Published in Geek Culture

JIN Follow

Dec 23, 2022 · 11 min read · · Listen

Save

Photo by Marten Newhall on Unsplash

Elasticsearch Architecture

It is a distributed search engine based on the Lucene library. It supports quasi-real-time data retrieval NRT (near real-time), processing structured and unstructured data, and providing

multitenant-capable full-text search engine capabilities with an HTTP web interface and schema-free JSON documents.

Please support me if you feel that I contribute value to you!

If you feel my articles are valuable to you, please become my referred members to support me. It can bring some income for me.

What is Elasticsearch

- It is a **real-time distributed storage, searches, and analysis engine**

1. Real-time
2. Distributed storage
3. Search
4. Analysis

- Most of the data searched from Elasticsearch can be filtered out according to the score as long as the high score is returned to the user
- Hence, relevant results can be found even with less accurate keywords

The Concept

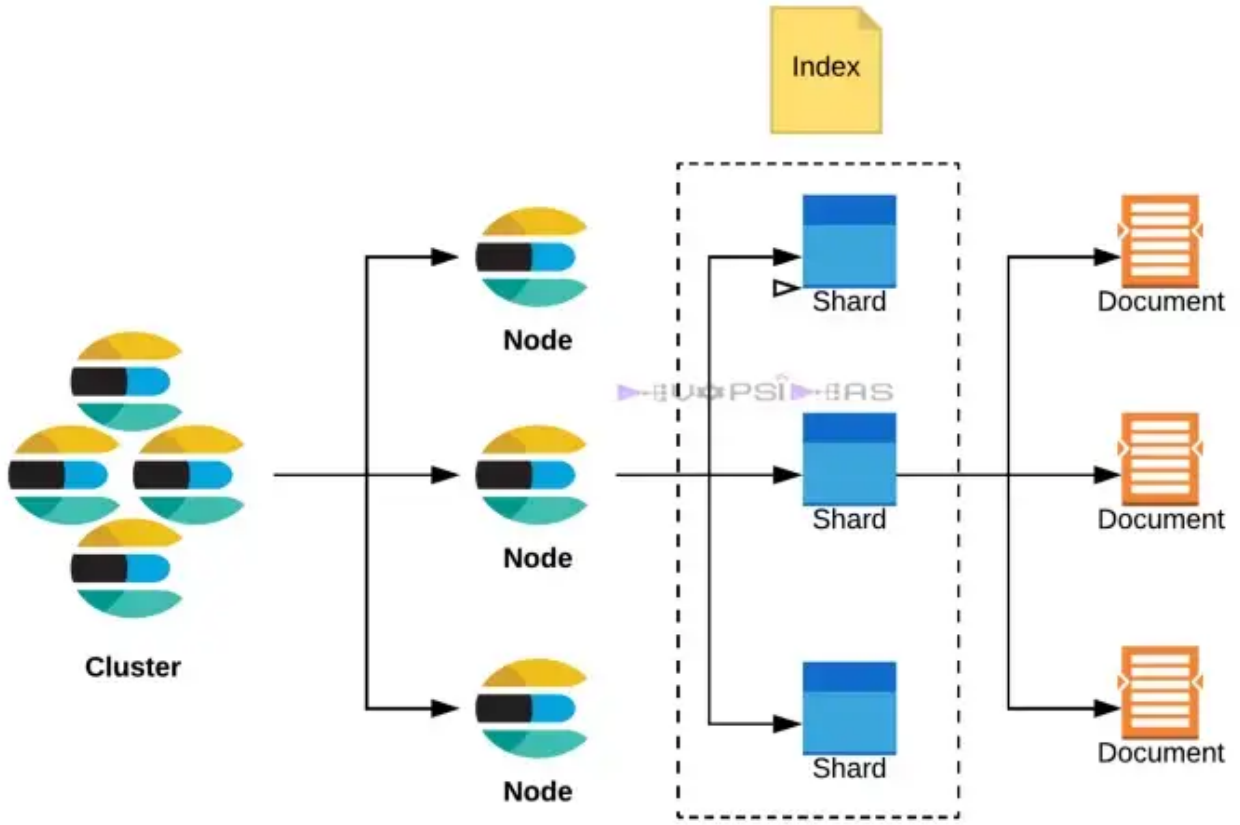


Image Credit: **Devopsideas**

1. **Near real-time (NRT)**

- It is a slight delay (typically 1 second) from indexing a document until it is searchable

2. **Node**

- It is an instance of a server that can run independently.
- It is identified by a unique name. It determines which servers in the network correspond to which nodes in the Elasticsearch cluster
- The unique name is important when you install multiple nodes in the same server although it is not recommended.
- A node is also a service unit that makes up a cluster.
- Its function is to store data and have indexing and searching capabilities.

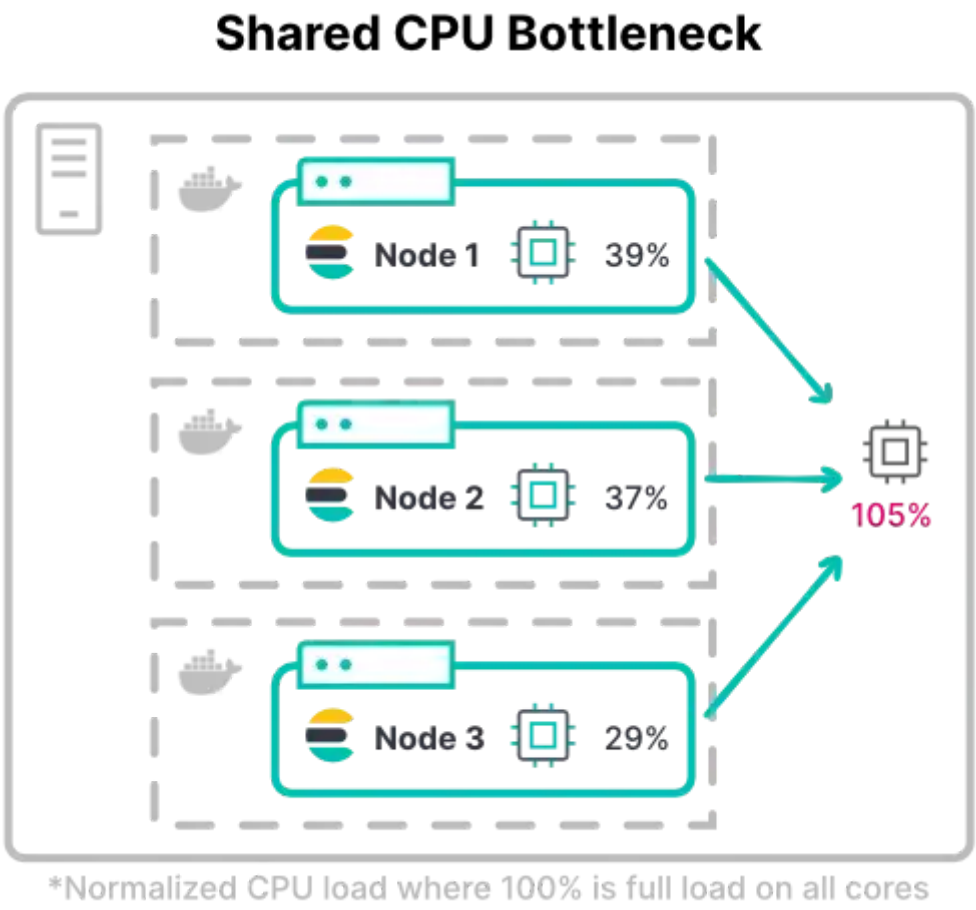


Image Credit: **Ryan Eno**

3. **Cluster**

- It is organized by one or more nodes, which jointly hold your entire data and provide indexing and search capabilities.
- One of the nodes is the master node, which is elected through elections
- A cluster is identified by a unique name. A node can only join a cluster by specifying its name.
- Its function is to provide collection indexing and search capabilities across all nodes for the entire data.

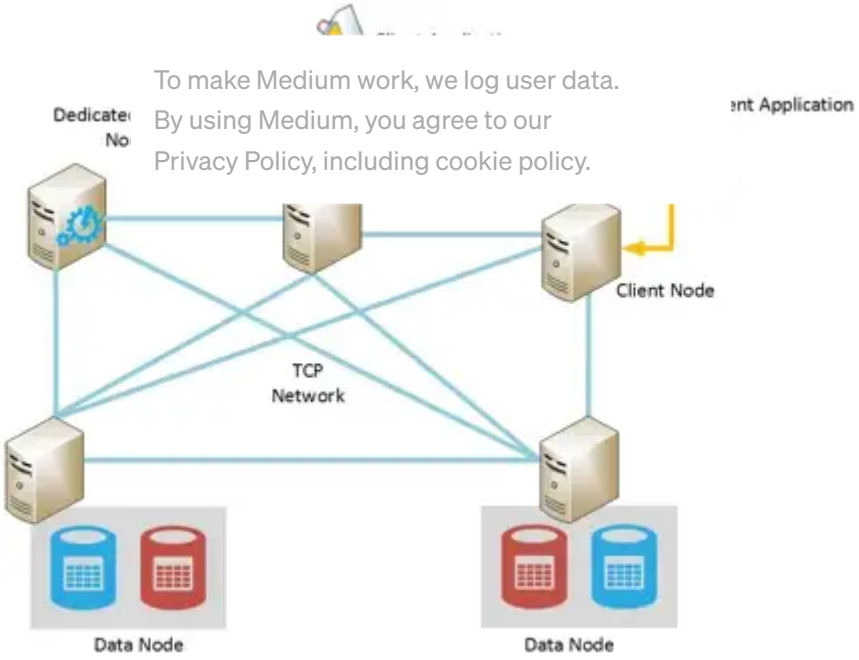


Image Credit: **K Hong**

The Fundamental Knowledge of System Design (13) — The Raft Consensus Algorithm

It's equivalent to Paxos in fault tolerance and performance.

medium.com

4. Index/Shards

- It is a collection of different types of documents with somewhat similar characteristics.
- It is identified by a name (must be all lowercase letters)
- It uses the fragmentation mechanism to improve retrieval performance.

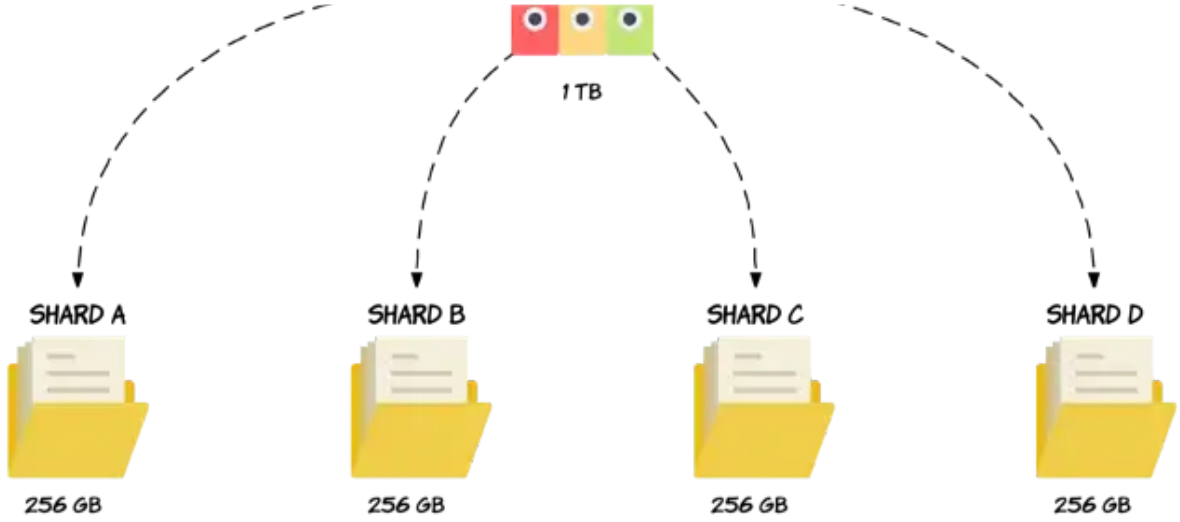


Image Credit: **Bo Andersen**

5. Type

- It is a logical classification/partition of the index
- Types are similar to the concept of tables in relational databases.
- It is used to classify documents

6. Document

- It is a basic information unit that can be indexed.
- It is defined in **JSON (Javascript Object Notation)** format.
- Each document has a unique identifier. Each document must be assigned an index type.
- Documents are similar to the concept of records in relational databases.

7. Mapping

- It is a way to define the relationship between a document and its fields.
- It contains information such as metadata fields, field lists, or attributes.

8. Fragmentation (Shards ->Lucene Index)

Image Credit: **Lakshya Bansal**

- Shard = Lucene Index
- An index can store a large amount of data beyond the hardware limit of a single node. However no node can process such large search requests, thereby the response is slow.
- To solve this problem, it is mainly used to split index data horizontally into multiple shards to reduce the pressure on individual nodes and improve query performance through parallel retrieval.
- The number of shards can only be specified before the index is created, and cannot be changed after the index is created.
- Each shard is itself a fully functional and independent “index” that can be placed on any node in the cluster.

9. Replica

- The number of replica fragments (≥0) must be specified when defining the index.
- The number of replicas can be changed dynamically

- In a network/cloud production environment, failures can happen anytime, where a shard/node is offline for some reason. So, a failover mechanism is recommended where copies of a shard. These copies are called replicas.

- Replicas are never placed on the same node as the primary shard
- When a node is damaged, it can be recovered from the replica.
- All read requests can run on all replicas in parallel to scale up the throughput

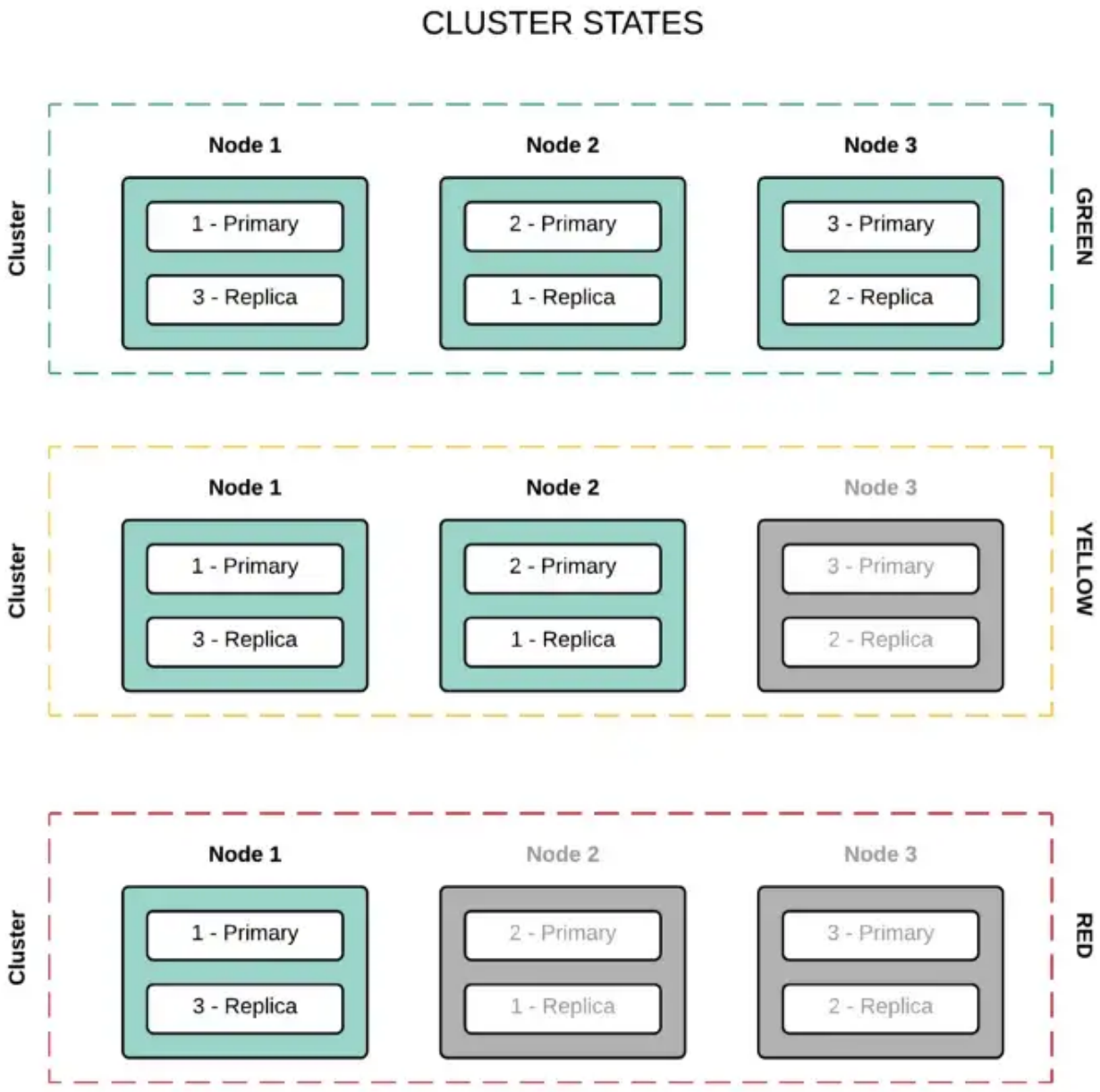


Image Credit: **Devopsideas**

10. Gateway

- It is the persistent s lex. It stores the index data in memory firs when the memory is full. It supports mu ed file system (Hadoop), and Amazon's S3 cloud storage system.

The source code:

GitHub — elastic/Elasticsearch: Free and Open, Distributed, RESTful Search Engine

Free and Open, Distributed, RESTful Search Engine. Contribute to elastic/Elasticsearch development by creating an...

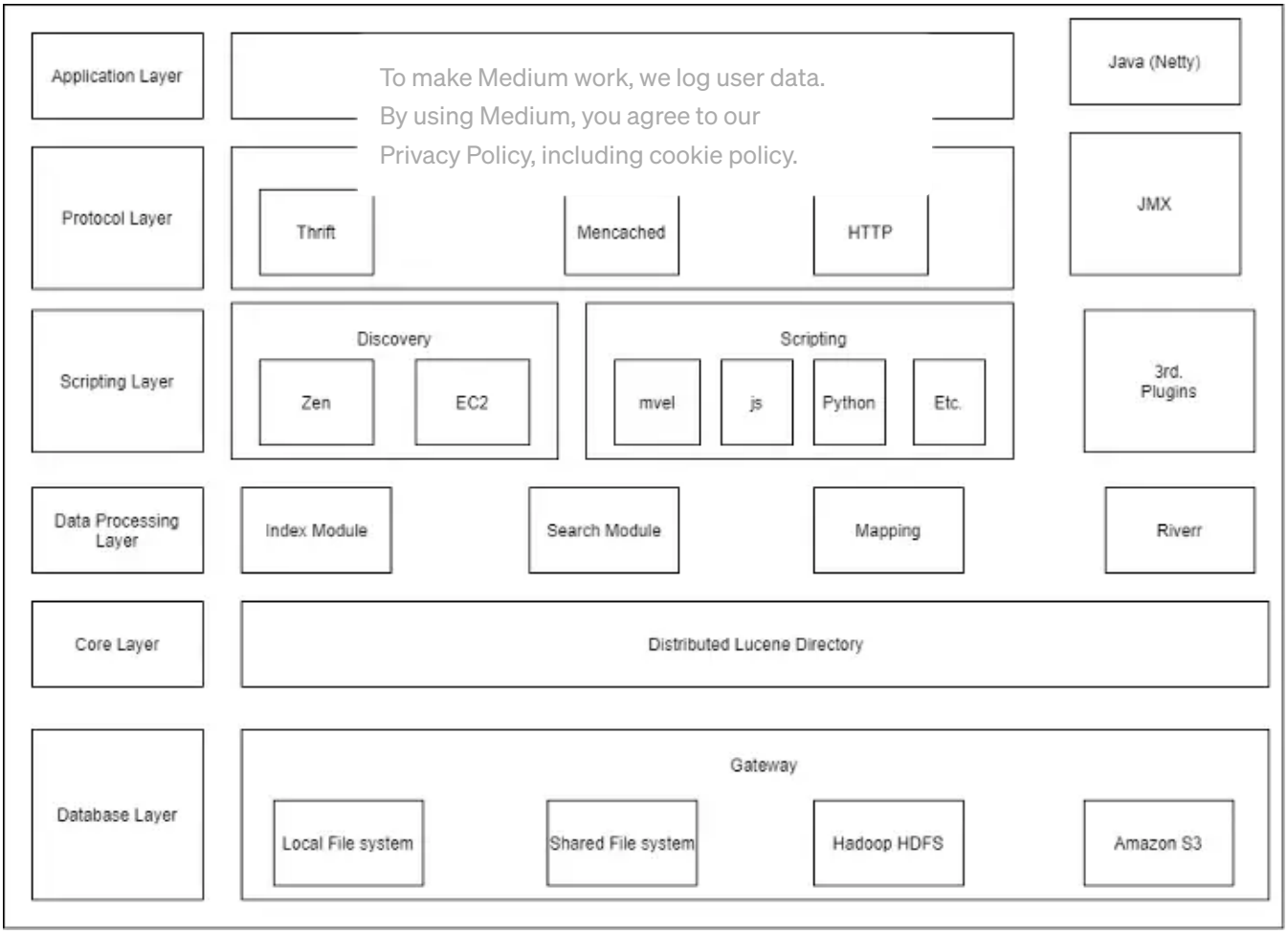
github.com

It is recommended that you understand the source code before understanding the architecture, which can better help you understand the overall architecture diagram.

Elasticsearch Functions

1. Full-text search engine
2. Solve complex queries
3. Solve complex calculation
4. Solve other operations that cannot be solved by traditional databases
5. Solve time-series data processing such as log process, infrastructure monitoring

Overall Architecture



Architecture Hierarchy

1. Application layer
2. Protocol layer
3. Discovery/scripting layer
4. Data processing layer
5. Core architecture layer
6. Data storage layer

Application Layer

1. **Restful APIs**

- The interface is pro [boilerplate: To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.] can interact with the Elasticsearch cluste PI.
- It includes CURD operations, index creation, and index deletion
- Java APIs: `TransportClient` (7. x obsolete, 8.0 removed, difficult to have a compatible version). So, `TransportClient` is used for old versions of Elasticsearch
- New Versions is `Java High Level REST Client`
- Elasticsearch provides various query interface

The Introduction to RESTful API

RESTful (Representational State Transfer) is a software architectural style that defines a set of constraints for the...

medium.com

Protocol Layer

1. **Thrift**

- It supports a variety of different programming languages, including C++, Java, Python, PHP, Ruby, etc.
- It allows users to select the type of transmission protocol between the client and the server, either text or binary type transmission protocols.
- In order to save bandwidth and improve transmission efficiency, the binary-type transmission protocol is mostly used.
- The text-type protocol is used based on the actual requirements of the project.
- It is also an RPC framework for large-scale cross-language service development.

2. Memcached

- It is a free open-sou emory object storage caching system whi ons and reduces database load

- It is also a key-value cache
- It is a text-line-based protocol
- Data persistence is not supported because the data is queried frequently, thereby leading to more reads and fewer writes. It is suitable for e-commerce scenarios. All data is lost after the server is shut down. So, it is very convenient for rapid development (frequent queries) and easy to use.

3. HTTP

- The API is provided externally in the form of the HTTP protocol and RESTful protocol in JSON format.

The fundamental knowledge of System Design — (1)
Today, I will share the fundamental knowledge of system design.
interviewnoodle.com

4. TCP

- The underlying communication protocol is **Netty** (a high-performance asynchronous I/O framework).
- Elasticsearch inserts the implementation of Netty into its own system in the form of plug-ins
- The bottom layer uses **kqueue** or **epoll** to achieve high multiplexing of I/O and uses **zero-copy buffer** technology to improve the efficiency of the CPU.

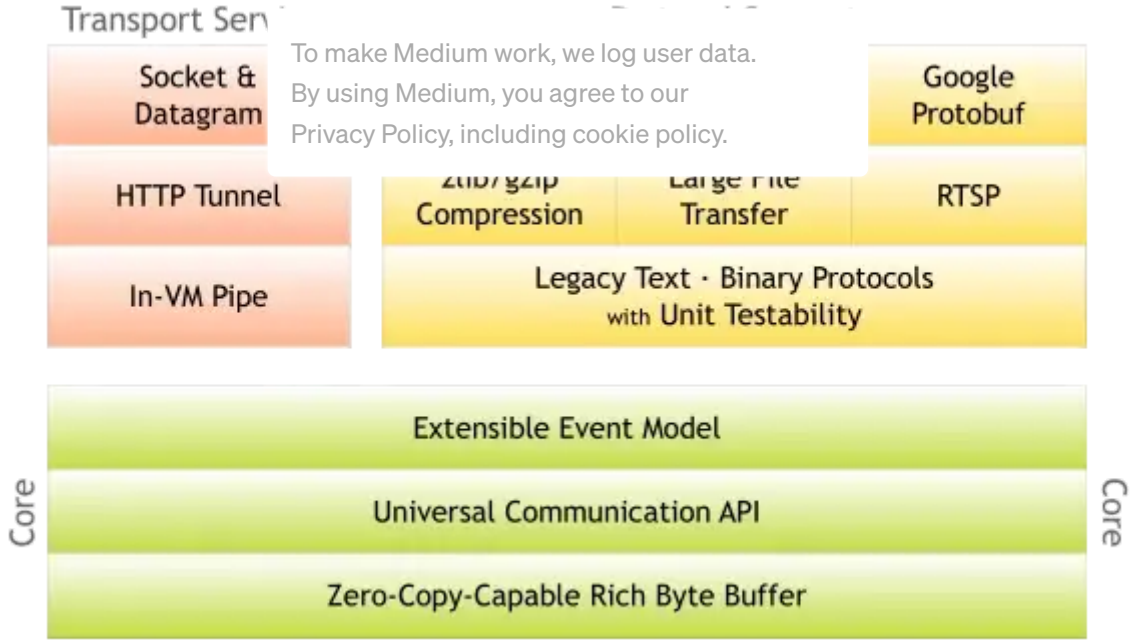


Image Credit: **Netty**

5. Java Management Extensions (JMX)

- It is a function introduced in J2SE 5.0 version.
- It is a service framework for embedding management functions for applications, devices, systems, etc. it can dynamically manage resources at runtime and dynamically obtain application status.
- It can flexibly develop seamlessly integrated system, network, and service management applications across a series of heterogeneous operating system platforms, system architecture, and network transmission protocols.
- It is divided into 3 layers (remote management layer, agent layer, and probe layer).

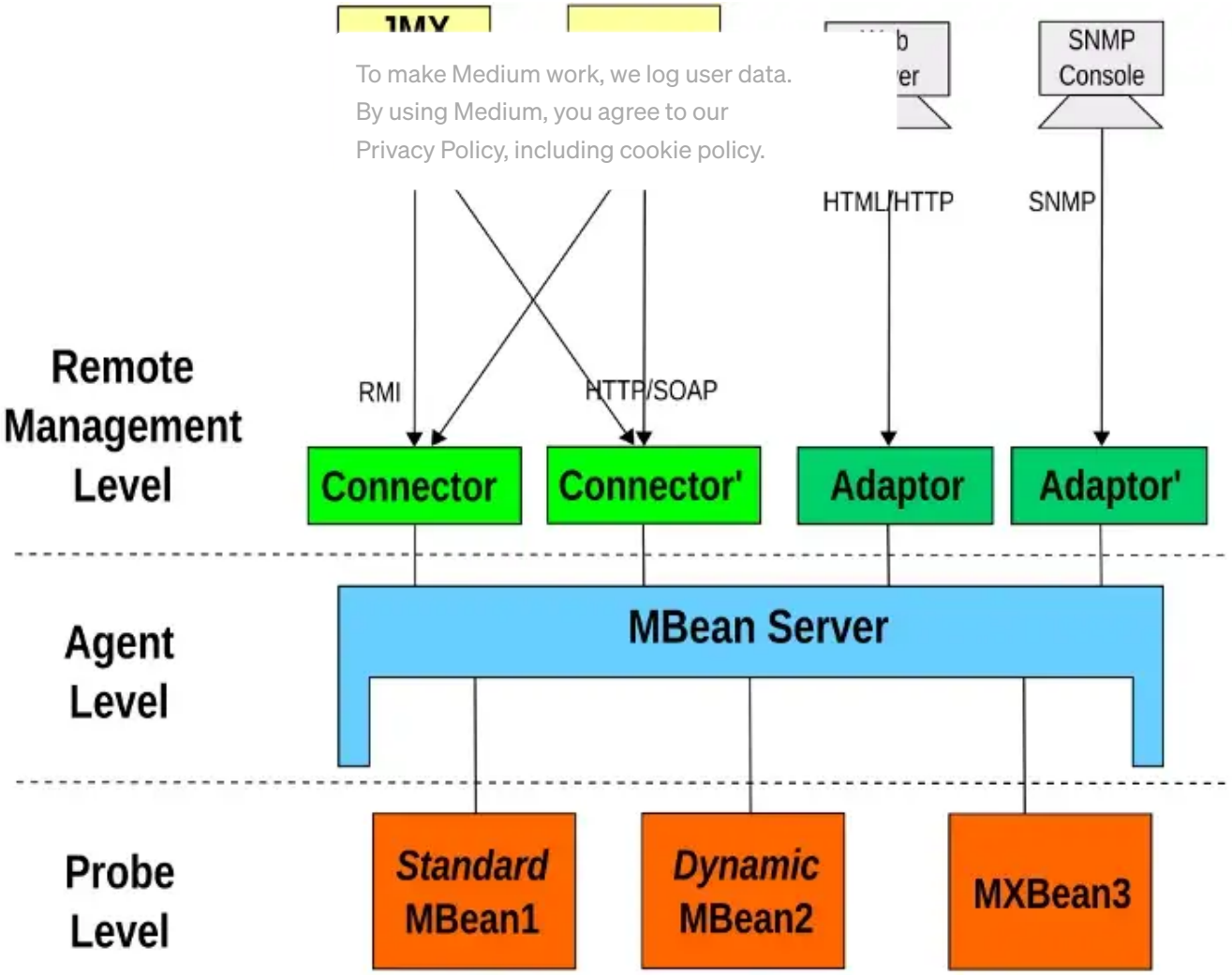


Image Credit: **Wikipedia**

Discovery/Scripting Layer

1. **ZenDiscovery**

- It is a special built-in discovery mechanism of Elasticsearch.
- It provides 2 discovery methods — Unicast and multicast.
- The main function is to discover the nodes in the cluster and elect the master node
- **Unicast** — A node sends a request to a server when joining an existing cluster, or forming a new cluster. When a node communicates via unicast, it will get the status of all nodes in the entire cluster, communicate with the master node, and join the cluster.

- **Multicast** — A node can send requests to multiple servers. However, it is not recommended for a [boilerplate] because it can generate a lot of unnecessary

2. Scripting

- It is used to solve complex business scenarios.
- However, it is rarely used in an actual production environment because it can lead to low performance.
- In non-complex business scenarios, basic operations can be done without scripting.

Data Processing layer

1. **Index Module**

- It is a module created by an index that controls all aspects related to the index.
- It can be divided into 2 parts: Static and dynamic
- Static — It is specified at creation time and cannot be modified
- Dynamic — It can be modified by updating the index configuration API

2. Search Module

- It allows users to perform a search query and return query results.
- The search is performed segment by segment because an index is composed of several segments. With the continuous growth of each segment, the delay might be lengthened due to the bottleneck point inside the disk.
- **fsync operation** — ensure that the segment can be physically written to the disk to truly avoid data loss, but it is time-consuming, so it cannot be performed once every time each piece of data is indexed. Indexing and searching for particular data leads to a large delay. So, the newly added data is stored in the indexing

buffer area and then rewritten into a segment, and directly written into the filesystem cache.

- As long as data of th ound in a short time without performing a full commit or fsync operation.

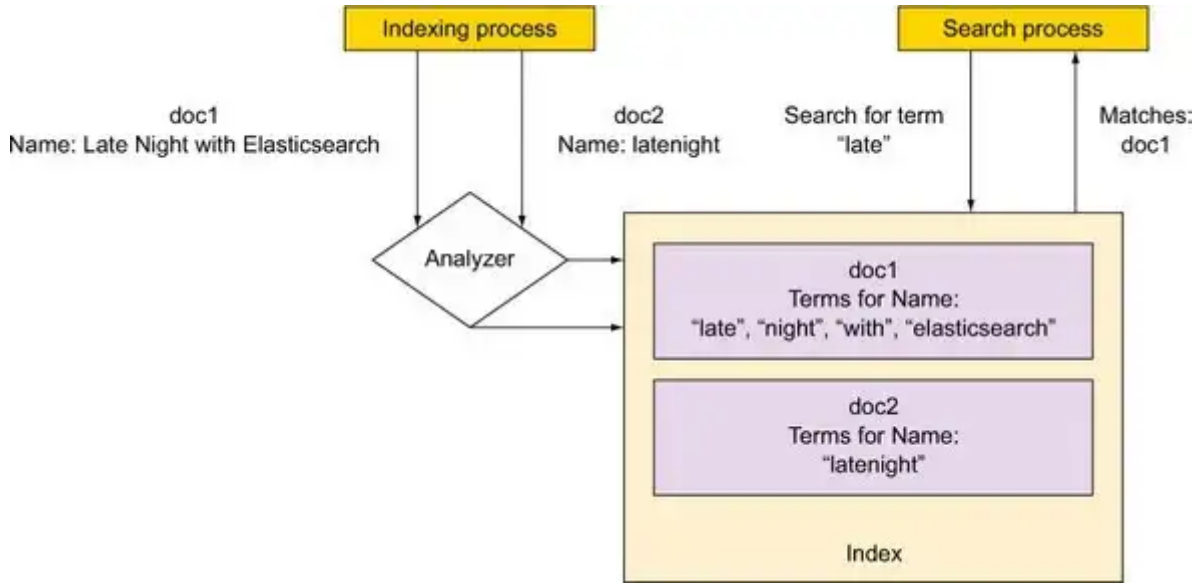


Image Credit: **Livebook**

Index Process

- When creating an index, a document is first stored in the main shard through routing rules.
- After that, the replica is also created.
- All index data is located in shards. The primary shard and replica shard store one copy. When the data of a replica shard or the primary shard is lost, it can be recovered in other nodes. This process is time-consuming. Before the recovery is completed, the entire system will be in a relatively dangerous state until the failover ends.
- The replica is also used to improve data reliability, that is, when there are a large number of read query requests, all read query requests can run on the replica in parallel.

Search Process

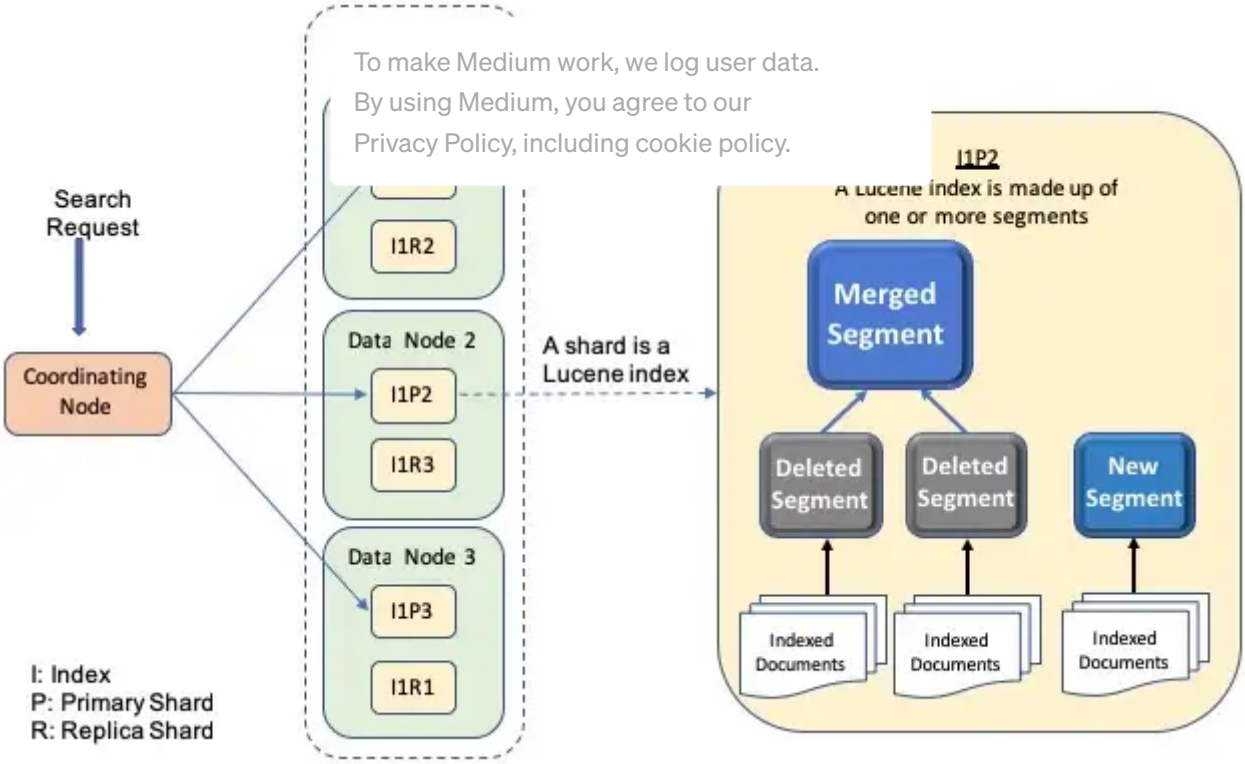


Image Credit: **Packtpub**

- When a search request is sent to a node, that node becomes the coordinator node.
- The coordinator node broadcast query requests to all relevant shards and aggregates their responses into a globally sorted set, which is returned to the client.

3. Mapping

- It is used to define how a document is mapped to the search module, including its searchable characteristics.
- It defines how to categorize the documents in an index into logical groups.

4. River

- It is used to obtain data from other data sources.
- It exists in the form of plug-ins.
- It includes: RabbitMQ, ActiveMQ, CSV, FileSystem, JDBC, GitHub, Kafka, etc

Core Architecture Laye

1. **Lucene**

- It is an open-source full-text search engine toolkit written based on Java.
- It is the most advanced and powerful search library.
- It takes a lot of learning costs to develop directly based on Lucene.
- The API is also complex because it requires users to have an in-depth understanding of the principles.

Data Storage Layer

1. **Gateways**

- **Local File System** — It allows applications to store and retrieve files on storage devices. Files are arranged in a hierarchical structure.
- **Distributed Hadoop** — It provides high-performance access to data across highly scalable Hadoop clusters.
- **Amazon S3 cloud object Storage** — It is an object storage service offering industry-leading scalability, data availability, security, and performance.

References

What is Apache Lucene that powers Elasticsearch🚀

Lucene is the heart of Elasticsearch, it is an open-source project maintained by Apache Foundation. We don't need to...

lakshyabansal.hashnode.dev

Understanding Sharding in Elasticsearch

Elasticsearch is extreme
architecture. One of the

codingexplained.com

ELK: Elasticsearch - 2020

bogotobogo.com site search: Elastic Search Logstash with Elastic Search Logstash, ElasticSearch, and Kibana 4...

www.bogotobogo.com

running-Elasticsearch-fun-profit

A book about running Elasticsearch Project maintained by Hosted on GitHub Pages - Theme by fdv mattgraham WIP, COVERS...

fdv.github.io

Field data types | Elasticsearch Guide [8.5] | Elastic

Each field has a field data type or field type. This type indicates the kind of data the field contains, such as...

www.elastic.co

Chapter 3. Indexing, updating, and deleting data · Elasticsearch in Action

Using mapping types to define multiple types of documents in the same index · Types of fields you can use in mappings ·...

livebook.manning.com

Elasticsearch Architectural Overview

Clusters, Nodes, Indices, Shards, and Documents

buildingvts.com

Elasticsearch Architect

Elasticsearch is a distributed search engine used for full-text search. In this section, we are going to discuss...

www.javatpoint.com

A Comprehensive Guide to OpenSearch and Elasticsearch™ Architecture

Elasticsearch is a search and analytics engine built with the Apache Lucene search library. It extends the search...

www.instaclustr.com

ElasticSearch Architecture Overview

An Elasticsearch index is a logical namespace to organize your data (like a database). And the data you put on it is a...

solutionhacker.com

Elasticsearch 101: Fundamentals & Core Components

Elasticsearch is currently the most popular way to implement free text search and analytics in applications. It is...

www.velotio.com

Introduction to the Elasticsearch Architecture

This article is an introduction to the physical architecture of Elasticsearch, being how documents are distributed...

codingexplained.com

ElasticSearch architecture with Mysql

Thanks for contributing an answer to Stack Overflow! Please be sure to answer the question. F

stackoverflow.com

An Advanced Elasticsearch Architecture for High-volume Reindexing

This article and much more are now part of my FREE EBOOK Running Elasticsearch for Fun and Profit available on Github...

thoughts.t37.net

If you've found any of my articles helpful or useful then please consider throwing a coffee my way to help support my work or give me patronage😊, by using

Patreon

Ko-fi.com

buymeacoffee

Last but not least, if you are not a Medium Member yet and plan to become one, I kindly ask you to do so using the following link. I will receive a portion of your membership fee at no additional cost to you.

Join Medium with my referral link — JIN

As a Medium member, a portion of your membership fee goes to writers you read, and you get full access to every story...

jinlow.medium.com

Elasticsearch System Architecture Database Search Engines

Distributed Systems

Enjoy the read? Reward the writer.[Beta]

Your tip will go to JIN through a third-party platform of their choice, letting them know you appreciate their story.

Give a tip

Sign up for Geek Culture Hits

By Geek Culture

Subscribe to receive top 10 most read stories of Geek Culture — delivered straight into your inbox, once a week. Take a look.

Your email

Get this newsletter

By signing up, you will create a Medium account if you don't already have one. Review our Privacy Policy for more information about our privacy practices.