INTRODUCTION TO HBASE

OVERVIEW OF HBASE

HBase is a distributed, scalable, NoSQL database designed to handle large amounts of sparse data. It is built on top of the Hadoop Distributed File System (HDFS) and provides real-time read and write access to large datasets. HBase is designed to store and manage big data across a distributed environment, offering high performance and scalability. It supports column-oriented storage, which allows for efficient data retrieval and high throughput.

KEY FEATURES:

- **COLUMN-FAMILY STORAGE**: Data is organized into column families, allowing efficient access to subsets of columns.
- **SCALABILITY**: Supports horizontal scaling by adding more RegionServers.
- **HIGH AVAILABILITY**: Distributed nature provides fault tolerance and high availability.

HBASE ARCHITECTURE

- **HBASE MASTER**: Manages the overall cluster, including table schema changes, load balancing, and RegionServer assignment. It coordinates with ZooKeeper to maintain cluster health.
- **REGIONSERVERS**: Handle the storage and retrieval of data. Each RegionServer manages multiple regions and handles read/write requests from clients.
- **REGIONS**: Divisions of a table where data is stored. Each region is a subset of a table's data and is split into smaller regions as the dataset grows.
- **COLUMN FAMILIES**: Logical groupings of columns in a table, providing efficient access to related data. Each column family is stored separately on disk.
- **HBASE ZOOKEEPER**: Coordinates distributed services and maintains configuration information, cluster state, and leader election.

INSTALLATION

PREREQUISITES:

- Ensure Hadoop and Java are installed and properly configured on all nodes.
- Configure ZooKeeper, as HBase depends on it for coordination.

DOWNLOAD HBASE:

- Obtain the latest HBase binary from the Apache HBase website.

CONFIGURATION:

- Edit **hbase-site.xml** to set properties such as **hbase.root.logger**, **hbase.zookeeper.quorum**, and **hbase.master**.
- Configure HDFS properties in **core-site.xml** and **hdfs-site.xml**.

START HBASE:

- Use **start-hbase.sh** to launch HBase services, including the Master and RegionServers.

VERIFY:

- Access the HBase web UI at **http://<hbase-master-host>:16010** to check the status of the cluster.

THE HBASEADMIN AND HBASE SECURITY

VARIOUS OPERATIONS ON TABLES

CREATING TABLES:

create 'myTable', 'cf1', 'cf2'

MODIFYING TABLES:

alter 'myTable', { NAME => 'cf3' }

DELETING TABLES:

disable 'myTable'

drop 'myTable'

HBASE GENERAL COMMAND AND SHELL

HBase Shell: An interactive command-line interface to perform operations on HBase tables.

CREATE TABLE:

create 'tableName', 'cf1', 'cf2'

INSERT DATA:

put 'tableName', 'rowKey', 'cf:qualifier', 'value'

RETRIEVE DATA:

get 'tableName', 'rowKey'

DELETE DATA:

delete 'tableName', 'rowKey', 'cf:qualifier'

LIST TABLES:

list

JAVA CLIENT API FOR HBASE

The Java Client API provides classes and methods for interacting programmatically with HBase. Key classes include:

- **HTABLE**: For performing operations on tables.
- **PUT**: To insert or update rows.
- **GET**: To retrieve rows.
- **RESULT**: Contains the result of a Get operation.

ADMIN API

The Admin API offers administrative functions to manage HBase resources:

- **HBASEADMIN**: Used for creating, modifying, and deleting tables.
- **HBASECONFIGURATION**: Manages HBase configuration settings.

CRUD OPERATIONS

- **CREATE**: Add new rows using Put instances.
- **READ**: Retrieve rows with the Get class.
- **UPDATE**: Modify existing rows with Put.
- **DELETE**: Remove rows using the Delete class.

HBASE – SCAN, COUNT, AND TRUNCATE

SCAN:

Retrieves a range of rows. Can specify filters to narrow down the results.

scan 'tableName'

COUNT:

Counts rows in a table or within a specified range.

count 'tableName'

TRUNCATE:

Clears all data from a table without dropping the table schema.

truncate 'tableName'

HBASE SECURITY

- **AUTHENTICATION**: Typically managed via Kerberos to ensure secure access.
- **AUTHORIZATION**: Controls access to data using table-level and column-family permissions.
- **DATA ENCRYPTION**: Encrypts data at rest and in transit using HBase's built-in encryption capabilities or external tools.