Hadoop

- An open source framework to handle massive amount of data in a distributed and scalable ways

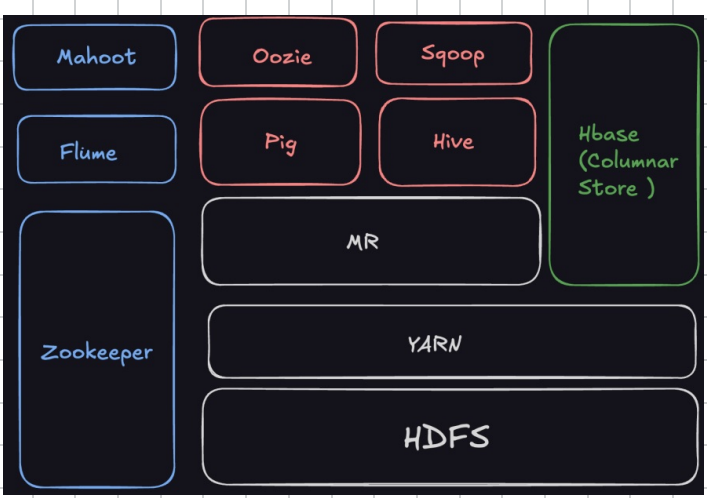
GFS → storing
MapReduce → Processing

- 2 tasks → massive data storage
  faster parallel processing

* Properties

- Scalability : Hadoop can scale horizontally
- Fault tolerance : Hadoop maintains copies / replicas to avoid failure if any single machine failed
- Distributed processing : Hadoop can process the data where it is stored
- Cost effectiveness : Inexpensive hardware / commodity machines can be used
- open source : Free to use and modify

* Hadoop Ecosystem



* Note : Hadoop properties
- Loosely coupled framework: we can remove components and it's still going to work
- Integration : can be connected to other frameworks easily

HDFS - distributed storage
MapReduce - divide into smaller tasks
YARN - decouple the resource management

Hive — Query engine (not a database)
Abstract MR by translating SQL queries into MR jobs

Pig — Abstraction of MR (MR performance but no Java/SQL)
High level scripting language for MR

Sqoop — Facilitate import/export between Hadoop and Relational DTB

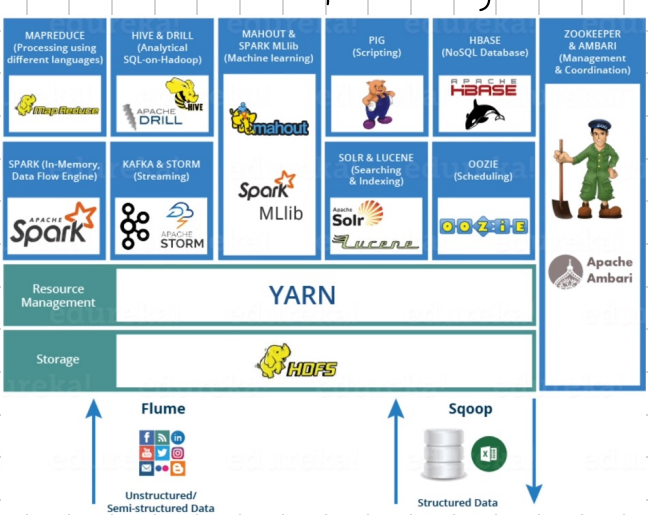
Oozie — Abstraction of MR
Use xml files for scheduling and automating our work
→ essential for scheduling complex workflows in ETL

Hbase — NoSQL DTB which allows real time read and write on HDFS

Mahout — DS component
provide ML libraries

Flume — Messaging queue
Collect logs or event data from various sources and deliver them to Hadoop and Hbase
Real-time analytics for monitoring and ingesting streaming data

Zookeeper — Coordinate distributed system to maintain consistency
Critical for ensuring reliability in Hadoop clusters



Storage: HDFS, Hbase
Processing: MR, Pig, Hive, Spark
Data ingestion: Flume, Sqoop
Coordination: Zookeeper
Workflow management: Oozie

Hadoop architecture

* Terminologies

1. File system : layer between software and hardware
   - Standalone
   - Distributed

2. Block - smallest unit of data storage in a file system

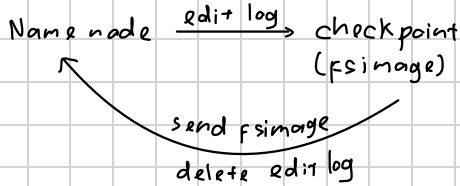
3. Process and Daemon Process
   - Process → program in execution
   - Daemon Process → a background process run without any user's intervention

HDFS architecture

Data node failure

	Temporary	Permanent
Cause	Network, software crashes,..	Hardware failures, disk corruption, decommissioning
Detection	Datanode stops sending signals (unavailable)	... for extended period (mark as dead)
Replication trigger	Failed blocks are marked as "under replicated", new replicas are created	... as "lost", new replicas are created

Secondary name node



Standby name node

- Real-time synchronizes metadata from the active namenode
- Take over if the active namenode fails, ensuring zero downtime