**Real-Time Credit Card Fraud Detection – Practical Use Case Flow**

**Step 1: Where is the data generated?**

**Scenario:**  
Imagine this is an **E-Commerce company (e.g., Flipkart, Amazon, or Myntra)** processing **millions of real-time transactions** per day.

**User Action:**  
A customer attempts to purchase a phone for ₹45,000 using a credit card.

**Example Transaction:**

json

{

"transaction\_id": "TXN9843012",

"user\_id": "USR87231",

"card\_number": "XXXX-XXXX-XXXX-3210",

"amount": 45000,

"location": "Mumbai",

"merchant\_type": "E-Commerce",

"timestamp": "2025-08-01T10:12:54Z",

"device\_id": "DEVICE123",

"ip\_address": "203.92.42.11"

}

This **raw transaction** is emitted **instantly** by the payment gateway system, POS system, or app backend.

**Step 2: How does the system capture this data?**

**Two possibilities:**

1. **Real payment gateway integration**
   * Webhook or streaming connector from Razorpay, Stripe, etc.
   * Pushes the data to a Kafka-compatible endpoint like **Azure Event Hubs**
2. **Simulated data using NiFi or Python**
   * For prototyping, NiFi or a Python script mimics this by pushing synthetic transactions into Event Hub

**Step 3: Apache NiFi or Python – Simulating or Processing Stream**

* NiFi (Producer role) can:
  + Receive HTTP POSTs from live payment APIs
  + Or generate test data using GenerateFlowFile
  + Push data to Event Hub using PublishKafkaRecord\_2\_0

**In production**, NiFi can act as the “glue” between different sources (e.g., API, DB, logs) and the Kafka/Event Hub.

**Step 4: Event Hub (Kafka-compatible Broker)**

* Topic: ecommerce-transactions
* Buffers and streams data in real-time

**Step 5: Azure Databricks – Streaming Ingestion**

* Reads from Event Hub
* Writes raw transaction stream to Delta Lake → **Bronze Table**
* Each event = one transaction

**Step 6: Great Expectations – Validate Quality**

* Ensure amount is positive
* Timestamps are valid
* Merchant type is known, etc.

**Step 7: ML-based Fraud Detection Logic**

* Detect anomalies like:
  + Multiple large transactions in short time
  + Mismatched locations and IPs
  + Blacklisted device ID or merchant
* Predict fraud using a trained PySpark ML model

**Step 8: Store Clean Data in Delta Lake – Gold Table**

plaintext

/mnt/gold/fraud\_results

Columns:

* transaction\_id
* amount
* location
* is\_fraud (True/False)
* confidence\_score
* timestamp

**Step 9: Unity Catalog – Lineage + Access Control**

* Gold table registered in Unity Catalog
* You can trace back:

bronze → silver → gold

**Step 10: Power BI Dashboard**

* Connected to **Databricks SQL Warehouse**
* Shows:
  + Live fraud transaction list
  + Fraud distribution by region
  + Volume of fraud over time
  + % fraud by merchant type