Real-Time E-commerce Data Processing and Analysis

Report

Of

Deployment Steps & Implementation

BIG DATA

Team Members

Mostafa Fathi Mohamed

Teacher Assistant in Nile University

Deployment Steps

1. **Docker-compose**
   - **Namenode**: Hadoop Namenode service.
     - Image: `bde2020/hadoop-namenode:2.0.0-hadoop3.2.1-java8`
     - Ports: `9870:9870`
     - Volumes: Binds to `hadoop_namenode` volume.
     - Environment: `CLUSTER_NAME=test`
     - Dependencies: None

   - **Datanode**: Hadoop Datanode service.
     - Image: `bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8`
     - Ports: `9864:9864`
     - Volumes: Binds to `hadoop_datanode` volume.
     - Environment: `SERVICE_PRECONDITION: "namenode:9870"`
     - Dependencies: Depends on the Namenode service.

   - **Resourcemanager**: Hadoop Resourcemanager service.
     - Image: `bde2020/hadoop-resourcemanager:2.0.0-hadoop3.2.1-java8`
     - Ports: `8088:8088`
     - Environment: `SERVICE_PRECONDITION: "namenode:9000 namenode:9870 datanode:9864"`
     - Dependencies: Depends on Namenode, Datanode services.

   - Nodemanager1: Hadoop Nodemanager service.
     - Image: `bde2020/hadoop-nodemanager:2.0.0-hadoop3.2.1-java8`
     - Ports: `8042:8042`
     - Environment: `SERVICE_PRECONDITION: "namenode:9000 namenode:9870 datanode:9864 resourcemanager:8088"`
     - Dependencies: Depends on Namenode, Datanode, Resourcemanager services.

- Historyserver: Hadoop Historyserver service.
  - Image: `bde2020/hadoop-historyserver:2.0.0-hadoop3.2.1-java8`
  - Ports: `8188:8188`
  - Volumes: Binds to `hadoop_historyserver` volume.

- Spark-master: Apache Spark Master service.
  - Image: `bde2020/spark-master:3.0.0-hadoop3.2`
  - Ports: `8080:8080`, `7077:7077`
  - Dependencies: Depends on Namenode, Datanode services.
  - Environment: Spark configuration.

- Spark-worker-1: Apache Spark Worker service.
  - Image: `bde2020/spark-worker:3.0.0-hadoop3.2`
  - Ports: `8081:8081`
  - Dependencies: Depends on Spark Master service.
  - Environment: Spark configuration.

- Zookeeper: Apache Zookeeper service.
  - Image: `wurstmeister/zookeeper:3.4.6`
  - Ports: `2181:2181`

- Kafka: Apache Kafka service.
  - Image: `wurstmeister/kafka:2.12-2.5.0`
  - Ports: `9092:9092`, `9093` (exposed)
  - Dependencies: Depends on Zookeeper service.
  - Environment: Kafka configuration.

- Volumes: Defines three volumes for Hadoop Namenode, Datanode, and Historyserver.

2. Hadoop:

- Download and install Hadoop using bde2020/hadoop-namenode:2.0.0-hadoop3.2.1-java8 image in the docker container.
- Configure Hadoop settings in hadoop-env.sh, core-site.xml, and hdfs-site.xml.
- Format the Hadoop Distributed File System (HDFS).
- Start Hadoop services using start-all.sh or individual commands (start-dfs.sh, start-yarn.sh).

3. Spark:

- Download and install Apache Spark using bde2020/spark-master:3.0.0-hadoop3.2 image in the docker container.
- Configure Spark settings in spark-env.sh.
- Set up Hadoop configuration in spark-defaults.conf.
- Start Spark services using start-master.sh and start-worker.sh.

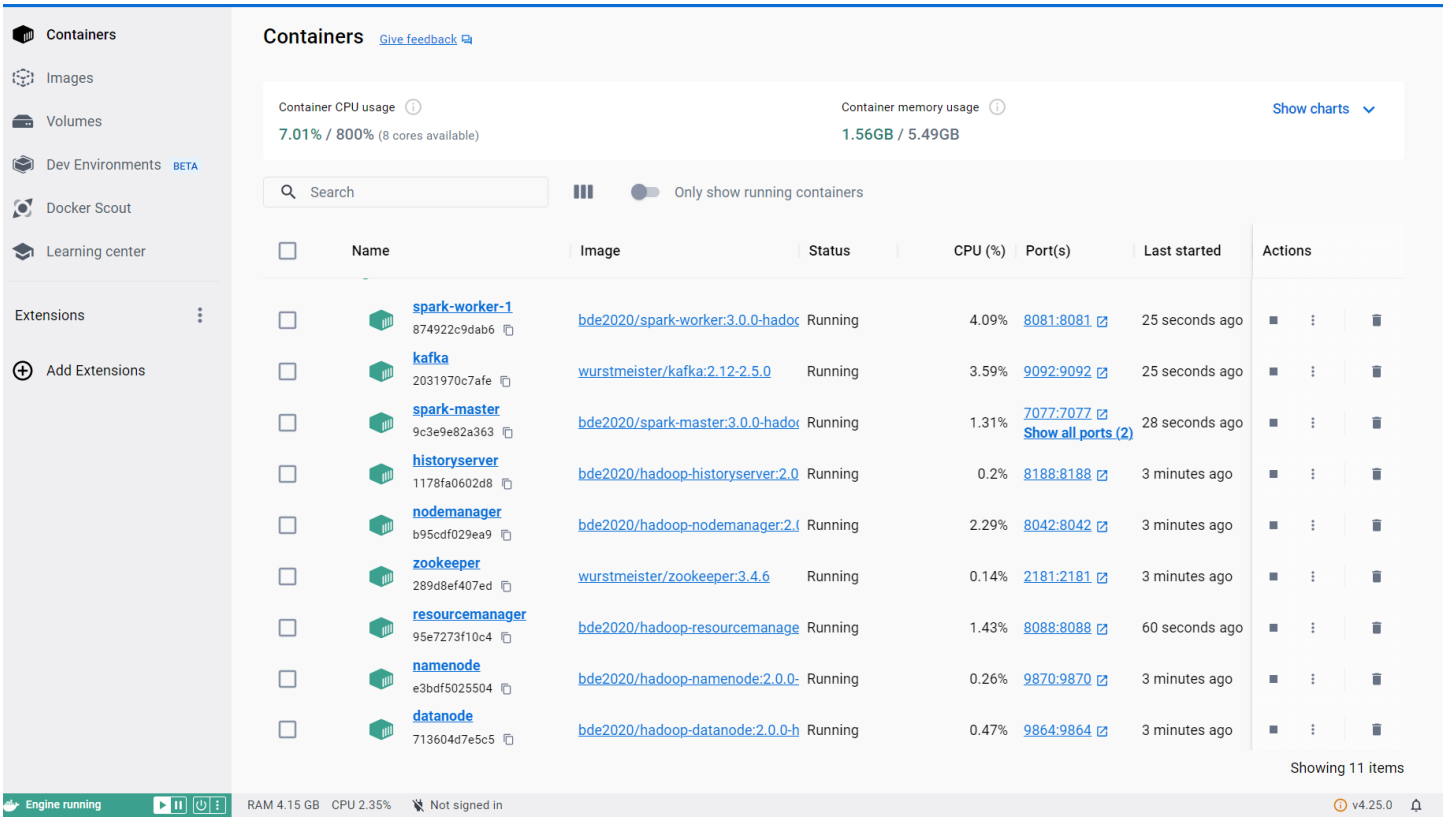
4. YARN:

- Ensure Hadoop is correctly configured and running.
- Configure YARN settings in yarn-site.xml.
- Start the ResourceManager and NodeManagers using start-yarn.sh.

5. **Kafka**:
   - **Download and Install Kafka**: Use the `wurstmeister/kafka:2.12-2.5.0` image to set up Kafka in a Docker container.
   - **Configure Kafka:** Adjust Kafka settings in the `server.properties` file within the Kafka container.
   - **Configure Zookeeper Connection:** Edit Kafka's `server.properties` to specify Zookeeper connection details: **KAFKA_ZOOKEEPER_CONNECT: zookeeper:2181**
6. **Zookeeper**:
   - Download and Install Zookeeper: Use the `wurstmeister/zookeeper:3.4.6` image to set up Zookeeper in a Docker container.
   - Configure Zookeeper: Customize Zookeeper settings in the `zoo.cfg` file within the Zookeeper container.
   - Start Zookeeper:
     - Ensure Zookeeper is correctly configured.
     - Start Zookeeper using the appropriate command within the container.

Implementations Steps

Using Docker-Compose to use spark, Hadoop, yarn, Kafka and Zookeeper: -



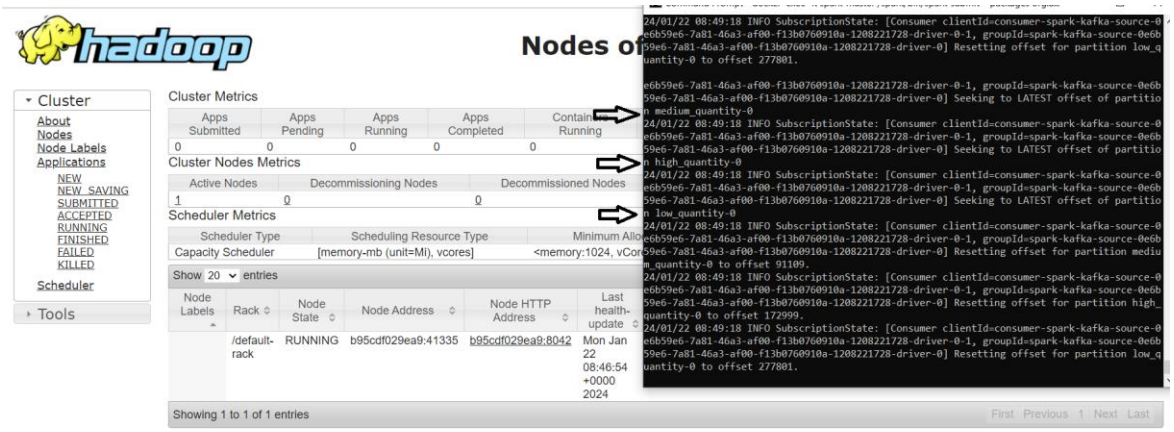
- **Kafka Producer**
  - Before use the script of Kafka producer, we should have a topics so we implement three topics to categories the data in it using the following:
    - ✓ docker exec -it kafka /opt/kafka/bin/kafka-topics.sh --create --topic low_quantity --bootstrap-server kafka:9092 --partitions 1 --replication-factor 1
    - ✓ docker exec -it kafka /opt/kafka/bin/kafka-topics.sh --create --topic medium_quantity --bootstrap-server kafka:9092 --partitions 1 --replication-factor 1
    - ✓ docker exec -it kafka /opt/kafka/bin/kafka-topics.sh --create --topic high_quantity --bootstrap-server kafka:9092 --partitions 1 --replication-factor 1

```
bash-5.1# kafka-topics.sh --list --bootstrap-server localhost:9092
__consumer_offsets
high_quantity
low_quantity
medium_quantity
topic_name
bash-5.1#
```

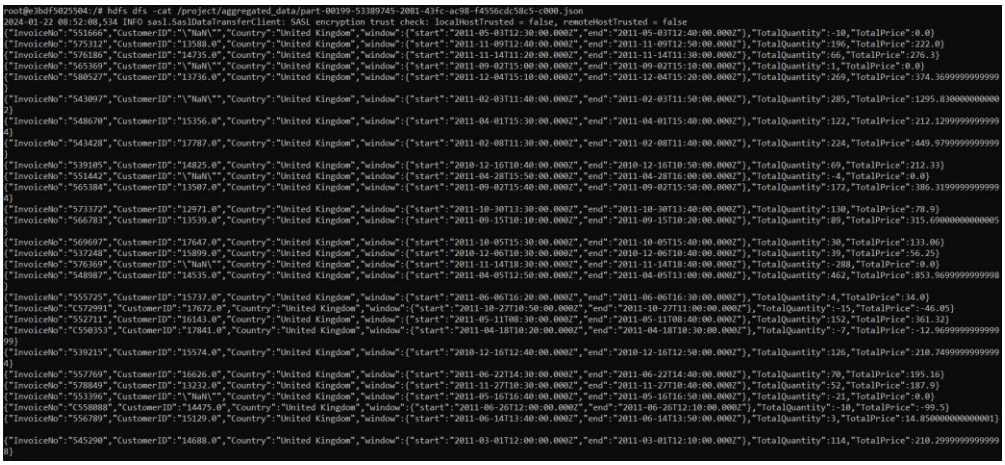
- copy the flat file and script from local device to spark-master container.
- After That execute our script in spark master container using the following command:
  - ✓ docker exec -it spark-master /spark/bin/spark-submit --packages org.apache.spark:spark-sql-kafka-0-10_2.12:3.0.0 /home/producer_script.py

Now, Data is categories in the three topics in **Zookeeper server**.

- **Spark**
  - Create and Determine data path from hdfs to store data in it after processing: - data_path = "hdfs://namenode:9000/project/aggregated_data"
  - Start executing our script with executing spark-master container and launch spark session: -
    - ✓ docker exec -it spark-master /spark/bin/spark-submit --packages org.apache.spark:spark-sql-kafka-0-10_2.12:3.0.0 /home/streaming_app.py



- After Appling processing using spark, we stored output in hdfs:-

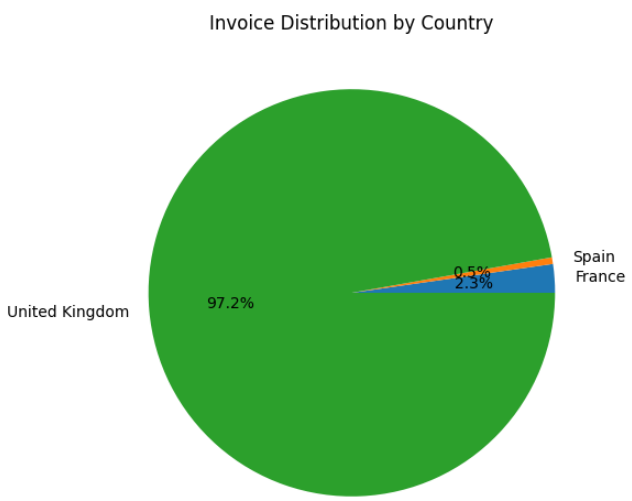


Now, we have filtered and aggregated data in HDFS, we want to apply our visualization to achieve the project goal.

Project Goal: -

- Which country among the following ("United Kingdom," "France," "Spain") has the highest number of billings?
- Display the top three customers with the highest invoice amounts from each of the countries "United Kingdom," "France," and "Spain."

Question 1: - as we note that highest number of billings come from 'United Kingdom'



Question 2: - The top three customers from "United Kingdom," "France," and "Spain."