Architecting the OpenSearch service at CERN

Sokratis Papadopoulos[1,*], *Pablo* Saiz[1,**], *Ulrich* Schwickerath[1,***], and *Emil* Kleszcz[1,****]

[1]CERN, Esplanades des Particules 1, 1211 Meyrin, Switzerland

Abstract. The centralised Elasticsearch service has been running at CERN since 2016, providing the search and analytics engine for numerous CERN users. The service has been based on the open-source version of Elasticsearch, surrounded by a set of external open-source plugins offering security, multi-tenancy, extra visualization types, and more. In October 2020, CERN embarked on an evaluation of OpenDistro for Elasticsearch, an alternative solution that used a different set of modules while retaining the core of open-source Elasticsearch. Notably, OpenDistro offered the advantage of bundling all components together, simplifying the deployment of new versions. This evaluation gained increased significance following a license change imposed by Elastic, the original creators of Elasticsearch. Consequently, the OpenDistro project was rebranded, now based on a forked version of Elasticsearch, called OpenSearch. Motivated by the license change and the streamlined deployment of the feature-rich OpenSearch project as a fully open-source environment, the decision was taken to migrate the service at CERN towards it. The migration required a complete architectural redesign to accommodate the new modules while upholding the established standards of resource efficiency. The new service not only introduced a wide range of additional capabilities but also resolved long-standing maintainability issues while meeting the growing demands of various use-cases. At the time of writing, CERN's service comprises 42 OpenSearch and 41 OpenDistro clusters in active production, plus 28 OpenSearch development clusters. This article covers the motivation, design, and implementation of this transition, highlighting the challenges encountered throughout the process.

1 Introduction

Elasticsearch has been widely used at CERN, currently counting over a hundred clusters, indexing 500 TBs of data and 1.2 trillion documents. It serves as a monitoring and data inspection tool for CERN IT and experiments like ALICE, ATLAS, CMS, LHCb, and NA62, while also powering the search engines of INSPIRE [1] and Zenodo [2]. Within CERN IT, Elasticsearch handles log monitoring and analysis for various IT services, including security, monitoring, and storage. Section 2 explains the reasons behind the migration from Elasticsearch to OpenSearch, Section 3 analyzes the new service architecture, Section 4 details the migration process, Section 5 outlines the roadmap and Section 6 provides a summary. To facilitate a better understanding of this article, relevant tools' definitions are provided now.

*e-mail: sokratis.papadopoulos@cern.ch
**e-mail: pablo.saiz@cern.ch
***e-mail: ulrich.schwickerath@cern.ch
****e-mail: emil.kleszcz@cern.ch

- **Elasticsearch**: A distributed search and analytics engine powered by Apache Lucene. It is widely used for log analytics and full-text search.
- **Kibana**: The web user interface used for data visualization and dashboard creation, seamlessly integrating with Elasticsearch.
- **OpenDistro for Elasticsearch**: An AWS-stewarded project, based on Apache 2.0 releases of Elasticsearch. It integrated plugins to the Elasticsearch core for extra features. However, it is now archived since Elastic has discontinued the Apache 2.0 releases.
- **OpenSearch**: A fork of Apache 2.0 licensed Elasticsearch core codebase. OpenSearch is an independent, open-source project continuing the development of Elasticsearch under the Apache 2.0 license. It has gained support from over 65 organizations [3], including CERN.
- **OpenSearch Dashboards**: A fork of Kibana, serves as the native data visualization tool.

2 Motivation for Migration

This section outlines the top five reasons driving the service at CERN from Elasticsearch to OpenSearch:

1. **Maintainability**: Requiring multiple external plugins to run the service made upgrading and integration efforts challenging, causing delays and impacting stability.

2. **Deployment**: OpenSearch provides native plugins for essential functionalities, streamlining the deployment process with pre-integrated RPM packages and eliminating the need for patching and separate plugin installations.

3. **Features**: The OpenSearch ecosystem offers a comprehensive set of built-in plugins, including security with encryption in both REST and Transport layer, alerting, index management, and anomaly detection, all licensed under Apache 2.0. Leveraging these plugins enhances the central service at CERN and supports customer requirements.

4. **Licensing**: Elastic's shift in licensing [4], discontinuing open-source development, and adopting restrictive licenses prompted the need to adopt OpenSearch, which continues development under the original Apache 2.0 license [5].

5. **Customers isolation**: In the legacy architecture [6], resource-intensive queries from one customer could impact the performance of others due to the shared-cluster model. The dedicated cluster approach followed on OpenSearch deployment ensures dedicated resources for each use-case, preventing such performance issues.

3 Service Design

CERN's legacy Elasticsearch service [6] [7] utilized Apache 2.0 releases from Elastic. It incorporated external plugins like ReadOnlyRest [8] for security and deployed Virtual Machines on-premise to create large clusters serving multiple use-cases with various endpoints. The new OpenSearch service differs significantly from the legacy deployment. Still entirely based on open-source solutions, with OpenSearch at core, a new security model and new hardware in-place, the OpenSearch service architecture is presented in detail throughout this section. The new service follows a dedicated clusters approach, minimizes external dependencies and is designed to provide flexibility and adaptability to individual user requirements.

3.1 Glossary of Terms

To understand the OpenSearch service architecture, the following terms are important:

- **Bundle**: A set of physical machines that run multiple OpenSearch clusters.
- **Cluster**: An independent OpenSearch cluster, dedicated to one customer/team.
- **Node**: An instance of the OpenSearch process.
- **Alias**: An alternative endpoint (URL) for the cluster.

3.2 Hardware Specification

The OpenSearch service at CERN is deployed and running on a pool of 156 physical machines distributed across three availability zones. Each machine is equipped with 64 CPU cores, 256 GB of RAM, and 10.5 TB of usable SSD space. The machines are managed by OpenStack Ironic [9] and unlike the legacy service, no virtualization layer is used. The machines' configuration is centrally managed with Puppet, using GitLab at CERN to store the configuration data. Secrets are managed with Teigi [8].

3.3 Deployment Strategy

The smallest deployment unit is a bundle consisting of three machines spread across the availability zones. Each bundle can host multiple clusters, accommodating different use-cases. Larger clusters requiring more resources are placed in dedicated bundles. The flexibility of the setup allows for the adjustment of resources based on individual requirements. Logical Volumes (LVs) efficiently manage storage across the available physical hard disks, allowing for easy expansion when the cluster is nearing capacity. For larger storage needs, external storage is used, connected to CephFS clusters. Figure 1 showcases the way OpenSearch clusters are configured within the hosts of a bundle. Specifically, it shows two example bundles with two defined OpenSearch clusters each. The number of OpenSearch nodes, amount of RAM, and disk space allocated to each cluster depend on the specific use-case. Larger clusters have more data nodes with larger disk space allocations, and the RAM allocation respects the maximum limit of 32 GB per process to optimize memory usage.

3.4 Authentication and Authorization

Authentication and authorization in the OpenSearch service are primarily handled by OpenSearch itself, while an Apache server in front of it acts as a reverse proxy for multiple clusters residing on the same host but on different ports. Authentication options include basic-auth, Kerberos, and OpenID. Authorization is performed within OpenSearch, with integration of CERN LDAP allowing access management based on CERN e-groups. Cluster owners are responsible for managing basic-auth user credentials, access rights, and document-level security. OpenID configuration is handled through the CERN application portal [10].

3.5 Monitoring

For machine-level monitoring, collectd daemons are used to continuously gather crucial machine metrics, which are then sent to our central monitoring infrastructure for analysis. For cluster-level monitoring, a dedicated OpenSearch cluster called *perfmon* is employed. Within this cluster, service-specific log data are collected with Filebeat, and then processed and organized by Logstash for easy access. Additionally, vital data like cluster state and thread pool information are captured, also processed through Logstash and stored in the *perfmon* cluster.

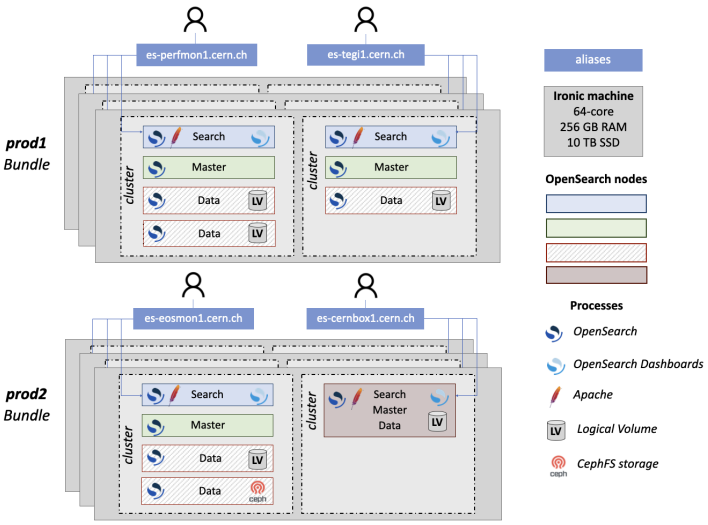

Figure 1. The OpenSearch service architecture at CERN

3.6 Operations

For service management tasks, a pool of Linux machines is used. These machines handle various operations, including KPI/SLI reporting, accounting and billing, cluster bootstrapping and maintenance, cluster configuration backup, and updating cluster index templates. These tasks are performed through authenticated cron jobs on the designated machines.

4 Migration Implementation

This section discusses the migration process from Elasticsearch to OpenDistro and OpenSearch, including the key takeaways and the timeline of the projects.

4.1 From Elasticsearch to OpenSearch

Due to the different security models followed by Elasticsearch and OpenSearch, an online migration or mixing of Elasticsearch and OpenSearch nodes was not feasible. As illustrated in Figure 2, a new OpenSearch cluster was bootstrapped for each use case under migration. All data, including index templates, retention policies, indices, and Kibana objects, were copied across. Live data were simultaneously being ingested on the old Elasticsearch and new OpenSearch endpoint. At this moment, old and new cluster held the same data and operated in parallel for a while. After configuring the new cluster and ensuring everything worked as expected, the old endpoint was decommissioned and its alias redirected to the new OpenSearch cluster.

The migration process had zero downtime and allowed ample time for testing and verification. In some cases, for use-cases with transient data, such as retaining data for only 30 days, the migration of index data was not necessary. Instead, the process allowed for 30 days to pass during simultaneous ingestion of live data, until both the old Elasticsearch and new OpenSearch endpoint showed the same data.

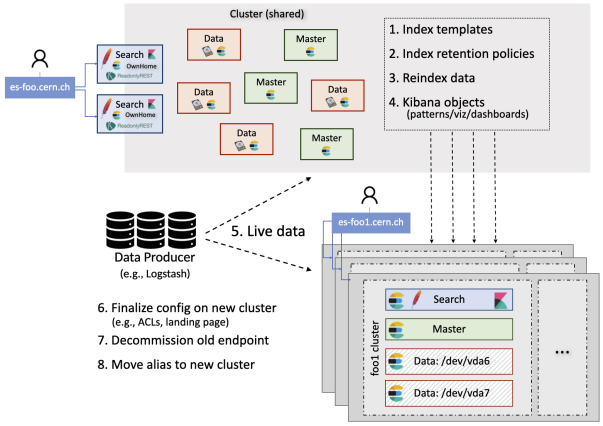

Figure 2. The migration process from Elasticsearch to OpenSearch

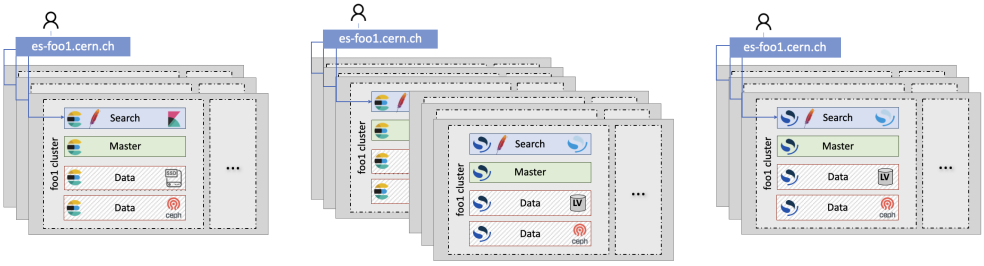

Figure 3. The migration process from OpenDistro to OpenSearch

4.2 From OpenDistro to OpenSearch

Since OpenDistro and OpenSearch use the same security model and can communicate with each other, an online update strategy can be followed for this migration. As illustrated in Figure 3, the procedure follows below steps:

1. New machines and the corresponding cluster OpenSearch nodes are bootstrapped. Firewall settings are adapted so that all machines can communicate among them.
2. The clusters initiate automatic data rebalancing, shifting shards to new nodes. The old nodes are gradually phased out until all data is exclusively on the new nodes.
3. The clusters aliases are redirected to the new machines, ensuring that new client connections are served by the new OpenSearch nodes.
4. Finally, the old OpenDistro processes can be safely stopped.

4.3 Key takeaways

The migration from Elasticsearch to OpenSearch and the ongoing migration from OpenDistro to OpenSearch provided valuable insights. Some key takeaways include:

- OpenSearch seamlessly integrates with a variety of tools used at CERN such as OpenID, LDAP, and Kerberos, requiring no custom solutions or additional integration work.
- Due to the lack of support for multiple instances in the upstream OpenSearch Puppet module [11] the team at CERN had to develop and maintain custom Puppet modules for OpenSearch and OpenSearch Dashboards.
- Latest releases of Elasticsearch client libraries, the Logstash Elasticsearch output plugin and the Elastic Beats products only establish connections to Elastic licensed clusters. Therefore, the OpenSearch community has taken the initiative to fork the client libraries [12] and develop a Logstash OpenSearch output plugin [13] for seamless communication with OpenSearch. For Beats, the only viable option is for data to be routed through Logstash before reaching OpenSearch clusters. Consequently, during the migration to OpenSearch, all customers are required to use OpenSearch client libraries and replace the Logstash Elasticsearch output plugin with the corresponding OpenSearch one.
- Situations arise when the original requester of the legacy Elasticsearch service has left the organization, requiring knowledge transfer to new personnel before proceeding with the migration. Furthermore, users may need to prioritize other critical activities, leading to the maintenance of five major versions simultaneously by the OpenSearch team at CERN for an extended duration and thus increasing the support load.
- The implementation of dedicated clusters requires users to abide by their assigned quotas. In the previous shared cluster model, if a user surpassed their allocated resources, it could be balanced by redistributing resources from other users who over-provisioned. However, this approach had a negative impact on the cluster's responsiveness to other users, compromising their experience. With dedicated clusters, each use-case has its own cluster and must adhere to the specified quotas. At the initial stages of the OpenSearch clusters, some adjustments may be required to align resources with the actual demand of each use-case. The goal is to achieve balance over time, thus ensuring optimal resource utilization.
- The migration to OpenSearch version 2 introduced a major breaking change by removing support for the *_type* field [14]. Consequently, in order to ensure compatibility, customers had to review their data producers (e.g., Logstash, Beats, Flume, etc.) to eliminate the use of the *_type* field whenever it was used.

4.4 Migration timeline

Figure 4 illustrates the number of clusters maintained by the CERN team in recent years, showcasing the migration journey from Elasticsearch to OpenDistro and OpenSearch. It is important to note a difference in the deployment model between Elasticsearch and OpenDistro/OpenSearch clusters. The legacy service followed a shared-cluster approach, where multiple use-cases utilized the same large cluster. In contrast, the new service employs dedicated clusters, varying in size, for each specific use case. This explains the significant difference in the number of deployed clusters between the past and present.

As depicted in the graph, the first OpenDistro clusters were bootstrapped in the summer of 2021, initiating the first cluster migrations from Elasticsearch to OpenDistro. At the start of 2022, when the first stable release of OpenSearch became available, the team swiftly transitioned its focus to OpenSearch and deployed already some test clusters. Consequently, the migration to OpenDistro has stopped and remaining clusters were instead migrated directly to OpenSearch. With the release of OpenSearch version 2 in August 2022, the already migrated clusters running version 1 were upgraded to the latest version. Simultaneously, the general

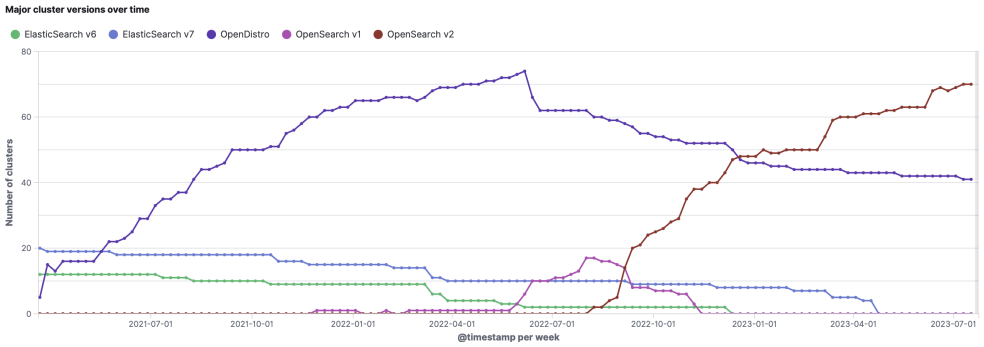

Figure 4. The major cluster versions over time for the service

migration away from Elasticsearch towards OpenSearch version 2 continued. Meanwhile, naturally all new use-cases were directly bootstrapped in OpenSearch version 2.

The primary objective was to decommission all Elasticsearch 6.8.0 clusters in the service, which was successfully accomplished by the end of 2022. Subsequently, the decommissioning of Elasticsearch version 7.1.1 was completed by the end of March 2023. The decommissioning target for the remaining OpenDistro clusters is set for Q2 2024.

5 Roadmap

Despite the limited dedicated personnel, there are numerous opportunities to further explore and enhance the service at CERN. The top six priorities are presented:

- **OpenDistro and AlmaLinux 9 migration**: The remaining 41 OpenDistro clusters should get migrated to OpenSearch, while the host machines will move from CentOS Stream 8 to AlmaLinux 9, in line with the recommendations of the CERN Linux Committee.
- **Further automation**: The *itostools* is a customised repository developed at CERN that includes scripts to assist with service management and operations. The plan is to further develop and expand this repository to automate various operational tasks.
- **OpenSearch community engagement**: Since May 2023, CERN has been recognized as an official OpenSearch partner. The goal is to enhance CERN's visibility within the OpenSearch community and actively participate in meetings and forums to contribute to the development and growth of the community.
- **Data streams**: Leveraging OpenSearch's data streams functionality simplifies cluster management for projects involving append-only log data.
- **Exploration of OpenSearch capabilities**: OpenSearch comes with an Observability suite, and numerous plugins like Index Management (including Snapshots), Anomaly Detection, and more, showing great potential in improving service quality.
- **Alternatives to Logstash & Beats**: Elastic's decision to distance themselves from OpenSearch caused compatibility issues with their products (Logstash, Beats). CERN relies on these tools to feed data to OpenSearch clusters. Thus, exploring alternatives is crucial for CERN's open-source standing and service continuity.

6 Conclusion

This paper provided an in-depth exploration of the successful migration process from Elasticsearch to OpenSearch at CERN. Motivated by maintainability, licensing changes, and improved features, the migration aimed to overcome legacy challenges and leverage OpenSearch's capabilities. The new service architecture adopts an uncommon deployment model with multiple nodes and clusters hosted on powerful machines, ensuring resource efficiency. The migration process, covering Elasticsearch to OpenSearch and OpenDistro to OpenSearch transitions, is detailed, including challenges and insights. The paper concludes with a roadmap for further automation, community engagement, and exploration of OpenSearch capabilities. Overall, it provides a comprehensive and informative account of the migration journey, making it a valuable resource for organizations considering or undergoing similar transitions.

References

[1] *INSPIRE: the leading information platform for High Energy Physics (HEP) literature*, https://inspirehep.net, [Online; accessed 21-July-2023]

[2] *Zenodo: a multi-disciplinary open repository maintained by CERN*, https://zenodo.org, [Online; accessed 21-July-2023]

[3] *Official OpenSearch partners*, https://opensearch.org/partners/, [Online; accessed 21-July-2023]

[4] *Elastic: Doubling down on open, Part II*, https://www.elastic.co/blog/licensing-change, [Online; accessed 21-July-2023]

[5] *The Apache v2.0 license*, https://www.apache.org/licenses/LICENSE-2.0, [Online; accessed 21-July-2023]

[6] P. Saiz, U. Schwickerath, *Large Elasticsearch cluster management*, in *EPJ Web of Conferences* (EDP Sciences, 2020), Vol. 245, p. 07021

[7] J.R. Andersson, J.A. Moya, U. Schwickerath, Frontiers in big Data **4**, 718879 (2021)

[8] U. Schwickerath, P. Saiz, Z. Toteva, *Securing and sharing Elasticsearch resources with Read-onlyREST*, in *EPJ Web of Conferences* (EDP Sciences, 2019), Vol. 214, p. 08032

[9] *Ironic: Bare Metal as a Service*, https://ironicbaremetal.org, [Online; accessed 21-July-2023]

[10] C.S.S.O. Team et al., Tech. rep. (2013)

[11] *Official OpenSearch puppet module*, https://github.com/voxpupuli/puppet-opensearch, [Online; accessed 21-July-2023]

[12] *OpenSearch language clients*, https://opensearch.org/docs/latest/clients/index/, [Online; accessed 21-July-2023]

[13] *Logstash output OpenSearch plugin*, https://github.com/opensearch-project/logstash-output-opensearch, [Online; accessed 21-July-2023]

[14] *OpenSearch version 2: breaking changes*, https://opensearch.org/docs/latest/breaking-changes, [Online; accessed 21-July-2023]