Interview Questions - Big Data

✅ Question 1: What are the 5 V's of Big Data? Why are they important?

Answer:

The 5 V's are key characteristics that define Big Data and help guide architectural and tool decisions:

1. **Volume** – Refers to the vast amounts of data generated every second.
    - Example: Facebook generates ~4 petabytes of data per day.
    - Importance: Systems must scale horizontally to store and process such massive volumes.
2. **Velocity** – The speed at which new data is generated and must be processed.
    - Example: Real-time sensor data in autonomous vehicles.
    - Importance: Drives the need for streaming frameworks like Apache Kafka and Flink.
3. **Variety** – Data comes in different formats: structured (tables), semi-structured (JSON, XML), and unstructured (videos, logs).
    - Importance: Storage systems and pipelines must support multiple formats without strict schemas.
4. **Veracity** – The trustworthiness and quality of data.
    - Importance: Poor data quality leads to bad analytics. Validation, cleansing, and lineage tracking are essential.
5. **Value** – The potential to derive insights and make data-driven decisions.
    - Example: Netflix uses data to improve recommendations.
    - Importance: Emphasizes that storing data isn't enough—you must extract actionable insights.

👉 These five dimensions help engineers and architects understand the complexity and design requirements for big data systems.

✅ Question 5: What is the difference between batch and stream processing? Which would you use for fraud detection?

Answer:

Batch Processing:

- Processes large datasets in chunks at scheduled intervals (e.g., hourly, nightly).
- High throughput but **higher latency**.
- Examples: Hadoop MapReduce, Spark (batch mode).
- Use cases: Reporting, data aggregation, machine learning model training.

Stream Processing:

- Processes data **as it arrives** in near real-time.
- Lower latency, often used in **event-driven** or **real-time systems**.
- Examples: Apache Flink, Kafka Streams, Apache Storm.
- Use cases: Live dashboards, monitoring systems, real-time recommendations.

Fraud Detection Use Case:

- Stream processing is the better choice.
    - Fraudulent activities (e.g., unusual login patterns, suspicious payments) need **immediate action**.
    - A delay in detection could lead to financial loss or security breaches.
    - You might use Kafka to ingest transaction events, and Flink to apply detection rules in real time.

👉 *That said, batch can be used to train fraud detection models using historical data, while stream processing applies them in real time.*