databricks

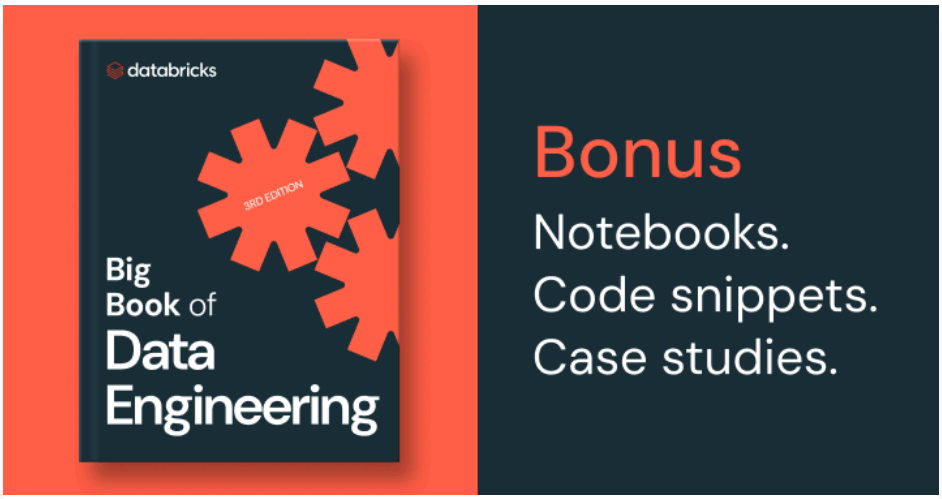
Apache Hive

What is Apache Hive?

Apache Hive is open-source data warehouse software designed to read, write, and manage large datasets extracted from the Apache Hadoop Distributed File System (HDFS) , one aspect of a larger Hadoop Ecosystem.

With extensive Apache Hive documentation and continuous updates, Apache Hive continues to innovate data processing in an ease-of-access way.

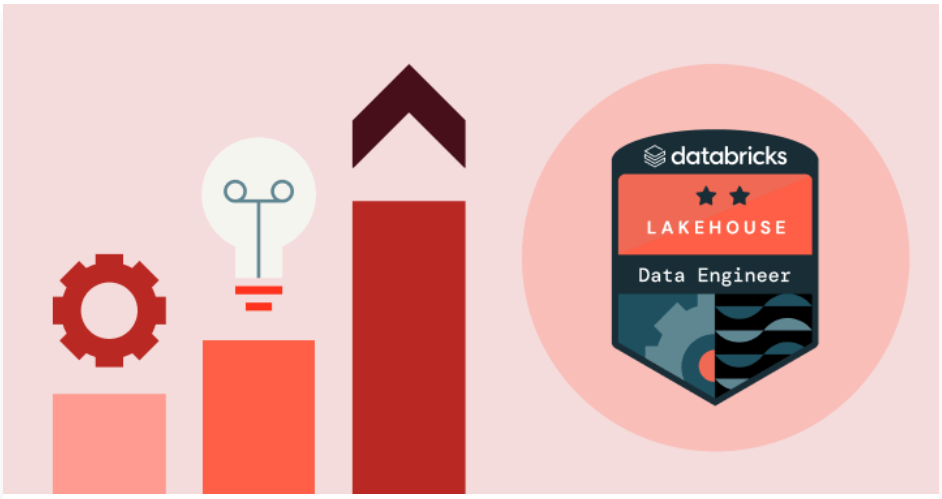
databricks



Big Book of Data Engineering

Fast-track your expertise with this essential guide for the AI era.

Read now →



Learn data engineering now

Watch 4 videos and pass a quiz to earn a badge.

Get started →

databricks

Get started with ETL

Learn about ETL pipelines with this O'Reilly technical guide.

Download now →

The History of Apache Hive

Apache Hive is an open source project that was conceived of by co-creators Joydeep Sen Sarma and Ashish Thusoo during their time at Facebook. Hive started as a subproject of Apache Hadoop, but has graduated to become a top-level project of its own. With the growing limitations of Hadoop and Map Reduce jobs and the increasing size of data from 10s of GB a day in 2006 to 1TB/day and to 15TB/day within a few years. The engineers at Facebook were unable to run the complex jobs with ease, giving way to the creation of Hive.

Apache Hive was created to achieve two goals - an SQL-based declarative language that also allowed engineers to be able to plug in their own scripts and programs when SQL did not suffice, which also enabled most of engineering world (SQL Skills based) to use Hive with minimal disruption or retraining compared to others.

developed in the walls of Facebook, Apache Hive is used and developed by other companies such as Netflix. Amazon maintains a software fork of Apache Hive included in Amazon Elastic MapReduce on Amazon Web Services.

What are some features of Hive?

Apache Hive supports the analysis of large datasets stored in Hadoop's HDFS and compatible file systems such as Amazon S3, Azure Blob Storage, Azure Data Lake Storage, Google Cloud Storage, and Alluxio.

It provides a SQL-like query language called HiveQL with schema-on-read and transparently converts queries to Apache Spark, MapReduce, and Apache Tez jobs. Other features of Hive include:

- Hive Data Functions help processing and querying big datasets. Some of the functionalities provided by these functions include string manipulation, date manipulation, type conversion, conditional operators, mathematical functions, and others
- Metadata storage in a relational database management system
- Different storage types such as Parquet, plain text, RCFile, HBase, ORC, and others
- Operating on compressed data stored into the Hadoop ecosystem using algorithms
- Built-in user-defined functions (UDFs) to manipulate dates, strings, and other data-mining tools. Hive supports extending the UDF set to handle use-cases not supported by built-in functions
- SQL-like queries (HiveQL), which are implicitly converted into MapReduce or Tez, or Spark jobs

databricks

Apache Hive components

The key components of the Apache Hive architecture are the Hive Server 2, Hive Query Language (HQL), the External Apache Hive Metastore, and the Hive Beeline Shell.

Hive Server 2

The Hive Server 2 accepts incoming requests from users and applications and creates an execution plan and auto generates a YARN job to process SQL queries. The server also supports the Hive optimizer and Hive compiler to streamline data extraction and processing.

Hive Query Language

By enabling the implementation of SQL-reminiscent code, the Apache Hive negates the need for long-winded JavaScript codes to sort through unstructured data and allows users to make queries using built-in HQL statements (HQL). These statements can be used to navigate large datasets, refine results, and share data in a cost-effective and time-efficient manner.

The Hive Metastore

The central repository of the Apache Hive infrastructure, the metastore is where all of the Hive's metadata is stored. In the metastore, metadata can also be formatted into Hive tables and partitions to compare data across relational databases. This includes table names, column names, data types, partition information, and data location on HDFS.

Hive Beeline Shell

In line with other database management systems (DBMS), Hive has its own built-in command-line interface where users can

Database Connectivity or Java Database Connectivity application.

How does Apache Hive software work?

The Hive Server 2 accepts incoming requests from users and applications before creating an execution plan and automatically generates a YARN job to process SQL queries. The YARN job may be generated as a MapReduce, Tez, or Spark workload.

This task then works as a distributed application in Hadoop. Once the SQL query has been processed, the results will either be returned to the end-user or application, or transmitted back to the HDFS.

The Hive Metastore will then leverage a relational database such as Postgres or MySQL to persist this metadata, with the Hive Server 2 retrieving table structure as part of its query planning. In some cases, applications may also interrogate the metastore as part of their underlying processing.

Hive workloads are then executed in YARN, the Hadoop resource manager, to provide a processing environment capable of executing Hadoop jobs. This processing environment consists of allocated memory and CPU from the various worker nodes in the Hadoop cluster.

YARN will attempt to leverage HDFS metadata information to ensure processing is deployed where the needed data resides, with MapReduce, Tez, Spark, or Hive can auto-generate code for SQL queries as MapReduce, Tez, or Spark jobs.

Despite Hive only recently leveraging MapReduce, most Cloudera Hadoop deployments will have Hive configured to use MapReduce, or sometimes Spark. Hortonworks (HDP) deployments normally have Tez set up as the execution engine.

databricks

types used by Apache Hive?

Through its use of batch processing, Apache Hive is able to efficiently extract and analyze petabytes of data at rapid speeds - making it ideal for not only processing the data but also running ad hoc queries.

The Apache Hive data types consist of five categories: Numeric, Date/Time, String, Complex, and Misc.

Numeric Data Types

As the name suggests, these data types are integer-based data types. Examples of these data types are 'TINYINT,' 'SMALLINT,' 'INT,' and 'BIGINT'.

Date/Time Data Types

These data types allow users to input a time and a date, with 'TIMESTAMP,' 'DATE,' and 'INTERVAL,' all being accepted inputs.

String Data Types

Again this type of data is very straightforward and allows for written text, or 'strings,' to be implemented as data for processing. String data types include 'STRING,' 'VARCHAR,' and 'CHAR.'

Complex Data Types

One of the more advanced data types, complex types record more elaborate data and consist of types like 'STRUCT', 'MAP,' 'ARRAY,' and 'UNION.'

Misc. Types

Data types that don't fit into any of the other four categories are known as miscellaneous data types and can take inputs such as 'BOOLEAN' or 'BINARY.'

In Apache Hive, Map Join is a feature employed to increase the speed and efficiency of a query by combining, or rather 'joining,' data from two tables whilst bypassing the Map-Reduce stages of the process.

What is a Relational Database Management System (RDBMS) and how does Apache Hive use it?

A Relational Database Management System (RDBMS) is a database model which operates by storing metadata in a row- or column-based table structure and allows for the connection and comparison of different datasets.

By using an RDBMS, Apache Hive can ensure that all the data is stored and processed safely, reliably, and accurately because integrated features such as role-based security and encrypted communications ensure that only the correct people are given access to the information extracted.

What's the difference between Apache Hive and a traditional RDBMS?

There are a few key differences between Apache Hive and an RDBMS:

- RDBMS functions work on read and write many times whereas Hive works on write once, read many times.
- Hive follows the schema-on-read rule, only meaning there is no data validation, checking or parsing, just copying/moving files. In traditional databases, a schema is applied to a table that enforces a schema on a write rule.

which other RDBMS may not need to.

Apache Hive vs. Apache Spark

An analytics framework designed to process high volumes of data across various datasets, Apache Spark provides a powerful user interface capable of supporting various languages from R to Python.

Hive provides an abstraction layer that represents the data as tables with rows, columns, and data types to query and analyze using an SQL interface called HiveQL. Apache Hive supports ACID transactions with Hive LLAP. Transactions guarantee consistent views of the data in an environment in which multiple users/processes are accessing the data at the same time for Create, Read, Update and Delete (CRUD) operations.

Databricks offers Delta Lake, which is similar to Hive LLAP in that it provides ACID transactional guarantees, but it offers several other benefits to help with performance and reliability when accessing the data. Spark SQL is Apache Spark's module for interacting with structured data represented as tables with rows, columns, and data types.

Spark SQL is SQL 2003 compliant and uses Apache Spark as the distributed engine to process the data. In addition to the Spark SQL interface, a DataFrames API can be used to interact with the data using Java, Scala, Python, and R. Spark SQL is similar to HiveQL.

Both use ANSI SQL syntax, and the majority of Hive functions will run on Databricks. This includes Hive functions for date/time conversions and parsing, collections, string manipulation, mathematical operations, and conditional functions.

There are some functions specific to Hive that would need to be converted to the Spark SQL equivalent or that don't exist in

This includes ANSI SQL aggregate and analytical functions. Hive is optimized for the Optimized Row Columnar (ORC) file format and also supports Parquet. Databricks is optimized for Parquet and Delta but also supports ORC. We always recommend using Delta, which uses open-source Parquet as the file format.

Apache Hive vs. Presto

A project originally established at Facebook, PrestoDB, more commonly known as Presto, is a distributed SQL query engine that allows users to process and analyze petabytes of data at a rapid speed. Presto's infrastructure supports the integration of both relational databases and non-relational databases from MySQL and Teradata to MongoDB and Cassandra.

Additional Resources

Hadoop to Lakehouse migration guide →

Migration hub →

Cloud Modernization Ebook: A business guide into the hidden value of migration from Hadoop →

On-demand quick demo's of the Databricks Lakehouse Platform →

Back to Glossary

databricks

Product

Solutions

Resources

About

Databricks Inc.
160 Spear Street, 15th Floor
San Francisco, CA 94105
1-866-330-0121

See Careers
at Databricks

© Databricks 2025. All rights reserved. Apache, Apache Spark, Spark, the Spark Logo, Apache Iceberg, Iceberg, and the Apache Iceberg logo are trademarks of the Apache Software Foundation.

Privacy Notice | Terms of Use | Modern Slavery Statement | California Privacy | Your Privacy Choices