elastic

Les vendredi noirs ?

Même pas peur !

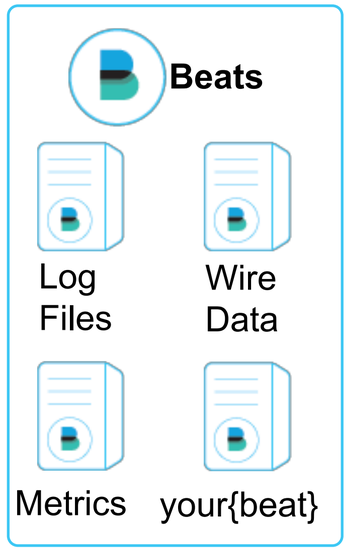
David Pilato
Developer | Evangelist, @dadoonet

DevOpsDays Geneva

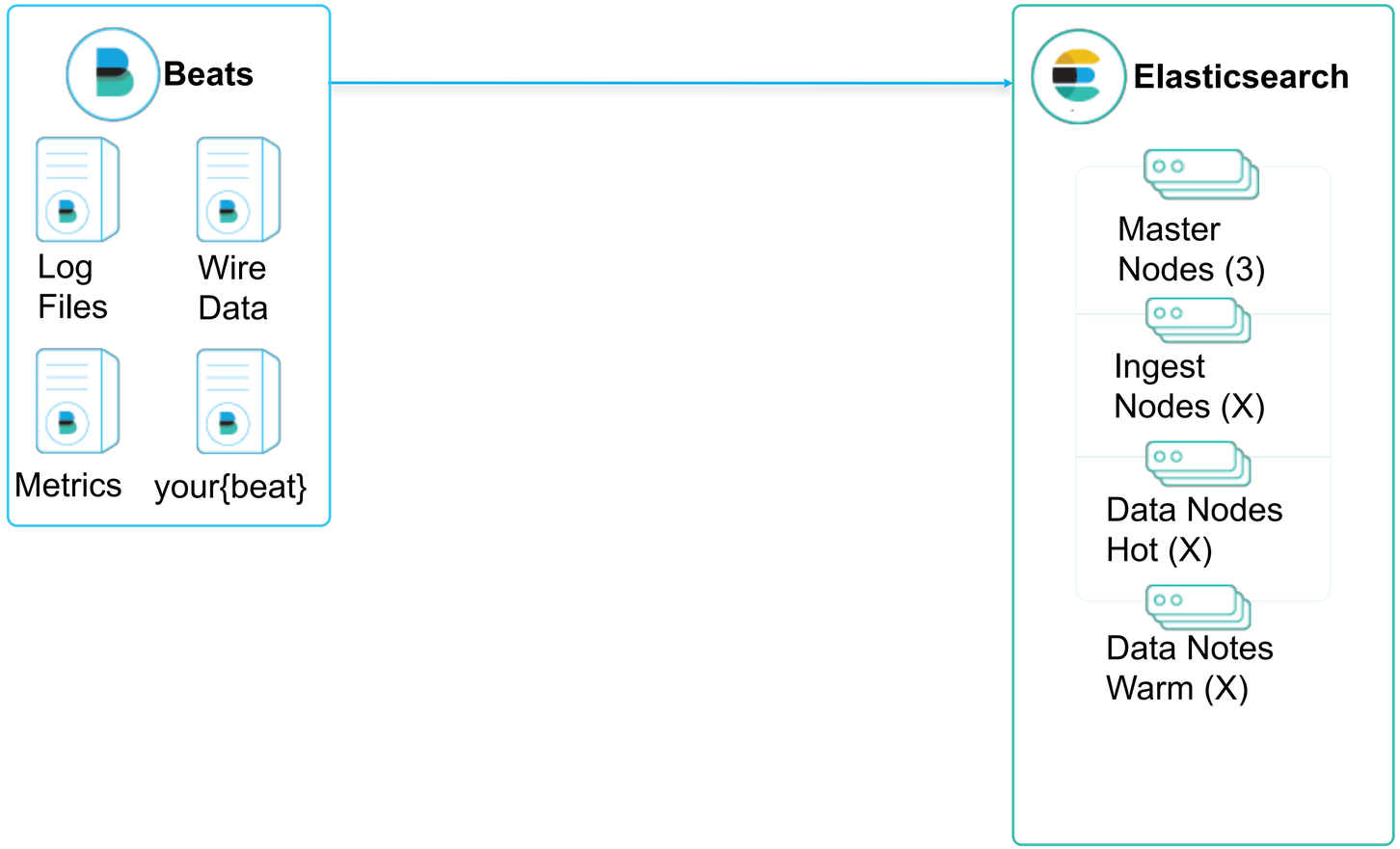
Data Platform Architectures

elastic

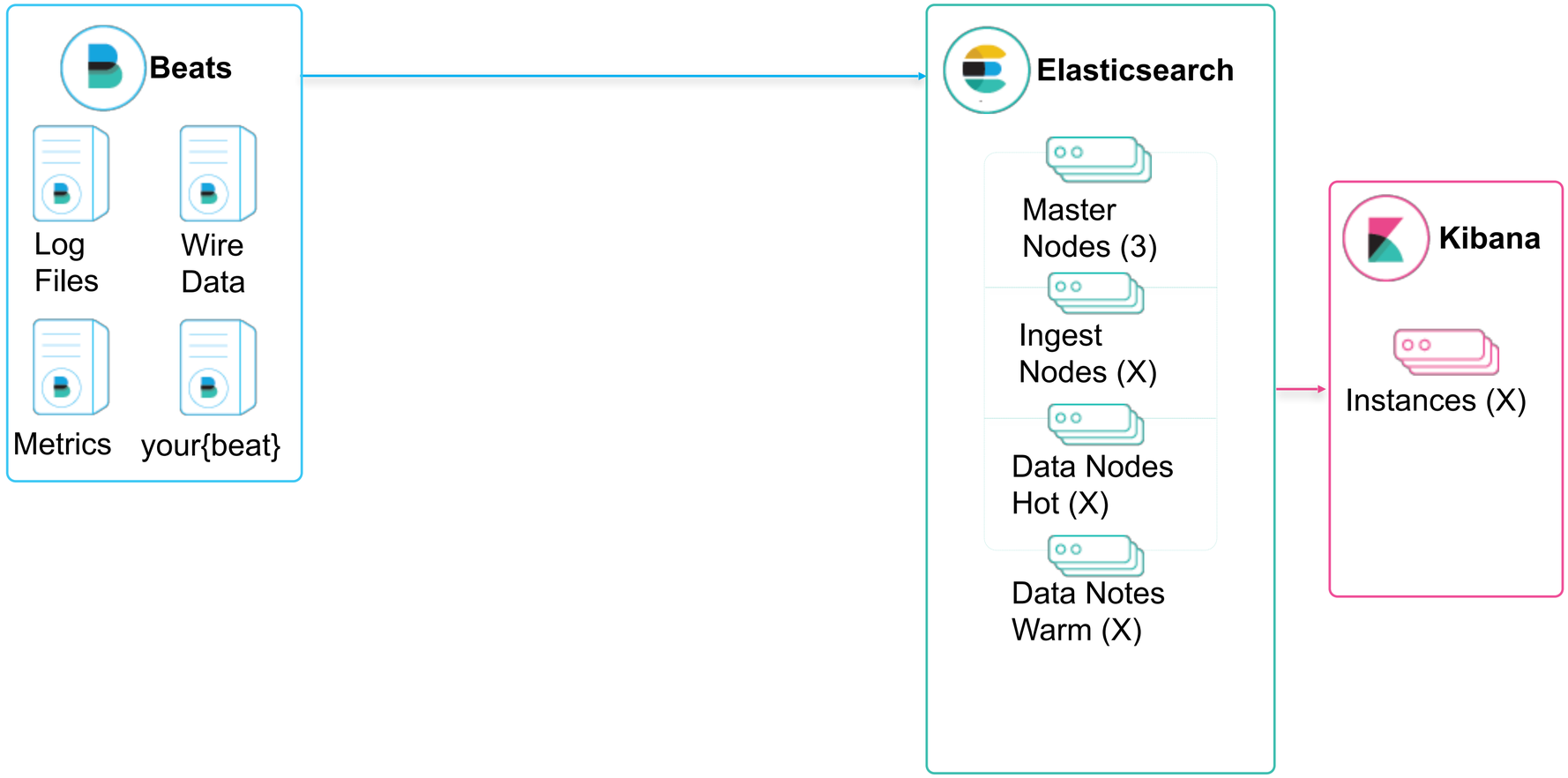
The Elastic Journey of Data



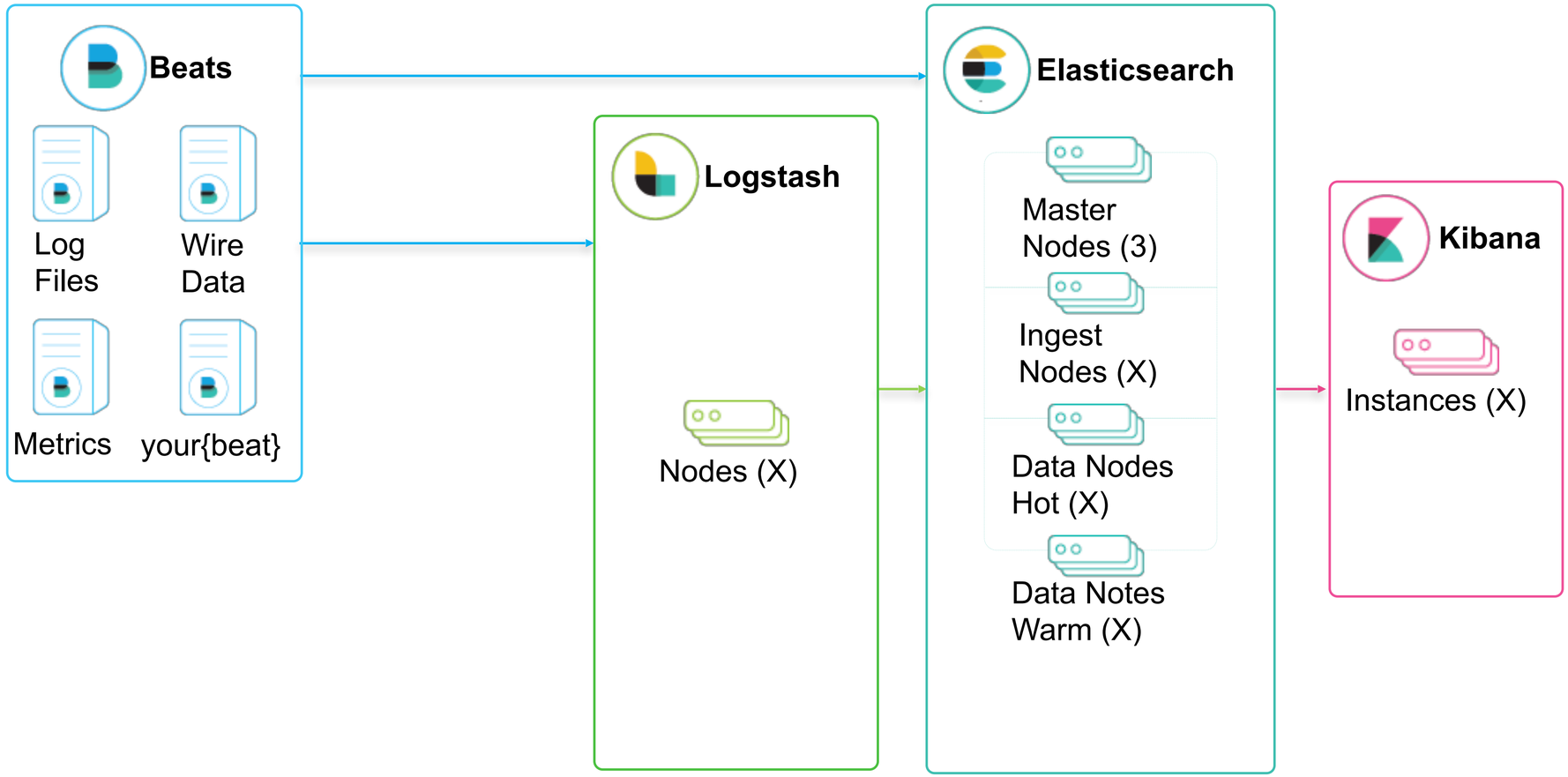
The Elastic Journey of Data



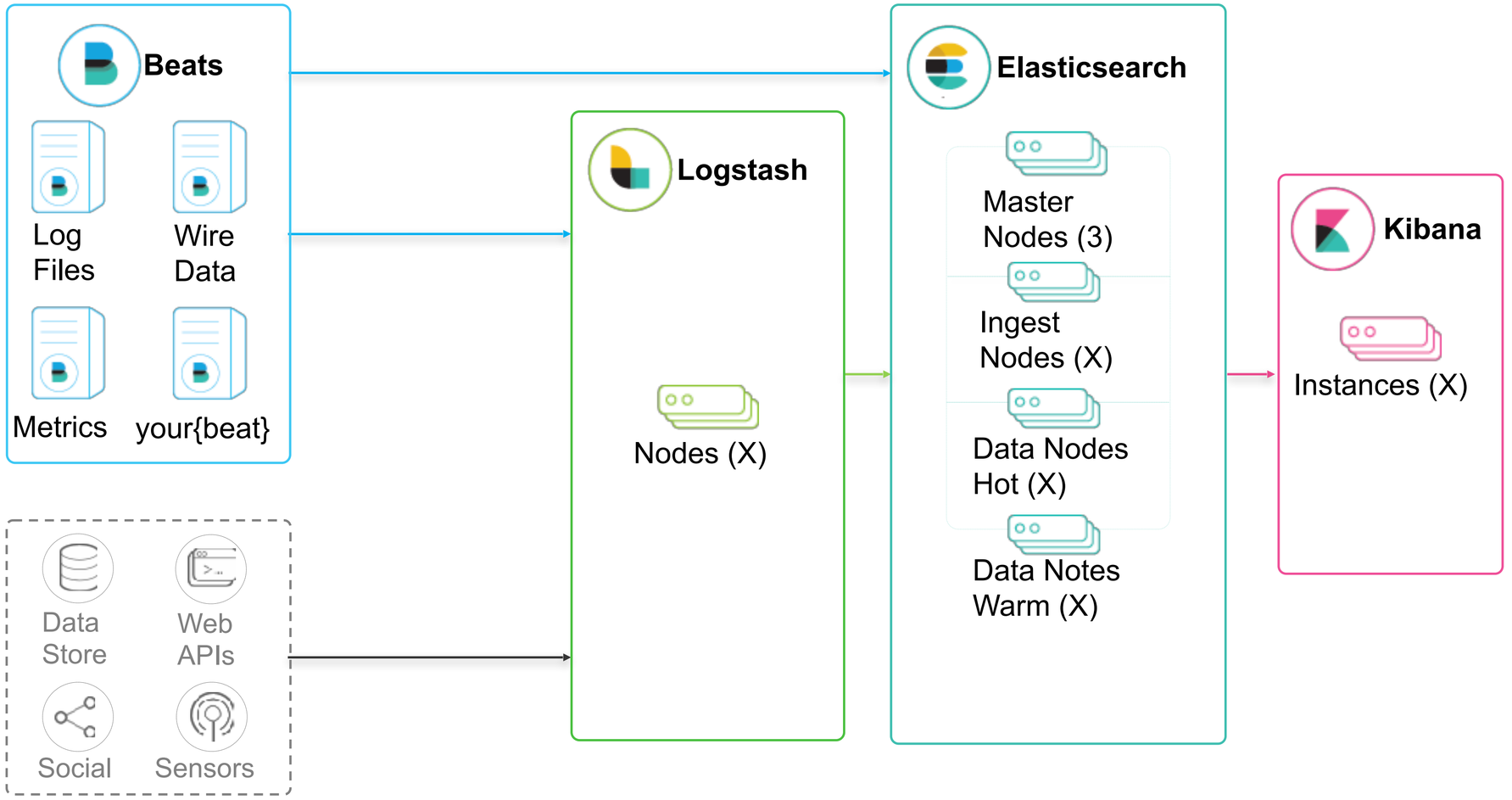
The Elastic Journey of Data



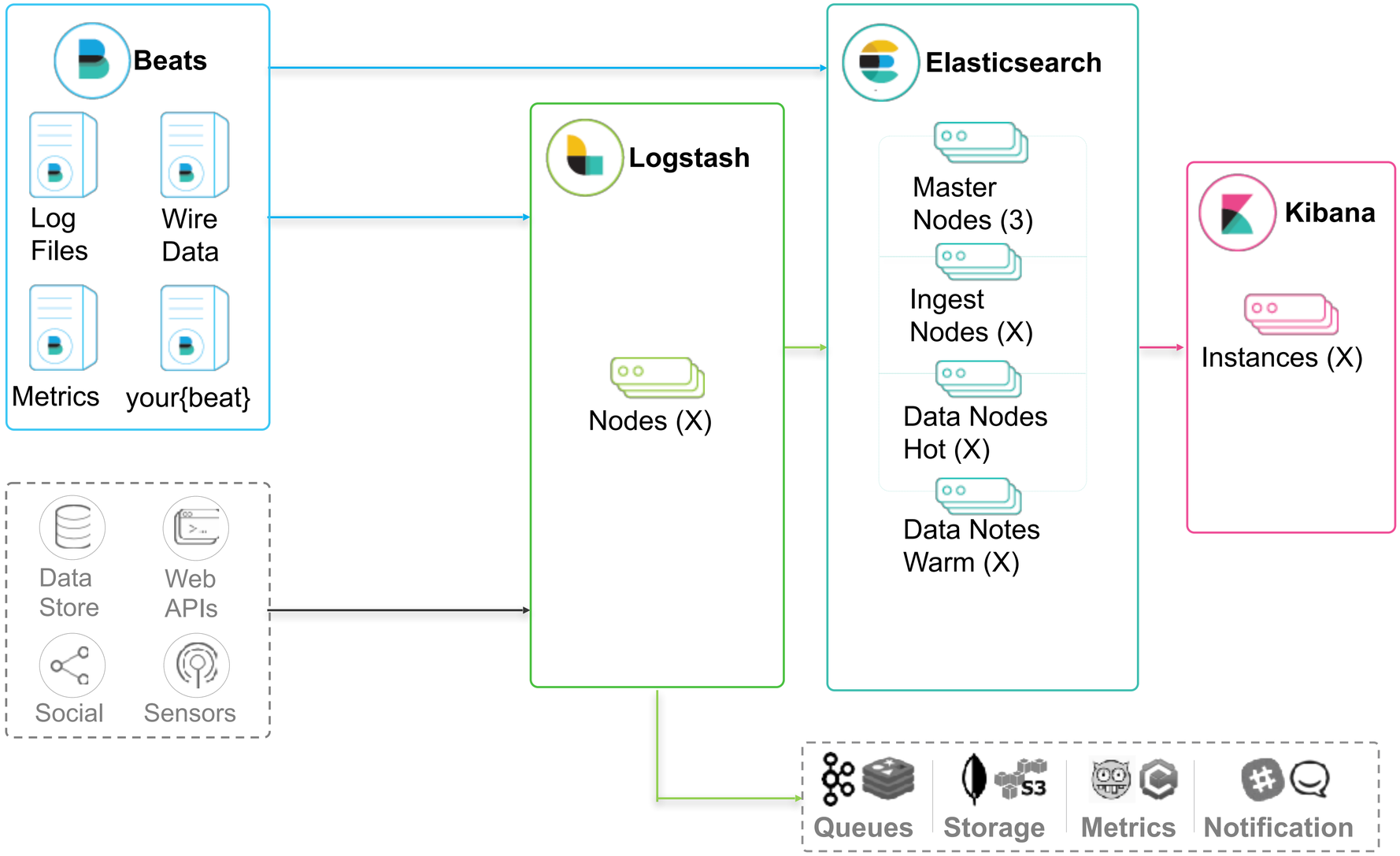
The Elastic Journey of Data



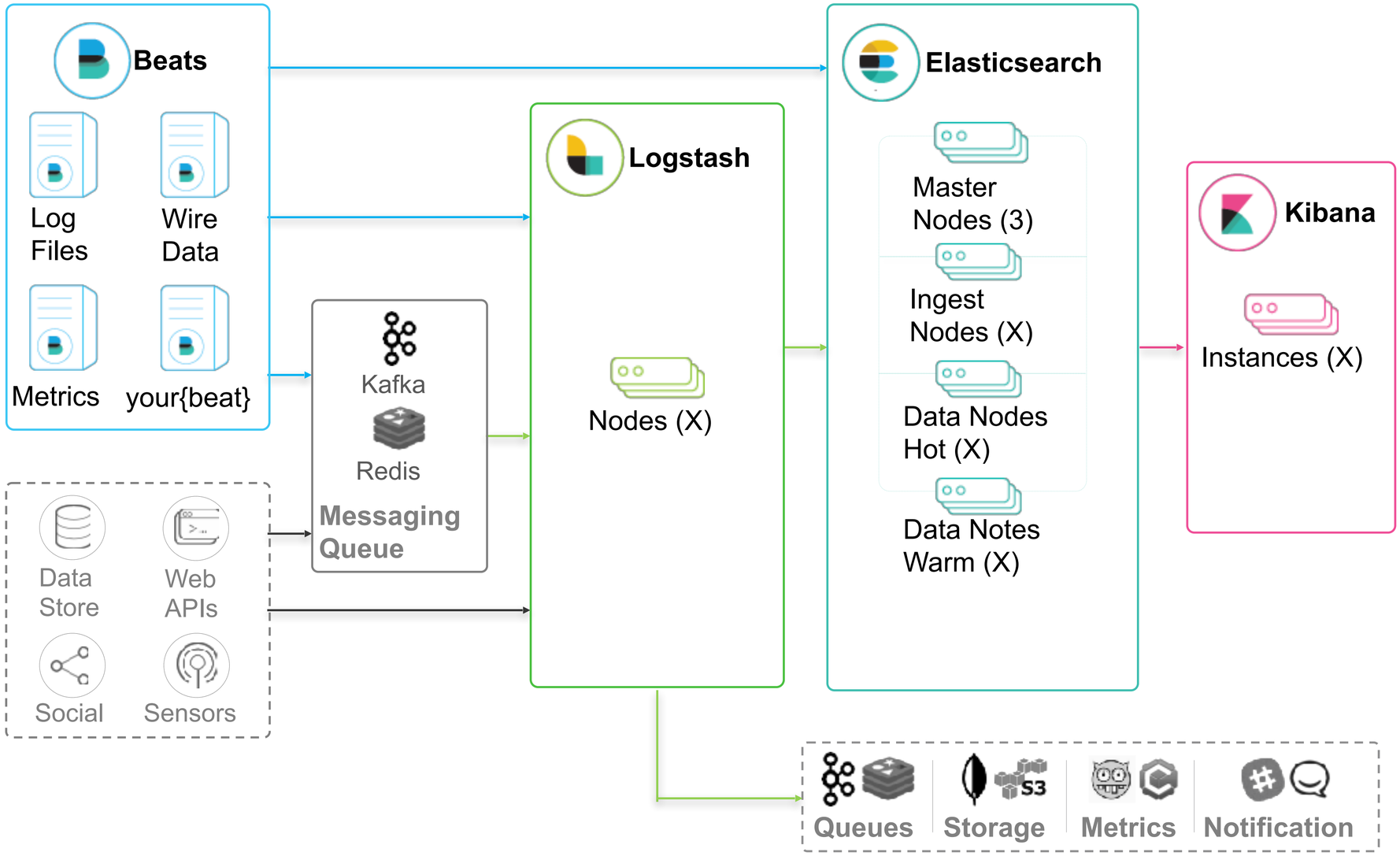
The Elastic Journey of Data

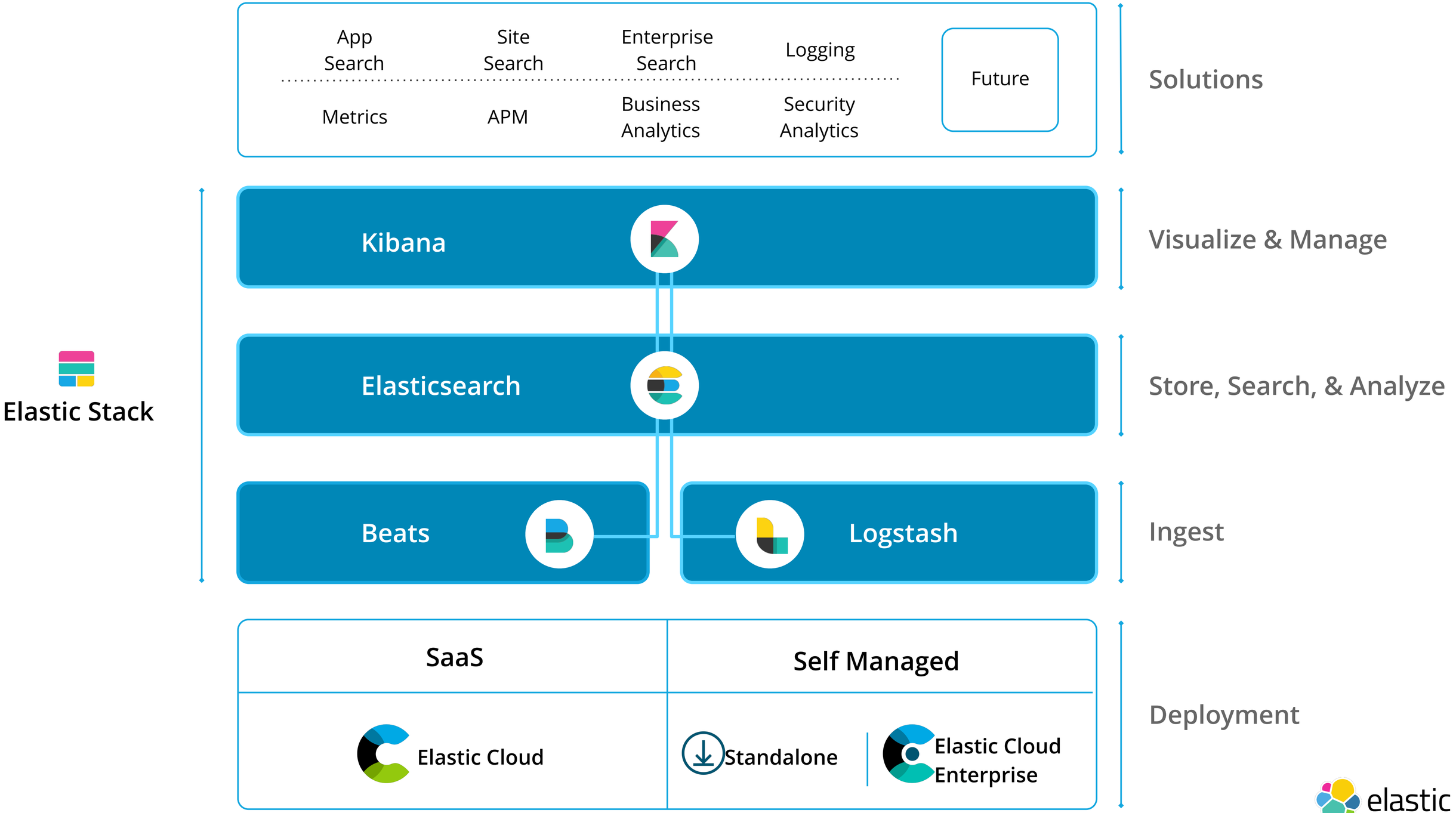


The Elastic Journey of Data



The Elastic Journey of Data

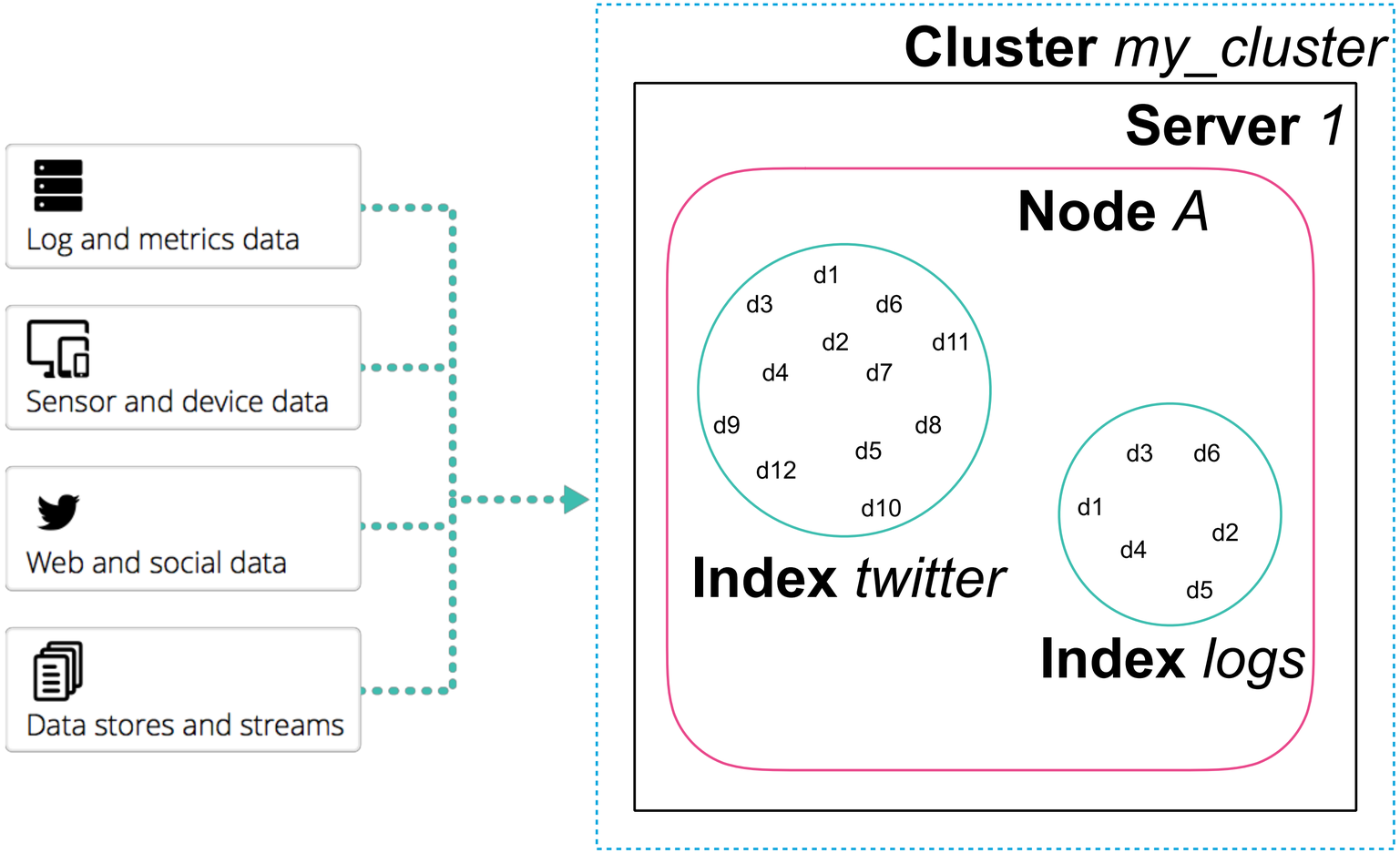



App Search
Site Search
Enterprise Search
Logging
Future
Metrics
APM
Business Analytics
Security Analytics
Solutions
Elastic Stack
Kibana
Visualize & Manage
Elasticsearch
Store, Search, & Analyze
Beats
Logstash
Ingest
SaaS
Self Managed
Elastic Cloud
Standalone
Elastic Cloud Enterprise
Deployment
elastic

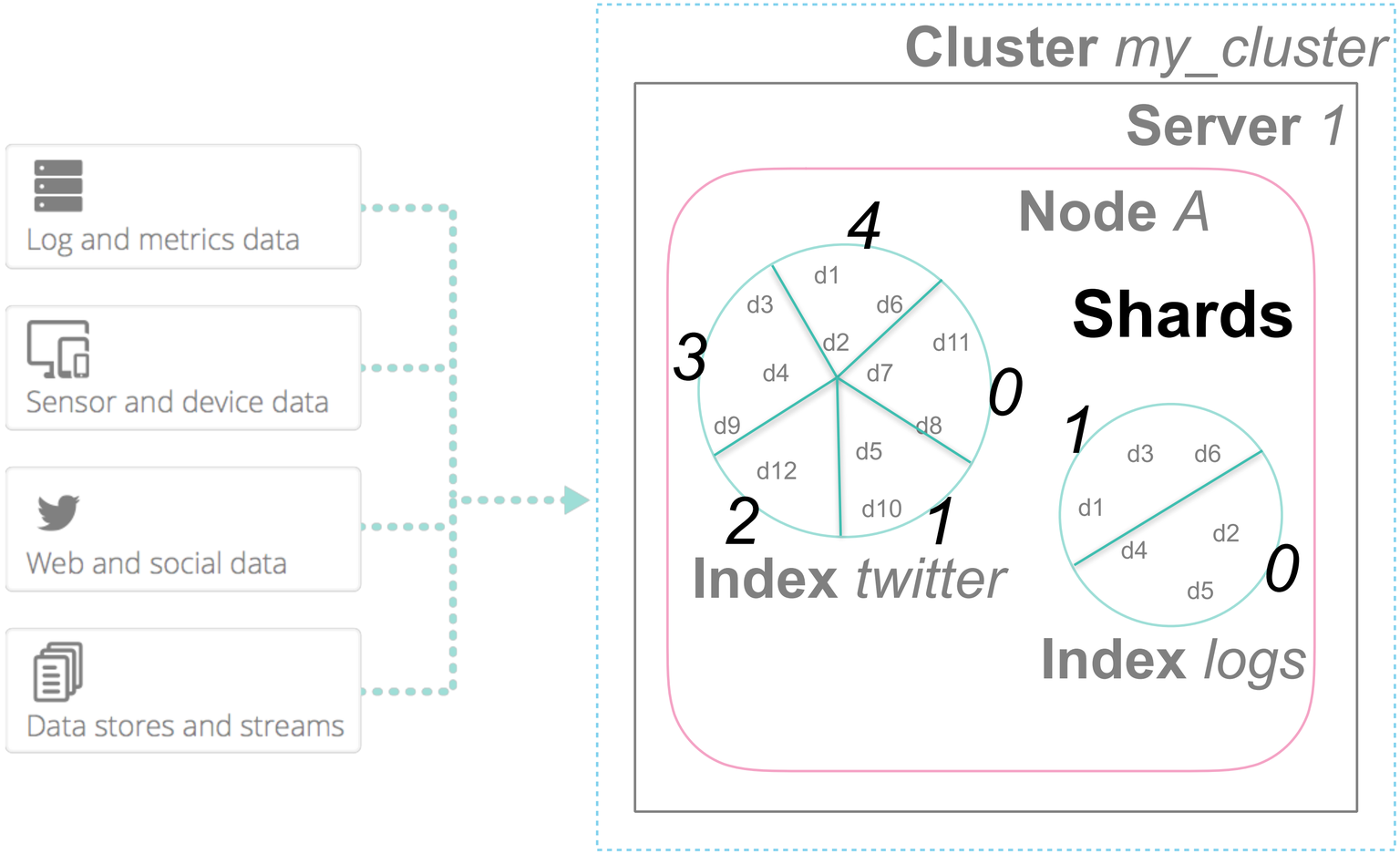
Elasticsearch Cluster Sizing

elastic

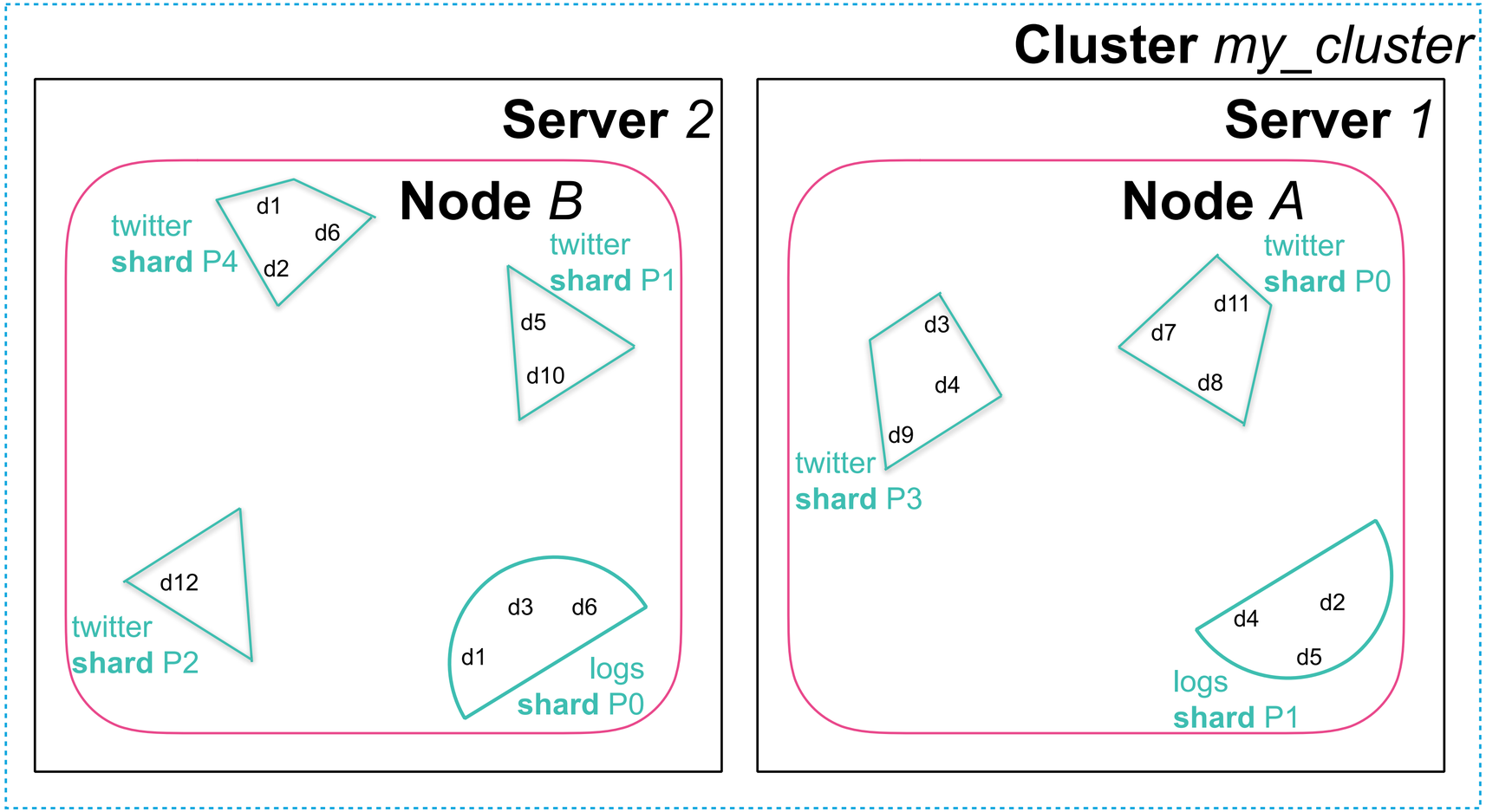
Terminology



Partition

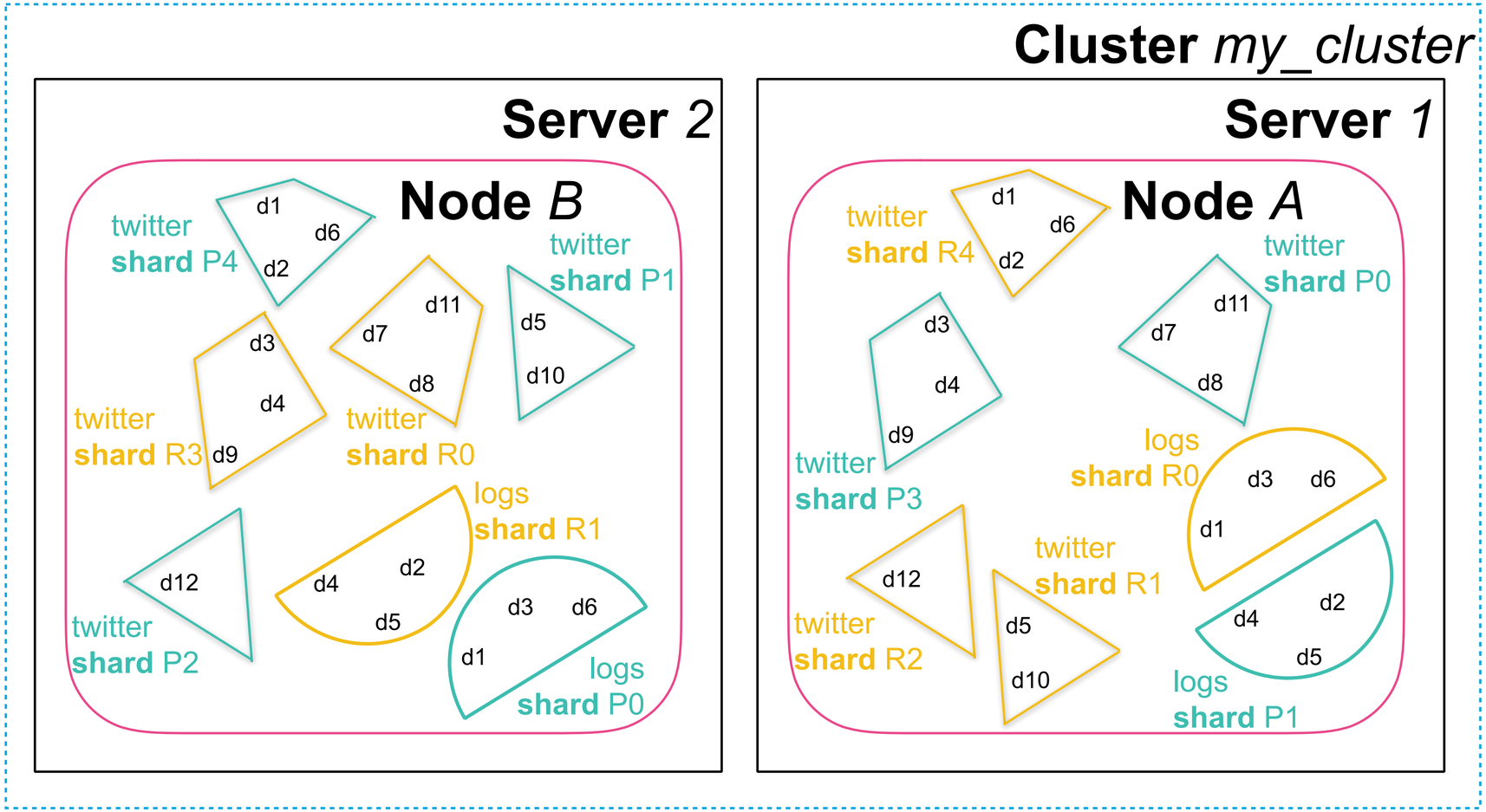


Distribution



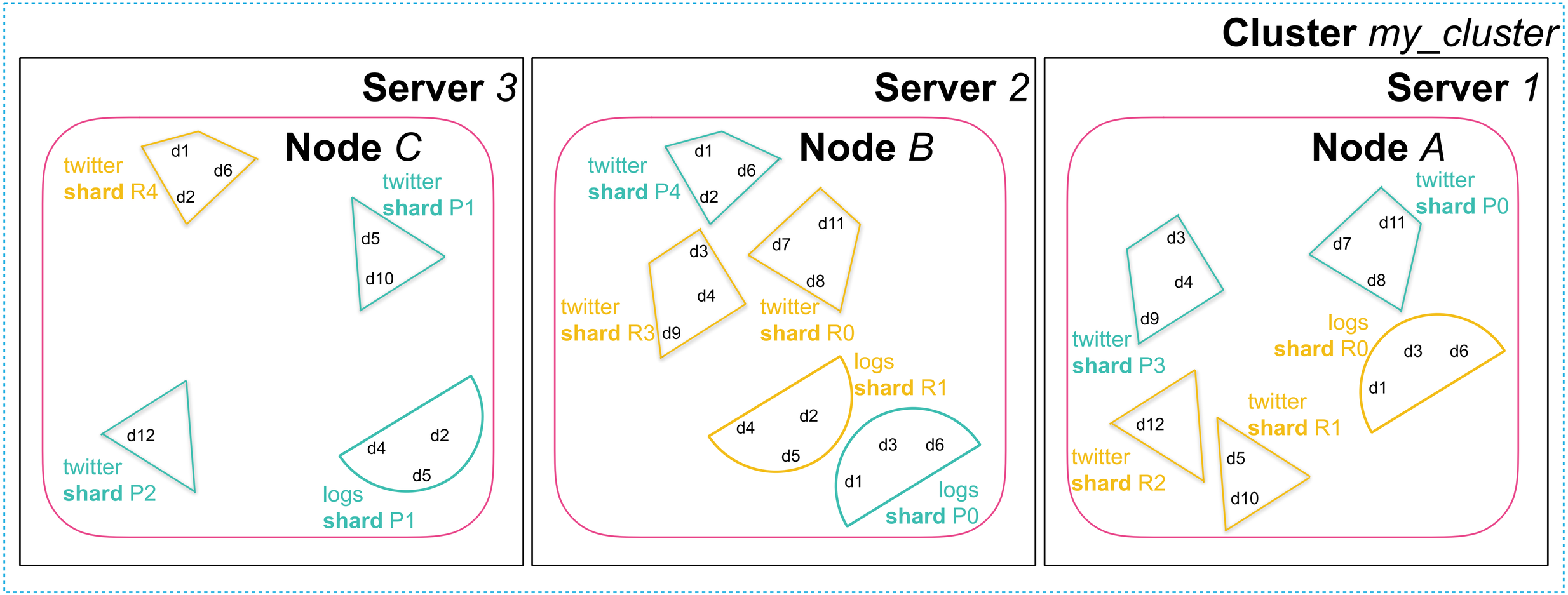
Replication

- Primaries
- Replicas



Replication

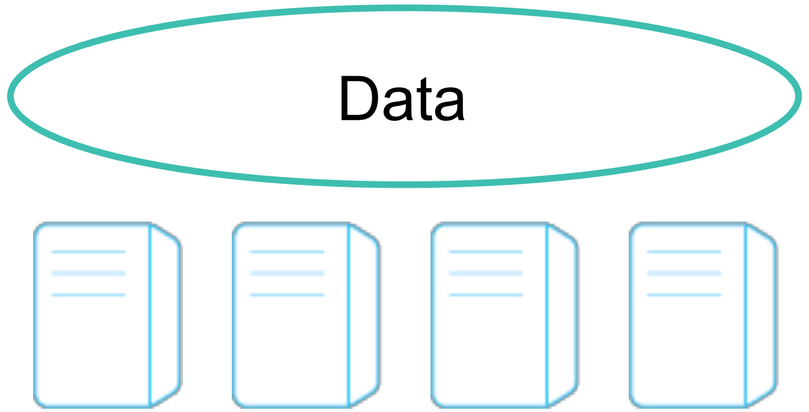
- Primaries
- Replicas



Scaling

Scaling

Scaling



Scaling

elastic

Scaling

- In Elasticsearch, **shards** are the **working unit**
- More data -> More shards

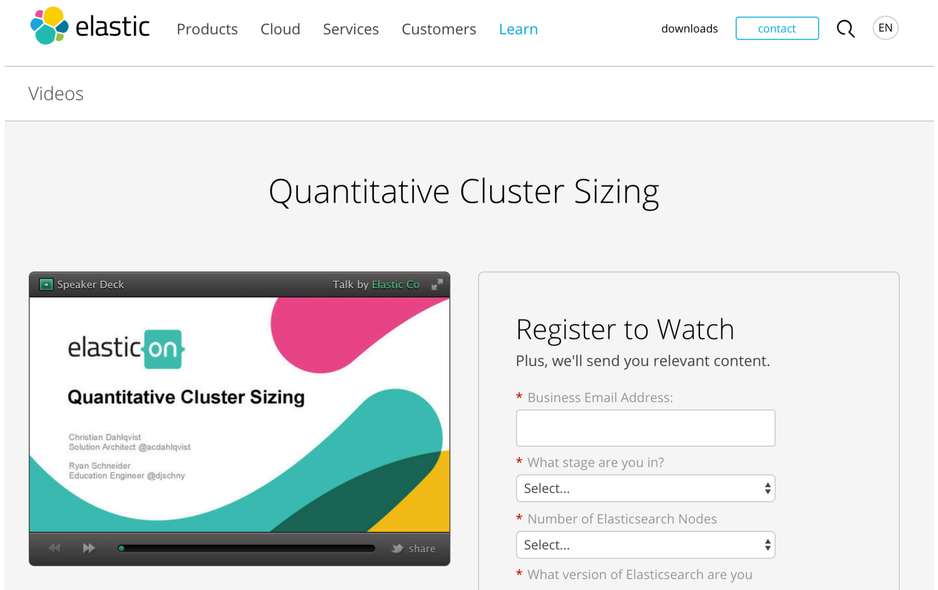
But how many shards?

How much data?

- ~1000 events per second
- 60s * 60m * 24h * 1000 events => ~87M events per day
- 1kb per event => ~82GB per day
- 3 months => ~7TB

Shard Size

- It depends on many different factors
  - document size, mapping, use case, kinds of queries being executed, desired response time, peak indexing rate, budget, ...
- After the shard sizing*, each shard should handle **45GB**
- Up to 10 shards per machine

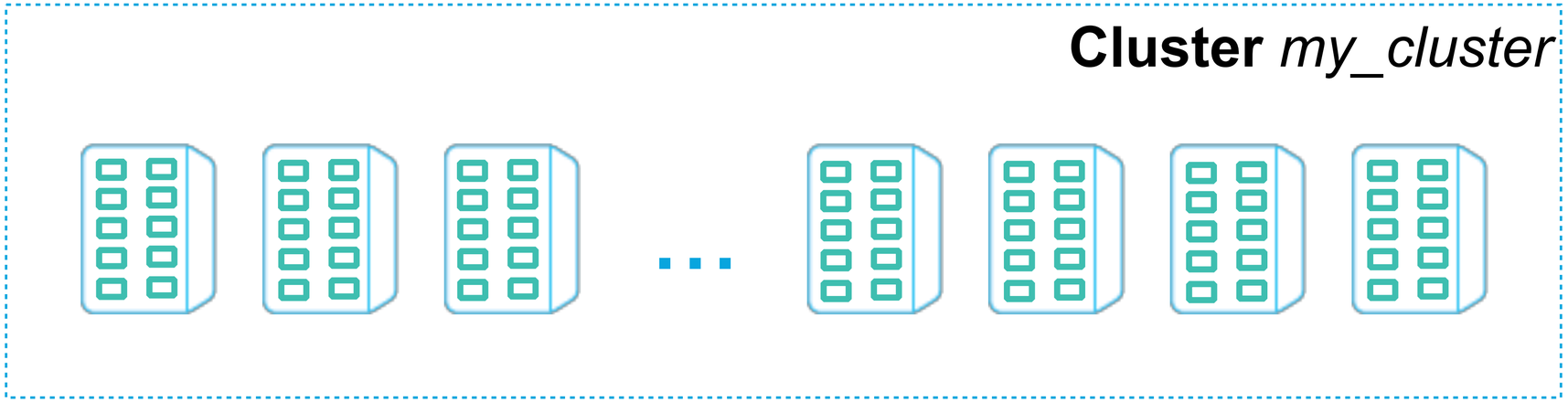


* https://www.elastic.co/elasticon/conf/2016/sf/quantitative-cluster-sizing

How many shards?

- Data size: ~7TB
- Shard Size: ~45GB*
- **Total Shards: ~160**

- Shards per machine: 10*
- **Total Servers: 16**



* https://www.elastic.co/elasticon/conf/2016/sf/quantitative-cluster-sizing

elastic

But...

- How many indices?
- What do you do if the daily data grows?
- What do you do if you want to delete old data?

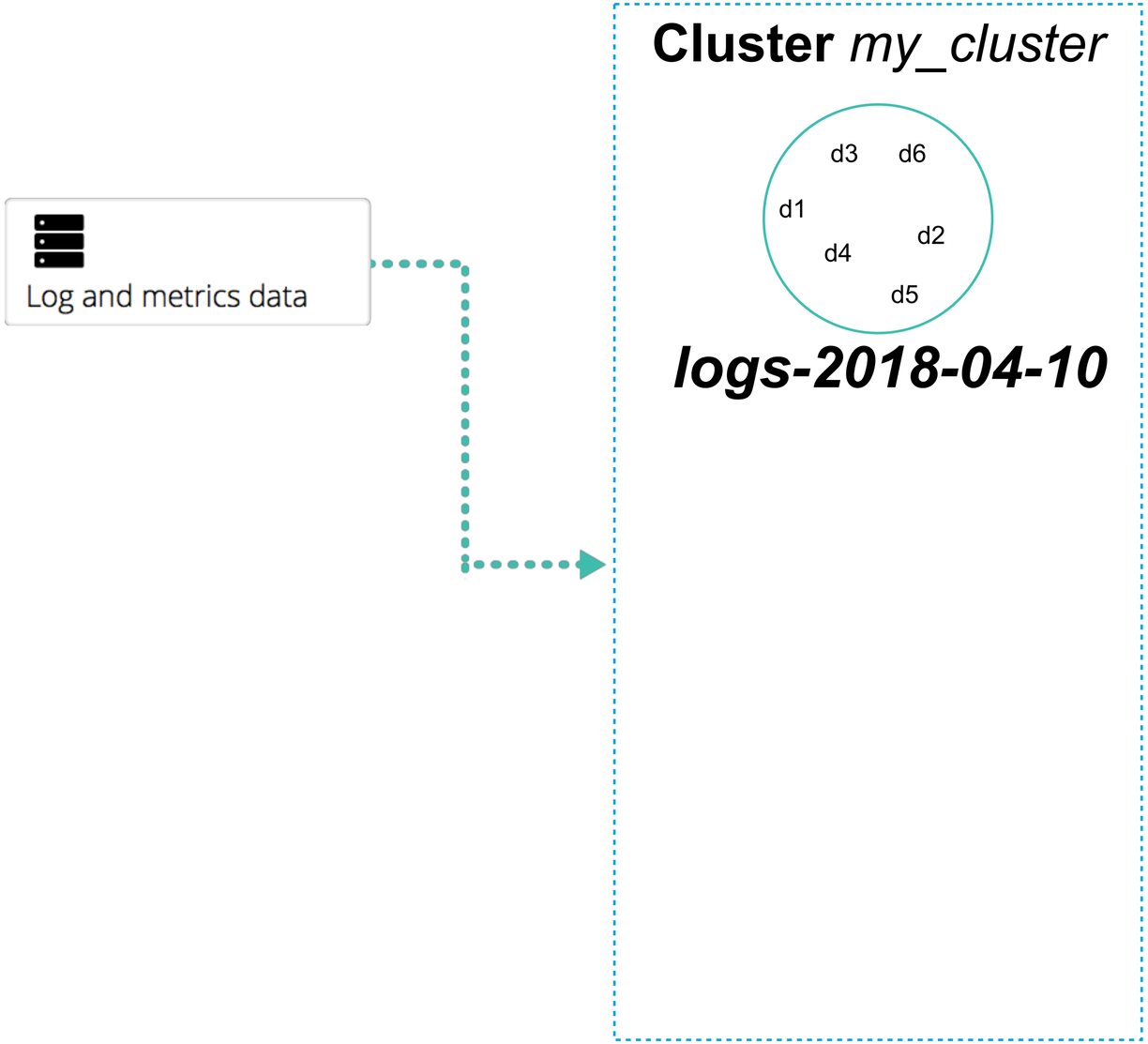
Time-Based Data

- Logs, social media streams, time-based events
- Timestamp + Data
- Do not change
- Typically search for recent events
- Older documents become less important
- Hard to predict the data size

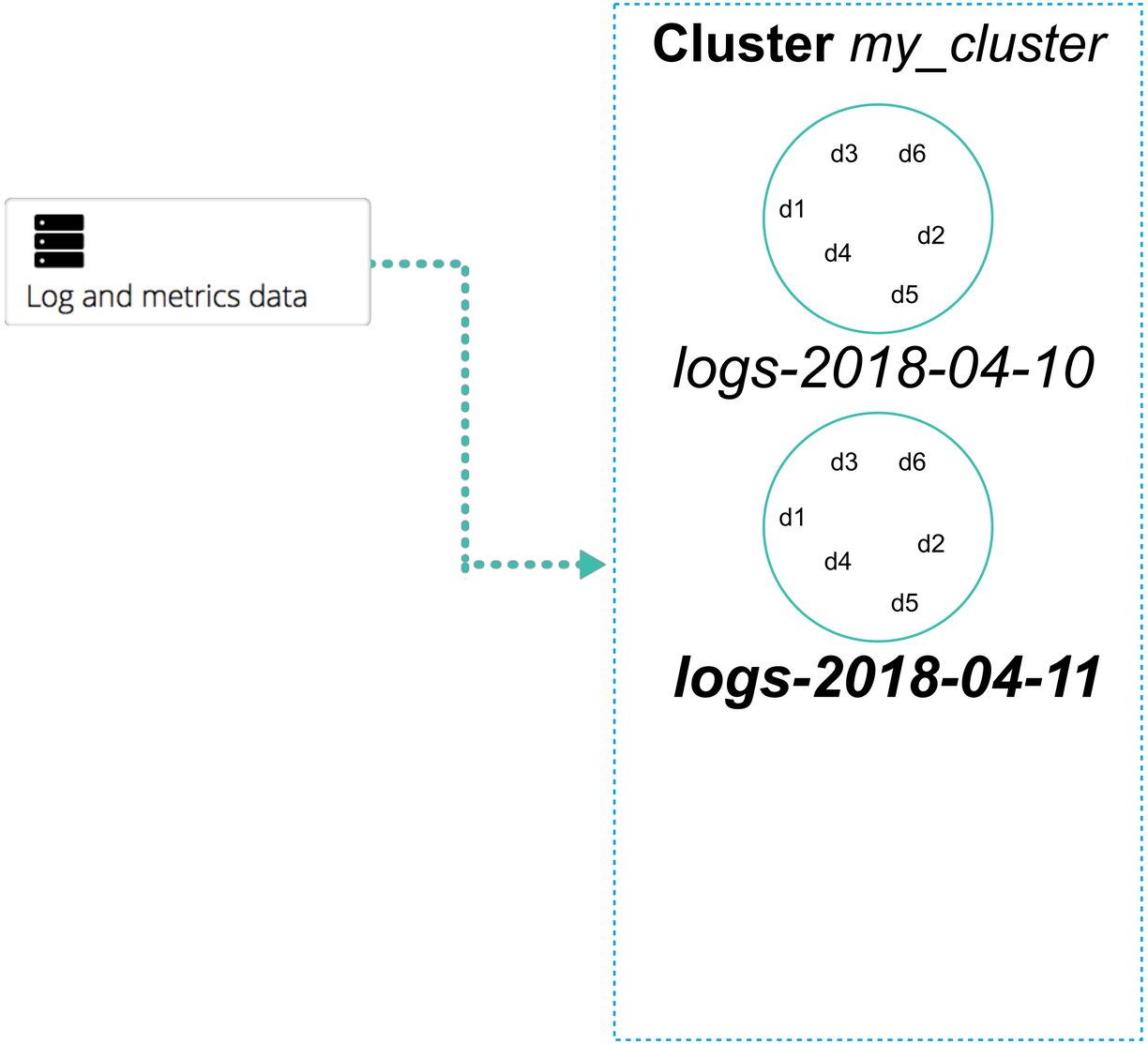
Time-Based Data

- Time-based Indices is the best option
  - create a new index each day, week, month, year, ...
  - search the indices you need in the same request

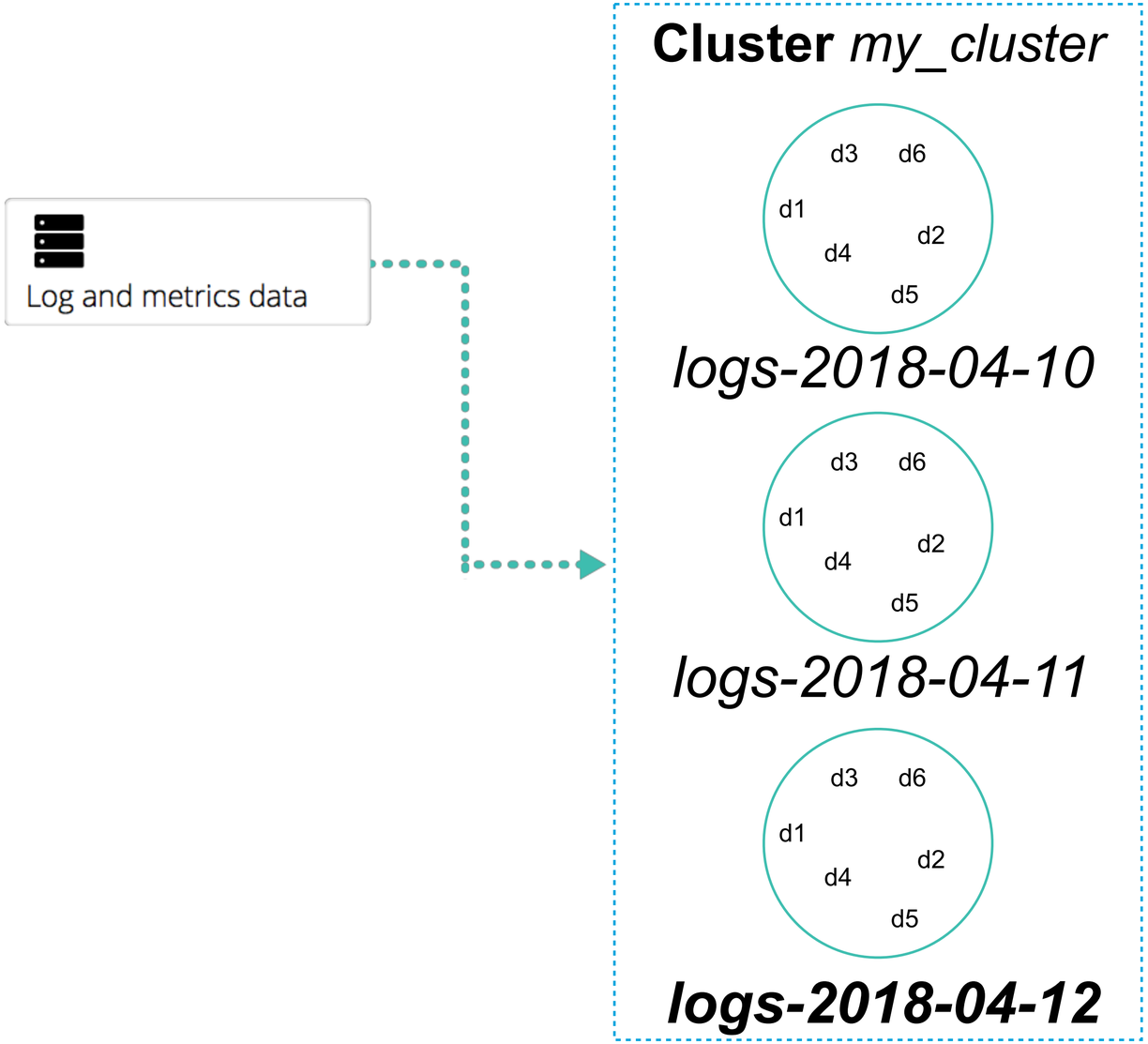
Daily Indices



Daily Indices



Daily Indices



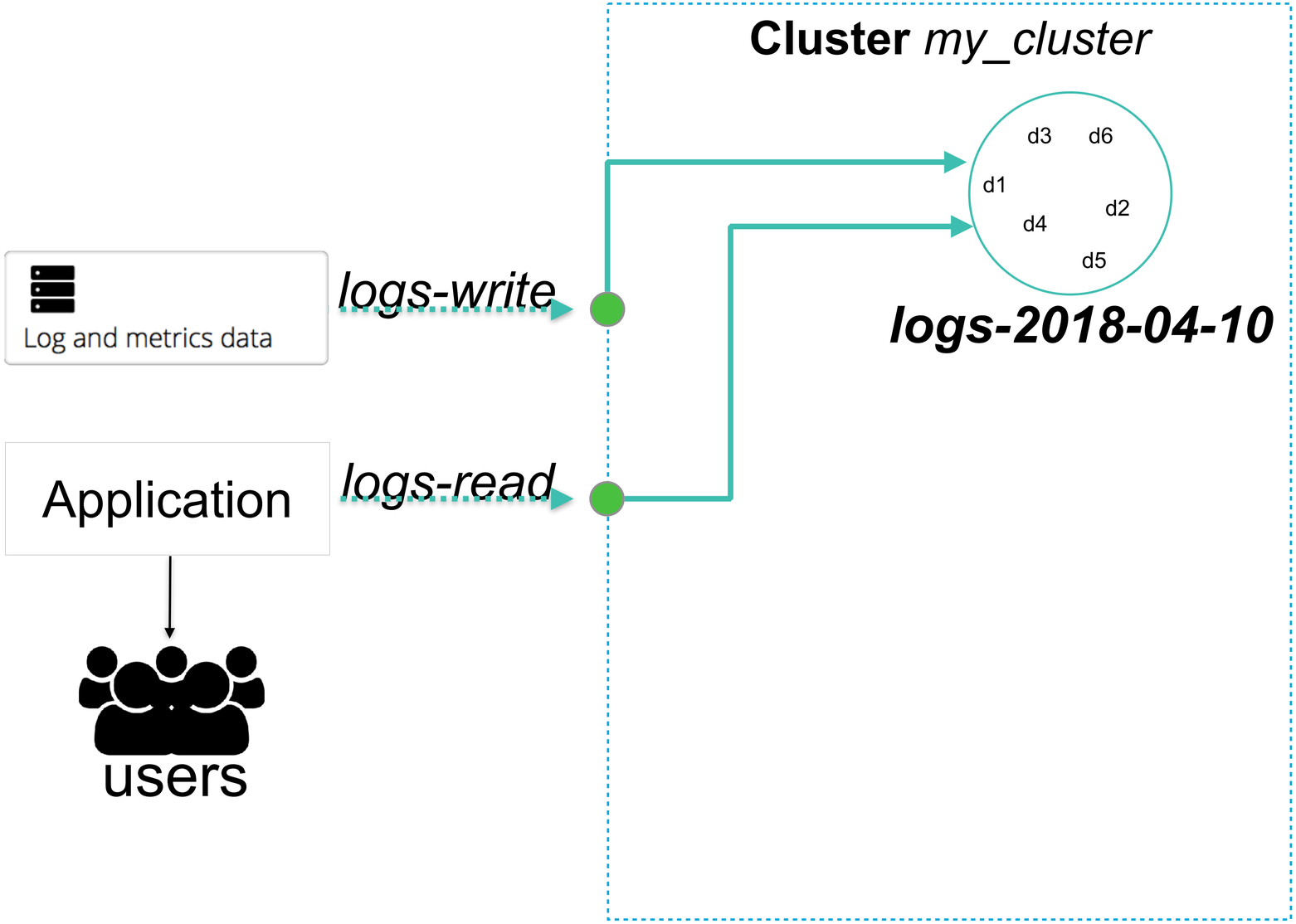
Templates

- Every new created index starting with '*logs-*' will have
  - 2 shards
  - 1 replica (for each primary shard)
  - 60 seconds refresh interval

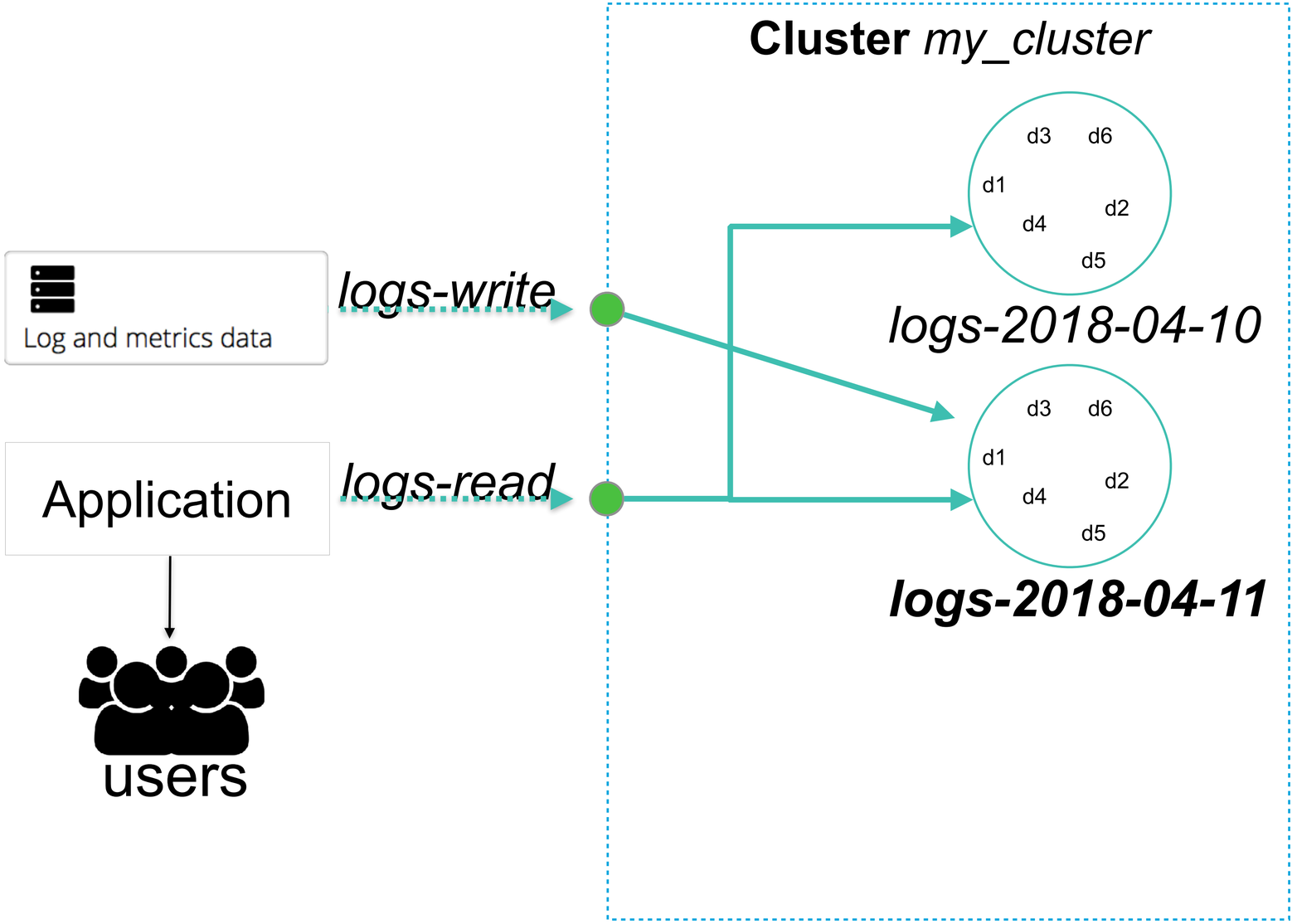
```
PUT _template/logs
{
  "template": "logs-*",
  "settings": {
    "number_of_shards":   2,
    "number_of_replicas": 1,
    "refresh_interval": "60s"
  }
}
```

More on that later

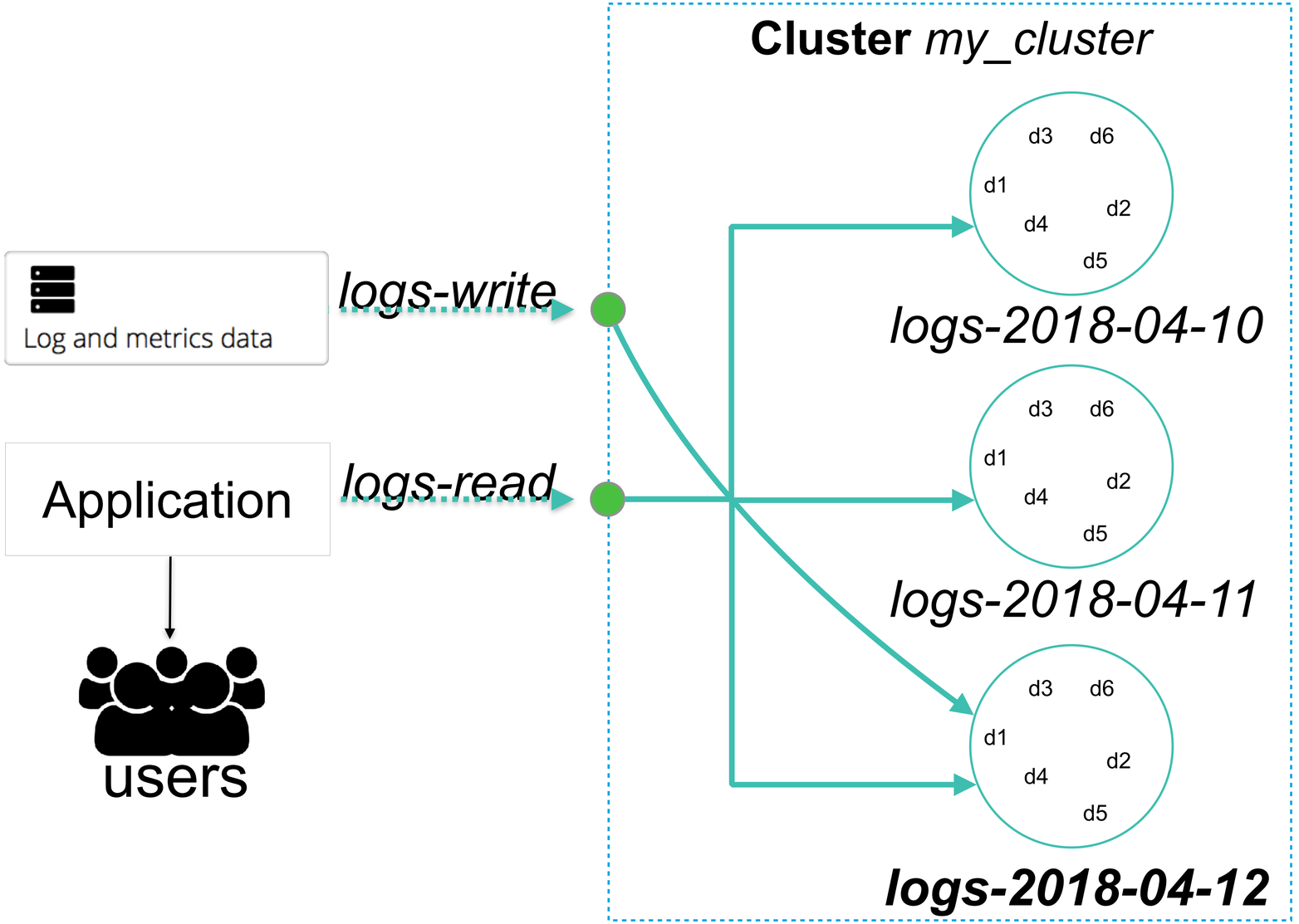
Alias


Cluster my_cluster
d3
d6
d1
d2
d4
d5
logs-2018-04-10
Log and metrics data
logs-write
Application
logs-read
users

Alias


Cluster my_cluster
Log and metrics data
logs-write
Application
logs-read
users
d3
d6
d1
d2
d4
d5
logs-2018-04-10
d3
d6
d1
d2
d4
d5
logs-2018-04-11

Alias


Cluster my_cluster
Log and metrics data
logs-write
Application
logs-read
users
d3
d6
d1
d2
d4
d5
logs-2018-04-10
logs-2018-04-11
logs-2018-04-12

Detour: Rollover API

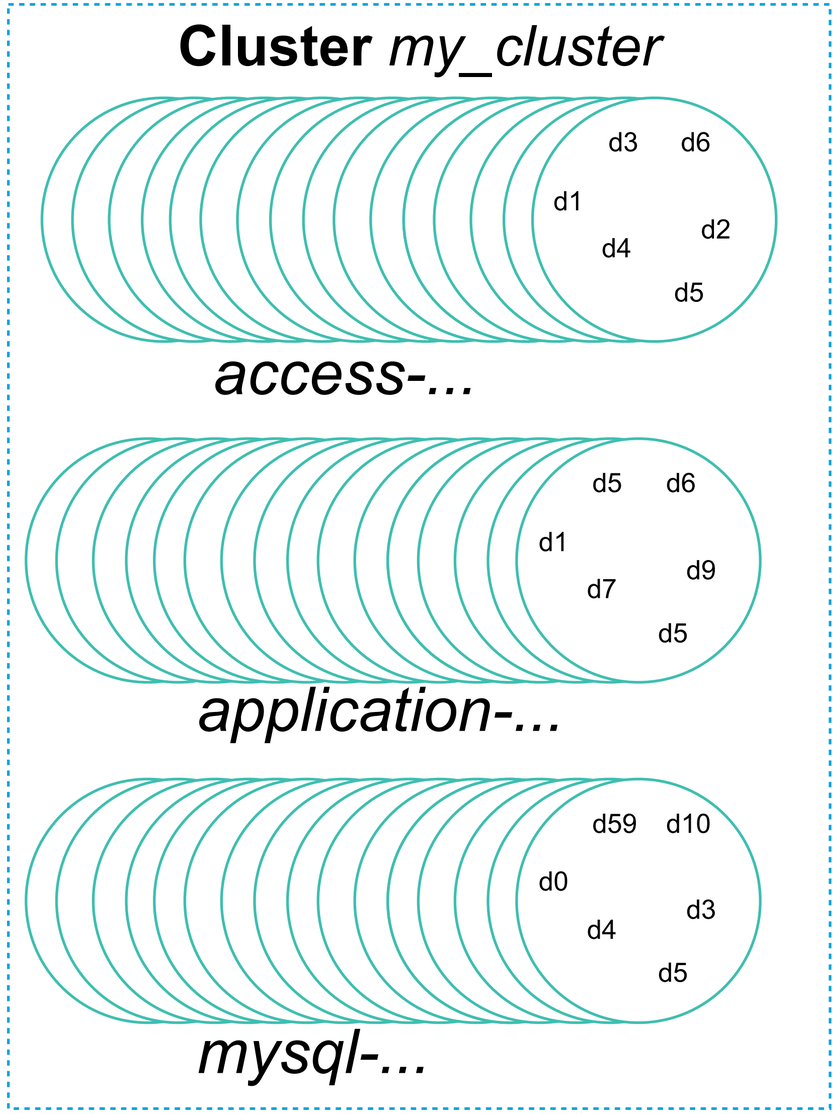
https://www.elastic.co/guide/en/elasticsearch/reference/6.3/indices-rollover-index.html

elastic

Do not Overshard

don't keep default values!

- 3 different logs
- 1 index per day each
- 1GB each
- 5 shards (default): so 200mb / shard vs 45gb
- 6 months retention
- **~900 shards for ~180GB**
- **we needed ~4 shards!**



Scaling

But what happens if we have 2M users?

Scaling

Scaling

Scaling

Shards are the working unit

- Primaries
  - More data -> More shards
  - **write throughput** (More writes -> **More primary shards**)
- Replicas
  - high availability (1 replica is the default)
  - **read throughput** (More reads -> **More replicas**)

Detour: Shrink API

https://www.elastic.co/guide/en/elasticsearch/reference/6.3/indices-shrink-index.html

elastic

Detour: Split API

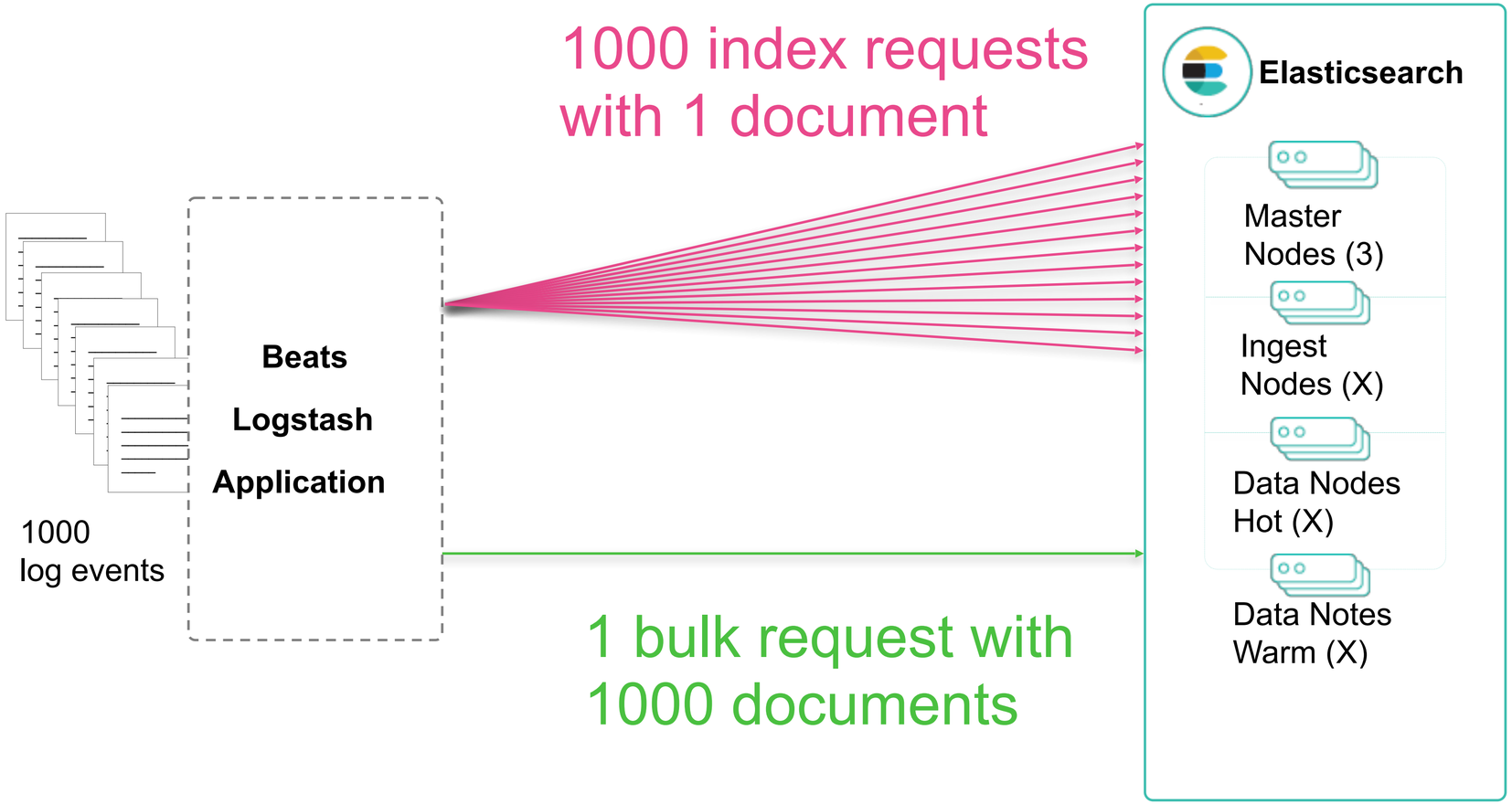
https://www.elastic.co/guide/en/elasticsearch/reference/6.3/indices-split-index.html

elastic

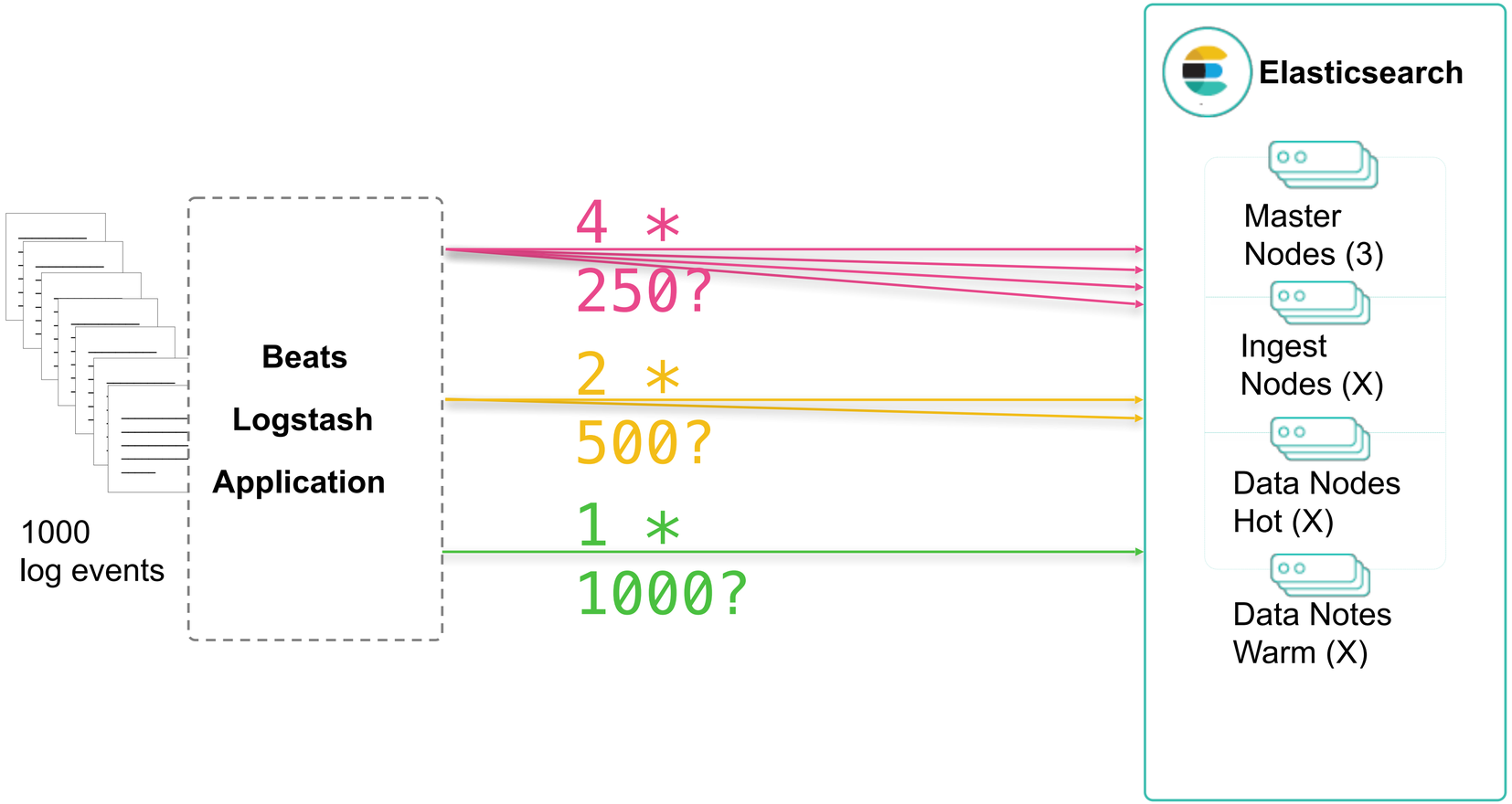
Optimal Bulk Size

elastic

What is Bulk?


1000 index requests
with 1 document
Elasticsearch
Master
Nodes (3)
Ingest
Nodes (X)
Data Nodes
Hot (X)
Data Notes
Warm (X)
Beats
Logstash
Application
1000
log events
1 bulk request with
1000 documents

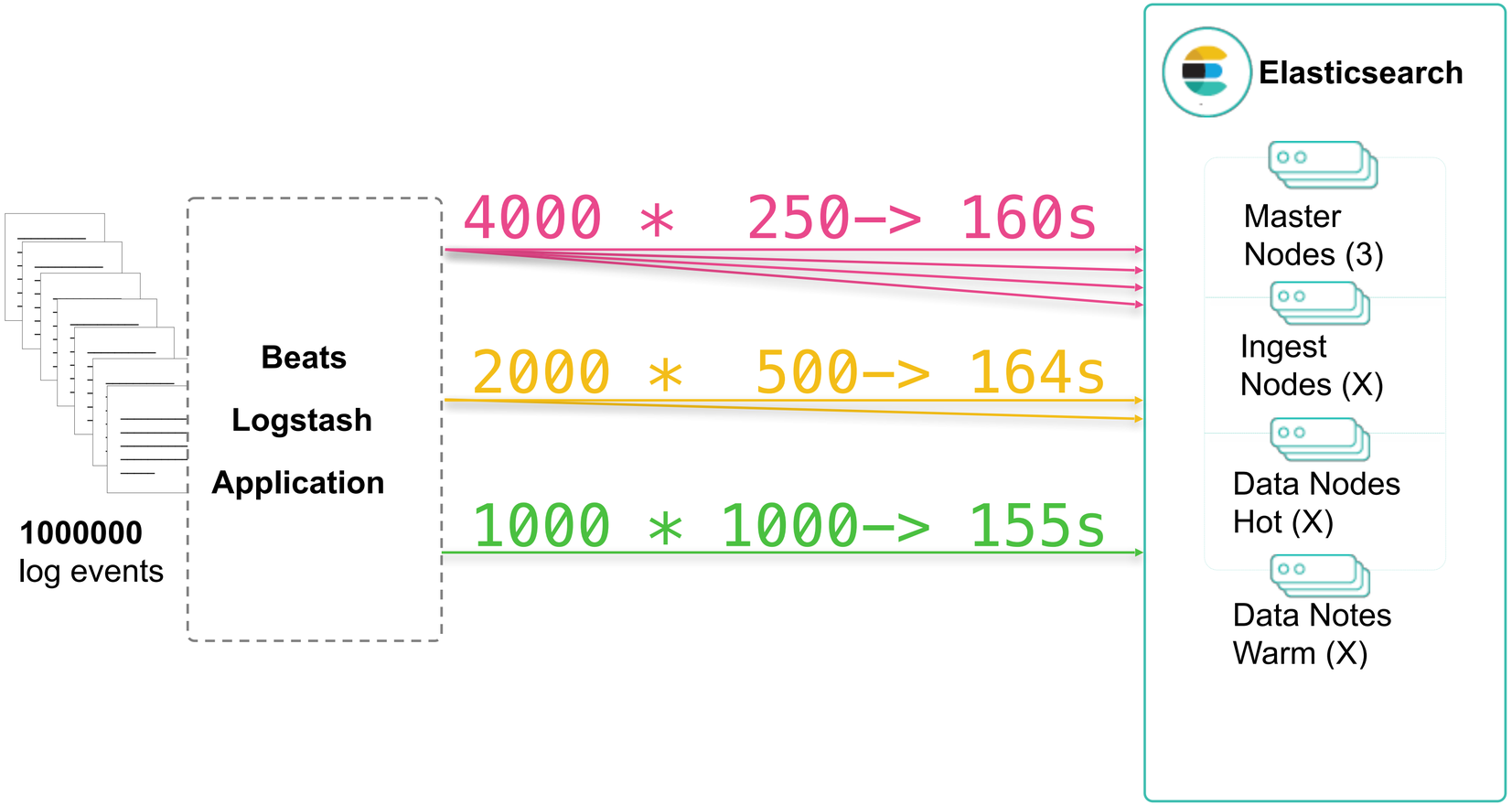
What is the optimal bulk size?



It depends...

- on your application (language, libraries, ...)
- document size (100b, 1kb, 100kb, 1mb, ...)
- number of nodes
- node size
- number of shards
- shards distribution

Test it ;)


Elasticsearch
Master Nodes (3)
Ingest Nodes (X)
Data Nodes Hot (X)
Data Notes Warm (X)
4000 * 250–> 160s
2000 * 500–> 164s
1000 * 1000–> 155s
Beats
Logstash
Application
1000000
log events

Test it ;)

```
input { stdin{} }

filter {}

output {
  elasticsearch {
    hosts => ["10.12.145.189"]
    flush_size => "${SIZE}"
} }
```

In Beats set "bulk_max_size"
in the output.elasticsearch

```
DATE=`date +%Y.%m.%d`
LOG=logs/logs.txt

exec_test () {
  curl -s -XDELETE "http://USER:PASS@HOST:9200/logstash-$DATE"
  sleep 10
  export SIZE=$1
  time cat $LOG | ./bin/logstash -f logstash.conf
}

for SIZE in 100 500 1000 3000 5000 10000; do
  for i in {1..20}; do
    exec_test $SIZE
done; done;
```

Test it ;)

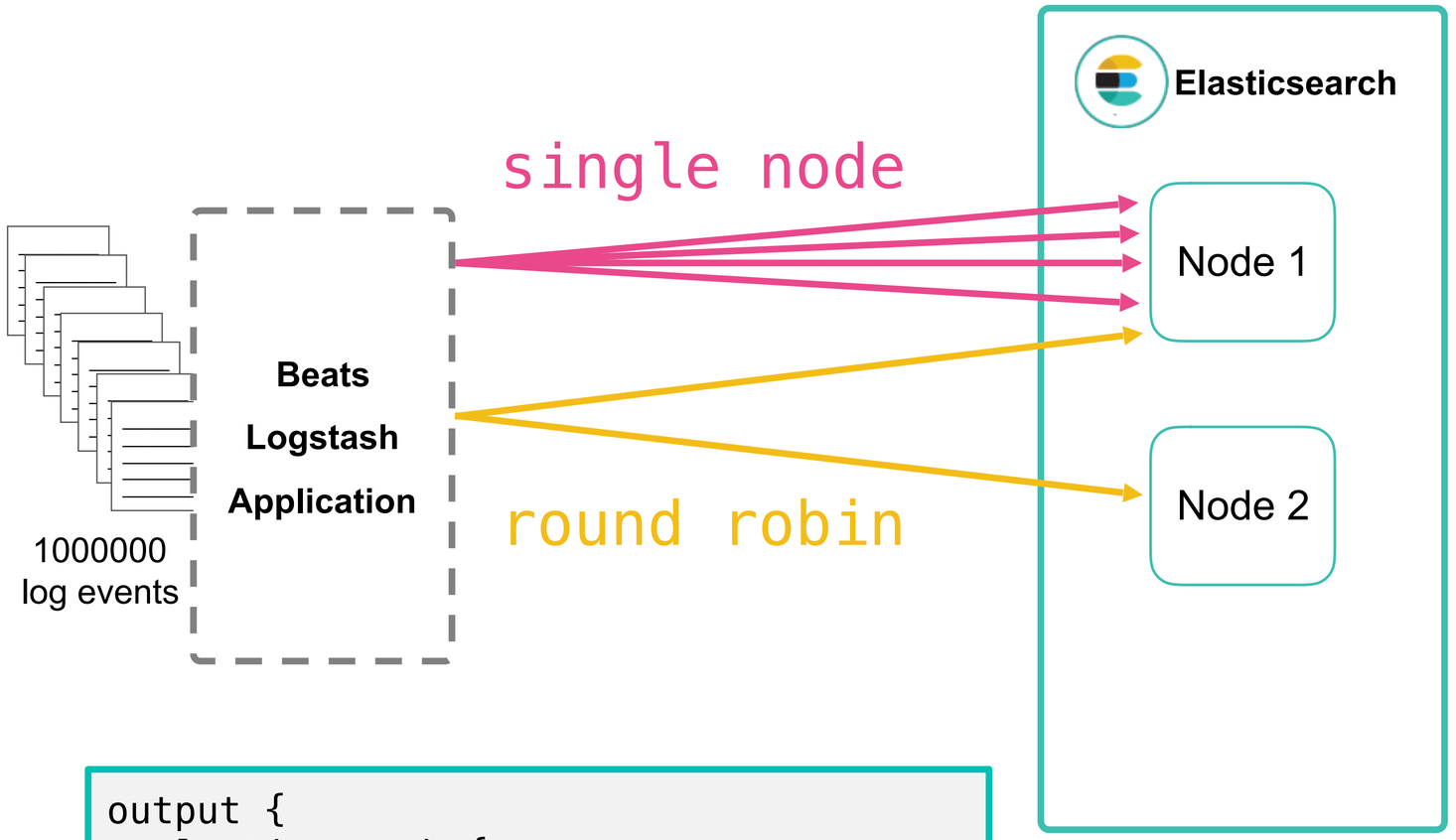
- 2 node cluster (m3.large)
  - 2 vCPU, 7.5GB Memory, 1x32GB SSD
- 1 index server (m3.large)
  - logstash
  - kibana

# docs	100	500	1000	3000	5000	10000
time(s)	191.7	161.9	163.5	160.7	**160.7**	161.5

Distribute the Load

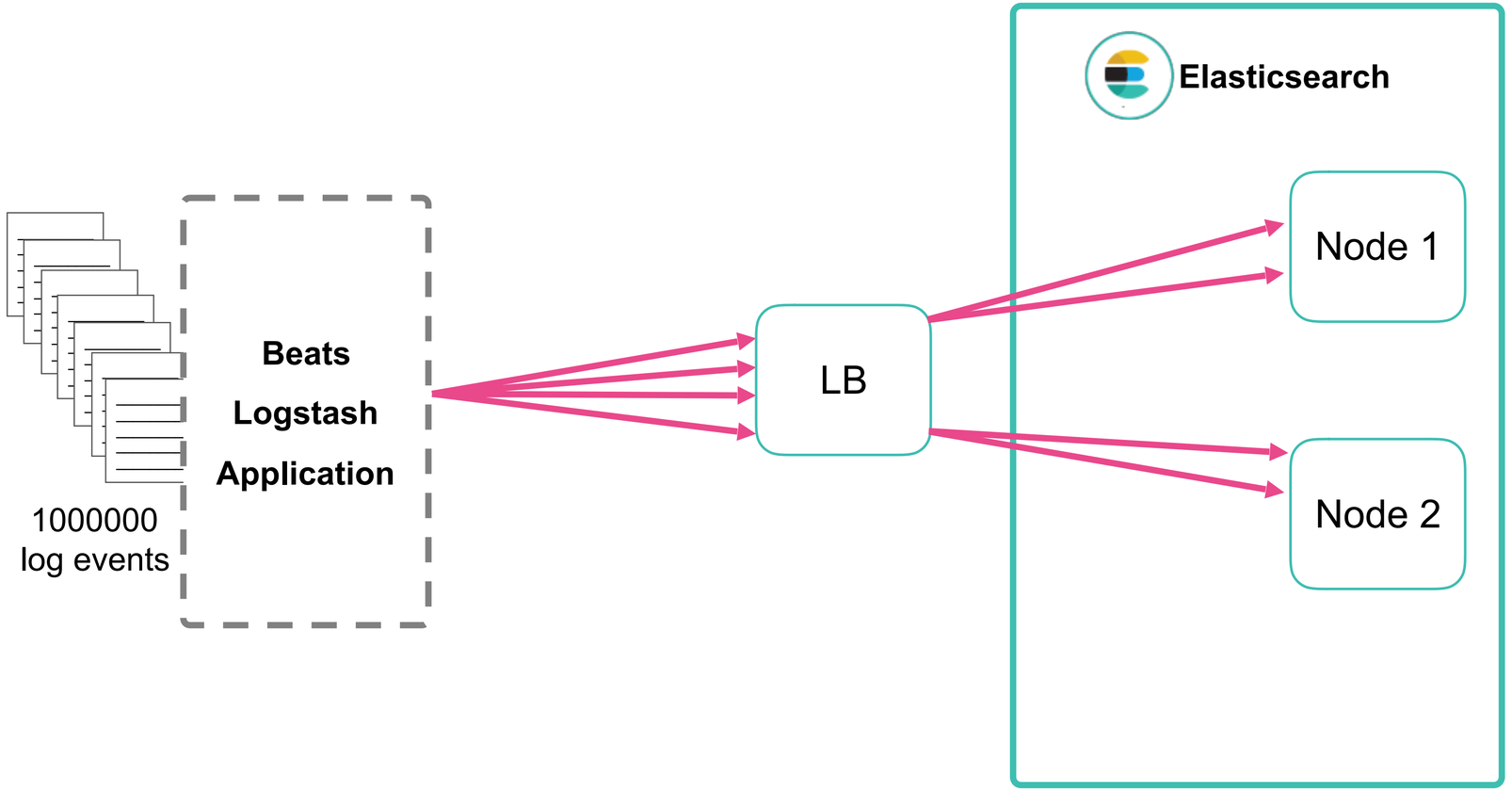
elastic

Avoid Bottlenecks

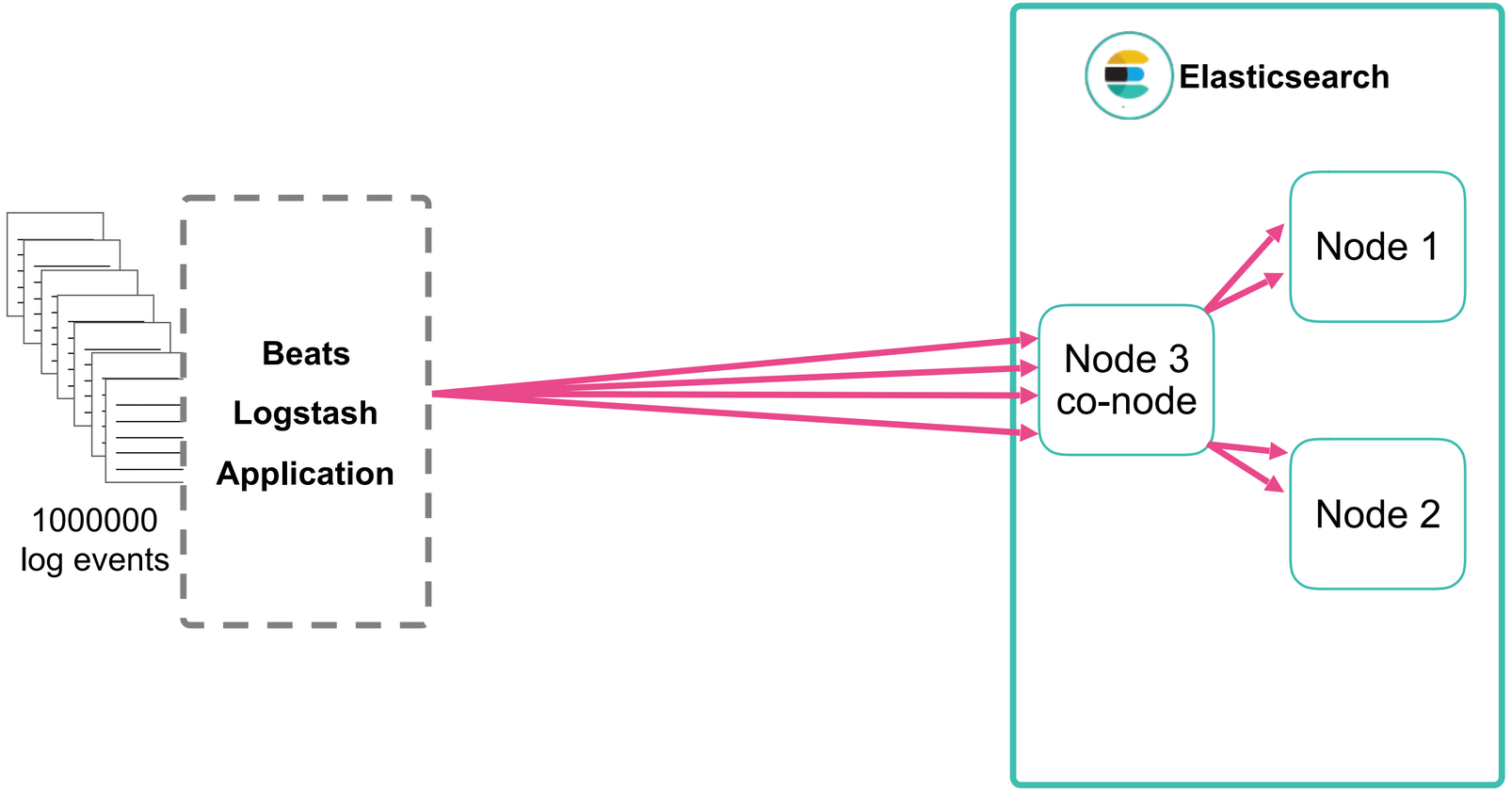



```
output {
  elasticsearch {
    hosts => ["node1","node2"]
} }
```

Load Balancer



Coordinating-only Node



Test it ;)

- 2 node cluster (m3.large)
  - 2 vCPU, 7.5GB Memory, 1x32GB SSD
- 1 index server (m3.large)
  - logstash (round robin configured)
    - `hosts => ["10.12.145.189", "10.121.140.167"]`
  - kibana

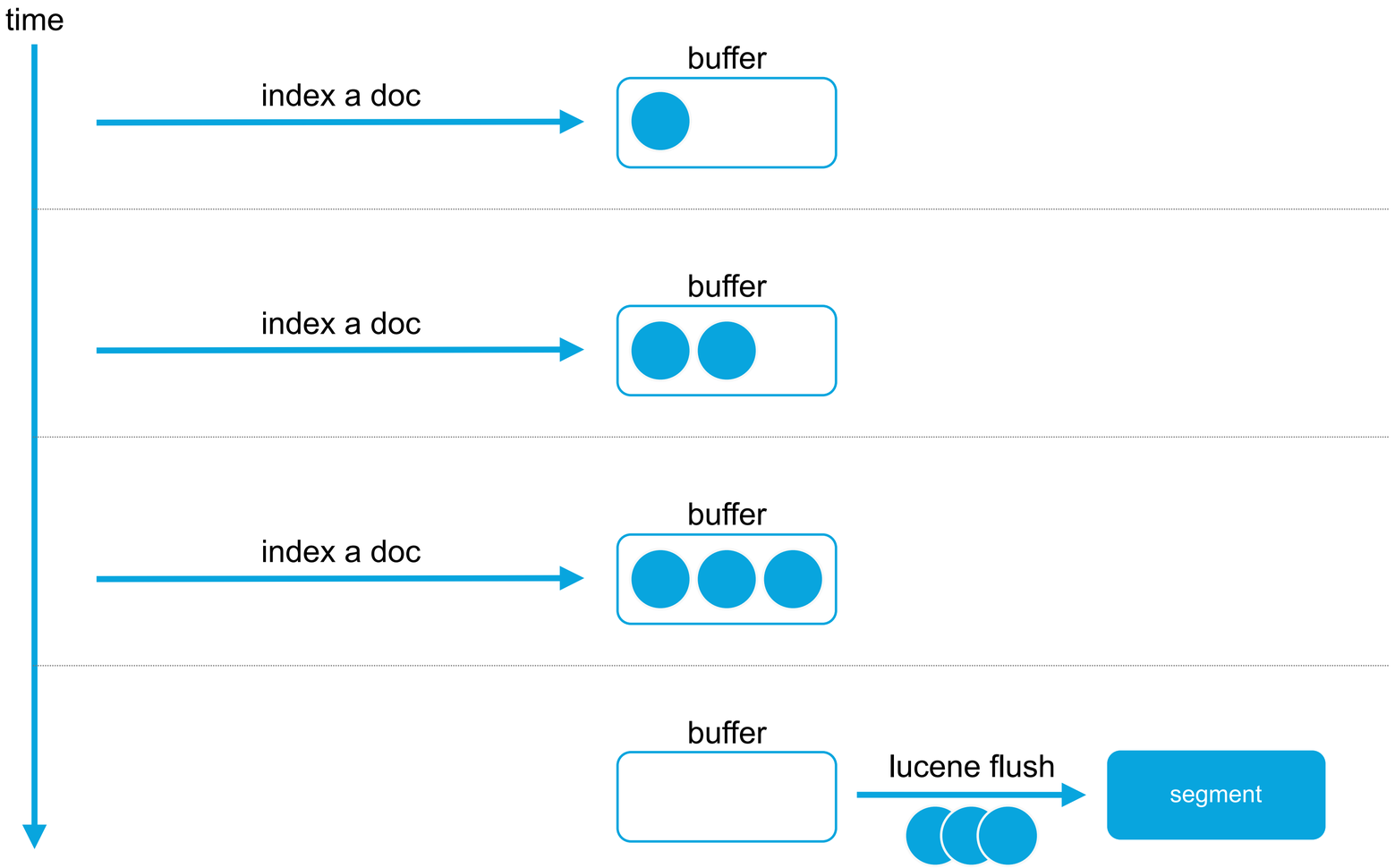
#docs
time(s)
	1000	5000	10000

NO Round Robin	163.5	160.7	161.5
Round Robin	161.3	158.2	159.4

Optimizing Disk IO

elastic

Durability


time
index a doc
buffer
index a doc
buffer
index a doc
buffer
buffer
lucene flush
segment

refresh_interval

- Dynamic per-index setting

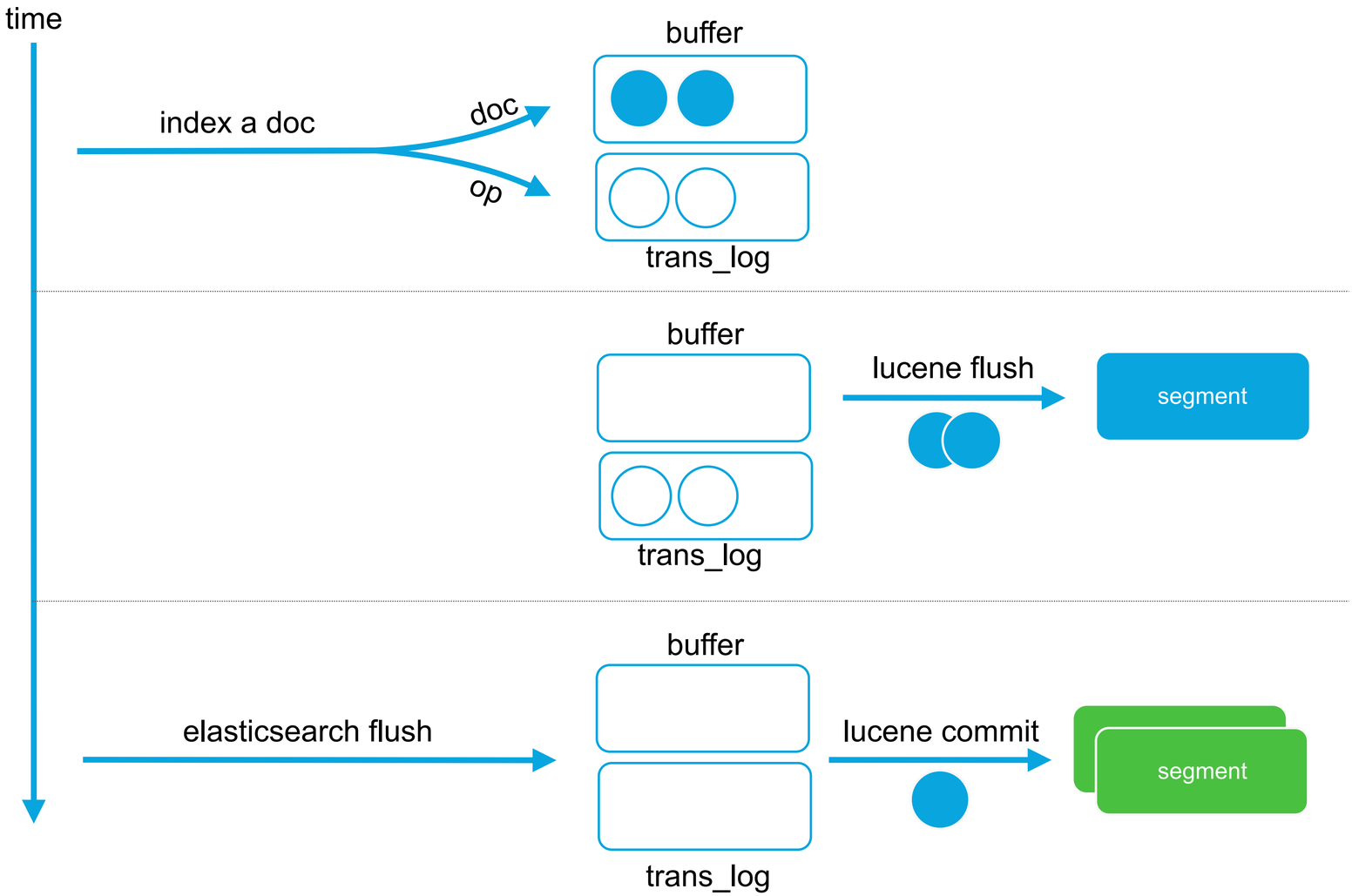
```
PUT logstash-2017.05.16/_settings
{
  "refresh_interval": "60s"
}
```

- Increase to get better write throughput to an index
- New documents will take more time to be available for Search.

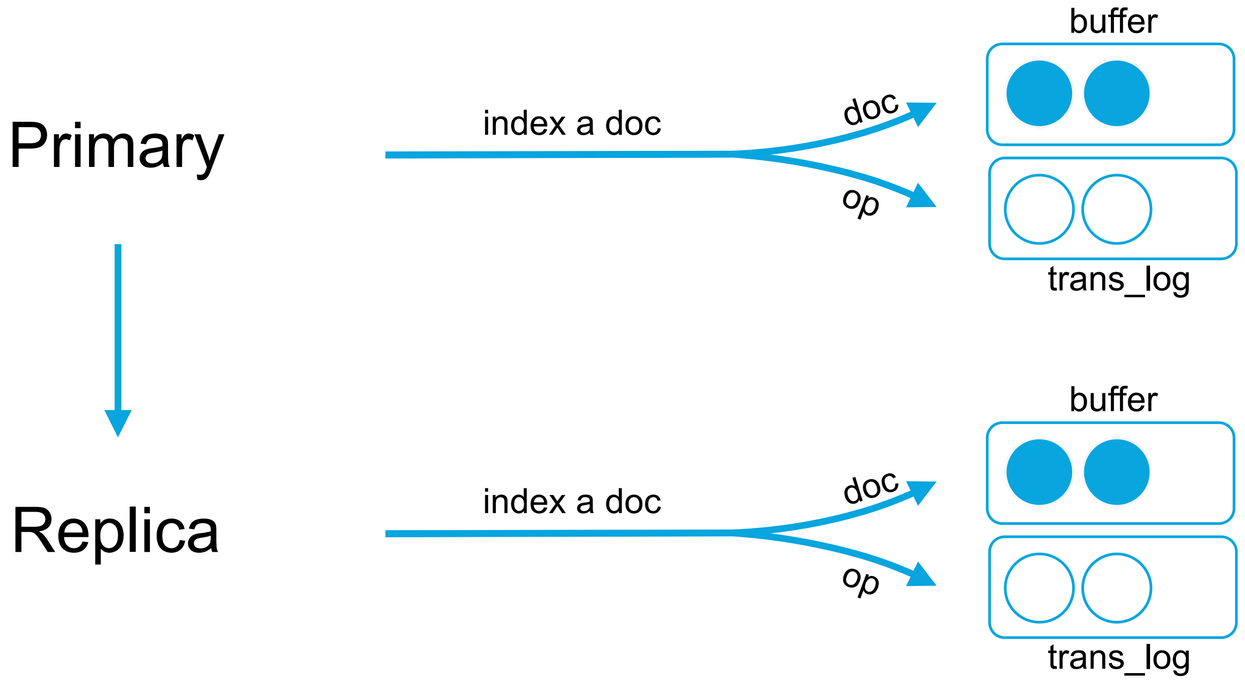
#docs
time(s)
	1000	5000	10000

1s refresh	161.3	158.2	159.4
60s refresh	156.7	152.1	152.6

Durability


time
index a doc
doc
op
buffer
trans_log
buffer
lucene flush
segment
trans_log
buffer
elasticsearch flush
lucene commit
segment
trans_log

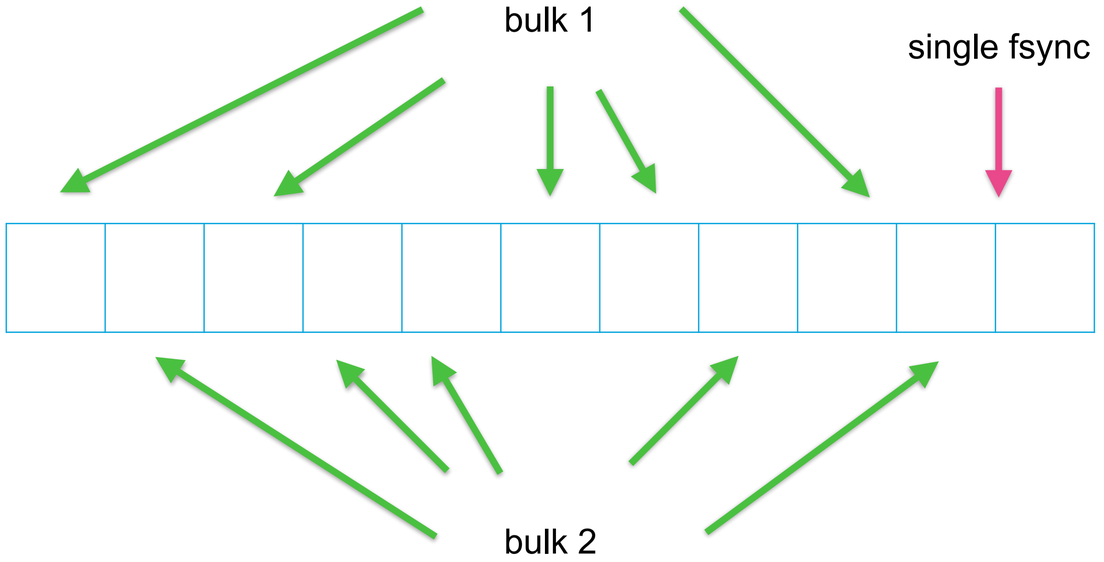
Translog fsync every 5s (1.7)



redundancy doesn’t help if all nodes lose power

Translog fsync on every request

- For low volume indexing, fsync matters less
- For high volume indexing, we can amortize the costs and fsync on every bulk
- Concurrent requests can share an fsync

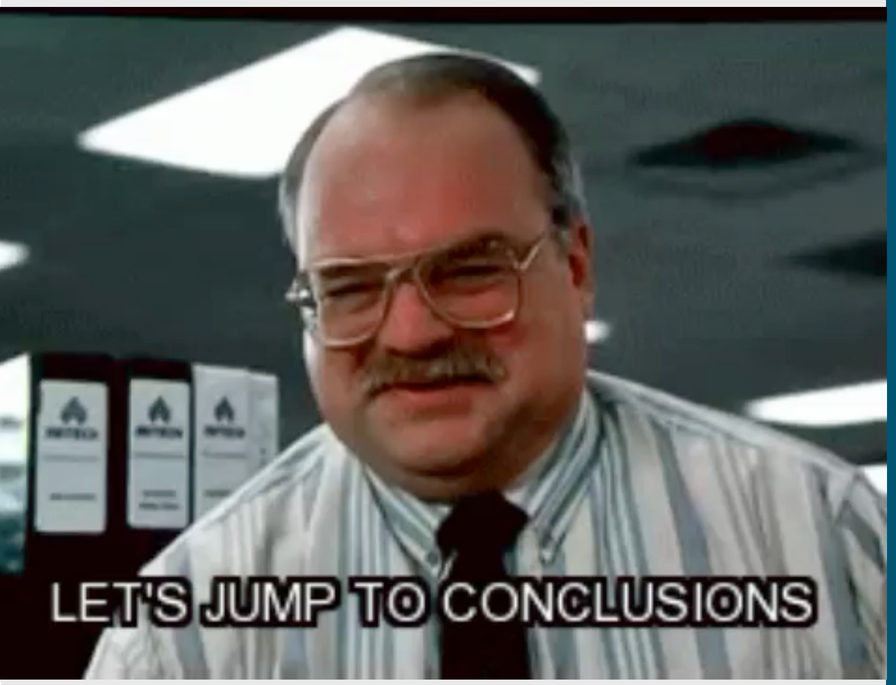


Async Transaction Log

- index.translog.durability
  - request (default)
  - async
- index.translog.sync_interval (only if async is set)
- Dynamic per-index settings
- **Be careful, you are relaxing the safety guarantees**

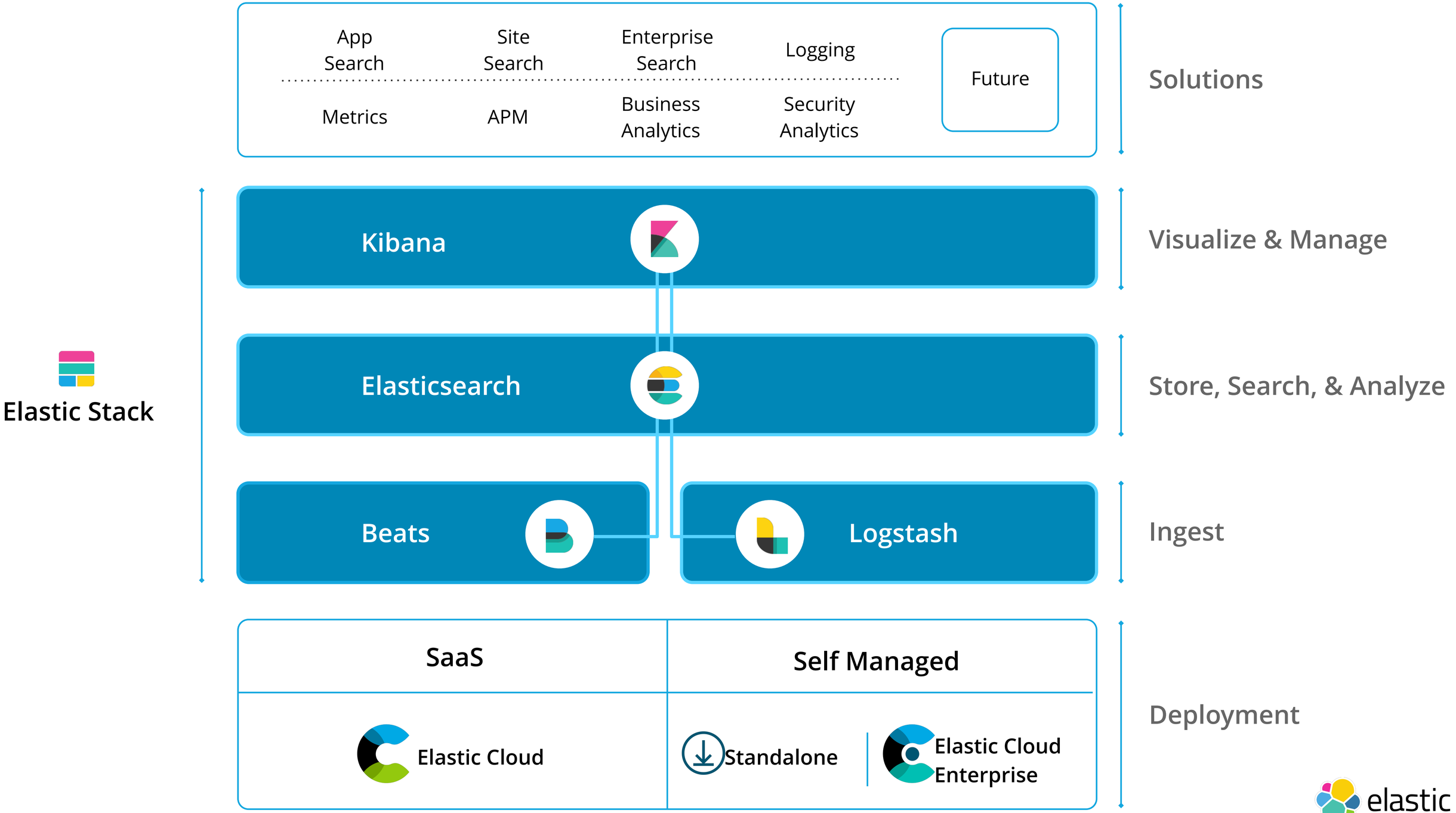
#docs
time(s)
	1000	5000	10000

Request fsync	161.3	158.2	159.4
5s sync	152.4	149.1	150.3



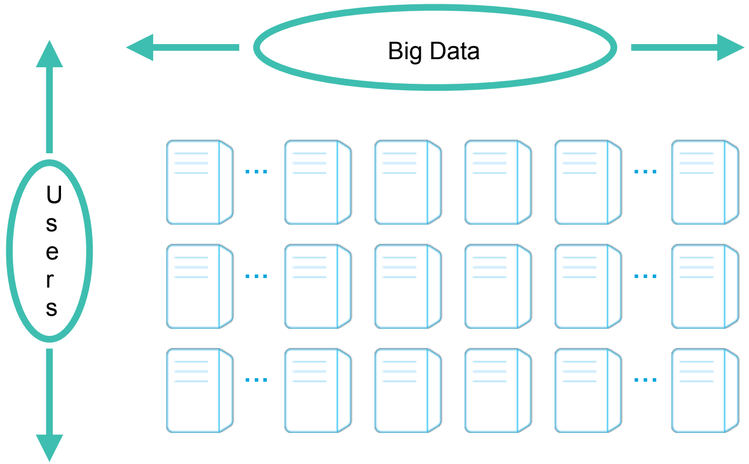
Final Remarks

elastic


App Search
Site Search
Enterprise Search
Logging
Future
Metrics
APM
Business Analytics
Security Analytics
Solutions
Elastic Stack
Kibana
Visualize & Manage
Elasticsearch
Store, Search, & Analyze
Beats
Logstash
Ingest
SaaS
Self Managed
Elastic Cloud
Standalone
Elastic Cloud Enterprise
Deployment
elastic

Final Remarks

- Primaries
  - More data -> More shards
  - Do not overshard!
- Replicas
  - high availability (1 replica is the default)
  - read throughput (More reads -> More replicas)



Final Remarks

- Bulk and Test
- Distribute the Load
- Refresh Interval
- Async Trans Log (**careful**)

#docs per bulk	1000	5000	10000
Default	163.5	**160.7**	161.5
RR+60s+Async5s	152.4	149.1	150.3

elastic

Les vendredi noirs ?

Même pas peur !

David Pilato
Developer | Evangelist, @dadoonet

DevOpsDays Geneva