Data Streams feature in OpenSearch

Emil Kleszcz

it-opensearch-experts@cern.ch

Agenda

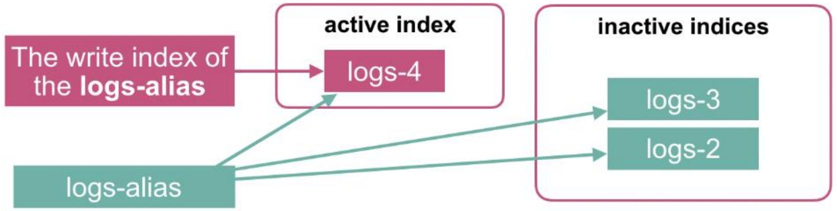
- Background
- What are Data Streams?
- When to use them?
- Why to use them?
- How can I use them?
- Demo
- Migrate your current data
- Summary

Background

- **Time series data** refer to data points recorded and organized in chronological order

- **Classic way of ingesting time-series data** with indices:
  - Create an index template (like a scheme in SQL)
  - Set up an ISM policy (e.g. retention)
  - Define a write alias and handle manually rollovers
  - Ingest data, splitting it across multiple indices over time

- **Challenges**:
  - Complex setup: requires configuring ISM, aliases, and templates
  - Manual oversight: error-prone rollover and lifecycle management
  - Scalability: high-throughput data ingestion needs extensive tuning



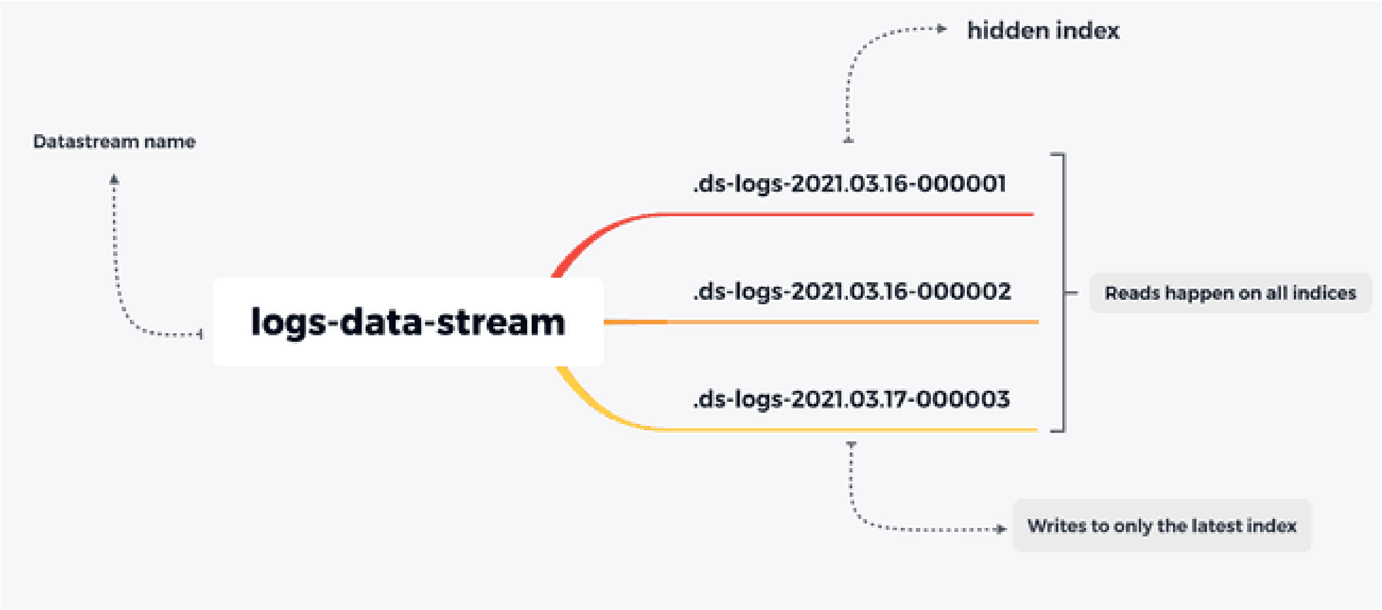
Background

- **Time-series data is voluminous and snowballs quickly**
- For example: For an average user, syslogs from MacBook might be anything from the range of 300MB-1GB per day.
  - Now multiply it with the number of days, it starts to look prominent…

- There are two ways of storing and managing time-series data in OS:
  - **Indexes**
    - Independent units requiring manual setup and management
  - **Data streams**
    - Automated abstraction managing time-based indices seamlessly

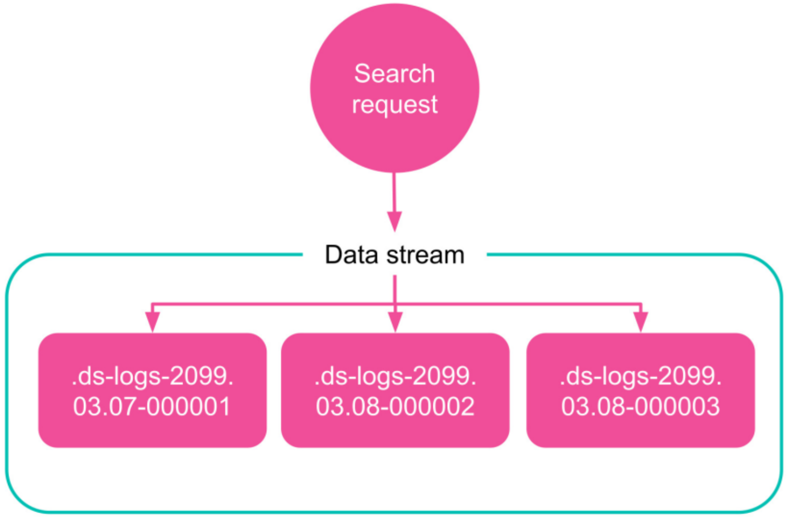
What are OpenSearch Data Streams?

- Designed for indexing and querying time-series append-only data
  - typically logs, metrics, or observability data
- Collection of hidden automatically generated indices
- Rolls over the index automatically based on the ISM policy

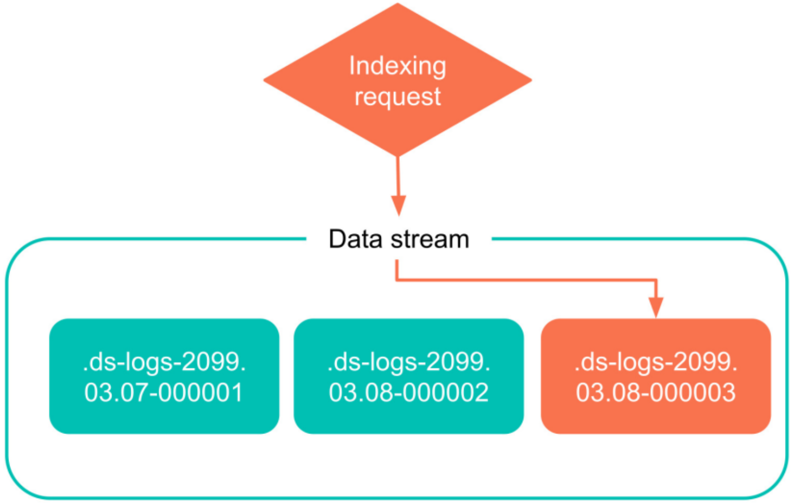

hidden index
Datastream name
.ds-logs-2021.03.16-000001
.ds-logs-2021.03.16-000002
.ds-logs-2021.03.17-000003
logs-data-stream
Reads happen on all indices
Writes to only the latest index

What are Data Streams?

- Made of a list of hidden indices (**backing indices**)
- **Read requests** are automatically routed to the proper backing indices
- **Write requests** are routed to the write index (most recent backing index) only



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



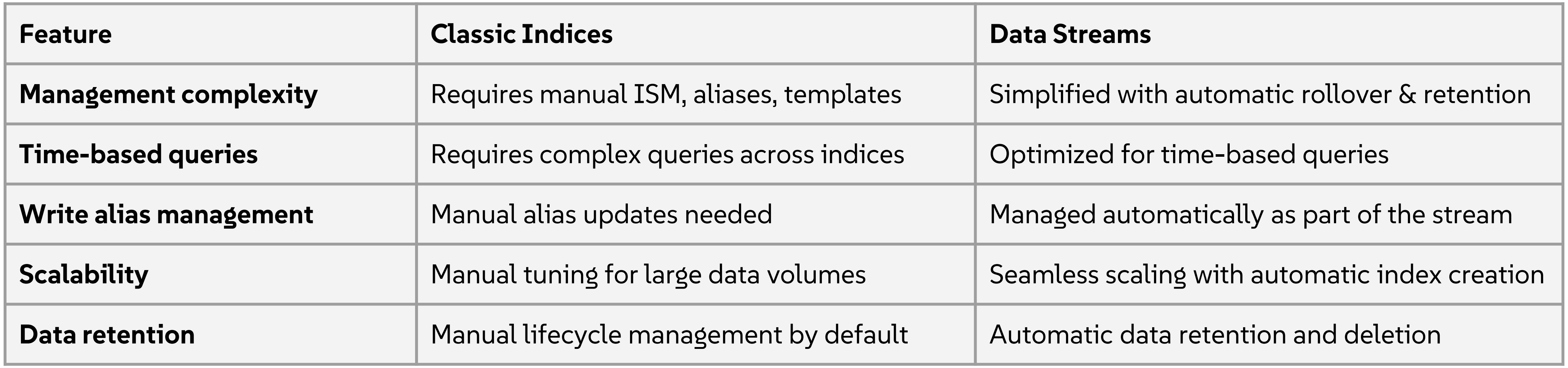
Indexing request
Data stream
.ds-logs-2099.03.07-000001
.ds-logs-2099.03.08-000002
.ds-logs-2099.03.08-000003

When to use Data Streams?

- **Time-series data**: logs, metrics, and traces
- For **append-only** logs with timestamp!
- Examples of logs:
  - Application (e.g. Nginx, Apache, logstash, systemd…)
  - Audit
  - Metrics (Prometheus sending to OpenSearch, etc.)
  - Stream of events (IoT, app telemetry, etc.)
  - System logs
- **When not to use?**
  - Mutable data
  - Non-time-series (no timestamp)
  - Very small static data volumes
  - Highly customised index management

Why use Data Streams?

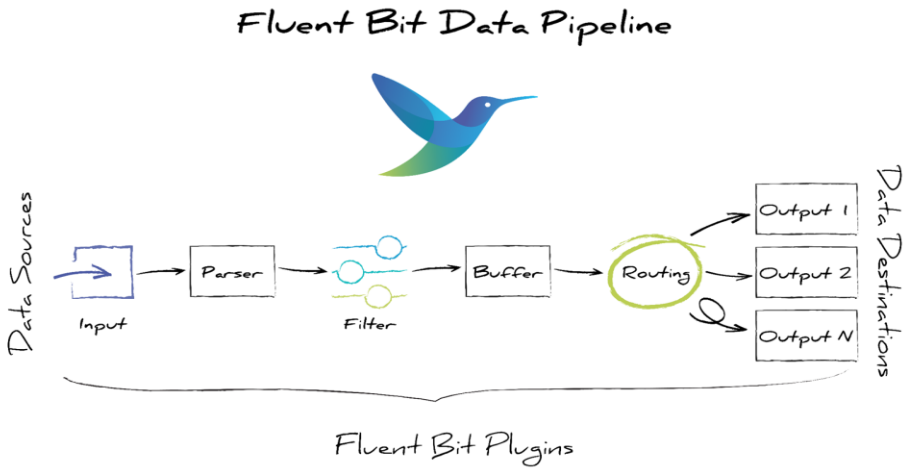
- **Previously:** creating a rollover index alias, defining a write index, and defining common mappings and settings for the backing indices



Feature	**Classic Indices**	**Data Streams**
Management complexity	Requires manual ISM, aliases, templates	Simplified with automatic rollover & retention
Time-based queries	Requires complex queries across indices	Optimized for time-based queries
Write alias management	Manual alias updates needed	Managed automatically as part of the stream
Scalability	Manual tuning for large data volumes	Seamless scaling with automatic index creation
Data retention	Manual lifecycle management by default	Automatic data retention and deletion

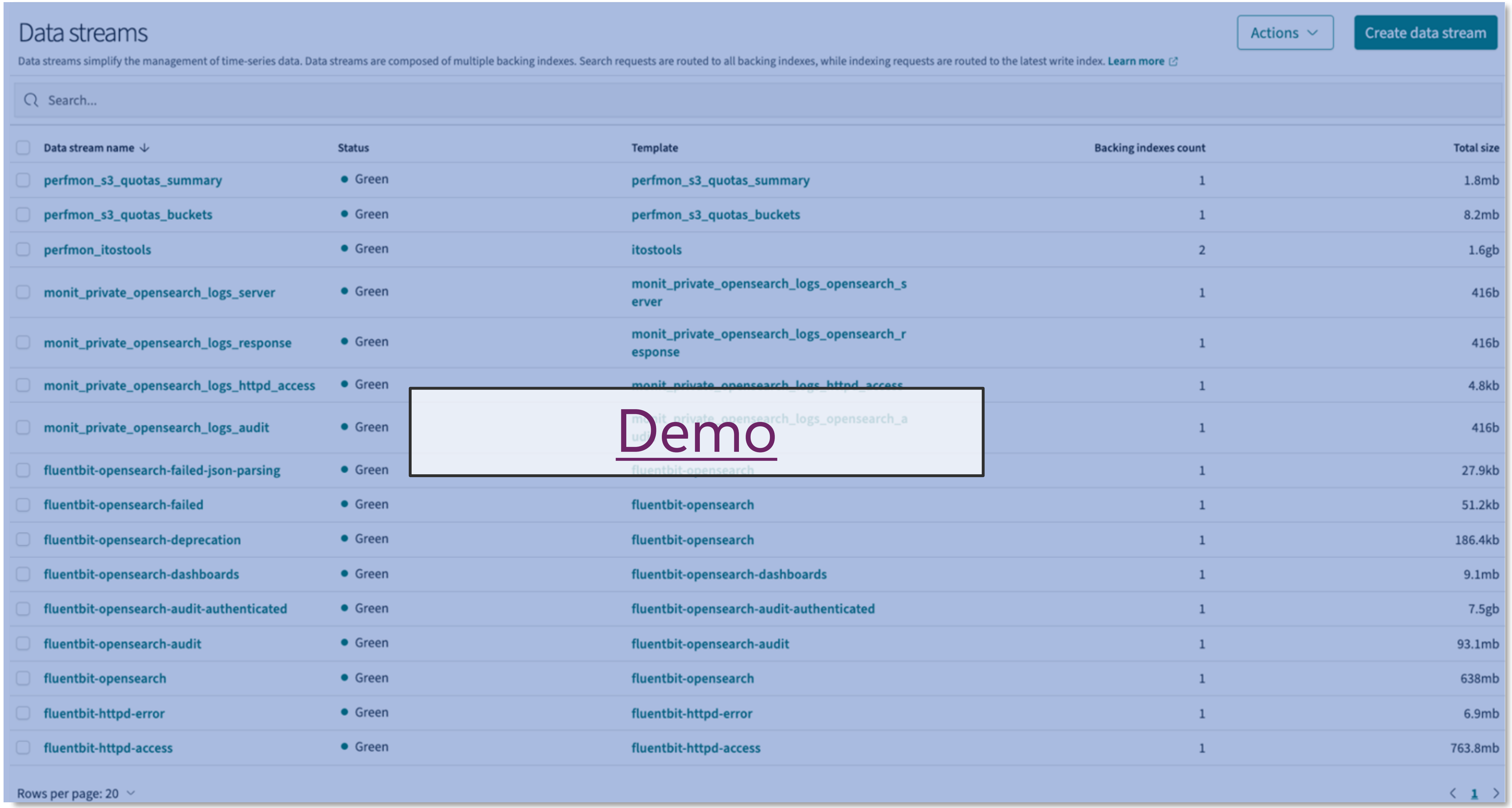
How to use Data Streams?

- OpenSearch treats indexed documents as immutable
  - Aligning well with append-only log use-cases
- Write once - read many
- Recommended to define a ISM retention policy
  - to manage lifecycle of the data hot->cold->delete
- Works with the popular ingestion tools such as:
  - Logstash, Fluent bit, vector.dev, Fluentd



```
# Fluent bit output plugin example
[OUTPUT]
    Name opensearch
    Match some_logs*
    Host os-playground.cern.ch
    Port 443
    Path /os
    Buffer_Size 128KB
    Logstash_Format Off
    Index fluentbit-logs
    Type _doc
    Time_Key @timestamp
    Time_Key_Format %Y-%m-%dT%H:%M:%S
    Time_Key_Nanos Off
    Tls On
    HTTP_User XXX
    HTTP_Passwd YYY
    Suppress_Type_Name On
    Workers 0
    Compress ""
    Write_Operation create
    Generate_ID Off
```

Demo



Data streams

Data streams simplify the management of time-series data. Data streams are composed of multiple backing indexes. Search requests are routed to all backing indexes, while indexing requests are routed to the latest write index. Learn more

□	Data stream name ↓	Status	Template	Backing indexes count	Total size
□	perfmon_s3_quotas_summary	● Green	perfmon_s3_quotas_summary	1	1.8mb
□	perfmon_s3_quotas_buckets	● Green	perfmon_s3_quotas_buckets	1	8.2mb
□	perfmon_itostools	● Green	itostools	2	1.6gb
□	monit_private_opensearch_logs_server	● Green	monit_private_opensearch_logs_opensearch_server	1	416b
□	monit_private_opensearch_logs_response	● Green	monit_private_opensearch_logs_opensearch_response	1	416b
□	monit_private_opensearch_logs_httpd_access	● Green	monit_private_opensearch_logs_httpd_access	1	4.8kb
□	monit_private_opensearch_logs_audit	● Green	monit_private_opensearch_logs_opensearch_audit	1	416b
□	fluentbit-opensearch-failed-json-parsing	● Green	fluentbit-opensearch	1	27.9kb
□	fluentbit-opensearch-failed	● Green	fluentbit-opensearch	1	51.2kb
□	fluentbit-opensearch-deprecation	● Green	fluentbit-opensearch	1	186.4kb
□	fluentbit-opensearch-dashboards	● Green	fluentbit-opensearch-dashboards	1	9.1mb
□	fluentbit-opensearch-audit-authenticated	● Green	fluentbit-opensearch-audit-authenticated	1	7.5gb
□	fluentbit-opensearch-audit	● Green	fluentbit-opensearch-audit	1	93.1mb
□	fluentbit-opensearch	● Green	fluentbit-opensearch	1	638mb
□	fluentbit-httpd-error	● Green	fluentbit-httpd-error	1	6.9mb
□	fluentbit-httpd-access	● Green	fluentbit-httpd-access	1	763.8mb

Demo

Migrate your current data from indexes to Data Streams

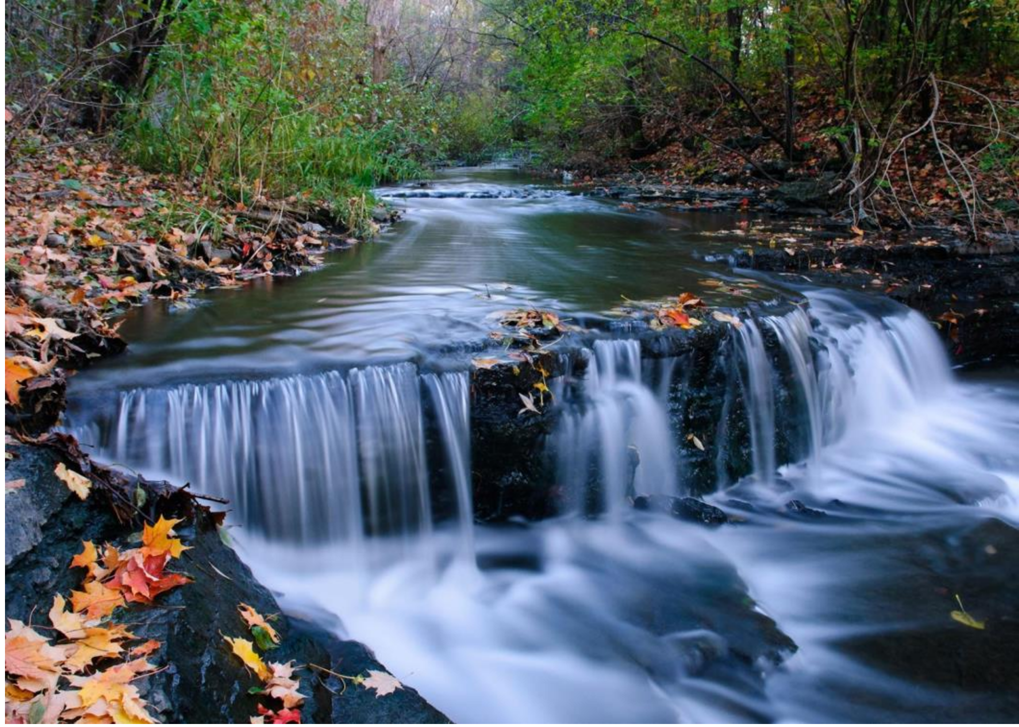
1. Create an index template of type Data Stream
2. Create a Data Stream:
   a. PUT _data_stream/logs-app
3. *Check if source index has field @timestamp and add it*
4. Reindex data

```
POST _reindex
{
     "source": {
          "index": "old-logs-index"
     },
     "dest": {
          "index": "logs-data-stream"
     }
}
```

1. Recommended: define ISM policy for data retention

Example of LanDB migration (TBs)

Summary



- Time-series data can be simplified with Data Streams
- Migration to Data Streams is easy
- Provides considerable improvement of operations
- Reach out to OpenSearch team for help
- Happy streaming! :-)

- Reach for more details:
  - https://opensearch.org/docs/latest/im-plugin/data-streams
  - https://opensearch.docs.cern.ch/data_ingestion/#data-streams