



Elasticsearch

Query Language: ES|QL

Philipp Krenn

Philipp
Krenn

@xeraa

Developer 🥑



Query DSL

GET _search

```
{  
  "query": {  
    "match": {  
      "quote": "I know kung-fu"  
    }  
  }  
}
```

Query DSL

GET _search

```
{  
  "aggs": {  
    "quotes-by-character": {  
      "terms": {  
        "field": "character"  
      }  
    }  
  }  
}
```



_query



_search



How many query languages use `_search`?



Query
String

Timelion

EQL

Query DSL

Vega

SQL

Lucene

KQL



_search



~17 months of development

<https://github.com/elastic/elasticsearch/commit/8a1022e1c75fd2b99fa77a7ec548b0b2979b4662>



ES|QL

- Language
- Engine
- Visualization

ES|QL

the language

ES|QL Features

- Unstructured and structured data
- Procedural piped language
- SQL-like filtering and data manipulation
- Lookups

Piped Query Language?

- Pipes in (Unix) shells
- Similar languages
 - Splunk's Search Processing Language (SPL)
 - Microsoft's Kusto Query Language (KQL)
 - Amazon's CloudWatch queries

Demo

[https://xeraa.net/talks/
elasticsearch-piped-query-language-esql/](https://xeraa.net/talks/elasticsearch-piped-query-language-esql/)

Enrich at Runtime

		A
		B
		C
		D

input table

ENRICH
command

		A		
		B		
		C		
		D		

output table

A			
B			
C			
D			

enrich index

enrich
policy

A		
B		
C		
D		

source index

A			
B			
C			
D			

source index

ES|QL

the engine

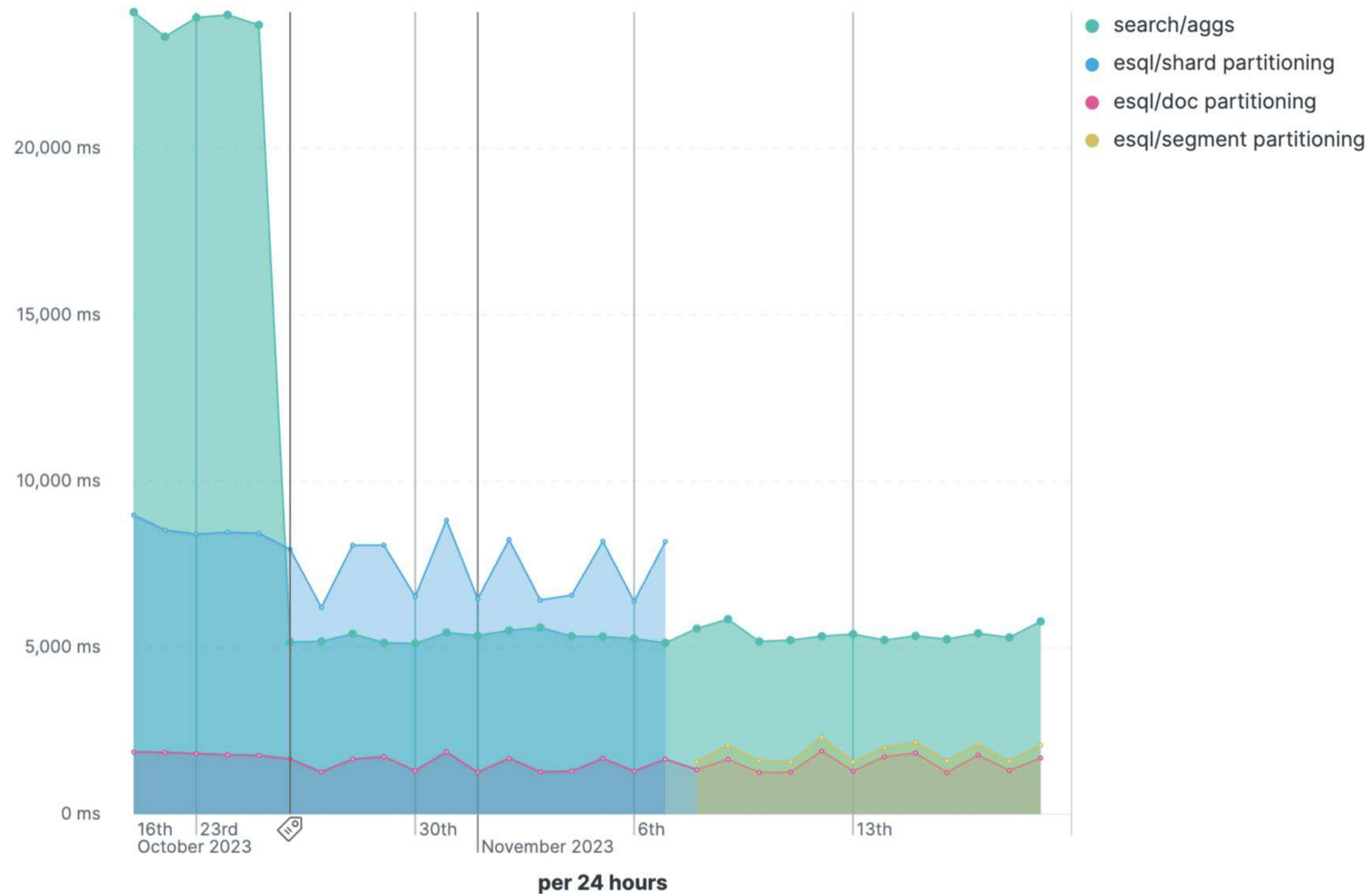
66

The new ES|QL execution engine was designed with performance in mind — it operates on blocks at a time instead of per row, targets vectorization and cache locality, and embraces specialization and multi-threading. It is a separate component from the existing Elasticsearch aggregation framework with different performance characteristics.

Benchmarks

<https://elasticsearch-benchmarks.elastic.co/#tracks/esql/nightly/default/30d>

nightly-esql/avg_tip_percent_eval ⓘ



ES|QL

the visualization

apache-logs

Filter your data using KQL syntax

Data views + Create a data view

Find a data view

.kibana-event-log-*

✓ apache-logs

apache-logs Temporary

apache-logs Temporary

apache-logs Temporary

elastiflix-movies

Try ES|QL Technical preview

geoip.region_name

http.request.method

http.response.body.bytes

http.response.status_code

1,000 hits Reset search



Aug 29, 2023 @ 00:00:00.000 - Nov 27, 2023 @ 03:48:08.679 (interval: Auto)

Documents Field statistics

1 field sorted

@timestamp

Document

Sep 12, 2023 @ 02:40:35.000 @timestamp Sep 12, 2023 @ 02:40:35.000 Body_kilobytes 8B geoip.contin

 geoip.country_iso_code US geoip.country_name United States geoip.local

 http.request.method POST http.response.body.bytes 8,771 http.response

Sep 12, 2023 @ 02:40:25.000 @timestamp Sep 12, 2023 @ 02:40:25.000 Body_kilobytes 22B geoip.contin

 geoip.country_iso_code US geoip.country_name United States geoip.local

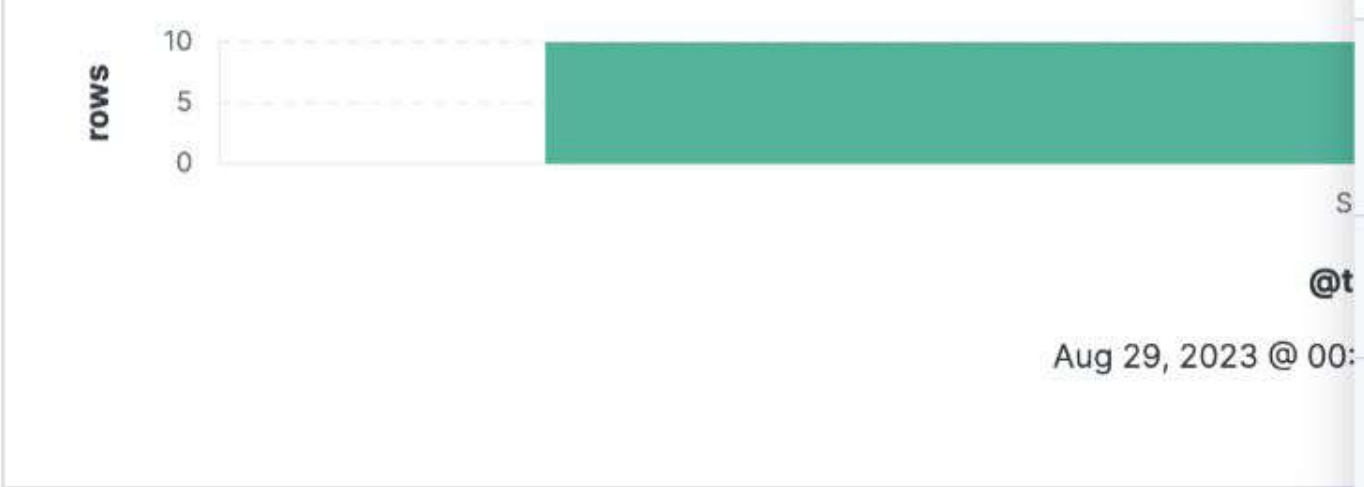
 http.request.method PUT http.response.body.bytes 23,265 http.response

ES|QL from apache-logs | limit 10

6 lines Last 90 days

- Search field names 0
- Available fields 23
- @timestamp
 - geop.city_name
 - geop.city_name.keyword
 - geop.continent_name
 - geop.continent_name.keyword
 - geop.country_iso_code
 - geop.country_iso_code.keyword
 - geop.country_name
 - geop.country_name.keyword
 - geop.location.lat
 - geop.location.lon
 - geop.region_iso_code
 - geop.region_iso_code.keyword
 - geop.region_name
 - geop.region_name.keyword
 - http.request.method
 - http.response.body.bytes
 - http.response.status_code
 - http.version
 - message

10 hits Reset search



@timestamp	Document
Sep 11, 2023 @ 23:54:05.000	@timestamp Sep 11, 2023 @ 23:54:05.000 E geop.country_name United Arab Emirates geop.region_iso_code AE-SH geop.location.lat 24.4668 geop.location.lon 54.3662 http.request.method GET http.response.status_code 200 http.version 1 message 175.233.253.216 - boehm8507 [11/Sep/2023:21:54:05 +0000] "GET /api/v1/health HTTP/1.0"
Sep 11, 2023 @ 23:54:15.000	@timestamp Sep 11, 2023 @ 23:54:15.000 Korea geop.location.lat 37.5111 geop.location.lon 127.0095 http.request.method GET http.response.status_code 200 http.version 1 message 175.233.253.216 - boehm8507 [11/Sep/2023:21:54:15 +0000] "GET /api/v1/health HTTP/1.0"
Sep 11, 2023 @ 23:54:25.000	@timestamp Sep 11, 2023 @ 23:54:25.000 http.request.method HEAD http.response.body.bytes 29,020 http.response.status_code 100 http.version 1 message 226.215.74.64 - - [11/Sep/2023:21:54:25 +0000] "HEAD /synergize/initiatives HTTP/1.0" 100 29020 source.address 226.215.74.64 url.original /synergize/initiatives
Sep 11, 2023 @ 23:54:35.000	@timestamp Sep 11, 2023 @ 23:54:35.000 geop.continent_name Asia geop.country_iso_code CN geop.country_name China a geop.location.lat 34.773 geop.location.lon 113.722 http.request.method POST http.response.body.bytes 12,578 http.response.status_code 204 http.version 2 message 121.237.56.83 - - [11/Sep/2023:21:54:35 +0000] "POST /systems..."
Sep 11, 2023 @ 23:54:45.000	@timestamp Sep 11, 2023 @ 23:54:45.000 geop.continent_name North America geop.country_iso_code US geop.country_name United States geop.location.lat 37.751 geop.location.lon -97.822 http.request.method GET

Rows per page: 100

ES|QL reference

Search

ES|QL

An ES|QL (Elasticsearch query language) query consists of a series of commands, separated by pipe characters: | . Each query starts with a **source command**, which produces a table, typically with data from Elasticsearch.

A source command can be followed by one or more **processing commands**. Processing commands can change the output table of the previous command by adding, removing, and changing rows and columns.

Source commands

- FROM
- ROW
- SHOW

Processing commands



```

1 from apache-logs
2 | where url.original == "/login"
3 | eval time_buckets = auto_bucket(@timestamp, 50, "2023-09-11T21:54:05.000Z", "2023-09-12T00:40:35.000Z")
4 | stats login_attempts = count(user.name) by time_buckets, user.name
5 | sort login_attempts desc
6 | limit 1000

```

6 lines @timestamp detected

Run query ⌘ + Enter

Search field names

0

ES|QL is currently in technical preview. Find more information in the documentation.

5 hits Reset search login_attempts over ti...



- haag8418
- goyette5638
- jacobson4857
- weimann5420

Selected fields 3

- login_attempts
- time_buckets
- user.name

Available fields 3

- login_attempts
- time_buckets
- user.name

Columns Sort fields

login_attempts

time_buckets

user.name



2 2023-09-11T21:55:00.000Z

haag8418



1 2023-09-11T22:00:00.000Z

goyette5638

ES|QL Last 90 days Refresh

```

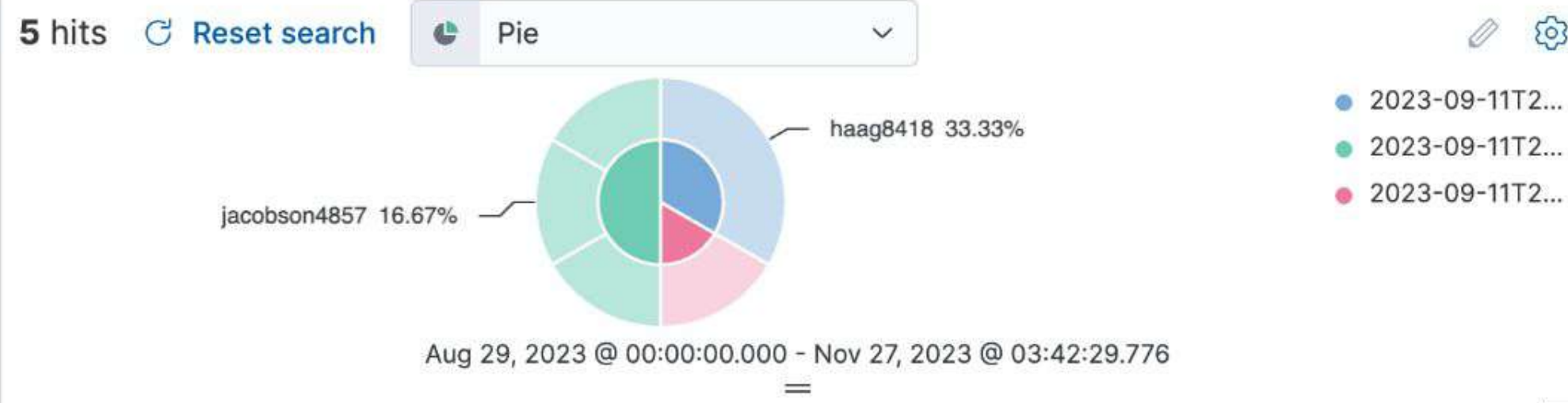
1 from apache-logs
2 | where url.original == "/login"
3 | eval time_buckets = auto_bucket(@timestamp, 50, "2023-09-11T21:54:05.000Z", "2023-09-12T00:40:35.000Z")
4 | stats login_attempts = count(user.name) by time_buckets, user.name
5 | sort login_attempts desc
6 | limit 1000

```

6 lines @timestamp detected Run query ⌘ + Enter

- Search field names 0
- Selected fields** 3
- login_attempts
 - time_buckets
 - user.name

- Available fields** 3
- login_attempts
 - time_buckets
 - user.name



Columns Sort fields

login_attempts	time_buckets	user.name
2	2023-09-11T21:55:00.000Z	haag8418
1	2023-09-11T22:00:00.000Z	goyette5638

ES|QL currently offers limited configuration options

Configuration options input fields

Pie chart configuration: apache-logs

Slice by configuration: time_buckets, user.name

Metric configuration: login_attempts

Cancel Apply and close

ES|QL

```

1 from apache-logs
2 | where url.original == "/login"
3 | eval time_buckets = auto_bucket(@timestamp, 50, "2023-09-11T21:54:05.000Z", "2023-09-12T00:40:35.000Z")
4 | stats login_attempts = count(user.name) by time_buckets, user.name
5 | sort login_attempts desc
6 | limit 1000

```

6 lines @timestamp detected

Search field names 0

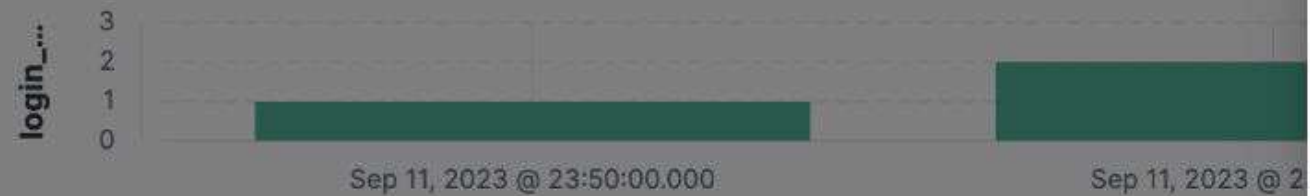
5 hits Reset search login_attempts over ti...

Selected fields 3

- login_attempts
- time_buckets
- user.name

Available fields 3

- login_attempts
- time_buckets
- user.name



Columns Sort fields

login_attempts	time_buckets
2	2023-09-11T21:54:05.000Z
1	2023-09-11T22:00:40.350Z

Create rule

Name

Tags (optional)

Elasticsearch query

Alert when matches are found during the latest query run. [Learn more](#)

Define your query using ES|QL

ES|QL editor

```

1 from apache-logs
2 | where url.original == "/login"
3 | eval time_buckets = auto_bucket(@timestamp, 50,
  "2023-09-11T21:54:05.000Z", "2023-09-12T00:40:35.000Z")
4 | stats login_attempts = count(user.name) by time_buckets,
  user.name

```

Cancel

Save

Observability

- Overview
- Alerts
- SLOs
- Cases

Logs

- Explorer **BETA**
- Stream
- Anomalies
- Categories

Infrastructure

- Inventory
- Metrics Explorer
- Hosts **BETA**

APM

- Services**
- Traces
- Dependencies

Synthetics

- Monitors **NEW**

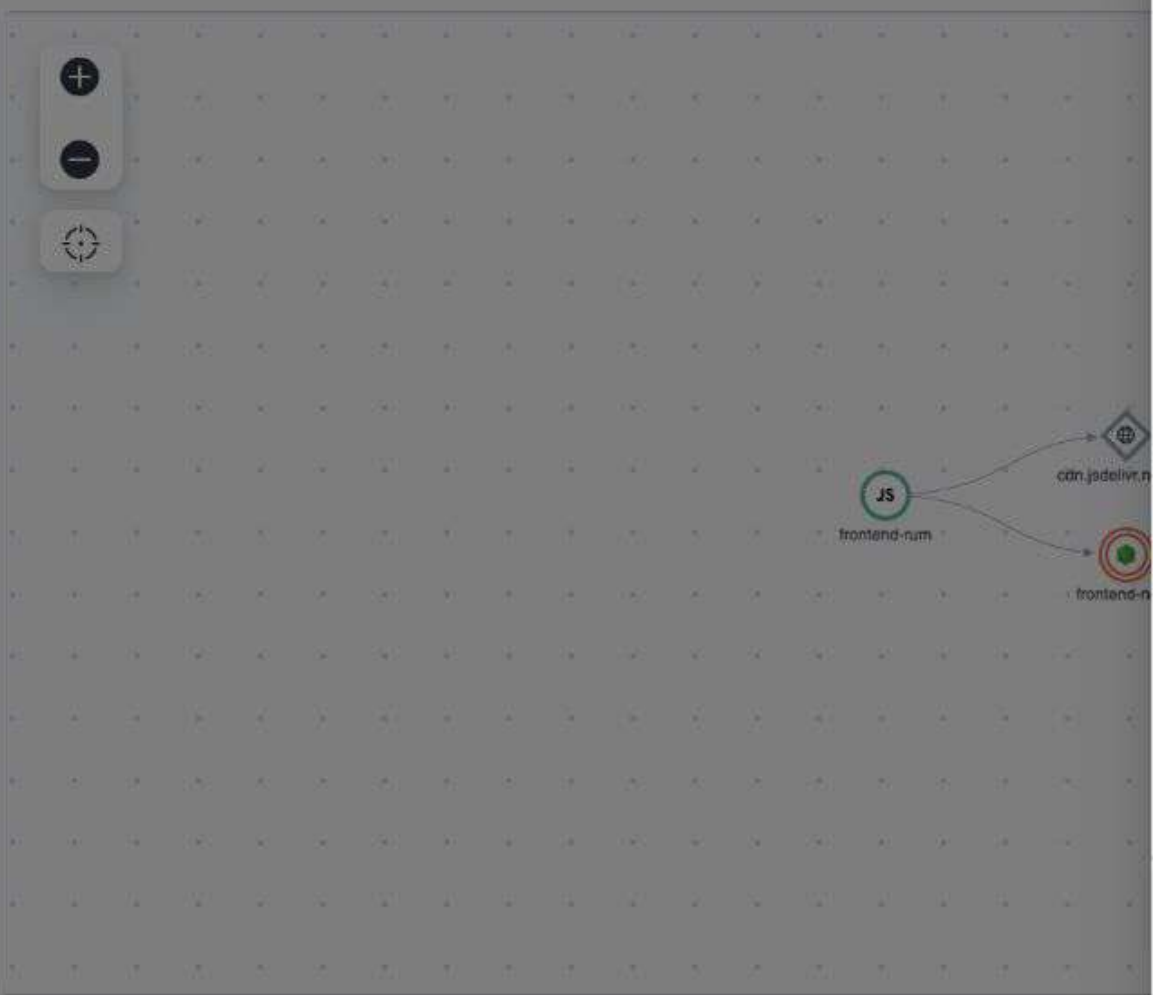
Services

All services Service groups

Inventory **Service Map**

Search transactions, errors and metrics (E.g. transaction.duration)

Comparison Day before



Open conversation

Generating an ES|QL Query for Calculating Median L...

You

can you generate an ES|QL query to calculate the median latency of all the services, please?

Show 4 events

Elastic Assistant

To calculate the median latency of all the services, you can use the **MEDIAN** aggregation function in a **STATS** command. However, I need to know the index pattern and the exact field names for the service and latency.

Here is a placeholder query:

```
FROM "<index-pattern>"
| STATS median_latency = MEDIAN("<latency-field>") BY "<service-field>"
```

[Run this query](#)

Please replace **<index-pattern>**, **<latency-field>**, and **<service-field>** with your actual index pattern and field names.

Send a message to the Assistant



```
1 from traces-apm.rum-default
```

1 line @timestamp found 1 warning

Submit feedback Run query + Enter

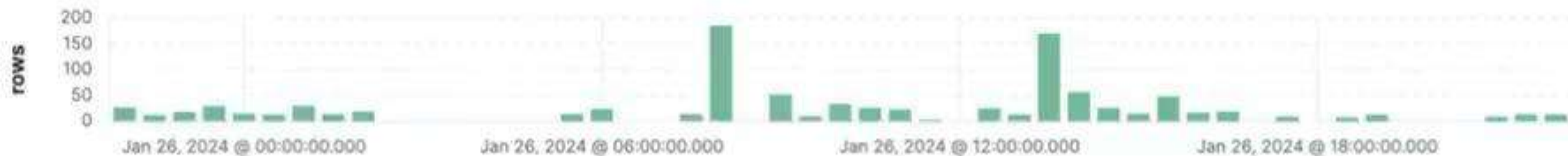
Search field names 0

Available fields 220

- @timestamp
- agent.activation_method
- agent.ephemeral_id
- agent.name
- agent.version
- child.id
- client.domain
- client.geo.city_name
- client.geo.continent_name
- client.geo.country_iso_code

ES|QL is currently in technical preview. Find more information in the documentation.

1,017 hits



@timestamp every 30 minute

Jan 25, 2024 @ 22:00:00.000 - Jan 26, 2024 @ 22:16:08.225

@timestamp

Document

Jan 26, 2024 @ 14:28:05.445 @timestamp Jan 26, 2024 @ 14:28:05.445 agent.name rum-js agent.version 5.16.0 client.geo.city_name W
arsaw client.geo.continent_name Europe client.geo.country_iso_code PL client.geo.country_name Polan

Conclusion

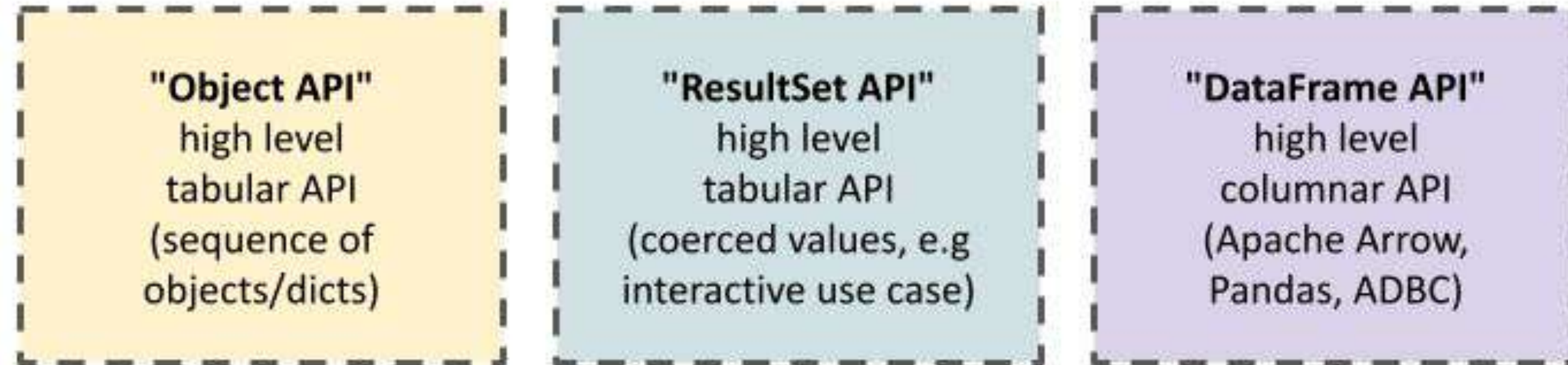
Limitations

elastic.co/guide/en/elasticsearch/reference/current/esql-limitations.html

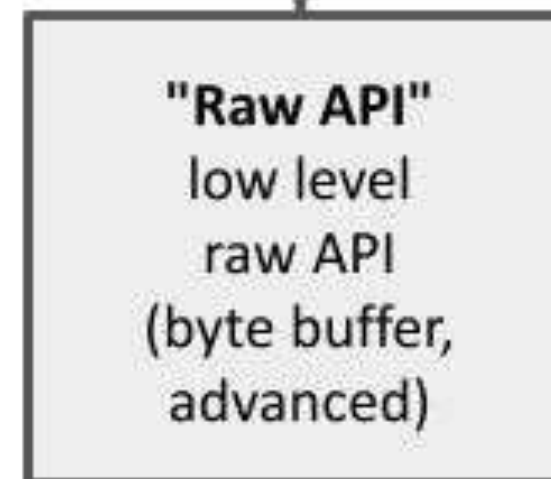
- Full-text search — keyword-based queries only
- flattened, nested,... fields
- async search, pagination
- Cross Cluster Query (CCQ)
- TSDS
- Language clients (raw API in 8.12)

Language Clients

High-level APIs focus on specialised use cases. These will be implemented in language clients as required, depending on ecosystem needs and idioms.



The low-level "raw" API exposes all details of the result data, but might not provide the smoothest developer experience. All language clients will have this API, auto-generated from the API spec.



Path to GA

- Stability:
 - performance improvements
 - memory tracking / circuit breakers
 - language consistency
- Better Kibana UX
- Docs



Try

<https://esql.demo.elastic.co>

[https://www.elastic.co/guide/en/elasticsearch/
reference/current/esql.html](https://www.elastic.co/guide/en/elasticsearch/reference/current/esql.html)



ES|QL

- Language
- Engine
- Visualization

Kahoot





Elasticsearch Piped Query Language: ES|QL

Philipp Krenn