# Elasticsearch - Speed is key

Alexander Reelsen
@spinscale

elastic

CRAFT

# Agenda

- Introduction
  - Elasticsearch

- Speed by Example
  - Search
  - Aggregations
  - Operating System
  - Distributed aspects

# About

- Me
  - joined in march 2013
  - working on Elasticsearch & Shield
  - Interested in all things scale & search
- Elastic
  - Founded in 2012
  - Behind: Elasticsearch, Logstash, Kibana, Marvel, Shield, ES for Hadoop, Elasticsearch clients
  - Support subscriptions
  - Public & private trainings

# About

- Me
  - joined in march 2013
  - working on Elasticsearch & Shield
  - Interested in all things scale & search

- Elastic
  - Founded in 2012
  - Behind: Elasticsearch, Logstash, Kibana, Marvel, Shield, ES for Hadoop, Elasticsearch clients
  - Support subscriptions
  - Public & private trainings

Elasticsearch - High level overview

# Elasticsearch is...

an open source, distributed, scalable,

highly available, document-oriented, RESTful

full text search engine

with real-time search and analytics capabilities

# Elasticsearch is...

an **open source**, distributed, scalable, highly available, document-oriented,

RESTful, full text search engine with real-time search and analytics capabilities

Apache 2.0 License

https://www.apache.org/licenses/LICENSE-2.0

# Elasticsearch is...

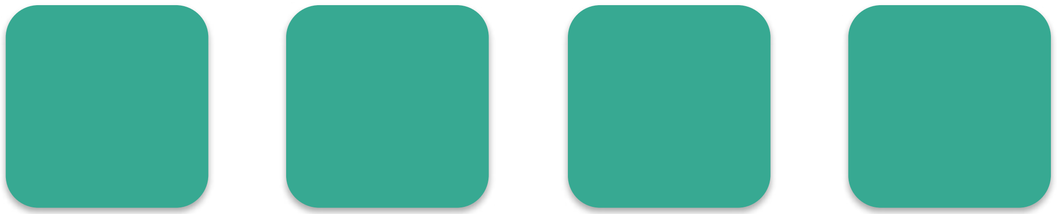
an open source, **distributed, scalable**, highly available, document-oriented,

RESTful, full text search engine with real-time search and analytics capabilities

# Elasticsearch is...

an open source, **distributed, scalable**, highly available, document-oriented,

RESTful, full text search engine with real-time search and analytics capabilities



# Elasticsearch is...

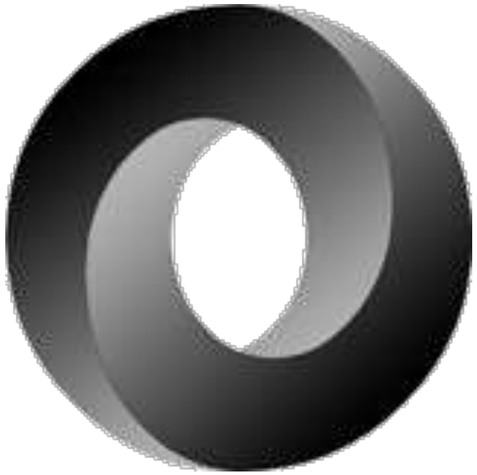
an open source, distributed, scalable, **highly available**, document-oriented,

RESTful, full text search engine with real-time search and analytics capabilities

# Elasticsearch is...

an open source, distributed, scalable, highly available, **document-oriented**,

RESTful, full text search engine with real-time search and analytics capabilities



Source: http://json.org/

```
{
  "name" : "Craft"
  "geo" : {
    "city" : "Budapest",
    "lat" : 47.49, "lon" : 19.04
  }
}
```

# Elasticsearch is...

an open source, distributed, scalable, highly available, document-oriented,

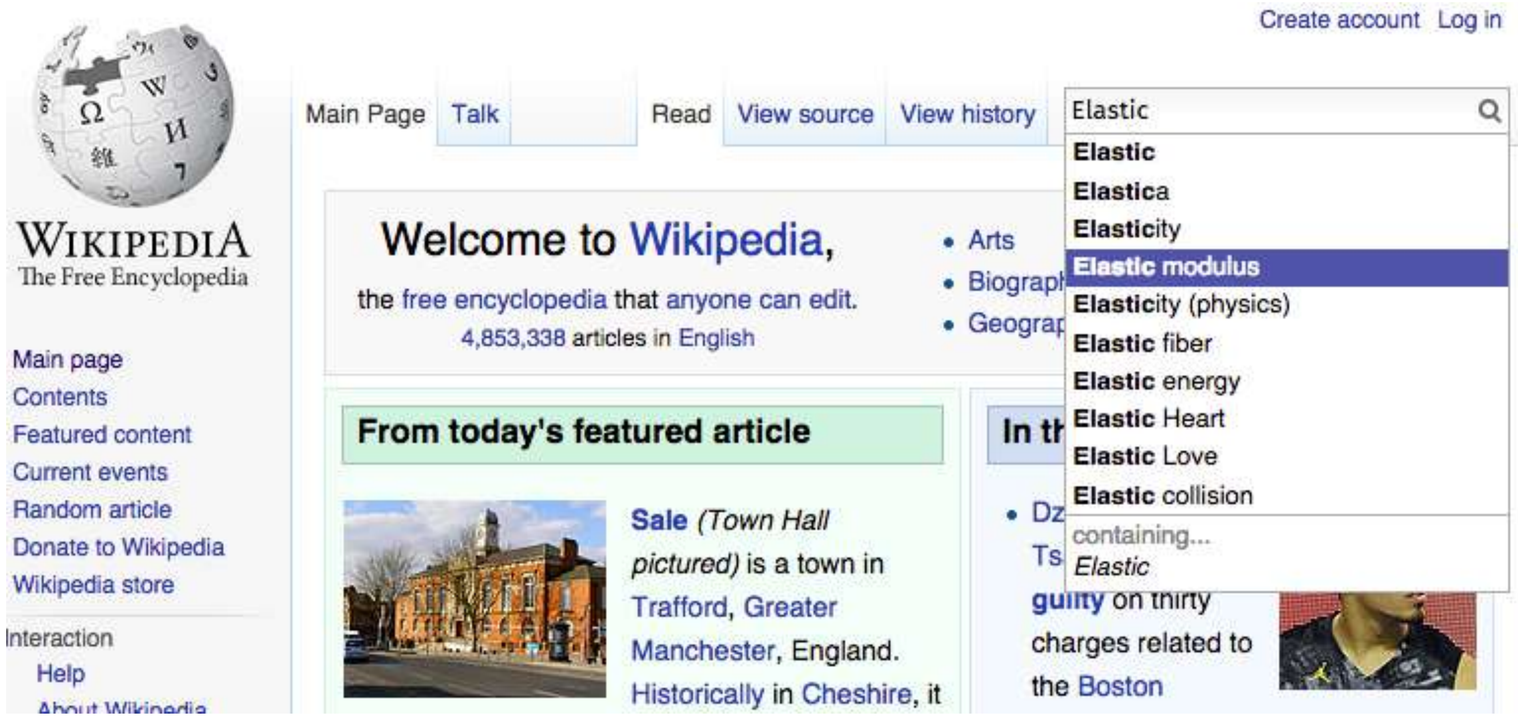
**RESTful**, full text search engine with real-time search and analytics capabilities


Source: https://httpwg.github.io/asset/http.svg

# Elasticsearch is...

an open source, distributed, scalable, highly available, document-oriented,

RESTful, **full text search engine** with real-time search and analytics capabilities



# Elasticsearch is...

an open source, distributed, scalable, highly available, document-oriented,

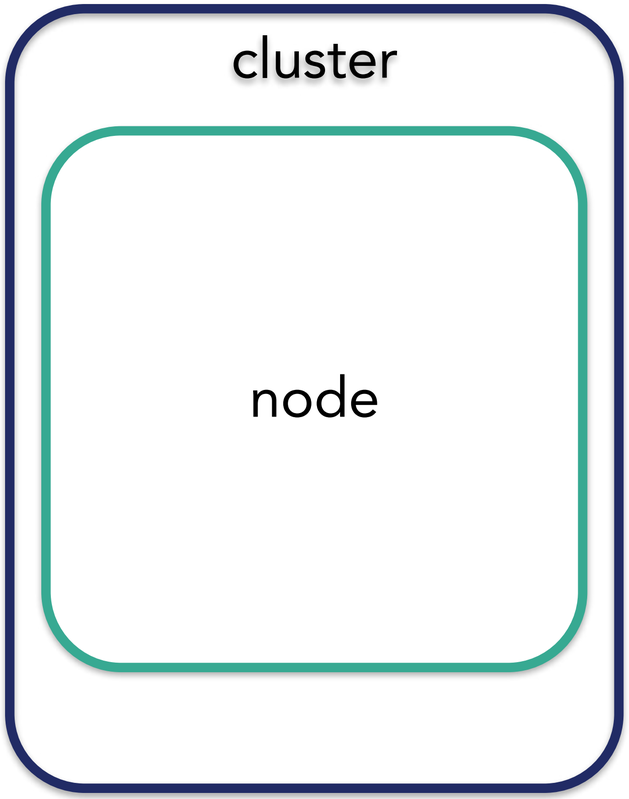
RESTful, full text search engine with **real-time search and analytics capabilities**

# Getting up and running... is easy

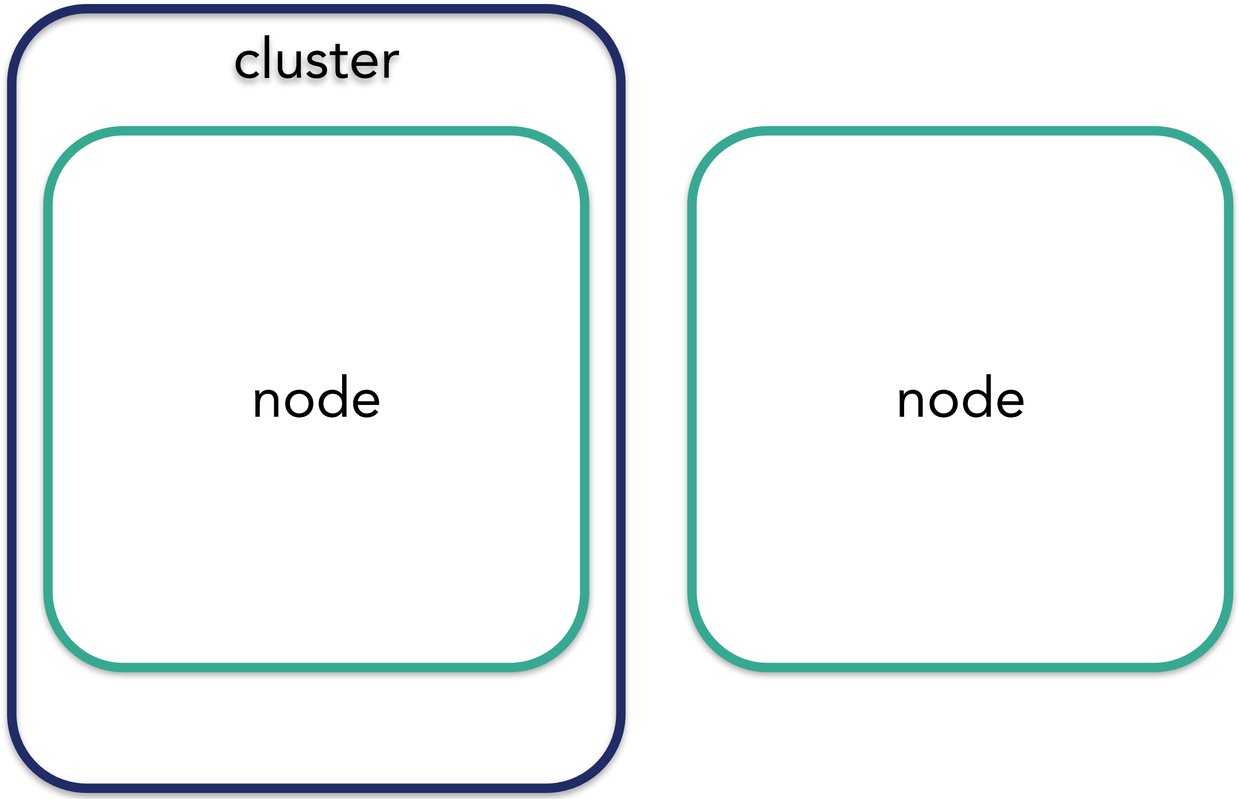
```
# wget https://download.elastic.co/elasticsearch/
elasticsearch/elasticsearch-1.5.1.zip

# unzip elasticsearch-1.5.1.zip
# cd elasticsearch-1.5.1

# ./bin/elasticsearch

# curl http://localhost:9200
```

# Scaling

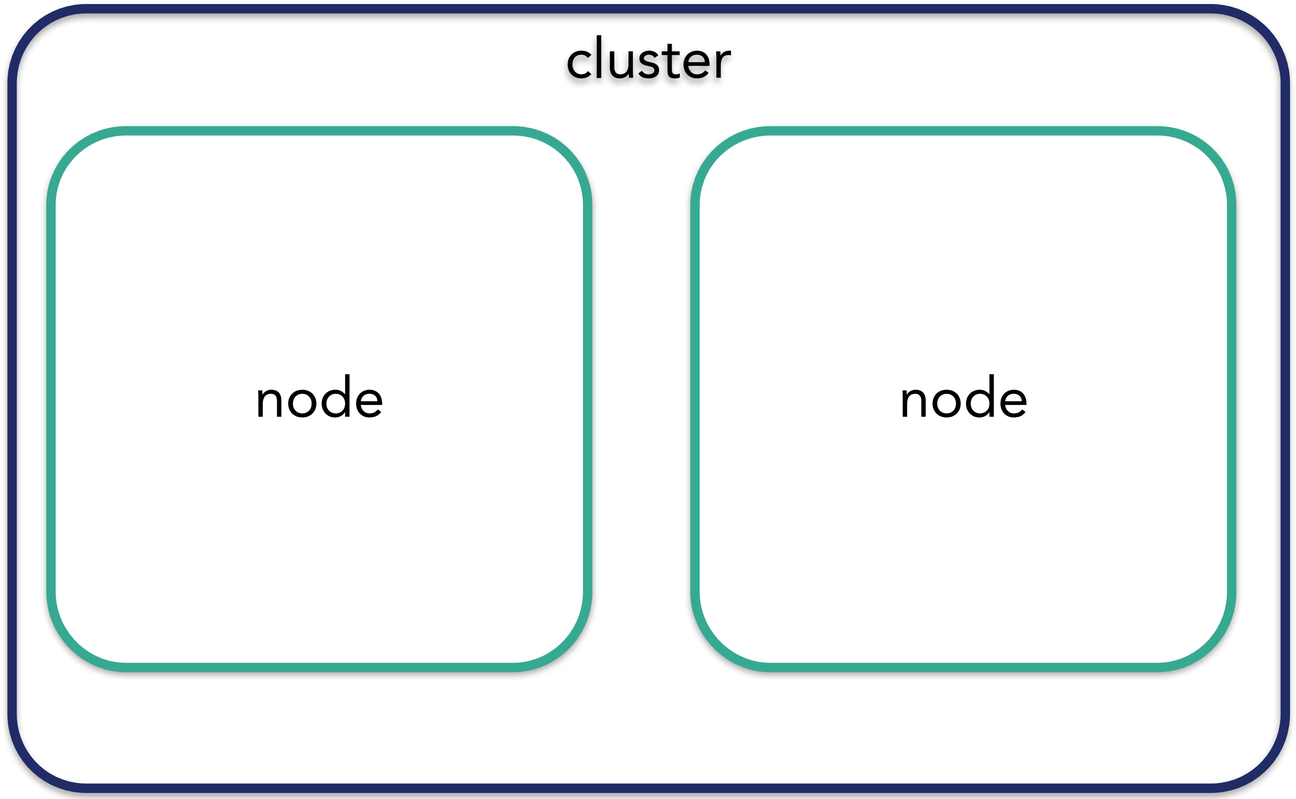
# Cluster: A collection of nodes



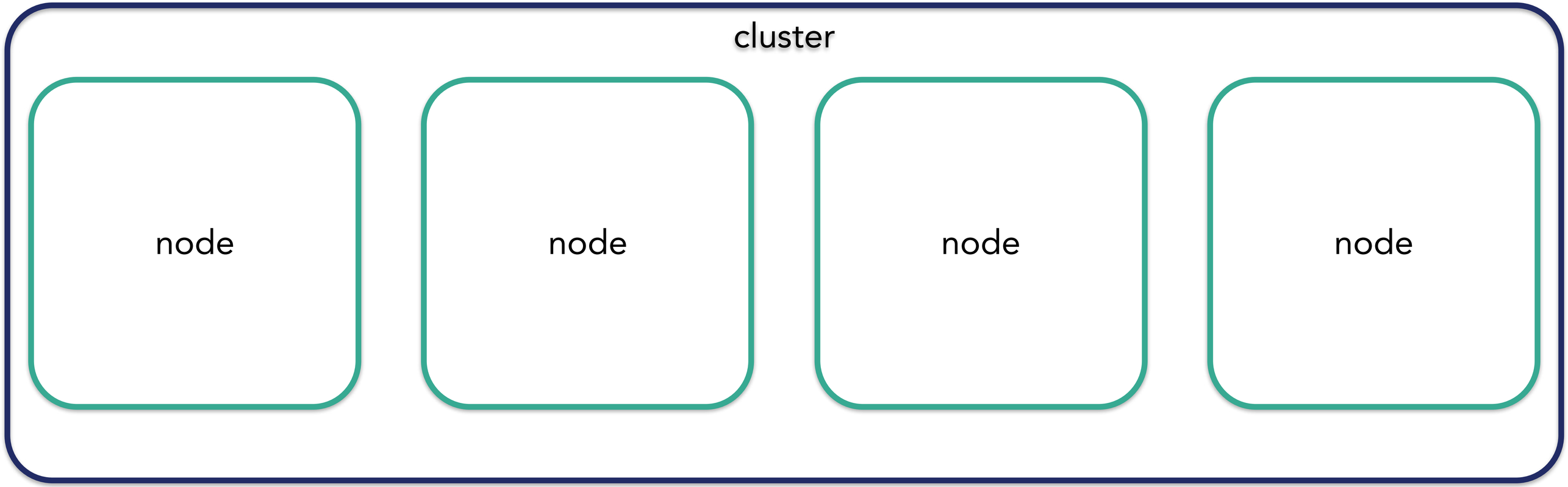
# Cluster: A collection of nodes



# Cluster: A collection of nodes

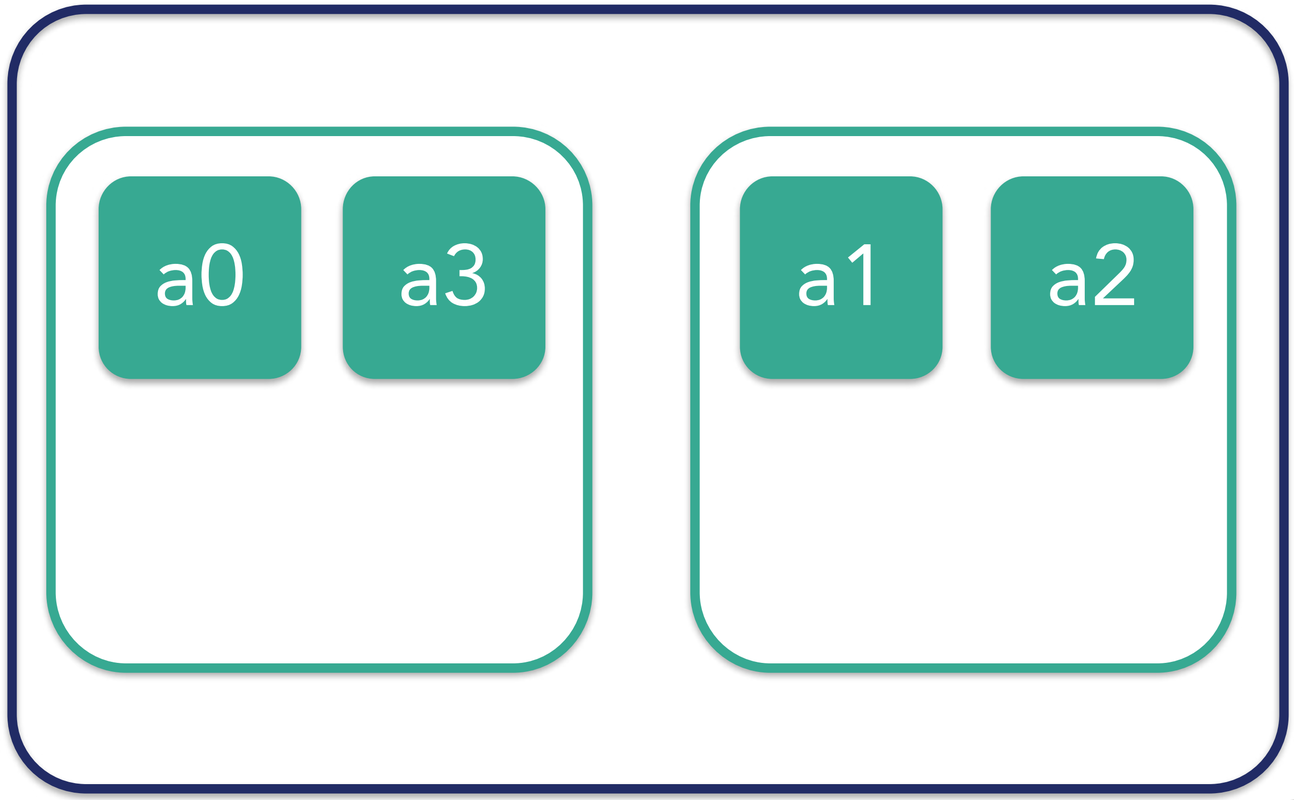


# Cluster: A collection of nodes



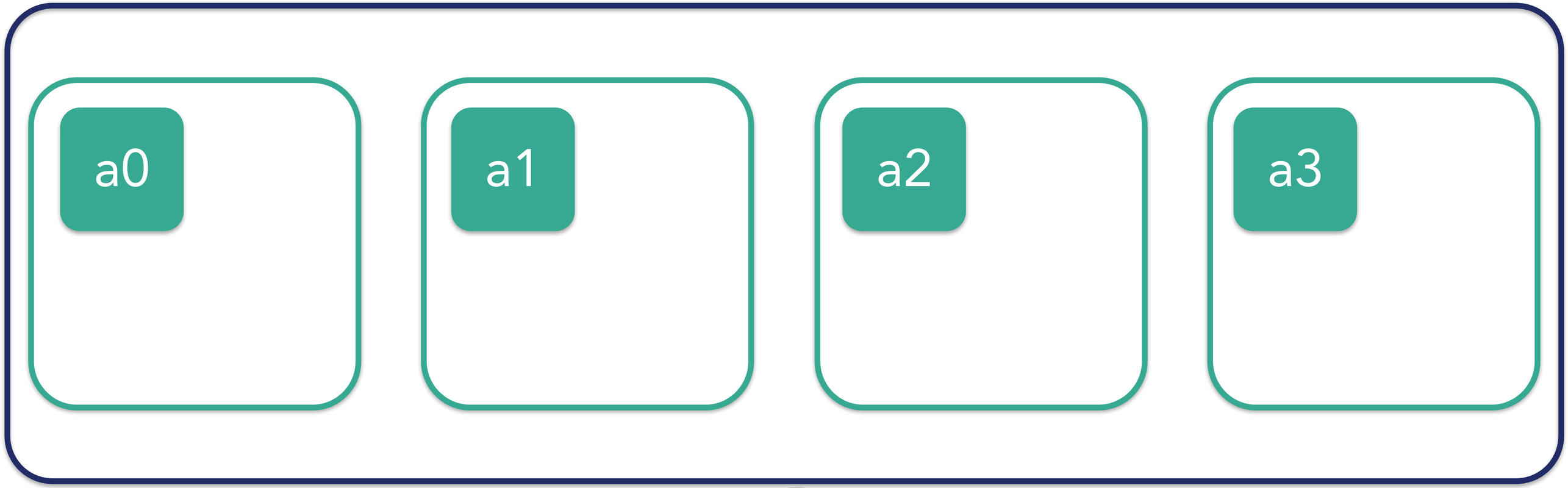
# Shards: Unit of scale

```
# curl -X PUT http://localhost:9200/a -d '
{ "index.number_of_shards" : 4 }'
```



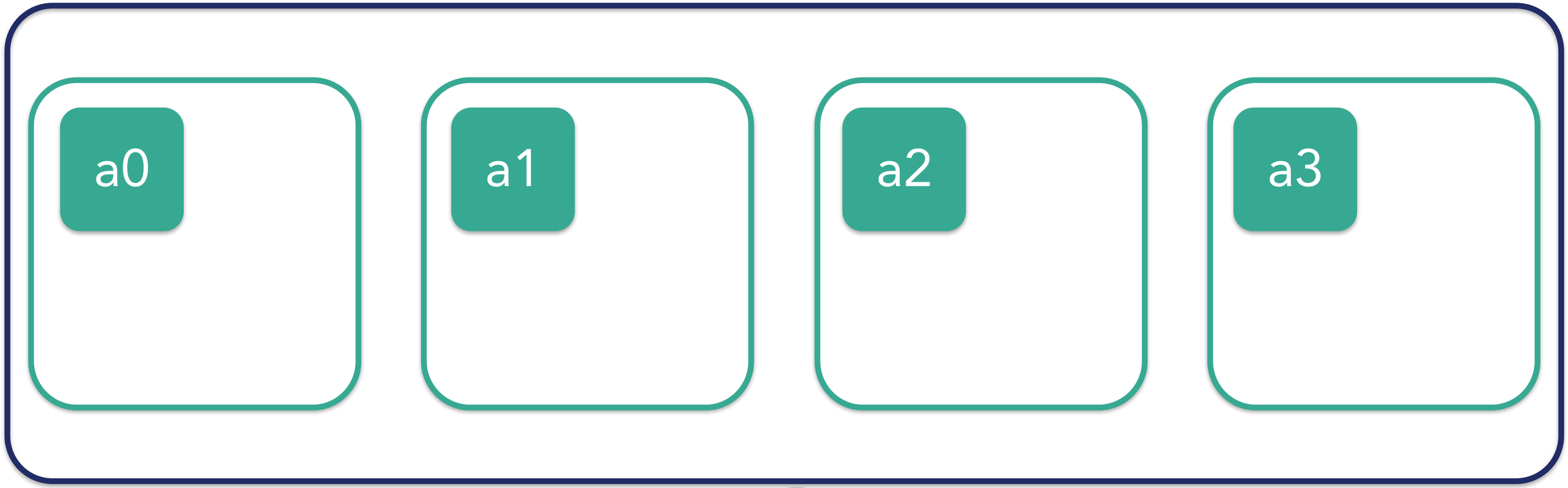
# Shards: Unit of scale

```
# curl -X PUT http://localhost:9200/a -d '
{ "index.number_of_shards" : 4 }'
```



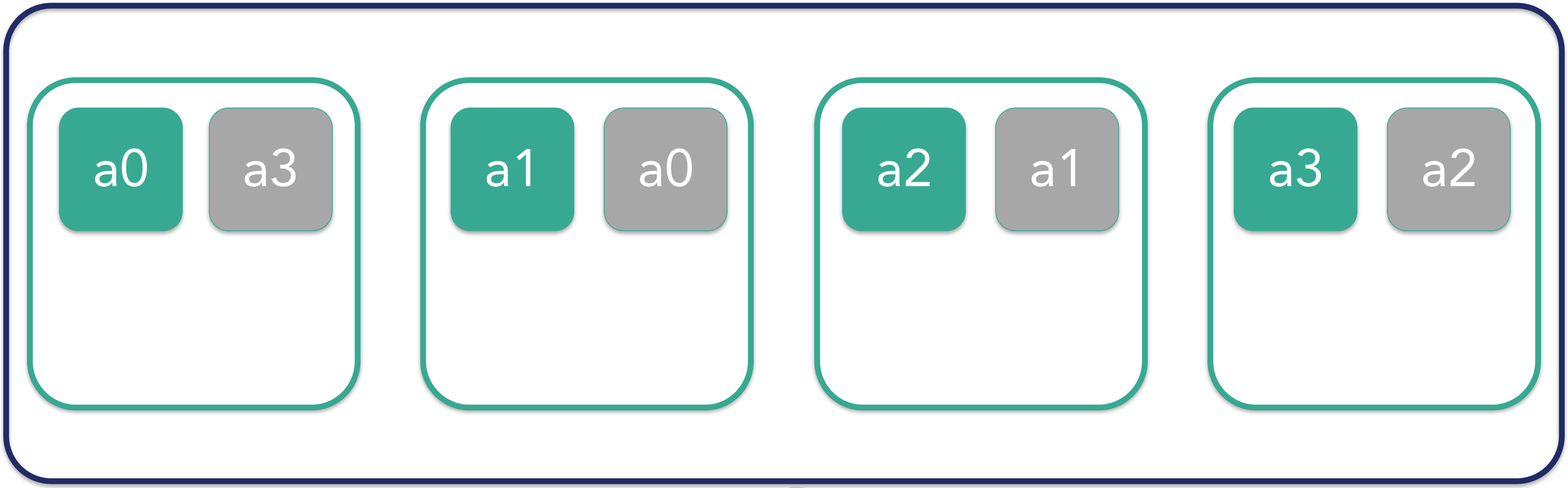
# Shards: Unit of scale

```
# curl -X PUT http://localhost:9200/a/_settings -d '
{ "index.number_of_replicas" : 1 }'
```



# Replication

```
# curl -X PUT http://localhost:9200/a/_settings -d '
{ "index.number_of_replicas" : 1 }'
```



# Search

# CRUD

```
PUT books/book/1
{
  "name" : "Elasticsearch - The definitive guide",
  "authors" : [ "Clinton Gormley", "Zachary Tong" ],
  "pages" : 722,
  "published_at" : "2015/01/31"
}
```

# CRUD

```
PUT books/book/1
{
  "name" : "Elasticsearch - The definitive guide",
  "authors" : [ "Clinton Gormley", "Zachary Tong" ],
  "pages" : 722,
  "published_at" : "2015/01/31"
}
```

```
GET books/book/1
```

# CRUD

```
PUT books/book/1
{
  "name" : "Elasticsearch - The definitive guide",
  "authors" : [ "Clinton Gormley", "Zachary Tong" ],
  "pages" : 722,
  "published_at" : "2015/01/31"
}
```

```
GET books/book/1
```

```
DELETE books/book/1
```

# CRUD

```
PUT books/book/1
{
  "name" : "Elasticsearch - The definitive guide",
  "authors" : [ "Clinton Gormley", "Zachary Tong" ],
  "pages" : 722,
  "published_at" : "2015/01/31"
}
```

```
GET books/book/1
```

```
DELETE books/book/1
```

```
GET books/book/_search?q=elasticsearch
```

# Searching

```
GET books/book/_search
{
  "query" : { "filtered" : {
    "query" : { "match" : { "name" : "elasticsearch" }},
    "filter" : {
      "range" : { "published_at" : { "gte" : "now-1y" } }
    }
  }}
}
```

# Searching

```
GET books/book/_search
{
   "query" :
      "query"
      "filter
         "rang
      }
   }}
}
```

```
{
   "took": 3, "timed_out": false,
   "_shards": { "total": 5, "successful": 5, "failed": 0 },
   "hits": {
      "total": 1, "max_score": 0.15342641,
      "hits": [ {
            "_index": "books", "_type": "book", "_id": "1",
            "_score": 0.15342641,
            "_source": {
               "name": "Elasticsearch - The definitive guide",
               "authors": [ "Clinton Gormley", "Zachary Tong" ],
               "pages": 722, "category": "search"
               "published_at": "2015/01/31",
} } ] } }
```

# Searching

```
GET books/book/_search
{
  "query" : { "filtered" : {
    "query" : { "match" : { "name" : "elasticsearch" }},
    "filter" : {
      "range" : { "published_at" : { "gte" : "now-1y" } }
    }
  }},
  "aggs" : {
    "category" : { "terms" : { "field" : "category" } }
  }
}
```

# Searching

```
GET books/book/_search
{
   "query" :
     "query"
     "filter
       "rang
     }
   }},
   "aggs" :
     "catego
   }
}
```

```
{
   "took": 3, "timed_out": false,
   "_shards": { "total": 5, "successful": 5, "failed": 0 },
   "hits": {
      "total": 1, "max_score": 0.15342641,
      "hits": [ ... ]
   },
   "aggregations": {
      "category": {
        "buckets": [
            { "key": "search", "doc_count": 1 },
            { ... }
] } } }
```

# Search

- Hits all relevant shards
- Searches for top-N results per shard
- Reduces to top-N total
- Gets top-N documents/data from relevant shards
- Returns data to requesting client

# Search on a single shard

- Lucene is doing the heavy lifting
- A single shard is a Lucene index
- Each field is its own inverted index and can be searched in

| term | docid |
|---|---|
| clinton | 1 |
| gormley | 1 |
| tong | 1 |
| zachary | 1 |

# Example: Return original JSON in search response

- It is very hard to reconstruct the original data from the inverted index
- Solution: Just store the whole document in its own field and retrieve it, when returning data to the client

```
{                                                          _source
  "name" : "Elasticsearch - The definitive guide",
  "authors" : [ "Clinton Gormley", "Zachary Tong" ],
  "pages" : 722,
  "published_at" : "2015/01/31"
}
```

# Example: `_all` field

```
{
  "name" : "Elasticsearch - The definitive guide",
  "authors" : [ "Clinton Gormley", "Zachary Tong" ],
  "pages" : 722,
  "published_at" : "2015/01/31"
}
```

```
Elasticsearch - The definitive guide
Clinton Gormley Zachary Tong
722
2015/01/31
```

`_all`

# Example: **copy_to** field (name & authors)

```
{
  "name" : "Elasticsearch - The definitive guide",
  "authors" : [ "Clinton Gormley", "Zachary Tong" ],
  "pages" : 722,
  "published_at" : "2015/01/31"
}
```

```
Elasticsearch - The definitive guide
Clinton Gormley Zachary Tong
```

**copy_to**

# Search: Using filters

- Filters do not contribute to score & can be cached using a BitSet
  - range filter for a date/price
  - term filter for a category
  - geo filter for a bounding box

| 0 | 1 | 1 | 0 | 0 | 1 |
|---|---|---|---|---|---|

```
{ "term" : { "category" : "search" } }
```

| 0 | 1 | 0 | 0 | 1 | 0 |
|---|---|---|---|---|---|

```
{ "term" : { "category" : "reduced" } }
```

# Filters: Missing fields

- Problem: How to search in an inverted index for non-existing fields (`exists` & `missing` filter)?
- Costly: Need to merge postings lists of all existing terms (expensive for high-cardinality fields!)
- Solution: Index document field names under `_field_names`

# Aggregations

# Aggregations

- Aggregations: Buckets & metrics
- Aggregations cannot make use of the inverted index
- Meet Fielddata: Uninverting the index
- Inverted index: Maps term to document id
- Fielddata: Maps document id to terms

# Aggregations

## Inverted Index

| term | docid |
|---|---|
| clinton | 1 |
| gormley | 1 |
| tong | 1 |
| zachary | 1 |

## Fielddata

| docid | term |
|---|---|
| 1 | Clinton Gormley, Zachary Tong |

# Aggregations

## Inverted Index

| term | docid |
| --- | --- |
| Clinton Gormley | 1 |
| Zachary Tong | 1 |

## Fielddata

| docid | term |
| --- | --- |
| 1 | Clinton Gormley, Zachary Tong |

# Aggregations: Fielddata

- Fielddata is an in-memory data structure, lazily constructed
- Easy to go OOM (wrong field or too many documents)
- Solution:
  - circuit breaker
  - **`doc_values`**: index-time data structure, no heap, uses the file system cache, better compression

# Aggregations: Probabilistic data structures

- Problem: Count distinct elements
- Naive: Load all data into a set, then check the size (distributed?)
- Solution: `cardinality` Aggregation, that uses HyperLogLog++
  - configurable precision, allows to trade memory for accuracy
  - excellent accuracy on low-cardinality sets
  - fixed memory usage: no matter if there are tens or billions of unique values, memory usage only depends on configured precision

# Aggregations: Probabilistic data structures

- Problem: Calculate percentiles
- Naive: Maintain a sorted list of all values
- Solution: `percentiles` Aggregation, that uses T-Digest
  - extreme percentiles are more accurate
  - small sets can be up to 100% accurate
  - while values are added to a bucket, the algorithm trades accuracy for memory savings

# Operating system & Hardware

# Elasticsearch can easily max out...

- CPU

  Indexing, searching, highlighting

- I/O

  Indexing, searching, merging

- Memory

  Aggregation, indices

- Network

  Relocation, Snapshot & Restore

# Hardware

- CPU: Threadpools are sized on number of cores
- Disk: SSD
- Memory: ∞
- Network: GbE or better

# Operating system

- file system cache
- file handles
- memory locking: `bootstrap.mlockall`
- dont swap, no OOM killer

# Distributed aspects

# Fallacies of distributed computing

- The network is reliable
- Latency is zero
- Bandwidth is infinite
- The network is secure
- Topology doesn't change
- There is one administrator
- Transport cost is zero
- The network is homogeneous

by Peter Deutsch
https://en.wikipedia.org/wiki/Fallacies_of_distributed_computing

# Wrapup

# Summary

- Speed is key!
- Search is a tradeoff: Query time vs. index time
- Benchmark your use-case
  - http://benchmarks.elasticsearch.org/

# Elasticsearch 2.x

- Automatic I/O throttling
- Clusterstate incremental updates
- Faster recovery
- Aggregations 2.0
- Merge queries and filters
- Reindex API
- Changes API
- Expression scripting engine

# Lucene 5.x

- Speed improvements in queries (must_not, sloppy phrase)
- Automated caching
- BitSet compression vastly improved (roaring bitsets)
- Index compression (on disk + memory)
- Indexing performance (adaptive merge throttling, SSD detection)
- Index safety: atomic commits, segment commit identifiers, verify integrity at merge
- ...

# Thanks for listening! Questions?

Alexander Reelsen
alex@elastic.co
@spinscale

We’re hiring
https://www.elastic.co/about/careers

We’re helping
https://www.elastic.co/subscriptions

# References


http://www.elastic.co/guide/en/elasticsearch/reference/current/mapping-source-field.html

http://www.elastic.co/guide/en/elasticsearch/reference/current/mapping-all-field.html

http://www.elastic.co/guide/en/elasticsearch/reference/current/mapping-field-names-field.html

http://www.elastic.co/guide/en/elasticsearch/reference/current/mapping-core-types.html#copy-to

http://www.elastic.co/guide/en/elasticsearch/reference/current/search-aggregations-metrics-cardinality-aggregation.html

http://www.elastic.co/guide/en/elasticsearch/guide/current/cardinality.html

http://www.elastic.co/guide/en/elasticsearch/guide/current/percentiles.html

http://www.elastic.co/guide/en/elasticsearch/reference/current/search-aggregations-metrics-percentile-aggregation.html

http://www.elastic.co/elasticon/2015/sf/elasticsearch-architecture-amusing-algorithms-and-details-on-data-structures/

http://speakerdeck.com/elastic/all-about-aggregations

http://www.elastic.co/elasticon/2015/sf/updates-from-lucene-land

http://speakerdeck.com/elastic/resiliency-in-elasticsearch-and-lucene

http://www.elastic.co/elasticon/2015/sf/level-up-your-clusters-upgrading-elasticsearch

http://speakerdeck.com/elasticsearch/maintaining-performance-in-distributed-systems