Optimize Elasticsearch: Synthetic Source and TSDS

Philipp Krenn @xeraa

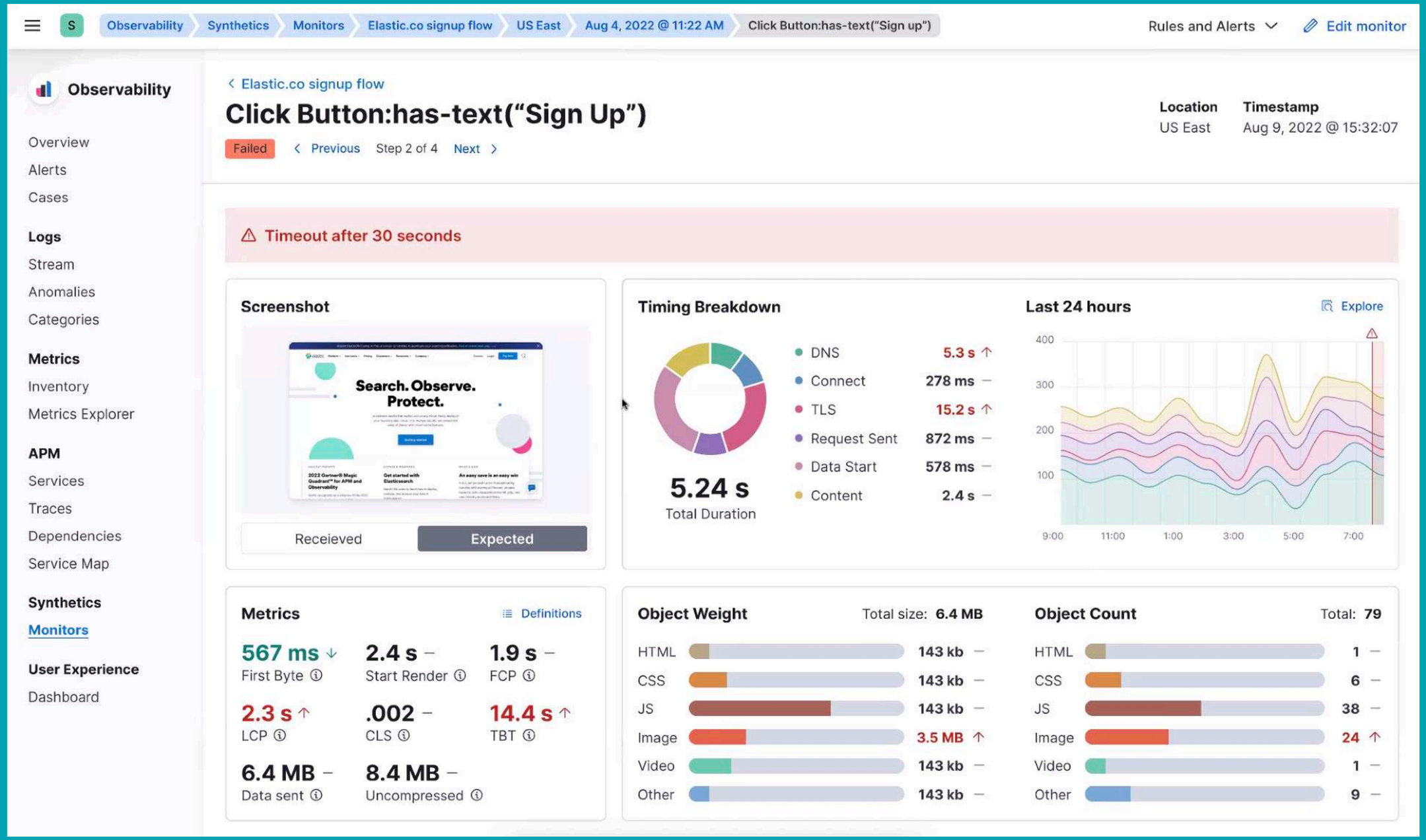
elastic

Developer 🥑

Synthetic Source

https://github.com/elastic/elasticsearch/issues/86603

Not Synthetic Monitoring



Not Elasticsearch Source Code

elastic / elasticsearch Public

Edit Pins | Watch 2.7k | Fork 22.3k | Starred 61.4k

Code | Issues 3.5k | Pull requests 511 | Actions | Projects 1 | Security | Insights

main | 210 branches | 347 tags | Go to file | Add file | Code

ywangd [Test] Ensure unique additional headers (#90781) ... 60d327d 3 hours ago 66,053 commits

.ci	Bump versions after 8.4.3 release	6 days ago
.github	GitHub Workflows security hardening (#90124)	8 days ago
.idea	Stop versioning Checkstyle IDE config (#87285)	4 months ago
benchmarks	Speed `getIntLE` from `BytesReference` (#90147)	20 days ago
build-conventions	Fix usage of `IndexAccessControl` in `CustomAuthorizationEngine`...	11 days ago
build-tools-internal	Create gradle plugin for ES stable plugins (#90355)	2 days ago
build-tools	Fix plugin wiring in yaml rest test plugin (#90818)	3 hours ago
ccr/images	[DOCS] Update remote cluster docs (#77043)	13 months ago
client	Assert wildcards are not expanded as specified by request options (...	5 days ago
dev-tools	Add convenience script for pruning old dev branch CI jobs	2 months ago
distribution	Update forbiddenapis to 3.4 (#90624)	5 days ago
docs	Fix quadratic complexity in SnapshotStatus serialization (#90795)	14 hours ago
gradle	Create gradle plugin for ES stable plugins (#90355)	2 days ago
libs	Update forbiddenapis to 3.4 (#90624)	5 days ago

About

Free and Open, Distributed, RESTful Search Engine

www.elastic.co/products/elasticsearch

java | search-engine | elasticsearch

Readme

View license

61.4k stars

2.7k watching

22.3k forks

Releases 98

Elasticsearch 8.4.3 Latest

6 days ago

+ 97 releases

Packages

No packages published

Publish your first package

Elasticsearch _source

_source field

edit

The _source field contains the original JSON document body that was passed at index time. The _source field itself is not indexed (and thus is not searchable), but it is stored so that it can be returned when executing *fetch* requests, like get or search.

If disk usage is important to you then have a look at synthetic _source which shrinks disk usage at the cost of only supporting a subset of mappings and slower fetches or (not recommended) disabling the _source field which also shrinks disk usage but disables many features.

Synthetic _source [preview]

edit

Though very handy to have around, the source field takes up a significant amount of space on disk. Instead of storing source documents on disk exactly as you send them, Elasticsearch can reconstruct source content on the fly upon retrieval. Enable this by setting mode: synthetic in _source:

```
PUT idx
{
  "mappings": {
    "_source": {
      "mode": "synthetic"
    }
  }
}
```

Copy as curl View in Console

Example

```
PUT movies/_doc/1
{
  "title": "Star Wars: A New Hope",
  "quote": "These are not the droids you are looking for"
}
```

Why _source?

_update, _update_by_query, _reindex

Highlighting

Source document

Why Mappings?

Dynamic vs static

Mapping for features & correctness

Optimization

`store`

Don't retrieve complete `_source`

Doesn't replace `_source`

Include or Exclude from `_source`

Mostly don't

Use `_source` filtering instead

Synthetic `_source`

Reconstruct at runtime from indexed data (small overhead)

`_update`, `_update_by_query`, `_reindex`, highlighting possible

Supported Fields

8.4: `boolean`, `byte`, `double`, `float`, `geo_point`, `half_float`, `integer`, `ip`, `keyword`, `constant_keyword`, `long`, `scaled_float`, `short`, `text` (with a keyword sub-field)

8.5 adds `aggregate_metric_double`, `date`, `date_nanos`, `dense_vector`, `histogram`, `version`, `match_only_text`, `ignore_above`

Limitations

Every field in the mapping must be supported

Normalizers on `keyword`, `flattened`,... not possible

Source Rewriting

Alphabetical order of attributes

```
{
  "foo.bar.baz": 1
}
```

⬇️

```
{
  "foo": {
    "bar": {
      "baz": 1
    }
  }
}
```

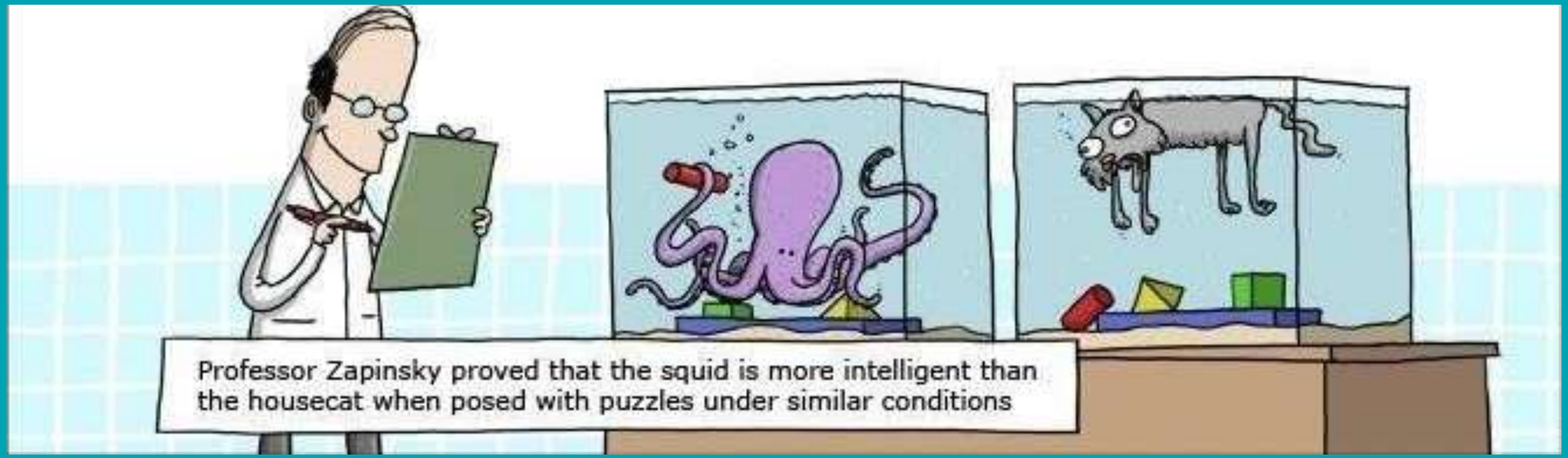
Source Rewriting

```
{
  "foo": [
    {
      "bar": 1
    },
    {
      "bar": 2
    }
  ]
}
```

⬇️

```
{
  "foo": {
    "bar": [1, 2]
  }
}
```

This Is Not a Benchmark

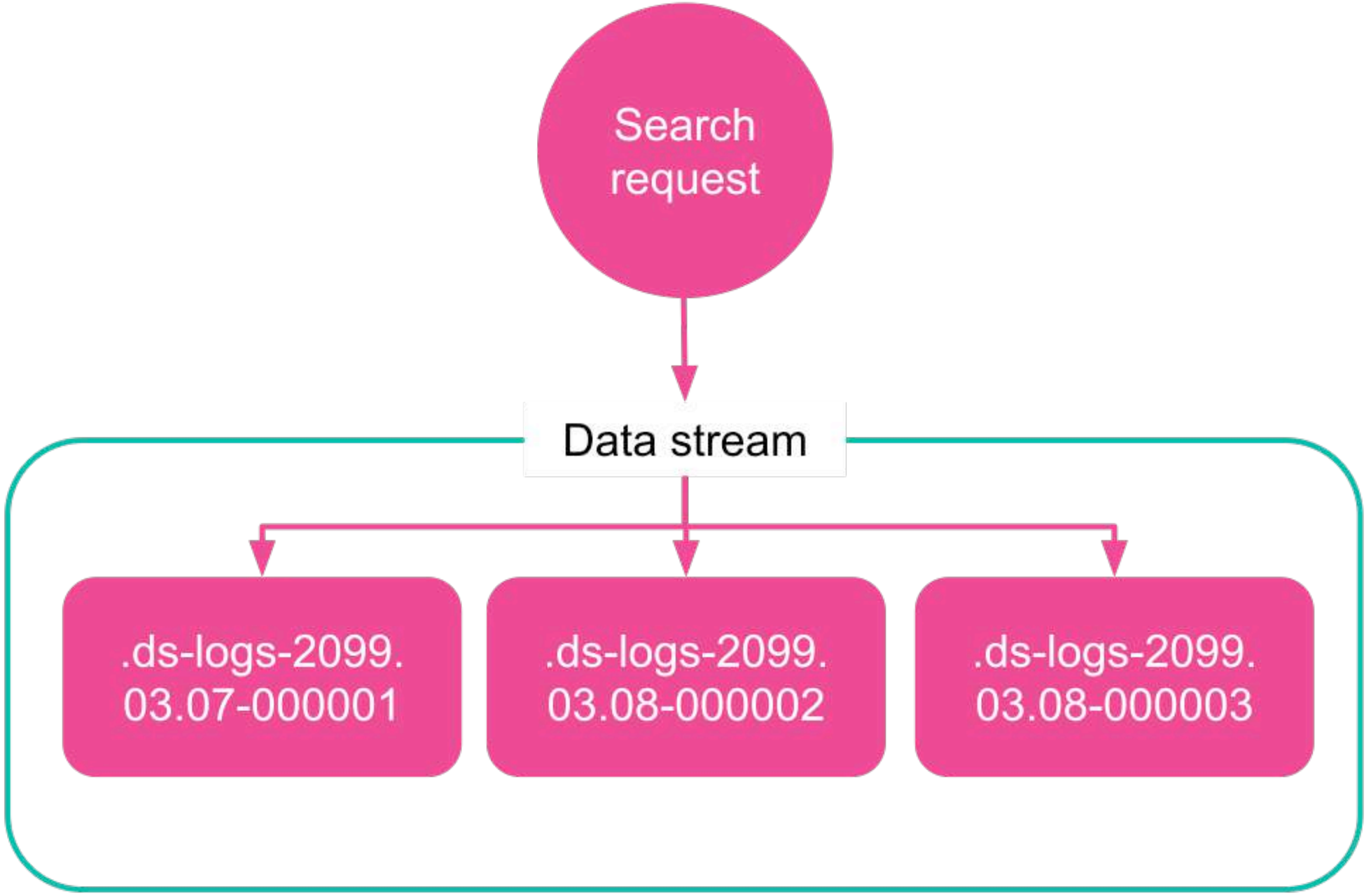


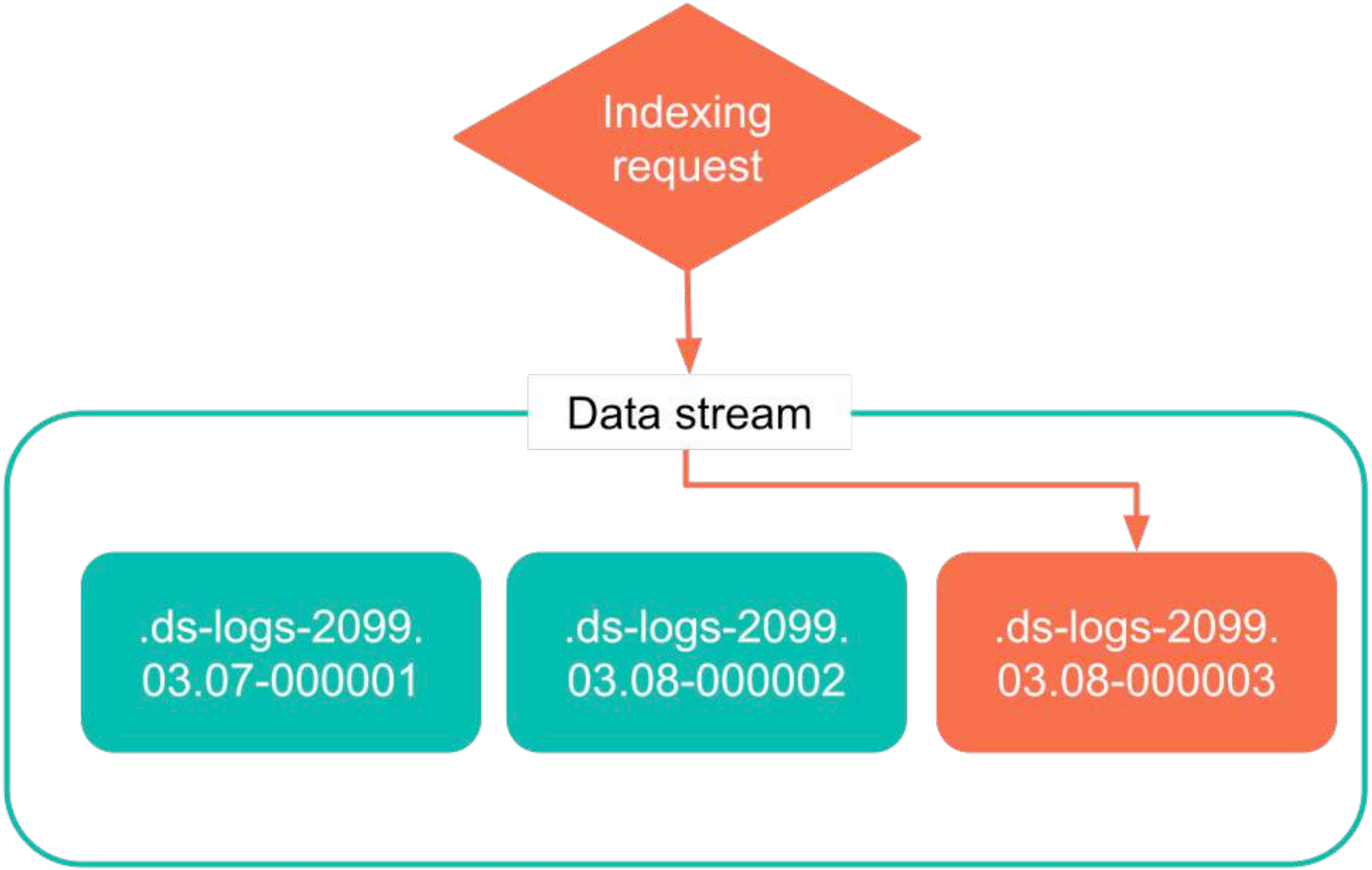
Benchmark

Your data

Proper shard size

Time Series Data Stream (TSDS)


Search request
Data stream
.ds-logs-2099.03.07-000001
.ds-logs-2099.03.08-000002
.ds-logs-2099.03.08-000003


Indexing request
Data stream
.ds-logs-2099.03.07-000001
.ds-logs-2099.03.08-000002
.ds-logs-2099.03.08-000003

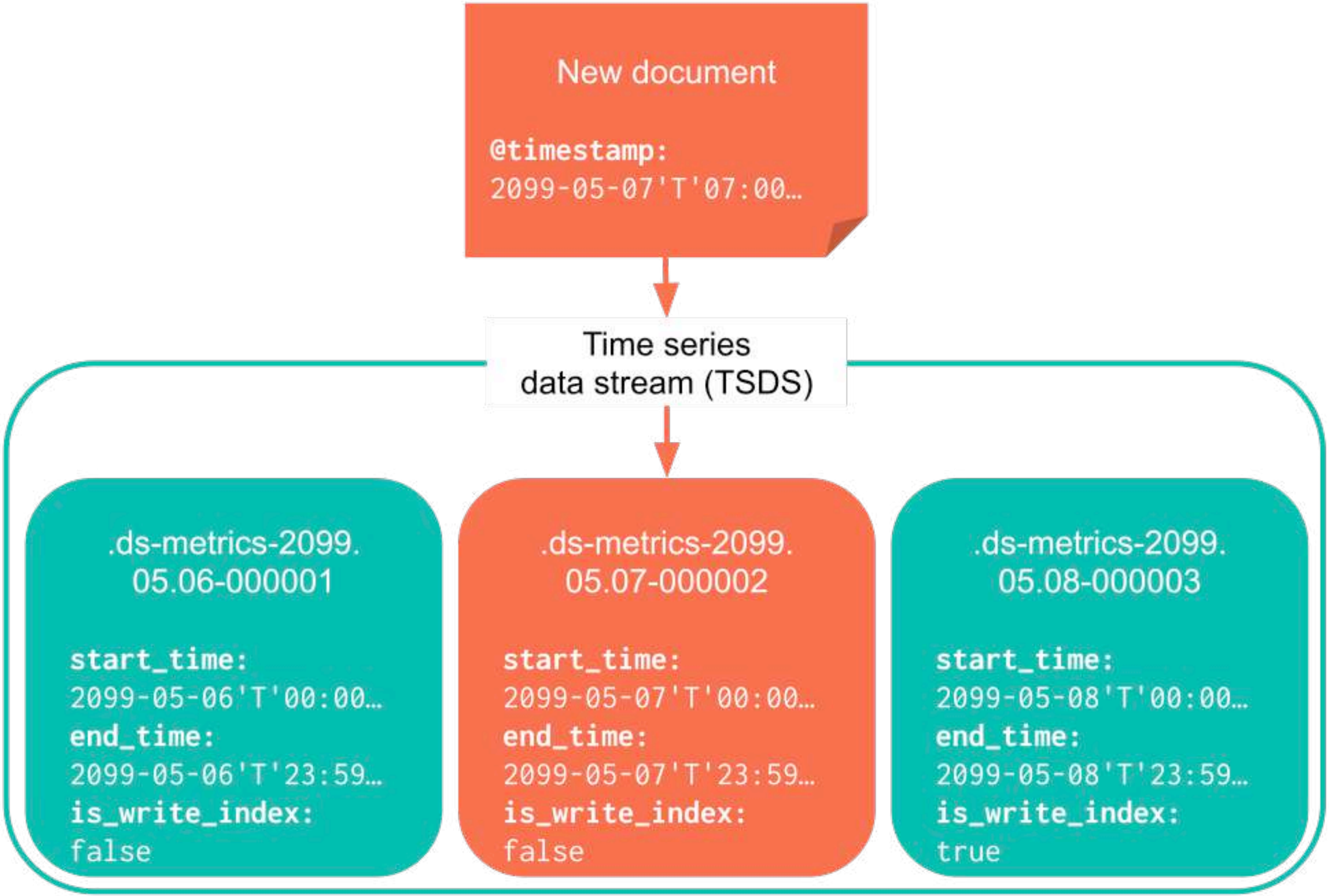
TSDS

Save disk

Use iff metrics in near real-time and `@timestamp` order

Supported Fields

`keyword`, `ip`, `byte`, `short`, `integer`, `long`, `unsigned_long`


New document
@timestamp:
2099-05-07'T'07:00…
Time series
data stream (TSDS)
.ds-metrics-2099.
05.06-000001
start_time:
2099-05-06'T'00:00…
end_time:
2099-05-06'T'23:59…
is_write_index:
false
.ds-metrics-2099.
05.07-000002
start_time:
2099-05-07'T'00:00…
end_time:
2099-05-07'T'23:59…
is_write_index:
false
.ds-metrics-2099.
05.08-000003
start_time:
2099-05-08'T'00:00…
end_time:
2099-05-08'T'23:59…
is_write_index:
true

index.look_ahead_time

Default 2h

Accepted ranges:
now - index.look_ahead_time to
now + index.look_ahead_time

Dimension Routing

Think custom routing

```
PUT _component_template/my-weather-sensor-mappings
{
  "template": {
    "mappings": {
      "properties": {
        "sensor_id": {
          "type": "keyword",
          "time_series_dimension": true
        },
        "location": {
          "type": "keyword",
          "time_series_dimension": true
        },
        "temperature": {
          "type": "half_float",
          "time_series_metric": "gauge"
        },
        "humidity": {
          "type": "half_float",
          "time_series_metric": "gauge"
        },
        "@timestamp": {
          "type": "date",
          "format": "strict_date_optional_time"
        }
      }
    }
  },
  "_meta": {
    "description": "Mappings for weather sensor data"
  }
}
```

```
PUT _component_template/my-weather-sensor-settings
{
  "template": {
    "settings": {
      "index.lifecycle.name": "my-lifecycle-policy",
      "index.look_ahead_time": "3h",
      "index.codec": "best_compression"
    }
  },
  "_meta": {
    "description": "Index settings for weather sensor data"
  }
}
```

```
PUT _index_template/my-weather-sensor-index-template
{
  "index_patterns": ["metrics-weather_sensors-*"],
  "data_stream": { },
  "template": {
    "settings": {
      "index.mode": "time_series",
      "index.routing_path": [ "sensor_id", "location" ]
    }
  },
  "composed_of": [ "my-weather-sensor-mappings", "my-weather-sensor-settings" ],
  "priority": 500,
  "_meta": {
    "description": "Template for my weather sensor data"
  }
}
```

```
PUT metrics-weather_sensors-dev/_bulk
{ "create":{ } }
{ "@timestamp": "2099-05-06T16:21:15.000Z", "sensor_id": "HAL-000001", "location": "plains", "temperature": 26.7,"humidity": 49.9 }
{ "create":{ } }
{ "@timestamp": "2099-05-06T16:25:42.000Z", "sensor_id": "SYKENET-000001", "location": "swamp", "temperature": 32.4, "humidity": 88.9 }
```

Index Sorting

Tradeoff: Ingestion overhead

Downsampling

aka rollup v2

Optimize Elasticsearch: Synthetic Source and TSDS

Philipp Krenn @xeraa