



# Elastic Stack: Past, Present, & Future

---

Medcl  
Elastic



# About me

- 曾勇 ( Medcl )
- Elastic Developer/Evangelist
- Creator of Elastic China Community
- Github
  - <http://github.com/medcl>
- Twitter/Weibo
  - @medcl



elastic

# Past

The history of Elastic Stack

# History of Elasticsearch

- In 2004, Shay Banon developed a product called **Compass**
- The need for ***scalability*** became a top priority
- In 2010, Shay completely rewrote Compass with two main objectives:
  - 1. ***distributed from the ground up in its design***
  - 2. ***easily used by any other programming language***



- He called it ***Elasticsearch***
- He also start a company around Elasticsearch, named **Elastic**
- Today Elasticsearch is the most popular enterprise search engine

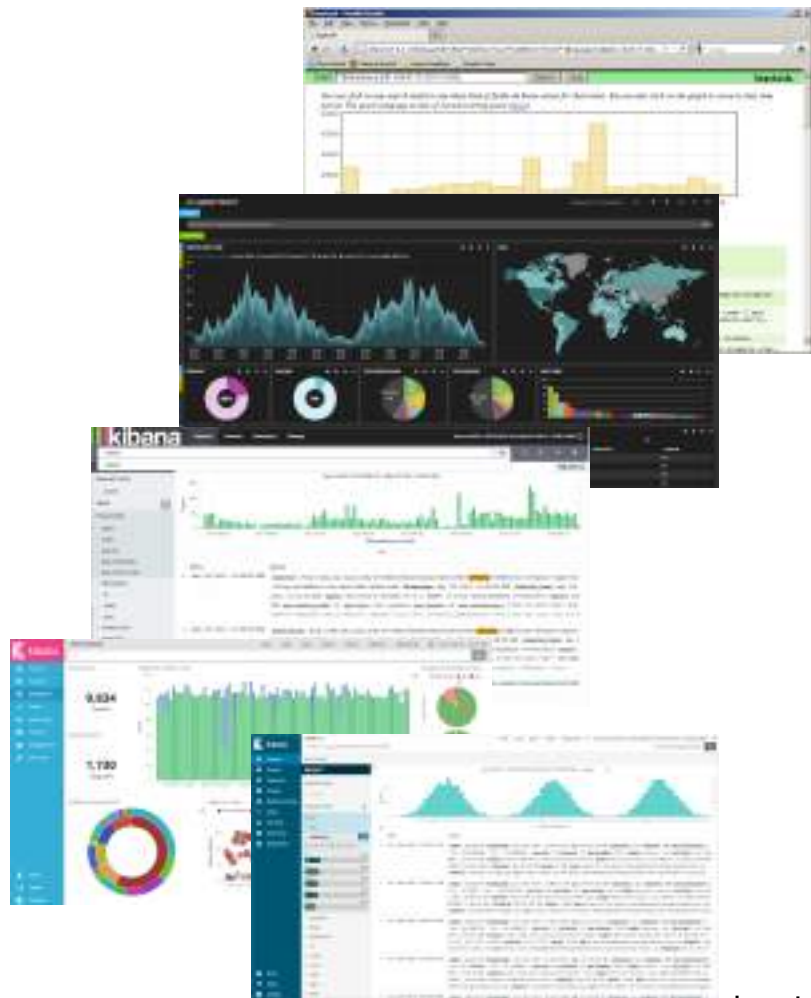


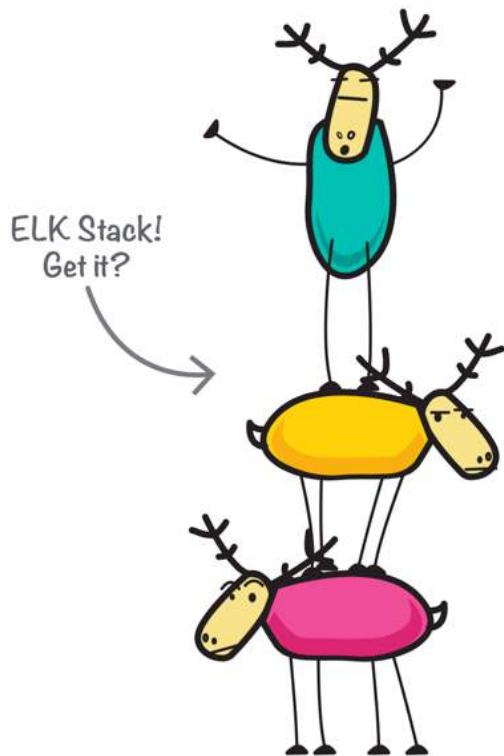
# Milestone of Elasticsearch

- **0.4:** first version was released in February, 2010
  - Distributed、 RESTful API、 Full Text Search、 Facet、 Geolocation
- **1.0:** released in January, 2014
  - Aggregations、 Tribe node、 Doc values、 Circuit breaker
- **2.0:** released in October, 2015
  - Pipeline Aggregations、 Query/Filter merging、 Hardening、 Performance and resilience
- **5.0:** released in October, 2016
  - New data structures、 Painless scripting、 Ingest node、 User friendly
- **6.0:** released in November, 2017

# Timeline

- 2011.5, Logstash 1.0, JRuby
- 2011,12, Kibana 1.0, PHP
- 2012.8, Kibana 2.0, Ruby
- 2013.1 Kibana Join Elastic
- 2013.4, Kibana 3.0, Angularjs
- 2013.8 Logstash Join Elastic
- 2014.10, Kibana 4.0, Nodejs
- 2015.10 Logstash 2.0





**E** Elasticsearch

**L** Logstash

**K** Kibana

# Timeline

- 2015.3, Found join Elastic
- 2015.5, Packetbeat Join Elastic
- 2016.9, Prelert join Elastic
- 2016.10, Elastic Stack release 5.0
- 2017.6, Opbeat join Elastic
- 2017.11, Swiftype join Elastic
- 2017.11, Elastic Stack release 6.0



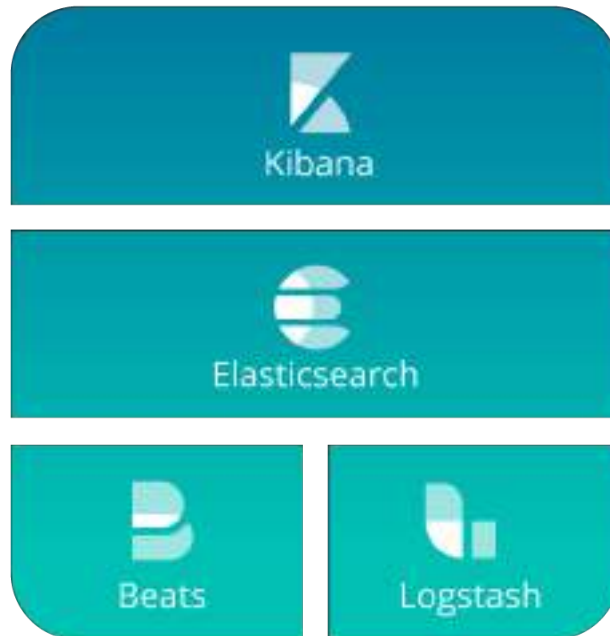


# Release together from 5.0



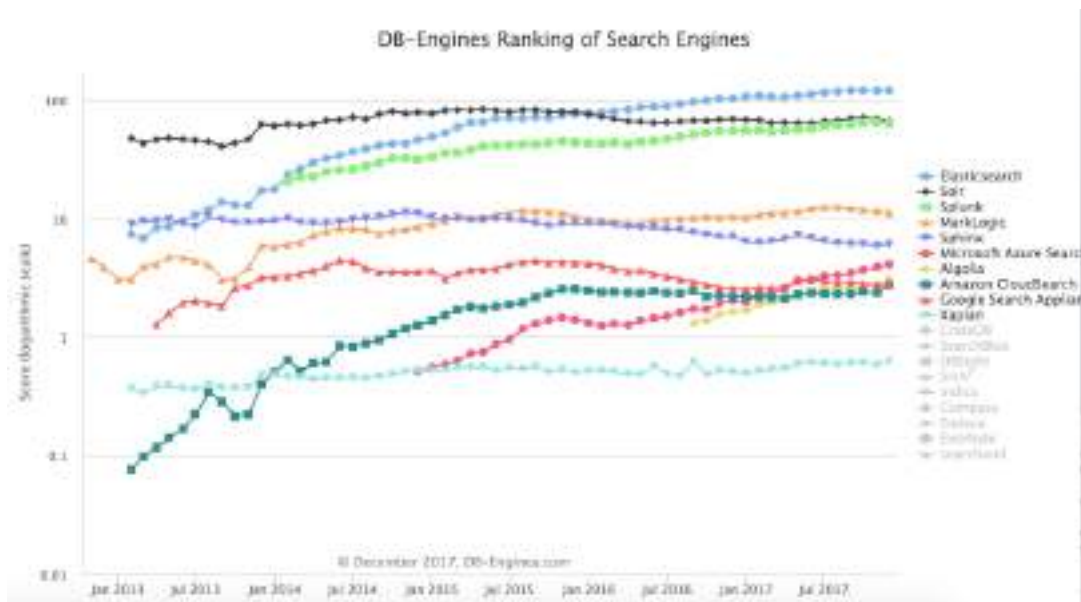
## Elastic Stack

100% open source



# Now, Elastic Stack is used for ...

- Application search
- Enterprise search
- Logging analysis
- Metrics analysis
- Security analysis
- Sentiment analysis
- APM
- ...



# Present

A better Elastic Stack



elasticsearch

# Removal of Type(6.0)

Index

Type

ID

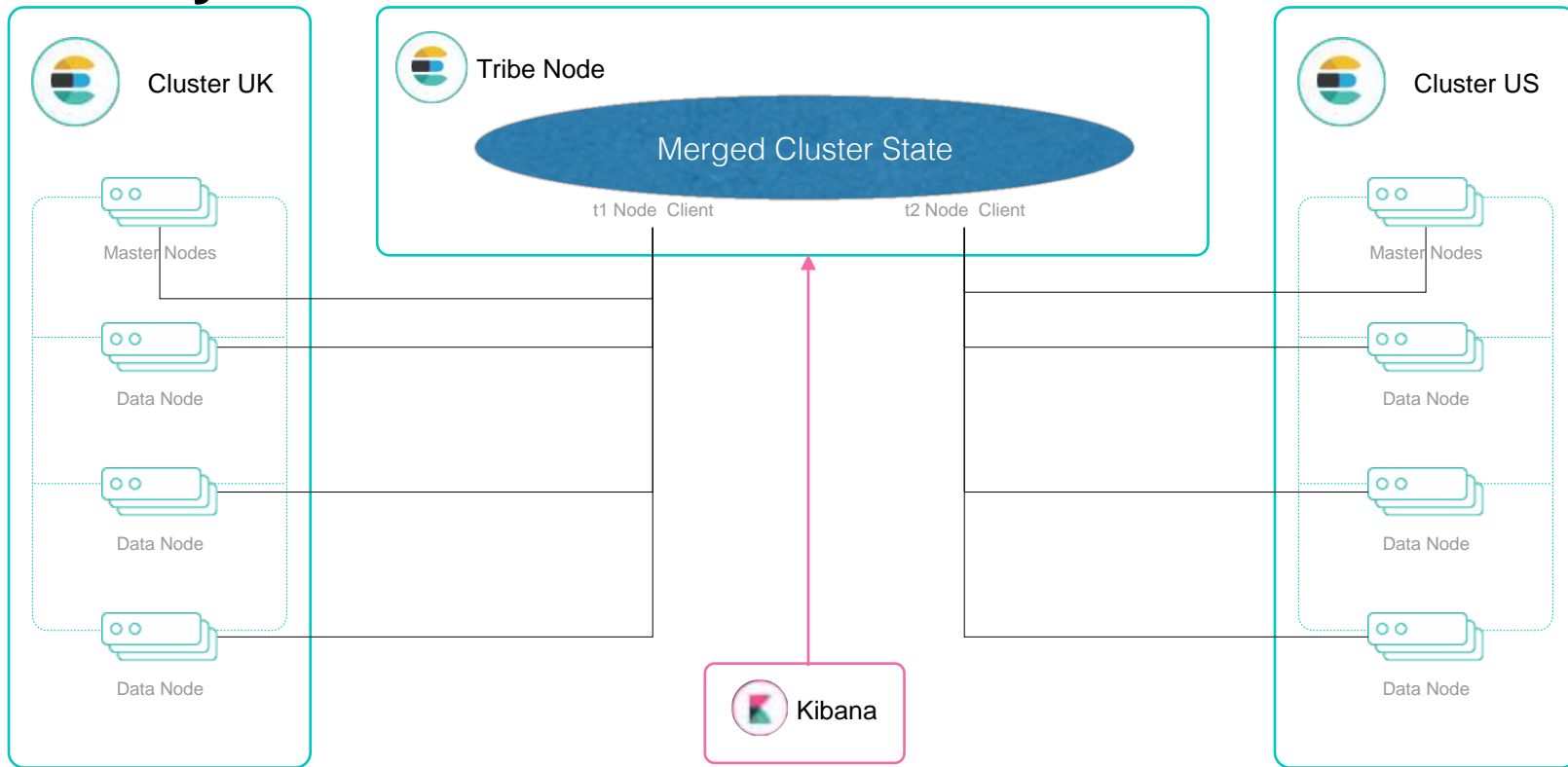
# Removal of Type(6.0)

Index

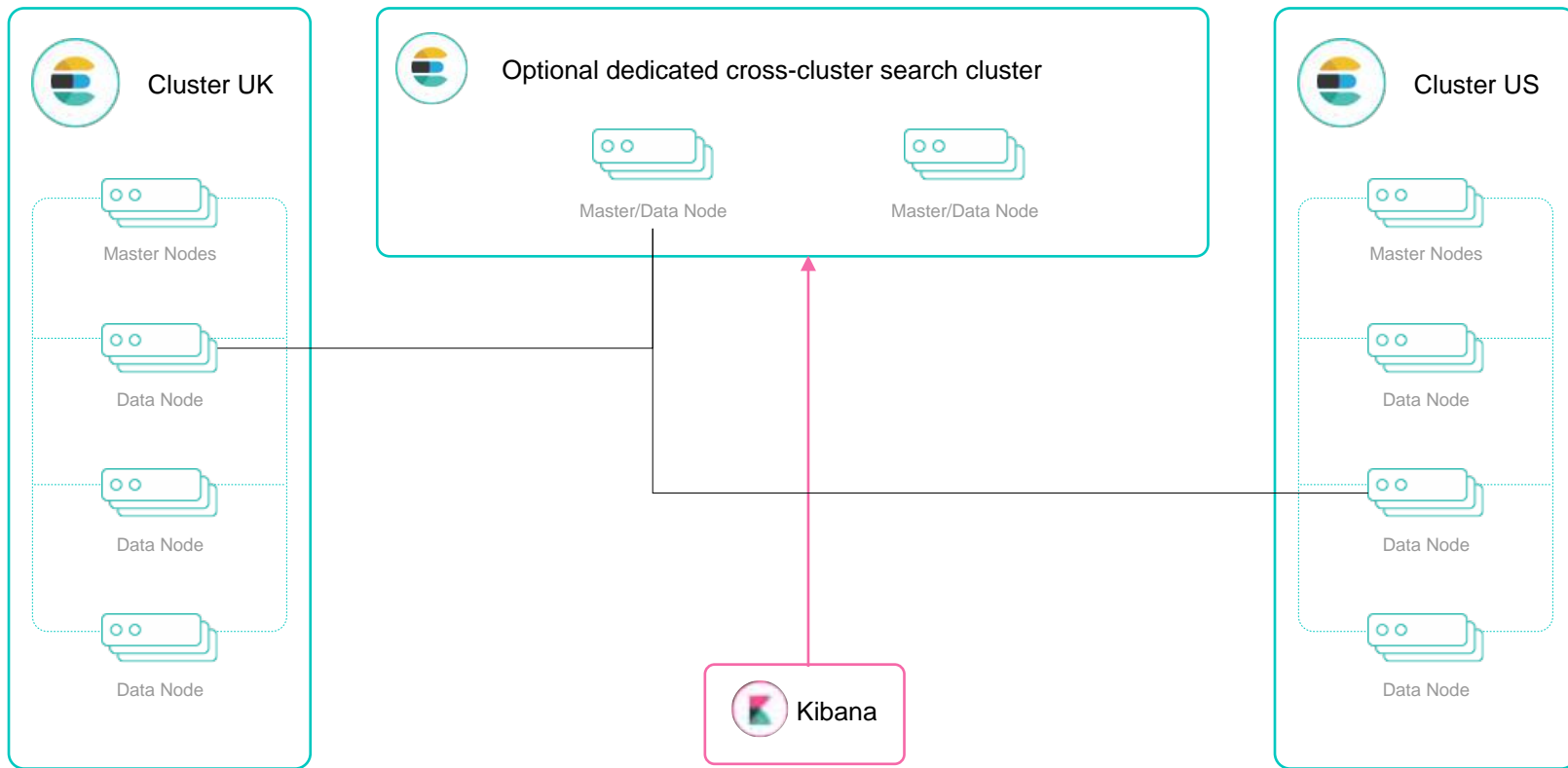
~~Type~~

ID

# Good bye! Tribe Node

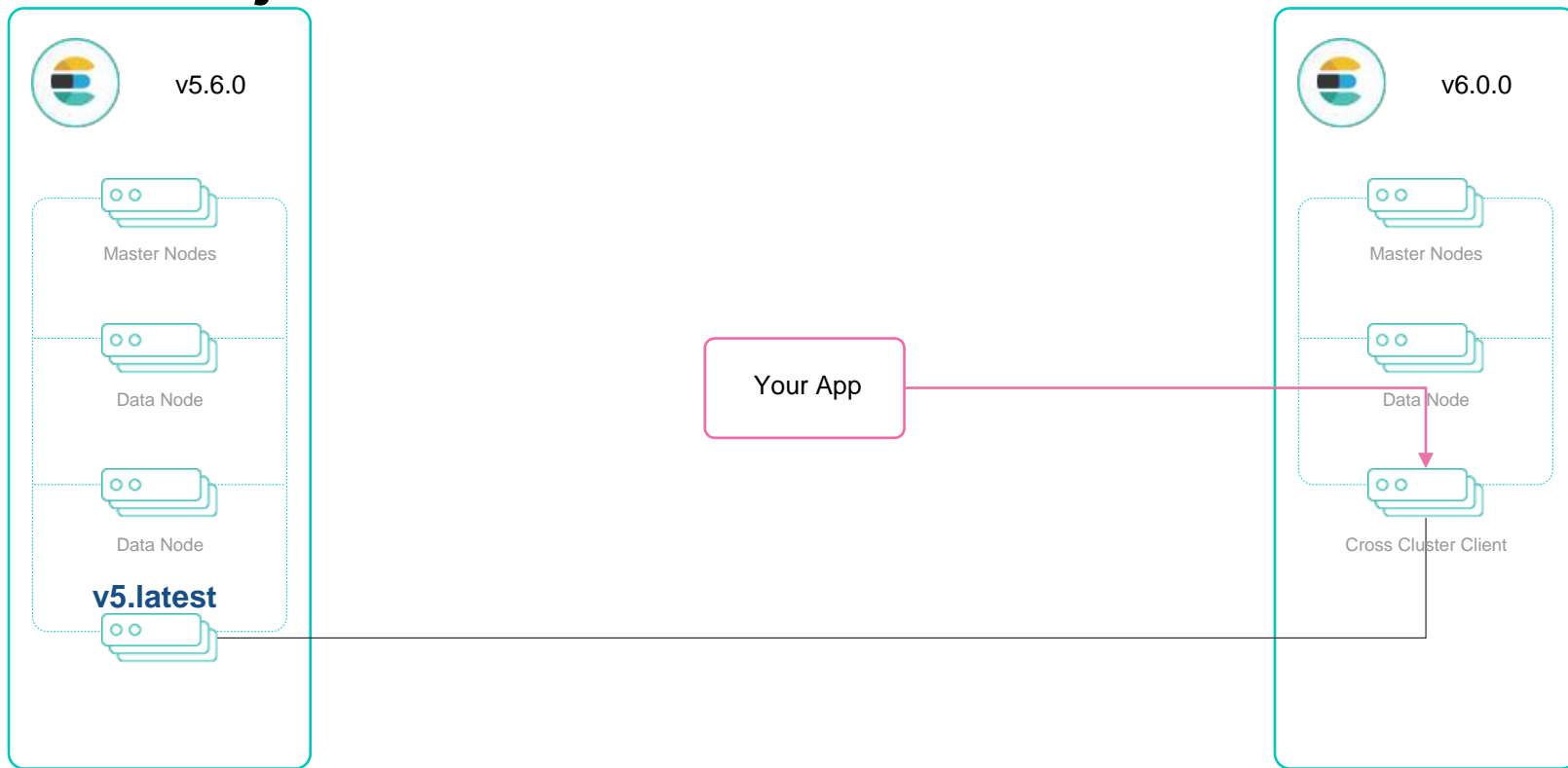


# Hello! Cross-Cluster Search





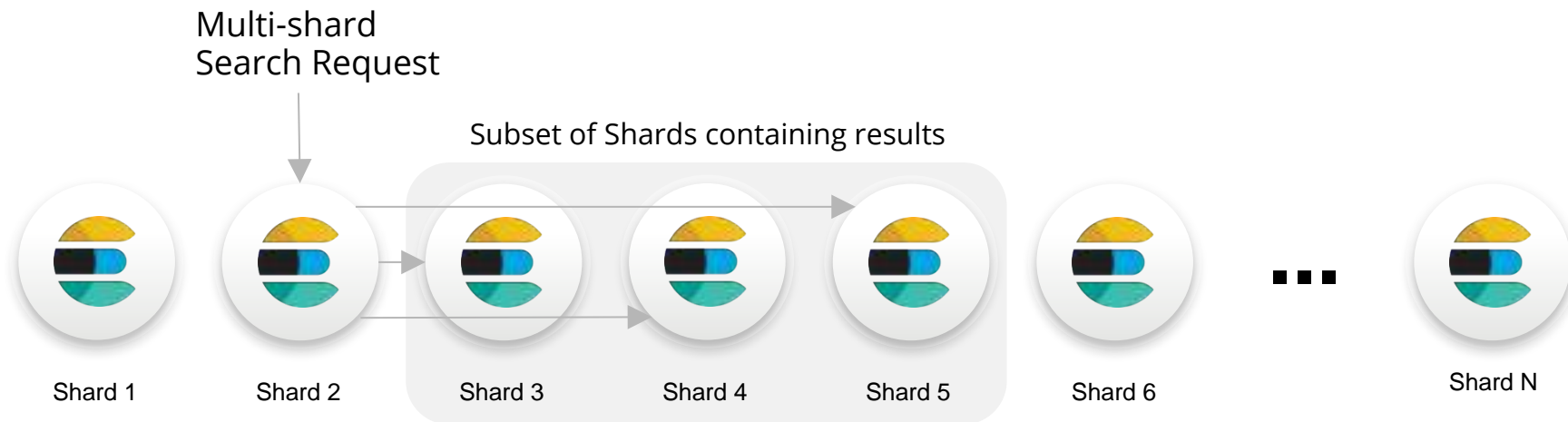
# Cross Major Version Search



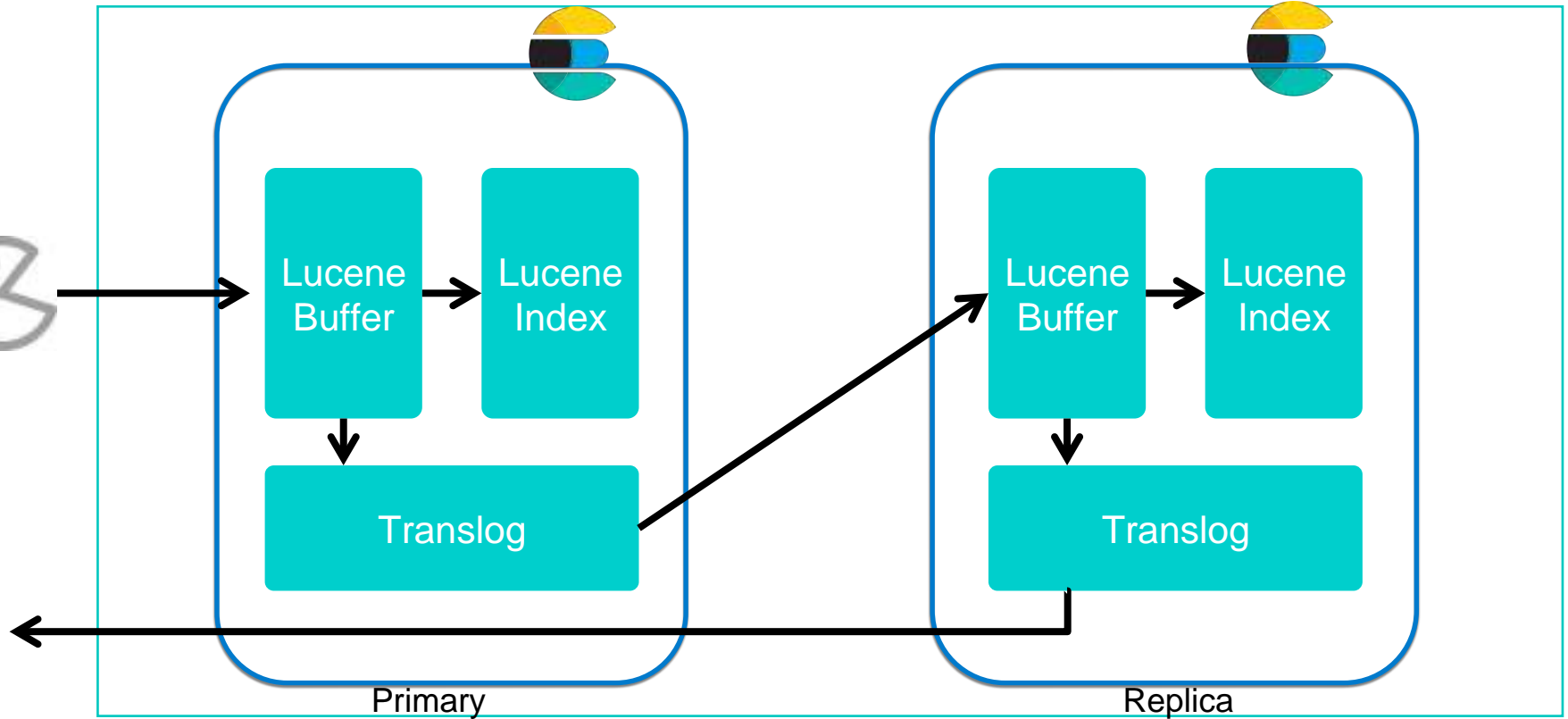
# Improved search scalability

Searches across many shards are more scalable:

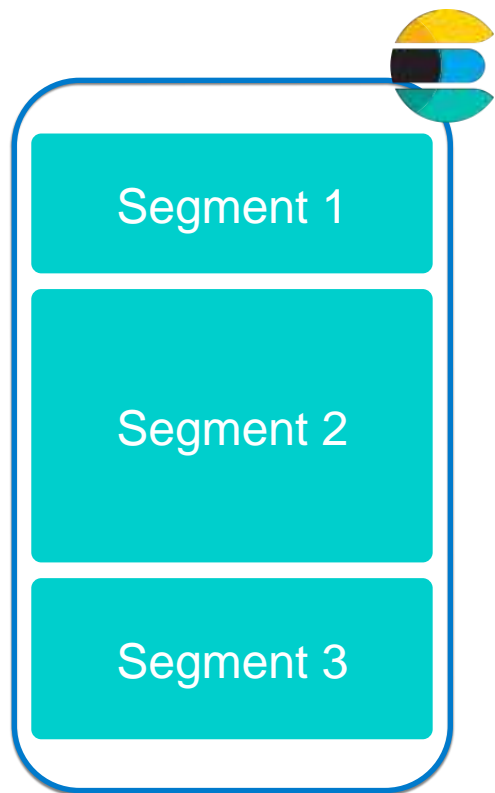
- Fast pre-check phase, exclude any shards that can't match query.
- Batched reduction of results, reduces memory usage on the coordinating node.
- Limits to the number of shards which are searched in parallel, so that a single query cannot dominate the cluster.



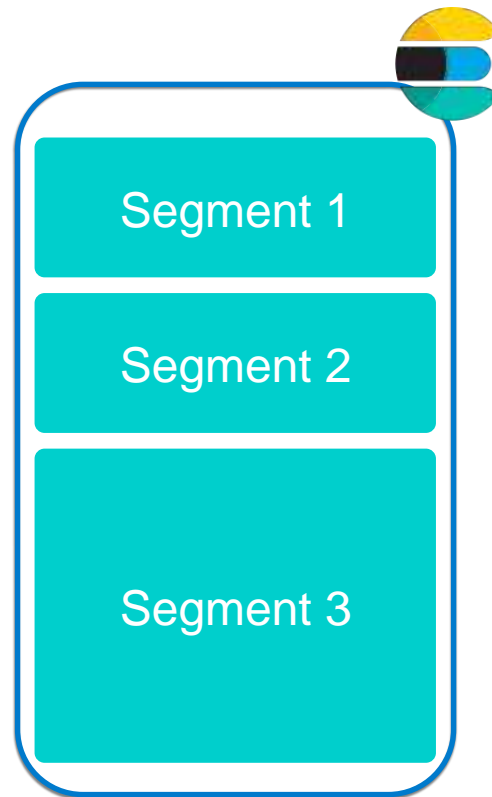
# How replication works



# Recovery (5.x)

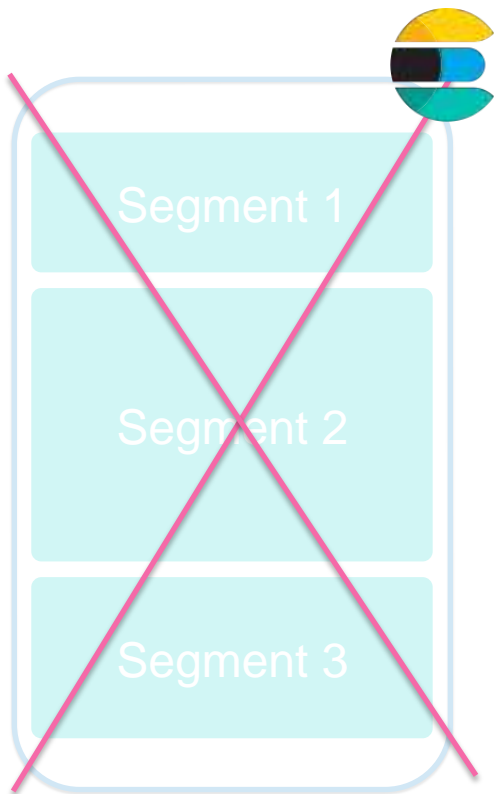


Primary

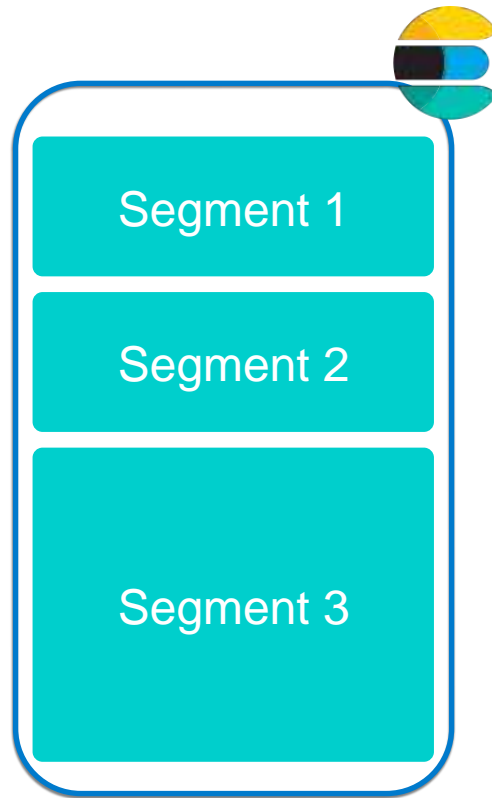


Replica

# Recovery (5.x)

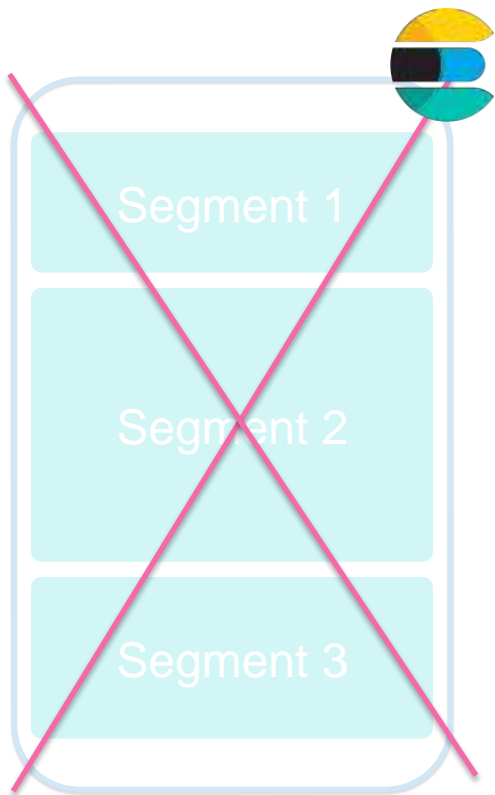


Offline

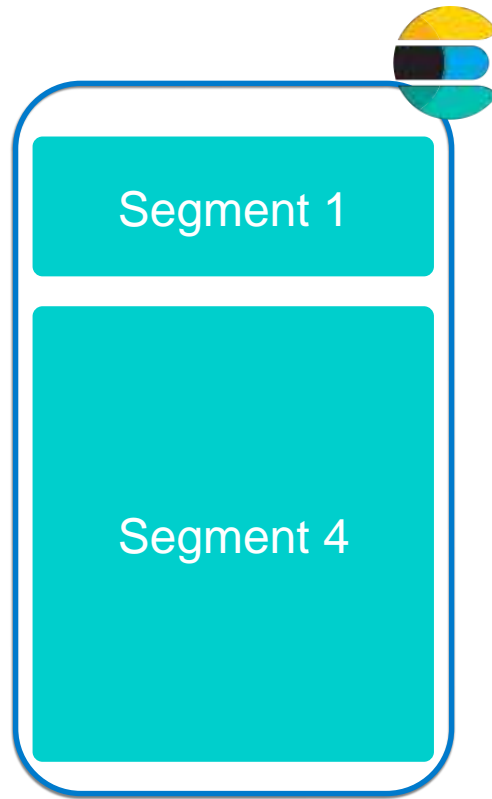


Primary

# Recovery (5.x)

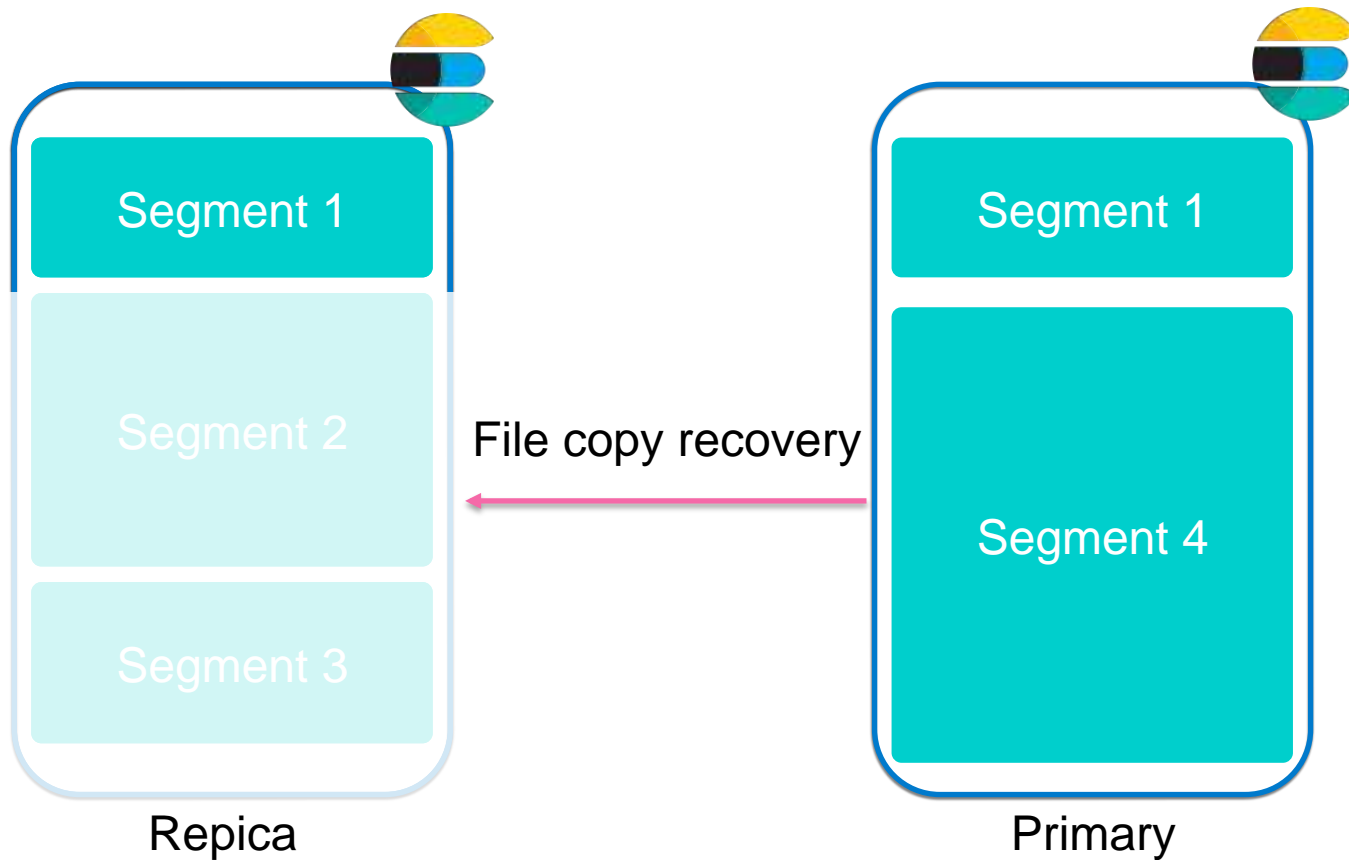


Offline

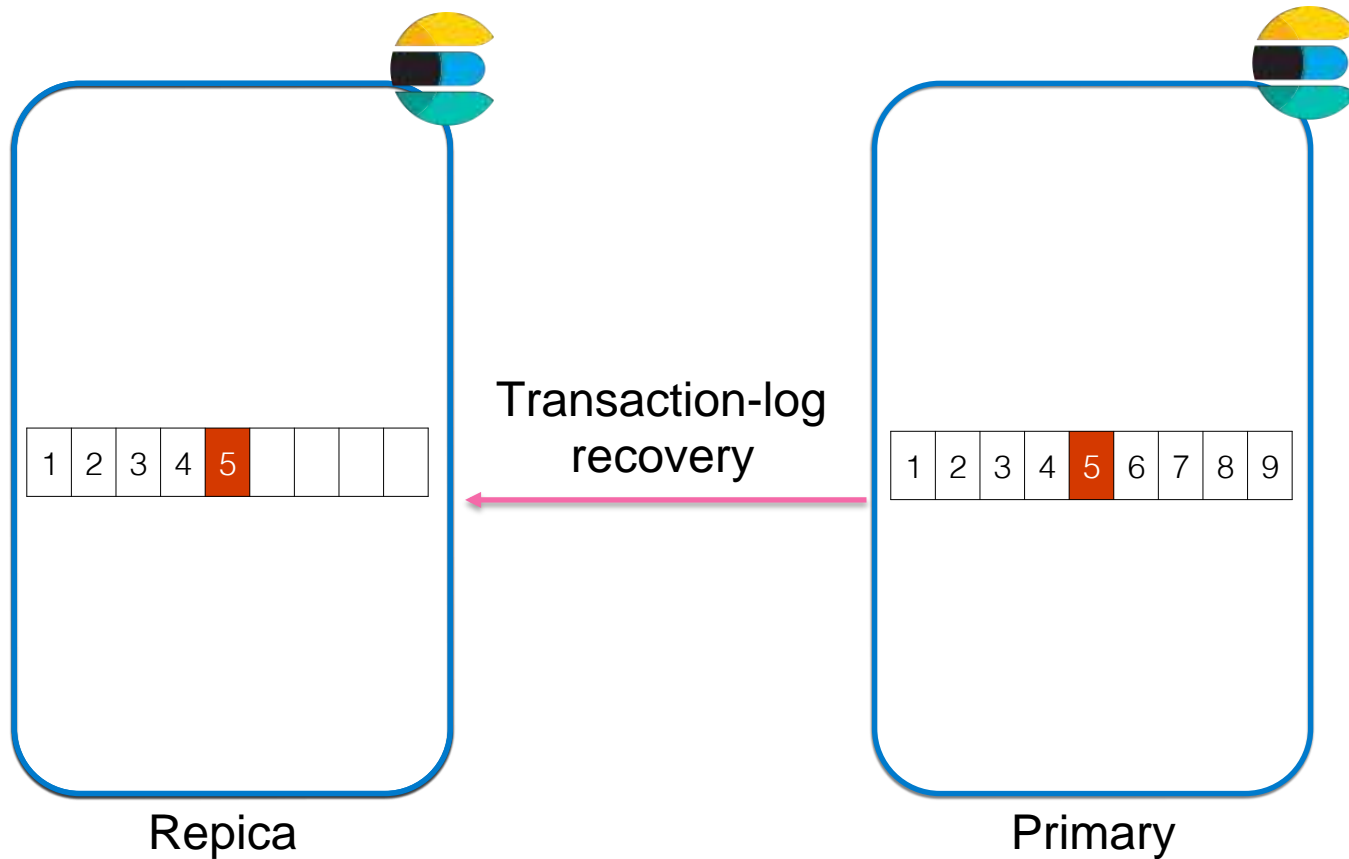


Primary

# Recovery (5.x)



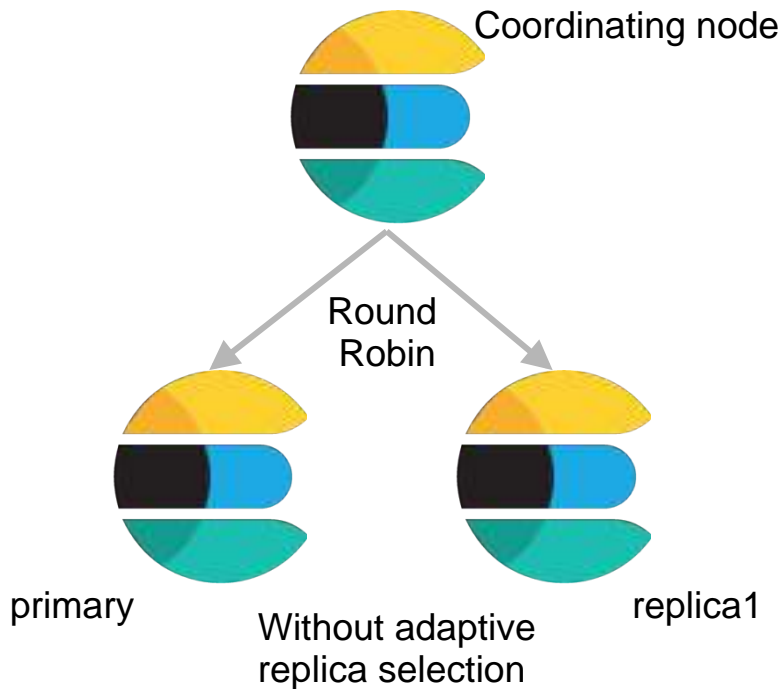
# Recovery (6.x)





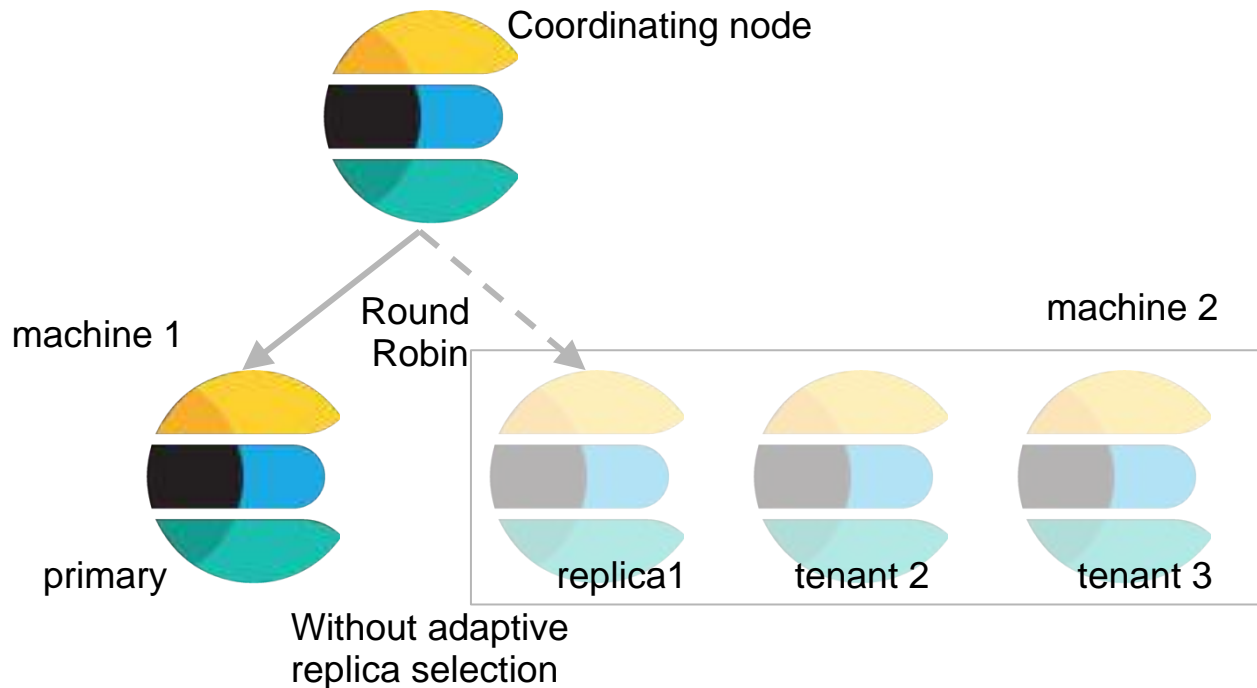
# Adaptive Replica Selection

Historic behavior is round robin



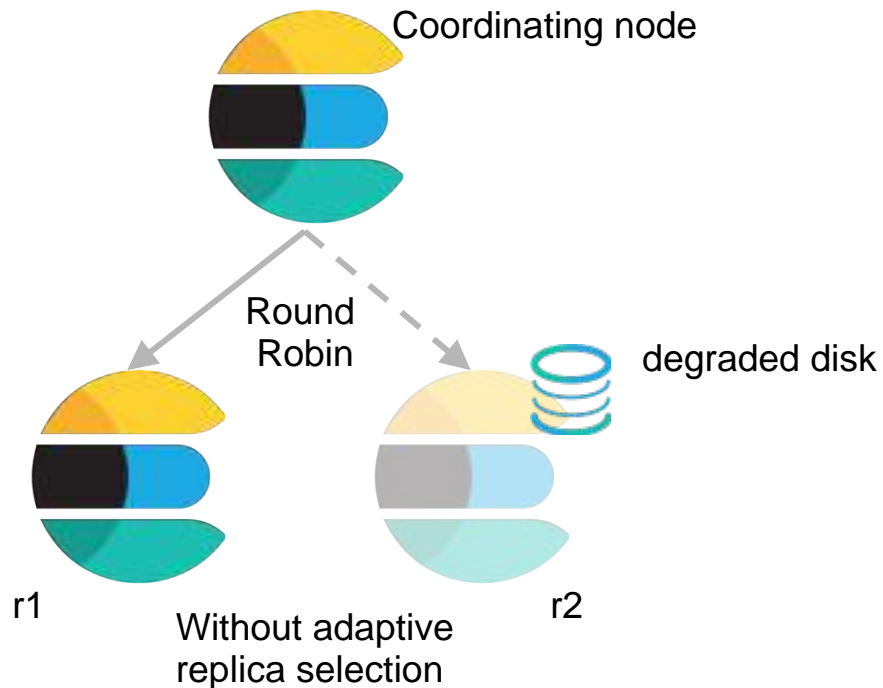
# Adaptive Replica Selection

But sometimes you're in a noisy-neighbor situation and that's not great



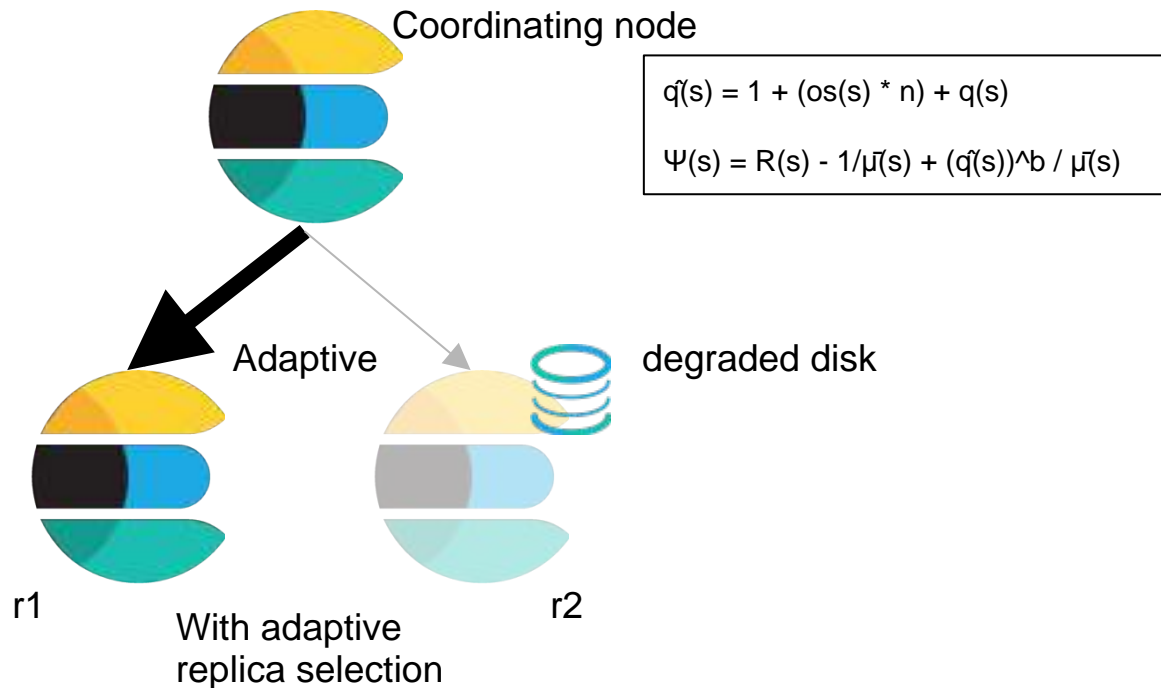
# Adaptive Replica Selection

Or you could have a degraded disk, causing slower response times



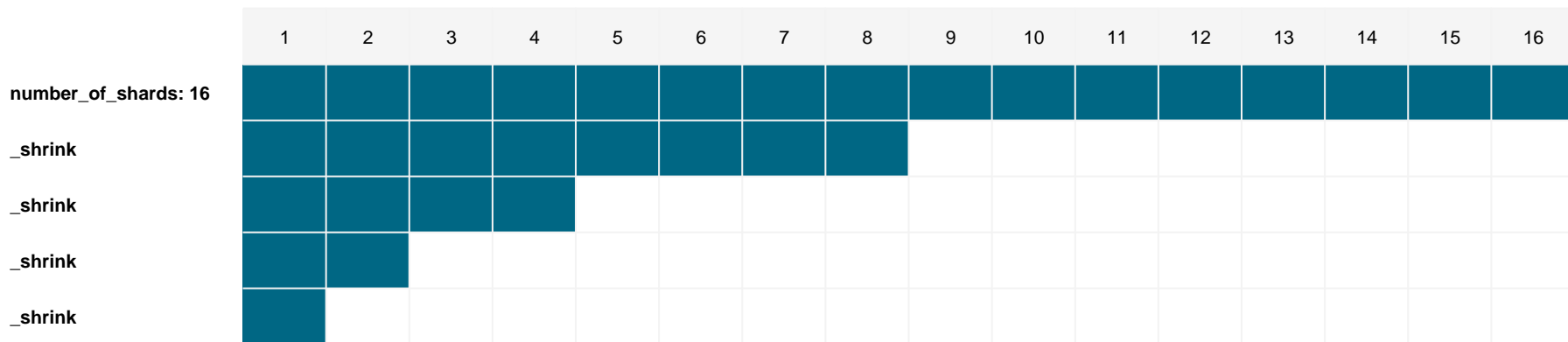
# Adaptive Replica Selection

Accounting for node performance in searches



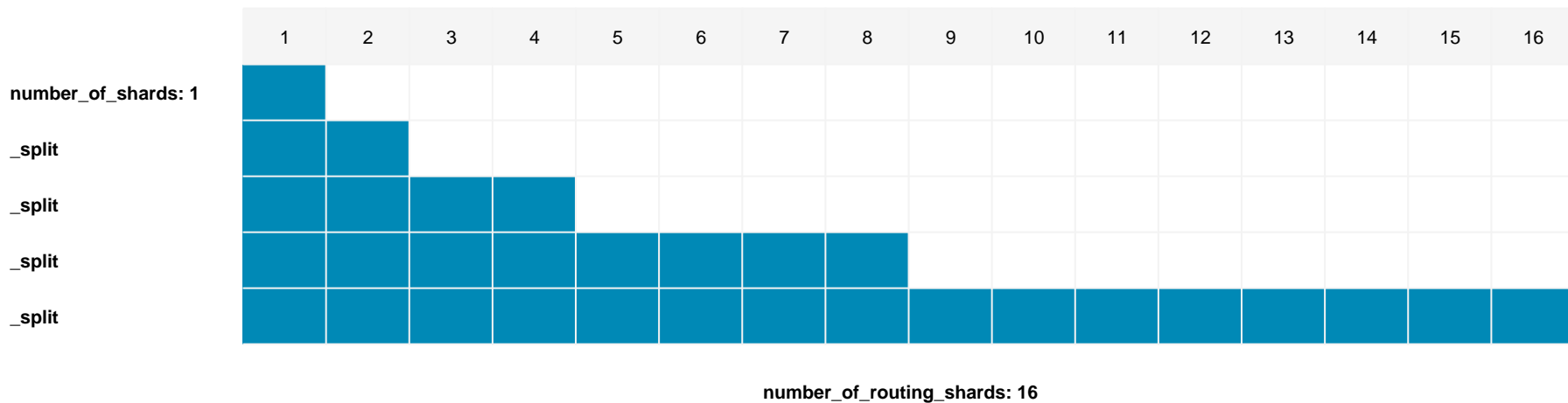
# Shard Shrinking

- Allows you to shrink an existing index into a new index with fewer primary shards
- Fast with hard-linking
- Copy of every shard in the index must be present on the same node



# Shard Splitting

- Fewer concerns up front on deciding correct number of shards
- Scale based on capacity demands
- Compliments shrink API and improves story on elastic scalability



# Composite Aggs

Let's aggregate pageviews for a Google Analytics type application

- Millions of URLs
- API/programmatic access to aggregation results

URL	Access Time
http://elastic.co	2017-12-15T12:10:30Z
https://www.elastic.co/guide/index.html	2017-12-15T12:10:40Z
http://elastic.co	2017-12-15T12:10:55Z



URL	Pageviews
http://elastic.co	20,000
https://www.elastic.co/guide/index.html	5,000
https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html	2,000

# Composite Aggs

Let's aggregate pageviews for a Google Analytics type application

```
GET page-views/_search
{
  "aggs" : {
    "my_buckets" : {
      "composite" : {
        "size": 10,
        "sources" : [
          { "url": { "terms" : { "field": "url", "order": "desc" } } }
        ]
      }
    }
  }
}
```



# Composite Aggs

Let's aggregate pageviews for a Google Analytics type application

```
GET page-views/_search
{
  "aggs" : {
    "my_buckets": {
      "composite" : {
        "size": 10,
        "after": { "url": "https://www.elastic.co/guide/en/elasticsearch/
reference/current/index.html" },
        "sources" : [
          { "url": { "terms" : { "field": "url", "order": "desc" } } } ]
      }
    }
  }
}
```

# Space-saving columnar store

Tapping into Lucene 7 goodness (sparse doc value)

- Better for storing sparse fields
- Save on disk space & file system cache

user	first	middle	last	age	phone
johns	Alex		Smith		
jrice	Jill	Amy	Rice		508.567.1211
mt123	Jeff		Twain	56	
sadams	Sue		Adams		
adoe	Amy		Doe	31	
lp12	Liz		Potter		

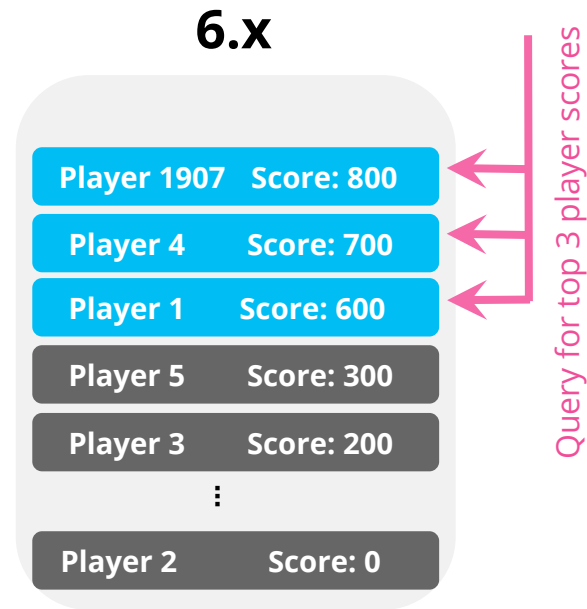
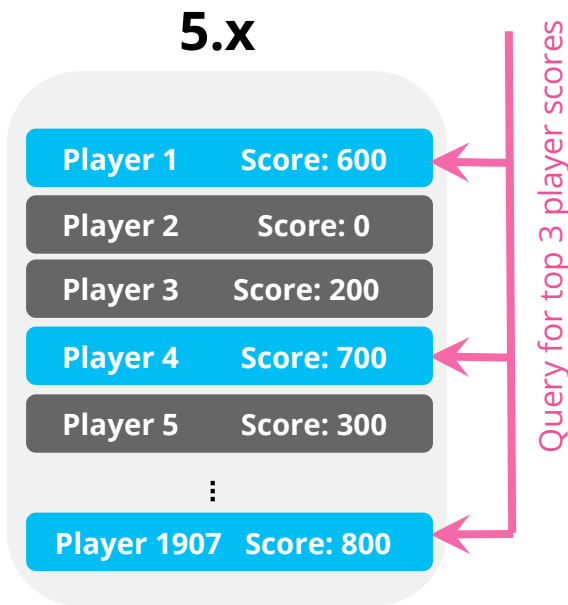
# Much speedier sorted queries

Tapping into Lucene 7 goodness (index sorting)

Sort at index time vs. query time

Optimize on-disk format for some use cases

Improve query performance at the cost of index performance



# Doc Values - Sparse Data (5.x)

Segment 1

ID	fname	lname
1	Shane	Connelly
2	Shay	Banon
3	Tanya	Bragin

Segment 2

ID	fname	lname	mi	state	city
4	Steve	Kearns	Null	Null	Boston
5	George	Burdell	P	GA	Null
6	Bill	Swerski	Null	Null	Chicago



Merged Segment 3

Docs	fname	lname	mi	state	city
1	Shane	Connelly	Null	Null	Null
2	Shay	Banon	Null	Null	Null
3	Tanya	Bragin	Null	Null	Null
4	Steve	Kearns	Null	Null	Boston
5	George	Burdell	P	GA	Null
6	Bill	Swerski	Null	Null	Baz

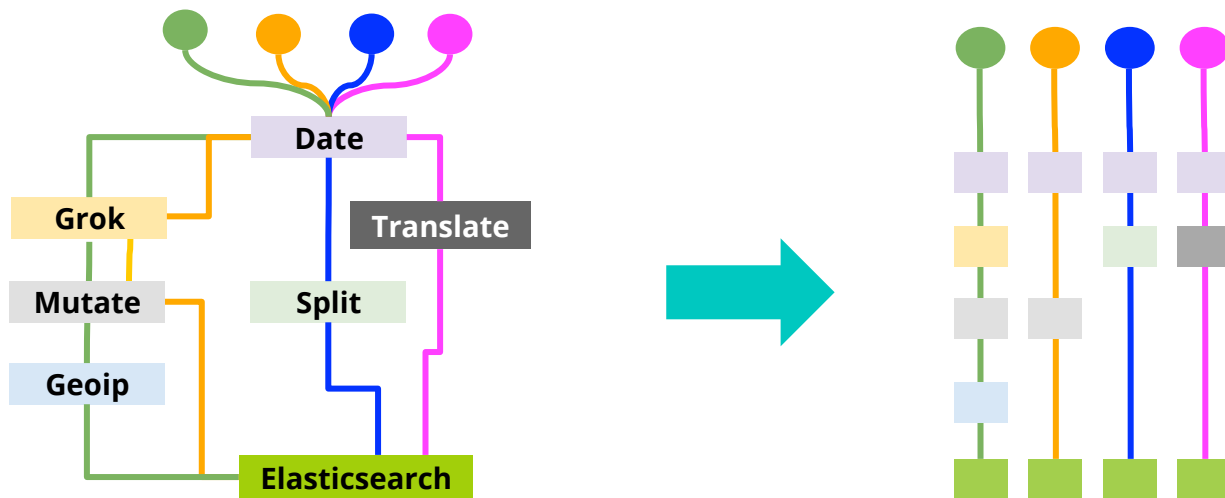


logstash

# Multiple Pipelines, One Logstash

Untangle complex Logstash configs with multiple pipelines

- Run multiple, distinct workloads on a single Logstash JVM
- Manage data flow per data source independently
- Track each pipeline separately with the new Pipeline Viewer



# Java execution engine (experimental, off by default)

Paves way for Java plugins

What is it

- Execution environment for Java plugins

Benefits

- Execute plugins in any JVM language

Guidance to customers

- Do not turn on in production!
  - Try in dev/test and report any issues
- `--experimental-java-execution`





beats



# Logging data

New in 6.1

FILEBEAT



WINLOGBEAT



## Infrastructure

### System

- Linux / MacOS
- Windows Events

### Containers

- Docker
- Kubernetes

## Applications

### Databases

- MySQL
- **PostgreSQL (6.1)**

### Queues

- Redis
- **Kafka (6.1)**

### Web / Proxy

- Apache
- Nginx
- **Traefik (6.1)**

### Elastic

- Elasticsearch\*
- Kibana\*
- **Logstash (6.1)**

# Metrics data

New in 6.1

HEARTBEAT METRICBEAT



## Infrastructure

### OS

- *System (uptime)*
- *Windows (service)*

### Containers

- Docker
- Kubernetes

### Virtualization

- vSphere

### Cloud metadata

- AWS
- GCP
- Azure
- DigitalOcean
- Alibaba

### Storage

- *Ceph (OSD)*

### Uptime

- Heartbeat

# Metrics data

New in 6.1

HEARTBEAT METRICBEAT



## Applications

### Datastores

- MySQL
- PostgreSQL
- MongoDB
- Couchbase
- Aerospike
- Memcached
- **Etcd (6.1)**

### Queues

- Kafka
- Redis
- *RabbitMQ (queue)*

### Elastic

- Elasticsearch
- Kibana
- **Logstash (6.1)**

### Custom metrics

- JMX/Jolokia
- PHP-FPM
- Golang
- Dropwizard
- *HTTP (server)*
- **Graphite (6.1)**

### Web servers

- Apache
- Nginx
- Other**
- HAProxy
- Zookeeper
- Prometheus



# Security Analytics Data

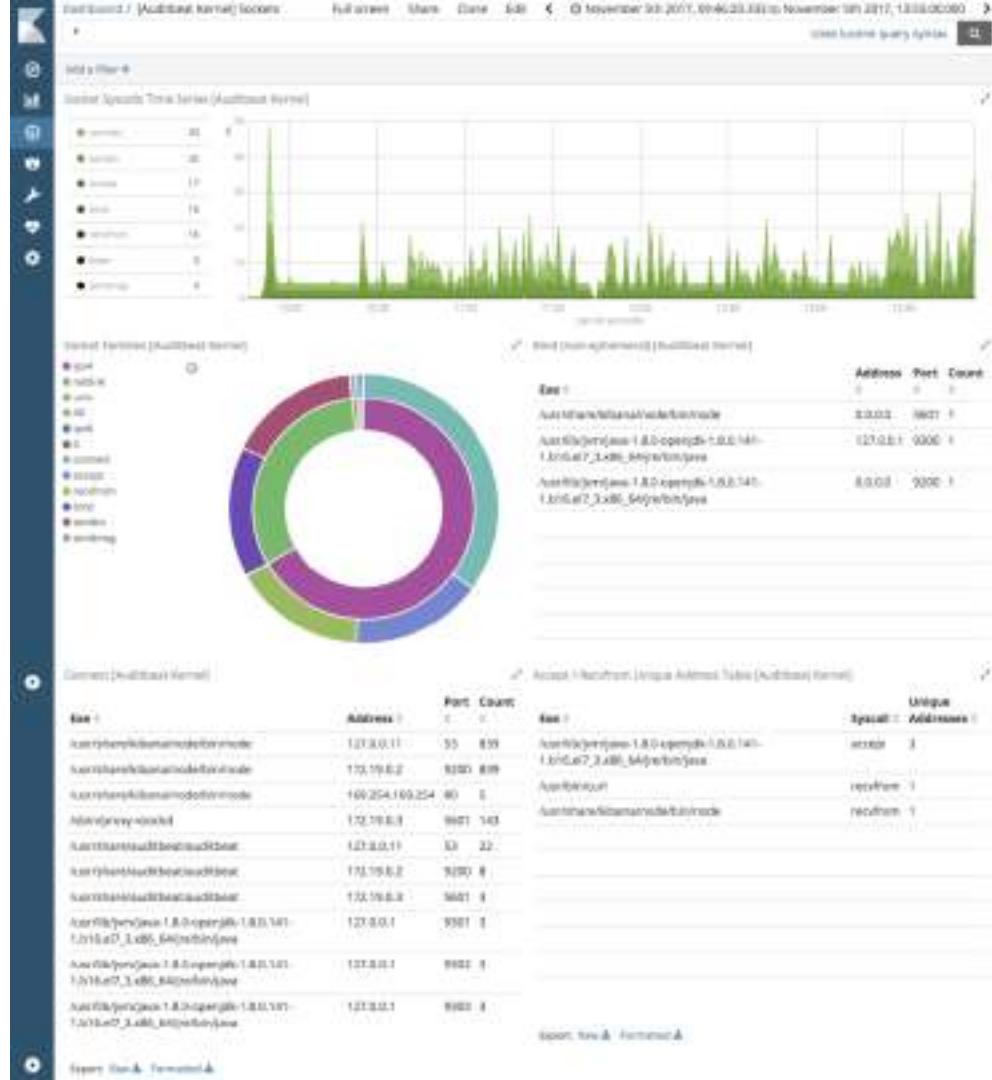
New in 6.1

## Packetbeat

- SSL envelope analysis

## Auditbeat

- Improved dashboards



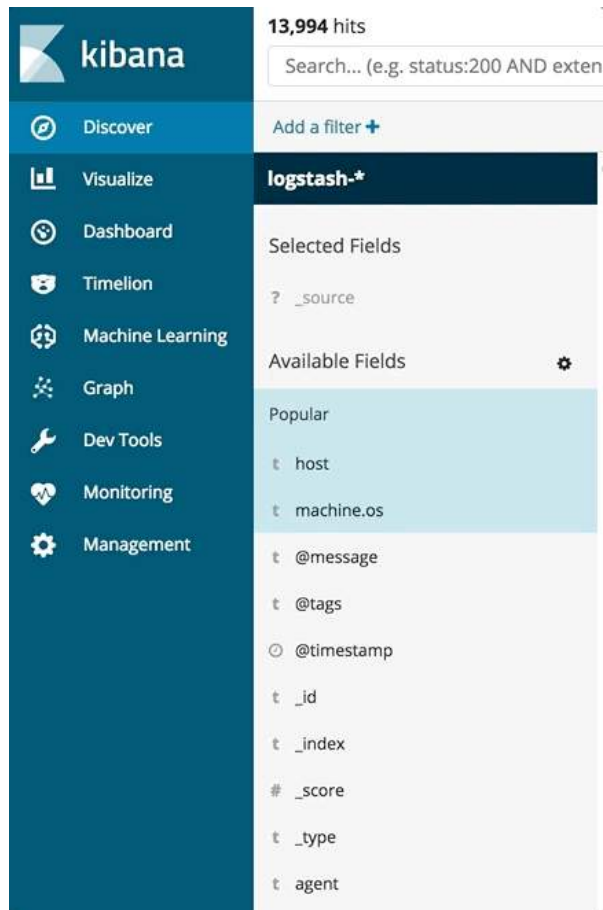


kibana

# Accessibility Initiative

## New & Improved in 6.0

- At Elastic, we have a very diverse and inclusive culture. We want to ensure our product is an extension of that and represents our Elastician values
- High contrast colors for the color blind
- Keyboard accessible
- Improved support for screen readers



The screenshot displays the Kibana search interface. On the left is a dark blue sidebar with navigation options: Discover (selected), Visualize, Dashboard, Timelion, Machine Learning, Graph, Dev Tools, Monitoring, and Management. The main content area shows a search for 'logstash-\*' with 13,994 hits. A search bar at the top contains the text 'Search... (e.g. status:200 AND exten'. Below the search bar is a section for 'logstash-\*' with 'Selected Fields' and 'Available Fields' sections. The 'Available Fields' section is expanded to show a list of fields: host, machine.os, @message, @tags, @timestamp, \_id, \_index, \_score, \_type, and agent. The 'Popular' section is also visible above the list.

# Full Screen Mode

## New & Improved in 6.0

- Full screen mode available for NOC's, SOC's and Kiosks
- Perfect for operations use case and "command centers"





# Kibana Home

The screenshot shows the Kibana Home dashboard. At the top left, it says "Welcome to Kibana". On the right, there is a link "Data stored in Elasticsearch?" and a button "See up-to-date patterns".

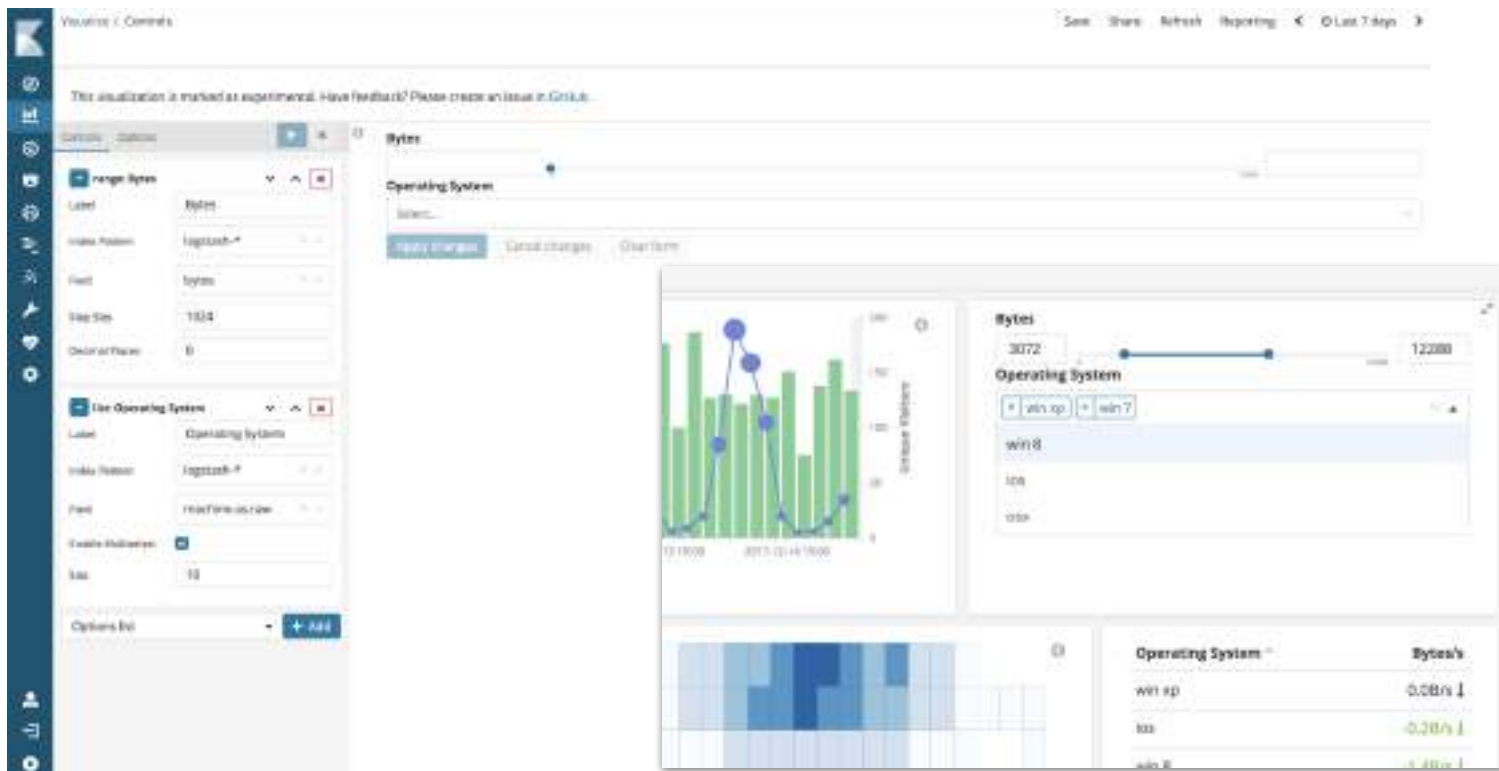
The dashboard is divided into two main sections:

- Visualize and Explore Data**
  - APM**: Automatically collect in-depth performance metrics and errors from inside your applications.
  - Discover**: Interactively explore your data by querying and filtering raw documents.
  - Machine Learning**: Automatically model the normal behavior of your time series data to detect anomalies.
  - Visualize**: Create visualizations and aggregate data stores in your Elasticsearch indices.
  - Dashboard**: Display and share a collection of visualizations and saved searches.
  - Graph**: Surface and analyze relevant relationships in your Elasticsearch data.
  - Timeline**: Use an expression language to analyze time series data and visualize the results.
- Manage and Administer the Elastic Stack**
  - Console**: Skip CLI and use the (DSL) interface to work with your data directly.
  - Monitoring**: Track the real-time health and performance of your Elastic Stack.
  - Security Settings**: Protect your data and evenly manage who has access to what with users and roles.
  - Index Patterns**: Manage the index patterns that help retrieve your data from Elasticsearch.
  - Saved Objects**: Import, export, and manage your saved searches, visualizations, and dashboards.
  - Watcher**: Detect changes in your data by creating, managing, and monitoring alerts.

At the bottom, there is a message "Didn't find what you were looking for?" and a button "View full directory of Kibana plugins".

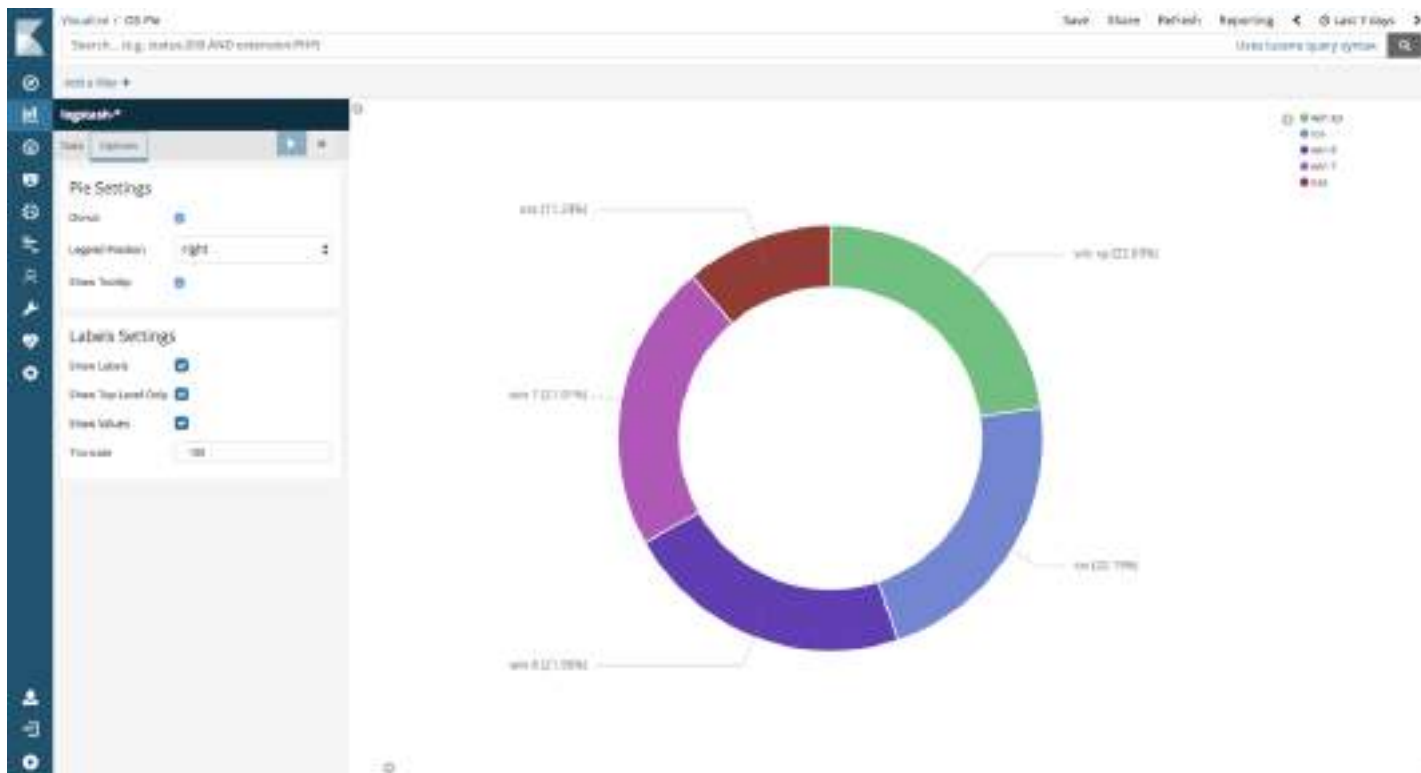
# Lab Visualizations

## Input Controls



# Pie Chart

## Data Labels



# Time Series Visual Builder

## Data Table

The screenshot displays the Elastic Time Series Visual Builder interface. At the top, the visualization is titled "Visualize / Bytes per second" and includes navigation options like "Save", "Share", "Refresh", and "Reporting". A warning message states: "This visualization is marked as experimental. Have feedback? Please create an issue in GitHub." Below this, the visualization type is set to "Table".

The data table shows the following metrics:

Operating System	Bytes/s
WIN XP	0.00%
ios	0.00%
WIN 8	1.40%
win 7	0.40%
ios	0.00%

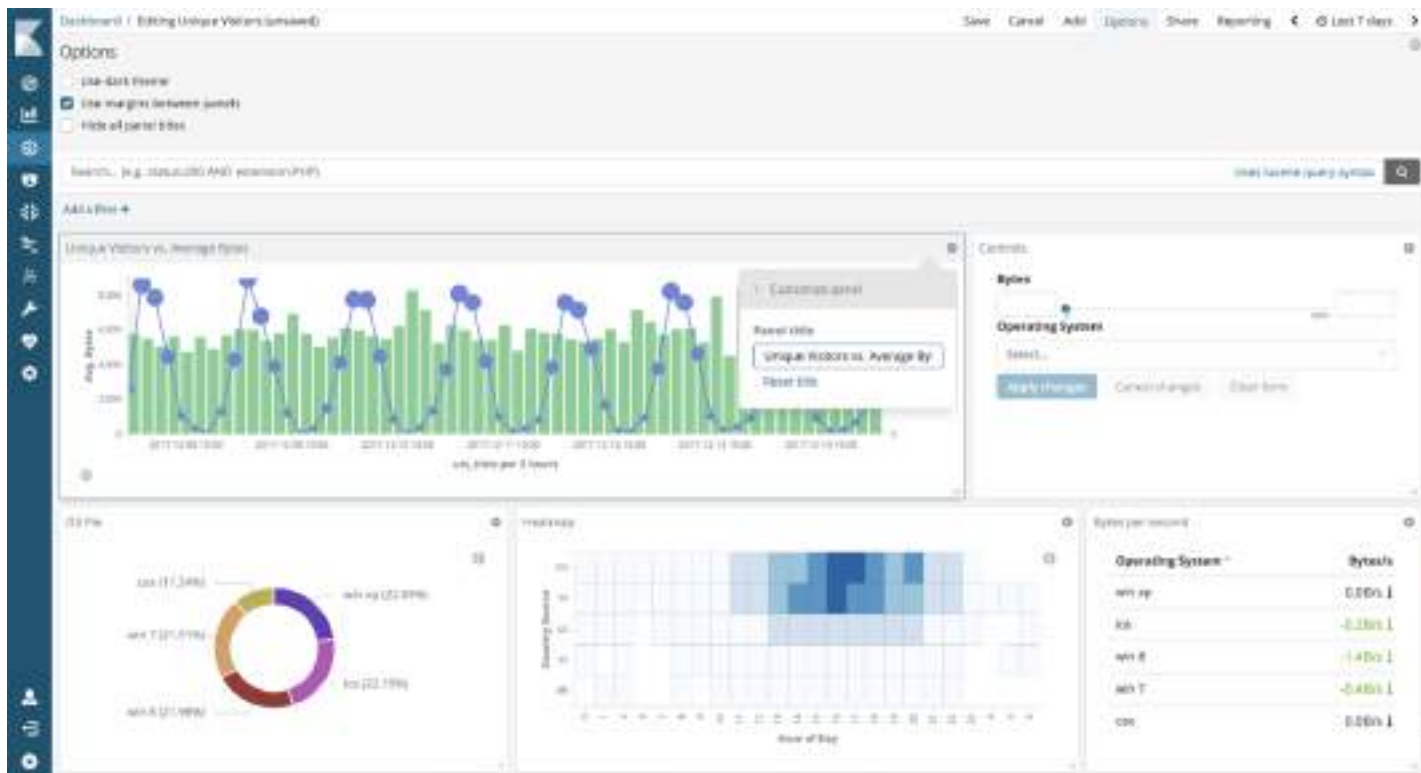
Below the table, there are configuration options for columns and metrics. The "Columns" section is titled "Panel Options" and includes a note: "For the table visualization you need to define a field to group by using a terms aggregation." The "Group by Field" is set to "machine.os" and the "Column Label" is "Operating System".

The "Metrics" section shows two metrics:

- Bytes/s**: Max - Digital
- Derivative**: Max of Bytes - 1s

# Dashboard Customization

Optional margins, customizable and hidden panel titles



# Future

What we are working on

# Kibana's new Experimental Query Language

- **Kuery Syntax:** `function("field", value)`
- **Like so:**
  - **Kuery:** `is("response", 200)`
    - Lucene: `response:200`
  - **Kuery:** `not(is("response", 404))`
    - Lucene: `!response:404`
  - **Kuery:** `range("bytes", gt=1000, lt=8000)`
    - Lucene: `bytes:[1000 to 8000]`
  - **Kuery:** `geoPolygon("geo.coordinates", "40.97, -127.26", "24.20, -84.375", "40.44, -66.09")`
    - Lucene: not supported

+ A lot of Lucene-style syntax still works in Kuery, including all of these examples

# Kibana Canvas





# And SQL

- Elasticsearch SQL
- Visualize in Kibana

The screenshot shows the SQLing interface in Kibana. At the top left, the title "SQLing" is displayed with navigation icons. The main content area is divided into two sections. The top section, titled "Just some SQL", contains a pie chart on the left and a text block on the right. The text block reads: "No biggie, just doing some SQL over here. Elasticsearch SQL is going to be a thing so I whipped up this function to let us write SQL queries. The SQL team is really doing some cool stuff and their data format matches up really well with Conna's [Statistics.js](#). Pretty jizzed to get this in everyone's hands soon. Things are looking good!". Below this is a bar chart with a y-axis ranging from 0 to 2500 and a x-axis with numerous labels rotated 45 degrees. The right sidebar contains configuration panels for "Dimensions & Measures", "Chart Style", "Text Settings", "Default Style", and "Legend". At the bottom, a code editor displays an Elasticsearch SQL query: 

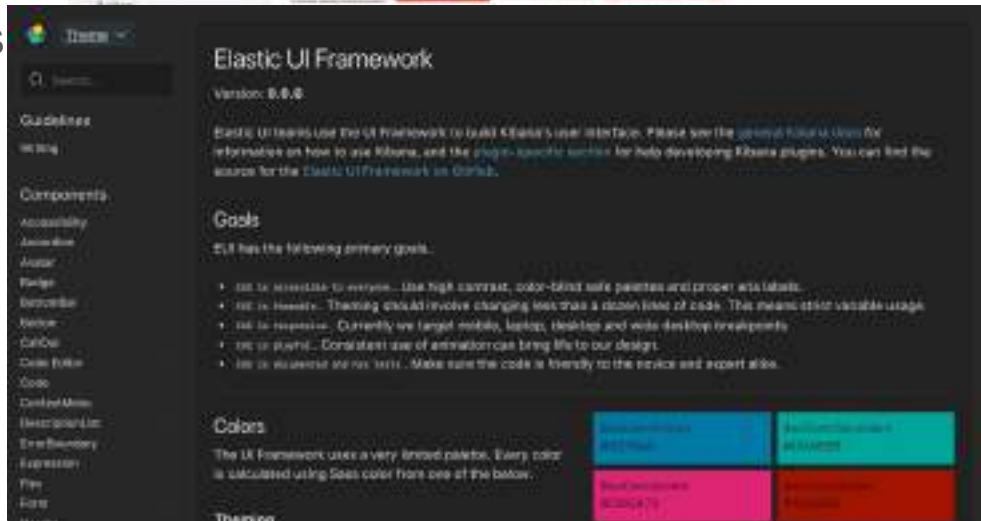
```
esSQL "SELECT 'monitor.host', count(*) as count from heartbeat nodes wr 'monitor.host' array 100 | pointsize color=monitor.host, x=monitor.host, y=count | plot font={font family="Open Sans", Helvetica, Arial, sans-serif} size=12 align="left" color="#FFFFFF" weight="normal" underline=false italic=false} legend=false defaultStyle={seriesStyle points="D" base=0.7} { number size="conna_element" overflow: visible; } .Plot-x-axis .Plot-y-axis { transform: rotate(110deg) translate(-100px, -0px); width: 100%; height: 100%; }
```

 Below the code editor are "Run" and "Close" buttons. The bottom of the image shows a navigation bar with icons for Home, Pages, Elements, Dashboard, Code, and Share, along with a "Give Feedback" link.

# Elastic UI Framework

- Kibana's user interface
- React components
- With many examples
- Best for develop Kibana plugins

- <https://github.com/elastic/eui>
- `npm install @elastic/eui`



# Cross Datacenter Replication

- Laying foundation of sequence numbers
  - Cross-datacenter replication
  - Changes API

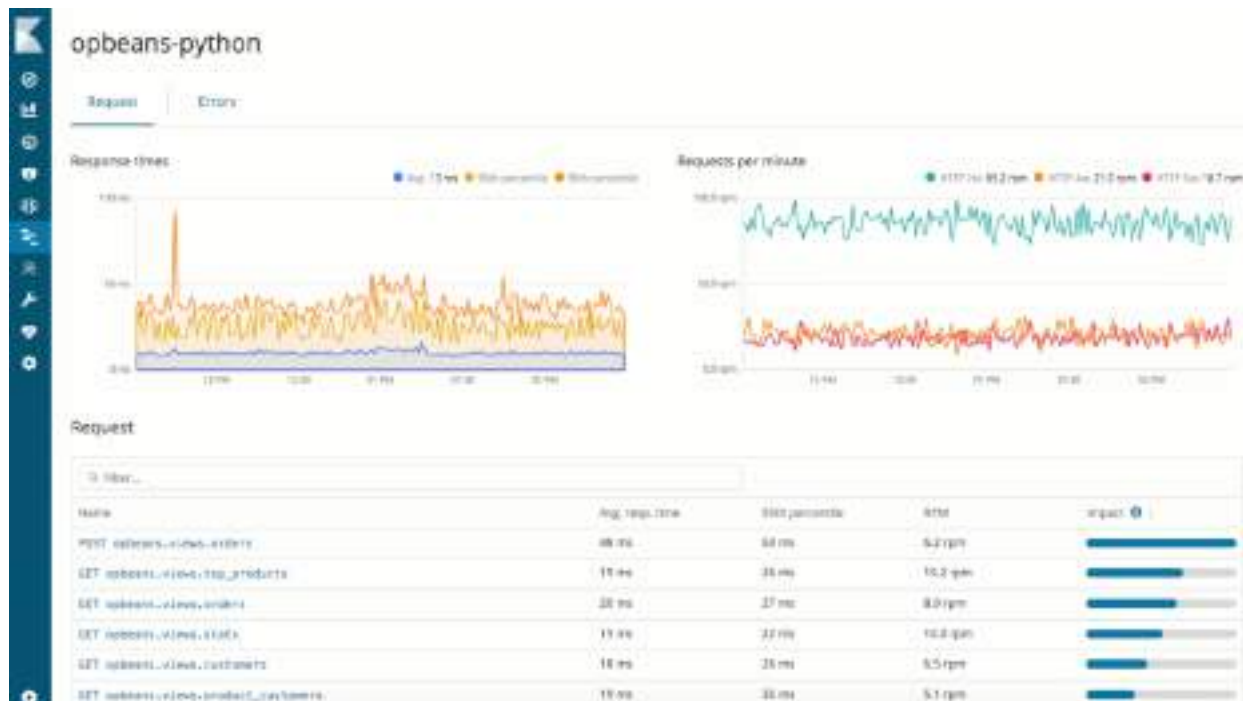


Replicate



# Elastic APM

- Nodejs
- Django
- Flask



<https://www.elastic.co/solutions/apm>



# THANK YOU

@elastic

[www.elastic.co](http://www.elastic.co)