Restaurant Data Analysis using Elasticsearch

Deepanshu Sehgal
Student ID: 40225333
Concordia University
Montreal, Canada
deepanshusehgal0069
@gmail.com

Varun Aggarwal
Student ID: 40225335
Concordia University
Montreal, Canada
varunsince1998
@gmail.com

Mukul Kumar
Student ID: 40225304
Concordia University
Montreal, Canada
mukulsharma1998
@gmail.com

Sushant Sharma
Student ID: 40227986
Concordia University
Montreal, Canada
sushant.sharma9871
@gmail.com

ABSTRACT

Nowadays there is a huge requirement for distributed system designs which refer to a collection of independent computers or nodes acting as a single coherent system. Many organizations around the world use multiple software applications that produce large quantities of data in the form of logs. As data generation and usage is increasing day by day, special big database systems are needed to store and manage these logs. Elasticsearch is a NoSQL system that stores large amounts of data by distributing them into smaller chunks called nodes and then processing them together simultaneously, thus achieving parallelization. Elasticsearch is a part of the ELK stack that includes Logstash which is responsible for the aggregation of data and moving it further onto Elasticsearch where the data is stored and queried. Finally, it reaches Kibana which acts as a frontend to show real-time analysis and insights of data in a visual dashboard. The ELK when implemented on Docker achieves virtualization as the nodes of ELK stacks are created as containers and thus enhancing the power of distributed computing. Elasticsearch is distributed in nature, and we intend to study and implement multiple concepts of distributed computing such as containerization, clustering, scalability, and fault tolerance with the help of the system created.

I. INTRODUCTION

After March 2020, COVID-19 sparked a global public health catastrophe and a series of additional concerns, including an economic downturn, unemployment, and mental instability. The lockdowns amid the pandemic has affected the lives of people in a such way that most of them around the world are suffering from anxiety, stress, worry and poignancy in addition to illness. The pandemic had ill effects on the restaurant and hospitality businesses as well. Lockdowns, social distancing, hygiene effective measures, travel restrictions and stay at home ordering were part of the measures to prevent COVID transmission which in turn led to the closure of many restaurants leading to heavy losses in the business. The new normal quoted by many led to significant new consumer trends, implicitly, new global trends across different industries affected by pandemic and the restaurant industry was no exception. The restaurant and hospitality businesses were at its peak before the peak covid era and every restaurant was forced to limit their business exclusively to just delivery at door step.

A. Problem Statement

The problem is to analyze the reviews of restaurants available for the pre-COVID era and visualize the data to show the trends of how good a restaurant is on the basis of user reviews. Users generally don't have real time access to the visual statistics which might help them to choose a restaurant to go to for specific purposes with the help of reviews that others have provided.

B. Objective

The major objective of the project is to analyze the precovid restaurant dataset fed into the NoSQL systems and get insights of data with help of visualizations on a dashboard. The aim is to included distributed system functionalities in a way such that the system is scalable, fault tolerant and reduce the processing time with the help of parallelization.

C. Scope

The scope of this project is to apply and practice the distributed system concepts with real applications, such as NoSQL systems for big data and appreciate the power of distributed computing in extracting, analyzing the present the big data.

D. Outcomes

The system is intended to produce implementation of ELK Stack which includes Elasticsearch, Logstash and Kibana. The system should be able to show real time insights in a visual way such that the proposed system is scalable by handling multiple nodes at the same time, fault tolerant with the system has no effect of failures in case a node fails to start. The system should be able to run the all the nodes in a parallel way concurrently and shows consistent information whenever it is made to run.

II. BACKGROUND

A. ELK Stack

ELK stack has become one of the most powerful and popular log analysis tool for software driven businesses with multiple organizations relying on ELK for log analysis and management. The three components of the stack include Elasticsearch, Logstash and Kibana which combine together to form a powerful solution for aggregating, managing and querying log data from the pre fed or cloud based IT environments. With the help of log analytics, the elk stack provides critical visibility of IT assets and infrastructure for software-dependent organizations, satisfying use cases like security analytics and application troubleshooting. It is open source which means its free to download and let the users install plug ins and even modify the source code.

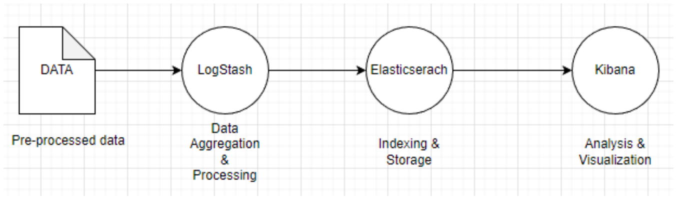


Fig. 1: ELK Stack Architecture

B. Logstash

Logstash released by Elastic is a server-side data processing pipeline that can ingest logs from a variety of data sources, applying transformations to the log data and send it to Elasticsearch cluster for indexing and analysis. Its main task is to read variety of data sources including system logs, application logs, windows event logs and more. Logstash is deployed with one server acting as a node based on user requirements for flexibility, scalability, and data privacy. The Logstash pipeline uses input plugins to aggregate data from multiple sources into a queue and the data in the queue is micro processed using the filter plugins and finally the output plugins format the logs and send them on to the Elasticsearch.

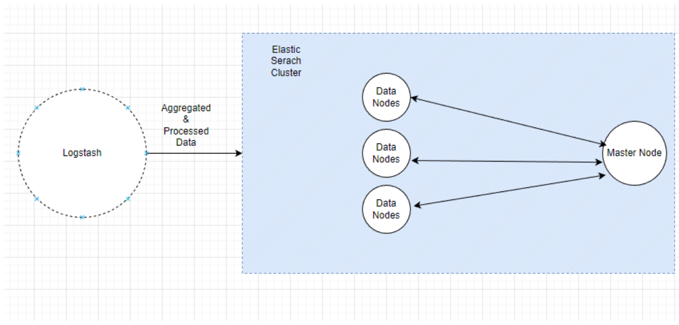


Fig. 2: Logstash

C. Elasticsearch

Elasticsearch is a NoSQL, distributed database that stores, retrieves, and manages the document- oriented and semi structured data. It is a Java based open-source tool that can search and index document files in diverse formats. It uses RESTful APIs to integrate with data ingestion tools like Logstash that send data to Elasticsearch in JSON document format. An Elasticsearch instance is known as node and these nodes can be deployed and managed in an Elasticsearch cluster, which is a coordinated group of nodes that share the same cluster.name attribute. The addition of more nodes to cluster increases the processing capabilities and reliability of the system.

D. Kibana

Kibana is a front-end tool for Elasticsearch which allows querying the underlying logs using Kibana Query Language. Performing text-based searches on the data and creating visualizations of the logs in form of line graphs, histograms, pie-charts and more are some of the key functionalities provided by Kibana. Kibana Dashboards let the user to view all the visualizations in one single window delivering real time analysis and insights as log data flows from Logstash into Elasticsearch.

III. DISTRIBUTED SYSTEM DESIGN OF ELASTIC SEARCH

A. Containerization

Containerization is a type of virtualization in which software code is packaged with just the OS libraries and dependencies needed to run the code, resulting in a single, lightweight executable called a container that reliably functions on any infrastructure. It enables developers to more quickly and securely construct and deploy apps. Docker is a collection of platform-as-a-service tools that distribute software in packages known as containers using OS-level virtualization. It is a set of software tools for creating, executing, and controlling containers on servers and in the cloud. The Docker Engine is the application used to host the containers.

B. Elasticsearch Cluster

Nodes present in an Elasticsearch cluster are combinations of master-eligible/non-master-eligible and data/non-data nodes. Every Elasticsearch cluster has a voting configuration, which is the collection of master-eligible nodes whose votes are taken into consideration while making decisions like choosing a new master or committing a change in cluster state. Only when the voting configuration receives responses from the majority (more than half) of the nodes is a decision made. To make the cluster as resilient as feasible, Elasticsearch responds to each node joining or leaving the cluster by automatically changing the voting configuration in accordance. Before you take further nodes out of the cluster, it's crucial to wait for this adjustment to finish.

C. Scalability

Elasticsearch is designed to scale to meet your demands and be always accessible. It accomplishes this by being distributed by nature. Elasticsearch automatically distributes your data and query load over all of the available nodes via sharding when you add more servers (or nodes) to a cluster to boost capacity. Elasticsearch data is kept in an index that has been divided into a number of shards, which distribute data throughout a cluster. These shards are replicated and dispersed throughout the cluster in order to provide high availability, resulting in horizontal scaling. Sharding also enhances performance by parallel processing of query in mutilple shards simultaneously.

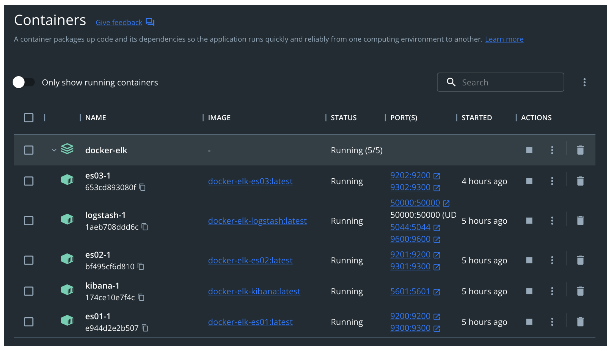
D. Fault Tolerance

Elasticsearch can provide redundancy, which both guards against hardware failures and boosts query capacity when nodes are added to a cluster, by distributing the documents in an index across several shards and distributing those shards across numerous nodes. Elasticsearch automatically migrates shards to rebalance the cluster as it expands or shrinks. Replicas are used to serve read requests, such as document retrieval or search requests, and to provide redundant copies of your data to guard against hardware failure. Shard replications increase fault tolerance since they continue to function even if the primary shard is down.

IV. DEMONSTRATION

A. Install Elasticsearch with Docker

Docker images for Elasticsearch are also accessible. Docker pull command can be used to get Elasticsearch for Docker via the Elastic Docker registry. Using Docker Compose and a docker-compose.yml file, you can quickly and easily set up a three-node Elasticsearch cluster in Docker. A three-node Elasticsearch cluster is created using this file. Node es01 runs on localhost:9200, and es02 and es03 communicate with es01 over the Docker network. To start the cluster, run docker-compose. To check if the nodes are operational, do a cat/nodes request.



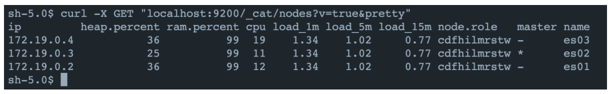
Containers Give feedback

A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. Learn more

Only show running containers

NAME	IMAGE	STATUS	PORT(S)	STARTED	ACTIONS
docker-elk	-	Running (5/5)			
es03-1 653cd893080f	docker-elk-es03:latest	Running	9202:9200 9302:9300	4 hours ago	
logstash-1 1aeb708ddd6c	docker-elk-logstash:latest	Running	50000:50000 50000:50000 (UD 5044:5044 9600:9600	5 hours ago	
es02-1 bf495cf6d810	docker-elk-es02:latest	Running	9201:9200 9301:9300	5 hours ago	
kibana-1 174ce10e7f4c	docker-elk-kibana:latest	Running	5601:5601	5 hours ago	
es01-1 e944d2e2b507	docker-elk-es01:latest	Running	9200:9200 9300:9300	5 hours ago	

Fig. 3: Containers in dockers

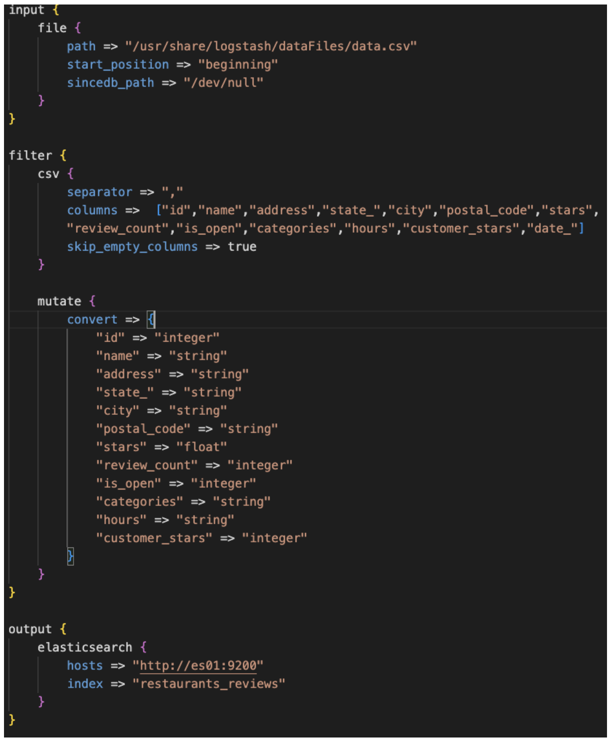


```
sh-5.0$ curl -X GET "localhost:9200/_cat/nodes?v=true&pretty"
ip          heap.percent ram.percent cpu load_1m load_5m load_15m node.role   master name
172.19.0.4            36          99  19    1.34    1.02     0.77 cdfhilmrstw -      es03
172.19.0.3            25          99  11    1.34    1.02     0.77 cdfhilmrstw *      es02
172.19.0.2            36          99  12    1.34    1.02     0.77 cdfhilmrstw -      es01
sh-5.0$
```

Fig. 4: Nodes

B. Loading the dataset

Cleaning and preprocessing of raw dataset is very important before loading it into the system. It is achived my handling null values and dropping unwanted columns. Logstash is used to load CSV File into Elasticsearch. Create a pipeline.config file with appropriate configuration of the index. Run the Logstash command in terminal to execute this config file and csv data will start loading in Elasticsearch cluster.

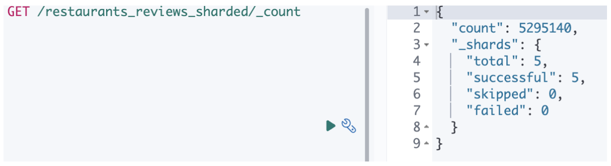


```
input {
    file {
        path => "/usr/share/logstash/dataFiles/data.csv"
        start_position => "beginning"
        sincedb_path => "/dev/null"
    }
}

filter {
    csv {
        separator => ","
        columns =>  ["id","name","address","state_","city","postal_code","stars",
        "review_count","is_open","categories","hours","customer_stars","date_"]
        skip_empty_columns => true
    }

    mutate {
        convert => {
            "id" => "integer"
            "name" => "string"
            "address" => "string"
            "state_" => "string"
            "city" => "string"
            "postal_code" => "string"
            "stars" => "float"
            "review_count" => "integer"
            "is_open" => "integer"
            "categories" => "string"
            "hours" => "string"
            "customer_stars" => "integer"
        }
    }
}

output {
    elasticsearch {
        hosts => "http://es01:9200"
        index => "restaurants_reviews"
    }
}
```

Fig. 5: Logstash Config File

C. Getting started with Kibana

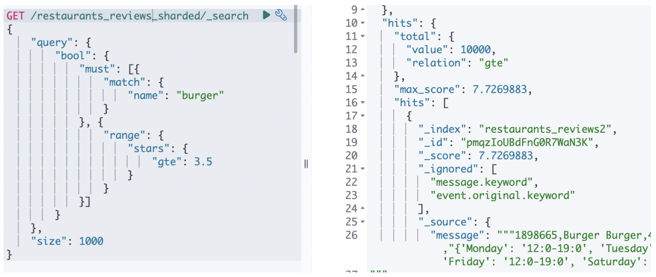
You can visualise your Elasticsearch data and move around the Elastic Stack with the help of Kibana, a free and open user interface. You can accomplish anything, from monitoring query load to comprehending how requests go through your apps. To access the console in Kibana, select Dev Tools from the left menu panel. Use GET, PUT POST command to interact with the index(csv dataset). Create complex searches with wildcard characters, searches across several fields, and more using the string query.



```
GET /restaurants_reviews_sharded/_count
```

```
{
  "count": 5295140,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  }
}
```

Fig. 6: Count Query



```
GET /restaurants_reviews_sharded/_search
{
  "query": {
    "bool": {
      "must": [{
        "match": {
          "name": "burger"
        }
      }, {
        "range": {
          "stars": {
            "gte": 3.5
          }
        }
      }]
    }
  },
  "size": 1000
}
```

```
  },
  "hits": {
    "total": {
      "value": 10000,
      "relation": "gte"
    },
    "max_score": 7.7269883,
    "hits": [
      {
        "_index": "restaurants_reviews2",
        "_id": "pmqzIoUBdFnG0R7WaN3K",
        "_score": 7.7269883,
        "_ignored": [
          "message.keyword",
          "event.original.keyword"
        ],
        "_source": {
          "message": """1898665,Burger Burger,4
            ,"{'Monday': '12:0-19:0', 'Tuesday'
            'Friday': '12:0-19:0', 'Saturday':
```

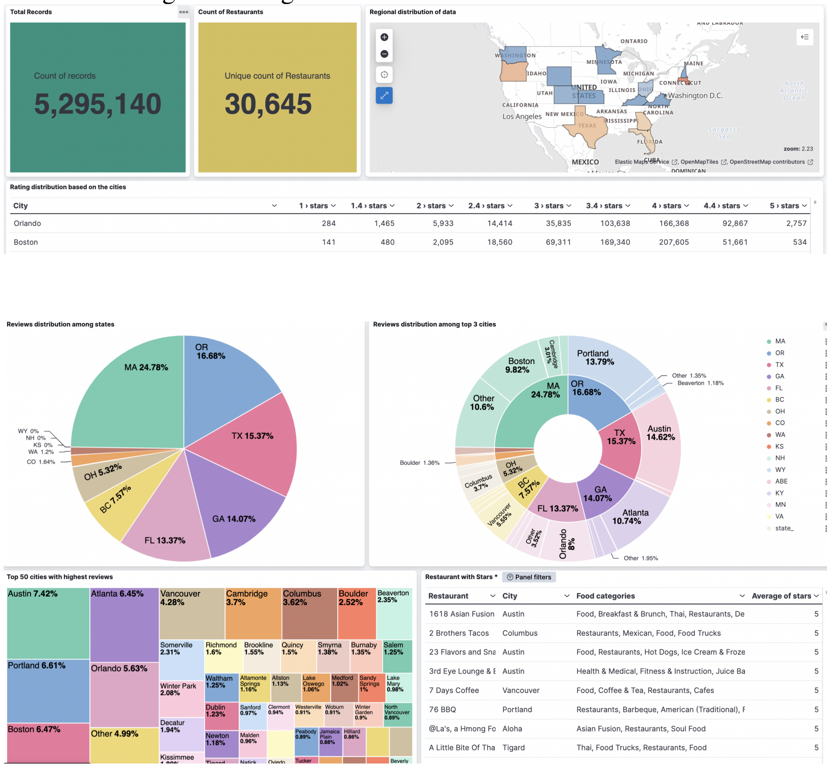
Fig. 7: Search Query

D. Sharding of the index

Elasticsearch data is kept in an index that has been divided into a number of shards and re which distribute data throughout a cluster. Create 5 shards of the index to divide the data into three equal parts among three nodes.

E. Vizualizing and Analyzing

Select Dashboard option from the left panel to create custom vizualizations of the dataset. Through numerous plots, charts, graphs, and maps, the user-friendly user interface enables the creation of diagrams using indexed Elasticsearch data.



Rating distribution based on the cities

City	1 › stars	1.4 › stars	2 › stars	2.4 › stars	3 › stars	3.4 › stars	4 › stars	4.4 › stars	5 › stars
Orlando	284	1,465	5,933	14,414	35,835	103,638	166,368	92,867	2,757
Boston	141	480	2,095	18,560	69,311	169,340	207,605	51,661	534

Restaurant with Stars * Panel filters

Restaurant	City	Food categories	Average of stars
1618 Asian Fusion	Austin	Food, Breakfast & Brunch, Thai, Restaurants, De	5
2 Brothers Tacos	Columbus	Restaurants, Mexican, Food, Food Trucks	5
23 Flavors and Sna	Austin	Food, Restaurants, Hot Dogs, Ice Cream & Froze	5
3rd Eye Lounge & E	Austin	Health & Medical, Fitness & Instruction, Juice Ba	5
7 Days Coffee	Vancouver	Food, Coffee & Tea, Restaurants, Cafes	5
76 BBQ	Portland	Restaurants, Barbeque, American (Traditional), F	5
@La's, a Hmong Fo	Aloha	Asian Fusion, Restaurants, Soul Food	5
A Little Bite Of Tha	Tigard	Thai, Food Trucks, Restaurants, Food	5

Fig. 8: Kibana Dashboard

V. CONCLUSION

As we studied in the above sections, we can infer that the ELK stack was implemented on Docker using the platform as a service (PaaS) thus achieving containerization. The system created was able to demonstrate and appreciate the power of distributed computing by achieving clustering through the Elasticsearch cluster, scalability, and parallelization with the help of sharding and fault tolerance by creating replicas of all the data stored in the nodes. The system also had a visualization dashboard using Kibana where one can get key insights and real-time analysis of the stored data.

VI. FUTURE WORKS

Deploy Elastic with cloud-native capabilities in any public cloud, including AWS, Azure, and GCP, to accelerate important results at scale and with security.
Implementing extra security measures in place, such as auditing, IP filtering, and role-based access control.