**BIG DATA-HADOOP**

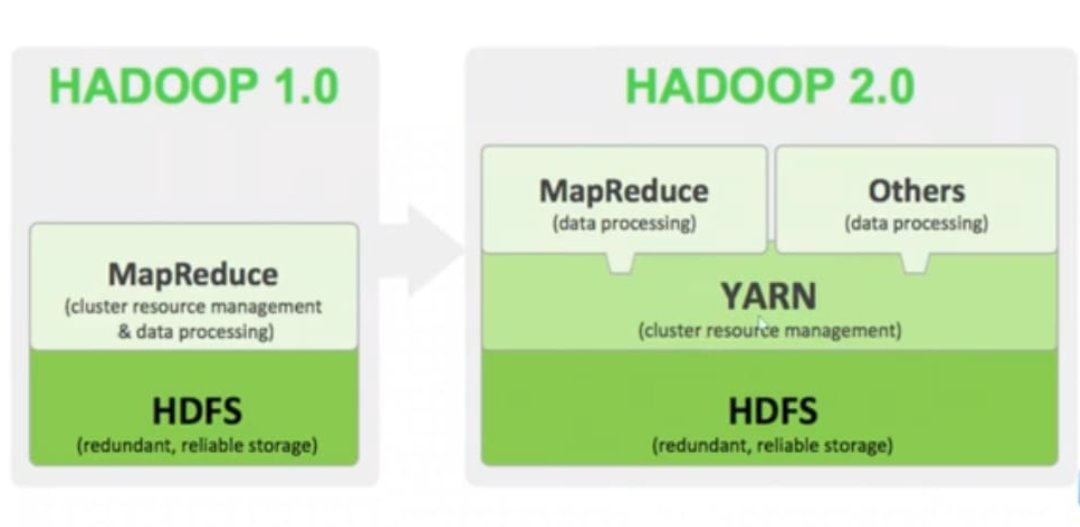
* Map reduce – Data processing + resource management
* HDFS – (Hadoop distubuted file system) – Data storage
* Yarn – (yet another resource negotiator):resource manager

**Hadoop 1.x**

* Map reduce – Data processing + resource management
* HDFS – (Hadoop distubuted file system) – Data storage

**Hadoop 2.x:**

* Map reduce – Data processing + resource management
* HDFS – (Hadoop distubuted file system) – Data storage
* Yarn – (yet another resource negotiator):resource manager



* **Hadoop supported files:**
* Json
* Csv
* Txt
* Xml
* Parquet
* Orc
* Sequence file
* Photos
* Pdf’s
* Table
* Database
* Social media
* **Features of Hadoop:**

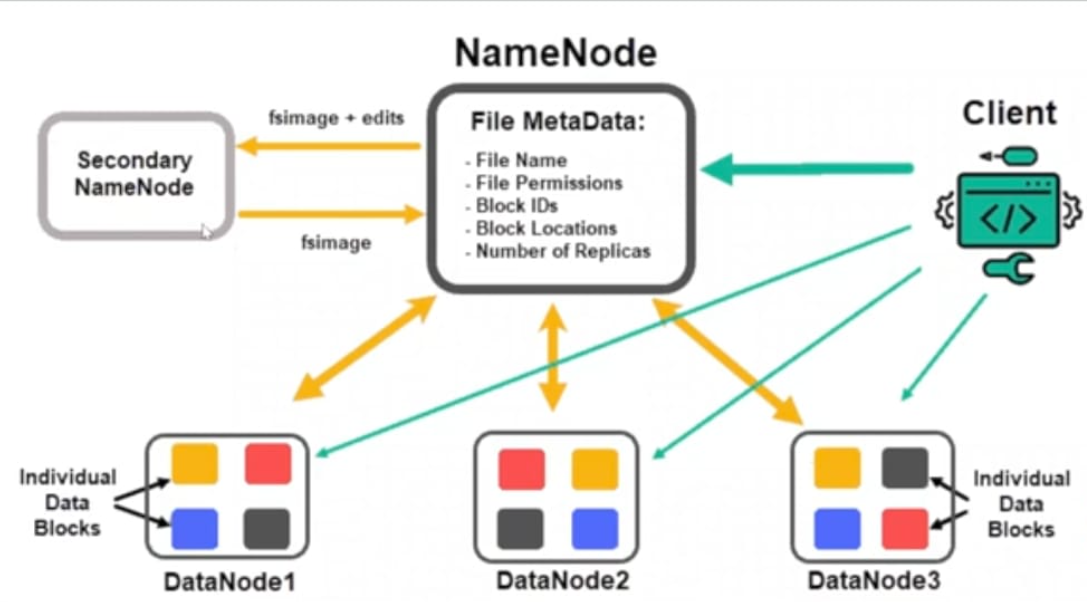
1. Reliable: handle failure. (Data replication)
2. Flexible: add more systems without down time
3. Economical: Commercial H/W used in cheap
4. Stable: reliable + consistency of the system, it will work without any expected error/failures

* **HDFS:**
* It is designed to store and manage large datasets/files across clusters.
* It is core component of the Hadoop eco system.
* It is a responsible for providing reliable and fault tolerance storage for big data applications.

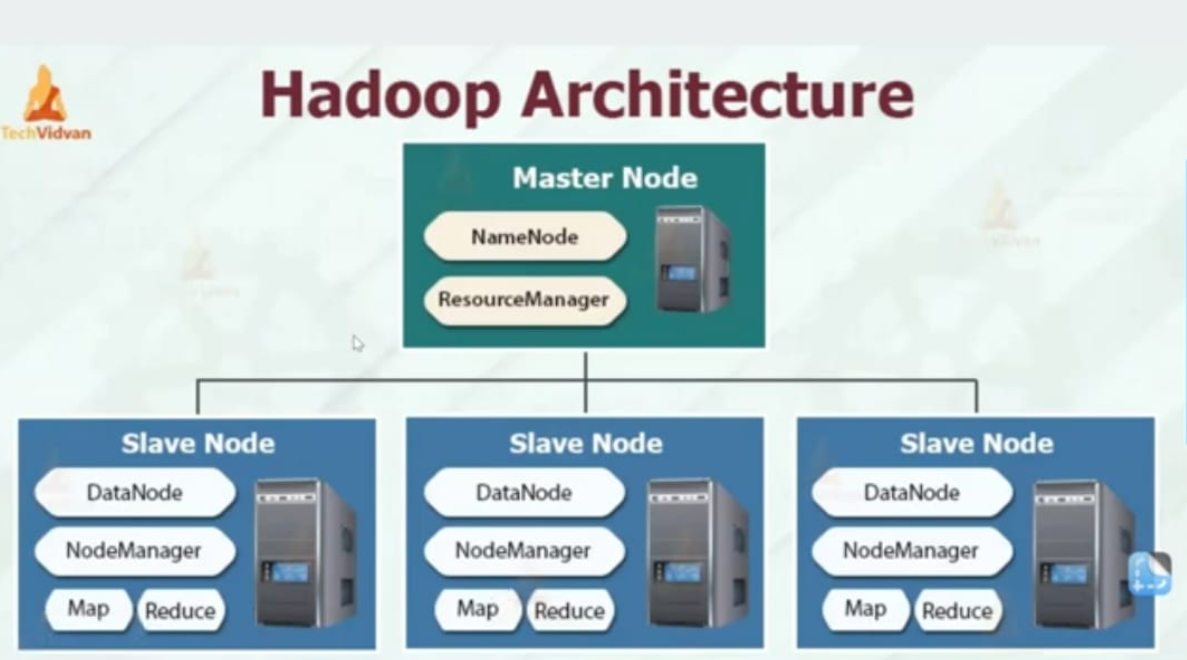
**Components:**

* Name node: Name node stores only metadata information

1. E.g. : actual metadata size 10 MB
2. Metadata: data about data
   * 1. File\_name : image
     2. File\_type: JPG
     3. File size: 10mb
     4. Storage\_location: pictures/images.jpg



* Data node: data node stores actual data



**20-12-2024**

* **Blocks:** any kind of data is stored in block wise in HDFS.(HDFS-site.xml)
* Hard disk:4 kb
* HDFS: 128 mb
* Image.jpg : 10mb
* **Replication:**
* Property in configuration file :3. It is possible to increase or decrease.(HDFS-site.xml)
* **Heart beats:**
* Data node sends heartbeats to name node every 3 seconds. Then name node knows that data nodes are available.
* **We can connect HDFS storage two ways:**
* 1.using cli hdfs commands
* 2.web browser
* **HDFS commands:**
* Ls – ltr
* Hdfs dfs -ls /
* Hdfs dfs -ls / user/
* Hdfs dfs -ls / user/cloudera/ -----------------------------------ls command (no files in cloudera)
* Hdfs dfs -chmod 777 /user/cloudera/honey----------------chmod 777 honey
* Folder creation in hdfs:
* Hdfs dfs -mkdir/user/cloudera/honey-----------------------mkdir honey(create folder)
* Hdfs dfs -rm -r /user/cloudera/honey---------------------------------rm-r honey
* Copy: cp linux\_source\_location linux\_target\_location
* Hdfs dfs -put local\_file\_system(lfs) Hadoop\_distrubuted\_file\_system(hdfs)
* Hdfs dfs -copyfromlocal local\_file\_system(lfs) Hadoop\_distrubuted\_file\_system(hdfs)
* Hdfs dfs -put file\_name.txt /user/cloudera
* Hdfs dfs -get local\_file\_system(lfs) Hadoop\_distrubuted\_file\_system(hdfs)
* Hdfs dfs -get /user/cloudera/ file\_name.txt
* Cat file\_name.txt
* Clear
* Hdfs dfs -CP /user/cloudera/ file\_name.txt /user/----------------------copy file command
* Hdfs dfs -mv /user/ file\_name.txt /user/hdfs/-----------------------------move file command
* Hdfs dfs -rm /user/ file\_name.txt /user/hdfs/-----------------------------remove file command
* Hdfs dfs -cat /user/cloudera/ file\_name.txt---------------------------create file command
* Hdfs dfs -chgrp Hadoop /user/cloudera/ file\_name.txt