Elasticsearch Stack

Benmounah Zakaria
Constantine 2 University
SDIA M1
2023-2024

PART 01

Introduction & basics

Introduction to Elasticsearch

- **Definition:** It is a software application that provides advanced search and data analysis capabilities. It parses vast amounts of data quickly and returns search results in real-time.

- **Open-source:** As an open-source tool, developers from around the world can contribute to its improvement, ensuring a wide array of features and a robust architecture.

Advanced Search Capabilities

- **Auto-completion:** Elasticsearch predicts what users intend to type, enhancing user experience.

- **Typo Correction:** Even if users make mistakes in their queries, Elasticsearch tries to find the most relevant results.

- **Relevance Adjustment:** It prioritizes results based on specific criteria, ensuring the most pertinent information is displayed first.

Beyond Search - Analytics

- **Structured Data Querying:** Besides textual data, Elasticsearch can analyze and query structured datasets like numbers or dates.

- **Visual Analytics:** Leveraging its data aggregation abilities, users can visualize data in various formats to derive insights.

Event Monitoring and Analysis

- **Real-time Monitoring:** As events occur, Elasticsearch can capture and analyze them in real-time, enabling immediate insights.

- **Sales Analysis:** By ingesting sales data, businesses can identify best-selling products, peak sales times, and other crucial sales metrics.

Anomaly Detection

- **Learning Patterns:** Over time, Elasticsearch identifies “normal” behavior based on historical data.

- **Immediate Alerts:** Upon detecting deviations from the norm, Elasticsearch can instantly send notifications, ensuring quick remedial actions.

Elasticsearch Internals

- **Document-based Storage:** Each piece of data is stored as a document, facilitating quick retrieval and analysis.

- **REST API:** This interface allows developers to interact with Elasticsearch, sending queries and receiving responses.

Technical Background

- **Built on Lucene:** Apache Lucene, a high-performance, full-featured text search engine library, is the foundation of Elasticsearch, contributing to its efficacy.

- **Scalability:** Elasticsearch can handle increasing amounts of data and demand, making it suitable for businesses of all sizes.

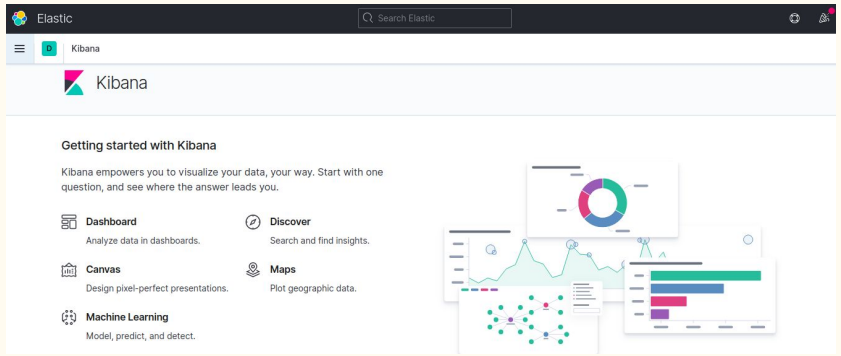
Potential applications

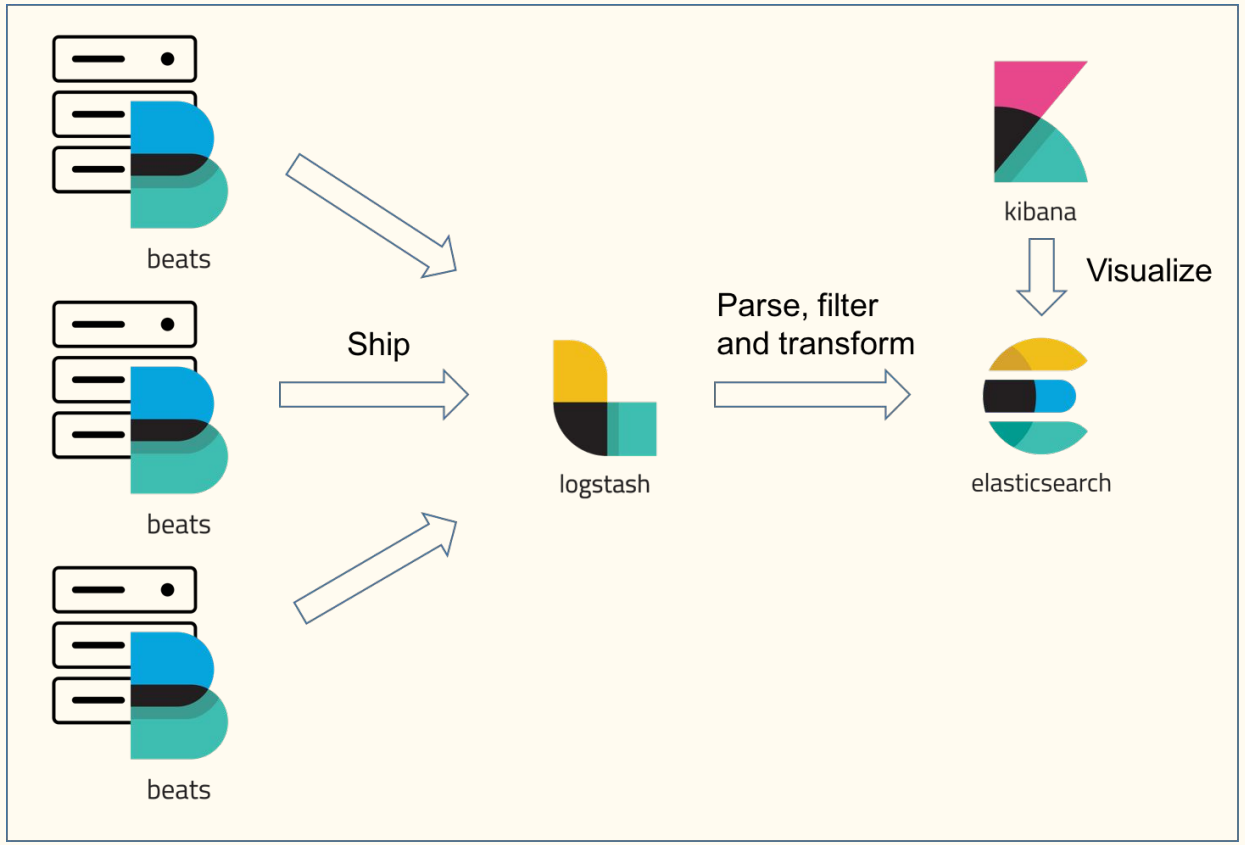
- Advanced Product Filtering
- Dynamic Pricing Analysis
- Customer Behavior Tracking
- Inventory Management and Forecasting
- Review & Rating Analysis
- Web security
- Web user behaviour

PART 02

Elastic stack
KIBANA

Elasticsearch ecosystem





Introduction to Kibana

What is Kibana?

- An integral part of the Elastic Stack.
- Specifically designed for visualizing Elasticsearch data.

Core Purpose:

- Transform raw data into meaningful visual insights.
- Helps in data exploration and understanding patterns.

Uses of Kibana

Diverse Visualization Tools:

- Bar graphs, coordinate maps, gauges, and more.

Real-Time Monitoring:

- Display instant data updates.
- Example: Track and show real-time website visitors' locations.

Browser Analytics:

- Understand user behavior and preferences.
- Optimize website design and functionality for the most popular browsers.

More Kibana Features

Change Detection & Forecasting:

- Predict future trends based on historical data.
- Set up alerts for anomalies or changes in patterns.

Elasticsearch Management:

- Simplify user authentication.
- Streamline authorization processes and permissions.

Web Interface:

- Direct, intuitive access to data in Elasticsearch.
- Reduce the learning curve for users unfamiliar with Elasticsearch's raw structure.

Kibana's Data Interaction

Data Retrieval:

- Extracts data from Elasticsearch seamlessly.
- Uses Elasticsearch's powerful REST API.

Query Interface:

- User-friendly GUI for building complex queries.
- No need for deep technical knowledge.

Benefits:

- Simplifies data extraction and visualization processes.
- Saves time and resources by negating the need for manual implementation.

Kibana Dashboards

Customizable & Role-Specific:

- Create unique dashboards tailored to different user roles.
- Display only relevant metrics and visualizations.

Examples:

- System Administrators: Monitor metrics like server uptime, CPU loads, memory usage.
- Developers: Track code exceptions, application downtimes, and API call frequencies.
- Management: Visualize company KPIs, such as sales trends, revenue growth, and customer demographics.

Data Diversity in Elasticsearch

Versatile Data Storage:

- Not limited to text-based data.
- Can store logs, metrics, and even geospatial information.

Beyond Search:

- While Elasticsearch excels in search, its analytics capabilities are equally robust.
- Paired with Kibana, it becomes a holistic data analysis tool.

Kibana Interface Overview

Visual Introduction:

Highlight different features and visualizations.

User Experience:

Emphasize the intuitive design and easy navigation.

Mention any customizable or drag-and-drop features.

Explore Kibana

Hands-on Exploration:

- Encourage users to dive into the public demo.
- Emphasize the benefits of real-world interaction.

What to Expect:

- Pre-configured dashboards to showcase Kibana's capabilities.
- Sample datasets to play with and visualize.

Link for Exploration:

- https://demo.elastic.co/app/kibana#/dashboard/1be3aae0-9406-11e8-8fa2-3d5f811fbd0f

PART 03

Elastic stack
LogStash

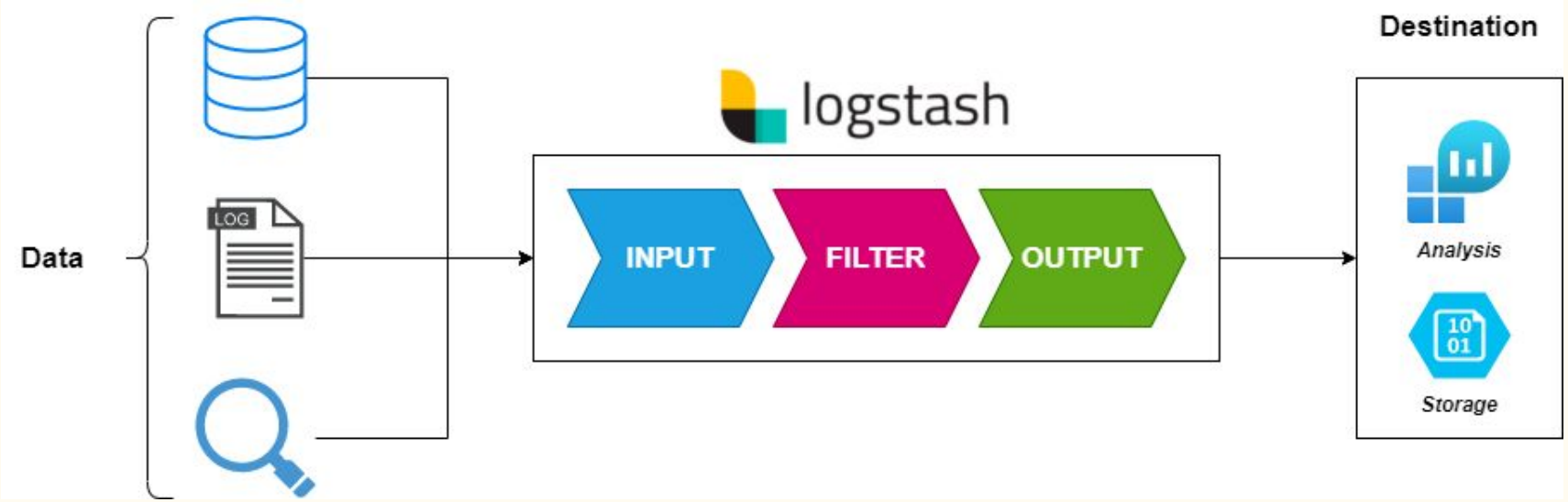
Introduction to Logstash

Definition:

- Originated for log processing for Elasticsearch.
- Today: A dynamic data processing tool.

Functionality:

- Interprets diverse data as 'events.'
- A broad event range: From logs to eCommerce orders.



Logstash's Evolution

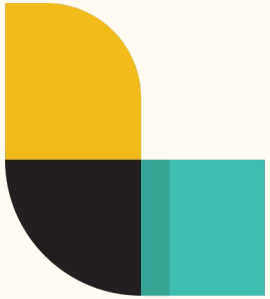
From Specific to General:

- Transition from a log-focused tool to a multi-purpose pipeline.
- A bridge between data sources and desired destinations.

Destinations Examples:

- Elasticsearch: For data visualization.
- Kafka queue: For data streaming.
- Email: For notifications/alerts.
- HTTP endpoint: For web-based data sharing.

What are Logstash Events?



- Logstash views data as "events".
- Events can range from log entries to chat messages, ecommerce orders, and more.

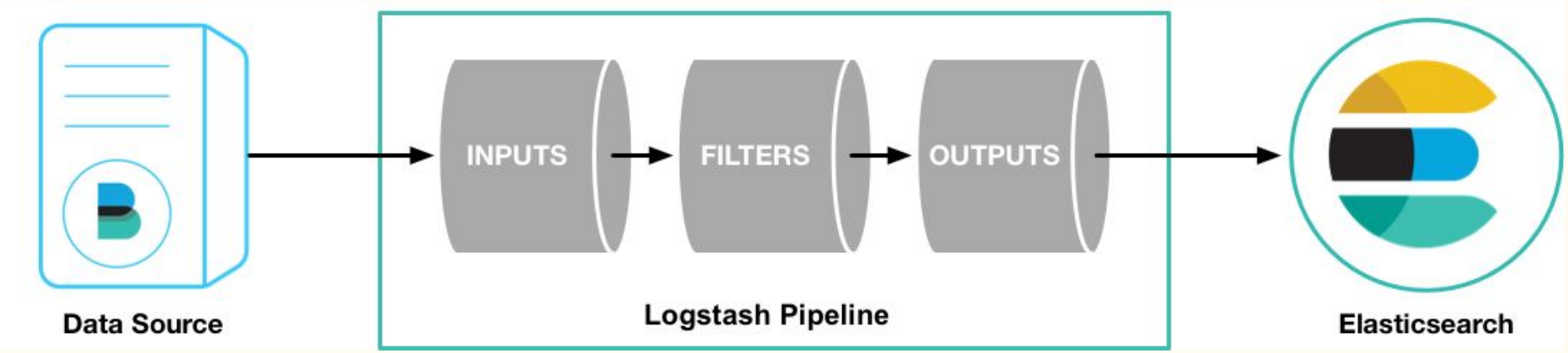
Anatomy of a Logstash Pipeline

Three Core Stages:

- Inputs: where data comes from.
- Filters: how data gets refined.
- Outputs: where data goes.

Plugins:

- Modular extensions to adapt and expand pipeline capabilities.



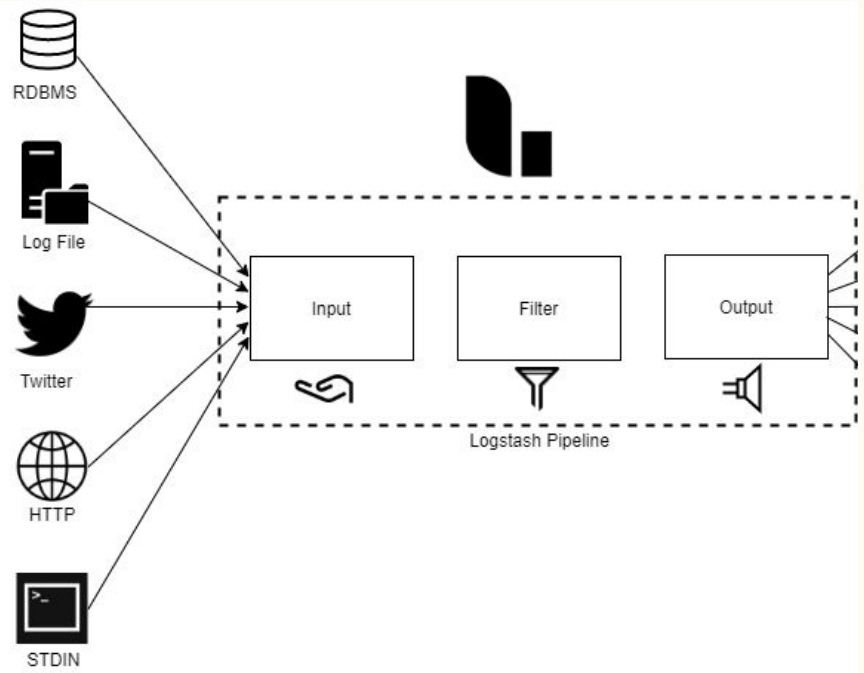
Input Plugins

Function:

- The starting point of data transformation.
- Facilitates diverse sources of data intake.

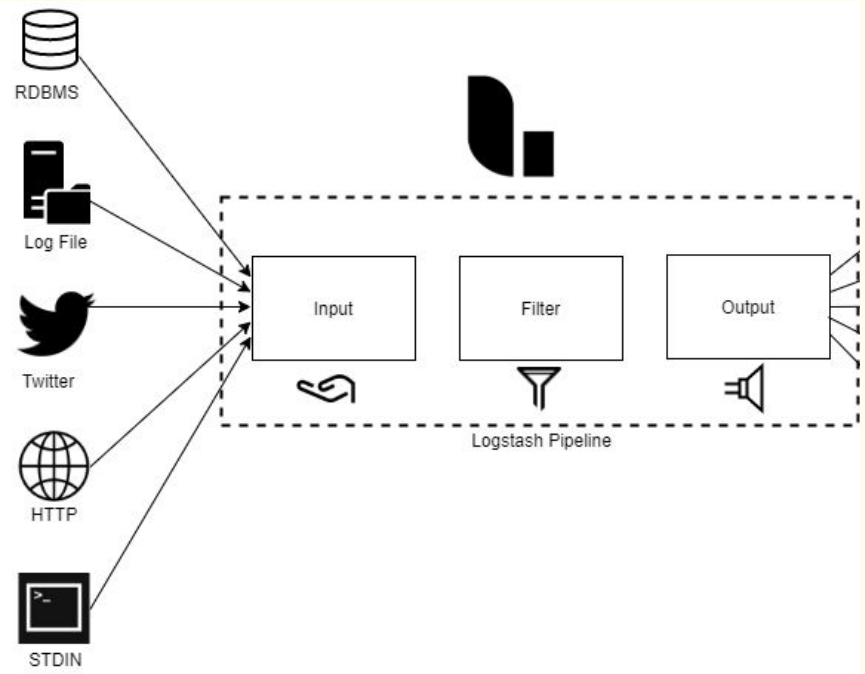
Examples:

- Files: Tail logs in real-time.
- HTTP requests: Direct data feeds.
- Kafka queues: Stream data from message brokers.



Filter Plugins

- Processing and transformation of events.
- Examples: Parsing CSV/XML/JSON, data enrichment (IP address lookup, database lookup).
- Image: A depiction of data being refined.



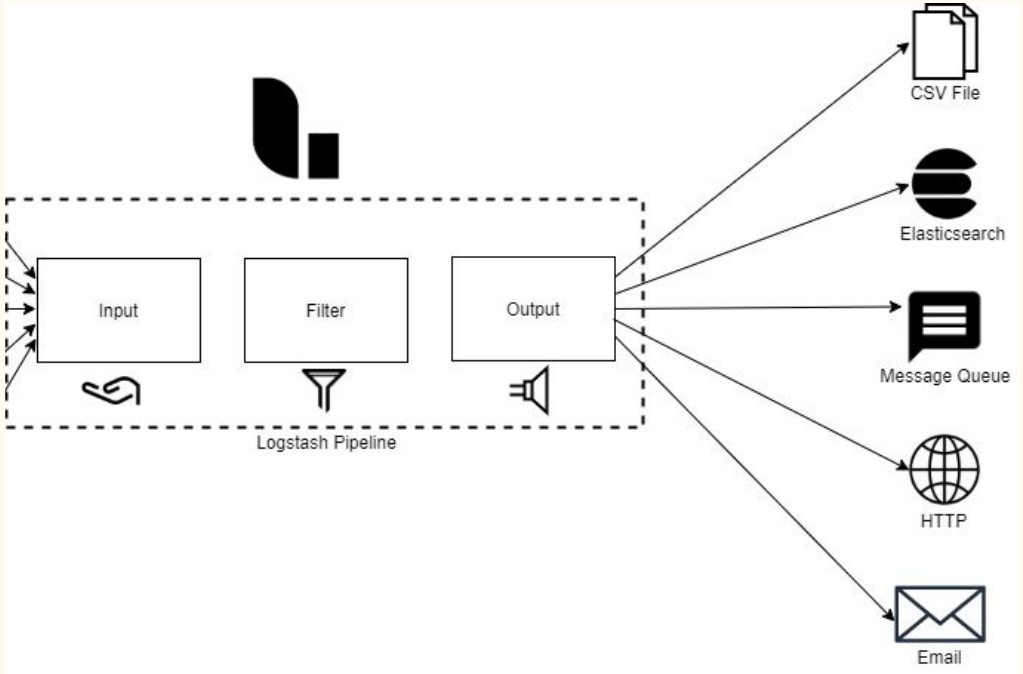
Output Plugins

Function:

- Destination directives for processed data.
- Multiple destinations possible.

Examples:

- Elasticsearch for visualization.
- AWS S3 for storage.
- Email for alerts.
- Image: Icons of various output destinations.



Input plugin example

- **file plugin:** Reads data from a file.
- **path:** Specifies the path to the file.
- **start_position:** Tells Logstash to start reading from the beginning of the file.
- **ignore_older:** Ensures that even older log files are processed from the beginning.

```
input {
file {
path => "/path/to/logfile.log"
start_position => "beginning"
ignore_older => 0
}
}
```

Filter plugin example

- **grok filter:** Parses unstructured log data into structured data.
- **match:** Uses the COMBINEDAPACHELOG pattern to parse common fields found in combined Apache logs (like client IP, user identifier, HTTP request, etc.).
- **date filter:** Converts and stores timestamps from the log data to ensure they're recognized by Elasticsearch in the correct format.
- **geoip filter:** Enriches data by converting IP addresses into geographical coordinates (latitude, longitude, location, etc.). This can be useful for visualizing log data on a map in Kibana.

```
filter {
grok { match => { "message" => "%{COMBINEDAPACHELOG}" } }
date { match => [ "timestamp", "dd/MMM/yyyy:HH:mm:ss Z" ]
remove_field => [ "timestamp" ] }
geoip { source => "clientip" }
}
```

Output plugin example

- **elasticsearch plugin:** Sends the processed data to an Elasticsearch instance.
- **hosts:** Specifies the Elasticsearch instance's address.
- **index:** Defines the naming convention for indices. Here, it creates an index for each day with a name like weblogs-2023.10.12.

```
output {
elasticsearch {
hosts => ["http://localhost:9200"]
index => "weblogs-%{+YYYY.MM.dd}"
}
}
```

Input & Output

192.168.1.1 - - [12/Oct/2023:14:23:45 +0000] "GET /index.html HTTP/1.1" 200 1234 "http://referrer.example.com" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"

Input


```
{
"@timestamp": "2023-10-12T14:23:45.000Z",
"clientip": "192.168.1.1",
"verb": "GET",
"request": "/index.html",
"httpversion": "1.1",
"response": 200,
"bytes": 1234,
"referrer": "http://referrer.example.com",
"agent": "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)",
"geoip": {
"ip": "192.168.1.1",
"latitude": XX.XXXX,
"longitude": XX.XXXX,
"location": [ XX.XXXX, XX.XXXX ],
"city_name": "City Name",
"region_name": "Region Name",
"country_name": "Country Name"
//... other geoip fields
}
}
```


Output

Advantages of Logstash

Scalability:

- Capability to run multiple pipelines in a single instance.
- Horizontal scalability for vast data.
- Dynamic pipelines with conditional statements.

Flexibility:

- User-friendly markup.
- Dynamic conditional operations to suit varying data needs.

Logstash in Action

Scenario:

- Tackling web server access logs.

Procedure:

- Read logs via the “file” input plugin.
- Extract data with Grok patterns.
- Segregate data into fields for detailed insights.

Beyond Just Logs

- Versatility in handling different data types beyond just logs.
- Potential for diverse applications and insights.

PART 04

Elastic stack
Beats

Introduction to Beats

- **Definition:** Beats are lightweight agents installed on servers.
- **Primary role:** To send data to Logstash or Elasticsearch.
- **Designed for specific purposes:** Each Beat targets a unique kind of data.
- **Key advantage:** Efficient data collection without the overhead of larger agents.

Filebeat

- Collects and ships log files.
- Targets specific logs: nginx, Apache, MySQL, and more.
- Useful for access logs, error logs, etc.

Filebeat - The Log Collector

- **Core Function:** Harvests log files and ships them.
- **Supported Logs:** Integrates seamlessly with nginx, Apache, MySQL, and more.
- **Practical Uses:** Centralizing access logs, error logs, and other operational logs.
- **Pre-packaged Modules:** Makes integration with common systems more effortless.

Exploring Metricbeat

- **System Metrics:** Monitors CPU, memory usage, and more.
- **Service Metrics:** Gathers data on the performance of running services.
- **Integration with Popular Services:** Modules for nginx, MySQL, and other popular services.
- **Result:** A comprehensive view of system and service health in near real-time.

Exploring Metricbeat

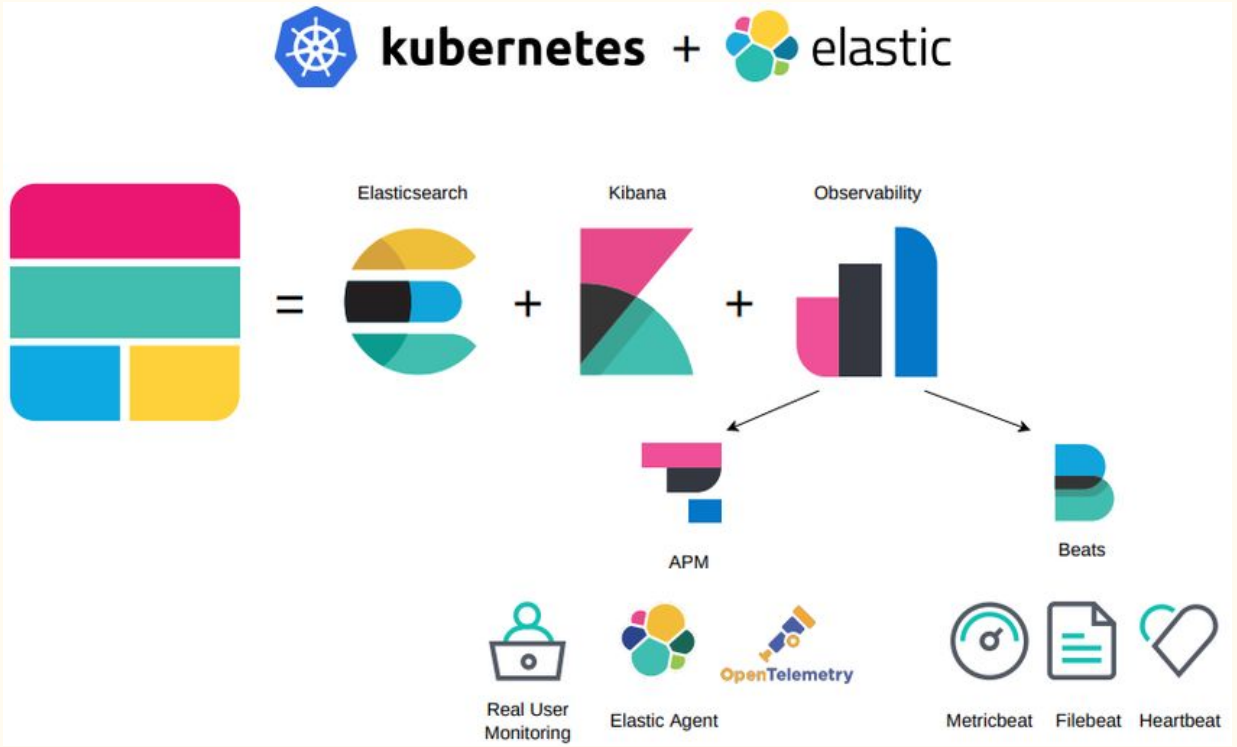
- **A Plethora of Beats:** Diverse tools for unique data collection needs.
- **Some examples:** Auditbeat (for audit data), Heartbeat (uptime monitoring), Packetbeat (network data), etc.
- Filebeat and Metricbeat remain the most popular and widely adopted.
- **Recommendation:** Explore the documentation to find the perfect Beat for specific use cases.

Conclusion and Takeaways

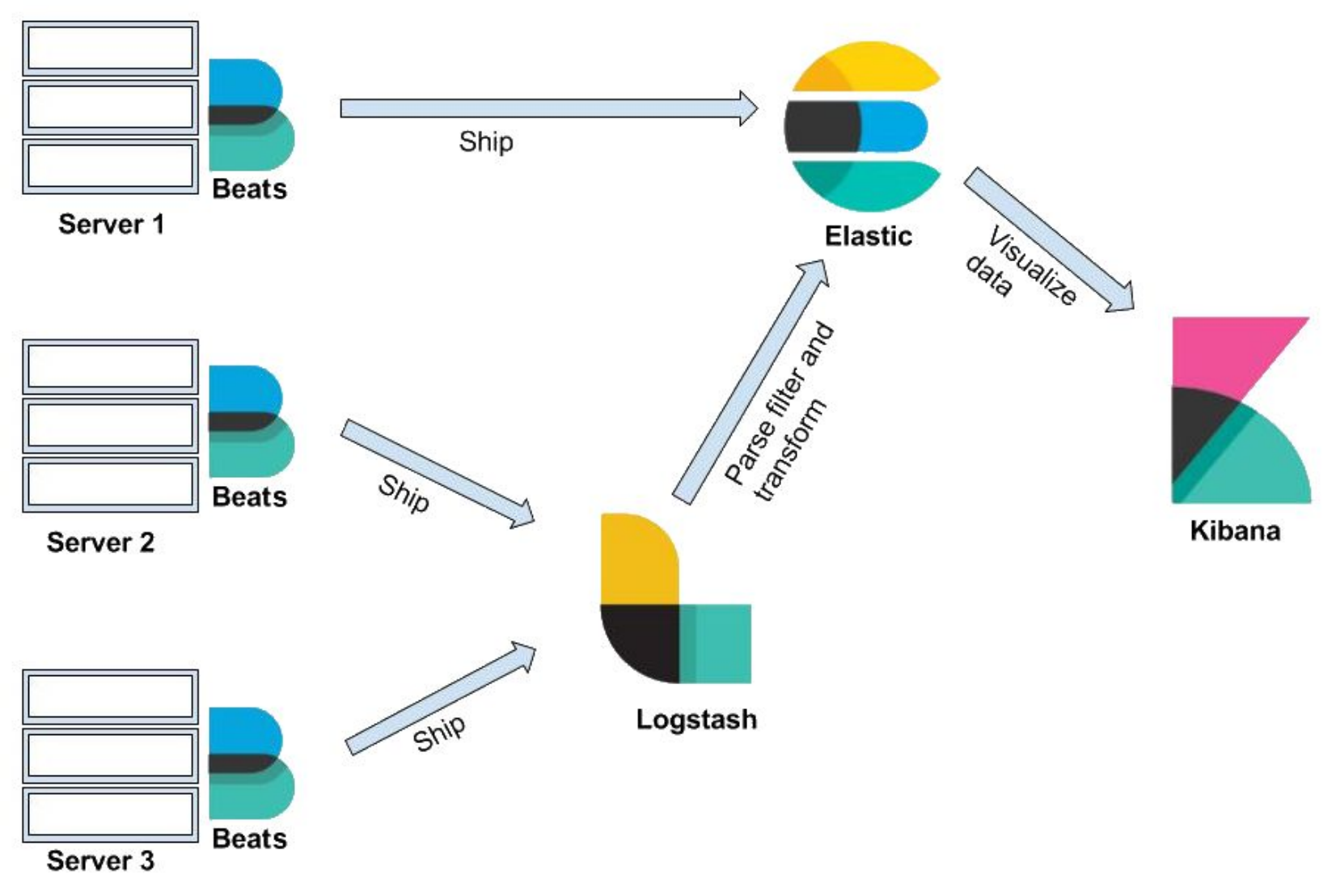
- **Simplified Data Collection:** Beats offer a streamlined way to gather data.
- **Scalability:** Lightweight nature ensures minimal impact on system resources.
- **Flexibility:** With various Beats available, there's likely one tailored for your specific needs.
- **Position in the Elastic Stack:** Crucial for data ingestion, laying the foundation for analysis and visualization in Kibana.

More elastic search stack

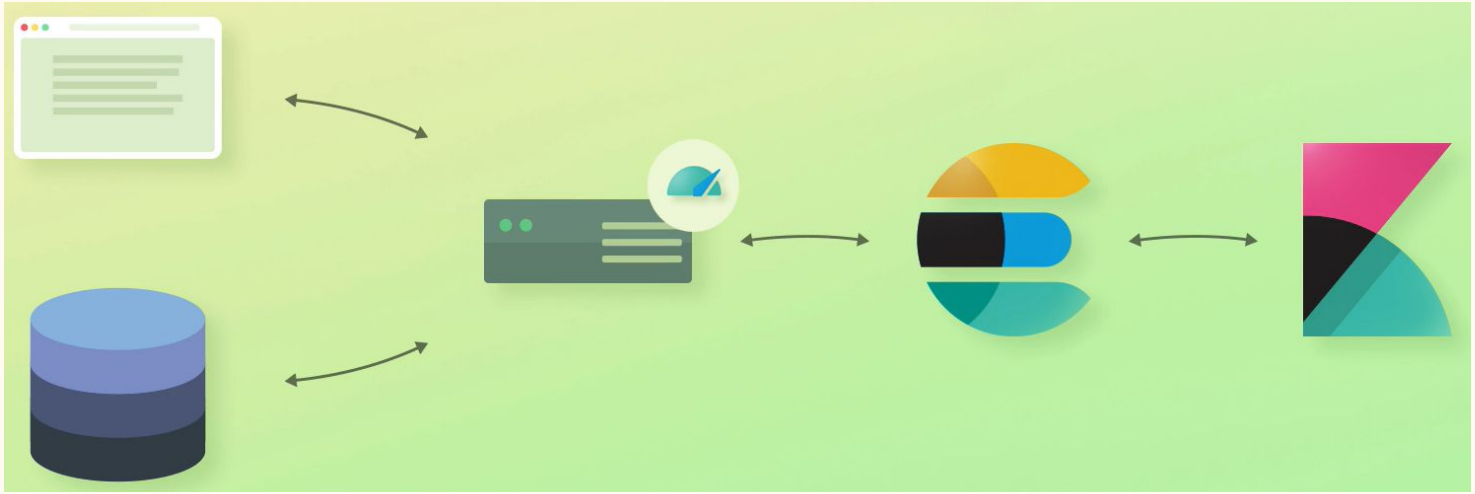
- x-pack
- Graph
- Elasticsearch SQL

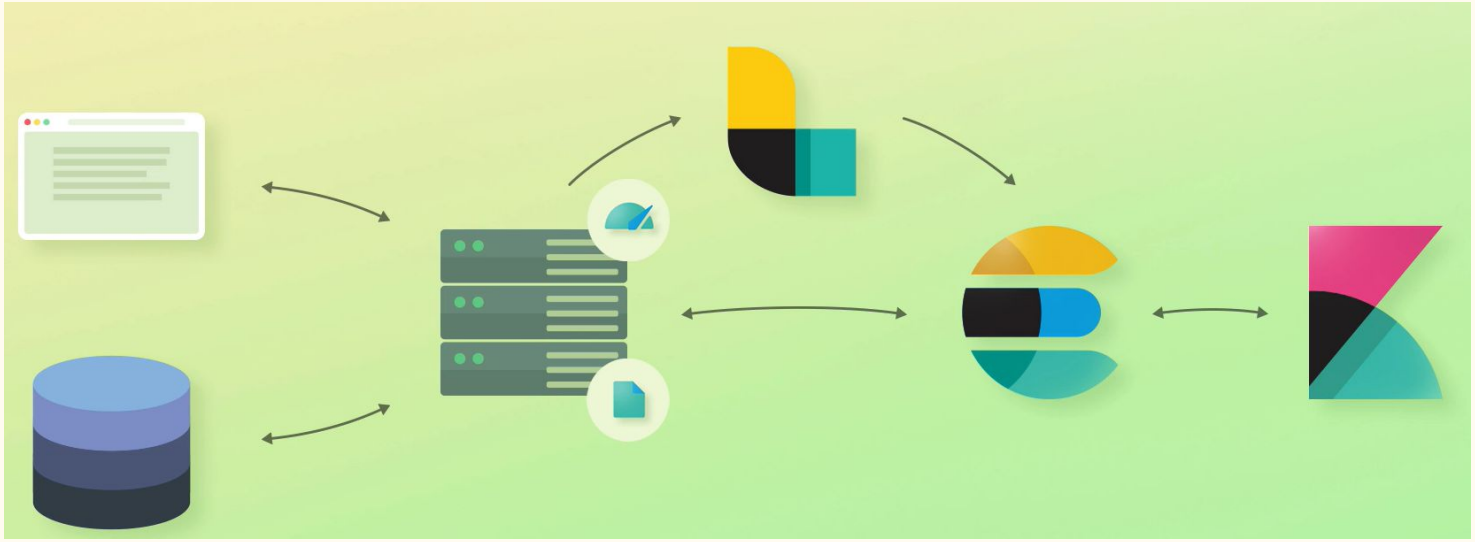


Synthesizing



E-commerce example





Amazon Project Nessie

Elastic stack setup

- Setup in windows
- Setup in iOS
- Use the cloud free trial solution
- Configure your own server

Elastic stack checking the server health

- GET /_cluster/health
- GET /_cat/nodes?v
- GET /_cat/indices?v&expand_wildcards=all

Elastic stack checking the server health

- GET /_cluster/health
- GET /_cat/nodes?v
- GET /_cat/indices?v&expand_wildcards=all