Serverless Observability: where SLOs meet transforms

Diana Todea - Site Reliability Engineer - Elastic
7th of February 2024

Site Reliability Engineer at Elastic for 2 years
Live in Valencia, Spain
Mother of 2 young children
Love traveling, reading and sports

UNSPLASH Juan Puyo

CONTEXT

The main issue our organisation faced when migrating to serverless was scaling. Due to the very large cross cluster search environment we needed our roll-up aggregations to support sucha complex environment.

Two methods of manipulating data:

-the rollup, which summarizes and stores historical data for later analysis (deprecated)

-the transforms, which convert existing Elasticsearch indices into summarized indices (current)

SUMMARY

SLO CONCEPTS

Service Level Indicators

A measure of the service level provided. Usually defined as a ratio of good over total events. Range between 0% and 100%. Examples: Availability, throughput, request latency, error rate.

Service Level Objectives

Target value for a service level measured by an SLI. Above the threshold the service is compliant. Example: 95% of the successful requests are served under 100ms.

Error budget

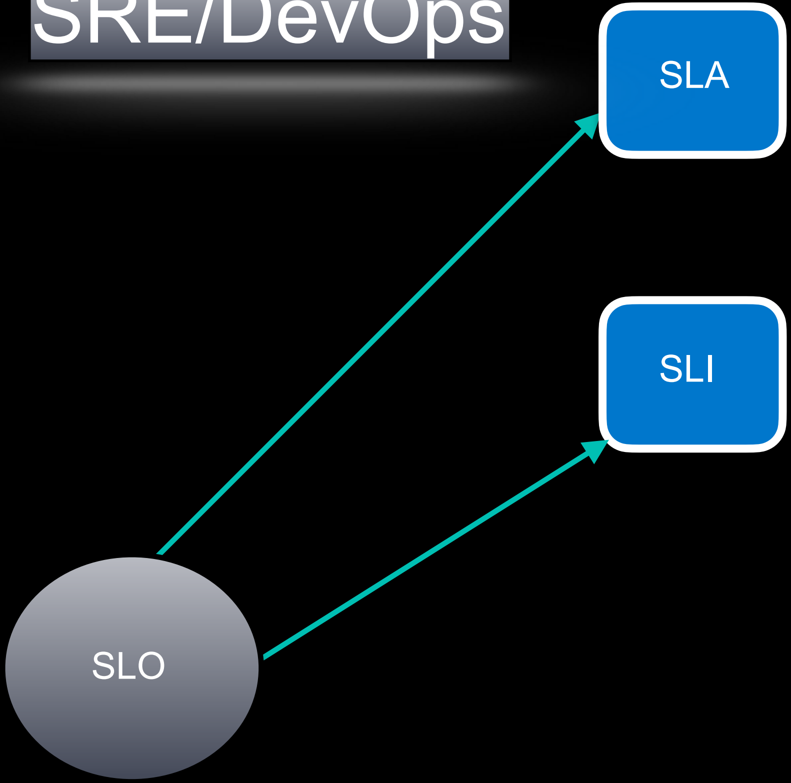
Defined as 100% minus the SLO. Quantity of errors that is tolerated.

Burn rate

The rate at which we are burning the error budget over a defined period of time. Very useful at alerting before exhausting the error budget.

Source: Google SRE book

SRE/DevOps



SLA is the agreement that your company makes with your clients.

SLIs represent the real numbers on your system’s performance.

SLOs are the goals your team must hit to meet the SLAs.

SLO is typically expressed as a percentage of time that a service is expected to meet a certain level of performance.

$$SLO = \frac{\text{Successful Operations}}{\text{Total Operations}} \times 100$$

Burn rate is a measure of how quickly a company is using up its cash reserves.

$$\text{Burn Rate} = \frac{\text{Initial Cash} - \text{Current Cash}}{\text{Time Period}}$$

Improve reliability and quality of services

GOOD SLO vs. BAD SLO

A bad SLO is **vague and subjective, it lacks quantifiable metrics, it has an undefined threshold and has no observation window**.

A good SLO is **specific and measurable, it's user-centric, it's quantifiable and achievable and it's timeframe defined.**

A well-defined SLO **focuses on a crucial aspect of service quality, provides clarity, measurability and alignment with user expectations, which are essential elements for effective monitoring and evaluation of service reliability**.

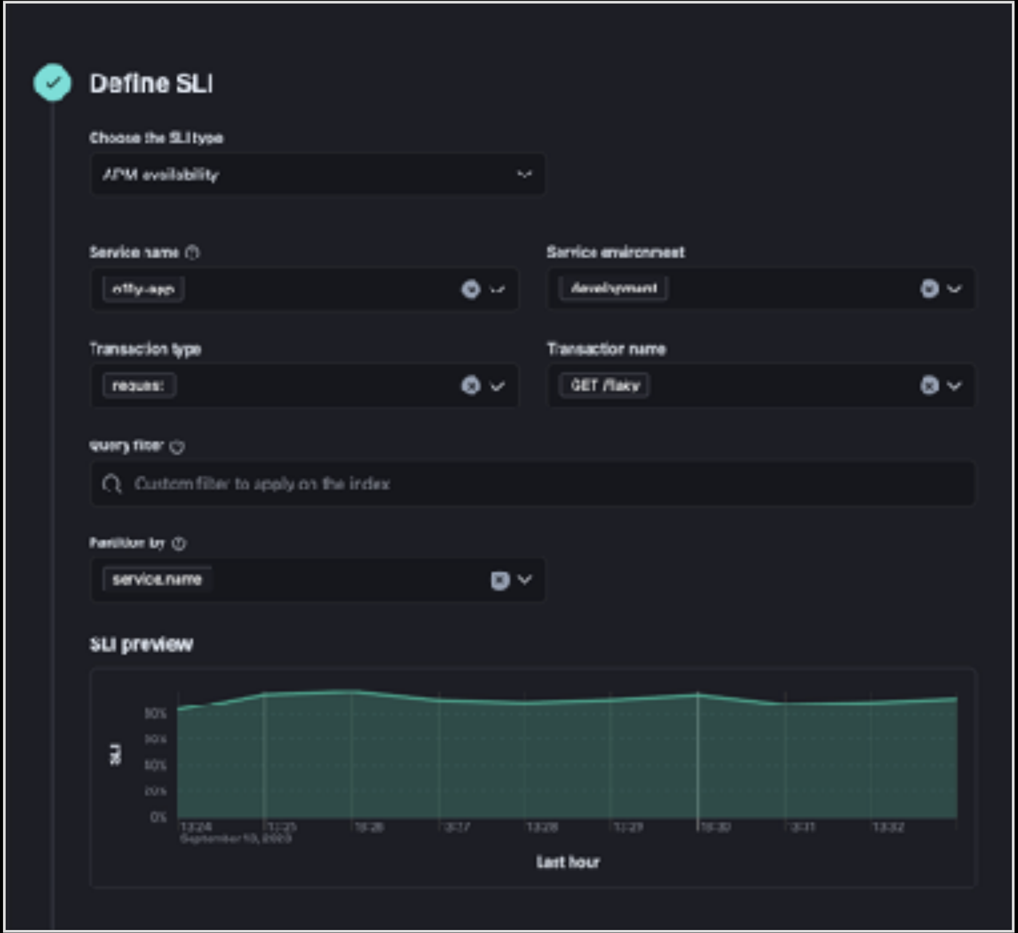
SLO INDICATOR TYPES

INDICATOR TYPES

- APM AVAILABILITY
- APM LATENCY
- CUSTOM KQL
- CUSTOM METRIC
- HISTOGRAM METRIC
- TIMESLICE METRICS

APM AVAILABILITY

- Designed to work exclusively with APM failed transaction rate
- Users can filter on service name, environment, transaction type and transaction name
- Support narrowing focus with a query filter
- Support partitioning by any field present in the APM data.
- Example SLOs: *99% of transactions are successful*



Define SLI

Choose the SLI type: APM availability

Service name: otty-app

Service environment: development

Transaction type: request

Transaction name: GET /flaky

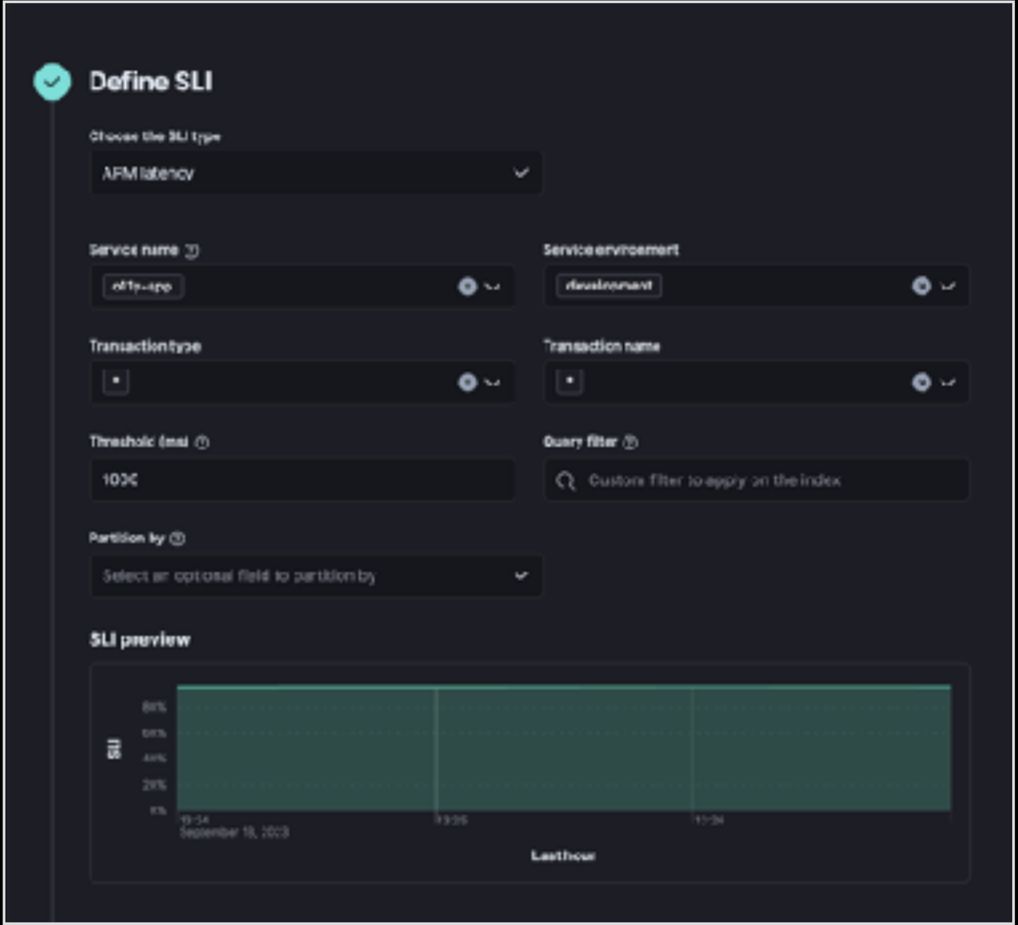
Query filter: Custom filter to apply on the index

Partition by: service.name

SLI preview

APM LATENCY

- Designed to work exclusively with APM transaction latency
- Users can filter on service name, environment, transaction type and transaction name
- Support narrowing focus with a query filter
- Support partitioning by any field present in the APM data.
- Example SLOs: *99% of transactions occur under 200ms*



Define SLI

Choose the SLI type: APM latency

Service name: olly-app

Service environment: development

Transaction type: *

Transaction name: *

Threshold (ms): 1000

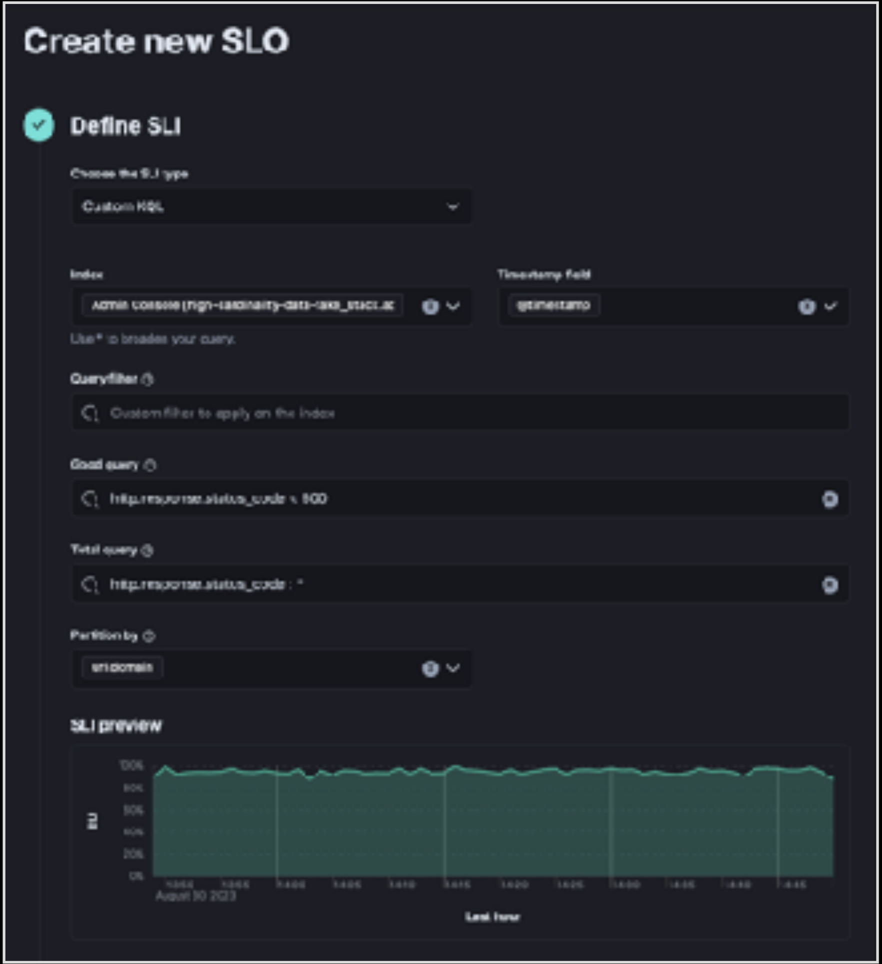
Query filter: Custom filter to apply on the index

Partition by: Select an optional field to partition by

SLI preview

CUSTOM KQL

- Designed to work exclusively with custom logs
- The document count for the good filter is used for the numerator
- The document count for the total filter is used for the denominator
- Support partitioning by any keyword present in the index pattern.
- Example SLOs: *99% of requests occur under 200ms, 99% of requests are successful, 99% of log messages are not errors*



Create new SLO

Define SLI

Choose the SLI type: Custom KQL

Index: Admin Console (high-cardinality-data-lake_stack...

Timestamp field: @timestamp

Use * to broaden your query.

Query filter: Custom filter to apply on the index

Good query: http.response.status_code < 500

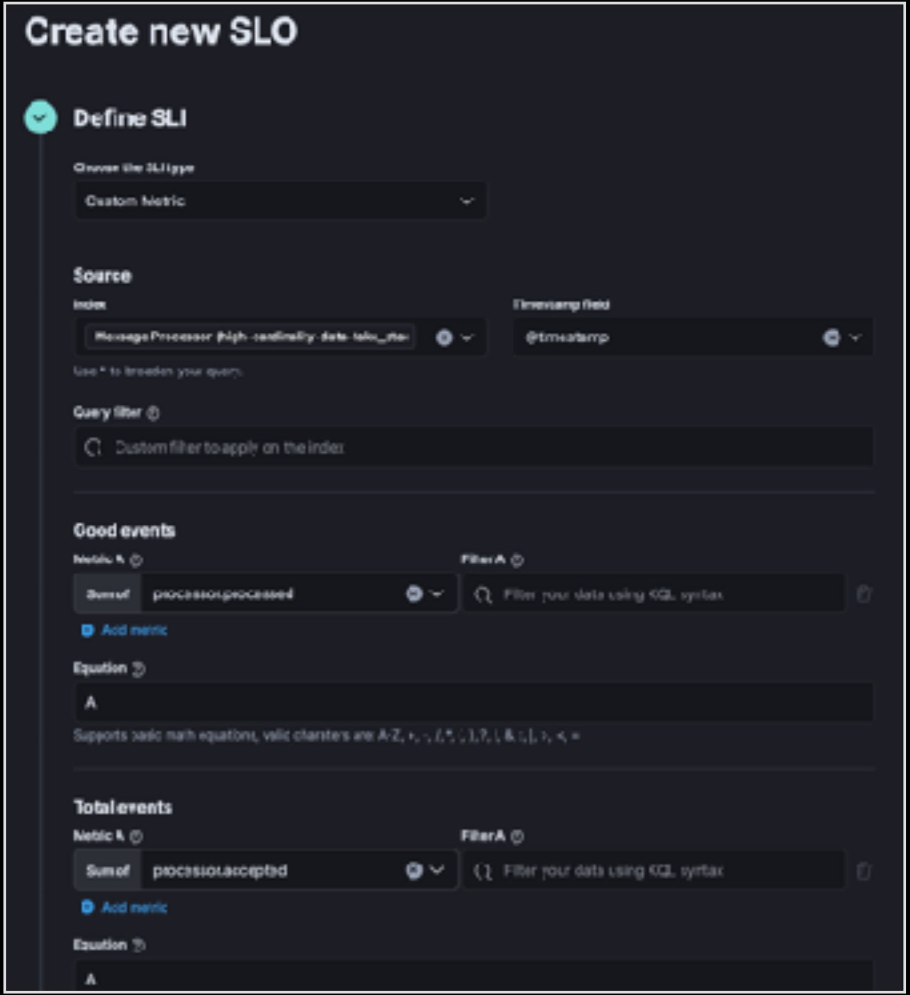
Total query: http.response.status_code : *

Partition by: url.domain

SLI preview

CUSTOM METRIC

- Designed to work with metric fields using a sum aggregation
- Supports basic math and boolean logic between multiple fields
- Support partitioning by any keyword present in the index pattern.
- Example SLOs: *99% of accepted events have been processed.*



Create new SLO

Define SLI

Choose the SLI type
Custom Metric

Source

Index: Message Processor (high-cardinality-data-take_me...
Timestamp field: @timestamp
Use * to broaden your query.

Query filter
Custom filter to apply on the index

Good events

Metric A: Sum of processor.processed
Filter A: Filter your data using KQL syntax
Add metric

Equation
A
Supports basic math equations, valid characters are: A-Z, +, -, /, *, (,), ?, !, &, :, |, >, <, =

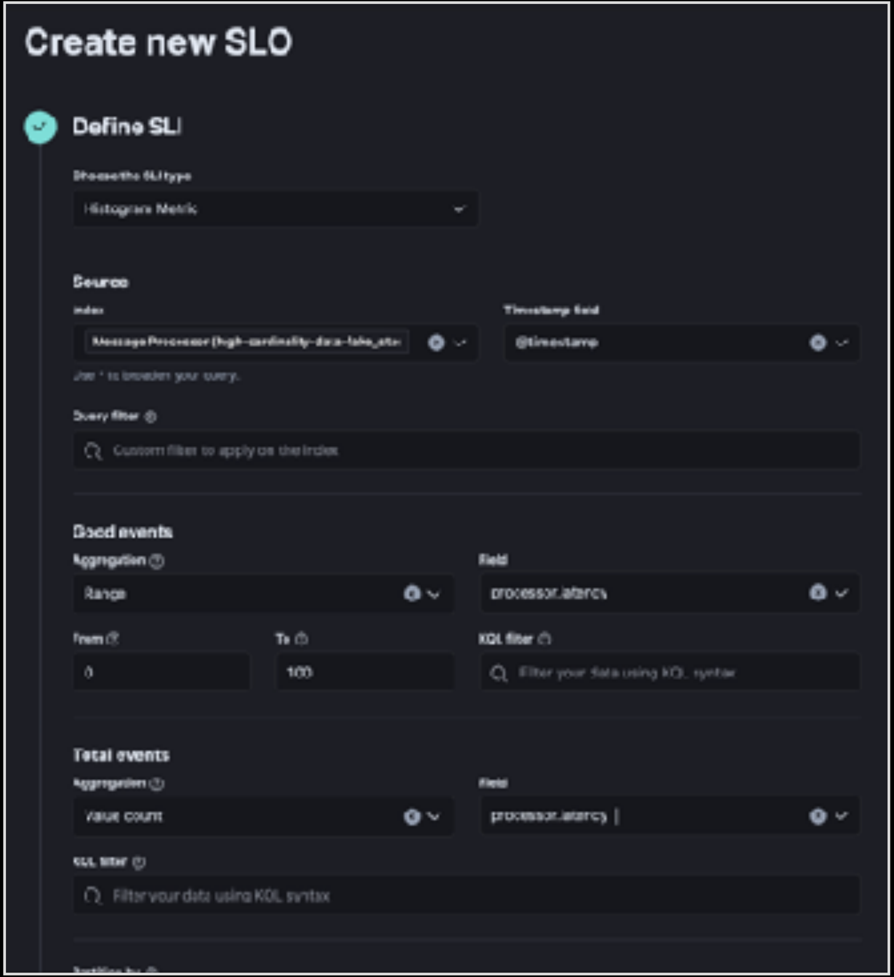
Total events

Metric A: Sum of processor.accepted
Filter A: Filter your data using KQL syntax
Add metric

Equation
A

HISTOGRAM METRIC

- Designed to work with histogram fields.
- Histogram fields are useful for storing pre-aggregated high resolution data with minimal footprint.
- Users can define a range for the values, the counts are used for the good/total events.
- Support partitioning by any keyword present in the index pattern.
- Example SLOs: *99% of events have a latency between 0ms and 100ms*


Create new SLO

Define SLI

Choose the SLI type: Histogram Metric

Source

Index: Message Processor (high-cardinality-data-fake_sta
Timestamp field: @timestamp

Use * to broaden your query.

Query filter: Custom filter to apply on the index

Good events

Aggregation: Range
Field: processor.latency
From: 0
To: 100
KQL filter: Filter your data using KQL syntax

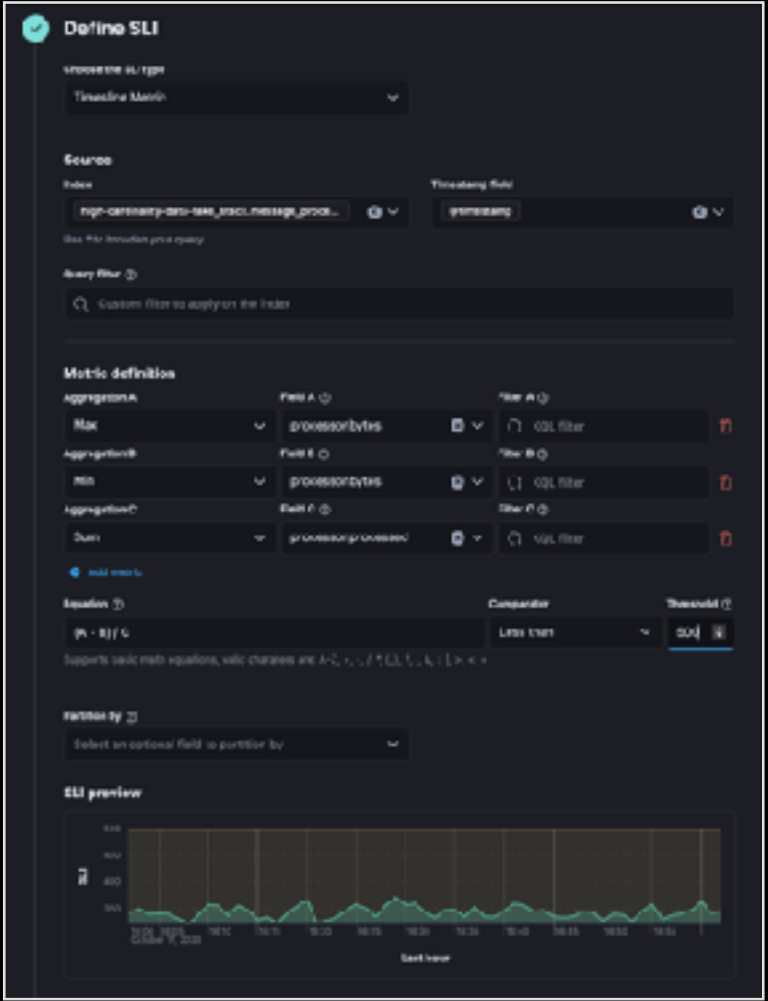
Total events

Aggregation: Value count
Field: processor.latency
KQL filter: Filter your data using KQL syntax

Partition by

TIMESLICE METRIC

- Supports multiple aggregations: average, min, max, sum, cardinality, last value, std. deviation, percentiles and document count.
- Due to the nature of the aggregations and their relationship with the bucket size, this indicator forces the user to use a timeslice budgeting method.
- Allows the user to define a custom equation with multiple aggregations along with a threshold for defining a “good slice”.



Define SLI

Choose the SLI type: Timeslice Metric

Source

Index	Timestamp field
high-cardinality-data-fake_stack.message_proce...	@timestamp

Query filter: Custom filter to apply on the index

Metric definition

Aggregation	Field	Filter
Aggregation A: Max	Field A: processor.bytes	Filter A: KQL filter
Aggregation B: Min	Field B: processor.bytes	Filter B: KQL filter
Aggregation C: Sum	Field C: processor.processed	Filter C: KQL filter

Add metric

Equation	Comparator	Threshold
(A - B) / C	Less than	500

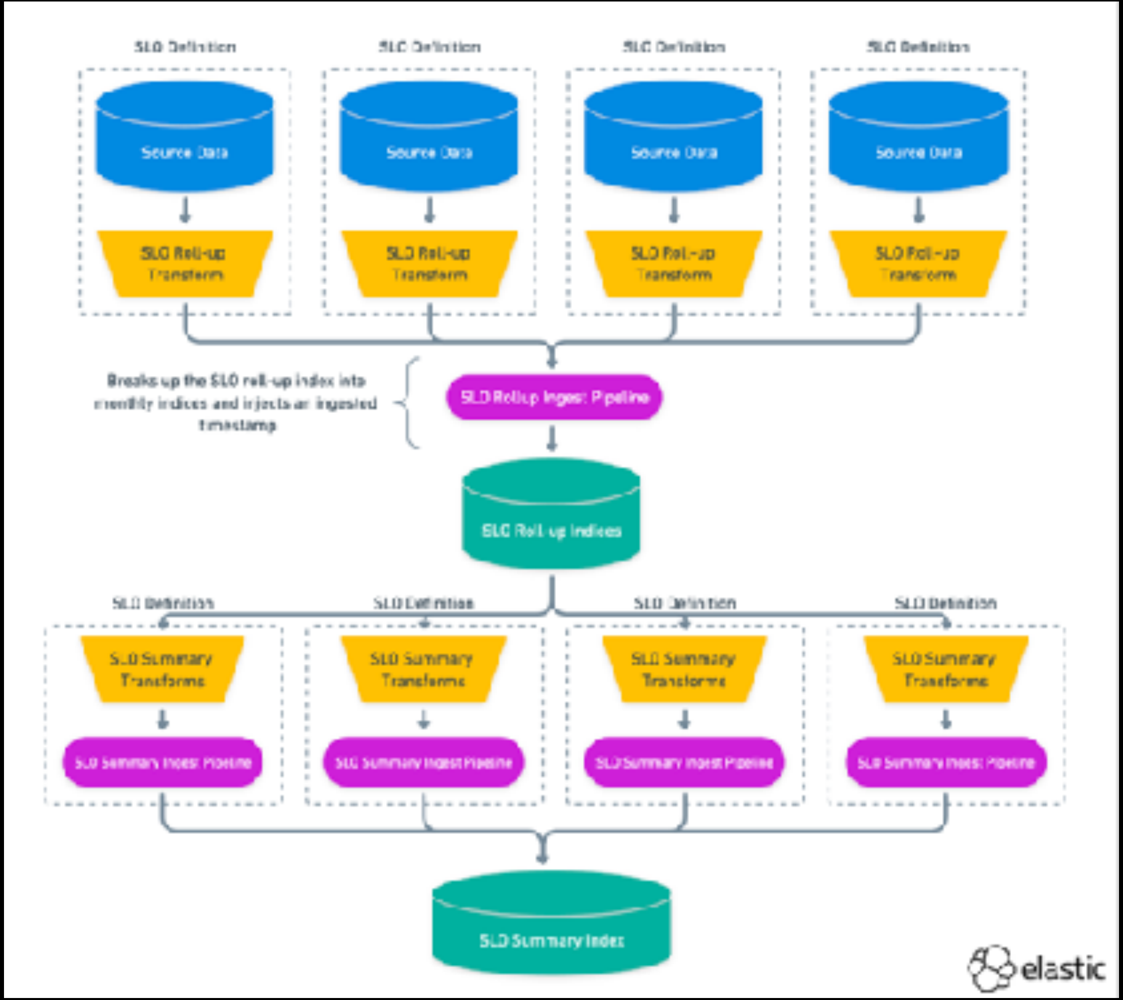
Partition by: Select an optional field to partition by

SLI preview

SLO ARCHITECTURE

SLOs rely on the **Transform** service to roll-up the source data into roll-up indices.

To support the group-by or the partition by feature we have added a second layer which summarizes the roll-up data into an entity-centric index for each SLO. This index also powers the search experience to allow users to search and sort by in any SLO dimension.



SLO Definition
Source Data
SLO Roll-up Transform
Breaks up the SLO roll-up index into monthly indices and injects an ingested timestamp
SLO Rollup Ingest Pipeline
SLO Roll-up Indices
SLO Definition
SLO Summary Transforms
SLO Summary Ingest Pipeline
SLO Summary Index
elastic

WHAT ARE TRANSFORMS?

Transforms are persistent tasks that enable you to convert existing Elasticsearch indices into summarized indices, which provide opportunities for new insights and analytics. For example, you can use transforms to pivot your data into entity-centric indices that summarize the behavior of users or sessions or other entities in your data. Or you can use transforms to find the latest document among all the documents that have a certain unique key.

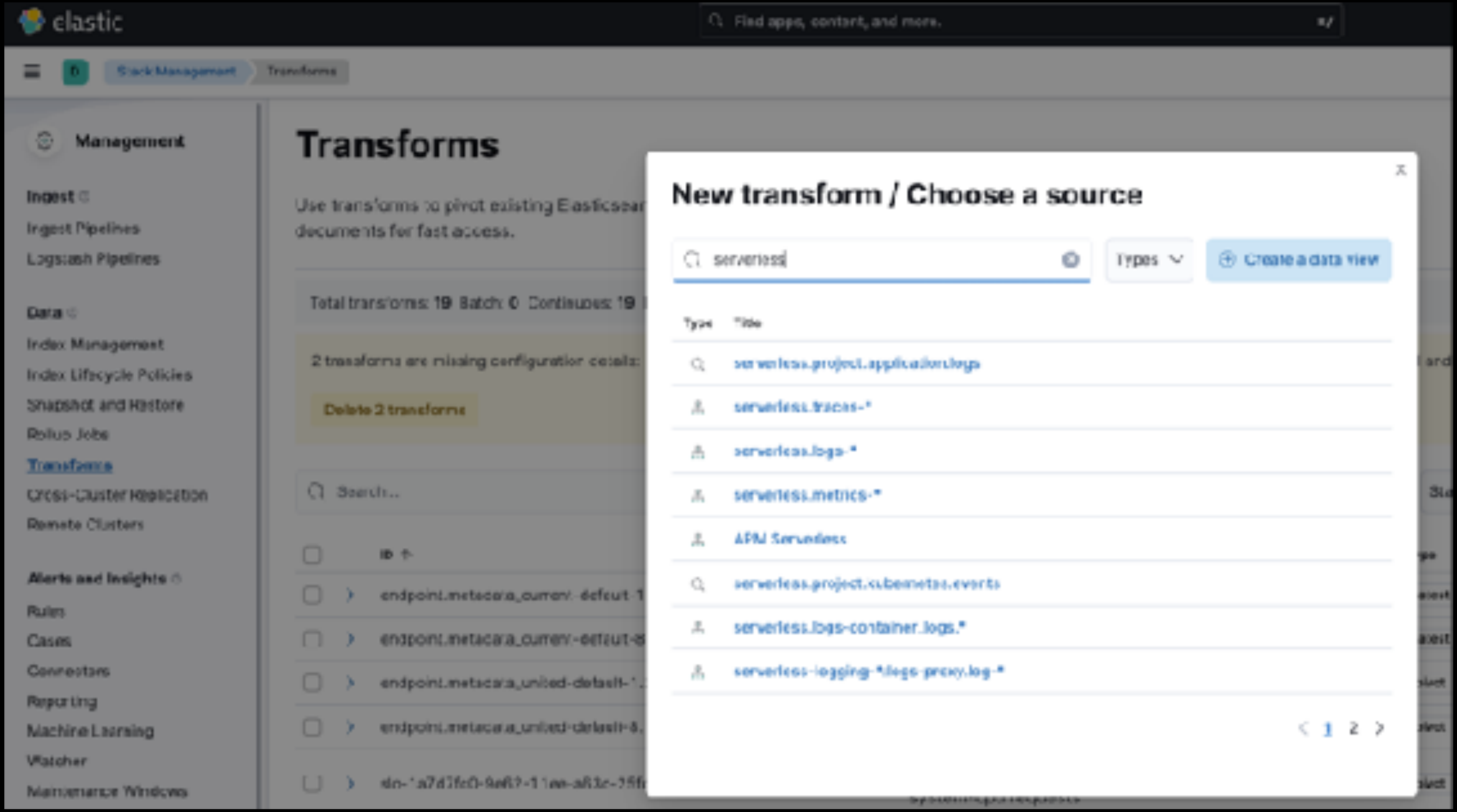
WHEN TO USE TRANSFORMS?

WHEN YOU NEED A COMPLETE **FEATURE INDEX** RATHER THAN A TOP-N SET OF ITEMS.

WHEN YOU NEED TO SORT AGRREGATION RESULTS BY A **PIPELINE AGGREGATION**.

WHEN YOU WANT TO CREATE **SUMMARY TABLES** TO OPTIMIZE QUERIES.

HOW TO CREATE TRANSFORMS?



elastic

Find apps, content, and more.

Stack Management › Transforms

Management

Ingest
- Ingest Pipelines
- Logstash Pipelines

Data
- Index Management
- Index Lifecycle Policies
- Snapshot and Restore
- Rollup Jobs
- Transforms
- Cross-Cluster Replication
- Remote Clusters

Alerts and Insights
- Rules
- Cases
- Connectors
- Reporting
- Machine Learning
- Watcher
- Maintenance Windows

Transforms

Use transforms to pivot existing Elasticsear
documents for fast access.

Total transforms: 19 Batch: 0 Continuous: 19

2 transforms are missing configuration details:

Delete 2 transforms

Search...

ID ↑
endpoint.metadata_current-default-1
endpoint.metadata_current-default-8
endpoint.metadata_united-default-1
endpoint.metadata_united-default-8
dn-1a7d7fc0-9a82-11ee-a83c-25f

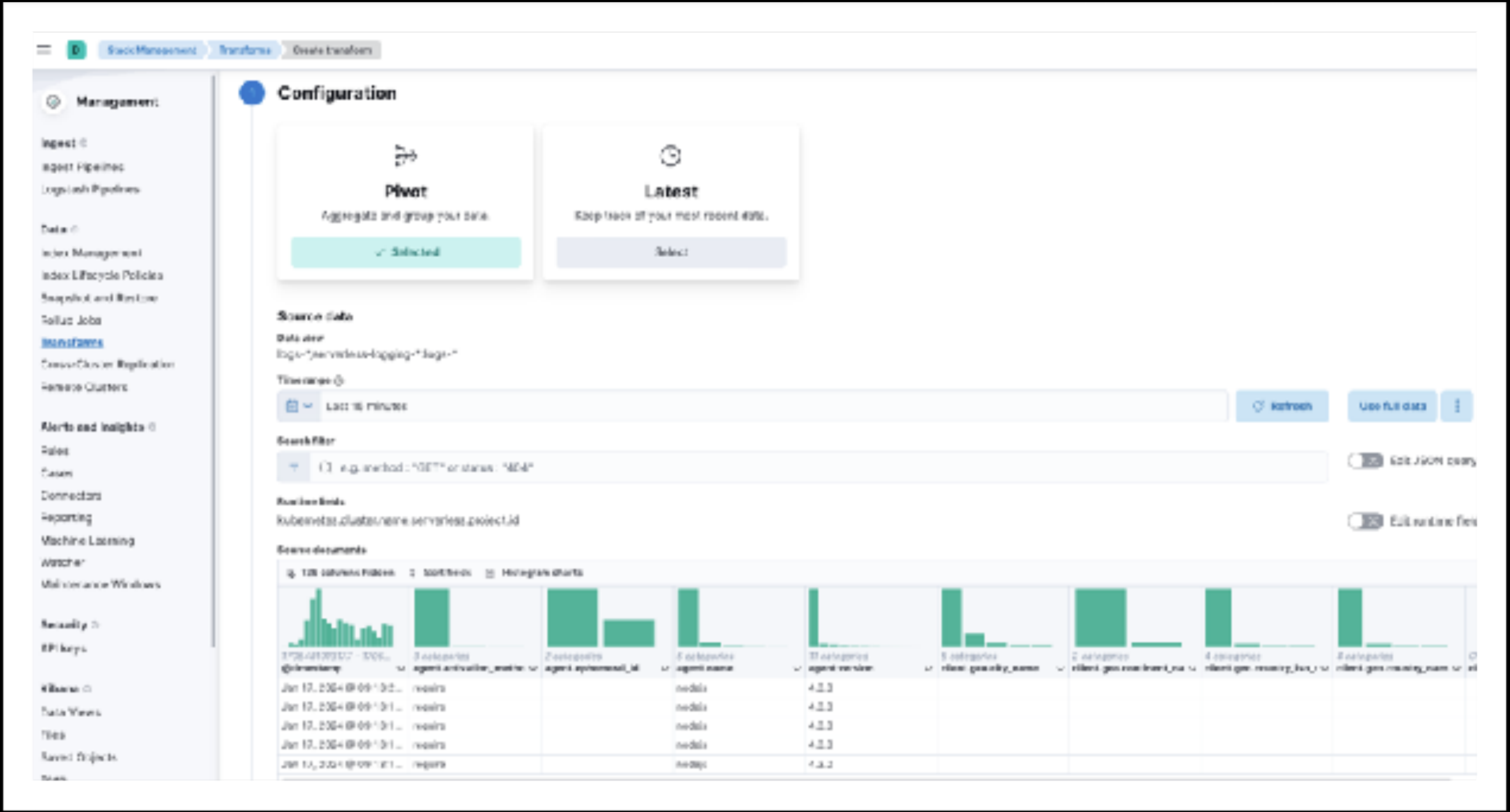
New transform / Choose a source

serverless

Types ⌄ | ⊕ Create a data view

Type	Title
🔍	serverless.project.application.logs
👤	serverless.traces-*
👤	serverless.logs-*
👤	serverless.metrics-*
👤	APM Serverless
🔍	serverless.project.kubernetes.events
👤	serverless.logs-container.logs.*
👤	serverless-logging-*/logs-proxy.log-*

‹ 1 2 ›



Stack Management > Transforms > Create transform

Management

Ingest
Ingest Pipelines
Logstash Pipelines

Data
Index Management
Index Lifecycle Policies
Snapshot and Restore
Rollup Jobs
Transforms
Cross-Cluster Replication
Remote Clusters

Alerts and Insights
Rules
Cases
Connectors
Reporting
Machine Learning
Watcher
Maintenance Windows

Security
API keys

Kibana
Data Views
Files
Saved Objects

Configuration

Pivot
Aggregate and group your data.
✓ Selected

Latest
Keep track of your most recent data.
Select

Source data

Data view
logs-*,serverless-logging-*,logs-*

Time range
Last 15 minutes
Refresh
Use full data

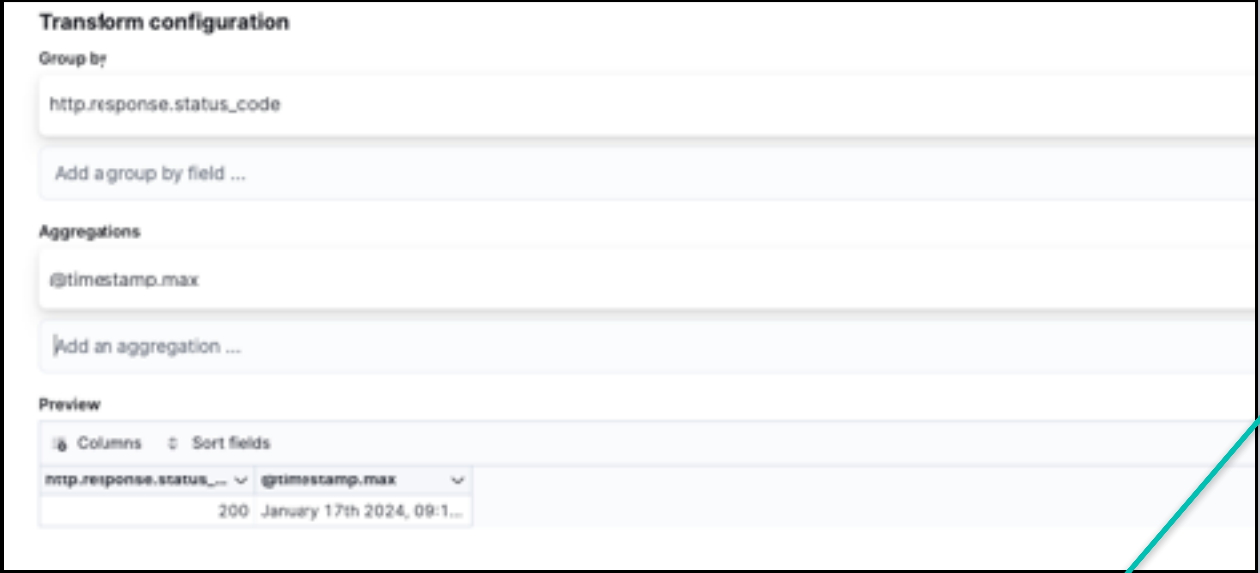
Search filter
e.g. method : "GET" or status : "404"
Edit JSON query

Runtime fields
kubernetes.cluster.name serverless.project.id
Edit runtime fields

Source documents

128 columns hidden · Sort fields · Histogram charts

@timestamp	agent.activation_method	agent.ephemeral_id	agent.name	agent.version	client.geo.city_name	client.geo.continent_na	client.geo.country_iso_c	client.geo.country_nam
Jan 17, 2024 @ 09:10:2…	require		nodejs	4.2.3				
Jan 17, 2024 @ 09:10:1…	require		nodejs	4.2.3				
Jan 17, 2024 @ 09:10:1…	require		nodejs	4.2.3				
Jan 17, 2024 @ 09:10:1…	require		nodejs	4.2.3				
Jan 17, 2024 @ 09:10:1…	require		nodejs	4.2.3				



Transform configuration

Group by

http.response.status_code

Add a group by field ...

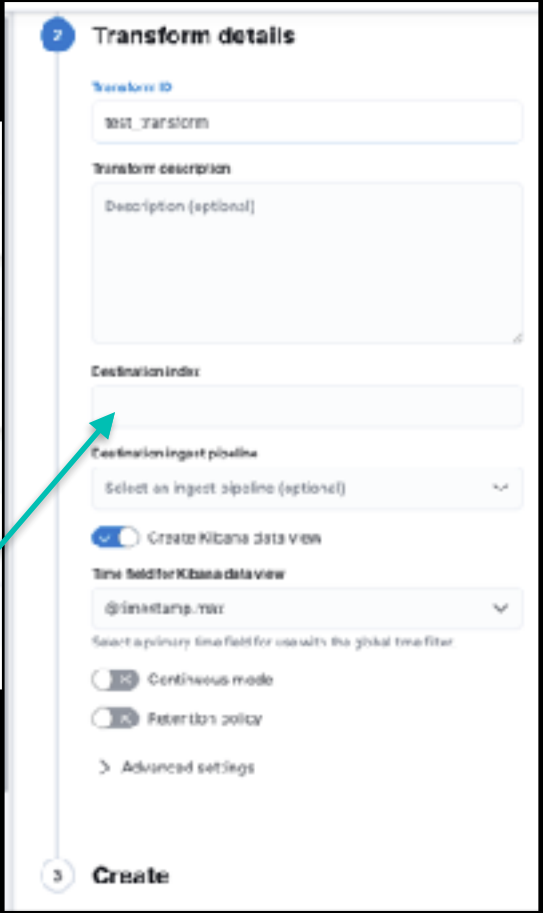
Aggregations

@timestamp.max

Add an aggregation ...

Preview

http.response.status_...	@timestamp.max
200	January 17th 2024, 09:1...



2 Transform details

Transform ID

test_transform

Transform description

Description (optional)

Destination index

Destination ingest pipeline

Select an ingest pipeline (optional)

Create Kibana data view

Time field for Kibana data view

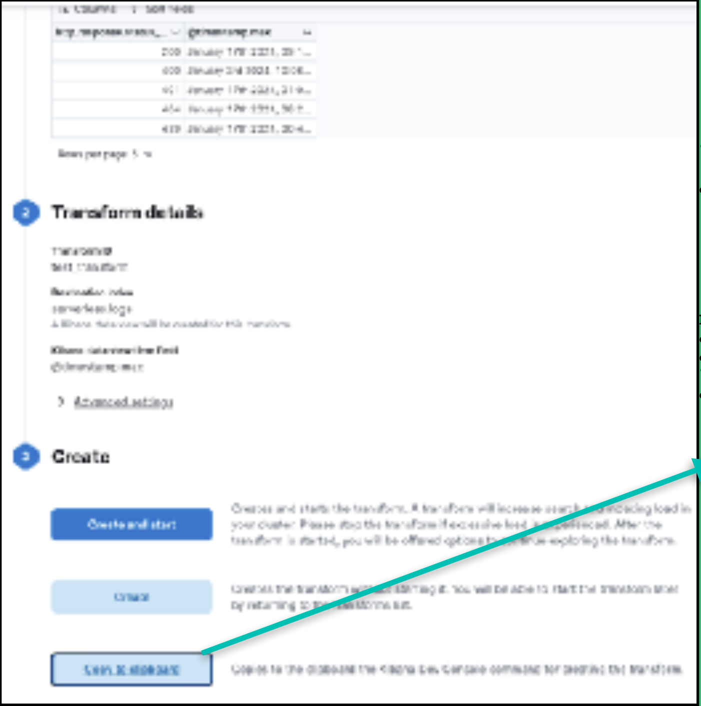
@timestamp.max

Select a primary time field for use with the global time filter.

Continuous mode

Retention policy

Add a destination index:
e.g. **serverless.logs**



Transform details

Transform ID
test_transform

Destination index
serverless.logs
A Kibana data view will be created for this transform

Kibana data view time field
@timestamp.max

Create

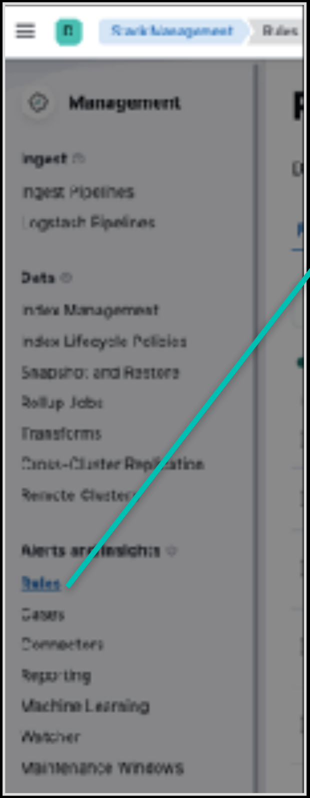
Create and start — Creates and starts the transform. A transform will increase search and indexing load in your cluster. Please stop the transform if excessive load is experienced. After the transform is started, you will be offered options to continue exploring the transform.

Create — Creates the transform without starting it. You will be able to start the transform later by returning to the transforms list.

Copy to clipboard — Copies to the clipboard the Kibana Dev Console command for creating the transform.

```
PUT _transform/test_transform
{
  "source": {
    "index": [
      "logs-*",
      "serverless-logging-*:logs-*"
    ],
    "runtime_mappings": {
      "kubernetes.cluster.name": {
        "script": {
          "source": "String k8sClusterNameLabel = 'labels.k8s_cluster_name';\nString orchestratorLabel =
'labels.orchestrator_cluster_name';\nString orchestratorField = 'orchestrator.cluster.name';\n\nif
(doc.containsKey(orchestratorField) && doc[orchestratorField].size() > 0) {\nemit(doc[orchestratorField].value);\n}
\nelse if (doc.containsKey(k8sClusterNameLabel) && doc[k8sClusterNameLabel].size() > 0)
{\nemit(doc[k8sClusterNameLabel].value);\n}\nelse if (doc.containsKey(orchestratorLabel) &&
doc[orchestratorLabel].size() > 0) {\nemit(doc[orchestratorLabel].value);\n}"
        },
        "type": "keyword"
      },
      "serverless.project.id": {
        "script": {
          "source": "String projectIdKey = 'project.id';\nString namespaceKey = 'kubernetes.namespace';\nString
handlingProjectKey = 'handling_project';\n\nif (doc.containsKey(projectIdKey) && doc[projectIdKey].size() > 0) {\n
emit(doc[projectIdKey].value);\n} else if (doc.containsKey(namespaceKey) && doc[namespaceKey].size() > 0 &&
doc[namespaceKey].value.indexOf('project-') == 0) {\n    emit(doc[namespaceKey].value.substring('project-'.length()));
\n} else if (doc.containsKey(handlingProjectKey) && doc[handlingProjectKey].size() > 0) {\n
emit(doc[handlingProjectKey].value);\n}"
        },
        "type": "keyword"
      }
    }
  },
  "pivot": {
    "group_by": {
      "http.response.status_code": {
        "terms": {
          "field": "http.response.status_code"
        }
      }
    },
    "aggregations": {
      "@timestamp.max": {
        "max": {
          "field": "@timestamp"
        }
      }
    }
  },
  "dest": {
    "index": "serverless.logs"
  },
  "settings": {
    "num_failure_retries": 3
  }
}
```

HOW TO GENERATE ALERTS FOR TRANSFORMS?



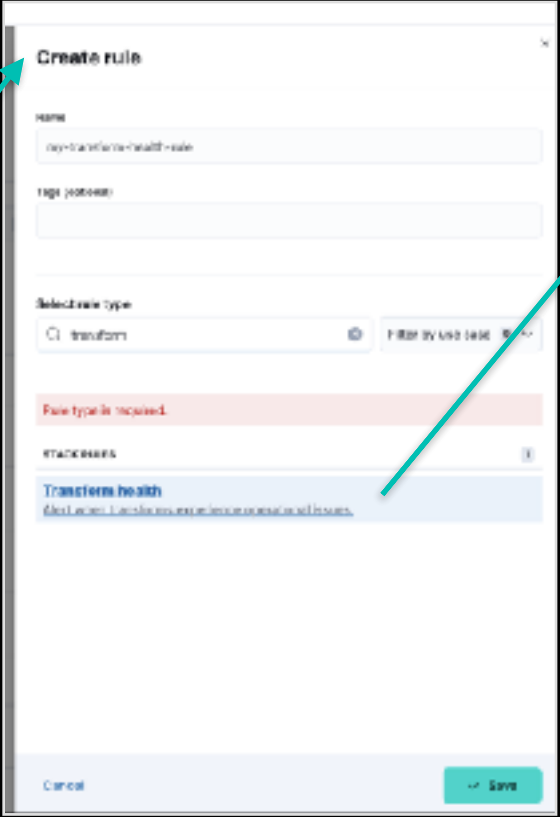
Stack Management > Rules

Management

Ingest
Ingest Pipelines
Logstash Pipelines

Data
Index Management
Index Lifecycle Policies
Snapshot and Restore
Rollup Jobs
Transforms
Cross-Cluster Replication
Remote Clusters

Alerts and Insights
Rules
Cases
Connectors
Reporting
Machine Learning
Watcher
Maintenance Windows



Create rule

Name: my-transform-health-rule

Tags (optional):

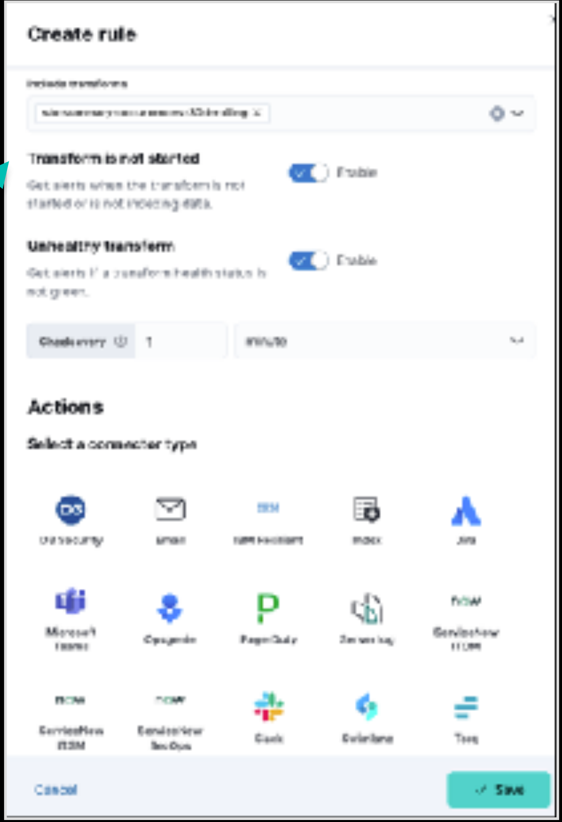
Select rule type: transform | Filter by use case

Rule type is required.

STACK RULES

Transform health
Alert when transforms experience operational issues.

Cancel | Save



Create rule

Include transforms: [illegible]

Transform is not started — Enable
Get alerts when the transform is not started or is not indexing data.

Unhealthy transform — Enable
Get alerts if a transform health status is not green.

Check every: 1 minute

Actions

Select a connector type

The SLOs APIs can be used to manage your SLOs using your infrastructure as code system (e.g. Terraform)

PUT _transform/<transform_id>

GET _transform/<transform_id>

DELETE _transform/<transform_id>

GET _transform/_all

*GET _transform/**

GET _transform/_stats

GET _transform/_all/_stats

GET _transform//_stats*

GET _transform/<transform_id>/_preview

POST _transform/<transform_id>/_reset

POST _transform/<transform_id>/_schedule_now

POST _transform/<transform_id>/_start

POST _transform/<transform_id>/_stop

POST _transform/<transform_id>/_update

POST _transform/_upgrade

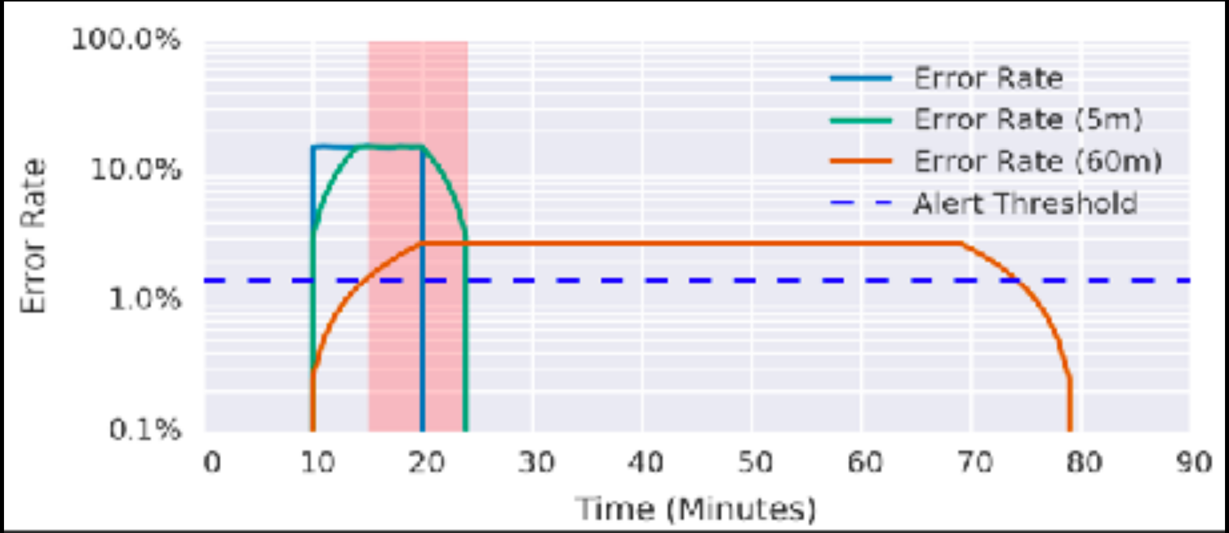
WHAT IS BURN RATE ALERTING?

Burn rate alerting calculates the rate at which SLOs are failing over multiple windows of time.

Burn rate alerting is less sensitive to short term fluctuations by focusing on sustained deviations.

Burn rate alerting can give you an indication of how severly the service is degrading and it helps prioritize multiple issues at the same time.

BURN RATES ALERTING WITH MULTIPLE WINDOWS



There are 2 windows for each severity: a short and a long.
The short window is 1/12 of the long window.

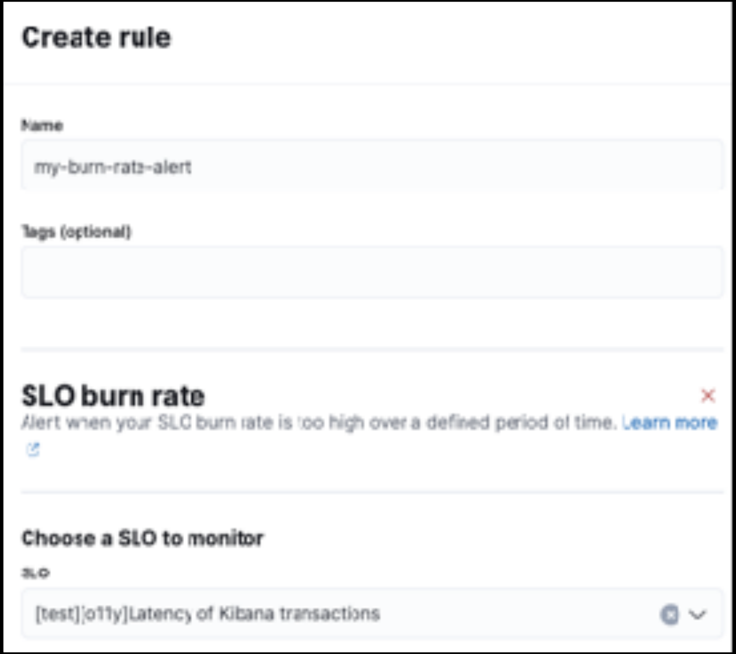
When the burn rate for both windows exceeds the threshold, the alert is triggered.

PROs: Reduced alert fatigue, improved user experience, a flexible alerting framework, good precision

CONs: Lots of options to configure

HOW TO CREATE A BURN RATE ALERT

How long you have until your error budget is exhausted.



Create rule

Name

my-burn-rate-alert

Tags (optional)

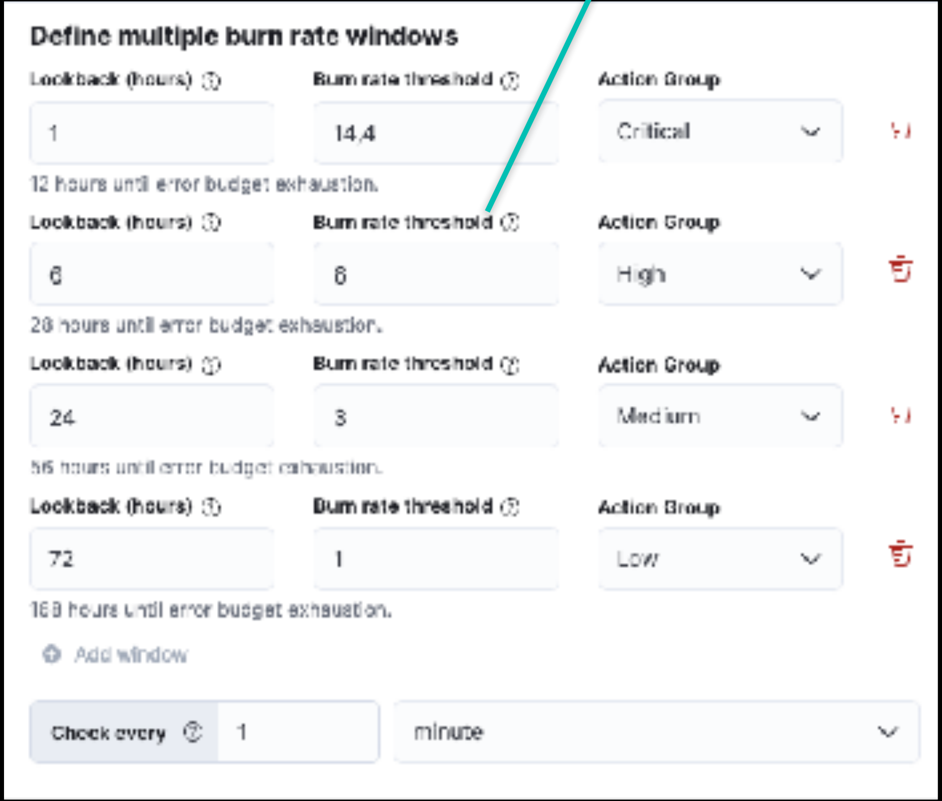
SLO burn rate

Alert when your SLO burn rate is too high over a defined period of time. Learn more

Choose a SLO to monitor

SLO

[test]]o11y]Latency of Kibana transactions



Define multiple burn rate windows

Lookback (hours)	Burn rate threshold	Action Group
1	14.4	Critical
6	6	High
24	3	Medium
72	1	Low

12 hours until error budget exhaustion.

28 hours until error budget exhaustion.

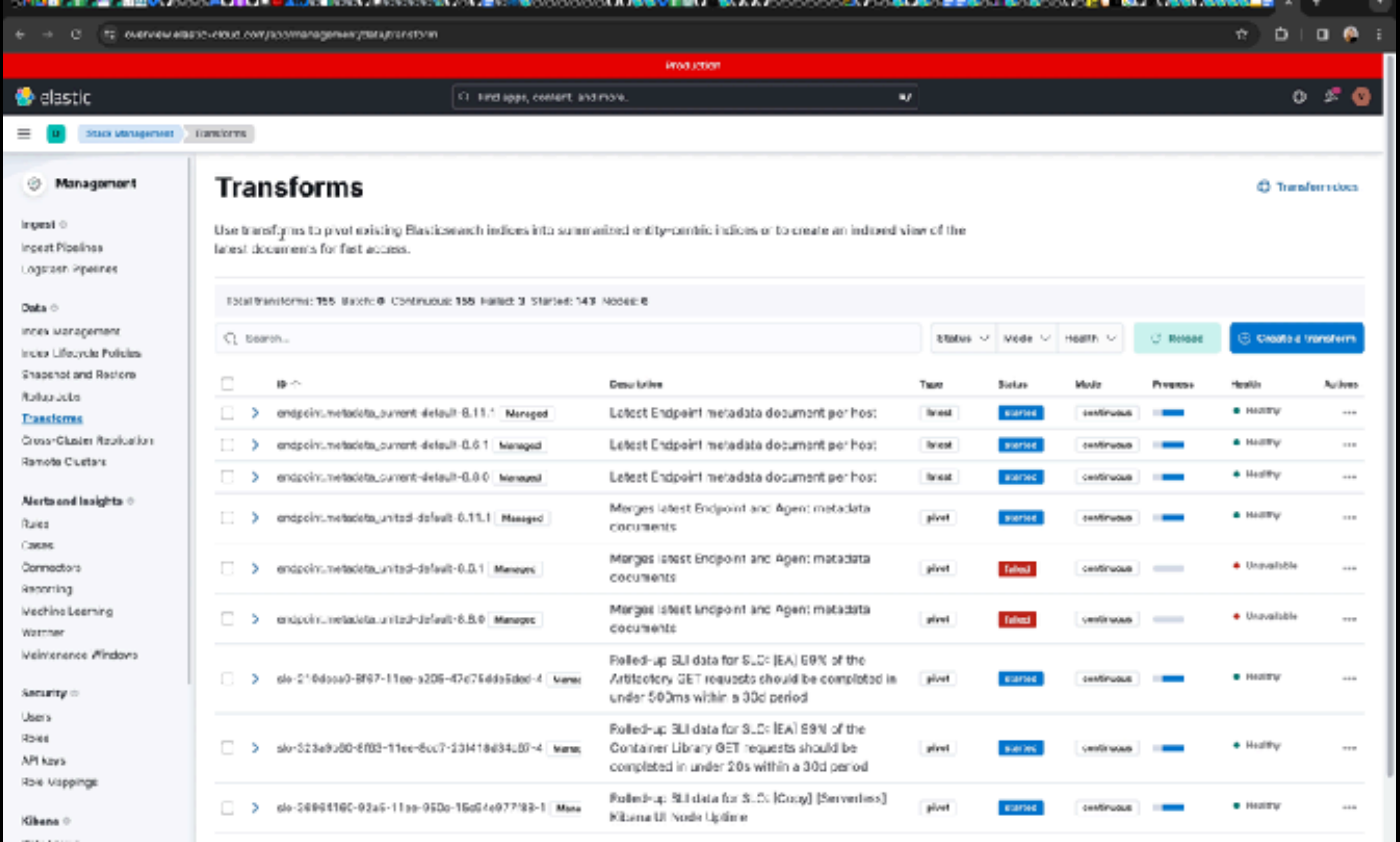
56 hours until error budget exhaustion.

168 hours until error budget exhaustion.

Add window

Check every 1 minute

DEMO TIME



Production

elastic

Stack Management > Transforms

Management

Ingest
Ingest Pipelines
Logstash Pipelines

Data
Index Management
Index Lifecycle Policies
Snapshot and Restore
Rollup Jobs
Transforms
Cross-Cluster Replication
Remote Clusters

Alerts and Insights
Rules
Cases
Connectors
Reporting
Machine Learning
Watcher
Maintenance Windows

Security
Users
Roles
API keys
Role Mappings

Kibana

Transforms

Transform docs

Use transforms to pivot existing Elasticsearch indices into summarized entity-centric indices or to create an indexed view of the latest documents for fast access.

Total transforms: 155 Batch: 0 Continuous: 155 Failed: 3 Started: 143 Nodes: 6

Search... Status Mode Health Reload Create a transform

ID	Description	Type	Status	Mode	Progress	Health	Actions
endpoint.metadata_current-default-8.11.1 Managed	Latest Endpoint metadata document per host	latest	started	continuous		● Healthy	···
endpoint.metadata_current-default-8.6.1 Managed	Latest Endpoint metadata document per host	latest	started	continuous		● Healthy	···
endpoint.metadata_current-default-8.8.0 Managed	Latest Endpoint metadata document per host	latest	started	continuous		● Healthy	···
endpoint.metadata_united-default-8.11.1 Managed	Merges latest Endpoint and Agent metadata documents	pivot	started	continuous		● Healthy	···
endpoint.metadata_united-default-8.6.1 Managed	Merges latest Endpoint and Agent metadata documents	pivot	failed	continuous		● Unavailable	···
endpoint.metadata_united-default-8.8.0 Managed	Merges latest Endpoint and Agent metadata documents	pivot	failed	continuous		● Unavailable	···
slo-210dcaa0-8f67-11ee-a206-47c75dda5ded-4 Manag	Rolled-up SLI data for SLO: [EA] 99% of the Artifactory GET requests should be completed in under 500ms within a 30d period	pivot	started	continuous		● Healthy	···
slo-323a9b80-8f03-11ee-8cc7-2314184a4a07-4 Manag	Rolled-up SLI data for SLO: [EA] 99% of the Container Library GET requests should be completed in under 20s within a 30d period	pivot	started	continuous		● Healthy	···
slo-56964160-93a6-11ee-950a-16a64e977183-1 Mana	Rolled-up SLI data for SLO: [Copy] [Serverless] Kibana UI Node Uptime	pivot	started	continuous		● Healthy	···

SLOS AND INCIDENT MANAGEMENT

DEVELOPERS

CUSTOMER SUPPORT/
EXPERIENCE TEAMS

SRE/DevOps

SLOS

DATA SCIENTISTS

PRODUCT MANAGERS

SRE/DEVOPS AND INCIDENT MANAGEMENT

- ✓ UNDERSTAND USER EXPECTATIONS
- ✓ DEFINE KEY SERVICE LEVEL INDICATORS
- ✓ QUANTIFY SERVICE LEVEL INDICATORS
- ✓ CONSIDER USER IMPACT
- ✓ BALANCE RELIABILITY AND INNOVATION
- ✓ USE HISTORICAL DATA FOR INFORMED DECISIONS
- ✓ ITERATE AND REFINE
- ✓ MONITOR AND ALERT
- ✓ DOCUMENT AND COMMUNICATE
- ✓ COLLABORATE ACROSS TEAMS

HAPPY TRANSFORM, HAPPY SLO

Photo by Nathan Dumlao on Unsplash

RESOURCES

1. https://www.elastic.co/guide/en/observability/current/slo.html
2. https://www.elastic.co/guide/en/observability/current/slo-create.html
3. https://www.elastic.co/guide/en/elasticsearch/reference/current/transforms.html
4. https://www.elastic.co/guide/en/elasticsearch/reference/current/transform-apis.html
5. https://docs.elastic.co/api-reference/observability/findslosop
6. https://www.elastic.co/guide/en/elasticsearch/reference/current/transform-scale.html
7. https://www.elastic.co/guide/en/elasticsearch/reference/current/transform-examples.html
8. https://www.elastic.co/guide/en/observability/current/slo-burn-rate-alert.html
9. https://www.elastic.co/guide/en/observability/current/observability-introduction.html
10. https://openapi.tools/

THANK YOU

https://github.com/didiViking
https://www.linkedin.com/in/diana-todea-b2a79968/