User Manual: Healthcare Big Data ETL Pipeline

1. Overview

This ETL pipeline is designed to process and analyze healthcare data, specifically utilizing the MIMIC-III Clinical Database. It leverages a Dockerized environment comprising Hadoop, Spark, and Hive to facilitate scalable data processing and analysis.

Objectives:

- Extract and preprocess data from the MIMIC-III Clinical Database.
- Transform and standardize data for analytical purposes.
- Load processed data into Hive for querying and analysis.

Technologies:

- **Docker & Docker Compose**: Containerization and orchestration.
- **Hadoop (HDFS)**: Distributed storage system.
- **Apache Spark**: Distributed data processing engine.
- **Apache Hive**: Data warehouse software for querying and managing large datasets.

2. System Requirements

Hardware:

- **CPU**: 4 cores or more
- **RAM**: 8 GB minimum (16 GB recommended)
- **Storage**: At least 20 GB free space

Software:

- **Operating System**: Linux, macOS, or Windows with Docker support
- **Docker**: Version 20.10 or higher
- **Docker Compose**: Version 1.29 or higher
- **Git**: For cloning repositories

3. Installation & Setup

A. Clone the Repository

```
git clone https://github.com/Marcel-Jan/docker-hadoop-spark.git
cd docker-hadoop-spark
```

B. Start the Docker Containers

```
docker-compose up -d
```

This command initializes the following services:

- **Hadoop HDFS**: Namenode and Datanode
- **Apache Spark**: Master and Worker nodes
- **Apache Hive**: Metastore and HiveServer2

Note: HiveServer2 may require manual startup if not configured to start automatically.

4. Accessing Services

- **Hadoop Namenode UI**: http://localhost:9870
- **Spark Master UI**: http://localhost:8080
- **HiveServer2 JDBC**: jdbc:hive2://localhost:10000

5. Data Ingestion

A. Obtain MIMIC-III Demo Dataset

1. Register and complete the required training on PhysioNet.
2. Download the MIMIC-III Clinical Database Demo v1.4 from PhysioNet.

B. Load Data into HDFS

1. Place the downloaded CSV files into a local directory, e.g., ~/mimic_data/.
2. Copy the data into HDFS:

```
docker exec -it namenode bash

hdfs dfs -mkdir /mimic

hdfs dfs -put /path/to/local/mimic_data/* /mimic/
```

6. Data Processing with Spark

A. Access Spark Shell

```
docker exec -it spark-master bash
spark-shell
```

B. Sample Data Processing

```
val df = spark.read.option("header", "true")
.csv("hdfs://namenode:9000/mimic/ADMISSIONS.csv")
df.printSchema()
df.show(5)
```

This example reads the ADMISSIONS.csv file from HDFS and displays its schema and first five rows.

7. Data Warehousing with Hive

A. Access Hive CLI

```
docker exec -it hive-server bash
hive
```

B. Create Hive Tables

```
CREATE DATABASE mimic;
USE mimic;

CREATE EXTERNAL TABLE ….
```

8. Troubleshooting

- **Docker Containers Not Starting**: Ensure no port conflicts and sufficient system resources.
- **HiveServer2 Not Running**: Manually start the service if not configured to start automatically.
- **HDFS Access Issues**: Verify that the data has been correctly uploaded to HDFS and that the paths are accurate.

9. Maintenance

- **Monitor Services**: Regularly check the UIs for Hadoop and Spark to monitor system health.
- **Data Backups**: Periodically back up HDFS data and Hive metastore.
- **Update Images**: Pull the latest Docker images to keep the environment up to date.

10. Glossary

- **ETL**: Extract, Transform, Load
- **HDFS**: Hadoop Distributed File System
- **Spark**: Open-source distributed general-purpose cluster-computing framework
- **Hive**: Data warehouse software that facilitates reading, writing, and managing large datasets
- **MIMIC-III**: Medical Information Mart for Intensive Care III, a large, publicly available database comprising de-identified health-related data