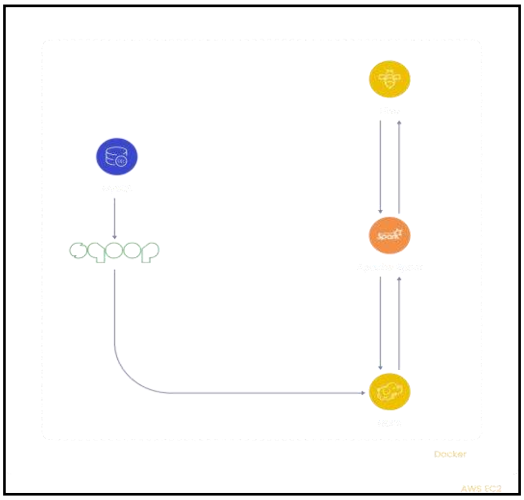
1. Hive Mini Project to Build a Data Warehouse for e-Commerce

In this hive project, you will design a data warehouse for e-commerce application to perform Hive analytics on Sales and Customer Demographics data using big data tools such as Sqoop, Spark, and HDFS.

- Create an AWS EC2 instance and launch it.
- Create docker images using docker-compose file on EC2 machine viassh.
- 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 demographicsinformation 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.



GCP and Azure

NOTE: For the commands that involve interacting with Virtual machine, please use the respective cloud provider syntax e.g. user@VM-Ip

In this hive project, you will design a data warehouse for e-commerce application to perform Hive analytics on Sales and Customer Demographics data using big data tools such as Sqoop, Spark, and HDFS.

For Google Cloud Platform (GCP) and Azure, sign in to the Console. Navigate to Compute Engine service (or Virtual Machine) and Create an Instance, that falls under the free tier. Select a suitable OS image, configure instance specifications, networking options, and storage preferences. Configure firewall rules to allow SSH access. Choose to either generate an SSH key or use an existing one. Review and create the instance. After provisioning, access the instance's external IP address to connect via SSH.

SSH to the machine and perform docker related tasks as given in the file *docker_setup_commands.txt*

SATISFY THE DEPENDENCIES

Need to store metastore dependent jar file into our Spark container. Use the PostgreSQL jar file - postgresql-42.3.1. Copy the JAR file from your local to your GCP / Azure VM.
From your VM instance home directory run this command-

docker cp /home/ec2-user/postgresql-42.3.1.jar hdp_spark-master:/spark/jars

Go to Spark shell using this command - *docker exec -i -t hdp_spark-master bash* - verify if the PostgreSQL jar is under the */spark/jars* directory

Now copy hive-site.xml to Spark directory (you can use the given file, copy it to compute machine and from there copy it to spark Sheel under /spark/conf OR copy it by going to hive prompt from your compute machine - the xml will be under /hive/conf)

UNDERSTAND THE BUSINESS PROBLEM

Refer to the Problem Statement or Business Objective section in the doc (Ecommerce DW - video notes.docx)

Talk about the Sales tables from the ER diagram a bit. We will need 9 tables to solve our Problem statement. We will be creating joins, views, etc. to solve our problem.

Log into the VM.
Follow the instructions in the file ec2 links and container commands.txt

To operate on XML data, we will need Spark

- Login to Spark container: docker exec -i -t hdp_spark-master bash
- Change to spark directory: cd Spark
- Run the spark shell: ./bin/spark-shell
- Now follow the commands in the file - 05_customer_demographic.scala

After completing the commands in 05_customer_demographic.scala, we will need to take care of integrating spark and hive. For that we need to make sure that there is PostgreSQL jar in /spark/jars folder in the spark container and inside /sparl/conf, hive-site.xml should be there (If it's not there, check in the hive container conf folder, you should have hive-site.xml. Copy to spark contain;'s confusion folder)

Post that, we need to copy the parquet files created previously using the commands in 05_customer_demographic.scala to the hive container. Container to container copy is not possible, so you need to copy to the VM first and from there copy to the hive container. (If needed you can use the commands in the file 06_File_copy_commands)

- After that go to the hive prompt from the hive container and follow the commands in the file - 07_customer_demograhics_creation.hql
- There on it is about performing the hive analytics. I have given the queries to perform the analysis based on the problem statements. You are free to create your own problem statements for analysis and then create queries to explain the same.