**Big Data Ecosystem Summary**

**1. HDFS (Hadoop Distributed File System)**

* Distributed storage system for big data.
* Splits large files into blocks (default 128MB) and stores across cluster nodes.
* Provides **fault tolerance** by replicating data (default 3 copies).
* High throughput, not optimized for small files.

**2. MapReduce**

* Programming model for **parallel batch processing** of large data on clusters.
* **Map Phase** → Break task into smaller parts, process in parallel.
* **Reduce Phase** → Aggregate results and output.
* Reliable but slower compared to modern engines (e.g., Spark).

**3. Apache Spark**

* In-memory processing engine → much faster than MapReduce.
* Supports **batch, stream, machine learning, and graph processing**.
* Components:
  + **Spark SQL** → Structured data queries.
  + **Spark Streaming** → Real-time processing.
  + **MLlib** → Machine learning.
  + **GraphX** → Graph analytics.

**4. Hive**

* Data warehouse tool on top of Hadoop.
* Provides **SQL-like interface (HiveQL)** for querying big data.
* Translates queries into MapReduce or Spark jobs.
* Ideal for **batch queries, reports, and analytics**.

**5. HBase**

* **NoSQL database** built on HDFS.
* Stores **large sparse datasets** in a column-oriented way.
* Provides **real-time read/write access** to big data.
* Suitable for **random access** and unstructured data.

**6. Zookeeper**

* Centralized service for **coordination and synchronization** in distributed systems.
* Manages configuration, naming, leader election, and cluster metadata.
* Ensures **high availability** and fault tolerance.

**🔑 Quick Takeaway**

* **HDFS** → Storage.
* **MapReduce** → Batch processing.
* **Spark** → Fast, unified engine (batch + streaming).
* **Hive** → SQL queries on big data.
* **HBase** → NoSQL database for real-time access.
* **Zookeeper** → Coordination service.