**Why Bigdata ?**

**Tools /products of Bigdata ?**

Hadoop

Spark

Snowflake

Redshift

Azure /AWS Databricks

Azure Synapse

Kafka

**Features of Bigdata :**

Horizontal scalability

Vertical scalability

Distributed storage

Massive parallel processing (MPP)

Replication

Fault tolerance

High availability

Low latency

========================================

Hadoop

Opensource framework , Java

Dough cutting

Data storage GFS -Google File System

Data processing -Mapreduce

Hadoop →

HDFS --> Hadoop Distributed File system

Mapreduce →

Daemons :

DFS -Data storage

1.Namenode

a. master node

B. Co-ordinates all the datanodes

C. Stores only the metadata

D. only one namenode in a cluster

E. High Availability (HA)

1. Active namenode and standby namenode

2. Datanode

---Actual data store

-1000s of datanode

Each datanode every 3 seconds sends a tcp signal to the namenode → Heartbeat

After 10 mins , that unresponsive data node is announced as a dead node

3. Secondary namenode

→ not a hot backup

→

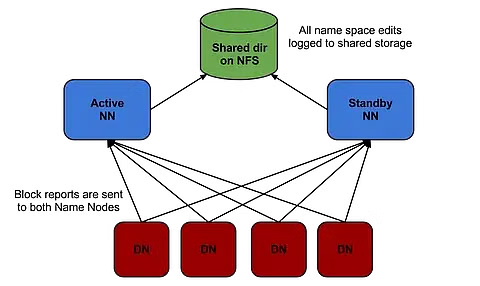
**YARN (Yet Another Resource Negotiator) → Data processing**

4. Resource manager

5.Node manager

Hadoop setup:

1. Single node cluster -- all 5 daemons runs in a same machine
2. Pseudo distributed environment
3. Multinode cluster -



**Hadoop Distribution**

1.Cloudera

2. Horton works

3. MapR

4.MS HDinsight

5.IBM Biginsights

6.AWS EMR

=========================================

HDFS ? SDR

C: sales.csv 300 mb D:

1. Sharding --block size 128mb -by default

A file is splitted into blocks/chunk/partition/piece

B1 -128mb

B2 --128 mb

B3 -44 mb

30 tb ⇒ block size → 256m,1g 128g, 512m

1. Distributed

Each blocks are distributed across the cluster

1. Replication factor → Fault tolerance

Each blocks are replicated among the cluster

3 times → by default

Max → 512 times

Replication → setrep → increase or decrease replication

1 gb ===> 3 gb

D1 b1 b3 d2 b1 b3 d3 b2

d4b2 d5b1 b3 d6 b3 b2

###to start the daemons

start-dfs.sh

Start-yarn,sh

## Open the web browser for hdfs web console and type

**localhost:50070**

hadoop fs -mkdir day1

WORM → Write Once Read Many

**/usr/local/hadoop-2.9.1/etc/hadoop/hdfs-site.xml**

Exercise Question:

Objective:

Create a directory structure in HDFS.

Upload a local file to HDFS.

List the contents of the HDFS directory.

Delete a file from HDFS.

Instructions:

Create a directory named "exercise\_dir" in the root of HDFS.

Inside "exercise\_dir," create two subdirectories named "input" and "output."

Upload a text file (you can create a dummy text file locally) to the "input" directory in HDFS.

List the contents of the "input" directory.

Move the uploaded file from the "input" directory to the "output" directory.

Confirm that the file is now in the "output" directory.

Delete the file from the "output" directory.