

Ecommerce-data-warehouse in HIVE using AWS

Agenda

We will dig deeper into some of the Hive's analytical features for this hive project. Using SQL is still highly popular, and it will be for the foreseeable future. Most big data technologies have been modified to allow users to interact with them using SQL. This is due to the years of experience and expertise put into training, acceptance, tooling, standard development, and re-engineering. So, in many circumstances, employing these excellent SQL tools to access data may answer many analytical queries without resorting to machine learning, business intelligence, or data mining.

This big data project will look at Hive's capabilities to run analytical queries on massive datasets. We will use the Adventure works dataset in a MySQL dataset for this project, and we'll need to ingest and modify the data. We'll use Adventure works sales and Customer demographics data to perform analysis and answer the following questions:

- Which age group of customers contribute to more sales?
- To find the upper and lower discount limits offered for any product
- Sales contributions by customer
- To Understand customer persona purchasing pattern based on gender, education and yearly income
- To find the sales contribution by customers on the overall year to date sales belong to categorized by same gender, yearly income.
- To identify the top performing territory based on sales
- To find the territory-wise sales and their adherence to the defined sales quota.

Aim

To perform Hive analytics on Sales and Customer Demographics data using big data tools such as Sqoop, Spark, and HDFS.

Data Description

Adventure Works is a free sample database of retail sales data. In this project, we will be only using Customer test, Individual test, Credit card, Sales order details, Store, Sales territory, Salesperson, Sales order header, Special offer tables from this database.

Tech Stack

→ Language: SQL, Scala

→ Services: AWS EC2, Docker, MySQL, Sqoop, Hive, HDFS, Spark

AWS EC2

Amazon EC2 instance is a virtual server on Amazon's Elastic Compute Cloud (EC2) for executing applications on the Amazon Web Services (AWS) architecture. Corporate customers can use the Amazon Elastic Compute Cloud (EC2) service to run applications in a computer environment. Amazon EC2 eliminates the need for upfront hardware investment, allowing customers to design and deploy projects quickly. Users can launch up to 10 virtual servers, configure security and networking, and manage storage on Amazon EC2.

Docker

Docker is a free and open-source containerization platform, and it enables programmers to package programs into containers. These standardized executable components combine application source code with the libraries and dependencies required to run that code in any environment.

MySQL

MySQL is a SQL (Structured Query Language) based relational database management system. The platform can be used for data warehousing, e-commerce, logging applications, etc.

Sqoop

Sqoop is a data transfer mechanism for Hadoop and relational database servers. It is used to import data from relational databases such as MySQL and Oracle into Hadoop HDFS, Hive, and export data from the Hadoop file system to relational databases.

Scala

Scala is a multi-paradigm, general-purpose, high-level programming language. It's an object-oriented programming language that also supports functional programming. Scala applications can be converted to bytecodes and run on the Java Virtual Machine (JVM). Scala is a scalable programming language, and Javascript runtimes are also available.

Hive

Apache Hive is a fault-tolerant distributed data warehouse that allows for massive-scale analytics. Using SQL, Hive allows users to read, write, and manage petabytes of data.

Hive is built on top of Apache Hadoop, an open-source platform for storing and processing large amounts of data. As a result, Hive is inextricably linked to Hadoop and is designed to process petabytes of data quickly. Hive is distinguished by its ability to query large datasets with a SQL-like interface utilizing Apache Tez or MapReduce.

Approach

- Create an AWS EC2 instance and launch it.
- Create docker images using docker-compose file on EC2 machine via ssh.
- Create tables in MySQL.
- Load data from MySQL into HDFS storage using Sqoop commands.
- Move data from HDFS to Hive.
- Integrate Hive into Spark.
- Using scala programming language, extract Customer demographics information from data and store it as parquet files.
- Move parquet files from Spark to Hive.
- Create tables in Hive and load data from Parquet files into tables.
- Perform Hive analytics on Sales and Customer demographics data.

Project Takeaways

- Understanding various services provided by AWS
- Creating an AWS EC2 instance
- Connecting to an AWS EC2 instance via SSH
- Introduction to Docker
- Visualizing the complete Architecture of the system
- Usage of docker-composer and starting all tools
- Copying a file from a local machine to an EC2 machine
- Understanding the schema of the dataset
- Data ingestion/transformation using Sqoop, Spark, and Hive
- Moving the data from MySQL to HDFS
- Creating Hive table and troubleshooting it
- Using Parquet and Xpath to access schema
- Understanding the use of GROUP BY, GROUPING SETS, ROLL-UP, CUBE clauses
- Understanding different analytic functions in Hive

Note:

- PostgreSQL jar file
 1. Command to move jar file from local machine to ec2 machine =>
`scp -r -i "demo_hive.pem" postgresql-42.3.1.jar
ec2-user@ec2-3-93-63-210.compute-1.amazonaws.com:/home/ec2-user/`
 2. Command to move jar file from ec2 machine to Spark container =>
`docker cp /home/ec2-user/postgresql-42.3.1.jar
hdp_spark-master:/spark/jars`
- Hive-site.xml file
 1. Command to copy hive-site.xml file from Hive container and paste it in ec2 machine =>
`docker cp ra_hive-server:/opt/hive/conf/hive-site.xml /home/ec2-user`
 2. Command to copy hive-site.xml file from ec2 machine and paste it in Spark container =>
`docker cp /home/ec2-user/hive-site.xml hdp_spark-master:/spark/conf`