ELK Tool

➔ **Elasticsearch :-** Elasticsearch is a NoSQL database and analytics engine, which can process any type of data, structured or unstructured, textual or numerical. it is free, open-source, and distributed in nature.
Elasticsearch is the main component of ELK Stack (also known as the Elastic Stack), which stands for Elasticsearch, Logstash, and Kibana. These free tools combine to offer scalability, speed and simple REST APIs for data analysis, visualization and storage. ELK Stack also offers a range of Beats (lightweight agents) to send data from various IT systems to Elasticsearch.

Elasticsearch is most commonly used for:
Enterprise search:- enabling fast text search of large, constantly updated data sets
Observability:- performing log analysis and assisting with monitoring of IT systems, identifying trends and problems
Security:- analyzing security events from multiple IT systems and security tools to identify threats and perform forensic analysis

"Elasticsearch comes with simple REST APIs and provides features for scalability and fast search."

Elasticsearch Architecture: 7 Key Components

- **Elasticsearch Cluster :-** An Elasticsearch cluster is composed of a group of nodes that store data. You can specify the number of nodes that start running with the cluster, as well as the IP address of the virtual or physical server. You can specify this information in the config/elasticsearch.yml file, which contains all configuration settings.
  Nodes in an Elasticsearch cluster are connected to each other, and each node contains a small chunk of cluster data. You can run as many clusters as needed. However, usually one node is sufficient. The system automatically creates a cluster when a new node starts. The nodes participate in the overall cluster processes in charge of searching and indexing.

- **Elasticsearch Node :-** A Node in Elasticsearch is a single server or process that is part of a cluster. It stores data and participates in the cluster's activities, such as indexing, searching, and replicating data.
  A node is identified by a unique name (automatically generated but can be customized).

  The three main options to configure an Elasticsearch node

  - **Master node :-** The Master Node is responsible for managing cluster-wide settings, such as creating or deleting indices, tracking the status of nodes, and maintaining cluster metadata..
    It does not hold data itself (unless configured to do so), but it is essential for cluster management.

  - **Data Node :-** The Data Node is where the actual data is stored and queried.

Data nodes hold indices, shards, and handle the CRUD operations (Create, Read, Update, Delete).
Most nodes in a typical Elasticsearch cluster are data nodes.

- **Client Node (Coordinating Node):-** serves as a load balancer that routs incoming requests to various cluster nodes**.** Client Node acts as a coordinator that receives client requests and forwards them to the appropriate data nodes.

- **Ingest Node:-** The Ingest Node is responsible for pre-processing and transforming documents before they are indexed in Elasticsearch. It can be configured to perform operations like enriching data, parsing logs, or modifying data (e.g., adding timestamps or geo-data)

➢ **<u>The port 9200 & 9300</u>**

- **Port 9200 (HTTP Communication or Client to Node) :-** Port 9200 is used for HTTP communication between client applications (like browsers, curl commands, Kibana, or other services) and Elasticsearch nodes.
  It is the default port for REST API requests.
  Clients interact with Elasticsearch over HTTP using this port for operations like indexing data, querying, Searching data,Indexing documents, Managing clusters, indices, and mappings
  Kibana: Kibana, which is the web interface for Elasticsearch, communicates with Elasticsearch nodes through port 9200 for querying data and visualizing results.
  Configuring firewall rules on 9200 port to limit access to specific IP addresses.

- **Port 9300: Transport Communication (Node to Node):-** Port 9300 is used for internal communication between Elasticsearch nodes within a cluster.
  It allows Elasticsearch nodes to share data about the cluster state, such as node discovery, replication ,metadata, shard allocation, and node health).
  Synchronize data replication between nodes.
  if a master node fails, the other nodes communicate via port 9300 to elect a new master.
  Since port 9300 is used for internal communication, access should be restricted to the internal network. By using SSL/TLS encryption and authentication.

➢ **Elasticsearch Documents :-** In Elasticsearch, a Document is a basic unit of information that can be indexed and stored. It represents a single record in a collection of data and is typically represented in JSON (JavaScript Object Notation) format.

1. Document Structure :- A document is a JSON object containing fields and values

```
{
 "title": "Elasticsearch Guide",
 "author": "John Doe",
 "publish_date": "2025-01-01",
 "content": "This is a guide to Elasticsearch."
}
```

2. Fields:- Each document can contain multiple fields (key-value pairs). Fields are used for storing data in a structured manner (e.g., strings, dates, numbers).

3. Indexing:- Documents are indexed in Elasticsearch to enable fast search and retrieval. The indexing process involves assigning a unique identifier (ID) to each document. Index similar like Row In DBMS . where each Row has Unique value and Store information for one thing.

4. Unique ID:- Each document has a unique ID within an index, which can either be auto-generated or explicitly defined. Similarly in Traditional Data base.

5. Document vs. Record:- A document in Elasticsearch is similar to a record in a traditional database table, but with more flexibility and scalability. It similarly Like Table which we make in Our DBMS.

6. Schema-less:- Elasticsearch is schema-less by default. This means documents can have different fields, but it automatically detects and maps field types.

7. Indexing and Searching:- Once documents are indexed, they can be searched efficiently using queries, making Elasticsearch a powerful tool for full-text search and analysis.

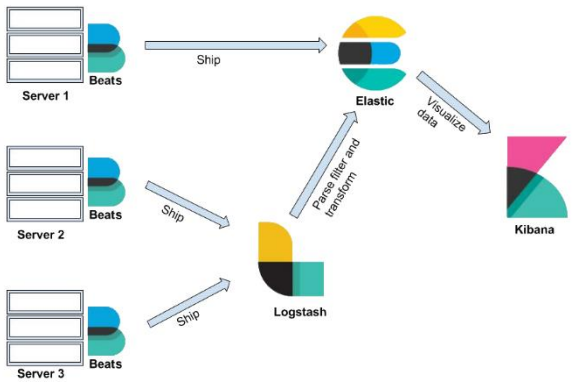
- **Elasticsearch Shards:-** In Elasticsearch, a shard is a basic unit of storage and distribution of data. When data is indexed in Elasticsearch, it is automatically divided into smaller pieces called shards. Each shard is a self-contained index that can be stored on any node in the cluster. Shards allow Elasticsearch to horizontally scale, making it capable of handling large volumes of data and queries efficiently.
  Shards are distributed across the nodes in the cluster. Elasticsearch automatically manages the allocation of shards based on available disk space and resources, ensuring efficient distribution and performance.
  Advantages of Sharding in Elasticsearch:
  1. **Horizontal Scalability:-** Elasticsearch distribute data at multiple machine using node so we can create and Delete node and shards too.
  2. **Improved Performance:-** Searching is happen parallelly in its shards and Index so that it speeds up query response times

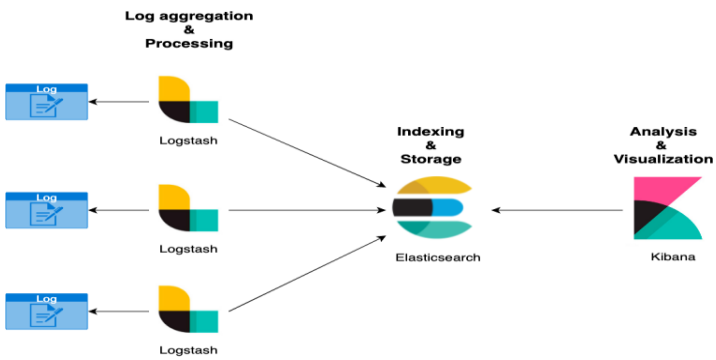
3. **Fault Tolerance and High Availability:-** each shards have its replica so In the case of primary shards failure, we have backup data in Replica shards.
4. **Better Resource Utilization:-** Sharding ensures that data is distributed evenly across nodes, preventing any single node from becoming a bottleneck and helping to optimize resource usage (e.g., CPU, memory, and disk).

- **Elasticsearch Replicas:-** In Elasticsearch, replicas are copies of index shards. Replicas are used as a fail-safe mechanism for backup and recovery purposes. Replicas are never placed on the node containing the original (primary) shards**.**
  To ensure availability, replicas are stored in different locations. You can define replicase after the index is created and create as many replicas as needed. This means you can store more replicas than primary shards.

- **How does Elasticsearch Work ?**
  Elasticsearch uses shipping agents, called beats OR elasticAegnt to transfer raw data from multiple sources into Elasticsearch. After data is shipped into Elasticsearch, the engine runs data ingestion processes, which parse, normalize, enrich, and prepare data for indexing. After the data is indexed, users can run complex queries and use aggregations to retrieve complex data summaries.

  For visualization and management, the Elastic Stack offers a tool called Kibana, which enables users to create real-time data visualizations, such as pie charts, maps, line graphs, and histograms. Kibana also lets you share dashboards, use Canvas to create custom dynamic infographics, and use Elastic Maps to visualize geospatial data.





Kibana and Its working

- **Kibana :-** Kibana is an open-source data visualization and exploration tool used to analyze and visualize data stored in Elasticsearch. It is a part of the Elastic Stack (formerly known as the ELK Stack, which consists of Elasticsearch, Logstash, and Kibana), helping

users to visualize large amounts of log and time-series data, create dashboards, and gain Insights.

- **key aspects of Kibana:-**
  1. **Data Visualization:-** Kibana is primarily used to visualize the data stored in Elasticsearch, providing a user-friendly interface to query, analyze, and visualize data.
  2. **Dashboards:-** It allows you to create and share dashboards, making it easier to display and interpret data.
  3. **Search & Analyze:-** Users can create search queries to filter and explore data stored in Elasticsearch in real-time.
  4. **Integration with Elastic Stack:-** Kibana seamlessly integrates with other Elastic Stack components like Elasticsearch, Logstash, and Beats.

➢ **Architecture of Kibana:-** Kibana follows a client-server architecture, where it acts as a web-based front-end interface for interacting with Elasticsearch. The core components of Kibana's architecture include:

- **Kibana Server:-** The server part is built using Node.js and is responsible for serving the Kibana UI (the dashboard) and handling requests from the web browser.
- **Elasticsearch:-** Kibana directly connects to an Elasticsearch cluster to retrieve and visualize data. All the data resides in Elasticsearch, and Kibana is responsible for presenting this data in a meaningful way.
- **Kibana Frontend:-** This is the user interface (UI) part that interacts with the Kibana server. It is a web-based application typically accessed via a browser.
- **Data Sources:-** Elasticsearch is the main data source, but Kibana can also interact with data from external sources, depending on plugins or integrations used.
- **Visualization Layer**:- Kibana provides various visualization options like bar charts, line charts, pie charts, maps, etc.
- **Kibana Plugins:-** Kibana supports various plugins (such as monitoring, reporting, alerting, and more) that enhance its functionality.

➢ **Kibana Ports:-** By default, Kibana runs on the following ports:
Default Port: 5601 (This is the default port for accessing Kibana's web interface)
Port Usage: This port can be changed in the configuration file if needed.

➢ **SSL Certificate and HTTPS Configuration:-** To secure communication between the Kibana server and its users, SSL (Secure Socket Layer) certificates can be configured. Kibana can be set up to use HTTPS instead of HTTP to encrypt traffic.

1. **Generate or Obtain an SSL Certificate:-** You need a valid SSL certificate to encrypt communication. You can generate a self-signed certificate or obtain one from a trusted Certificate Authority (CA).

2. **Configure Kibana to Use SSL:-** • Edit the kibana.yml configuration file to set up SSL for the Kibana server.

   Set the following properties:

   server.ssl.enabled: true (Enables SSL)

   server.ssl.certificate: /path/to/certificate.crt (Path to the certificate**)**

   server.ssl.key: /path/to/certificate.key (Path to the private key)

After configuration, restart the Kibana server to apply changes.

- **Kibana Features:-**
  - **Dashboards:** Create and share custom dashboards with visualizations.
  - **Real-Time Data Analysis**:- Kibana provides real-time insights, allowing you to monitor systems and logs in near real-time.
  - **Search Interface:-** Perform complex searches using Lucene query language or the more advanced KQL (Kibana Query Language).
  - **Advanced Visualizations:-** isualize data with different chart types, tables, and maps.
  - **Alerting:-** Kibana integrates with Elasticsearch alerting mechanisms to notify users of important changes or anomalies in the data.

- **Common Use Cases of Kibana:-**
  - **Log Analysis:** Often used for monitoring server logs, application logs, and security logs.
  - **System Monitoring**: Visualize metrics such as CPU usage, memory usage, disk I/O, and more from various sources (e.g., Beats).
  - **Security and Audit Data:-** Monitor and analyze data related to security, including intrusion detection, failed login attempts, etc.
  - **Business Intelligence**:- Kibana can be used for visualizing and analyzing business metrics, trends, and KPIs.

- **Kibana Plugins and Integrations:-** Kibana is highly extensible, with a variety of plugins and integrations available:
  - **Elasticsearch Plugins**:- Integrates seamlessly with Elasticsearch, including features like alerting, machine learning, etc.
  - **Beats and Logstash Integration:-** Kibana is often used with Beats and Logstash to ingest and process log data into Elasticsearch
  - **Machine Learning:-** Kibana can use machine learning features in the Elastic Stack to automatically detect anomalies in data.

- **Conclusion**

  Kibana is a powerful tool for data visualization and exploration in the Elastic Stack. It allows users to create dashboards, perform advanced searches, and visualize data stored in Elasticsearch. With features like SSL configuration, real-time data analysis, and integrations with other tools, Kibana is an essential tool for IT operations, security analysis, and business intelligence.

Logstash

- **What is Logstash?**

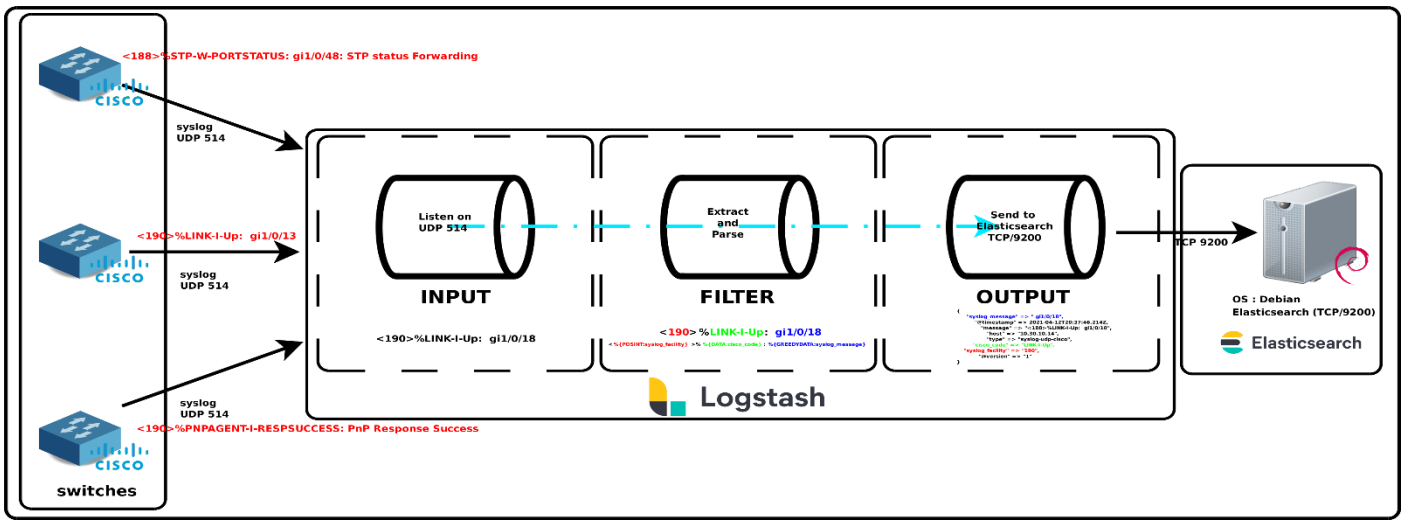
  Logstash is a server-side data processing pipeline that ingests data from multiple sources, transforms it, and then forwards it to Elasticsearch (or other storage systems) for storage, analysis, or further processing.

  It is one of the core components of the Elastic Stack (formerly known as ELK Stack, which includes Elasticsearch, Logstash, and Kibana). Logstash is primarily used for data collection, transformation, and routing.

  Logstash supports a wide variety of input, filter, and output plugins.

  - **Data Ingestion:-** Logstash is capable of ingesting data from various sources such as logs, metrics, databases, and APIs.
  - **Data Processing:-** It processes the data by applying filters to transform it into a structured format.
  - **Routing and Output:-** The processed data is then forwarded to one or more output destinations like Elasticsearch, a file, or another service

- **Architecture of Logstash:-** Logstash follows a Pipeline Architecture that consists of three main stages
  - **Input:-** The input stage is responsible for collecting or receiving data from various sources such as log files, message queues, databases, and other services.
    Common input plugins include beats, file, jdbc, syslog, etc.
  - **Filter:-** The filter stage processes the incoming data to transform it into a desired format.
    Filters can modify, enrich, or parse the data
    Common filter plugins include grok (for pattern matching), mutate (for modifying data), date (for date parsing), geoip, and csv.
  - **Output:-** The output stage is where processed data is sent to various destinations for further processing or storage.
    Common output plugins include Elasticsearch, file, stdout, email, and Kafka.



- **Logstash Ports** :- Logstash typically runs on port 5044 when receiving data from Beats (like Filebeat) or other services. However, Logstash itself does not have a default port for

general use, as its function is to listen for data on various ports depending on the input plugins.

- **Beats input (default):** 5044 (when using Filebeat, Metricbeat, or other Beats to send logs)
- **Custom Ports:** You can configure Logstash to listen on different ports or use different protocols based on the input plugins you are using (e.g., HTTP, TCP, UDP, etc.).

```
input {
    beats {
        port => 5044
    }
}
```

- **Logstash Features**:- Logstash provides a wide range of features that make it a powerful tool for data collection and transformation:
  - **Pluggable Architecture:** It supports a variety of plugins to connect to different data sources, apply transformations, and output data to multiple destinations.
  - **Multiple Input/Output Sources:-** It can integrate with various data sources like logs, databases, and message brokers, and output data to systems like Elasticsearch, Kafka, and more.
  - **Data Transformation:** Logstash enables extensive data transformation using filter plugins, such as Grok, Mutate, and Date.
  - **Event Routing**: Logstash can filter, modify, and route events to different outputs based on conditions or tags.

- **Common Use Cases of Logstash:-**
  - **Log Collection and Processing:-** Collect and aggregate logs from servers, applications, and network devices.
  - **Data Enrichment:** Enrich data with additional information, such as geo-location or IP geolocation, from external databases or services.
  - **Event Filtering:-** Filter out irrelevant or unnecessary events and forward important ones to Elasticsearch or other destinations.
  - **Metrics Aggregation:** Collect and aggregate system or application metrics, and send them to Elasticsearch for further analysis.

- **Conclusion**

Logstash is an essential part of the **Elastic Stack**, providing powerful data processing capabilities. It is primarily used for **ingesting**, **transforming**, and **routing** data from various sources into Elasticsearch or other destinations. Its flexible plugin architecture, support for real-time data processing, and extensive input/output integrations make it a versatile tool for log and data management.

Thanks and Regards

Karan Neelkanth
Software Quality Engineer
INDD, Indore
9111392442

References

Google.com
https://bluexp.netapp.com/blog/cvo-blg-elasticsearch-concepts-deployment-options-and-best-practices
https://chatgpt.com/
https://copilot.cloud.microsoft/?fromcode=cmc&redirectid=69B276614C6F4B89B2DC2BFB2307487A&auth=2