

**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:

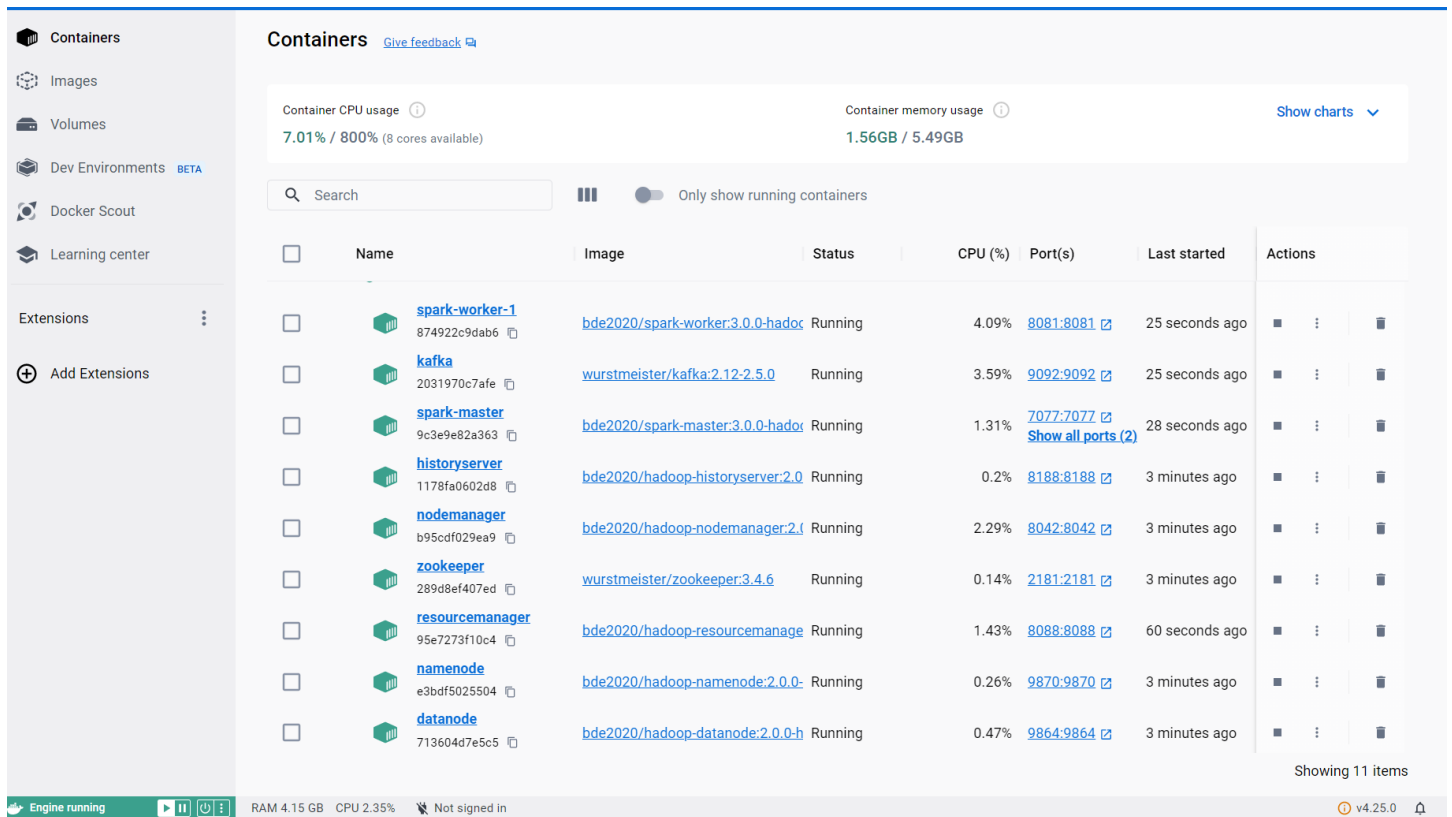
- **Download and Install Kafka:** Use the `wurstmeister/kafka:2.12-2.5.0` image to set up Kafka in a Docker container.
- **Configure Kafka:** Adjust Kafka settings in the `server.properties` file within the Kafka container.
- **Configure Zookeeper Connection:** Edit Kafka's `server.properties` to specify 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: -



The screenshot displays the Docker Desktop interface. On the left, a sidebar contains navigation options: Containers, Images, Volumes, Dev Environments (marked BETA), Docker Scout, and Learning center. Below this is an 'Extensions' section with an 'Add Extensions' button. The main area is titled 'Containers' and includes a search bar and a toggle for 'Only show running containers'. It shows a list of 11 running containers with columns for Name, Image, Status, CPU (%), Port(s), Last started, and Actions. The containers listed are: spark-worker-1, kafka, spark-master, historyserver, nodemanager, zookeeper, resourcemanager, namenode, and datanode. At the bottom, a status bar indicates 'Engine running', 'RAM 4.15 GB', 'CPU 2.35%', and 'Not signed in'.

Name	Image	Status	CPU (%)	Port(s)	Last started
spark-worker-1	bde2020/spark-worker:3.0.0-hadoop	Running	4.09%	8081:8081	25 seconds ago
kafka	wurstmeister/kafka:2.12-2.5.0	Running	3.59%	9092:9092	25 seconds ago
spark-master	bde2020/spark-master:3.0.0-hadoop	Running	1.31%	7077:7077	28 seconds ago
historyserver	bde2020/hadoop-historyserver:2.0	Running	0.2%	8188:8188	3 minutes ago
nodemanager	bde2020/hadoop-nodemanager:2.0	Running	2.29%	8042:8042	3 minutes ago
zookeeper	wurstmeister/zookeeper:3.4.6	Running	0.14%	2181:2181	3 minutes ago
resourcemanager	bde2020/hadoop-resourcemanager:2.0	Running	1.43%	8088:8088	60 seconds ago
namenode	bde2020/hadoop-namenode:2.0.0	Running	0.26%	9870:9870	3 minutes ago
datanode	bde2020/hadoop-datanode:2.0.0	Running	0.47%	9864:9864	3 minutes ago

### ■ 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#
```

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

- 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`



Cluster

About Nodes

Node Labels

Applications

NEW  
NEW SAVING  
SUBMITTED  
ACCEPTED  
RUNNING  
FINISHED  
FAILED  
KILLED

Scheduler

Tools

## Nodes of

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running
0	0	0	0	0

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes
1	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation
Capacity Scheduler	[memory-mb (unit-Mi), vcores]	<memory.1024, vcores

Show 20 entries

Node Labels	Rack	Node State	Node Address	Node HTTP Address	Last health update
/default-rack	RUNNING	b95cd029ea9:41335	b95cd029ea9:8042		Mon Jan 22 08:46:54 +0000 2024

Showing 1 to 1 of 1 entries

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

```

2024-01-22 08:52:08.534 info SAS: /api/agents/registered/data?offset=13389745&max=431&size=4556&id=580...
{"InvoicerNo": "551665", "CustomerID": "1", "Name": "H", "Country": "United Kingdom", "Window": "1", "Start": "2011-05-03T12:30:00.000Z", "End": "2011-05-03T17:40:00.000Z", "TotalQuantity": 10, "TotalPrice": 0.3},
{"InvoicerNo": "551665", "CustomerID": "1", "Name": "H", "Country": "United Kingdom", "Window": "1", "Start": "2011-05-03T17:40:00.000Z", "End": "2011-05-03T18:00:00.000Z", "TotalQuantity": 196, "TotalPrice": 22.0},
{"InvoicerNo": "576185", "CustomerID": "1", "Name": "H", "Country": "United Kingdom", "Window": "1", "Start": "2011-11-14T17:00:00.000Z", "End": "2011-11-14T17:30:00.000Z", "TotalQuantity": 66, "TotalPrice": 27.6},
{"InvoicerNo": "563569", "CustomerID": "1", "Name": "H", "Country": "United Kingdom", "Window": "1", "Start": "2011-09-02T15:00:00.000Z", "End": "2011-09-02T15:10:00.000Z", "TotalQuantity": 1, "TotalPrice": 0.0},
{"InvoicerNo": "568527", "CustomerID": "1", "Name": "H", "Country": "United Kingdom", "Window": "1", "Start": "2011-12-04T15:30:00.000Z", "End": "2011-12-04T15:30:00.000Z", "TotalQuantity": 269, "TotalPrice": 374.36999999999999},
{"InvoicerNo": "543697", "CustomerID": "1", "Name": "H", "Country": "United Kingdom", "Window": "1", "Start": "2011-02-02T17:40:00.000Z", "End": "2011-02-02T17:50:00.000Z", "TotalQuantity": 285, "TotalPrice": 1295.8300000000000},
{"InvoicerNo": "548670", "CustomerID": "1", "Name": "H", "Country": "United Kingdom", "Window": "1", "Start": "2011-04-04T17:30:00.000Z", "End": "2011-04-04T17:50:00.000Z", "TotalQuantity": 122, "TotalPrice": 121.22999999999999},
{"InvoicerNo": "543428", "CustomerID": "1", "Name": "H", "Country": "United Kingdom", "Window": "1", "Start": "2011-02-02T17:30:00.000Z", "End": "2011-02-02T17:40:00.000Z", "TotalQuantity": 224, "TotalPrice": 49.97999999999999},
{"InvoicerNo": "539105", "CustomerID": "1", "Name": "H", "Country": "United Kingdom", "Window": "1", "Start": "2010-12-16T10:00:00.000Z", "End": "2010-12-16T10:50:00.000Z", "TotalQuantity": 49, "TotalPrice": 216.33},
{"InvoicerNo": "551443", "CustomerID": "1", "Name": "H", "Country": "United Kingdom", "Window": "1", "Start": "2011-04-04T17:30:00.000Z", "End": "2011-04-04T17:50:00.000Z", "TotalQuantity": 4, "TotalPrice": 0.8},
{"InvoicerNo": "563584", "CustomerID": "1", "Name": "H", "Country": "United Kingdom", "Window": "1", "Start": "2011-09-02T15:00:00.000Z", "End": "2011-09-02T15:10:00.000Z", "TotalQuantity": 172, "TotalPrice": 339.99999999999999},
{"InvoicerNo": "573732", "CustomerID": "1", "Name": "H", "Country": "United Kingdom", "Window": "1", "Start": "2011-10-30T17:30:00.000Z", "End": "2011-10-30T17:40:00.000Z", "TotalQuantity": 130, "TotalPrice": 78.9},
{"InvoicerNo": "566783", "CustomerID": "1", "Name": "H", "Country": "United Kingdom", "Window": "1", "Start": "2011-09-15T10:30:00.000Z", "End": "2011-09-15T10:50:00.000Z", "TotalQuantity": 89, "TotalPrice": 35.60000000000000},
{"InvoicerNo": "566697", "CustomerID": "1", "Name": "H", "Country": "United Kingdom", "Window": "1", "Start": "2011-10-05T15:30:00.000Z", "End": "2011-10-05T15:40:00.000Z", "TotalQuantity": 30, "TotalPrice": 133.00},
{"InvoicerNo": "537248", "CustomerID": "1", "Name": "H", "Country": "United Kingdom", "Window": "1", "Start": "2010-12-06T10:30:00.000Z", "End": "2010-12-06T10:40:00.000Z", "TotalQuantity": 39, "TotalPrice": 56.25},
{"InvoicerNo": "576369", "CustomerID": "1", "Name": "H", "Country": "United Kingdom", "Window": "1", "Start": "2011-11-14T18:30:00.000Z", "End": "2011-11-14T18:40:00.000Z", "TotalQuantity": 288, "TotalPrice": 0.8},
{"InvoicerNo": "548087", "CustomerID": "1", "Name": "H", "Country": "United Kingdom", "Window": "1", "Start": "2011-04-04T17:30:00.000Z", "End": "2011-04-04T17:50:00.000Z", "TotalQuantity": 462, "TotalPrice": 853.36999999999998},
{"InvoicerNo": "555725", "CustomerID": "1", "Name": "H", "Country": "United Kingdom", "Window": "1", "Start": "2011-06-06T16:20:00.000Z", "End": "2011-06-06T16:30:00.000Z", "TotalQuantity": 14, "TotalPrice": 34.0},
{"InvoicerNo": "555931", "CustomerID": "1", "Name": "H", "Country": "United Kingdom", "Window": "1", "Start": "2011-05-11T17:00:00.000Z", "End": "2011-05-11T17:10:00.000Z", "TotalQuantity": 45, "TotalPrice": 66.33},
{"InvoicerNo": "552711", "CustomerID": "1", "Name": "H", "Country": "United Kingdom", "Window": "1", "Start": "2011-05-11T17:00:00.000Z", "End": "2011-05-11T17:10:00.000Z", "TotalQuantity": 151, "TotalPrice": 361.92},
{"InvoicerNo": "535835", "CustomerID": "1", "Name": "H", "Country": "United Kingdom", "Window": "1", "Start": "2011-04-18T10:20:00.000Z", "End": "2011-04-18T10:30:00.000Z", "TotalQuantity": 7, "TotalPrice": 12.369999999999999},
{"InvoicerNo": "538215", "CustomerID": "1", "Name": "H", "Country": "United Kingdom", "Window": "1", "Start": "2010-12-16T10:30:00.000Z", "End": "2010-12-16T10:50:00.000Z", "TotalQuantity": 126, "TotalPrice": 210.74999999999999},
{"InvoicerNo": "557869", "CustomerID": "1", "Name": "H", "Country": "United Kingdom", "Window": "1", "Start": "2011-06-22T14:30:00.000Z", "End": "2011-06-22T14:40:00.000Z", "TotalQuantity": 78, "TotalPrice": 195.16},
{"InvoicerNo": "578649", "CustomerID": "1", "Name": "H", "Country": "United Kingdom", "Window": "1", "Start": "2011-11-22T10:30:00.000Z", "End": "2011-11-22T10:40:00.000Z", "TotalQuantity": 52, "TotalPrice": 187.9},
{"InvoicerNo": "535986", "CustomerID": "1", "Name": "H", "Country": "United Kingdom", "Window": "1", "Start": "2011-05-16T16:00:00.000Z", "End": "2011-05-16T16:10:00.000Z", "TotalQuantity": 24, "TotalPrice": 0.9},
{"InvoicerNo": "555888", "CustomerID": "1", "Name": "H", "Country": "United Kingdom", "Window": "1", "Start": "2011-06-02T17:00:00.000Z", "End": "2011-06-02T17:10:00.000Z", "TotalQuantity": 10, "TotalPrice": -0.5},
{"InvoicerNo": "565789", "CustomerID": "1", "Name": "H", "Country": "United Kingdom", "Window": "1", "Start": "2011-06-01T13:40:00.000Z", "End": "2011-06-01T13:50:00.000Z", "TotalQuantity": 1, "TotalPrice": 14.050000000000001},
{"InvoicerNo": "545290", "CustomerID": "1", "Name": "H", "Country": "United Kingdom", "Window": "1", "Start": "2011-03-01T12:00:00.000Z", "End": "2011-03-01T12:10:00.000Z", "TotalQuantity": 114, "TotalPrice": 210.99999999999999}

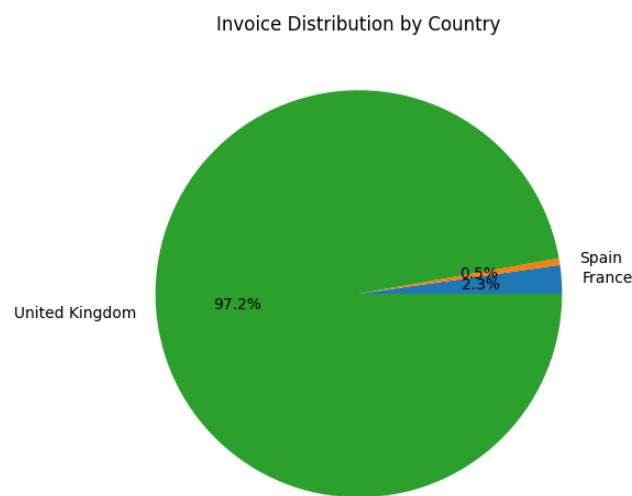
```

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."\*\*\*

