

ClickHouse — Online Analytical Processing Database Management System

Elasticsearch — Distributed Search Engine

Prepared by : Hagar Muhamed

Lecture 9





مجموعة جامعة
الجميع الذكية
للتعليم القابضة



ClickHouse — Online Analytical Processing Database Management System &
Elasticsearch — Distributed Search Engine

Lecture 9

Prepared by
Dr. Hagar Muhamed

www.midocean.ae

ClickHouse Overview

ClickHouse Architecture and Basic Features
Enhanced Features of ClickHouse
Elasticsearch

Chapter

09

Content



What Is ClickHouse?

ClickHouse is a high-performance, column-oriented SQL database management system (DBMS) for online analytical processing (OLAP). It is available as both an [open-source software](#) and a [cloud offering](#).

Chapter

09

ClickHouse is independent of the Hadoop big data system and features ultimate compression rate and fast query. It supports SQL query and delivers superior query performance, especially the aggregation analysis and query performance based on large and wide tables. Its query speed is one order of magnitude faster than that of other analytical databases.

What is OLAP?

OLAP scenarios require real-time responses on top of large datasets for complex analytical queries with the following characteristics:

- Datasets can be massive - billions or trillions of rows
- Data is organized in tables that contain many columns
- Only a few columns are selected to answer any particular query
- Results must be returned in milliseconds or seconds

Column-Oriented vs Row-Oriented Databases

In a row-oriented DBMS, data is stored in rows, with all the values related to a row physically stored next to each other.

In a column-oriented DBMS, data is stored in columns, with values from the same columns stored together.

Why is ClickHouse so fast?

ClickHouse uses all available system resources to their full potential to process each analytical query as fast as possible. This is made possible due to a unique combination of analytical capabilities and attention to the low-level details required to implement the fastest OLAP database.

Chapter

09

ClickHouse Advantages

Comprehensive DBMS
functions

Column-oriented
storage and data
compression

Vectorized execution
engine

Relational model and
SQL query

Data sharding and
distributed query

Excellent
performance

Comprehensive DBMS functions

ClickHouse has comprehensive database management functions, including the basic functions of a DBMS:

Data Definition Language (DDL): allows databases, tables, and views to be dynamically created, modified, or deleted without restarting services.

Data Manipulation Language (DML): allows data to be queried, inserted, modified, or deleted dynamically.

Permission control: supports user-based database or table operation permission settings to ensure data security.

Data backup and restoration: supports data backup, export, import, and restoration.

Distributed management: provides the cluster mode to automatically manage multiple database nodes.

Column-oriented storage and data compression

ClickHouse is a database that uses column-based storage. Data is organized by column. Data in the same column is stored together, and data in different columns is stored in different files.

During data query, column-based storage can reduce the data scanning range and data transmission size, thereby improving data query efficiency.

Columnar databases store data in the same column together and data in different columns separately. Columnar databases are more suitable for online analytical processing (OLAP) scenarios.

- **Vectorized execution engine**

- ClickHouse uses CPU's Single Instruction Multiple Data (SIMD) for vectorized execution. SIMD is an implementation mode that uses a single instruction to operate multiple pieces of data and improves performance with data parallelism. The principle of SIMD is to implement parallel data operations at the CPU register level.

- **Relational model and SQL query**

- ClickHouse uses relational models so the cost of migrating systems built on traditional relational databases or data warehouses to ClickHouse is lower.
- ClickHouse uses SQL as the query language and provides standard SQL query APIs for existing third-party analysis visualization systems to easily integrate with ClickHouse.

- **Data sharding and distributed query**

- The ClickHouse cluster consists of one or more shards, and each shard corresponds to one ClickHouse service node. The maximum number of shards depends on the number of nodes (one shard corresponds to only one service node).
- ClickHouse introduces the concepts of local table and distributed table. A local table is equivalent to a data shard. A distributed table does not store any data. It is an access proxy of the local table and functions as the sharding middleware. With distributed tables, multiple data shards can be accessed by using the proxy, thereby implementing distributed query.

ClickHouse Overview

ClickHouse Architecture and Basic Features

Enhanced Features of ClickHouse

Elasticsearch

Chapter

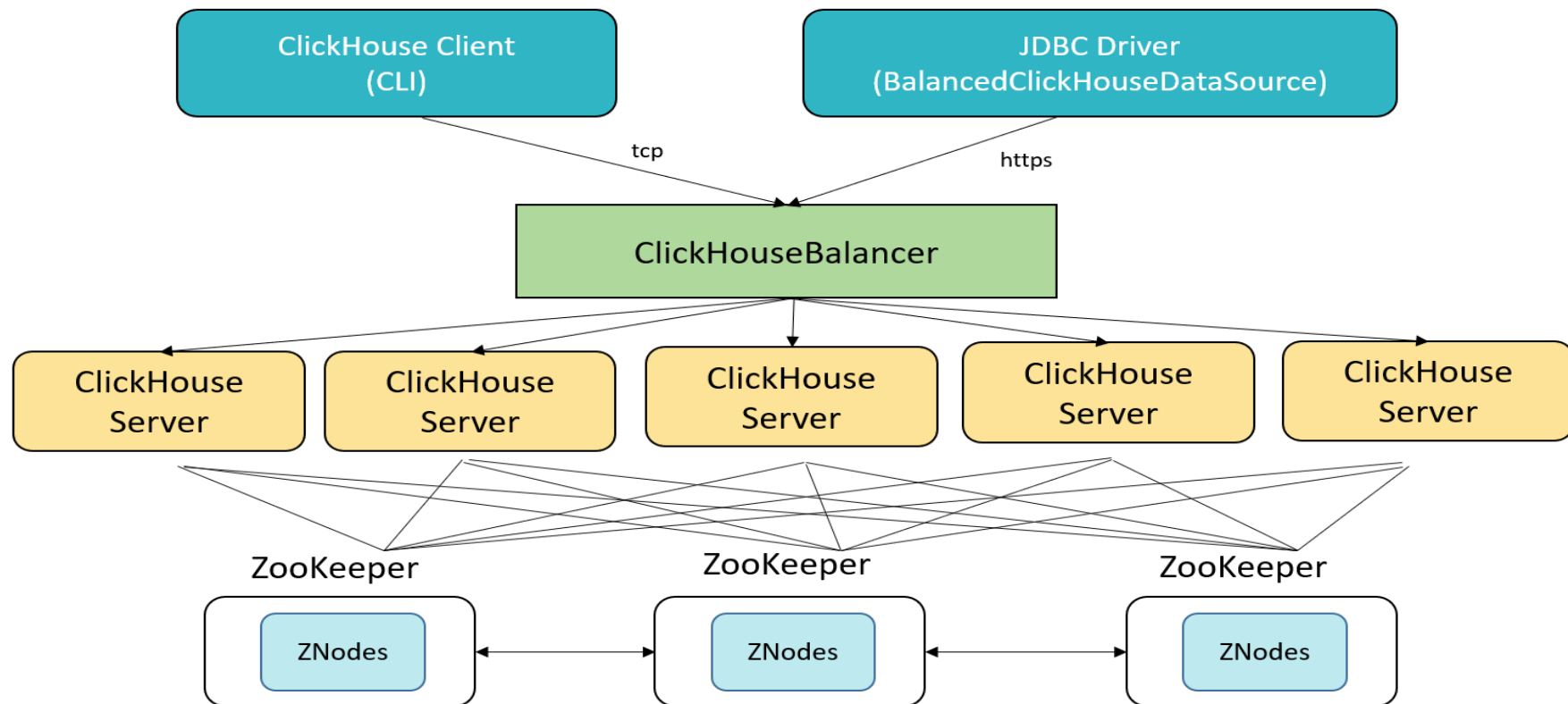
09

Content



Chapter
09

ClickHouse Architecture



SQL Capability

ClickHouse supports a declarative query language based on SQL. Supported queries include **GROUP BY**, **ORDER BY**, **FROM**, **JOIN**, **IN**, and uncorrelated subqueries. The language is basically compatible with standard syntax. It supports views and materialized views, but does not support standard **UPDATE** or **DELETE**, which is implemented through **ALTER TABLE**. Correlated (dependent) subqueries and window functions are not supported.

Common SQL syntax of ClickHouse

CREATE DATABASE: Creating a database

CREATE TABLE: Creating a table

INSERT INTO: Inserting data into a table

SELECT: Querying table data

ALTER TABLE: Modifying a table structure

DESC: Querying a table structure

Data Shards vs. Replicas

From the data perspective

Assume that N ClickHouse nodes form a cluster and each node in the cluster has a table Y with the same structure. If data in table Y of $N1$ is completely different from the data in table Y of $N2$, $N1$ and $N2$ are shards of each other. If their data is the same, they are replicas of each other. In other words, data between shards is different, but data between replicas is exactly the same.

From the function perspective

The main purpose of using replicas is to prevent data loss and increase data storage redundancy, while the main purpose of using shards is to implement horizontal data partitioning.

DROP: Deleting a table

SHOW: Displaying information about databases and tables

ClickHouse Overview
ClickHouse Architecture and Basic Features
Enhanced Features of ClickHouse
Elasticsearch

Chapter

09

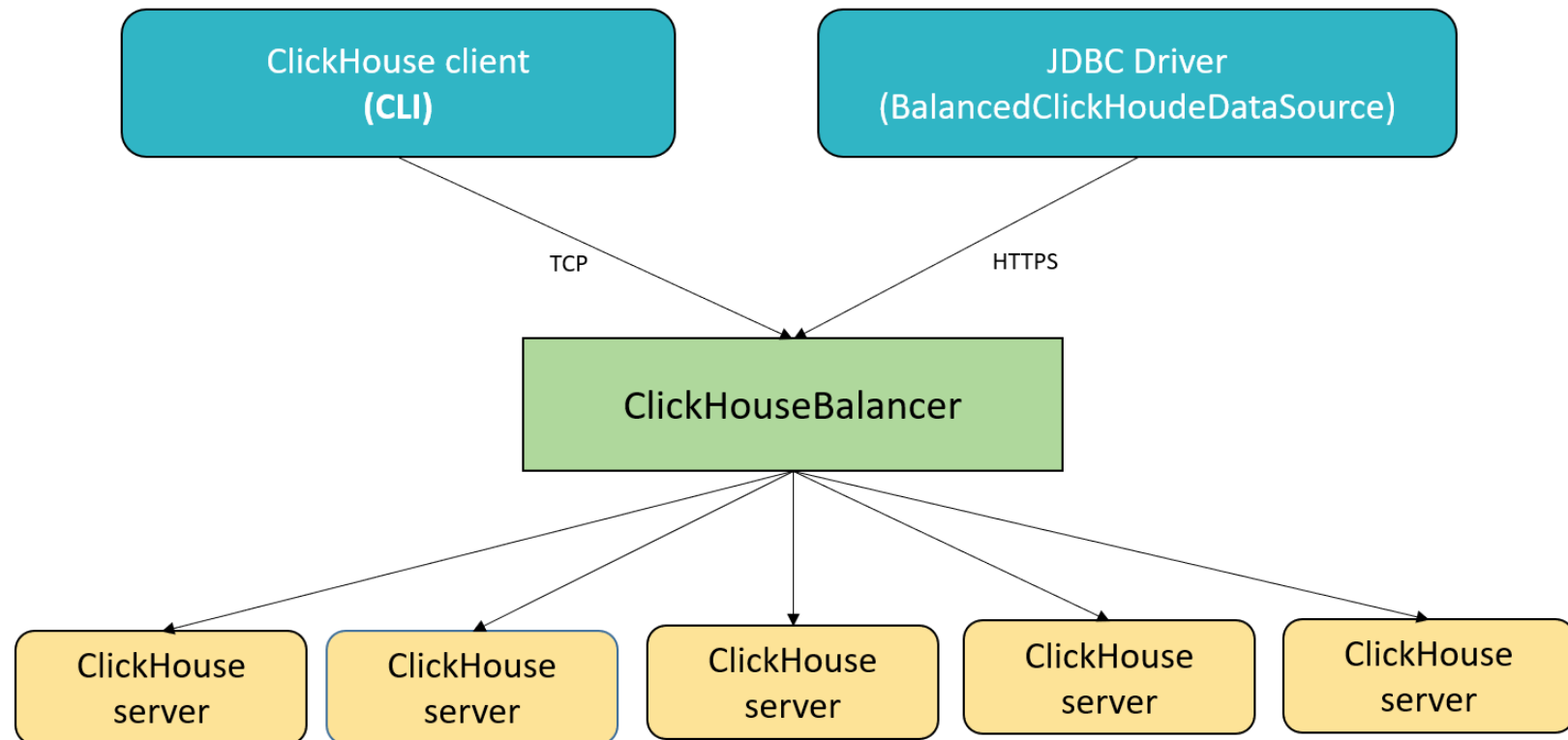
Content



Chapter

09

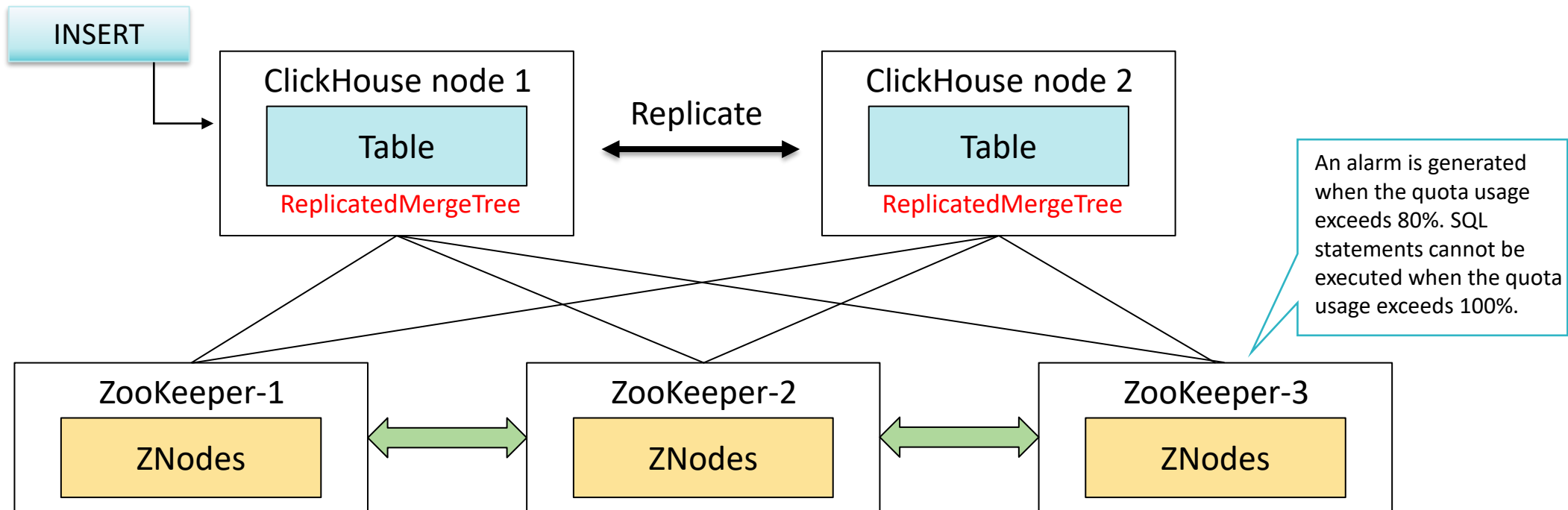
Load Balancing



Chapter

09

Automatic Cluster Topology and ZooKeeper Overload Prevention



ClickHouse
Elasticsearch Overview
System Architecture
Key Features

Chapter

09

Content



Elasticsearch is a high-performance Lucene-based full-text search service. It is a distributed RESTful search and data analysis engine and can also be used as a NoSQL database.

- Lucene extension

- Seamless switchover between the prototype environment and production environment

- Horizontal scaling

- Support for structured and unstructured data

Elasticsearch Features

High performance

- Search results can be obtained immediately, and the inverted index for full-text search is implemented.

Scalability

- Horizontal scaling is supported. Elasticsearch can run on hundreds or thousands of servers.

Relevance

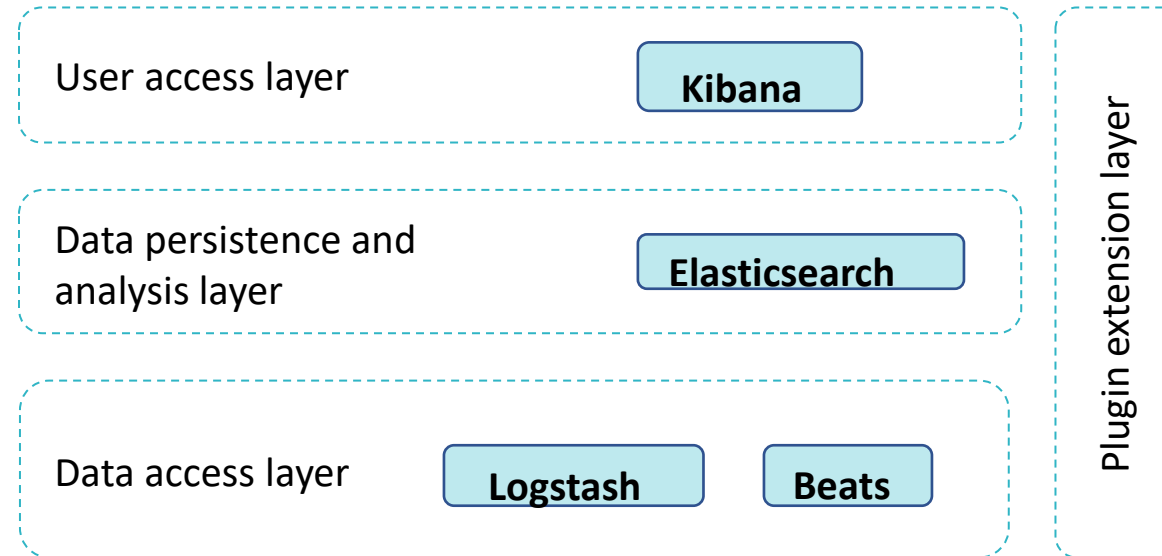
- Searches results are sorted based on elements (from word frequency).

Reliability

- Faults (such as hardware faults and network segmentation) are automatically detected, ensuring cluster (and data) security and availability.

- Elasticsearch is used for log search and analysis, spatiotemporal search, time sequence search, and intelligent search.
 - **Complex data types:** Structured data, semi-structured data, and unstructured data need to be queried. Elasticsearch can perform a series of operations such as cleansing, word segmentation, and inverted index creation on the preceding data types, and then provide the full-text search capability.
 - **Diversified search criteria:** Full-text search criteria contain words or phrases.
 - **Write and read:** The written data can be searched in real time.

Elasticsearch Ecosystem



ClickHouse
Elasticsearch Overview
System Architecture
Key Features

Chapter

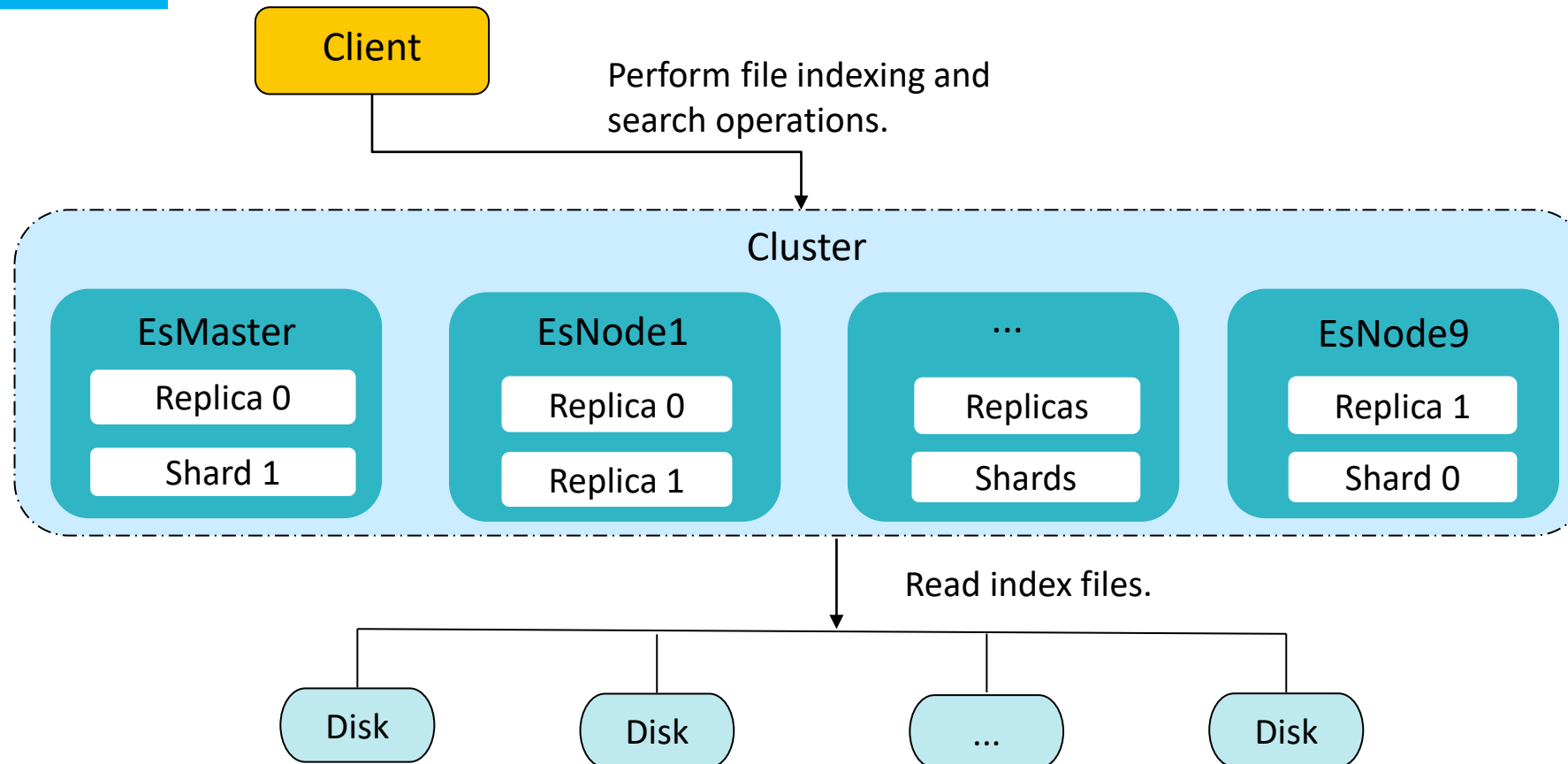
09

Content



Chapter
09

Elasticsearch System Architecture

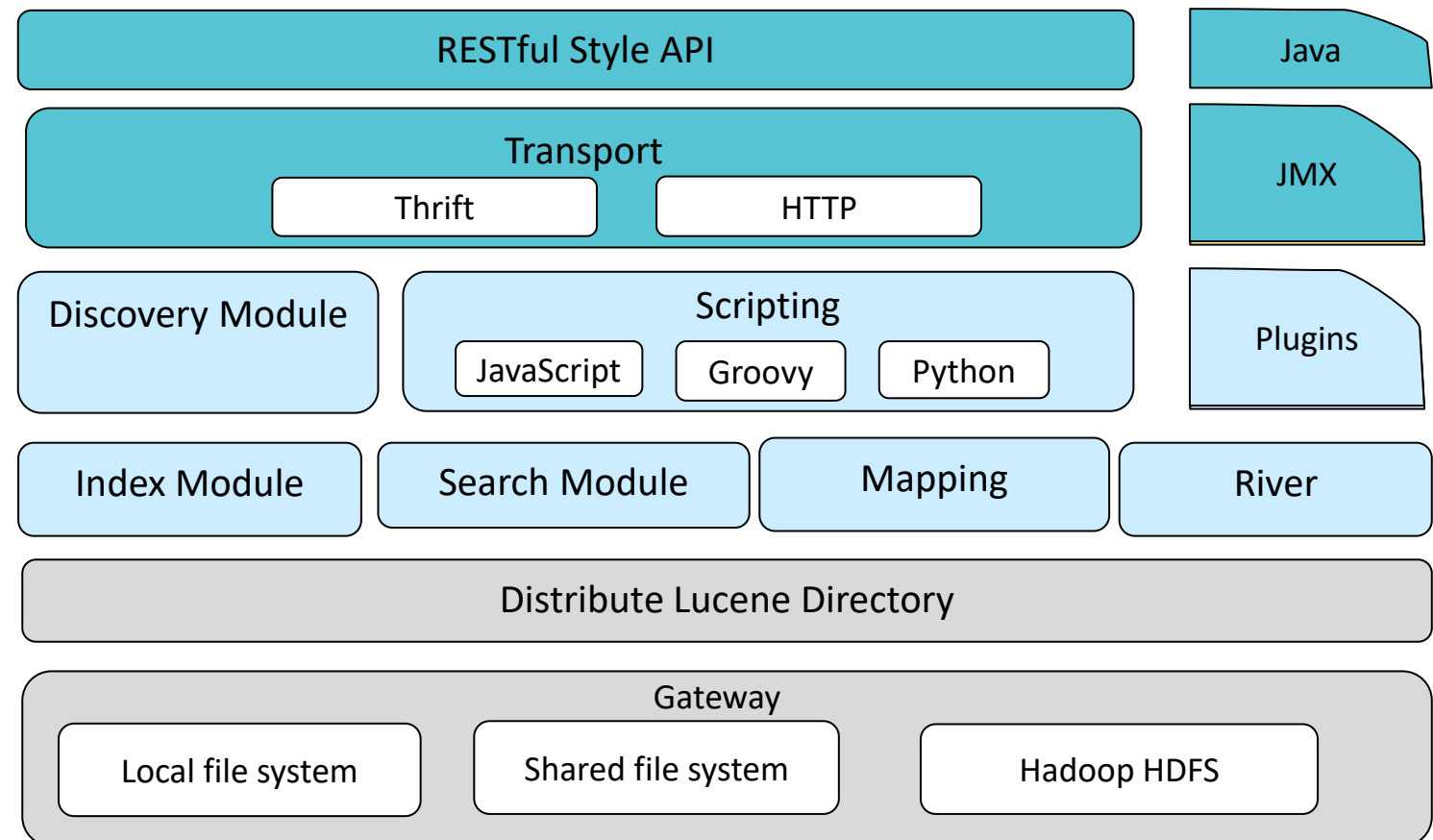


Chapter

09

Elasticsearch Internal Architecture

- Elasticsearch provides RESTful APIs or APIs for other languages (such as Java).
- The cluster discovery mechanism is used.
- Script languages are supported.
- The underlying layer is based on Lucene, ensuring absolute independence of Lucene.
- Indexes are stored in local files, shared files, and HDFS.



Basic Concepts of Elasticsearch (1)

- Index
 - A logical namespace in Elasticsearch
- Type
 - Used to store different types of documents.
- Document
 - A basic unit that can be indexed
- Mapping
 - Used to restrict the field type

Basic Concepts of Elasticsearch (2)

- **Cluster**
 - Each cluster contains multiple nodes, one of which is the master node (the rest are slave nodes). The master node can be elected.
- **EsNode**
 - Elasticsearch node. A node is an Elasticsearch instance.
- **EsMaster**
 - Master node that temporarily manages cluster-level changes, such as creating or deleting indexes and adding or removing nodes. A master node does not involve in document-level change or search. When the traffic increases, the master node does not affect the cluster performance.
- **shards**
 - Index shard. Elasticsearch splits a complete index into multiple shards and distributes them on different nodes.

Basic Concepts of Elasticsearch (3)

- Replicas
 - Index replica. Elasticsearch allows you to set multiple replicas for an index. Replicas can improve the fault tolerance of the system. When a shard on a node is damaged or lost, the data can be recovered from the replica
- Recovery
 - Data recovery or re-distribution. When a node is added to or deleted from the cluster, Elasticsearch redistributes shards based on the load of the node. When a failed node is restarted, data will be recovered.

Basic Concepts of Elasticsearch (4)

Gateway

Mode for storing Elasticsearch index snapshots. By default, Elasticsearch stores indexes in the memory and only makes them persistent on the local hard disk when the memory is full. The gateway stores index snapshots,

Transport

Interaction mode between an Elasticsearch internal node or cluster and the client. By default, internal nodes use the TCP protocol for interaction.

ClickHouse
Elasticsearch Overview
System Architecture
Key Features

Chapter

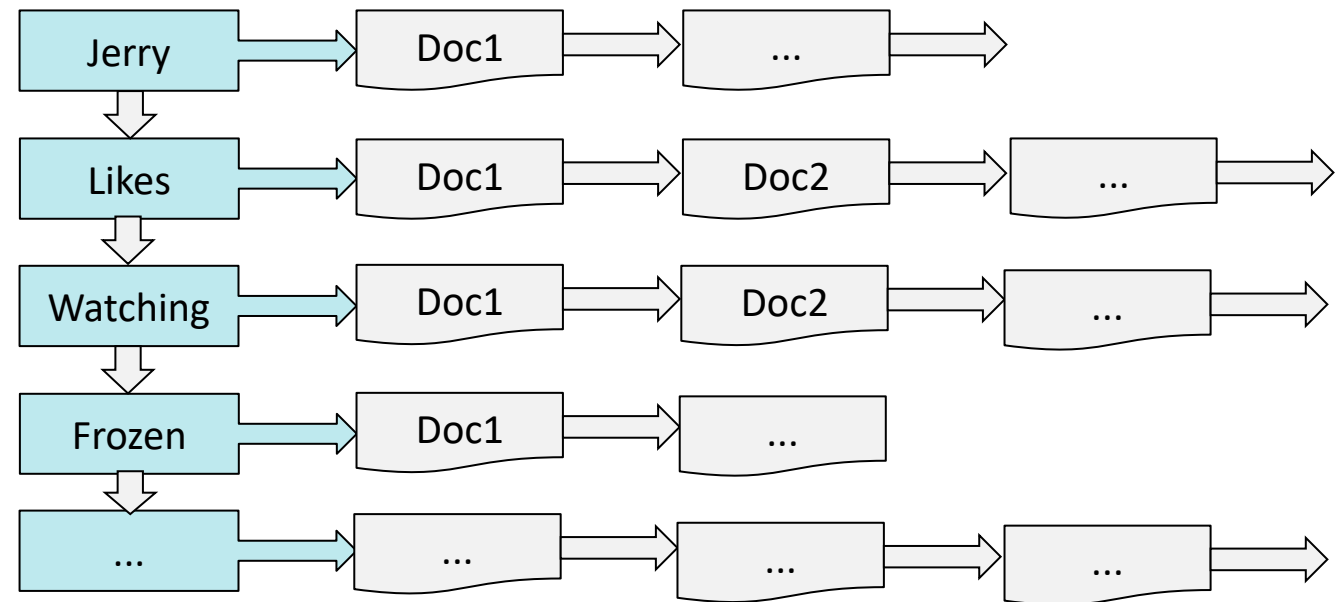
09

Content



Elasticsearch Inverted Index

- Forward index: Values are searched for based on keys. That is, specific information that meets the search criteria is located based on keys.
- Inverted index: Keys are searched for based on values. In full-text search, a value is the searched keyword. Corresponding documents are located based on values.



Elasticsearch Capacity Expansion

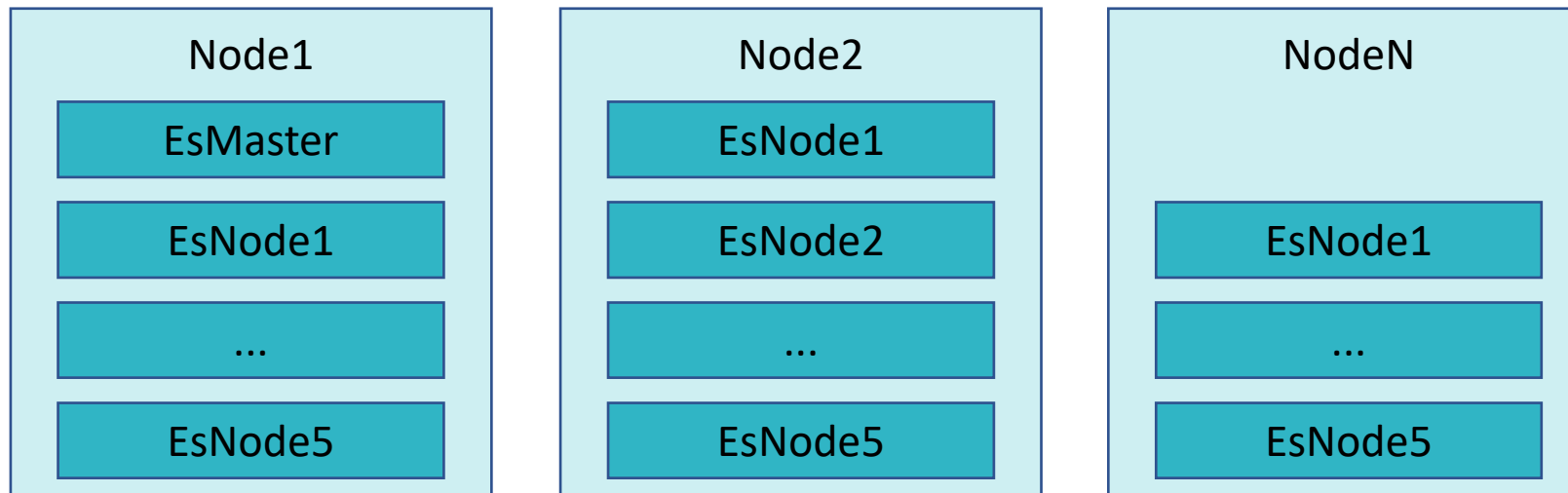
- Scenarios:
 - High physical resource consumption such as high CPU and memory usages of Elasticsearch service nodes, and insufficient disk space
 - Excessive index data volume for one Elasticsearch instance, such as 1 billion data records or 1 TB data
- Methods:
 - Add EsNode instances.
 - Add nodes with EsNode instances.
- After capacity expansion, use the automatic balancing policy.

Elasticsearch Capacity Reduction

- Scenarios:
 - OS reinstallation on nodes required
 - Reduced amount of cluster data
 - Out-of-service
- Method:
 - Delete an Elasticsearch instance on the Cloud Search Service (CSS) console.
- Precautions:
 - Ensure that replicas in the shard of the instance to be deleted exist in another instance.
 - Ensure that data in the shard of the instance to be deleted has been migrated to another node.

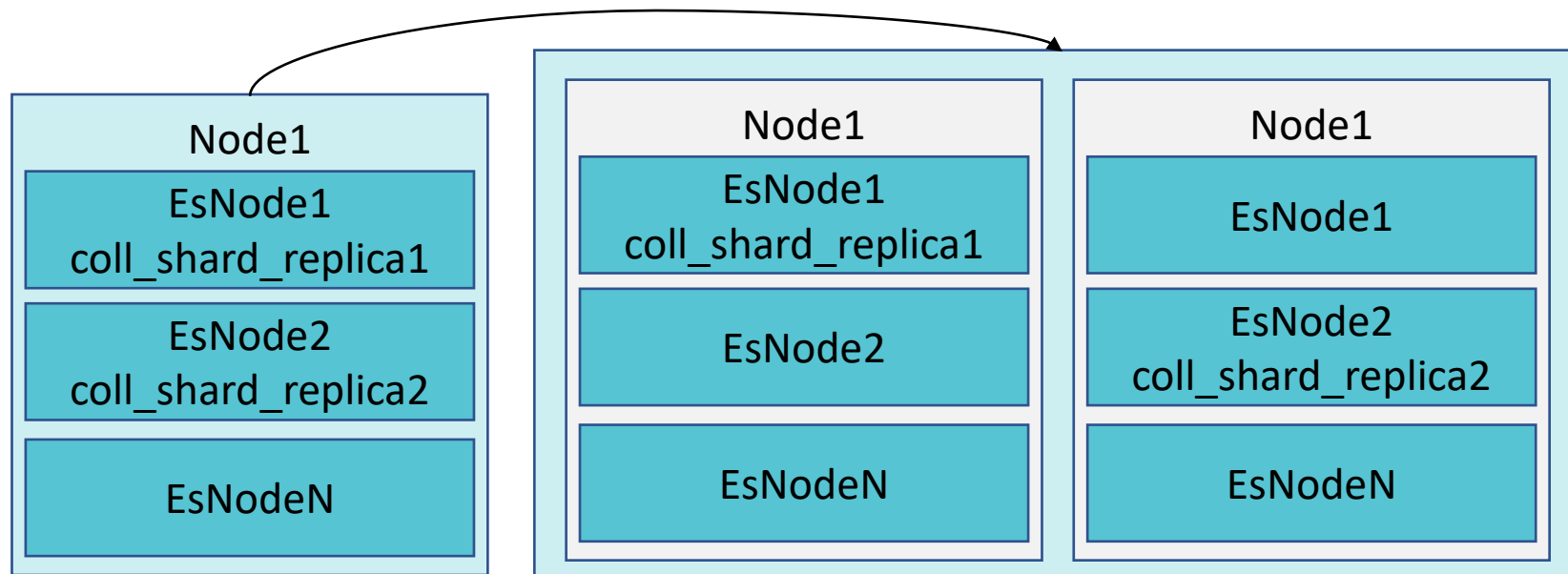
Elasticsearch Multi-Instance Deployment on a Node

Multiple Elasticsearch instances can be deployed on one node, and are differentiated from each other based on the IP address and port number. This method increases the usages of the single-node CPU, memory, and disk, and also improves the indexing and search capability of Elasticsearch.



Elasticsearch Cross-Node Replica Allocation Policy

- When multiple instances are deployed on a single node with multiple replicas, if the replicas can only be allocated across instances, a single point of failure may occur. To solve this problem, set parameter **cluster.routing.allocation.same_shard.host** to **true**.



Summary:

ClickHouse is an open-source columnar database management system designed for high-performance analytics. It excels at processing large volumes of data quickly and efficiently, making it ideal for real-time analytics and data warehousing. ClickHouse supports a variety of data formats.

Elasticsearch is a distributed, open-source search and analytics engine. It is designed for full-text search, real-time data indexing, and analysis of large datasets. for building search functionality in applications, including filtering, aggregations, and geospatial search. Elasticsearch's distributed architecture provides scalability and fault tolerance, making it a popular choice for organizations dealing with diverse data sources and complex querying needs.



مجموعة جامعة
الجميع الذكية
للتعليم القابضة

MIDOCEAN
UNIVERSITY
BY EVERYONE'S SMART UNIVERSITY



Thank You

