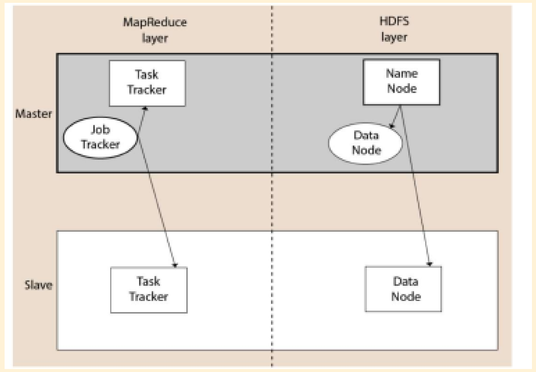


1. Modes of Hadoop Setup

Mode	Simple Meaning
Standalone Mode	Hadoop runs on **one system** (your laptop). ➜ No HDFS. ➜ No daemons (NameNode, DataNode). ➜ Used for **learning, testing, debugging**. ➜ **Fastest** mode because everything is local. **Example:** Student writing simple WordCount program on laptop.
Pseudo-Distributed Mode	Hadoop runs **all daemons separately** on **one system**. ➜ Simulates a mini cluster. ➜ Namenode, Datanode, Resource Manager all run as separate Java processes. **Example:** Developer testing small jobs on laptop before real deployment.
Fully-Distributed Mode	Hadoop runs on **multiple machines** (real cluster). ➜ Some computers act as **Master** (NameNode). ➜ Others act as **Slave** (DataNode). **Example:** Netflix running Hadoop on 1000+ machines to process video data.

2. Setting Up Hadoop in the Cloud

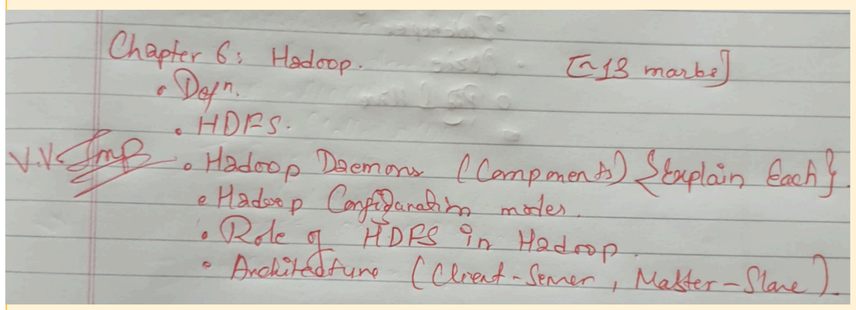
✅ Yes, we can run Hadoop in the cloud!

➡️ Use cloud services like **Amazon EMR**, **Google Cloud Dataproc**, or **Azure HDInsight**.

Why use cloud?

- No need to buy servers.
- Add or remove machines easily.
- Pay only for how much you use.

Example: A startup using Amazon EMR to analyze customer data without buying any computers.



The Hadoop ecosystem is a collection of tools and technologies built around the Hadoop core components, providing a comprehensive framework for large-scale data processing and analytics. Some key components of the Hadoop ecosystem include:

(a) **HDFS (Hadoop Distributed File System):** The primary storage system used by Hadoop for distributed storage of large datasets across multiple machines.

(b) **MapReduce:** A programming model and processing engine for distributed processing of large datasets in parallel across a cluster of computers.

(c) **YARN (Yet Another Resource Negotiator):** A resource management and job scheduling framework in Hadoop that allows multiple data processing engines like MapReduce, Apache Spark, Apache Flink, etc., to run on the same Hadoop cluster.

(d) **Apache Hive:** A data warehousing infrastructure built on top of Hadoop for querying and analyzing large datasets stored in HDFS using a SQL-like language called HiveQL.

(e) **Apache HBase:** A distributed, scalable, and column-oriented NoSQL database that runs on top of Hadoop.

6

7. Compare and Contrast the different mode of Hadoop Enviroment setup. Can we step up Hadoop in the cloud? If so, how and explain the scenarios when these different mode are preferable? [4+4]

6. Briefly describe the daemons of Hadoop. Explain the role of HDFS (Hadoop Distributed File System) in Hadoop and write down the syntax for file upload, download, list and view content of file commands for HDFS. [10]

6. Explain about the configuration modes of Hadoop. Give an overview of Hadoop ecosystem. [6+6]

1. Why Hadoop is used in big Data Analysis? How is Big Data Analytics helpful in increasing business revenue? Explain a suitable example. [10]

3. Explain the respective components of HDFS and YARN. How client writes data into HDFS? Explain with a suitable block diagram. [10]

6. List down the different components installed in the hadoop cluster. Explain its workflow. How fault tolerance and scalability is handled by the hadoop cluster? [2+6+4]

6. Explain various components of Hadoop in brief. [10]

2. **Define HDFS? How client reads data from HDFS? Explain with the help of suitable block diagram.** **[10]**

2. Define DFS. How client writes data in HDFS? Explain with the help of suitable block diagram. [10]

7. Clock synchronization in DFS may be the big challenge. How this clock synchronization problem can be solved? [10]

6. a) Explain in brief five daemons of Hadoop. [8]
   b) What is the role of Hadoop Distributed File System in Hadoop? [4]

7. **What are different daemons in HADOOP cluster? Explain each in details.** **[3+7]**

6. What are the components of Hadoop? Explain each in brief. [10]

3. Define DFS. How client writes data in HDFS? Explain with help of suitable block diagram. [10]
4. Clock synchronization in DFS may be the big challenge. How this clock synchronization problem can be solved? [1+5]

7. Explain various components of Hadoop in brief. [10]

8. What are the components of the Hadoop? For a hadoop cluster with 128 MB block size, how many mappers will hadoop mapreduce form while performing mapper function on 1 GB of data. Justify with explanation. [10]

5. List out the HADOOP daemons. How HADOOP and GFS are similar interms of design architecture. [2+8]

daemons refer to the background processes or services that run on different nodes of a Hadoop cluster to perform various tasks related to distributed storage and processing of data. These daemons collectively manage the storage, processing, and coordination within the Hadoop ecosystem. Each daemon serves a specific purpose and contributes to the overall functioning of the Hadoop cluster. The daemons typically run continuously in the background to ensure the smooth operation of the Hadoop cluster.

1. **NameNode**: The **NameNode** serves as the master node in the Hadoop Distributed File System (HDFS). Its primary responsibility lies in managing metadata, which includes crucial information about the data stored across the Hadoop cluster. This metadata encompasses details such as the location of data blocks within DataNodes, file block division, and system performance metrics. Essentially, the NameNode directs DataNode daemons, which reside on slave nodes, to handle low-level I/O tasks. However, it's important to note that the NameNode represents a single point of failure within the system due to its critical role in coordinating data storage and retrieval operations.

2. **Secondary NameNode**: Complementing the NameNode, the **Secondary NameNode** acts as a backup, safeguarding the metadata stored by the NameNode. It continuously reads metadata from the NameNode's RAM and archives it onto the disk, ensuring that in the event of a system failure or crash, this backed-up metadata can be utilized to restore the system's state and facilitate the creation of a new master node.

3. **DataNode**: On the other hand, the **DataNode** is responsible for storing the actual data within the Hadoop cluster. It operates as a daemon on slave nodes, managing the storage and retrieval of data blocks as instructed by the NameNode. Each DataNode stores data blocks locally on disk and communicates directly with clients, facilitating read and write operations. Additionally, DataNodes play a crucial role in data replication and report back to the NameNode regarding any local changes or updates.

4. **JobTracker (Resource Manager):** Moving beyond storage, the **Job Tracker** (also known as the Resource Manager) acts as the central coordinator for MapReduce jobs within the Hadoop cluster. It oversees the allocation of resources to applications running on the cluster, creating and managing jobs, and monitoring their execution. The Job Tracker, residing on the NameNode, assigns tasks to TaskTrackers, and in case of task failures, it redistributes tasks across the cluster for efficient execution.

5. **TaskTracker (Node Manager)**: Finally, the **Task Tracker** (Node Manager) operates as a slave node component within the MapReduce layer. Task Trackers manage the execution of individual MapReduce tasks on slave nodes, reporting task statuses back to the Job Tracker. In the event of a task failure, the Job Tracker reschedules the task for execution. Each Task Tracker spawns a separate JVM process to ensure fault tolerance and manages memory resources within the node.