Big Data Ecosystem part 2

1. Introduction to Big Data

- **Definition:** Big Data means datasets that are so **large, fast, and diverse** that traditional relational databases cannot store or process them efficiently.
- **4 V's of Big Data:**
  - **Volume** → Massive size of data (TB, PB, EB). Example: Facebook stores petabytes of user posts daily.
  - **Velocity** → Speed of data generation (real-time stock trades, IoT sensors).
  - **Variety** → Different types of data: structured (tables), semi-structured (JSON, XML), unstructured (images, videos, logs).
  - **Value** → Extracting useful insights (predicting customer behavior, fraud detection).
- **Why Big Data matters?**
  - Businesses use it for **decision making, personalization, fraud detection, recommendation engines, AI training, and healthcare predictions**.

---

2. Hadoop Distributed File System (HDFS)

- **What it is:** A **distributed storage system** designed to run on clusters of commodity hardware (cheap servers).
- **How it works:**
  - Data is split into **blocks** (default size 128MB/256MB).
  - Blocks are stored across multiple machines (**DataNodes**).
  - A **NameNode** keeps track of metadata (which block is stored where).
  - **Replication:** Each block is replicated (usually 3 copies) to avoid data loss.
- **Architecture:**
  - **NameNode (Master):** Stores file system metadata. If it fails → Secondary/Standby NameNode takes over.
  - **DataNodes (Workers):** Store actual data blocks and report to NameNode.

- **Advantages:** Fault-tolerant, scalable, cheap to expand.
- **Use Case:** Companies like LinkedIn or Twitter store huge logs and analytics data in HDFS.

---

3. Apache ZooKeeper

- **What it is:** A **centralized coordination service** for distributed applications like Hadoop, HBase, and Kafka.
- **Why needed?** Distributed systems have many nodes — ZooKeeper helps keep them **synchronized, consistent, and fault-tolerant**.
- **Functions:**
  - **Configuration Management:** Keeps cluster settings consistent across nodes.
  - **Leader Election:** Chooses a leader node automatically if the active one fails.
  - **Synchronization:** Helps multiple nodes work in coordination (like booking systems avoiding double-booking).
  - **Naming Service:** Maintains names/IDs for nodes.
- **Use Case:** In HBase, ZooKeeper helps track the master and region servers.

---

4. HBase

- **What it is:** A **NoSQL (non-relational) database** built on top of HDFS.
- **Key Characteristics:**
  - Modeled after **Google Bigtable**.
  - Stores data in **tables** with rows and columns, but columns are grouped into **Column Families**.
  - Designed for **real-time read/write access** to big datasets.
  - Can handle **billions of rows and millions of columns**.
- **Why HBase (not Hive or RDBMS)?**
  - Relational DBs fail when data is too large and schema changes frequently.

    - Hive is for batch analysis, HBase is for real-time queries.
- **Use Cases:**
    - Facebook Messenger uses HBase to store billions of messages.
    - IoT companies use it for **time-series data**.

---

5. Hive

- **What it is:** A **data warehouse tool** built on top of Hadoop.
- **Main Purpose:** Querying and analyzing large datasets stored in HDFS using a SQL-like language (**HiveQL**).
- **How it works:**
    - You write HiveQL (similar to SQL).
    - Hive converts it into **MapReduce, Tez, or Spark jobs** internally.
    - Results are stored back in HDFS.
- **Key Features:**
    - Supports **structured and semi-structured data**.
    - Provides functions like GROUP BY, JOIN, ORDER BY.
    - Good for batch processing, not real-time.
- **Use Cases:**
    - Data analysts running reports (sales trends, user activity).
    - Companies like Netflix use Hive to analyze viewing behavior.

---

6. Apache Spark

- **What it is:** A **unified big data processing engine** that is much faster than MapReduce.
- **Why fast?**
    - Uses **in-memory computation** (keeps data in RAM instead of writing intermediate results to disk like MapReduce).

- **Main Components:**
  - **Spark Core:** Basic execution engine.
  - **Spark SQL:** Run SQL queries on big data.
  - **Spark Streaming:** Process real-time data streams.
  - **MLlib:** Machine learning library.
  - **GraphX:** Graph processing (like social network analysis).
- **Advantages over MapReduce:**
  - Faster (up to 100x).
  - Supports batch, streaming, ML, and graph — all in one.
- **Use Cases:**
  - Uber uses Spark Streaming for **real-time ride matching**.
  - Banks use Spark MLlib for **fraud detection**.

---

7. MapReduce

- **What it is:** The **original programming model** in Hadoop for distributed data processing.
- **How it works (Steps):**
  1. **Map Phase:** Input data is divided into small chunks → processed in parallel → output is in (key, value) pairs.
  2. **Shuffle & Sort Phase:** System groups values by key.
  3. **Reduce Phase:** Aggregates values for each key → final result.
- **Example (Word Count):**
  - **Map:** "hello world hello" → (hello, 1), (world, 1), (hello, 1)
  - **Reduce:** (hello, [1,1]) → (hello, 2), (world, [1]) → (world, 1)
- **Limitations:**
  - Disk-based (slower than Spark).
  - Hard to program (requires Java).

- **Importance:** Even though Spark is now preferred, MapReduce introduced the foundation of **parallel data processing**.

✅ Final Quick Summary for Interviews

- **Big Data** → Huge, fast, diverse data.
- **HDFS** → Distributed storage system.
- **ZooKeeper** → Cluster coordination service.
- **HBase** → NoSQL, real-time database on HDFS.
- **Hive** → SQL-like query tool for batch analytics.
- **Spark** → Fast in-memory processing engine, supports SQL, streaming, ML.
- **MapReduce** → Original Hadoop processing model (batch, slower, disk-based).