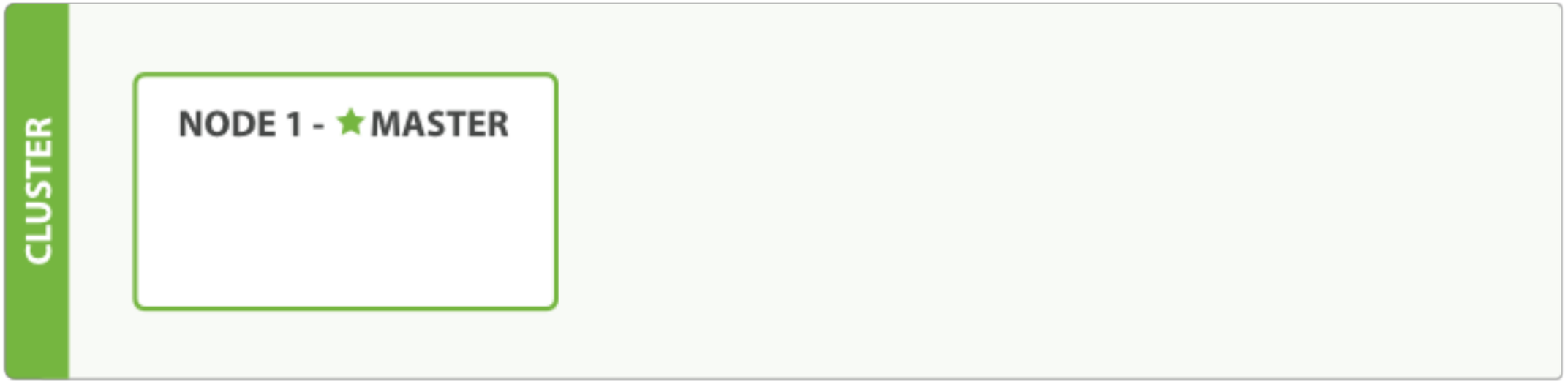

POST /databases/nosql
Client
Elasticsearch
201 Created

INITECH


POST /databases/nosql
Client
???
201 Created

Single node

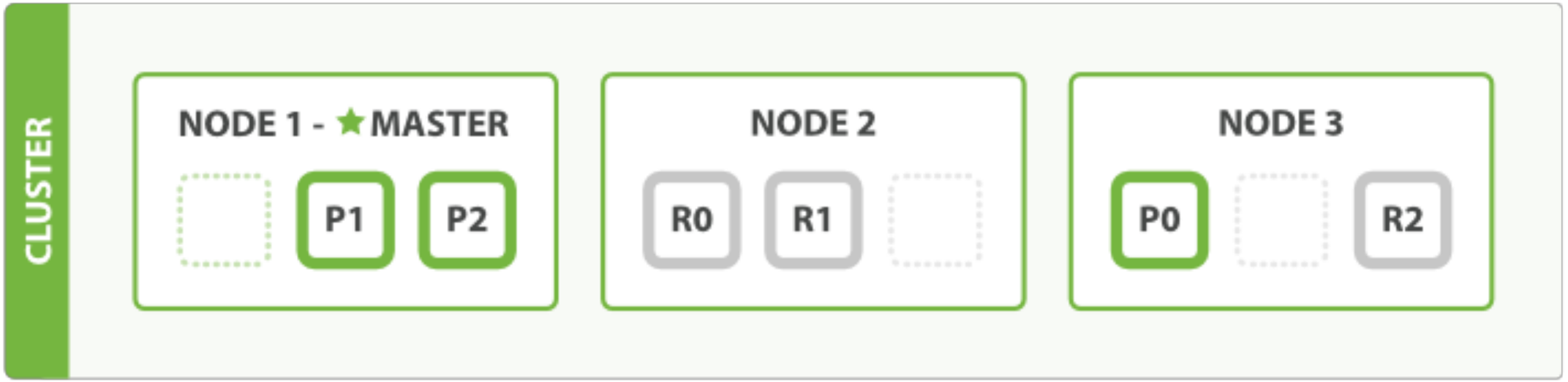
```
$ cd elasticsearch-<version>
$ ./bin/elasticsearch
```



elastic

Cluster

No more broadcasting



elastic

Node Types

Master-eligible node

Default by default

Data node

Default by default

Client node

Smart load balancer

Ingest node

Parse and enrich

ML node

Machine learning

Discovery

Zen discovery

Peer to peer network

Unicast Who to contact
Ping Discover each other

One master node

Elected or joined to

Three dedicated master nodes for production

Healthcheck

Master pings all nodes, they report back

discovery.zen.no_master_block

write | all

discovery.zen.minimum_master_nodes

$$\frac{master\_eligible\_nodes}{2} + 1$$

Split brain 💀

```
GET /_cluster/health
```

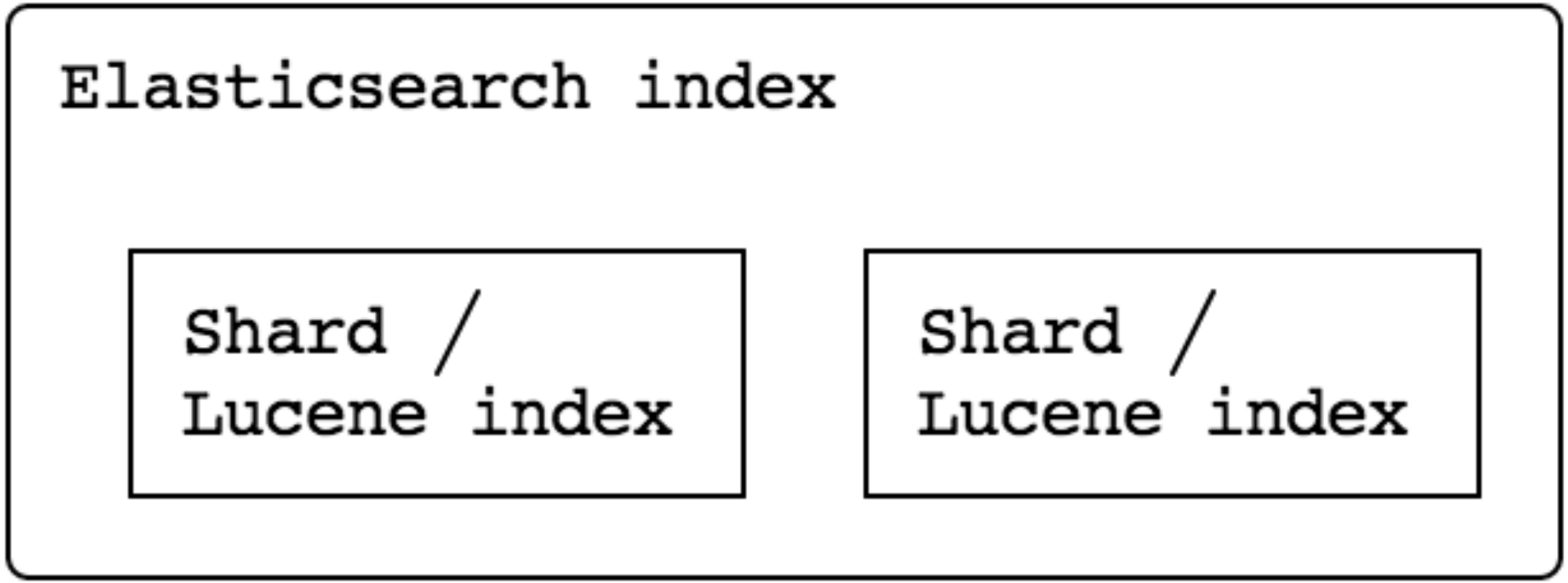
green yellow red

Alternative discovery

Azure, EC2, GCE

Indexing a Document

YOU KEEP USING THAT WORD
I DO NOT THINK IT MEANS WHAT YOU THINK IT MEANS


Elasticsearch index
Shard /
Lucene index
Shard /
Lucene index

Document

Unique combination: `_index` `_type` `_id`

PS: Types will be removed

POST /databases/nosql

vs

PUT /databases/nosql/elasticsearch

Autogenerated ID

20 characters, URL-safe, Base64-encoded, GUID strings

`AVfD6ukyeuK3k1LGtSwT`

Consistent hashing

Before 2.0: djb2

Ignoring `_routing`

```c
unsigned long
hash(unsigned char *str)
{
    unsigned long hash = 5381;
    int c;

    while (c = *str++)
        hash = ((hash << 5) + hash) + c; /* hash * 33 + c */

    return hash;
}
```

elastic

Consistent hashing

Current default: murmur3

https://github.com/elastic/elasticsearch/blob/5.4/core/src/main/java/org/elasticsearch/common/hash/MurmurHash3.java

Consistent hashing

Better distribution

100,000 incremental IDs
https://github.com/elastic/elasticsearch/pull/7954

Consistent hashing

3 shards

murmur3 [33185, 33347, 33468]
djb2 [30100, 30000, 39900]

Consistent hashing

5 shards

murmur3 [19933, 19964, 19940, 20030, 20133]
djb2 [20000, 20000, 20000, 20000, 20000]

Consistent hashing

33 shards

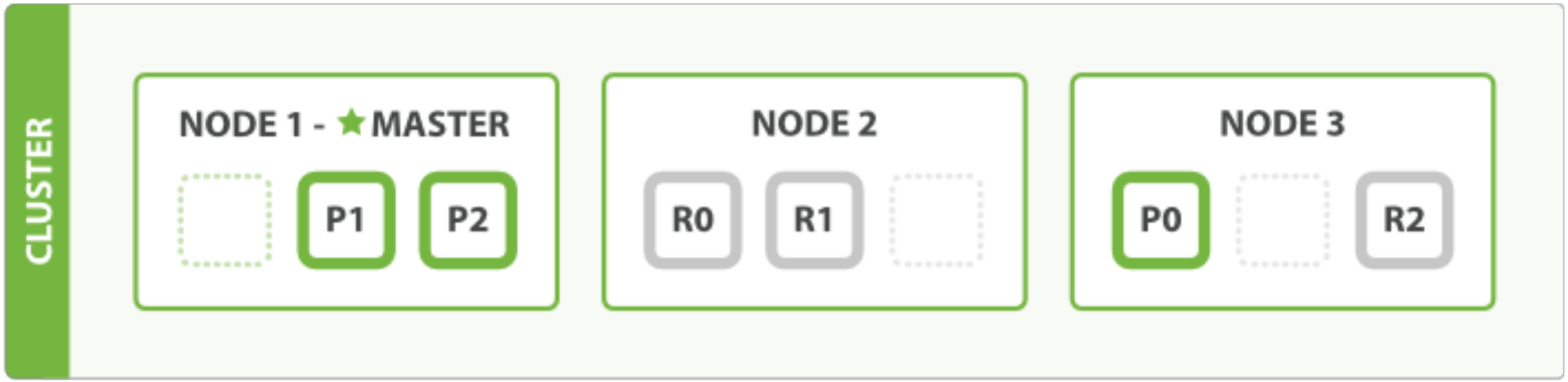
murmur3 [2999, 3096, 2930, 2986, 3070, 3093, 3023, 3052, 3112, 2940, 3036, 2985, 3031, 3048, 3127, 2961, 2901, 3105, 3041, 3130, 3013, 3035, 3031, 3019, 3008, 3022, 3111, 3086, 3016, 2996, 3075, 2945, 2977]
djb2 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 900, 900, 900, 900, 1000, 1000, 10000, 10000, 10000, 10000, 9100, 9100, 9100, 9100, 9000, 9000, 0, 0, 0, 0, 0, 0]

Shard decision

```
shard = hash(doc_id) % (num_of_primary_shards)
```

Write

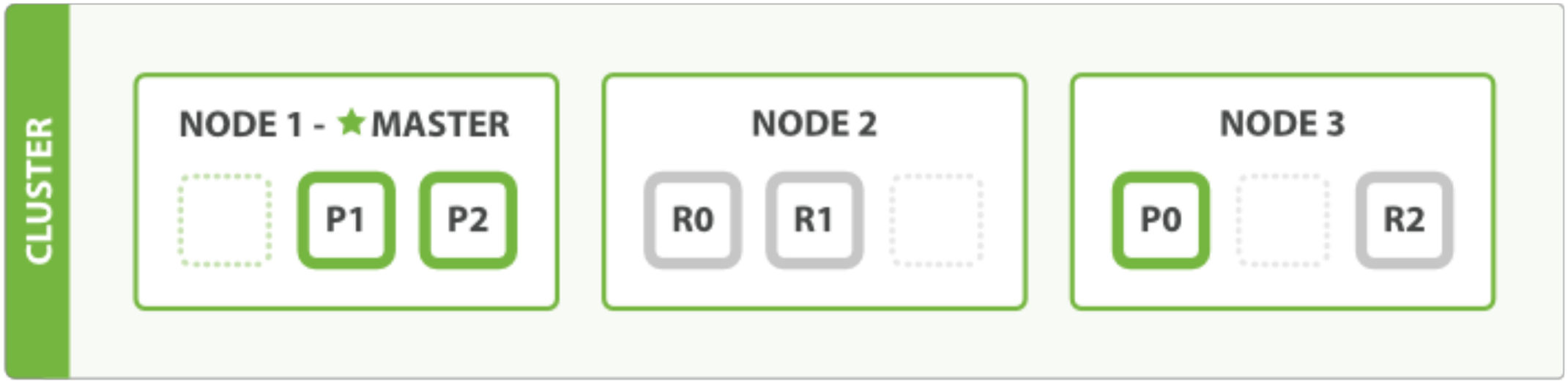
Coordinating Node, Hash, Primary, Replica(s)



elastic

Get & Aggregate

Coordinating Node, Hash, Shard



elastic

Optimistic concurrency control

`_version`

Flaky nodes

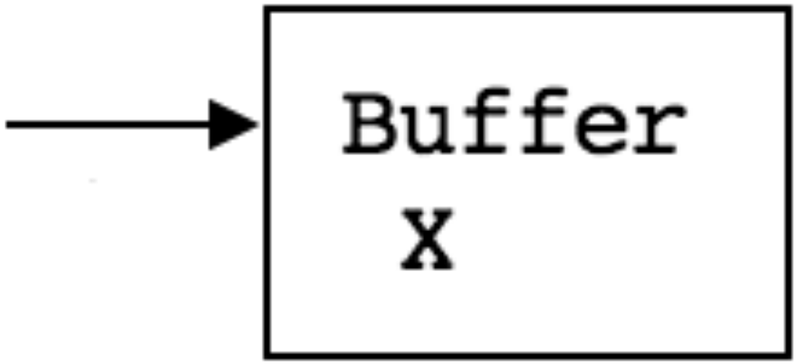
```
index.unassigned.node_left.delayed_timeout: 1m
```

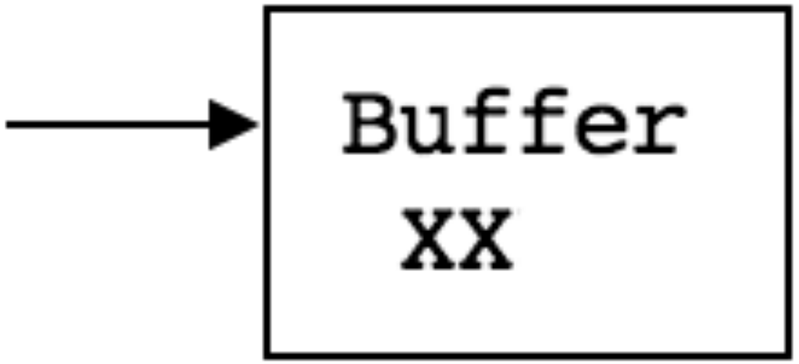
Sequency numbers

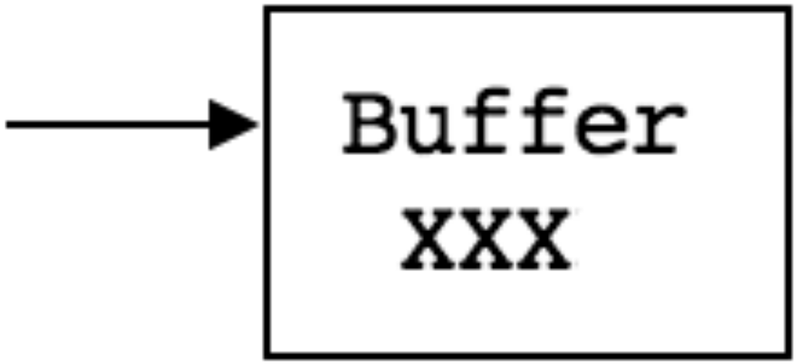
Quick recovery in 6.0

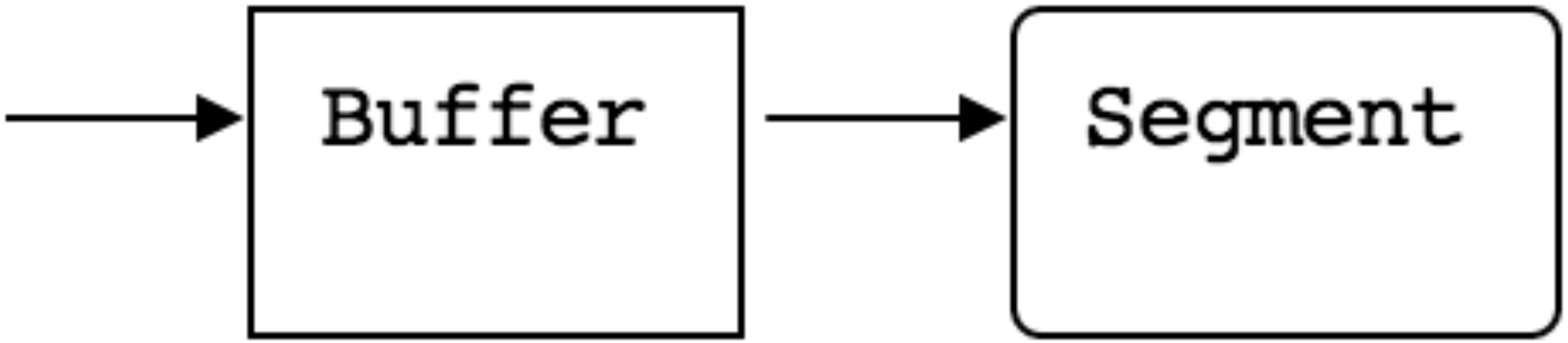
Lucene Segment

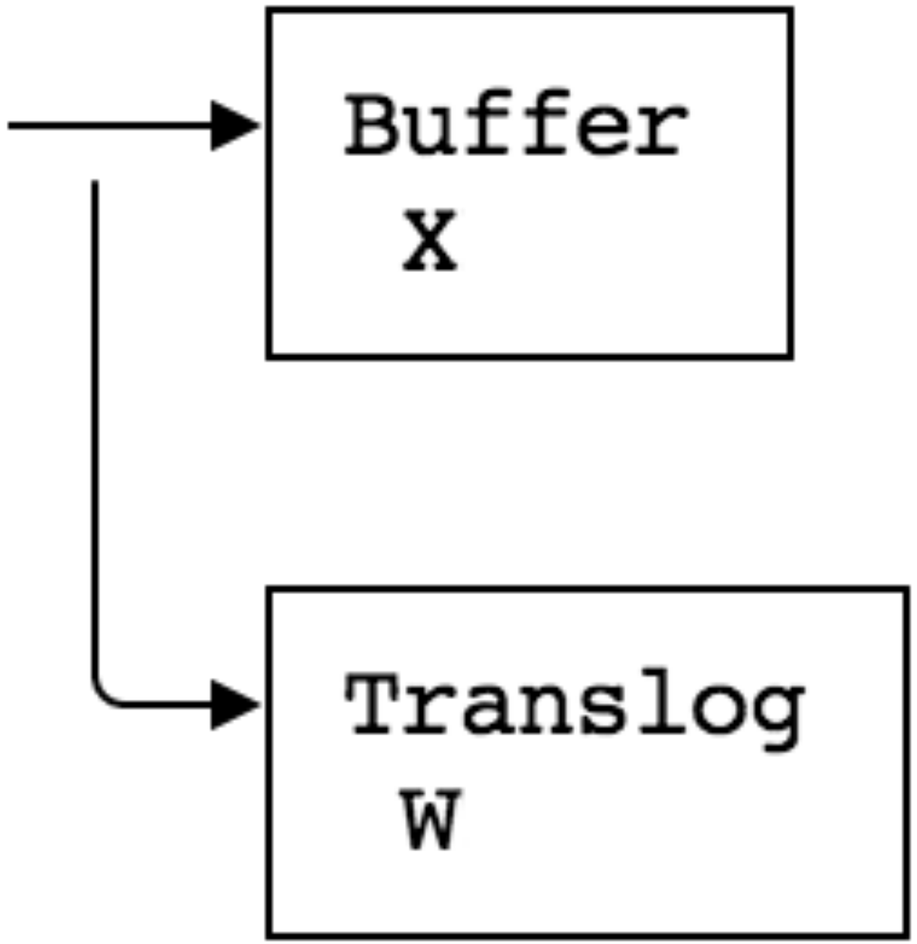
Lucene at work

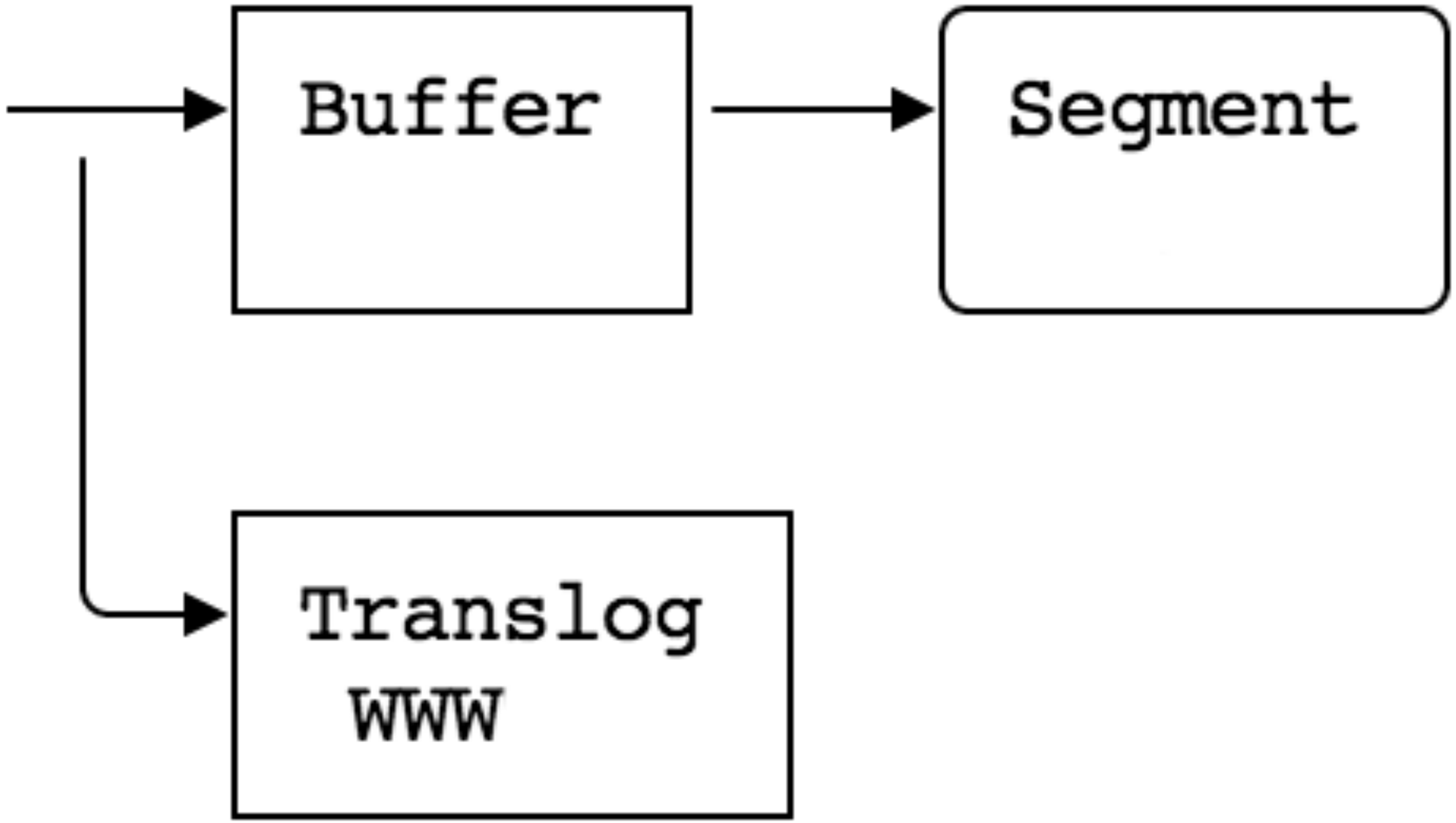

Buffer
X

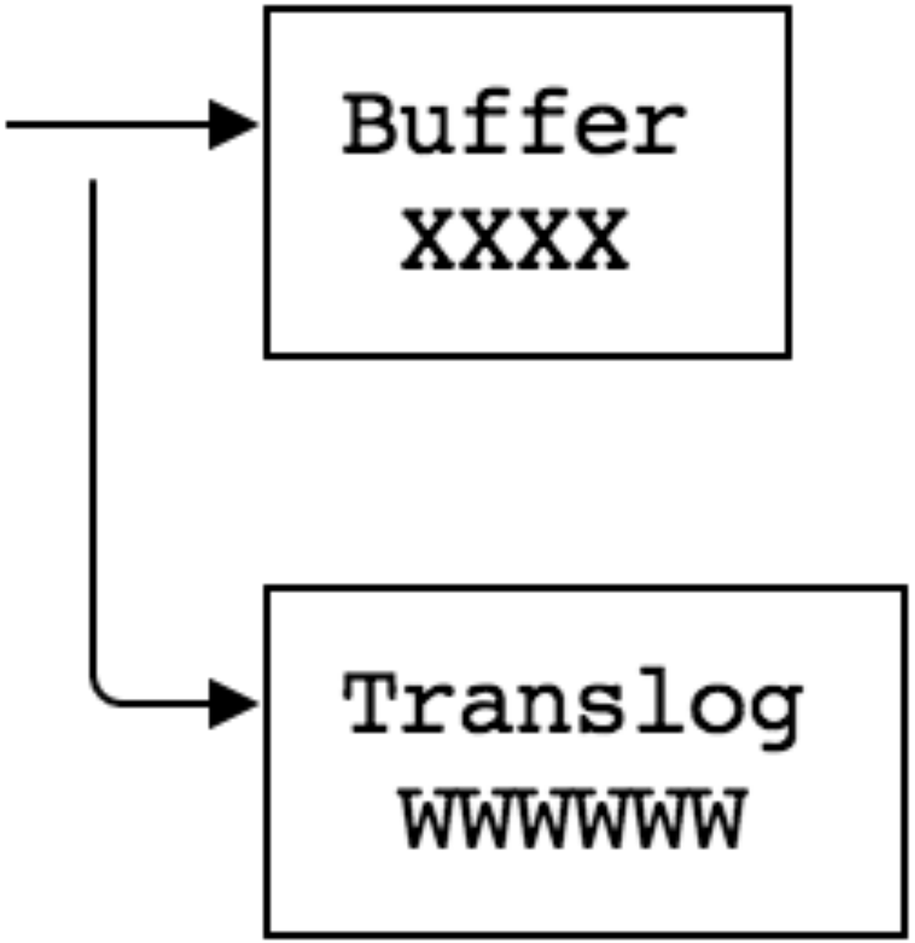

Buffer
XX

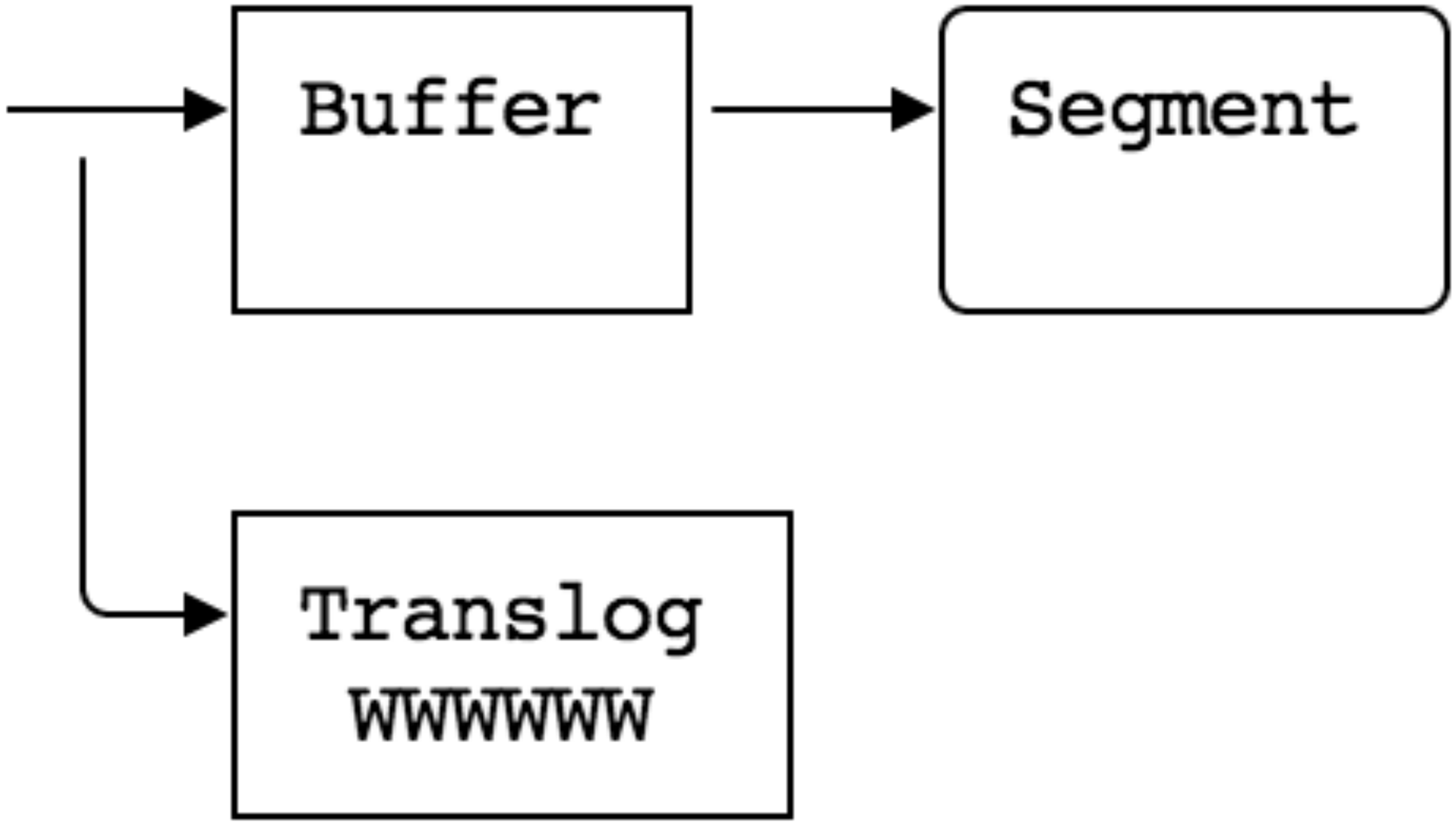

Buffer
XXX

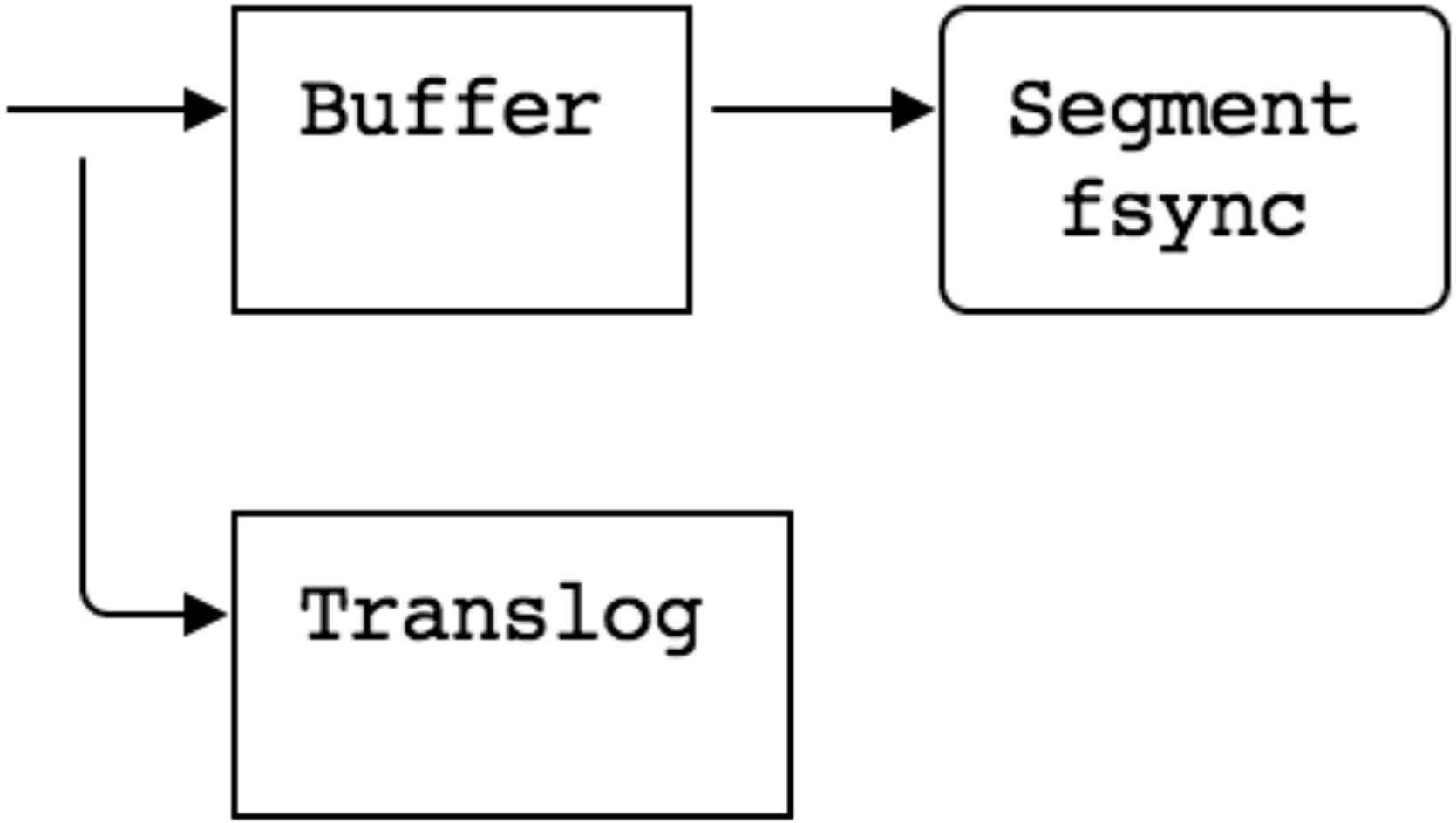

Buffer
Segment


Buffer
X
Translog
W


Buffer
Segment
Translog
WWW


Buffer
XXXX
Translog
WWWWWW


Buffer
Segment
Translog
WWWWWW


Buffer
Segment
fsync
Translog

Segments are immutable

`index.refresh_interval`

7.0: `index.search.idle.after`

Tombstone file

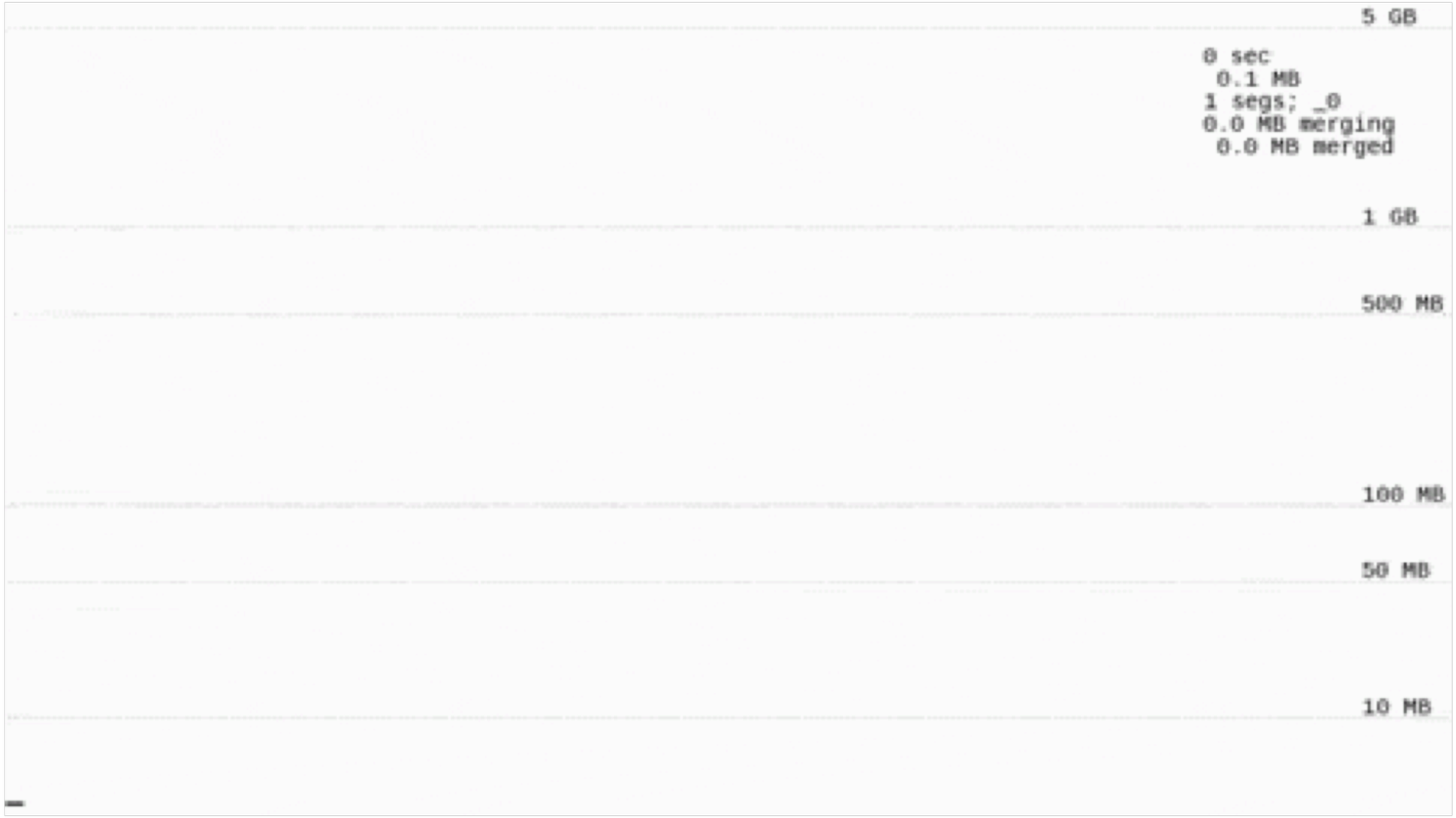
Marks deleted documents

Merge

Combine (and clean up) segments

Visualize merges

http://blog.mikemccandless.com/2011/02/visualizing-lucenes-segment-merges.html

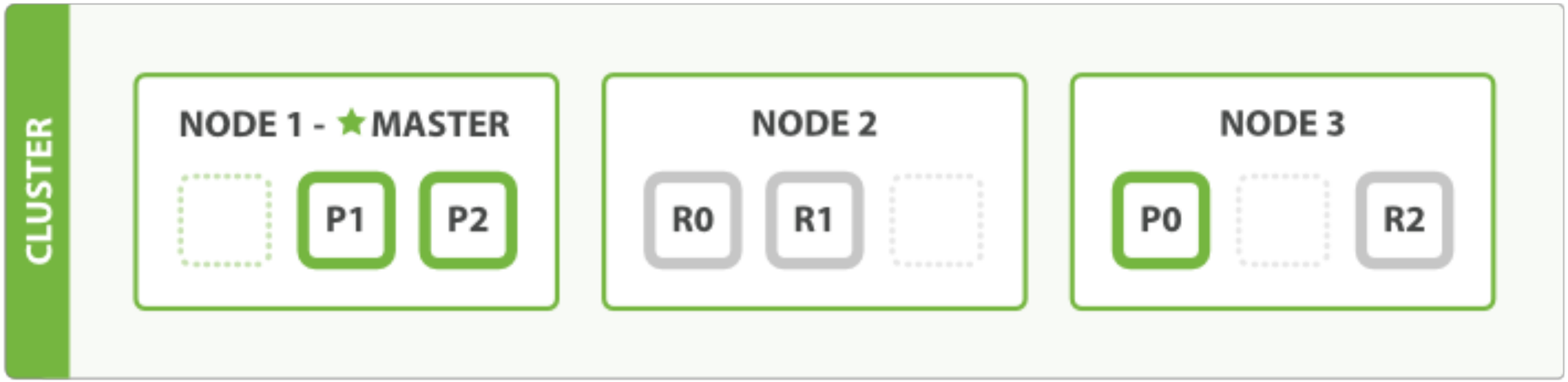

5 GB
0 sec
0.1 MB
1 segs; _0
0.0 MB merging
0.0 MB merged
1 GB
500 MB
100 MB
50 MB
10 MB

elastic

Searching

Search

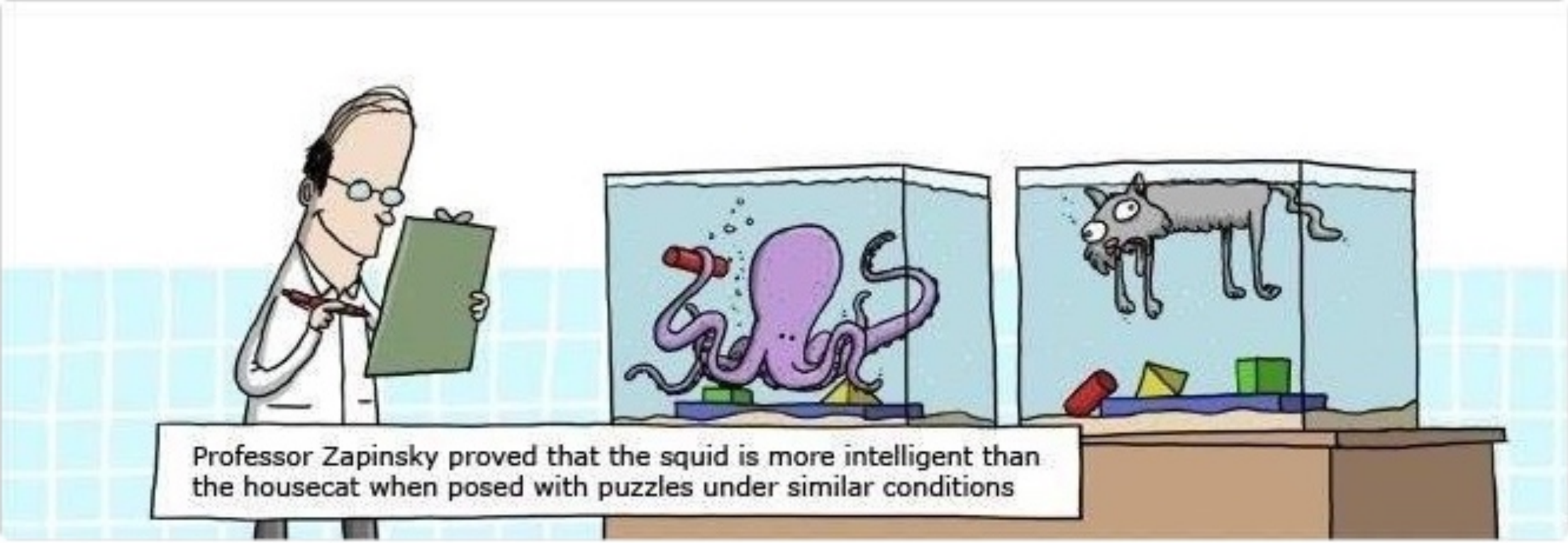
Coordinating Node, Query then Fetch



elastic

Benchmarks

Fair
Reproducible
Close to Production


Professor Zapinsky proved that the squid is more intelligent than the housecat when posed with puzzles under similar conditions

Full-Text Search

elastic

elasticsearch

elastic

Who uses a Database?

elastic

Who uses Search?

elastic

elastic

Store

Apache Lucene
Elasticsearch

Example

These are not the droids you are looking for.

`html_strip` Char Filter

These are not the droids you are looking for.

`standard` Tokenizer

These are not the droids you are looking for

`lowercase` Token Filter

these are not the droids you are looking for

`stop` Token Filter

droids you looking

`snowball` Token Filter

droid you look

Analyze

```
GET /_analyze
{
  "analyzer": "english",
  "text": "These are not the droids you are looking for."
}
```

```
{
  "tokens": [
    {
      "token": "droid",
      "start_offset": 18,
      "end_offset": 24,
      "type": "<ALPHANUM>",
      "position": 4
    },
    {
      "token": "you",
      "start_offset": 25,
      "end_offset": 28,
      "type": "<ALPHANUM>",
      "position": 5
    },
    ...
  ]
}
```

```
GET /_analyze
{
  "char_filter": [
    "html_strip"
  ],
  "tokenizer": "standard",
  "filter": [
    "lowercase",
    "stop",
    "snowball"
  ],
  "text": "These are <em>not</em> the droids you are looking for."
}
```

```
{
  "tokens": [
    {
      "token": "droid",
      "start_offset": 27,
      "end_offset": 33,
      "type": "<ALPHANUM>",
      "position": 4
    },
    {
      "token": "you",
      "start_offset": 34,
      "end_offset": 37,
      "type": "<ALPHANUM>",
      "position": 5
    },
    ...
  ]
}
```

Stop Words

a an and are as at be but by for if in into is it no not of on or such that the their then there these they this to was will with

https://github.com/apache/lucene-solr/blob/master/lucene/analysis/common/src/java/org/apache/lucene/analysis/en/EnglishAnalyzer.java#L44-L50

Always Use Stop Words?

To be, or not to be.

Languages

Arabic, Armenian, Basque, Brazilian, Bulgarian, Catalan, CJK, Czech, Danish, Dutch, English, Finnish, French, Galician, German, Greek, Hindi, Hungarian, Indonesian, Irish, Italian, Latvian, Lithuanian, Norwegian, Persian, Portuguese, Romanian, Russian, Sorani, Spanish, Swedish, Turkish, Thai

Language Rules

English: `Philipp's` → `philipp`

French: `l'église` → `eglis`

German: `äußerst` → `ausserst`

More Language Plugins

Core: ICU (Asian languages), Kuromoji (advanced Japanese), Phonetic, SmartCN, Stempel (Polish), Ukrainian

Community: Hebrew, Vietnamese, Network Address Analysis, String2Integer,...

German

```
GET /_analyze
{
  "analyzer": "german",
  "text": "Das sind nicht die Droiden, nach denen du suchst."
}
```

```
{
  "tokens": [
    {
      "token": "droid",
      "start_offset": 19,
      "end_offset": 26,
      "type": "<ALPHANUM>",
      "position": 4
    },
    {
      "token": "den",
      "start_offset": 33,
      "end_offset": 38,
      "type": "<ALPHANUM>",
      "position": 6
    },
    {
      "token": "such",
      "start_offset": 42,
      "end_offset": 48,
      "type": "<ALPHANUM>",
      "position": 8
    }
  ]
}
```

German with the English Analyzer

da sind nicht die droiden nach denen du suchst

elastic

German Stop Words

https://github.com/apache/lucene-solr/blob/master/lucene/analysis/common/src/resources/org/apache/lucene/analysis/snowball/german_stop.txt

Detect Languages

https://github.com/spinscale/elasticsearch-ingest-langdetect

```
PUT _ingest/pipeline/langdetect-pipeline
{
  "description": "A pipeline to detect languages",
  "processors": [
    {
      "langdetect" : {
        "field" : "quote",
        "target_field" : "language"
      }
    }
  ]
}
```

```
POST _ingest/pipeline/langdetect-pipeline/_simulate
{
  "docs": [
    {
      "_source": {
        "quote": "Das sind nicht die Droiden, nach denen du suchst."
      }
    }
  ]
}
```

```
{
  "docs": [
    {
      "doc": {
        "_index": "_index",
        "_type": "_type",
        "_id": "_id",
        "_source": {
          "language": "de",
          "quote": "Das sind nicht die Droiden, nach denen du suchst."
        },
        "_ingest": {
          "timestamp": "2018-10-26T00:06:42.320613Z"
        }
      }
    }
  ]
}
```


elastic

Phonetic

```
GET /_analyze
{
  "tokenizer": "standard",
  "filter": [
    {
      "type": "phonetic",
      "encoder": "beider_morse",
      "languageset": "any"
    }
  ],
  "text": "These are not the droids you are looking for."
}
```

Phonetic

... drDts drits drots iou ari ori loknk...

Another Example

Obi-Wan never told you what happened to your father.

Another Example

obi wan never told you what happen your father

Another Example

No. I am your father.

Another Example

i am your father

Inverted Index

	ID 1	ID 2	ID 3
am	0	0	1[2]
droid	1[4]	0	0
father	0	1[9]	1[4]
happen	0	1[6]	0
i	0	0	1[1]
look	1[7]	0	0
never	0	1[2]	0
obi	0	1[0]	0
told	0	1[3]	0
wan	0	1[1]	0
what	0	1[5]	0
you	1[5]	1[4]	0
your	0	1[8]	1[3]

To / The Index

```
PUT /starwars
{
  "settings": {
    "number_of_shards": 1,
    "analysis": {
      "filter": {
        "my_synonym_filter": {
          "type": "synonym",
          "synonyms": [
            "father,dad",
            "droid => droid,machine"
          ]
        }
      },
```

elastic

```
    "analyzer": {
      "my_analyzer": {
        "char_filter": [
          "html_strip"
        ],
        "tokenizer": "standard",
        "filter": [
          "lowercase",
          "stop",
          "snowball",
          "my_synonym_filter"
        ]
      }
    }
  }
},
```

elastic

```
"mappings": {
  "_doc": {
    "properties": {
      "quote": {
        "type": "text",
        "analyzer": "my_analyzer"
      }
    }
  }
}
}
```

Synonyms

Index `synonym` **or query time** `synonym_graph`

```
GET /starwars/_mapping

GET /starwars/_settings
```

```
PUT /starwars/_doc/1
{
  "quote": "These are <em>not</em> the droids you are looking for."
}
PUT /starwars/_doc/2
{
  "quote": "Obi-Wan never told you what happened to your father."
}
PUT /starwars/_doc/3
{
  "quote": "<b>No</b>. I am your father."
}
```

```
GET /starwars/_doc/1

GET /starwars/_doc/1/_source
```

Multi Lingual

Index

```
PUT /starwars_en/_doc/1
```

~~Type~~

Field

```
{ "quote_en": "...", "quote_de": "..." }
```

PS: Single Type per Index

Search

```
POST /starwars/_search
{
  "query": {
    "match_all": { }
  }
}
```

GET **VS** POST

```
{
  "took": 1,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "failed": 0
  },
  "hits": {
    "total": 3,
    "max_score": 1,
    "hits": [
      {
        "_index": "starwars",
        "_type": "_doc",
        "_id": "2",
        "_score": 1,
        "_source": {
          "quote": "Obi-Wan never told you what happened to your father."
        }
      },
      ...
```


elastic

```
POST /starwars/_search
{
  "query": {
    "match": {
      "quote": "droid"
    }
  }
}
```

```
{
  "took": 2,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "failed": 0
  },
  "hits": {
    "total": 1,
    "max_score": 0.39556286,
    "hits": [
      {
        "_index": "starwars",
        "_type": "_doc",
        "_id": "1",
        "_score": 0.39556286,
        "_source": {
          "quote": "These are <em>not</em> the droids you are looking for."
        }
      }
    ]
  }
}
```


elastic

```
POST /starwars/_search
{
  "query": {
    "match": {
      "quote": "dad"
    }
  }
}
```

```
...
  "hits": {
    "total": 2,
    "max_score": 0.41913947,
    "hits": [
      {
        "_index": "starwars",
        "_type": "_doc",
        "_id": "3",
        "_score": 0.41913947,
        "_source": {
          "quote": "<b>No</b>. I am your father."
        }
      },
      {
        "_index": "starwars",
        "_type": "_doc",
        "_id": "2",
        "_score": 0.39291072,
        "_source": {
          "quote": "Obi-Wan never told you what happened to your father."
        }
      }
    ]
  }
}
```


elastic

```
POST /starwars/_doc/0/_explain
{
  "query": {
    "match": {
      "quote": "dad"
    }
  }
}
```

```
{
  "_index": "starwars",
  "_type": "_doc",
  "_id": "0",
  "matched": false
}
```

```
POST /starwars/_doc/1/_explain
{
  "query": {
    "match": {
      "quote": "dad"
    }
  }
}
```

```
{
  "_index": "starwars",
  "_type": "_doc",
  "_id": "1",
  "matched": false,
  "explanation": {
    "value": 0,
    "description": "no matching term",
    "details": []
  }
}
```

```
POST /starwars/_doc/2/_explain
{
  "query": {
    "match": {
      "quote": "dad"
    }
  }
}
```

```
{
  "_index": "starwars",
  "_type": "_doc",
  "_id": "2",
  "matched": true,
  "explanation": {
    ...
```

elastic

```
POST /starwars/_search
{
  "query": {
    "match": {
      "quote": "machine"
    }
  }
}
```

```
{
  "took": 2,
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 1,
    "max_score": 1.2499592,
    "hits": [
      {
        "_index": "starwars",
        "_type": "_doc",
        "_id": "1",
        "_score": 1.2499592,
        "_source": {
          "quote": "These are <em>not</em> the droids you are looking for."
        }
      }
    ]
  }
}
```


elastic

```
POST /starwars/_search
{
  "query": {
    "match_phrase": {
      "quote": "I am your father"
    }
  }
}
```

elastic

```
{
  "took": 3,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "failed": 0
  },
  "hits": {
    "total": 1,
    "max_score": 1.5665855,
    "hits": [
      {
        "_index": "starwars",
        "_type": "_doc",
        "_id": "3",
        "_score": 1.5665855,
        "_source": {
          "quote": "<b>No</b>. I am your father."
        }
      }
    ]
  }
}
```

```
POST /starwars/_search
{
  "query": {
    "match_phrase": {
      "quote": {
        "query": "I am father",
        "slop": 1
      }
    }
  }
}
```

elastic

```
{
  "took": 16,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "failed": 0
  },
  "hits": {
    "total": 1,
    "max_score": 0.8327639,
    "hits": [
      {
        "_index": "starwars",
        "_type": "_doc",
        "_id": "3",
        "_score": 0.8327639,
        "_source": {
          "quote": "<b>No</b>. I am your father."
        }
      }
    ]
  }
}
```

```
POST /starwars/_search
{
  "query": {
    "match_phrase": {
      "quote": {
        "query": "I am not your father",
        "slop": 1
      }
    }
  }
}
```

elastic

```
{
  "took": 5,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "failed": 0
  },
  "hits": {
    "total": 1,
    "max_score": 1.0409548,
    "hits": [
      {
        "_index": "starwars",
        "_type": "_doc",
        "_id": "3",
        "_score": 1.0409548,
        "_source": {
          "quote": "<b>No</b>. I am your father."
        }
      }
    ]
  }
}
```

```
POST /starwars/_search
{
  "query": {
    "match": {
      "quote": {
        "query": "van",
        "fuzziness": "AUTO"
      }
    }
  }
}
```

```
{
  "took": 14,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "failed": 0
  },
  "hits": {
    "total": 1,
    "max_score": 0.18155496,
    "hits": [
      {
        "_index": "starwars",
        "_type": "_doc",
        "_id": "2",
        "_score": 0.18155496,
        "_source": {
          "quote": "Obi-Wan never told you what happened to your father."
        }
      }
    ]
  }
}
```


elastic

```
POST /starwars/_search
{
  "query": {
    "match": {
      "quote": {
        "query": "ovi-van",
        "fuzziness": 1
      }
    }
  }
}
```

```
{
  "took": 109,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "failed": 0
  },
  "hits": {
    "total": 1,
    "max_score": 0.3798467,
    "hits": [
      {
        "_index": "starwars",
        "_type": "_doc",
        "_id": "2",
        "_score": 0.3798467,
        "_source": {
          "quote": "Obi-Wan never told you what happened to your father."
        }
      }
    ]
  }
}
```

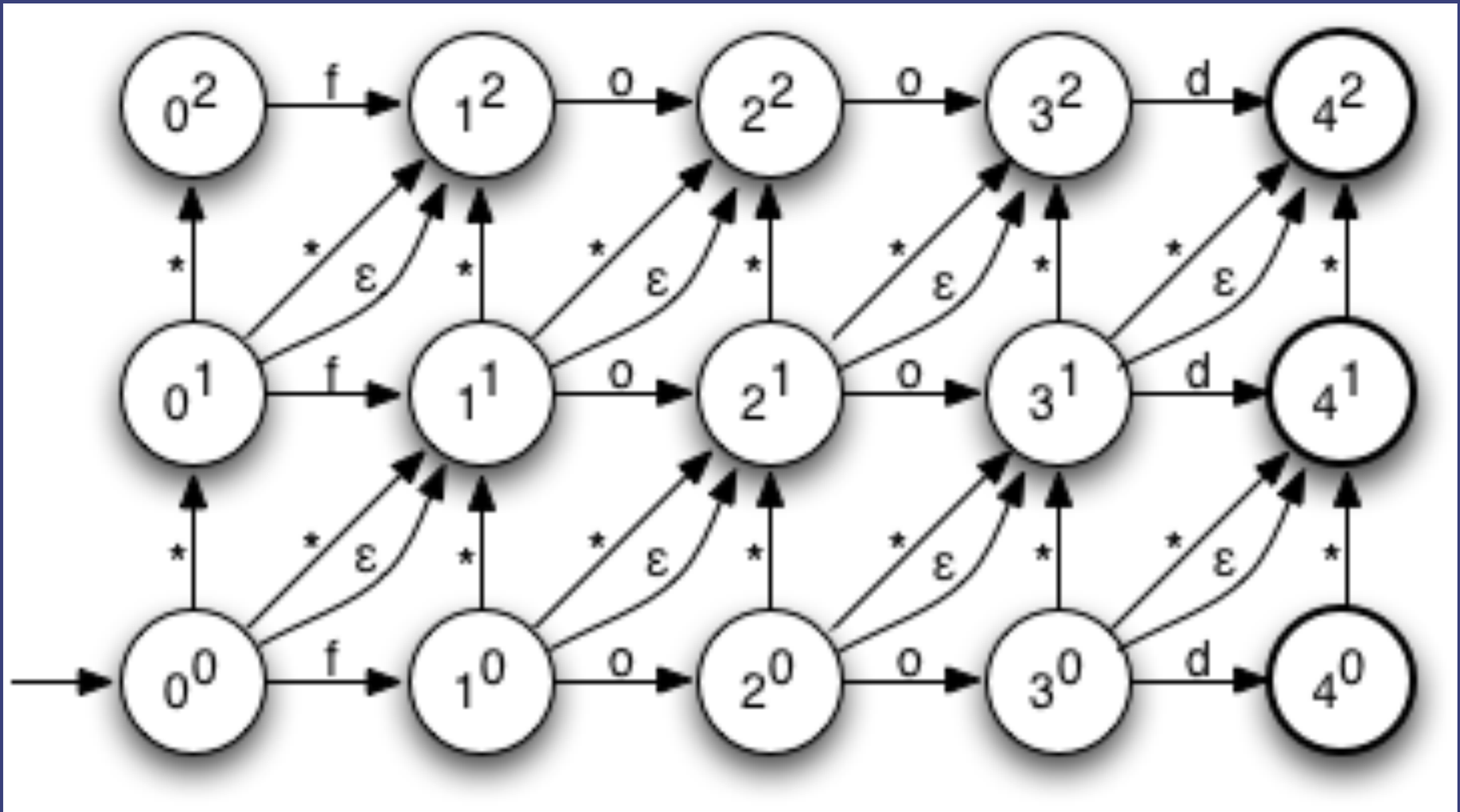

elastic

FuzzyQuery History

http://blog.mikemccandless.com/2011/03/lucenes-fuzzyquery-is-100-times-faster.html

Before: Brute force

Now: Levenshtein Automaton



http://blog.notdot.net/2010/07/Damn-Cool-Algorithms-Levenshtein-Automata

```
SELECT *
  FROM starwars
 WHERE quote LIKE "?an" OR
       quote LIKE "V?n" OR
       quote LIKE "Va?"
```

elastic

Score

Term Frequency / Inverse Document Frequency (TF/IDF)

Search one term

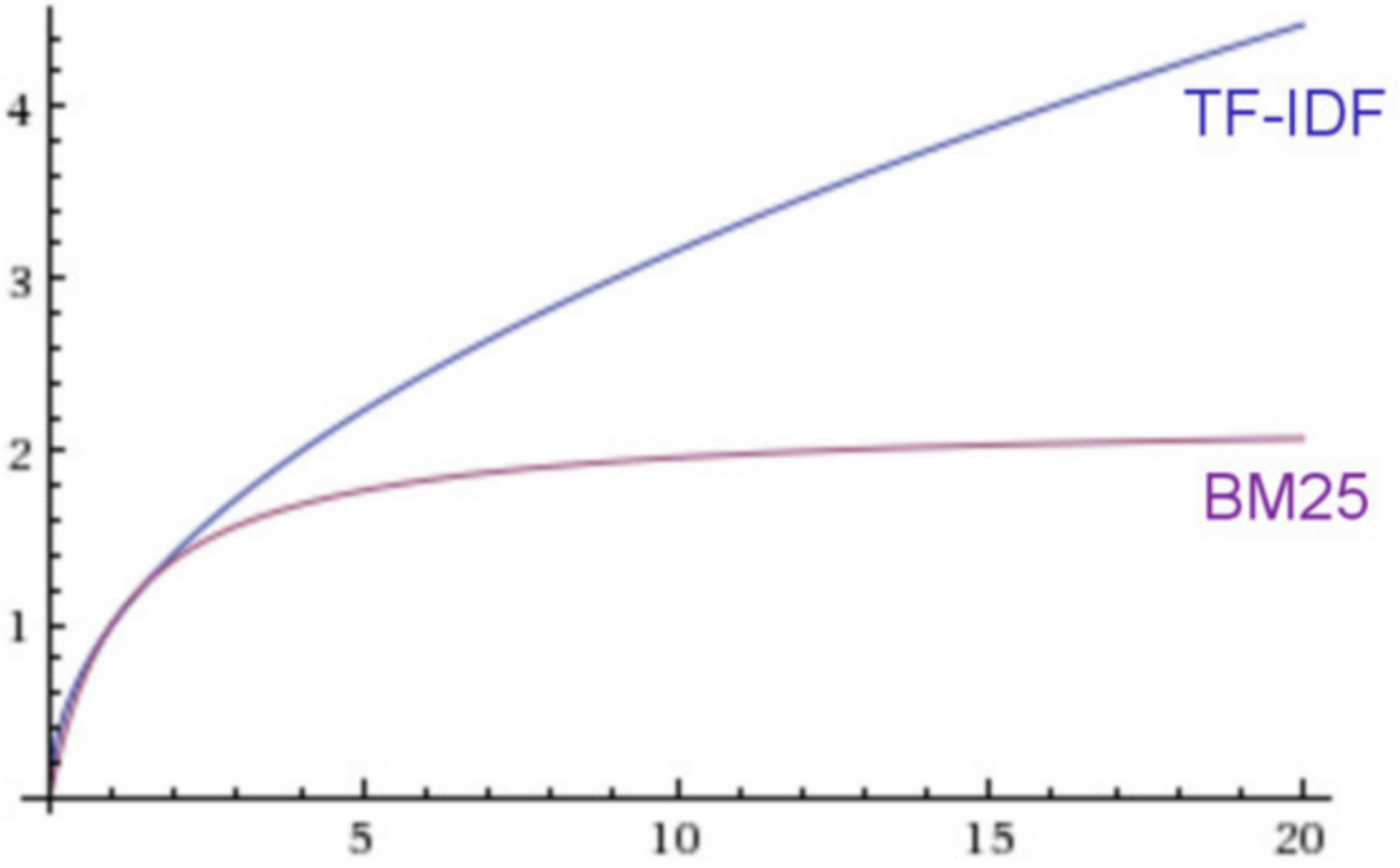
BM25

Default in Elasticsearch 5.0

https://speakerdeck.com/elastic/improved-text-scoring-with-bm25

Term Frequency

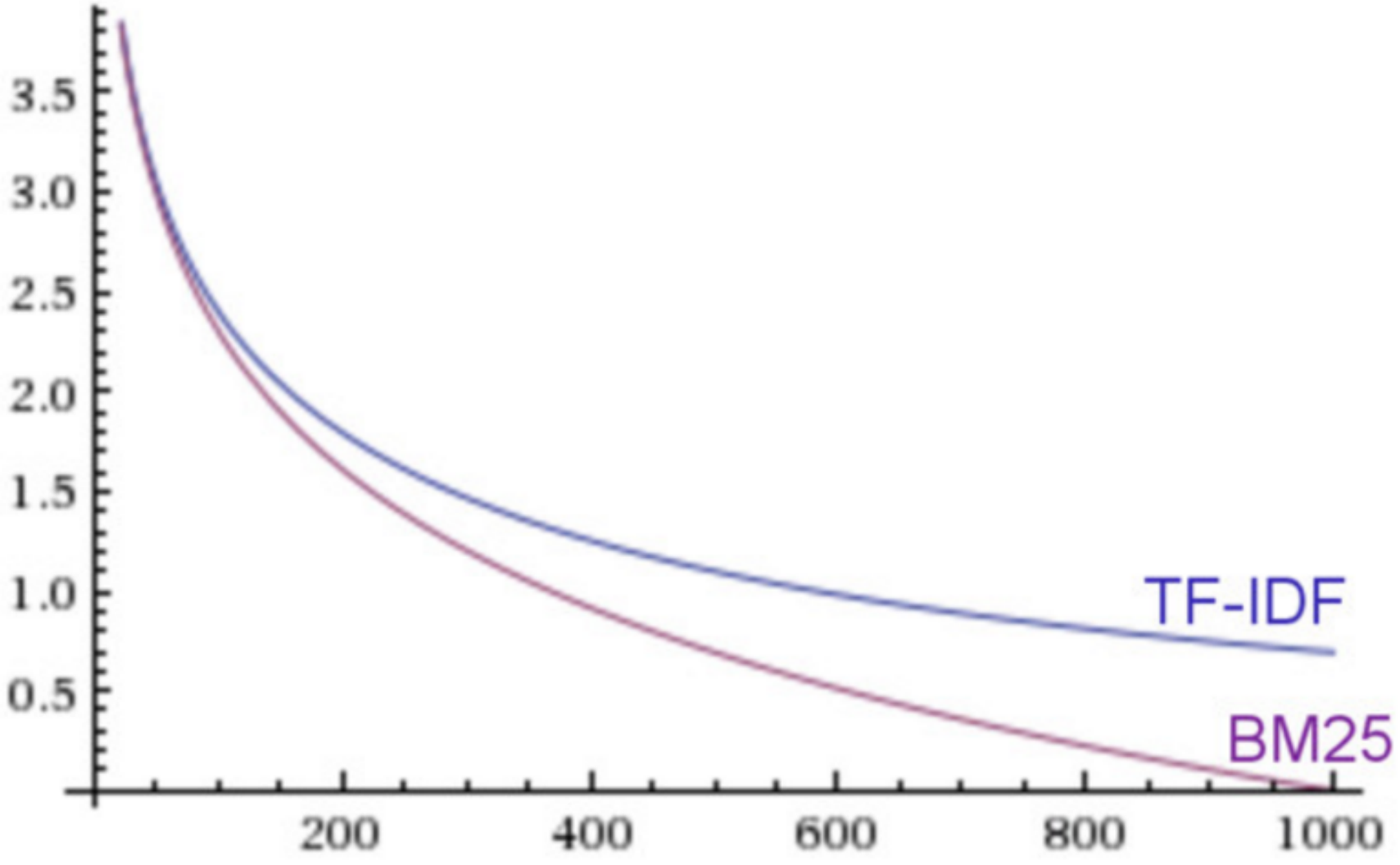
$$tf(t\ in\ d) = \sqrt{frequency}$$



elastic

Inverse Document Frequency

$$idf(t) = 1 + log(\frac{numDocs}{docFreq + 1})$$



elastic

Field-Length Norm

$$norm(d) = \frac{1}{\sqrt{numTerms}}$$

```
POST /starwars/_search?explain=true
{
  "query": {
    "match": {
      "quote": "father"
    }
  }
}
```

```
...
"_explanation": {
  "value": 0.41913947,
  "description": "weight(Synonym(quote:dad quote:father) in 0) [PerFieldSimilarity], result of:",
  "details": [
    {
      "value": 0.41913947,
      "description": "score(doc=0,freq=2.0 = termFreq=2.0\n), product of:",
      "details": [
        {
          "value": 0.2876821,
          "description": "idf(docFreq=1, docCount=1)",
          "details": []
        },
        {
          "value": 1.4569536,
          "description": "tfNorm, computed from:",
          "details": [
            {
              "value": 2,
              "description": "termFreq=2.0",
              "details": []
            },
            ...
```


elastic

Score

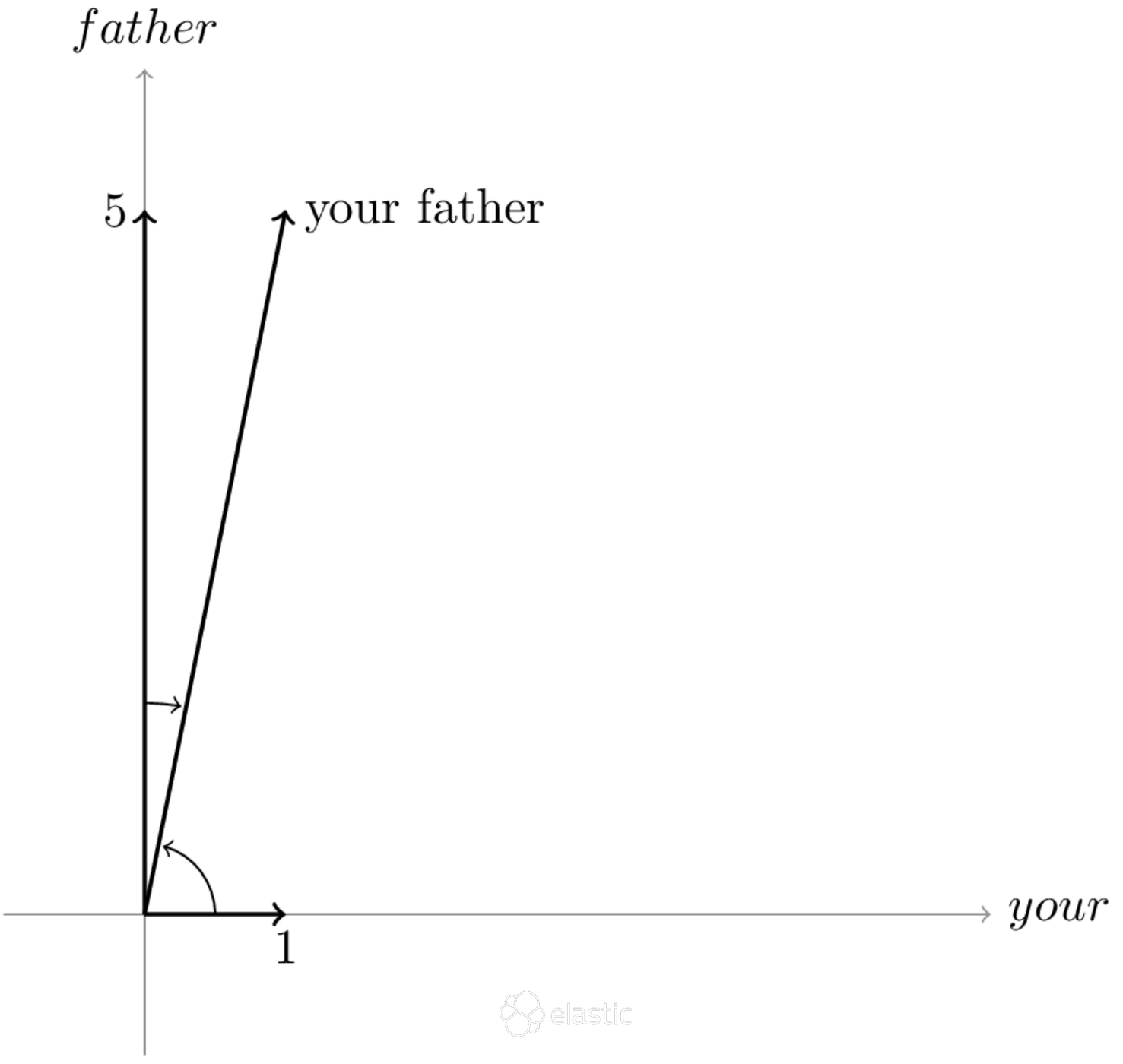
0.41913947: i am your father

0.39291072: obi wan never told you what happen your father

Vector Space Model

Search multiple terms

Search your father


father
5
your father
your
1

Coordination Factor

Reward multiple terms

Search for 3 terms

1 term: $X \cdot \frac{1}{3}$

2 terms: $(X + Y) \cdot \frac{2}{3}$

3 terms: $(X + Y + Z) \cdot \frac{3}{3}$

Practical Scoring Function

Putting it all together

elastic

```
score(q,d)  =
            queryNorm(q)
          · coord(q,d)
          · Σ (
                tf(t in d)
              · idf(t)²
              · t.getBoost()
              · norm(t,d)
            ) (t in q)
```

Function Score

Script, weight, random, field value, decay (geo or date)

```
POST /starwars/_search
{
  "query": {
    "function_score": {
      "query": {
        "match": {
          "quote": "father"
        }
      },
      "random_score": {}
    }
  }
}
```

elastic

Compare Scores

"100% perfect" vs a "50%" match

Don't do this. Seriously. Stop trying to think about your problem this way, it's not going to end well.

— https://wiki.apache.org/lucene-java/ScoresAsPercentages

elastic

```
GET /starwars/_analyze
{
  "analyzer" : "my_analyzer",
  "text": "These are my father's machines."
}
```

```
{ "tokens": [
    {
      "token": "my",
      "start_offset": 10,
      "end_offset": 12,
      "type": "<ALPHANUM>",
      "position": 2
    },
    {
      "token": "father",
      "start_offset": 13,
      "end_offset": 21,
      "type": "<ALPHANUM>",
      "position": 3
    },
    {
      "token": "dad",
      "start_offset": 13,
      "end_offset": 21,
      "type": "SYNONYM",
      "position": 3
    },
    {
      "token": "machin",
      "start_offset": 22,
      "end_offset": 30,
      "type": "<ALPHANUM>",
      "position": 4
    }
] }
```

```
PUT /starwars/_doc/4
{
  "quote": "These are my father's machines."
}
```

```
POST /starwars/_search
{
  "query": {
    "match": {
      "quote": "my father machine"
    }
  }
}
```

```
"hits": {
  "total": 4,
  "max_score": 2.92523,
  "hits": [
    {
      "_index": "starwars",
      "_type": "_doc",
      "_id": "4",
      "_score": 2.92523,
      "_source": {
        "quote": "These are my father's machines."
      }
    },
    {
      "_index": "starwars",
      "_type": "_doc",
      "_id": "1",
      "_score": 0.8617505,
      "_source": {
        "quote": "These are <em>not</em> the droids you are looking for."
      }
    },
    ...
```


elastic

2.92523 == 100%

```
DELETE /starwars/_doc/4

POST /starwars/_search
{
  "query": {
    "match": {
      "quote": "my father machine"
    }
  }
}
```

```
"hits": {
  "total": 3,
  "max_score": 1.2499592,
  "hits": [
    {
      "_index": "starwars",
      "_type": "_doc",
      "_id": "1",
      "_score": 1.2499592,
      "_source": {
        "quote": "These are <em>not</em> the droids you are looking for."
      }
    },
    ...
```


elastic

1.2499592 == 43% or 100%?

```
PUT /starwars/_doc/4
{
  "quote": "These droids are my father's father's machines."
}

POST /starwars/_search
{
  "query": {
    "match": {
      "quote": "my father machine"
    }
  }
}
```

```
"hits": {
  "total": 4,
  "max_score": 3.0068164,
  "hits": [
    {
      "_index": "starwars",
      "_type": "_doc",
      "_id": "4",
      "_score": 3.0068164,
      "_source": {
        "quote": "These droids are my father's father's machines."
      }
    },
    {
      "_index": "starwars",
      "_type": "_doc",
      "_id": "1",
      "_score": 0.89701396,
      "_source": {
        "quote": "These are <em>not</em> the droids you are looking for."
      }
    },
    ...
```


elastic

3.0068164 == 103%?

PS: Shards

Default?

Effect on IDF?

Distributed Frequency Search

```
GET starwars/_search?search_type=dfs_query_then_fetch
{
  ...
}
```

Don't use `dfs_query_then_fetch` *in production. It really isn't required.*

— *https://www.elastic.co/guide/en/elasticsearch/guide/current/relevance-is-broken.html*

More Search

Highlighting

```
POST /starwars/_search
{
  "query": {
    "match": {
      "quote": "father"
    }
  },
  "highlight": {
    "type": "unified",
    "pre_tags": [
      "<tag>"
    ],
    "post_tags": [
      "</tag>"
    ],
    "fields": {
      "quote": {}
    }
  }
}
```

elastic

```
...
"hits": {
  "total": 3,
  "max_score": 0.631961,
  "hits": [
    {
      "_index": "starwars",
      "_type": "_doc",
      "_id": "4",
      "_score": 0.631961,
      "_source": {
        "quote": "These droids are my father's father's machines."
      },
      "highlight": {
        "quote": [
          "These droids are my <tag>father's</tag> <tag>father's</tag> machines."
        ]
      }
    },
...
```

Boolean Queries

`must` `must_not` `should` `filter`

```
POST /starwars/_search
{
  "query": {
    "bool": {
      "must": {
        "match": {
          "quote": "father"
        }
      },
      "should": [
        {
          "match": {
            "quote": "your"
          }
        },
        {
          "match": {
            "quote": "obi"
          }
        }
      ]
    }
  }
}
```

elastic

```
...
"hits": {
  "total": 3,
  "max_score": 2.117857,
  "hits": [
    {
      "_index": "starwars",
      "_type": "_doc",
      "_id": "2",
      "_score": 2.117857,
      "_source": {
        "quote": "Obi-Wan never told you what happened to your father."
      }
    },
    {
      "_index": "starwars",
      "_type": "_doc",
      "_id": "3",
      "_score": 1.3856719,
      "_source": {
        "quote": "<b>No</b>. I am your father."
      }
    },
    ...
```


elastic

```
POST /starwars/_search
{
  "query": {
    "bool": {
      "filter": {
        "match": {
          "quote": "father"
        }
      },
      "should": [
        {
          "match": {
            "quote": "your"
          }
        },
        {
          "match": {
            "quote": "obi"
          }
        }
      ]
    }
  }
}
```

elastic

```
...
"hits": {
  "total": 3,
  "max_score": 1.6694657,
  "hits": [
    {
      "_index": "starwars",
      "_type": "_doc",
      "_id": "2",
      "_score": 1.6694657,
      "_source": {
        "quote": "Obi-Wan never told you what happened to your father."
      }
    },
    {
      "_index": "starwars",
      "_type": "_doc",
      "_id": "3",
      "_score": 0.8317767,
      "_source": {
        "quote": "<b>No</b>. I am your father."
      }
    },
```


elastic

Named Queries & `minimum_should_match`

```
POST /starwars/_search
{
  "query": {
    "bool": {
      "must": {
        "match": { "quote": "father" }
      },
      "should": [
        {
          "match": {
            "quote": { "query": "your", "_name": "quote-your" }
          }
        },
        {
          "match": {
            "quote": { "query": "obi", "_name": "quote-obi" }
          }
        },
        {
          "match": {
            "quote": { "query": "droid", "_name": "quote-droid" }
          }
        }
      ],
      "minimum_should_match": 2
    }
  }
}
```

elastic

```
...
  "hits": {
    "total": 1,
    "max_score": 2.117857,
    "hits": [
      {
        "_index": "starwars",
        "_type": "_doc",
        "_id": "2",
        "_score": 2.117857,
        "_source": {
          "quote": "Obi-Wan never told you what happened to your father."
        },
        "matched_queries": [
          "quote-obi",
          "quote-your"
        ]
      }
    ]
  }
}
```

Boosting

>1 increase, <1 decrease, <0 punish

<0 removed in 7.0

```
POST /starwars/_search
{
  "query": {
    "bool": {
      "must": {
        "match": {
          "quote": "father"
        }
      },
      "should": [
        {
          "match": {
            "quote": "your"
          }
        },
        {
          "match": {
            "quote": {
              "query": "obi",
              "boost": 3
            }
          }
        }
      ]
    }
  }
}
```

elastic

```
...
"hits": {
  "total": 3,
  "max_score": 4.2368493,
  "hits": [
    {
      "_index": "starwars",
      "_type": "_doc",
      "_id": "2",
      "_score": 4.2368493,
      "_source": {
        "quote": "Obi-Wan never told you what happened to your father."
      }
    },
    {
      "_index": "starwars",
      "_type": "_doc",
      "_id": "3",
      "_score": 1.3856719,
      "_source": {
        "quote": "<b>No</b>. I am your father."
      }
    },
    ...
```


elastic

Search for `father`
but prefer `father father`

```
POST /starwars/_search
{
  "query": {
    "bool": {
      "must": {
        "match": {
          "quote": "father father"
        }
      }
    }
  }
}
```

elastic

```
...
  "hits": [
    {
      "_index": "starwars",
      "_type": "_doc",
      "_id": "4",
      "_score": 1.263922,
      "_source": {
        "quote": "These droids are my father's father's machines."
      }
    },
    {
      "_index": "starwars",
      "_type": "_doc",
      "_id": "3",
      "_score": 1.1077905,
      "_source": {
        "quote": "<b>No</b>. I am your father."
      }
    },
```


elastic

```
POST /starwars/_search
{
  "query": {
    "bool": {
      "must": {
        "match": {
          "quote": "father"
        }
      },
      "should": {
        "match_phrase": {
          "quote": "father father"
        }
      }
    }
  }
}
```

elastic

```
...
  "hits": {
    "total": 3,
    "max_score": 9.146545,
    "hits": [
      {
        "_index": "starwars",
        "_type": "_doc",
        "_id": "4",
        "_score": 9.146545,
        "_source": {
          "quote": "These droids are my father's father's machines."
        }
      },
      {
        "_index": "starwars",
        "_type": "_doc",
        "_id": "3",
        "_score": 1.0454913,
        "_source": {
          "quote": "<b>No</b>. I am your father."
        }
      },
      ...
```


elastic

Suggestion

Suggest a similar text

`_search` **end point**

`_suggest` **deprecated since 5.0**

```
POST /starwars/_search
{
  "query": {
    "match": {
      "quote": "drui"
    }
  },
  "suggest": {
    "my_suggestion" : {
      "text" : "drui",
      "term" : {
        "field" : "quote"
      }
    }
  }
}
```

elastic

```
...
  "hits": {
    "total": 0,
    "max_score": null,
    "hits": []
  },
  "suggest": {
    "my_suggestion": [
      {
        "text": "drui",
        "offset": 0,
        "length": 4,
        "options": [
          {
            "text": "droid",
            "score": 0.5,
            "freq": 1
          }
        ]
      }
    ]
  }
}
```

elastic

Multiple Suggesters

term phrase completion context

NGram

Partial matches

Edge Gram

```
GET /_analyze
{
  "char_filter": [
    "html_strip"
  ],
  "tokenizer": {
    "type": "ngram",
    "min_gram": "3",
    "max_gram": "3",
    "token_chars": [
      "letter"
    ]
  },
  "filter": [
    "lowercase"
  ],
  "text": "These are <em>not</em> the droids you are looking for."
}
```

```
{
  "tokens": [
    {
      "token": "the",
      "start_offset": 0,
      "end_offset": 3,
      "type": "word",
      "position": 0
    },
    {
      "token": "hes",
      "start_offset": 1,
      "end_offset": 4,
      "type": "word",
      "position": 1
    },
    {
      "token": "ese",
      "start_offset": 2,
      "end_offset": 5,
      "type": "word",
      "position": 2
    },
    {
      "token": "are",
      "start_offset": 6,
      "end_offset": 9,
      "type": "word",
      "position": 3
    },
    ...
```

```
GET /_analyze
{
  "char_filter": [
    "html_strip"
  ],
  "tokenizer": {
    "type": "edge_ngram",
    "min_gram": "1",
    "max_gram": "3",
    "token_chars": [
      "letter"
    ]
  },
  "filter": [
    "lowercase"
  ],
  "text": "These are <em>not</em> the droids you are looking for."
}
```

```
{
  "tokens": [
    {
      "token": "t",
      "start_offset": 0,
      "end_offset": 1,
      "type": "word",
      "position": 0
    },
    {
      "token": "th",
      "start_offset": 0,
      "end_offset": 2,
      "type": "word",
      "position": 1
    },
    {
      "token": "the",
      "start_offset": 0,
      "end_offset": 3,
      "type": "word",
      "position": 2
    },
    {
      "token": "a",
      "start_offset": 6,
      "end_offset": 7,
      "type": "word",
      "position": 3
    },
    {
      "token": "ar",
      "start_offset": 6,
      "end_offset": 8,
      "type": "word",
      "position": 4
    },
    ...
```

Combining Analyzers

Reindex

Store multiple times

Tune BM25

Combine scores

BM25 Revisited

$$\sum_{i}^{n} IDF(q_i) \frac{f(q_i, D) * (k1 + 1)}{f(q_i, D) + k1 * (1 - b + b * \frac{fieldLen}{avgFieldLen})}$$

https://www.elastic.co/blog/practical-bm25-part-2-the-bm25-algorithm-and-its-variables

elastic

b field length amplification

Default 0.75

k1 term frequency saturation

Default 1.2

```
PUT /starwars_v42
{
  "settings": {
    "number_of_shards": 1,
    "index": {
      "similarity": {
        "default": {
          "type": "BM25",
          "b": 0,
          "k1": 0
        }
      }
    },
```

elastic

```
"analysis": {
  "filter": {
    "my_synonym_filter": {
      "type": "synonym",
      "synonyms": [
        "father,dad",
        "droid => droid,machine"
      ]
    },
    "my_ngram_filter": {
      "type": "ngram",
      "min_gram": "3",
      "max_gram": "3",
      "token_chars": [
        "letter"
      ]
    }
  },
```

```
"analyzer": {
  "my_lowercase_analyzer": {
    "char_filter": [
      "html_strip"
    ],
    "tokenizer": "whitespace",
    "filter": [
      "lowercase"
    ]
  },
  "my_full_analyzer": {
    "char_filter": [
      "html_strip"
    ],
    "tokenizer": "standard",
    "filter": [
      "lowercase",
      "stop",
      "snowball",
      "my_synonym_filter"
    ]
  },
```

```
        "my_ngram_analyzer": {
          "char_filter": [
            "html_strip"
          ],
          "tokenizer": "whitespace",
          "filter": [
            "lowercase",
            "stop",
            "my_ngram_filter"
          ]
        }
      }
    }
},
```

elastic

```
"mappings": {
  "_doc": {
    "properties": {
      "quote": {
        "type": "text",
        "fields": {
          "lowercase": {
            "type": "text",
            "analyzer": "my_lowercase_analyzer"
          },
            "full": {
              "type": "text",
              "analyzer": "my_full_analyzer"
            },
            "ngram": {
              "type": "text",
              "analyzer": "my_ngram_analyzer"
} } } } } } }
```

elastic

```
POST /_reindex
{
  "source": {
    "index": "starwars"
  },
  "dest": {
    "index": "starwars_v42"
  }
}
```

Aliases

Atomic `remove` **and** `add`

Point to multiple indices (read-only)

```
PUT _alias
{
  "actions": [
    {
      "add": {
        "index": "starwars_v42",
        "alias": "starwars_extended"
      }
    }
  ]
}
```

```
POST /starwars/_search
{
  "query": {
    "match": {
      "quote": "droid"
    }
  }
}
```

```
"hits": [
  {
    "_index": "starwars",
    "_type": "_doc",
    "_id": "4",
    "_score": 1.1533037,
    "_source": {
      "quote": "These droids are my father's father's machines."
    }
  },
  {
    "_index": "starwars",
    "_type": "_doc",
    "_id": "1",
    "_score": 1.1295731,
    "_source": {
      "quote": "These are <em>not</em> the droids you are looking for."
    }
  }
]
```

```
POST /starwars_extended/_search
{
  "query": {
    "match": {
      "quote.full": "droid"
    }
  }
}
```

```
"hits": [
  {
    "_index": "starwars_v42",
    "_type": "_doc",
    "_id": "1",
    "_score": 0.6931472,
    "_source": {
      "quote": "These are <em>not</em> the droids you are looking for."
    }
  },
  {
    "_index": "starwars_v42",
    "_type": "_doc",
    "_id": "4",
    "_score": 0.6931472,
    "_source": {
      "quote": "These droids are my father's father's machines."
    }
  }
]
```

There are no "best" b and k1 values

```
POST /starwars_extended/_search?explain=true
{
  "query": {
    "multi_match": {
      "query": "obiwan",
      "fields": [
        "quote",
        "quote.lowercase",
        "quote.full",
        "quote.ngram"
      ],
      "type": "most_fields"
    }
  }
}
```

elastic

```
...
"hits": {
  "total": 1,
  "max_score": 0.4912064,
  "hits": [
    {
      "_shard": "[starwars_v42][2]",
      "_node": "BCDwzJ4WSw2dyoGLTzwlqw",
      "_index": "starwars_v42",
      "_type": "_doc",
      "_id": "2",
      "_score": 0.4912064,
      "_source": {
        "quote": "Obi-Wan never told you what happened to your father."
      },
      ...
```


elastic

Whitespace Tokenizer

```
"weight(
  Synonym(quote.ngram:biw quote.ngram:iwa quote.ngram:obi quote.ngram:wan)
in 0) [PerFieldSimilarity], result of:"
```

```
POST /starwars_extended/_search
{
  "query": {
    "multi_match": {
      "query": "you",
      "fields": [
        "quote",
        "quote.lowercase^5",
        "quote.full",
        "quote.ngram"
      ],
      "type": "best_fields"
    }
  }
}
```

elastic

```
"hits": [
  {
    "_index": "starwars_v42",
    "_type": "_doc",
    "_id": "1",
    "_score": 3.465736,
    "_source": {
      "quote": "These are <em>not</em> the droids you are looking for."
    }
  },
  {
    "_index": "starwars_v42",
    "_type": "_doc",
    "_id": "2",
    "_score": 3.465736,
    "_source": {
      "quote": "Obi-Wan never told you what happened to your father."
    }
  },
  {
    "_index": "starwars_v42",
    "_type": "_doc",
    "_id": "3",
    "_score": 0.35667494,
    "_source": {
      "quote": "<b>No</b>. I am your father."
    }
  }
]
```


elastic

Multi Match Type

`best_fields` **Score of the best field (default)**

`cross_fields` **All terms in at least one field**

`most_fields` **Score sum of all fields**

`phrase`

Different Analyzers for Indexing and Searching

Per query

In the mapping

```
POST /starwars_extended/_search
{
  "query": {
    "match": {
      "quote.ngram": {
        "query": "the",
        "analyzer": "standard"
      }
    }
  }
}
```

elastic

```
...
"hits": [
  {
    "_index": "starwars_extended",
    "_type": "_doc",
    "_id": "2",
    "_score": 0.38254172,
    "_source": {
      "quote": "Obi-Wan never told you what happened to your father."
    }
  },
  {
    "_index": "starwars_extended",
    "_type": "_doc",
    "_id": "3",
    "_score": 0.36165747,
    "_source": {
      "quote": "<b>No</b>. I am your father."
    }
  }
]
...
```


elastic

Edge Gram vs Trigram

Test a setting before adding a field

Shingle Token Filter

Shingles (token ngrams) from a token stream

```
POST /starwars_extended/_close
PUT /starwars_extended/_settings
{
  "index": {
    "similarity": {
      "default": {
        "type": "BM25",
        "b": null,
        "k1": null
      }
    }
  },
```

```
"analysis": {
  "filter": {
    "my_edgegram_filter": {
      "type": "edge_ngram",
      "min_gram": 3,
      "max_gram": 10
    },
    "my_shingle_filter": {
      "type": "shingle",
      "min_shingle_size": 2,
      "max_shingle_size": 2
    }
  },
```

```
"analyzer": {
  "my_edgegram_analyzer": {
    "char_filter": [
      "html_strip"
    ],
    "tokenizer": "standard",
    "filter": [
      "lowercase",
      "my_edgegram_filter"
    ]
  },
```

```
        "my_shingle_analyzer": {
          "type": "custom",
          "tokenizer": "standard",
          "filter": [
            "lowercase",
            "my_shingle_filter"
          ]
        }
      }
    }
}

POST /starwars_extended/_open
```

```
GET starwars_extended/_analyze
{
  "text": "Father",
  "analyzer": "my_edgegram_analyzer"
}
```

```
{
  "tokens": [
    {
      "token": "fat",
      "start_offset": 0,
      "end_offset": 6,
      "type": "<ALPHANUM>",
      "position": 0
    },
    {
      "token": "fath",
      "start_offset": 0,
      "end_offset": 6,
      "type": "<ALPHANUM>",
      "position": 0
    },
    {
      "token": "fathe",
      "start_offset": 0,
      "end_offset": 6,
      "type": "<ALPHANUM>",
      "position": 0
    },
    {
      "token": "father",
      "start_offset": 0,
      "end_offset": 6,
      "type": "<ALPHANUM>",
      "position": 0
    }
  ]
}
```


elastic

```
PUT /starwars_extended/_doc/_mapping
{
  "properties": {
    "quote": {
      "type": "text",
      "fields": {
        "edgegram": {
          "type": "text",
          "analyzer": "my_edgegram_analyzer",
          "search_analyzer": "standard"
        },
        "shingle": {
          "type": "text",
          "analyzer": "my_shingle_analyzer"
        }
      }
    }
  }
}
```

elastic

```
PUT /starwars_extended/_doc/5
{
  "quote": "I find your lack of faith disturbing."
}
PUT /starwars_extended/_doc/6
{
  "quote": "That... is your failure."
}
```

```
GET /starwars_extended/_doc/5/_termvectors
{
  "fields": [
    "quote.edgegram"
  ],
  "offsets": true,
  "payloads": true,
  "positions": true,
  "term_statistics": true,
  "field_statistics": true
}
```

```
{
  "_index": "starwars_v42",
  "_type": "_doc",
  "_id": "5",
  "_version": 1,
  "found": true,
  "took": 3,
  "term_vectors": {
    "quote.edgegram": {
      "field_statistics": {
        "sum_doc_freq": 26,
        "doc_count": 2,
        "sum_ttf": 26
      },
      "terms": {
        "dis": {
          "doc_freq": 1,
          "ttf": 1,
          "term_freq": 1,
          "tokens": [
            {
              "position": 6,
              "start_offset": 26,
              "end_offset": 36
            }
          ]
        },
        "dist": {
          "doc_freq": 1,
          "ttf": 1,
          ...
```


elastic

```
POST /starwars_extended/_search
{
  "query": {
    "match": {
      "quote": "fail"
    }
  }
}
```

```
POST /starwars_extended/_search
{
  "query": {
    "match": {
      "quote.lowercase": "fail"
    }
  }
}
```

```
POST /starwars_extended/_search
{
  "query": {
    "match": {
      "quote.full": "fail"
    }
  }
}
```

```
POST /starwars_extended/_search
{
  "query": {
    "match": {
      "quote.ngram": "fail"
    }
  }
}
```

```
"hits": [
  {
    "_index": "starwars_v42",
    "_type": "_doc",
    "_id": "6",
    "_score": 1.8400999,
    "_source": {
      "quote": "That... is your failure."
    }
  },
  {
    "_index": "starwars_v42",
    "_type": "_doc",
    "_id": "5",
    "_score": 1.442779,
    "_source": {
      "quote": "I find your lack of faith disturbing."
    }
  }
]
```


elastic

```
POST /starwars_extended/_search
{
  "query": {
    "match": {
      "quote.edgegram": "fail"
    }
  }
}
```

```
"hits": [
  {
    "_index": "starwars_v42",
    "_type": "_doc",
    "_id": "6",
    "_score": 1.0114291,
    "_source": {
      "quote": "That... is your failure."
    }
  }
]
```

Updating Missing Fields

Expensive

```
POST /starwars_extended/_update_by_query
{
  "query": {
    "bool": {
      "must_not": {
        "exists": {
          "field": "quote.edgegram"
        }
      }
    }
  }
}
```

Shingles: Context Should Matter

```
POST /starwars_extended/_search
{
  "query": {
    "bool": {
      "must": {
        "match": {
          "quote.lowercase": "these droids are"
        }
      }
    }
  }
}
```

```
"hits": [
  {
    "_index": "starwars_v42",
    "_type": "_doc",
    "_id": "1",
    "_score": 2.1837702,
    "_source": {
      "quote": "These are <em>not</em> the droids you are looking for."
    }
  },
  {
    "_index": "starwars_v42",
    "_type": "_doc",
    "_id": "4",
    "_score": 2.137744,
    "_source": {
      "quote": "These droids are my father's father's machines."
    }
  }
]
```

```
POST /starwars_extended/_search
{
  "query": {
    "bool": {
      "must": {
        "match": {
          "quote.shingle": "these droids are"
        }
      }
    }
  }
}
```

```
"hits": [
  {
    "_index": "starwars_v42",
    "_type": "_doc",
    "_id": "4",
    "_score": 3.1811738,
    "_source": {
      "quote": "These droids are my father's father's machines."
    }
  },
  {
    "_index": "starwars_v42",
    "_type": "_doc",
    "_id": "1",
    "_score": 2.6568544,
    "_source": {
      "quote": "These are <em>not</em> the droids you are looking for."
    }
  }
]
```

Decompounding

Commonly in German, Scandinavian languages, Finnish, Korean

```
PUT /decompound_en
{
  "settings": {
    "number_of_shards": 1,
    "analysis": {
      "filter": {
        "british_decompounder": {
          "type": "hyphenation_decompounder",
          "hyphenation_patterns_path": "hyph/en_GB.xml",
          "word_list": [ "death", "star" ]
        }
      },
      "analyzer": {
        "british_decompound": {
          "type": "custom",
          "tokenizer": "standard",
          "filter": [
            "lowercase",
            "british_decompounder"
          ]
        }
      }
    }
  }
}
```

elastic

```
GET /decompound_en/_analyze
{
  "analyzer" : "british_decompound",
  "text" : "deathstar"
}
```

```
{
  "tokens": [
    {
      "token": "deathstar",
      "start_offset": 0,
      "end_offset": 9,
      "type": "<ALPHANUM>",
      "position": 0
    },
    {
      "token": "death",
      "start_offset": 0,
      "end_offset": 9,
      "type": "<ALPHANUM>",
      "position": 0
    },
    {
      "token": "star",
      "start_offset": 0,
      "end_offset": 9,
      "type": "<ALPHANUM>",
      "position": 0
    }
  ]
}
```


elastic

German Dictionaly (LGPL)

https://github.com/uschindler/german-decompounder

```
PUT /decompound_de
{
  "settings": {
    "number_of_shards": 1,
    "analysis": {
      "filter": {
        "german_decompounder": {
          "type": "hyphenation_decompounder",
          "word_list_path": "dictionary-de.txt",
          "hyphenation_patterns_path": "hyph/de_DR.xml",
          "only_longest_match": true,
          "min_subword_size": 4
        },
        "german_stemmer": {
          "type": "stemmer",
          "language": "light_german"
        }
      },
      "analyzer": {
        "german_decompound": {
          "type": "custom",
          "tokenizer": "standard",
          "filter": [
            "lowercase",
            "german_decompounder",
            "german_normalization",
            "german_stemmer"
          ]
        }
      }
    }
  }
}
```

elastic

```
GET /decompound_de/_analyze
{
  "analyzer" : "german_decompound",
  "text" : "Todesstern"
}
```

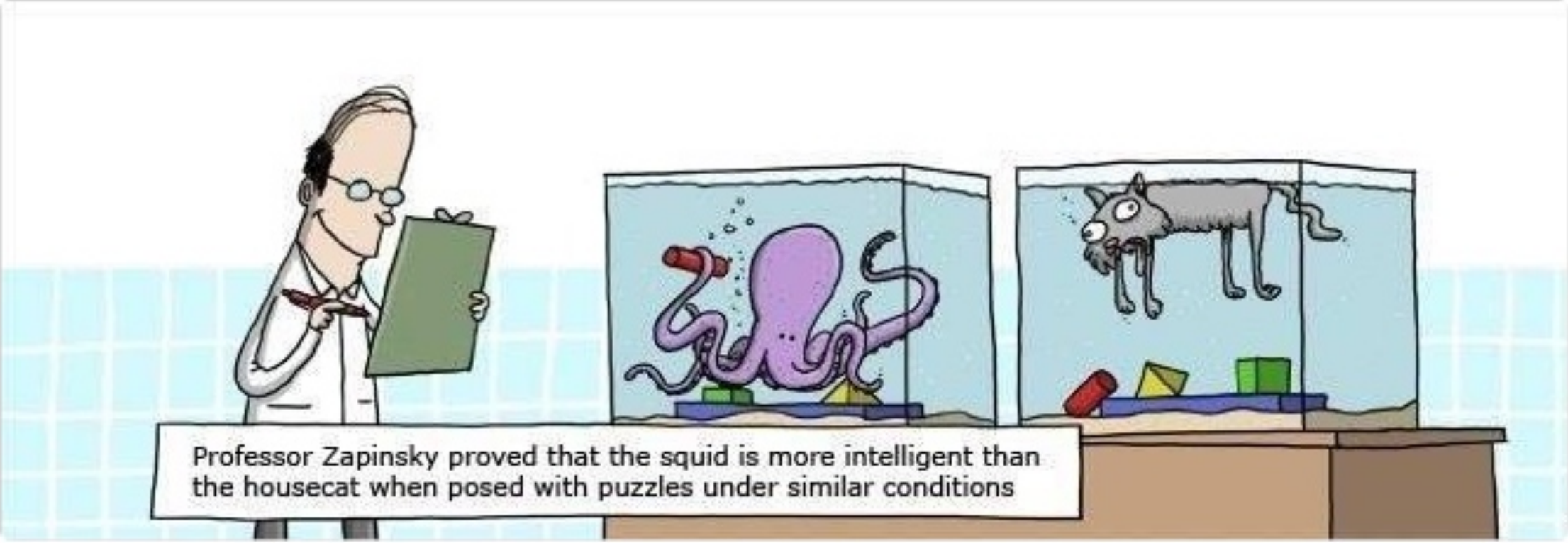
```
{
  "tokens": [
    {
      "token": "todesst",
      "start_offset": 0,
      "end_offset": 10,
      "type": "<ALPHANUM>",
      "position": 0
    },
    {
      "token": "tod",
      "start_offset": 0,
      "end_offset": 10,
      "type": "<ALPHANUM>",
      "position": 0
    },
    {
      "token": "stern",
      "start_offset": 0,
      "end_offset": 10,
      "type": "<ALPHANUM>",
      "position": 0
    }
  ]
}
```


elastic

Without Word Lists

https://github.com/jprante/elasticsearch-analysis-decompound

Performance


Professor Zapinsky proved that the squid is more intelligent than the housecat when posed with puzzles under similar conditions

elastic

ONE MILLION
DOCUMENTS
memegenerator.net

Conclusion

Indexing

Formatting

Tokenize

Lowercase, Stop Words, Stemming

Synonyms

elastic

Scoring

Term Frequency

Inverse Document Frequency

Field-Length Norm

Vector Space Model

Advanced Queries

Highlighting

Suggestions

NGrams, Edge Grams

Multiple Analyzers

Advanced Queries

Reindex & Alias

Update by Query

Shingles

Decompound

elastic

Monitor Your Apps

with the

elastic stack

elastic

Honest Status Page
@honest_update
Follow
We replaced our monolith with micro services so that every outage could be more like a murder mystery.
RETWEETS
2,882
LIKES
2,256
4:10 PM - 7 Oct 2015
18
2.9K
2.3K

THIS IS FINE.

TH-
THIS IS NOT FINE!!
OH MY GOD EVERY THINGS ON FIRE
WHAT THE HELL IS MY PROBLEM

elastic

ALL THE THINGS!

Disclaimer

I build highly monitored Hello World apps

Agenda

Monitor Java (preconfigured)

Some Security

Monitor PHP (configure yourself)

Code

https://github.com/xeraa/microservice-monitoring

Cloud
elastic

Amazon Lightsail

TERRAFORM

elastic

elastic cloud

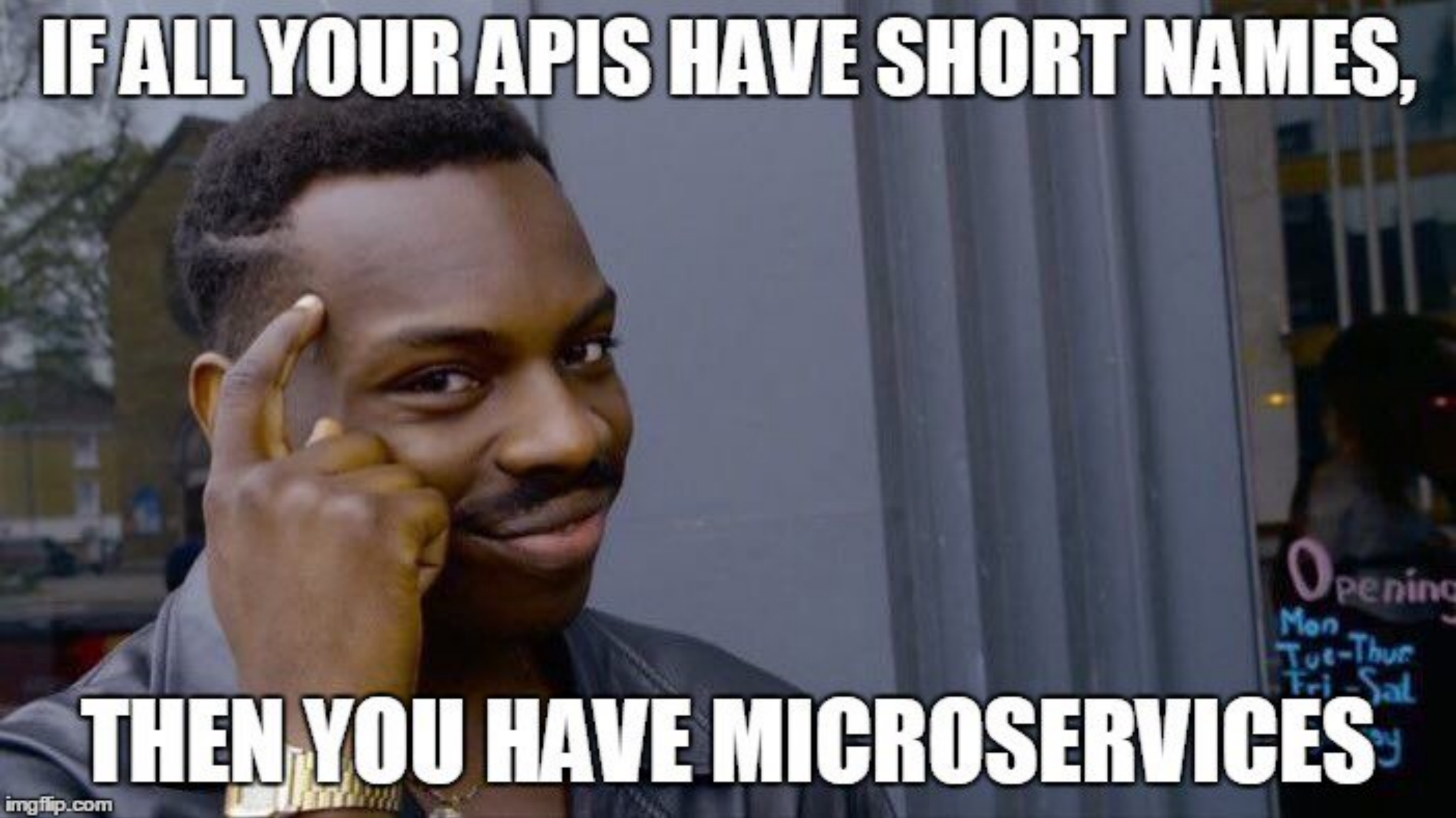
elastic

Java Application

elastic

Simple

No discovery, load-balancing,...


IF ALL YOUR APIS HAVE SHORT NAMES,
THEN YOU HAVE MICROSERVICES
imgflip.com

Monitor Java
elastic

Kibana Monitoring

Overview of the Elastic Stack components

elastic

Metricbeat System

[Metricbeat System] Overview and [Metricbeat System] Host overview dashboards

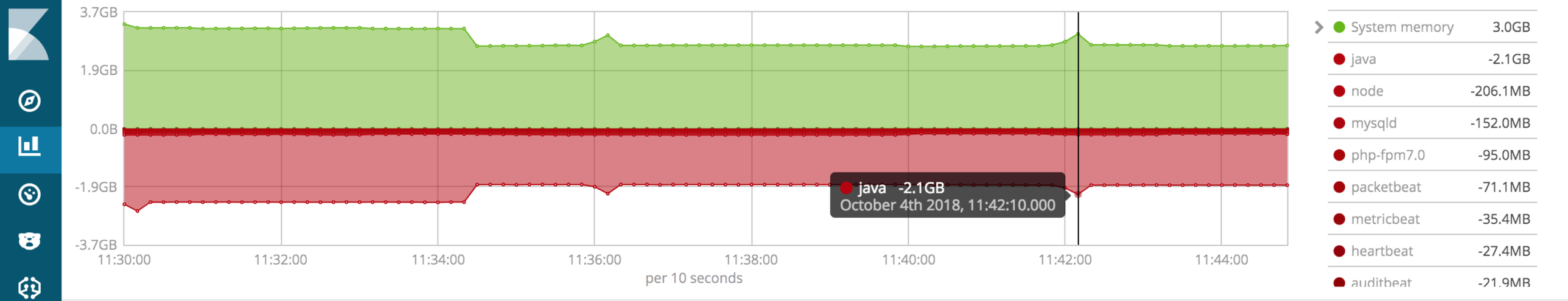
See the memory spike every 5min

Time Series Visual Builder

Sum of
system.memory.actual.used.bytes

Sum of system.process.memory.rss.bytes grouped by the term system.process.name and moved to the negative y-axis with a Math step

elastic



Auto Apply ▶ Apply Changes The changes will be automatically applied.

Data Panel Options Annotations

System memory

Metrics Options

Aggregation: Sum

Field: system.memory.actual.used.bytes

Group By: Everything

Process memory

Metrics Options

Packetbeat

Call /, /good, /bad, and /foobar

[Packetbeat] Overview, [Packetbeat] Flows, [Packetbeat] HTTP, and [Packetbeat] DNS Tunneling dashboards

elastic

Packetbeat

Raw events in Discover

Process enrichment for nginx, Java, and the APM server

Filebeat Modules

[Filebeat Nginx] Access and error logs, [Filebeat System] Syslog dashboard, and [Osquery Result] Compliance pack dashboards

Filebeat

Raw events in Discover

/good: MDC logging under json.name and the context view for one log message

meta.* and host.* information

elastic

Filebeat

/bad and /null: Stacktraces by filtering down on application:java and json.severity:ERROR

Visualize json.stack_hash

Visualize / New Visualization (unsaved)

Save Share Refresh Reporting Auto-refresh Last 7 days

Search... (e.g. status:200 AND extension:PHP) Uses lucene query syntax

Add a filter +

filebeat-*

Data Metrics & Axes Panel Settings

Metrics

Y-Axis Count

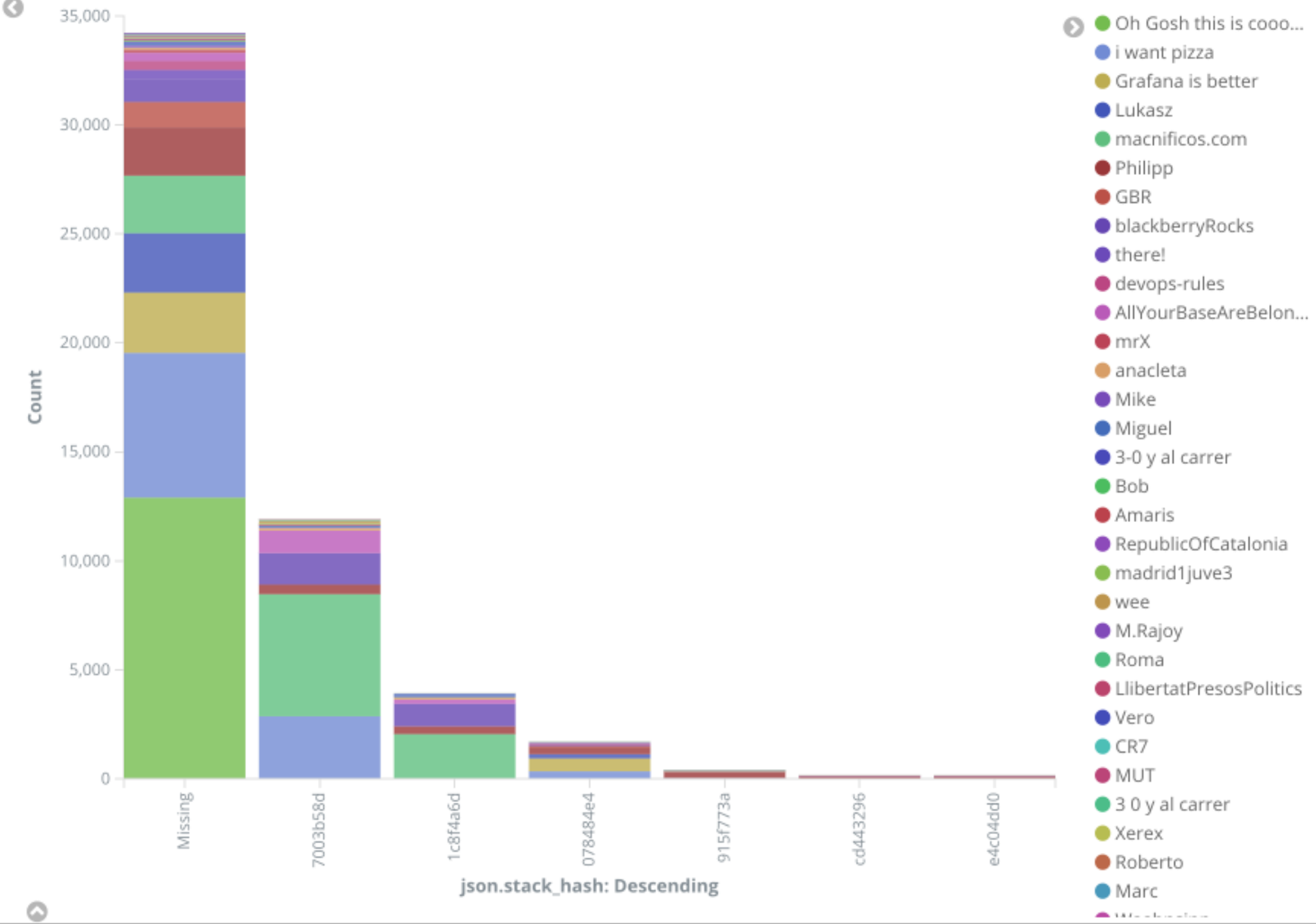
Add metrics

Buckets

X-A... json.stack_hash: Descending

Split Se... json.name: Descending

Add sub-buckets



Heartbeat

Heartbeat HTTP monitoring dashboard

Stop and start the frontend application while auto refreshing

Metricbeat nginx

[Metricbeat Nginx] Overview dashboard

elastic

Metricbeat HTTP

/health and /metrics endpoints

Collected information in Discover

Metricbeat JMX

Same data

Visualize the heap usage: jolokia.metrics.memory.heap_usage.used divided by the max of jolokia.metrics.memory.heap_usage.max

Annotations

Add changes from the events index

Heap usage

Metrics Options

Aggregation: Average
Field: jolokia.metrics.memory.heap_usage.used

Aggregation: Max
Field: jolokia.metrics.memory.heap_usage.max

Aggregation: Math

Variables

used: Average of jolokia.metrics.memory.heap_usage.used

max: Max of jolokia.metrics.memory.heap_usage.max

Expression

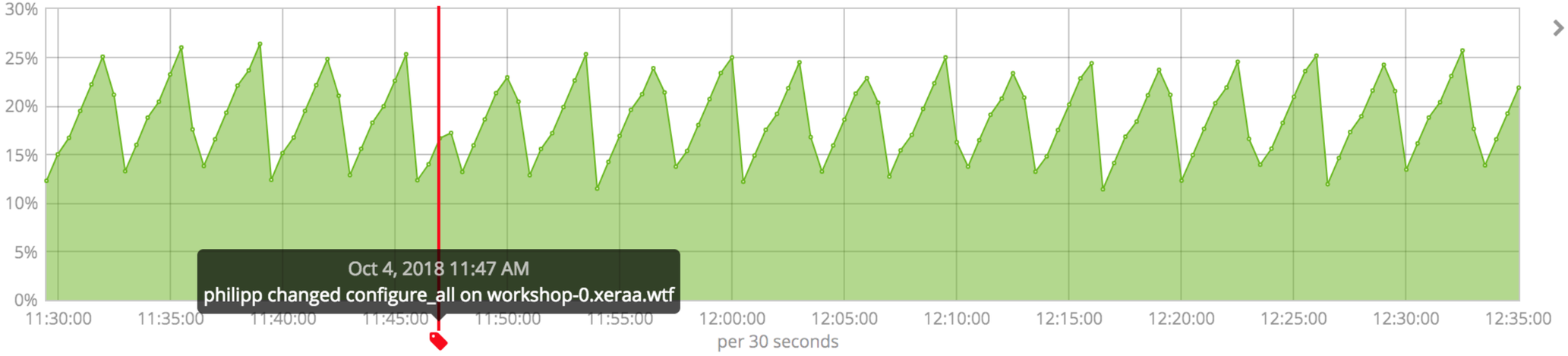
```
params.used/params.max
```

This field uses basic math expressions (see TinyMath) - Variables are keys on the params object, i.e. params.<name> To access all the data useparams._all.<name>.values for an array of the values and params._all.<name>.timestampsfor an array of the timestamps. params._timestampis available for the current bucket's timestamp,params._index is available for the current bucket's index, and params._intervals available for the interval in milliseconds.

Group By: Everything

This visualization is marked as experimental. Have feedback? Please create an issue in GitHub.

Time Series | Metric | Top N | Gauge | Markdown | Table



Auto Apply | Apply Changes | The changes will be automatically applied.

Data | Panel Options | Annotations

Data Sources

Index Pattern (required): events

Time Field (required): @timestamp

Query String:

Ignore Global Filters: Yes / No

Ignore Panel Filters: Yes / No

Icon (required): Tag

Fields (required - comma separated paths): application,user,host

Row Template (required - eg.{{field}}): {{user}} changed {{application}} on {{host}}

Some Security

elastic

Filebeat Modules

[Filebeat Auditd] Audit Events, [Filebeat System] New users and groups, and [Filebeat System] Sudo commands dashboards

elastic

https://github.com/linux-audit

"auditd is the userspace component to the Linux Auditing System. It's responsible for writing audit records to the disk. Viewing the logs is done with the ausearch or aureport utilities."

Auditd Monitors

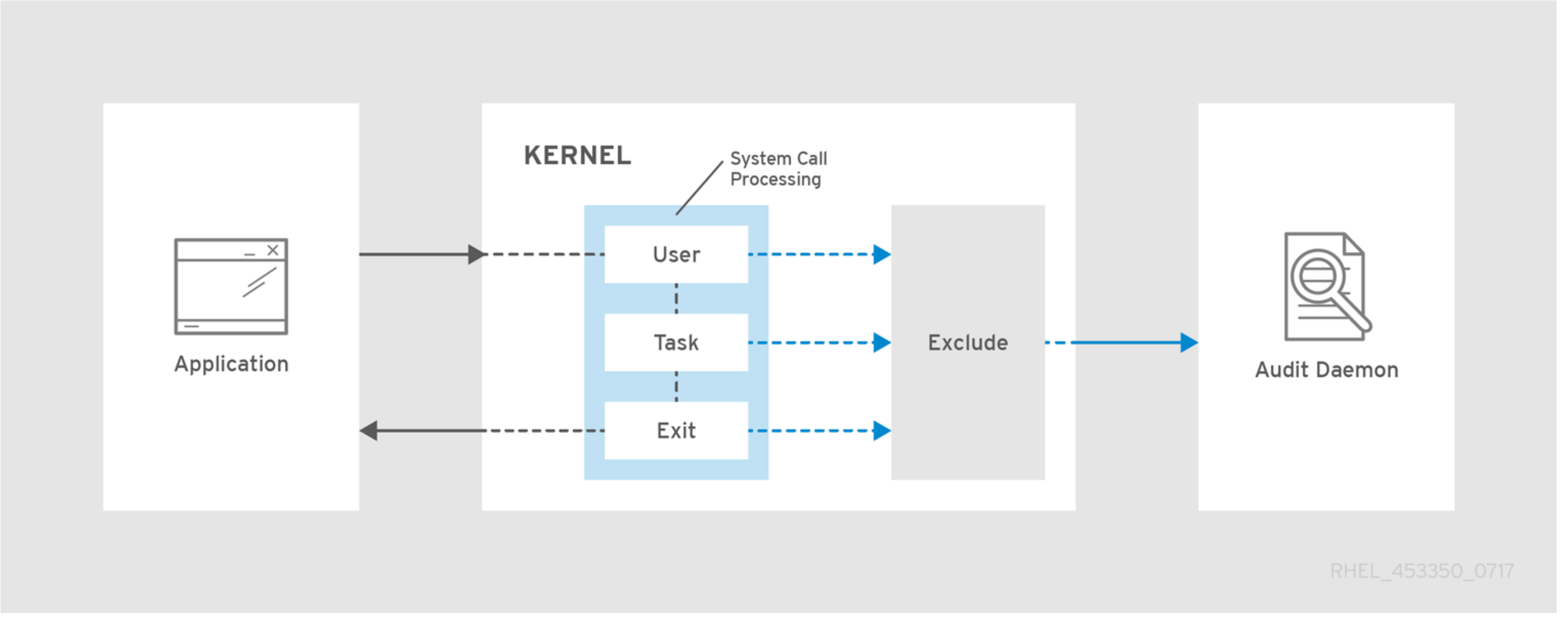
File and network access

System calls

Commands run by a user

Security events

elastic



https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/security_guide/chap-system_auditing

Understanding Logs

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/security_guide/sec-understanding_audit_log_files

elastic

Auditbeat

[Auditbeat Auditd] Overview dashboard

elastic

Fail SSH

ssh elastic-user@xeraa.wtf
with a bad password

[Filebeat System] SSH login attempts
dashboard

elastic

Success

ssh elastic-user@xeraa.wtf with a good password

Run service nginx restart and pick the elastic-admin user

Audit Event

[Auditbeat Auditd] Executions dashboard filter elastic-user

elastic

Audit Event

cat /etc/passwd

Filter for tags is developers-passwd-read in Discover

Power Abuse

ssh elastic-admin@xeraa.wtf

sudo cat /home/elastic-user/secret.txt

Tag power-abuse in Discover

File Integrity

Change something in
/var/www/html/index.html

[Auditbeat File Integrity] Overview
dashboard

elastic

Monitor PHP

elastic

Heartbeat

Add HTTP port 88

Packetbeat

Add HTTP on port 88

Metricbeat

php-fpm

```
- module: php_fpm
  metricsets: ["pool"]
  period: 10s
  status_path: "/status"
  hosts: ["http://localhost"]
```

Filebeat

Collect /var/www/html/silverstripe/logs/*.json

MORE
elastic

Alerting[a]

Name

Heap

Select an Index

metricbeat-*

Broad searches can be done by adding * to your query

Select a time field

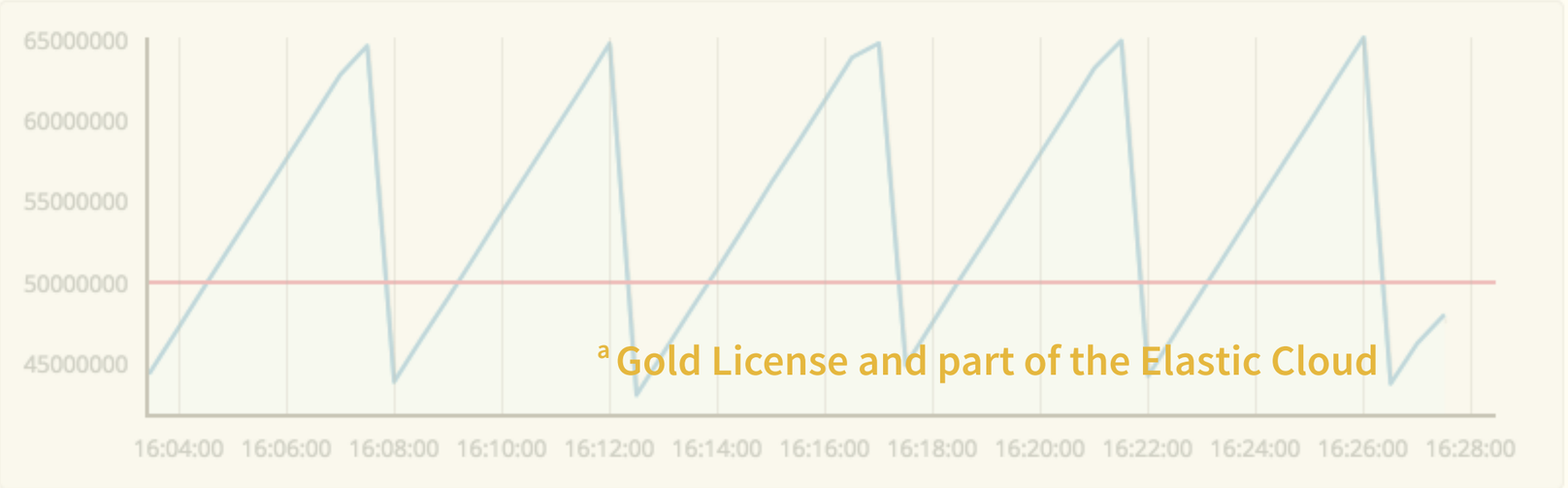
@timestamp

Run this wat

Matching the following condition

WHEN max() OF jolokia.metrics.memory.heap_usage.used GROUPED OVER top 3 beat.name IS ABOVE 50000000 FOR THE LAST 5 minutes

beat.name (1 of 3): frontend.xeraa.wtf



[a] Gold License and part of the Elastic Cloud

Name

Heap

Select an Index

metricbeat-*

Broad searches can be done by adding * to your query

Select a time field

@timestamp

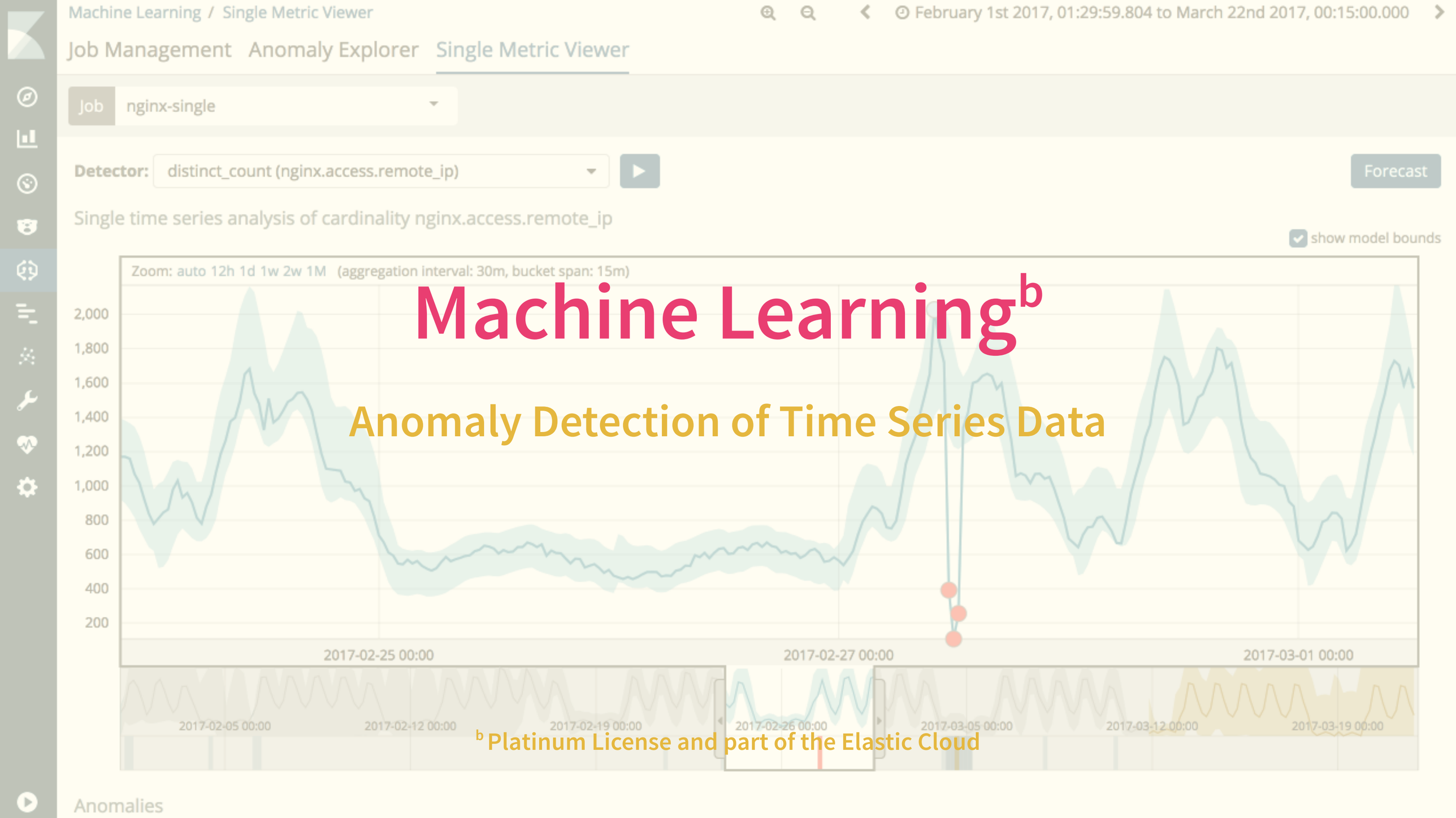
Run this wat

1

Matching the following condition

WHEN max() OF jolokia.metrics.memory.heap_usage.used GROUPED OVER top 5 'beat.name' IS ABOVE 50000000 FOR THE LAST 5 minutes

beat.name (1 of 3): frontend.xeraa.wtf



Machine Learning[b]

Anomaly Detection of Time Series Data

[b] Platinum License and part of the Elastic Cloud

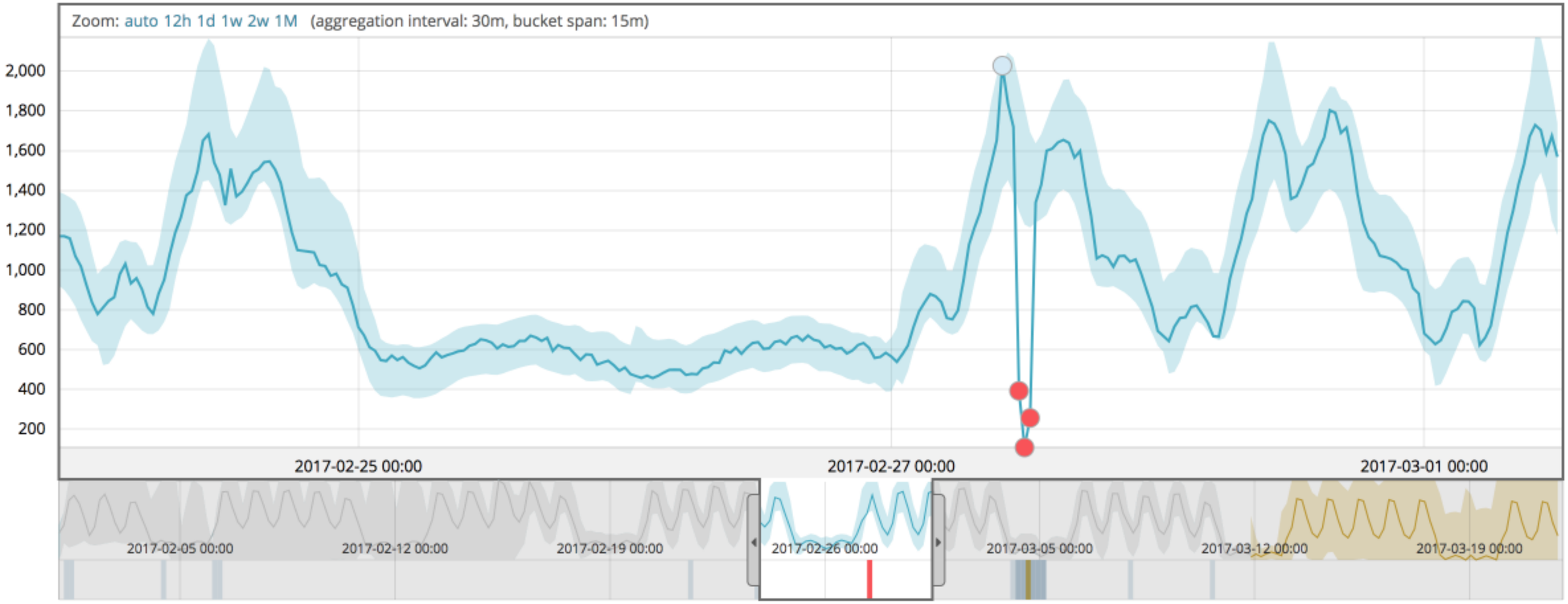
Job nginx-single

Detector: distinct_count (nginx.access.remote_ip)

Forecast

Single time series analysis of cardinality nginx.access.remote_ip

show model bounds



Anomalies

Conclusion

elastic

System metrics & network

Filebeat modules & Auditbeat

Application logs

Uptime

Application metrics

Request tracing

Code

https://github.com/xeraa/microservice-monitoring

Questions?

Philipp Krenn @xeraa

PS: Sticker

elastic