Big Data Ecosystem – Beginner Introduction

1. Introduction to Big Data

- **Big Data** refers to extremely large datasets that cannot be processed or managed by traditional databases due to their **Volume (size), Velocity (speed), Variety (different formats), and Value (usefulness)**.
- Examples: social media posts, IoT sensor data, online transactions, log files.
- **Goal:** Store, process, and analyze this data to extract insights and make better decisions.

---

2. Hadoop Distributed File System (HDFS)

- **HDFS** is the **storage layer** of the Hadoop ecosystem.
- It stores huge datasets by **splitting them into blocks** (default 128MB/256MB) and distributing them across a cluster of machines.
- **Key Features:**
    - **Replication:** Each block is copied (default 3 times) to ensure fault tolerance.
    - **Scalability:** Can add more nodes easily to store more data.
    - **Master-Slave Architecture:**
        - **NameNode** – Master, stores metadata (file names, locations).
        - **DataNodes** – Slaves, actually store the data blocks.

---

3. ZooKeeper

- **ZooKeeper** is a **coordination service** used in distributed systems like Hadoop, HBase, and Kafka.
- **Why it's needed?**
    - Distributed systems have many nodes; ZooKeeper helps them **communicate, stay synchronized, and handle failures**.
- **Main Functions:**
    - Configuration management (keeps track of cluster settings).

- Leader election (chooses a master node when needed).
- Synchronization (ensures data consistency).

---

4. HBase

- **HBase** is a **NoSQL database** that runs on top of HDFS.
- It provides **real-time read and write access** to large amounts of sparse data (data with many empty values).
- **Features:**
  - Modeled after Google's Bigtable.
  - Stores data in **tables with rows and columns** (but not like traditional SQL).
  - Good for applications like messaging, IoT data, or time-series data.

---

5. Hive

- **Hive** is a **data warehouse tool** built on top of Hadoop.
- It allows users to query and analyze big datasets using **HiveQL (similar to SQL)**.
- **Key Points:**
  - Translates SQL-like queries into **MapReduce/Spark jobs** behind the scenes.
  - Great for batch processing and analytics (not real-time).
  - Used by analysts who know SQL but not Java/MapReduce.

---

6. Apache Spark

- **Spark** is a **fast, general-purpose big data processing engine**.
- It improves upon MapReduce by storing data in **memory (RAM)** for faster computation.
- **Features:**
  - Supports **batch processing, streaming, machine learning, and graph processing**.

- Much faster than MapReduce because it avoids writing intermediate results to disk.
- Provides APIs in Java, Python, Scala, and R.

---

7. MapReduce

- **MapReduce** is the **programming model** originally used in Hadoop to process big data.
- **How it works:**
  - **Map step:** Splits data into smaller tasks, processes them in parallel, and outputs key-value pairs.
  - **Reduce step:** Collects and combines results to produce the final output.
- Example: Word count program
  - **Map:** Break text into words → (word, 1)
  - **Reduce:** Sum counts for each word → (word, total count)
- Spark is now more popular, but MapReduce introduced the foundation of distributed processing.

✅ **Summary for Interviews**

- **Big Data** = handling massive datasets.
- **HDFS** = storage system.
- **ZooKeeper** = cluster coordinator.
- **HBase** = NoSQL database.
- **Hive** = SQL-like querying tool.
- **Spark** = fast processing engine.
- **MapReduce** = older batch-processing model