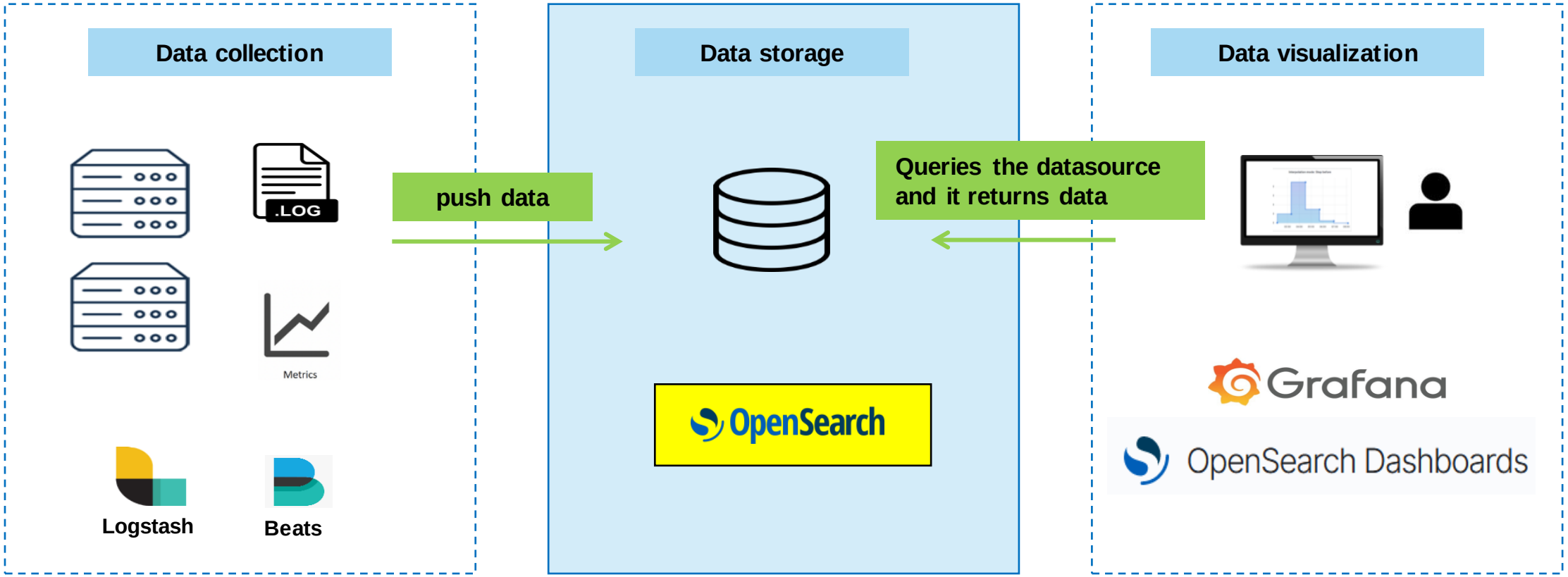
Karlsruhe Institute of Technology

OpenSearch

Introduction and overview

Components of monitoring architecture



Agenda

- What is OpenSearch?
- History of OpenSearch
- Use cases
- Features
- Basic concepts:
  - Cluster and nodes
  - Indices and documents
  - Shards
- Adding data to OpenSearch

https://opensearch.org/

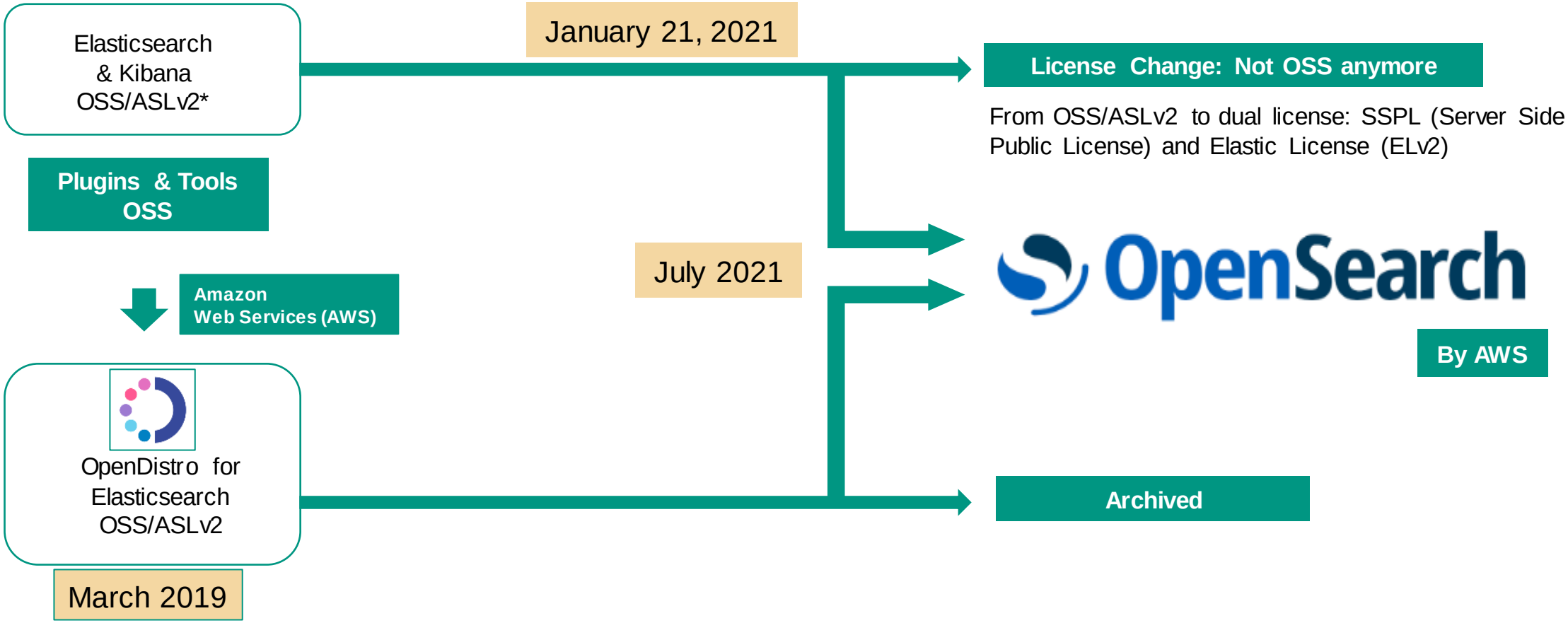
OpenSearch is a community-driven, Apache 2.0-licensed open source search and analytics suite that makes it easy to ingest, search, visualize, and analyze data.

What is OpenSearch?

- Launched in July 2021 by Amazon Web Services (AWS)
- It is a forked project based on old versions of Elasticsearch and Kibana (v. 7.10.2)
- Interact via REST API
- Built in Java using Lucene
- Consists of:
  - a search engine, **OpenSearch** (based on Elasticsearch);
  - a visualization user interface, **OpenSearch Dashboards** (based on Kibana);
  - as well as a series functionality-adding tools and plugins.

From Elasticsearch to OpenSearch

History of OpenSearch



*OSS = Open-source software
ASLv2 = Apache Software License version 2.0

Why was OpenSearch created?

- **Elasticsearch** made its debut in 2010 and established itself as the most popular search engine. Early iterations of Elasticsearch were open source and a combination of multiple products known as the **ELK Stack** (Elasticsearch, Logstash, Kibana).
- **AWS launched in 2019 Open Distro for Elasticsearch,** a value-added distribution of Elasticsearch 100% open source.
- **On January 21 2021**, Elastic NV announced that they would change their software licensing strategy and not release new versions of Elasticsearch and Kibana under the permissive Apache License, Version 2.0 (ASLv2).
- The move away from open source code by Elastic prompted **AWS** to roll out an open source, community-driven fork of Elasticsearch and Kibana called **OpenSearch**.

Uses cases

Logging
Centralize and search application logs at scale

Infrastructure monitoring
Analyze system metrics (CPU and memory usage)

Application monitoring
Monitor software services and applications in real-time

Security analytics
Network traffics, authentication logs

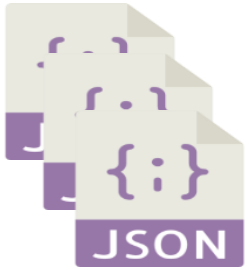
Business analytics
Examining business data for the purpose of gaining useful information

Full-text search
Application, websites

Root causes analysis
Quickly identify to resolve issues

Features

Document oriented and schema-free



Horizontal Scalability

RESTful API
JSON over HTTP

Used to index any kind of heterogeneous data

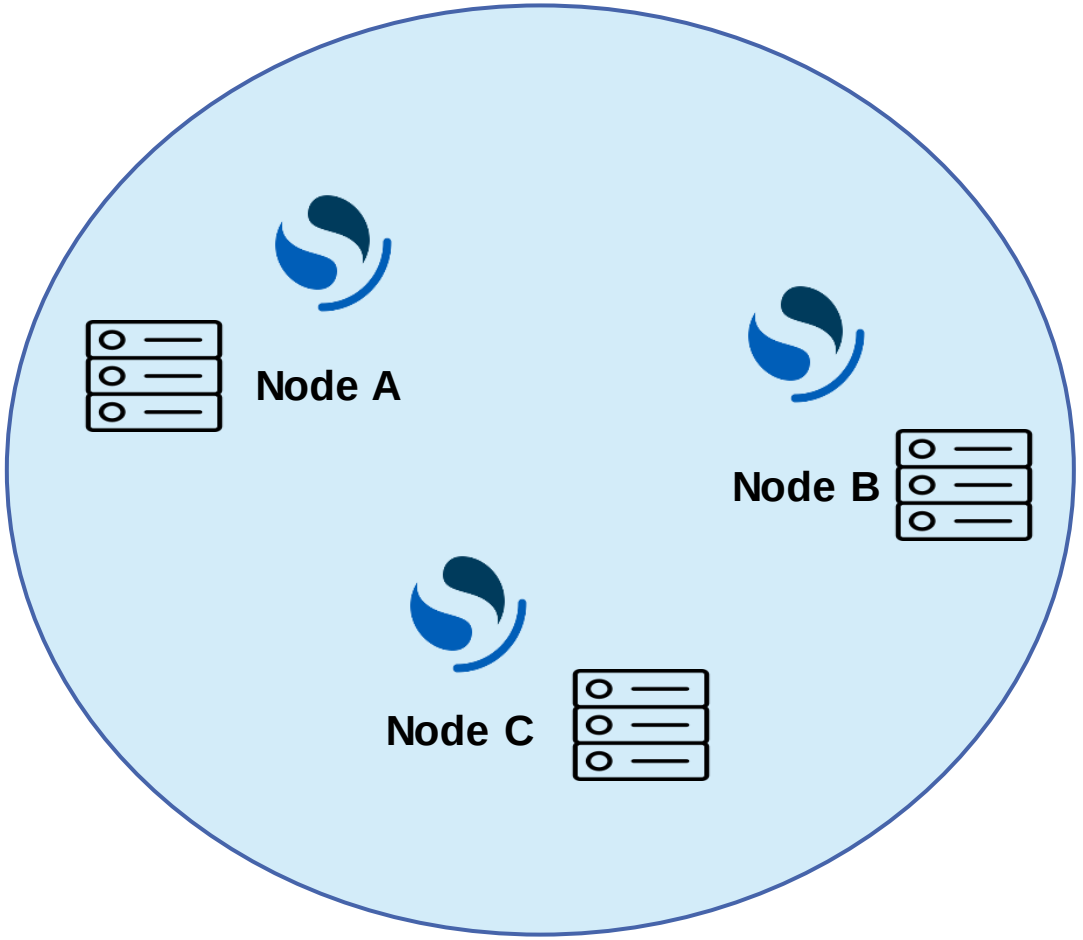
Security and Access Control

High Availability

Basic Concepts: Cluster and nodes

Cluster and nodes

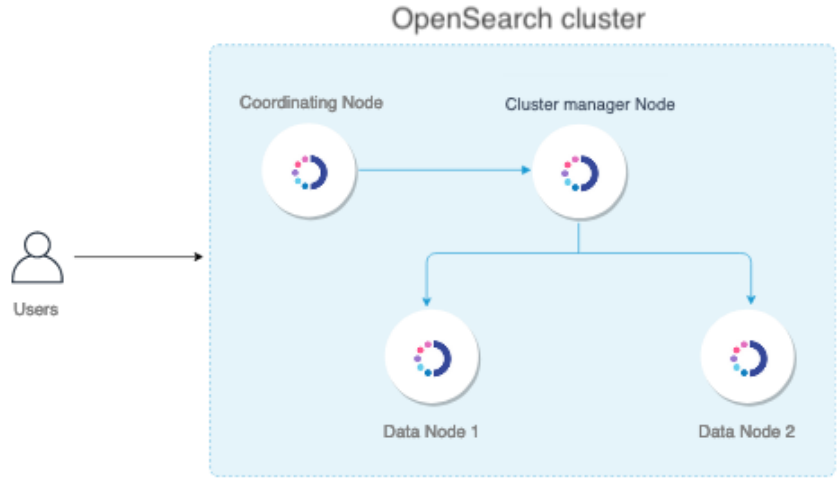
- The distributed design means that you interact with OpenSearch *clusters*.
- Each cluster is a collection of one or more *nodes*, servers that store your data and process search requests.



OpenSearch Cluster with 3 nodes

Cluster and nodes

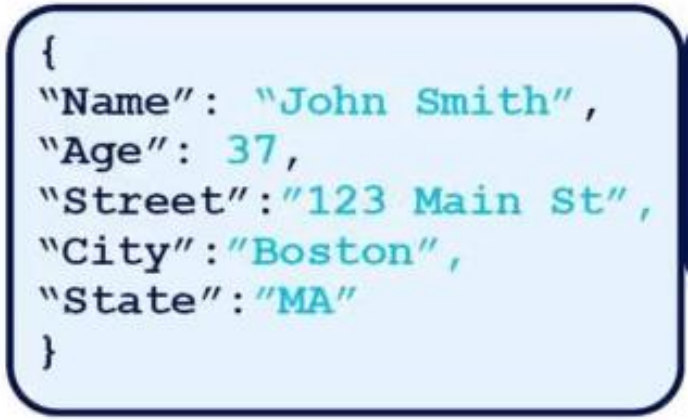
- The cluster is horizontally scalable, and adding additional nodes to the cluster allows the cluster capacity to increase linearly while maintaining similar performance.
- Nodes in the cluster can be differentiated based on the specific type of operations that they perform (cluster manager, data, ingest, coordinating node).



Basic Concepts: Data organizations

Indices and documents

- OpenSearch is a document oriented database.
- Documents are JSON structures that hold a collection of fields and their values.
- A document is like a row in a table in a relational database.



Indices and documents

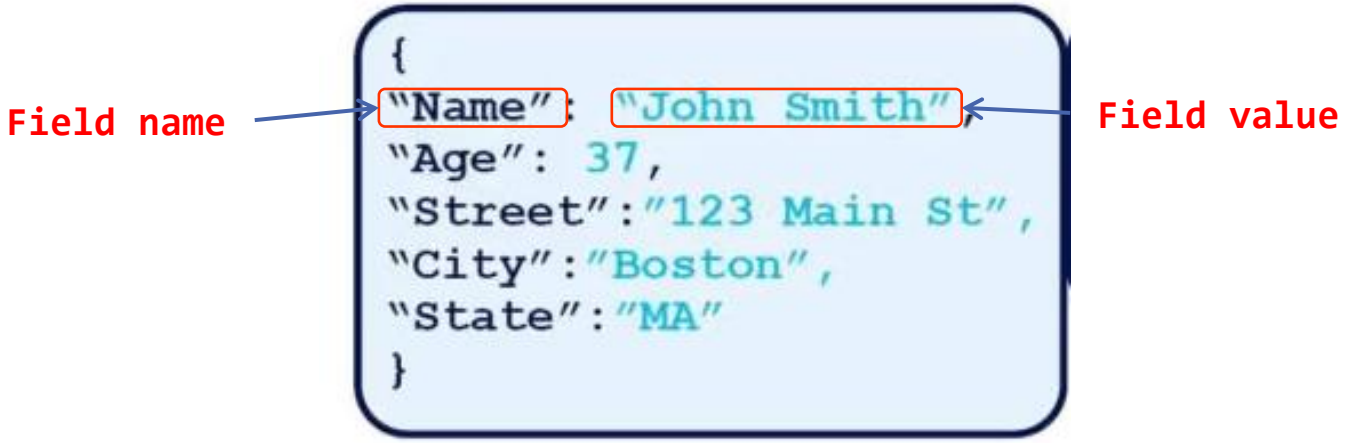
- OpenSearch organizes data into *indices*.
- An index is a logical *collection of documents*.
- An index is like to a database table.
- Although an index could have documents with entirely different content, for efficiency purposes, you would typically store similar documents in the same index.

Indices and documents

As an example, the following people index has documents about persons. Though the documents will all have similar content, some fields may only appear only in certain documents.

Fields

- A document contains a list of fields or *key-value pairs*.
- A field is similar to a column in a table in a relational database.
- Fields can be of several different types such as numbers, text, keywords, geo points, etc.



Mapping

- OpenSearch also has the ability to be *schema-less*, which means that documents can be indexed without explicitly specifying how to handle each of the different fields that might occur in a document.
- When **dynamic mapping** is enabled (by default), OpenSearch automatically detects and adds new fields to the index.
- However, if you know exactly what types your data falls under and want to enforce that standard, then you can use **explicit mappings**.

```
PUT sample-index1
{
  "mappings": {
    "properties": {
      "year":    { "type" : "text" },
      "age":     { "type" : "integer" },
      "director":{ "type" : "text" }
    }
  }
}
```

RDBMS vs Document-Oriented Database

RDBMS

Accounts				
ID	Firstname	Lastname	Email	Age
1	Virginia	Alaya	virginiaala…	25
2	John	Oneill	johnoneill..	23
3	Bruce	Smith	bsmith@...	33
4	Mike	Paul	mikepaul..	25

column=>field

Hobby	
ID	Hobby
1	Volleyball
2	Football
3	Hiking

row=>document

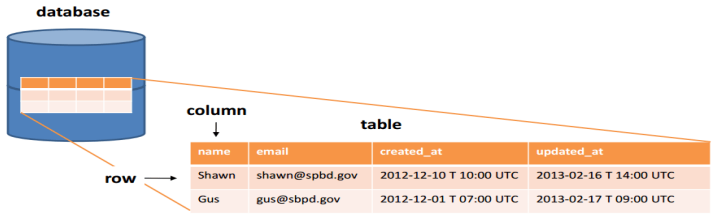
OpenSearch

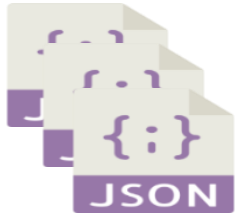
table=>index

- A *document* is similar to a *row* in the relational databases
- *Fields* are similar to *columns* in the relational databases
- An *index* is like to a database *table*

RDBMS vs Document-Oriented Database

Relational model	Document model
Stores data in a schema that uses tables with columns and rows	Stores entities as documents – like a JSON documents
Predefined schema	Dynamic schema
Based structured query language (SQL)	Querying through an API or unstructured query language
Vertically scalable	Horizontally scalable
It is slower	It is faster than relational model
Supports complex joins	Does not support complex joins

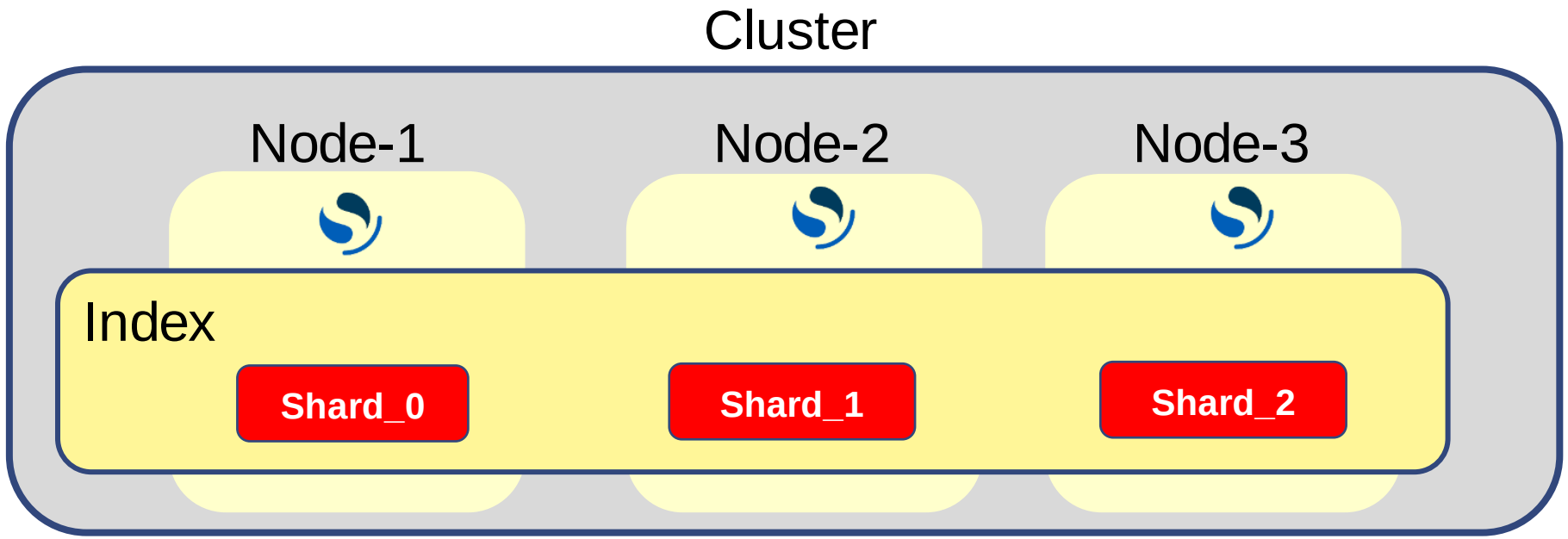




Basic Concepts: Internal Data Structures

Shards

- The shard is the atomic part of an index, which can be distributed over the cluster.
- Every index can be split into several shards to be able to distribute data.
- OpenSearch distributes shards almost all nodes in the cluster and can move shards automatically from one node to another in case of node failure, or the addition of new nodes.

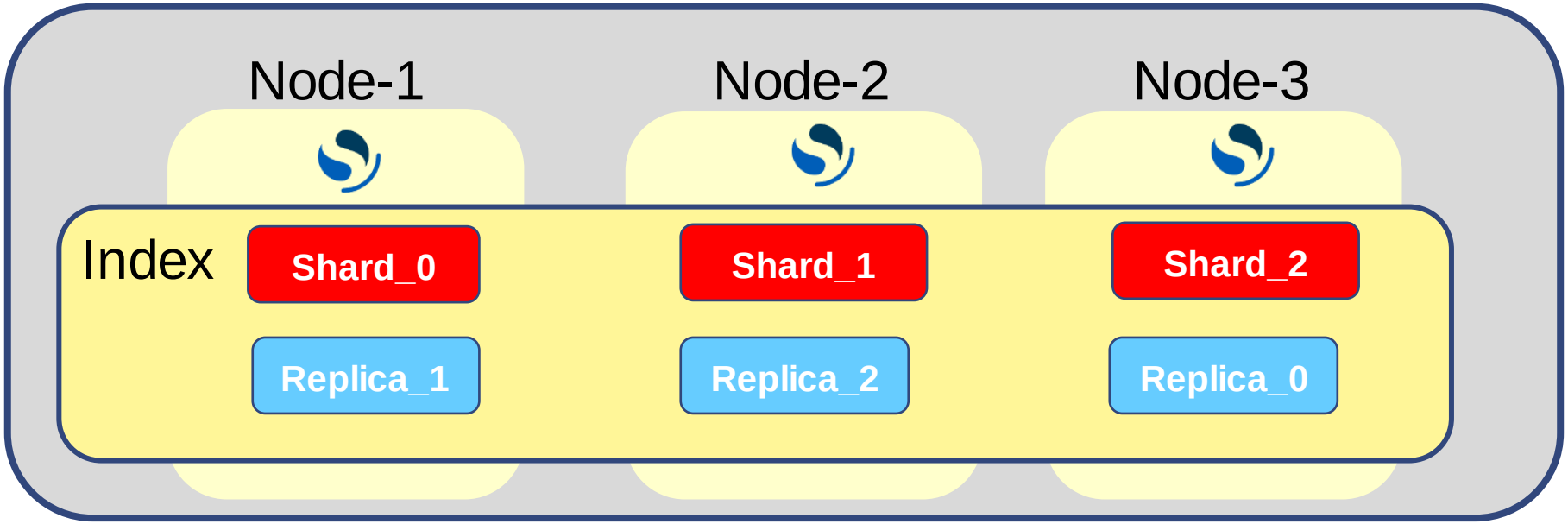


Index with 3 primary shards

Primary and Replica Shards

There are two types of shards: primaries and replicas.

- **Primary shard:** each document in an index belongs to one primary shard.
- **Replica shard:** a replica shard is a copy of a primary shard.
  - ☑ increase failover: a replica shard can be promoted to a primary shard.
  - ☑ increase performance: get and search requests can be handled by primary or replica shards.



Index with 3 primary shards and 1 replica each

Adding data to OpenSearch

There are a number of options for getting your data into OpenSearch, which is commonly referred to as **ingesting or indexing data.**

For example:

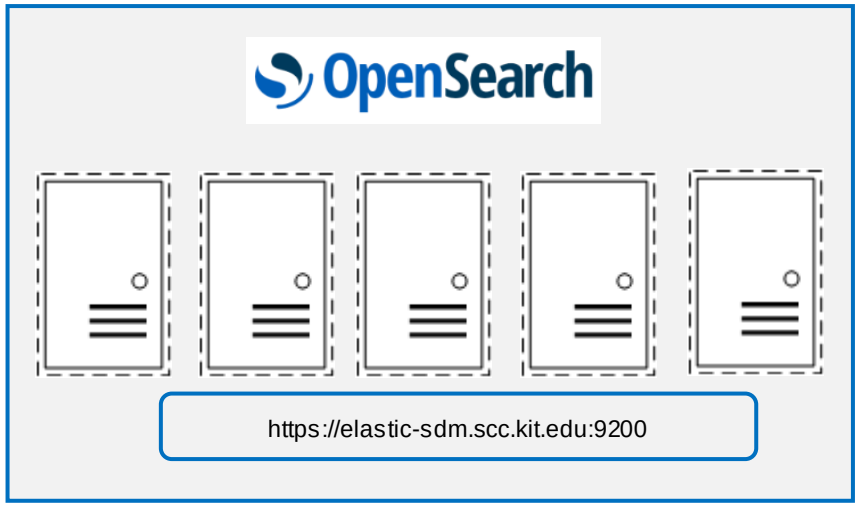
- Send data directly to OpenSearch from your application using one of the **language clients** (Python, Java, Ruby, PHP, etc...)
- Use **Agents and Ingestion Tools** for timestamped data:
  - **Beats** are data shippers designed to collect and ship a particular type of data from a server. Install a separate Beat for each type of data you want to collect
  - **Logstash** is an open source data collection engine which transforms and prepares data and supports a wide variety of data sources.

OpenSearch Cluster at SDM

GridKa
- Storage
- dCache
- Worker Nodes
- HTCondor

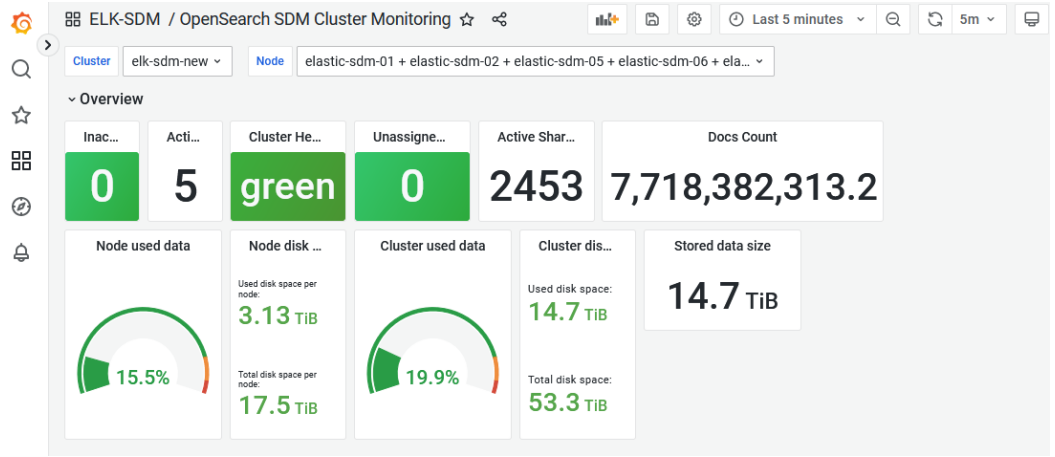
TAPE HPSS

LSDF WebDav



- 5-node cluster
- About 17,5 TB space per node
- Authentication and Authorization via KIT OpenID
- Deployment and configuration of OpenSearch via Puppet
- Security plugin enabled

Contact: elk-sdm-team@lists.kit.edu



Grafana OpenSearch SDM Cluster Dashboard

Useful links

- OpenSearch SDM Cluster Documentation:
https://docs-sdm.scc.kit.edu/DocumentationELKClusterSDM/

- OpenSearch project website and technical documentation:
https://opensearch.org/
https://opensearch.org/docs/latest/

- OpenSearch Community forum:
https://forum.opensearch.org/

- Elasticsearch project:
https://www.elastic.co/