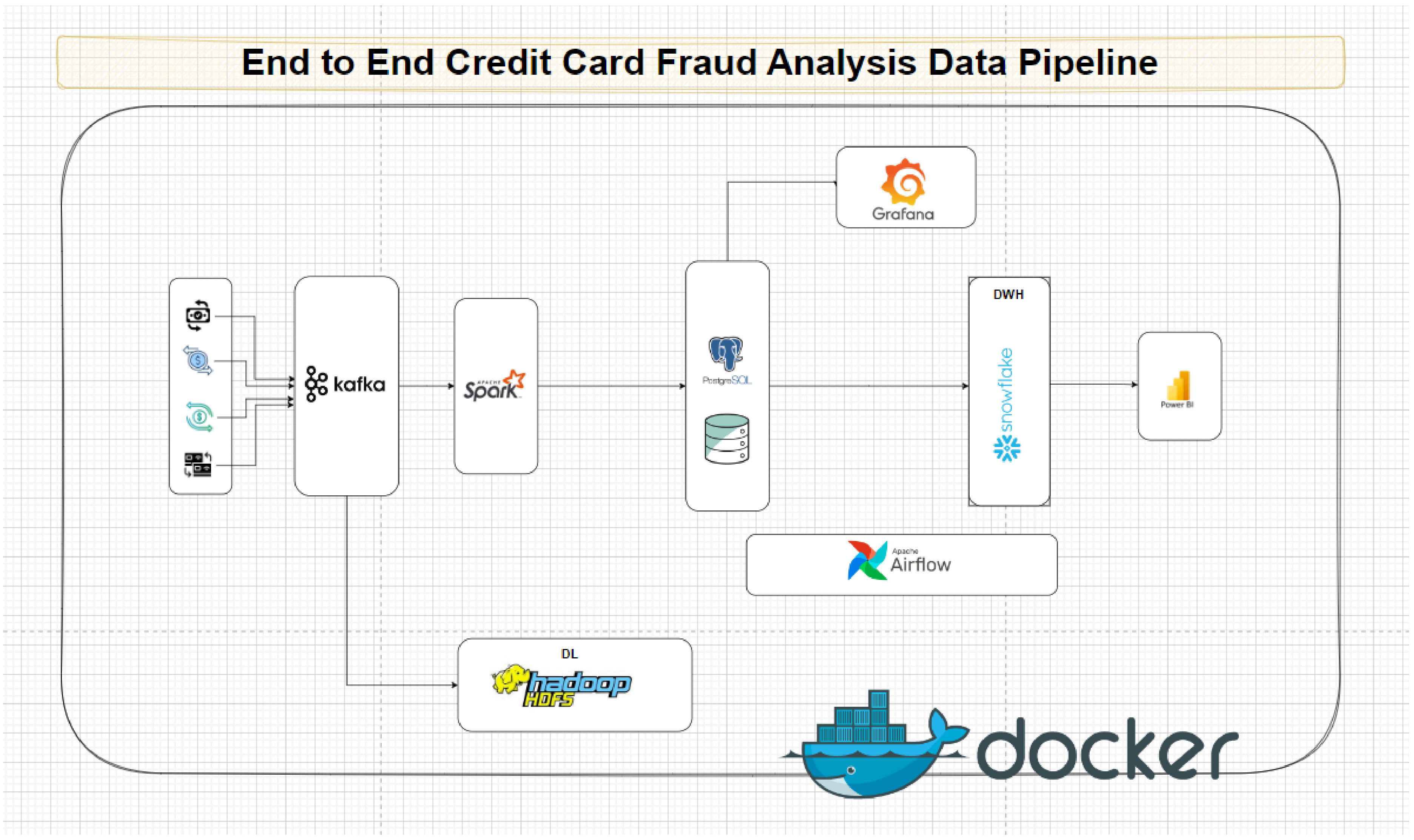
End to End Credit Card Fraud Analysis Data Pipeline

Project Overview: Credit Card Fraud Analysis Data Pipeline

Use Case: Detect fraudulent financial transactions in real time using a streaming pipeline.

Project Architecture



Work Flow

- Create Database and Data Warehouse Schema
- Create a Kafka Producer to simulate the sources streams (Point Of Sales)
- Create a unified topic called [Transactions] to receive all Credit Card transactions
- push events to Kafka topic using Kafka producer
- use spark structured streaming to transform and clean streamed data
- insert transactions to Postgres (staging layer) using spark streaming
- use Grafana to visualize streaming data
- use airflow and SQL to move data from Postgres to Snow-Flake DWH each day at mid night
- create Kafka-python consumer to store row data to HDFS acting as DL
- we can store the training data set on HDFS to train the spark ml model
- pull data from DWH to power bi to analysis it

🔷 Data **Fields :**

- index - Unique Identifier for each row
- trans_date_trans_time - Transaction DateTime
- cc_num - Credit Card Number of Customer
- merchant - Merchant Name
- category - Category of Merchant
- amt - Amount of Transaction
- first - First Name of Credit Card Holder
- last - Last Name of Credit Card Holder
- gender - Gender of Credit Card Holder
- street - Street Address of Credit Card Holder
- city - City of Credit Card Holder
- state - State of Credit Card Holder
- zip - Zip of Credit Card Holder
- lat - Latitude Location of Credit Card Holder
- long - Longitude Location of Credit Card Holder
- city_pop - Credit Card Holder's City Population
- job - Job of Credit Card Holder
- dob - Date of Birth of Credit Card Holder
- trans_num - Transaction Number
- unix_time - UNIX Time of transaction
- merch_long - Longitude Location of Merchant
- is_fraud - Fraud Flag ⟵ Target Class
- https://www.kaggle.com/datasets/kartik2112/fraud-detection?select=fraudTrain.csv

---

🔧 Tools & Technologies:

Layer	Tools
Data Ingestion	Apache Kafka
Stream Processing	Apache Spark Structured Streaming
Database	PostgreSQL
Data Warehouse	Snow-Flake
Detection Logic	Spark SQL / MLlib
Visualization	Power-BI
Orchestration	Apache Airflow
Containerization	Docker + Docker Compose

Building a Development Environment Using Docker

Docker Compose Services:

- `zookeeper`, `kafka`
- `spark-master`, `spark-worker`
- `Postgres Operational DB`, `Postgres Metadata for Airflow`
- `Name Node`, `Data Node` , `Node Manger` , `Recourse Manger`
- `airflow-webserver`, `scheduler`