

Designing a Data Warehouse for Collected Data About User Activity in Social Networks Using Elasticsearch

Iryna Mysiuk ¹

¹ Ivan Franko National University of Lviv

1 Universytetska Street, Lviv, 79000, Ukraine

DOI: [10.22178/pos.94-13](https://doi.org/10.22178/pos.94-13)

JEL Classification: D83, L86


Received 30.06.2023

Accepted 28.07.2023

Published online 31.07.2023

Corresponding Author:

iruna.musyk8@gmail.com

© 2023 The Author. This article is licensed under a [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) 

Abstract. In this paper, a data storage data warehouse is designed to store collected data from social networks. Creating indexes with data and selecting a configuration with the appropriate number of shards and replicas is described – the primary states of the cluster and possibilities of its scaling. The features of working with the non-relational Elasticsearch database are described when working with data on user activity in social network posts. Among social networks, Facebook and Instagram were chosen for analysis. The paper describes the advantages and disadvantages of using such a data store compared to Apache Kafka. Analysed existing data insertion Application Program Interfaces (APIs) and data visualisation tools integrated with Elasticsearch. The study describes the use of the Bulk API to insert many records at once into a database. The designed data warehouse uses Kibana, a data visualisation and analytics tool integrated with the selected database. Also, it is shown the ability to insert and view logs using Elasticsearch, Logstash, and Kibana (ELK stack). Tested data ingest by logging into the database using Beats. The obtained results can help implement a system for analysing user activities from social network data based on Elasticsearch as a central component.

Keywords: social networks; data warehouse; data analytics; big data processing; system design.

INTRODUCTION

In recent years, there has been an increase in the number of social networks. This may be due to their popularity due to the rapid exchange of information and global coverage of various blogging trends. As a result, specific dependencies can be established in different periods of user activity in social networks. The problem is the large amount of data and the difficulty of obtaining it from web pages. Access to the data is quite limited, but some activity data can be collected. This can be the number of likes, comments, shares, etc. To obtain data from social networks, a program developed based on the Selenium library is used [1, 2].

Collected data can be saved to a file or in databases. Usually, the semi-structured data was stored in a file with the format Comma Separated Values (CSV). Searching for data in a file is more complicated than in databases with filters and possible division into more atomic values, taking

into account recommendations for a better search.

In recent works [3, 4], Kafka has been actively used to monitor data and store the history of additions.

Non-relational databases provide more flexible options for searching data by several parameters. The Elasticsearch, Kibana, and Logstash (EKL) technology stack allows you to monitor the process of data filling and analysis of added information in the storage.

This paper aims to design a data warehouse based on the EKL stack to store collected data from social networks about user activity.

The main tasks in the work are the following:

- describe a process of data transformation and ingestion data;
- design data storage and description of the data ingestion process in Elasticsearch.

METHODS

A data warehouse is a data management system for data analysis based on stored data. The main components are data sources, a data warehouse with raw and metadata, data marts and a view of results with accounting, reporting and mining. Data sources can be flat files (the database is saved to a file) and operating systems with data collected in the staging area. The data warehouse's central element is a database containing meta, raw and summary data. Data warehouses can be divided into separate data marts for purchasing, sales and inventory. The final element of such a system is the reporting layer for business analytics.

A data warehouse can be implemented with the EKL stack. Elasticsearch is the central element. Elasticsearch is a non-relational database that uses a search engine in JavaScript Object Notation (JSON) format of a web service developed in the REST (Representational State Transfer) architectural style, based on the Lucene library. All data is divided between data indexes, each of which houses a specific set of data (filter data, main content, etc.). In addition, there is a possibility of data recovery [5]. Possible configurations include setting the number of shards and replicas for each index. The status of the index indicates incorrect configuration settings or problems in operation. Among the possible positions of the index are green – everything is fine, yellow – disproportionate configuration of the index with shards and replicas, and red – problems with work. Elasticsearch contains many developed Application Program Interfaces (APIs) for data insertion, information, search, and transformation. This set of REST APIs makes it possible to use Elasticsearch features. Logstash is a process of collecting logs from data sources. Kibana is a visualisation tool for reporting analysis. REST API is built using Hypertext Transfer Protocol (HTTP) methods: GET, POST, PUT, DELETE and others.

Bash scripts were used in the work to transform data and use curls for bulk operations in Elasticsearch. Scripting is a set of commands on the command line. This monotonous process allows you to automate data conversion into JSON format and send it to the database.

In general, after conducting the Apache Kafka and EKL stack benchmarking for multiple data collection from the social network, the following conclusions can be drawn:

– Apache Kafka is often compared to Elasticsearch as a technology that can be used as an alternative. The specifics and purpose of these tools are quite different. Apache Kafka can be considered the best for monitoring and logging but as a search engine and setting up customised search queries – Elasticsearch [6]. Each selected tool can be more helpful or practical, depending on the task. In working with data on user activity in social networks, a more important mission for further analysis is data search and analytics.

– Kafka is primarily not used as a database but as a change monitoring system. One of the disadvantages of Elasticsearch is the difficulty of grouping data with connections [7].

RESULTS AND DISCUSSION

Data transformation and ingestion process. The machine learning process requires testing data, which will later be used to classify the data. Therefore, storing such a data set for automatic search and replenishment is more convenient using the database, as shown in Fig. 1, collected data in a file for transformation and storage in a database.

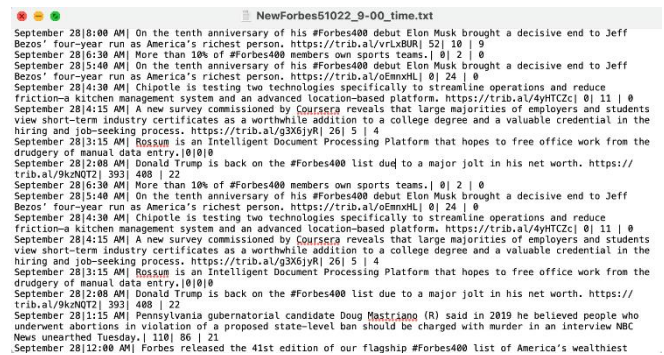


Figure 1 – An input file with data to send to the database

The raw data is converted into the desired format. To send data in a batch, we can group it into a JSON file of limited size. After that, the data from the converted file is transferred to the bulk endpoint, shown in Figure 2.

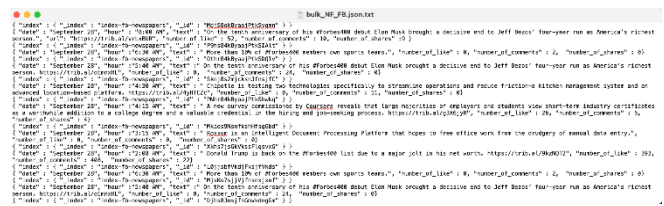


Figure 2 – Transformed data to use it in Bulk API

This data processing and transformation stage is necessary to prepare the data for the ingest process. When data is loaded into the database, the data load statuses are returned in the response. Thus, a report is obtained on the success of downloading data or errors during some downloads. In case of an unsuccessful download, we can perform a re-download with corrections.

Data storage design and the process of ingesting data to the Elasticsearch data store. Designing such a data warehouse for a specific task will allow to optimise the work of analysis and bringing data to a single data presentation format. Also, it will be possible to identify data with the wrong quality and other defects during processing at the data loading stage. The flow is shown in Figure 3, from downloading data from files from social networks to the corresponding analysis results.

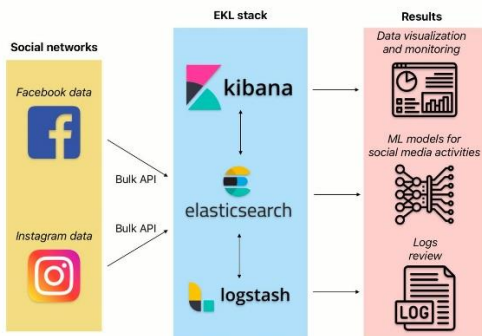


Figure 3 – The general view of the data warehouse

After downloading the Elasticsearch installation package, the installation process takes place with the generation of a password and a unique token for integration with Kibana [8, 9]. After starting Elasticsearch, the logs will display events and all changes made with the REST API, as shown in Figure 4. The installed version of Elasticsearch is 8.9.0.



Figure 4 – Elasticsearch event logging view

The installation process of Kibana is similar to Elasticsearch. The techniques are all logged in the console, and changes and errors can be reported, as shown in Figure 5. Kibana opens in the browser because it visualises all events that could be performed using the REST API in Elasticsearch.

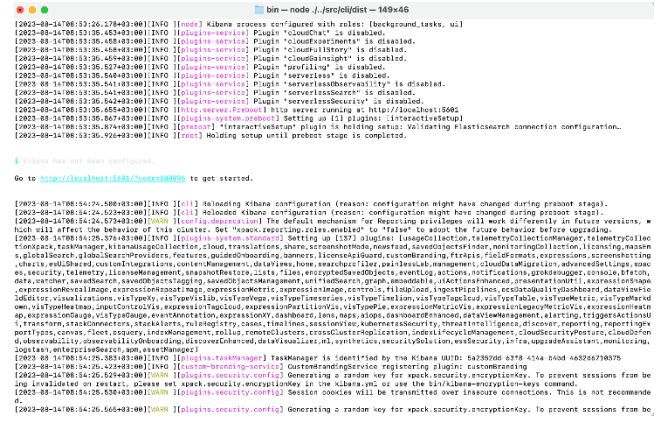


Figure 5 – Kibana event logging view

The index creation process is automated using a script but can be performed manually using the PUT method with the specified number of shards and the selection of these parameters affects the state of the index. Data index settings can be changed during system operation. As shown in Figure 6, the selected configuration may differ, and the Postman tool was used, which is designed to work with the API. The tool helps configure requests through the user interface, not the command line.

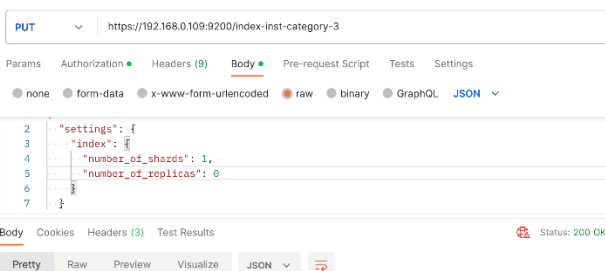


Figure 6 – An example of a request to configure the data index index-inst-category-3 in Elasticsearch

Based on the collected data from social networks [1, 2], it is possible to fill the database. Data on the news from well-known publications that published in Facebook posts Forbes,

New York Times, Reuters, and Washington Post are added to the relevant indexes index-fb-newspapers, index-fb-newspapers-1, index-fb-newspapers-2, index-fb-newspapers-3. In the same way, data were collected on the keywords astronomy, medicine and information technologies from Instagram social network y индекс index-inst-category-1, index-inst-category-2, index-inst-category-3. As shown in Figure 7, the indexes are filled with data with the specified main characteristics after data ingestion. This process took place using text data transformed into JSON format and sent via the Bulk API. Statuses are different for indexes. This has been changed to demonstrate the existence of other states. This signals the number of shards and replicas is more than needed and should be changed. These data changes can be tracked using a web page from Kibana and requesting the REST API with a display of all existing data indexes.

Distribution by indexes can be done differently, provided that data is not added to only one index. There are recommendations for the optimal loading of data indexes and their settings.

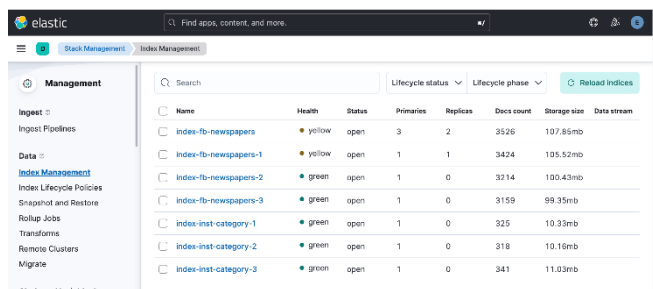


Figure 7 – Viewing existing indices and their parameters through the Kibana tool

An example of the record structure in the created Elasticsearch database index is shown in Figure 8. Numerical values are the number of comments, likes and links under the post with the text value of the news in the centre. There is also a specified date and time when information was collected to search for time dependencies.

As we can see in Figure 9, in Kibana, a customised dashboard can be created to monitor changes in data and the system. In addition, the user can create charts that will be updated after a specified time and signal critical modifications based on the set filters. Changes in logs and metrics can be displayed on such charts. In addition to primitive graphs, it is possible to show dependencies and connections when parsing text.

As a result, the capabilities offered make it possible to analyse without developing new additional applications.

```

{
  "_index": "index-fb-newspapers",
  "_id": "Othr84kByaaJPtkSDQlv",
  "_score": 1,
  "_source": {
    "date": "September 28",
    "hour": "5:40 AM",
    "text": " On the tenth anniversary of his #Forbes400 debut Elon Musk brought a decisive end to Jeff Bezos' four-year run as America's richest person. https://trib.al/oEmnxHL",
    "number_of_like": 0,
    "number_of_comments": 24,
    "number_of_shares": 0
  }
},
{
  "_index": "index-fb-newspapers",
  "_id": "PNhr84kByaaJPtkShwqk",
  "_score": 1,
  "_ignored": [
    "text.keyword"
  ],
  "_source": {
    "date": "September 28",
    "hour": "4:15 AM",
    "text": " A new survey commissioned by Coursera reveals that large majorities of employers and students view short-term industry certificates as a worthwhile addition to a college degree and a valuable credential in the hiring and job-seeking process. https://trib.al/g3X6jyR",
    "number_of_like": 26,
    "number_of_comments": 5,
    "number_of_shares": 4
  }
}
    
```

Figure 8 – An example of the data structure of an added user activity record from a social media news post to Elasticsearch

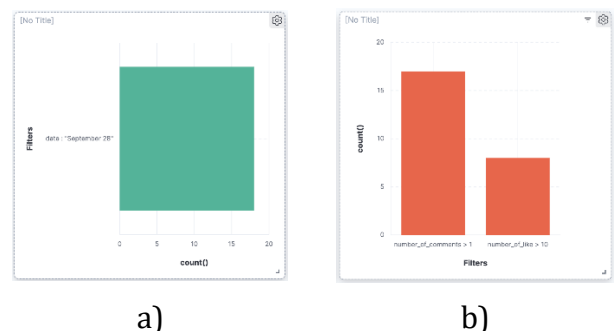


Figure 9 – Examples of customisation of visualisations for analytics on the Kibana dashboard: a) – the number of records read from the news on September 28, b) – the number of comments and likes, respectively, on September 28 in posts

Modern developments are well thought out, allowing data warehouse creation at a higher level. In addition to using tools for processing Kafka data streaming, the EKL stack can be used as a searchable data store.

In designing a data warehouse for storing and analysing users' activities in social networks, the EKL stack allows you to cover the main tasks. This technology stack is freely available and, in most cases, costs nothing.

Compared to Kafka, Elasticsearch with integrated Kibana and Logstash can visualise results for analysis and store data. To work with text, taking into account the search engine built in Elasticsearch, it is possible to carry out a more in-depth analysis of readers in the future.

One of the essential and challenging operations is the configuration and integration between these technology stacks. This is because each EKL stack runs in a separate console and has its settings. The advantage of such a solution is the possibility of using the same technologies in cloud services for global access to results.

Non-relational Elasticsearch database based on JSON format will provide more structured data exchange. The division into the number of indexes with stored data can be expanded to be used for more information from more sources and topics.

CONCLUSIONS

The paper describes converting textual semi-structured data into JSON format and sending groups to the Elasticsearch database. Thus, this process is automated using a bash script and is fast enough to populate the database. The method of installing and configuring the main parameters is described. The data warehouse was also designed, and the process of data work in Elasticsearch and data visualisation in Kibana was described.

Indexes are divided according to the principle of data collection sources. The work results can be used as test data for machine learning models implemented for natural language processing.

REFERENCES

1. Mysiuk I., Mysiuk R., & Shuvar R. (2023). Collecting and analyzing news from newspaper posts in facebook using machine learning. *Artificial Intelligence*, 28(1), 147–154. doi: [10.15407/jai2023.01.147](https://doi.org/10.15407/jai2023.01.147)
2. Mysiuk, I., Mysiuk, R., Shuvar, R., & Yuzevych, V. (2022). Methods of Analytics of Big Data of Popular Electronic Newspapers on Facebook. *Electronics and Information Technologies*, 19, 66-74. doi: [10.30970/eli.19.6](https://doi.org/10.30970/eli.19.6)
3. Manias, G., Mavrogiorgou, A., Kiourtis, A., Kakomitas, D., & Kyriazis, D. (2021). Real-Time Kafka-Based Topic Modeling and Identification of Tweets. *2021 IEEE International Conference on Progress in Informatics and Computing (PIC)*. doi: [10.1109/pic53636.2021.9687024](https://doi.org/10.1109/pic53636.2021.9687024)
4. Raptis, T. P., & Passarella, A. (2023). A Survey on Networked Data Streaming with Apache Kafka. *IEEE Access*, 1–1. doi: [10.1109/access.2023.3303810](https://doi.org/10.1109/access.2023.3303810)
5. Mysiuk, R., & Yuzevych, V. (2023). Recover Data about Detected Defects of Underground Metal Elements of Constructions in Amazon Elasticsearch Service. *Path of Science*, 9(1), 1011–1019. doi: [10.22178/pos.89-9](https://doi.org/10.22178/pos.89-9)
6. SaaSHub. (2023). *ElasticSearch VS Kafka*. Retrieved from <https://www.saashub.com/compare-elasticsearch-vs-kafka>
7. MirBozorgi. (2023). *Spark, Kafka, Cassandra and Elasticsearch applications*. Retrieved from <https://mirbozorgi.com/spark-vs-kafka-vs-cassandra>
8. Elastic. (2023). *Download Elasticsearch*. Retrieved from <https://www.elastic.co/downloads/elasticsearch>
9. Maffeo, L. (2019, July 11). *How to install Elasticsearch on MacOS*. Retrieved from <https://opensource.com/article/19/7/installing-elasticsearch-macos>
10. Elastic. (2023). *Download Kibana*. Retrieved from <https://www.elastic.co/downloads/kibana>