



ELK STACK

ELK Stack:

is a combination of 3 open-source tools for log analysis.



ELASTICSEARCH



- *An Apache Lucene based search engine*
- *It is open source & developed using Java*



LOGSTASH



- *Tool for collecting & monitoring logs from remote machines*
- *Is a data pipeline for Elasticsearch*



KIBANA



- *Data exploration & visualization tool*
- *Used for log & time series analytics, application monitoring & operational intelligence*



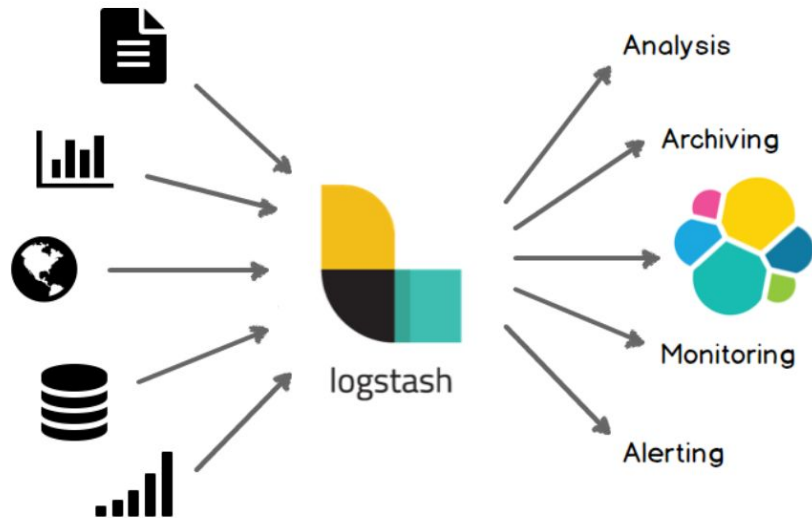
How Do They Work Together?



Logstash

Logstash is an open source data collection engine with real-time pipelining capabilities. Logstash can dynamically unify data from disparate sources and normalize the data into destinations of your choice.

While Logstash originally drove innovation in log collection, its capabilities extend well beyond that use case. Any type of event can be enriched and transformed with a broad array of input, filter, and output plugins, with many native codecs further simplifying the ingestion process.





How Logstash Works

The Logstash event processing pipeline has three stages: inputs → filters → outputs. Inputs generate events, filters modify them, and outputs ship them elsewhere. Inputs and outputs support codecs that enable you to encode or decode the data as it enters or exits the pipeline without having to use a separate filter.

Input/Filter/Output plugins

<https://www.elastic.co/guide/en/logstash/current/input-plugins.html>
[/filter-plugins.html](#)
[/output-plugins.html](#)

```
input {
  file {
    path => "/tmp/access_log"
    start_position => "beginning"
  }
}

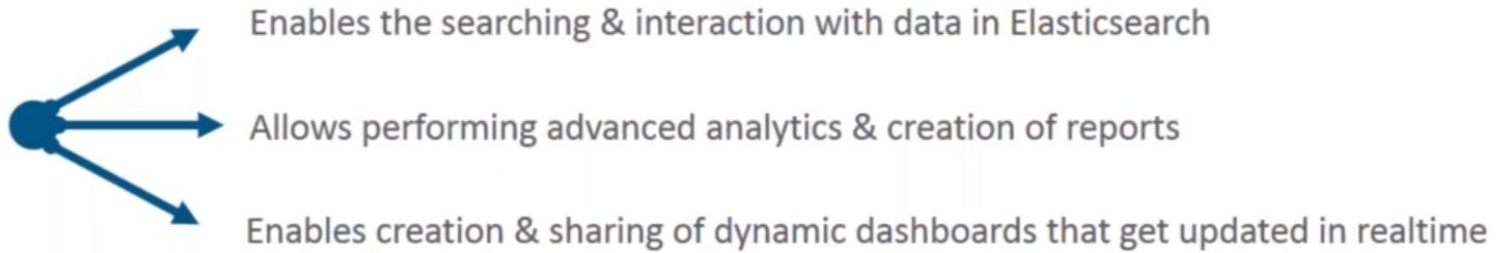
filter {
  if [path] =~ "access" {
    mutate { replace => { "type" => "apache_access" } }
    grok {
      match => { "message" => "%{COMBINEDAPACHELOG}" }
    }
  }
  date {
    match => [ "timestamp" , "dd/MMM/yyyy:HH:mm:ss Z" ]
  }
}

output {
  elasticsearch {
    hosts => ["localhost:9200"]
  }
  stdout { codec => rubydebug }
}
```

Role of KIBANA in ELK



KIBANA

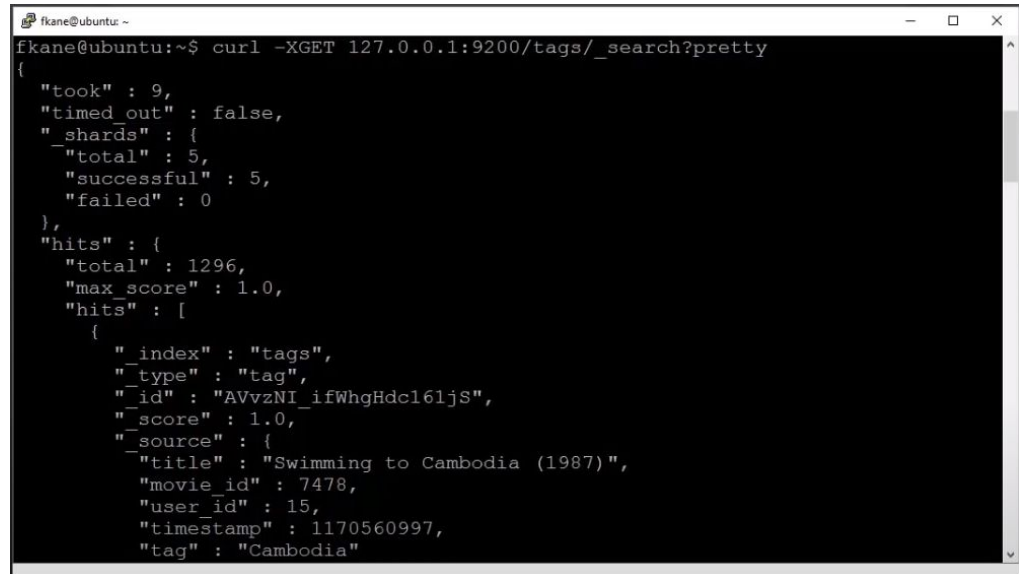




What is Elasticsearch

Elasticsearch is a distributed, open source search and analytics engine for all types of data, whether textual, numerical, geographic, structured and unstructured. Works with Restful services. It is easy to scale due to its architecture.

In simple terms, it is a database that stores and manages document-based and semi-structured data.



```
fkane@ubuntu: ~  
fkane@ubuntu:~$ curl -XGET 127.0.0.1:9200/tags/_search?pretty  
{  
  "took" : 9,  
  "timed_out" : false,  
  "_shards" : {  
    "total" : 5,  
    "successful" : 5,  
    "failed" : 0  
  },  
  "hits" : {  
    "total" : 1296,  
    "max_score" : 1.0,  
    "hits" : [  
      {  
        "index" : "tags",  
        "_type" : "tag",  
        "_id" : "AVvzNI_ifWhgHdc161jS",  
        "_score" : 1.0,  
        "_source" : {  
          "title" : "Swimming to Cambodia (1987)",  
          "movie_id" : 7478,  
          "user_id" : 15,  
          "timestamp" : 1170560997,  
          "tag" : "Cambodia"  
        }  
      }  
    ]  
  }  
}
```



Why Elasticsearch is used

Products that include e-commerce and search engines with large databases face problems, including the retrieval of product information that takes too long. This leads to poor user experience and therefore closes potential customers.

Relational database runs relatively slow when it comes to very large data and fetching search results via database queries

If we need to search for text between large blocks of data, then Elasticsearch may be the right choice for us. It is usually a faster solution than Hadoop / Spark / Flink etc.



Elasticsearch is used for?

- Application search
- Website search
- Logging and log analytics
- Geospatial data analysis and visualization
- Business analytics
- Much more....

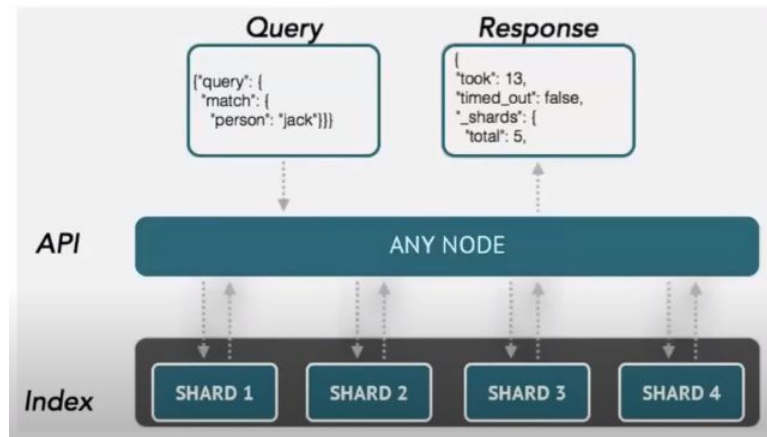
How Elasticsearch works

Raw data flows into Elasticsearch from variety of sources, including logs, system metrics and web applications.

Data ingestion is the process by which this raw data is parsed, normalized, and enriched before it is indexed in Elasticsearch.

Once indexed in Elasticsearch, users can run complex queries against their data and use aggregations to retrieve complex summaries of their data.

From Kibana, users can create powerful visualizations of their data, share dashboards, and manage the Elastic Stack.



Logical Concepts of Elasticsearch



documents

Documents are the things you're searching for. They can be more than text – any structured JSON data works. Every document has a unique ID, and a type.



types

A type defines the schema and mapping shared by documents that represent the same sort of thing. (A log entry, an encyclopedia article, etc.)



indices

An index powers search into all documents within a collection of types. They contain inverted indices that let you search across everything within them at once.

***Elasticsearch is moving toward eliminating the concept of types. In Elasticsearch 6, only one type is allowed per index.**



What is an inverted index

Document 1:

Space: The final frontier. These are the voyages...

Document 2:

He's bad, he's number one. He's the space cowboy with the laser gun!

Inverted index

space:	1, 2
the:	1, 2
final:	1
frontier:	1
he:	2
bad:	2

...



Basic Concepts of Elasticsearch

Cluster : is a collection of one or more servers that together hold entire data and give federated indexing and search capabilities across all servers. For relational databases, the node is DB instance. There can be N nodes with the same cluster name.

Node: A node is a single server that holds some data and participates on the cluster's indexing and querying. A node can be configured to join a specific cluster by the particular cluster name. A single cluster can have as many nodes as we want. A node is simply one Elasticsearch instance.

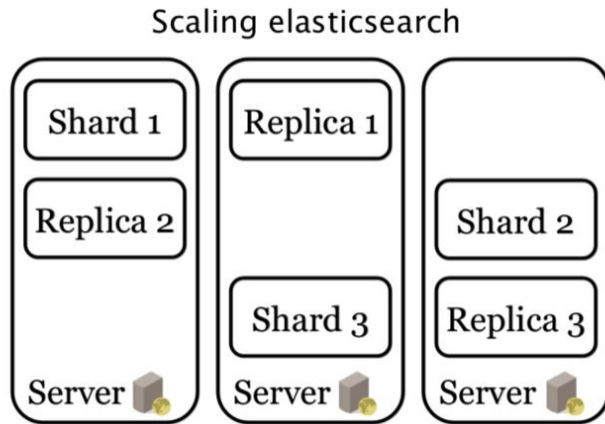
.

Basic Concepts of Elasticsearch(Cont.)

Index: The index is a collection of documents that have similar characteristics.

Shard: A shard is a subset of documents of an index. An index can be divided into many shards.

Replica: There is a replica-shard structure that allows one or more copies of index shards to be created in case the shard becomes disabled



How Elasticsearch scales

An index is split into shards.

Documents are hashed to a particular shard.

Each shard may be on a different node in a cluster.

Every shard is a self-contained Lucene index of its own.





Elasticsearch REST APIs

Index(PUT/POST) API: It helps to add or update the JSON document in an index when a request is made to that respective index with specific mapping

GET API: It helps to extract type JSON object by performing a get request for a particular document.

DELETE API: You can delete a particular index, mapping or a document by sending a HTTP DELETE request to Elasticsearch.

UPDATE API: Update the docs/data put into elasticsearch.

BULK API: If you have a lot of documents to index, you can submit them in batches with the bulk API. Using bulk to batch document operations is significantly faster than submitting requests individually as it minimizes network roundtrips.