**Rishabh Johri**

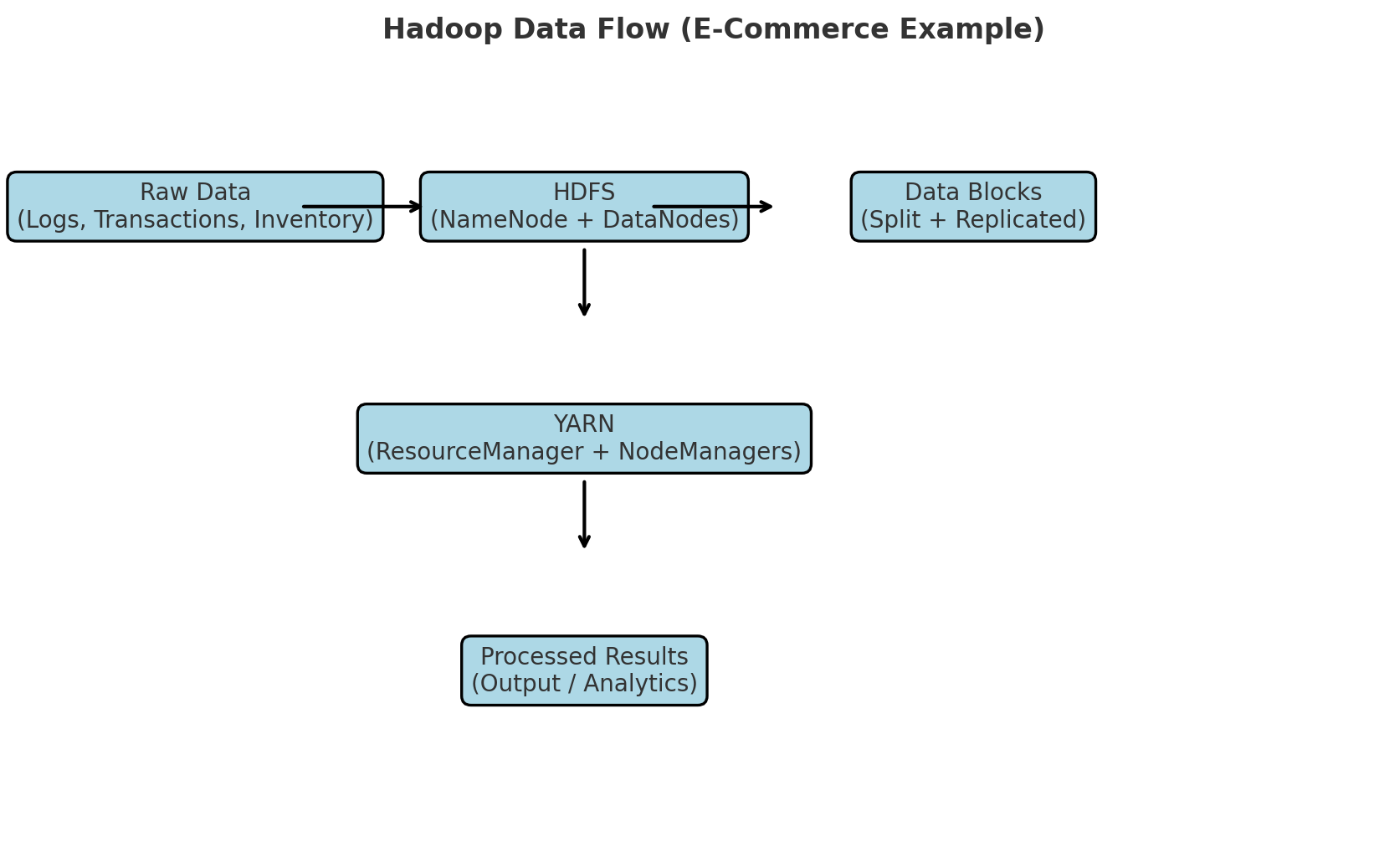
[**rjohri@deloitte.com**](mailto:rjohri@deloitte.com)

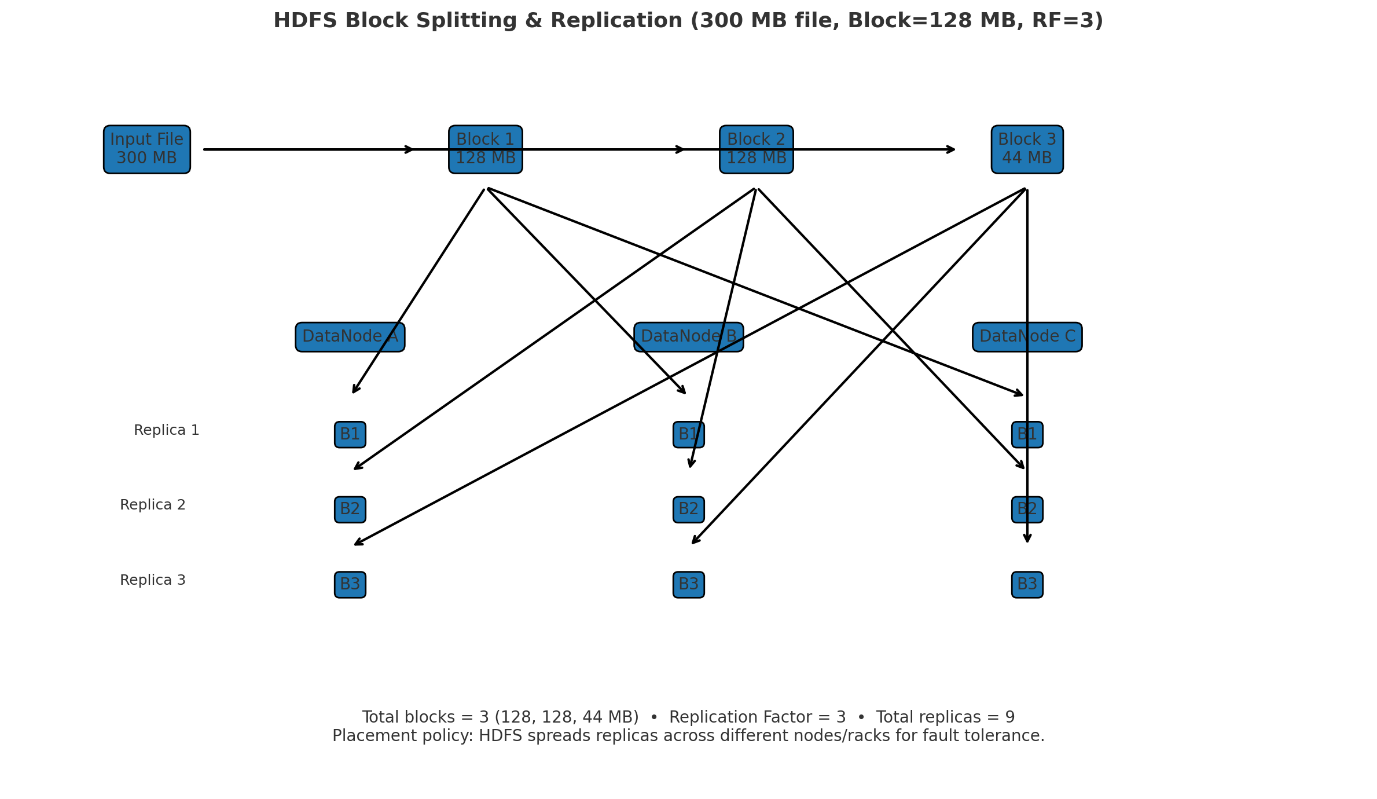
**1. Map Hadoop Components to Responsibilities**

* **File storage** -> **HDFS (NameNode + DataNodes)**
* **Metadata management** ->**NameNode** (stores file system namespace, metadata about blocks, permissions)
* **Cluster resource allocation** -> **ResourceManager** (manages YARN cluster resources)
* **Task execution** -> **NodeManager** (executes tasks on worker nodes)
* **Job tracking/history** -> **JobHistory Server** (keeps logs of completed jobs)
* **Backup metadata** -> **Secondary NameNode** (periodically merges edit logs with fsimage; helps in recovery)

**2. Data Flow in Hadoop**

1. **Data Input**:
   * Raw data (logs, transactions, inventory) ingested into **HDFS**.
2. **Splitting into Blocks**:
   * Files are split into fixed-size blocks (default **128 MB**).
3. **Storage & Replication**:
   * Each block is stored on **DataNodes**, with replication factor (default **3**).
   * Metadata (which block is where) is stored in **NameNode**.
4. **Processing via YARN**:
   * **ResourceManager** decides which node runs which task.
   * **NodeManagers** execute map/reduce or Spark jobs on local blocks.
5. **Output**:
   * Processed results written back to HDFS or external systems.





**3. Replication Factor Example**

* File size = **300 MB**
* Block size = **128 MB**
* Blocks created = 3
  + Block1 = 128 MB
  + Block2 = 128 MB
  + Block3 = 44 MB

With replication factor = 3:

* Total stored = **(3 blocks × 3 copies) = 9 block replicas** distributed across DataNodes.

**4. Failure Case**

* If **NameNode fails** -> cluster becomes unusable (metadata lost).
* **Secondary NameNode / Standby NameNode** helps recover by maintaining updated metadata snapshots.

**5. Analogy (Library System)**

* **Books = Data blocks**
* **Librarian = NameNode** (knows where every book is stored)
* **Assistants = DataNodes** (actually store and bring books)
* **Library manager = ResourceManager** (decides how assistants work)
* **Task supervisors = NodeManagers** (make sure assistants do their job)
* **Record keeper = JobHistory Server** (keeps logs of finished work)