

## Types of data used by flipkart

Log files ----->	Unstructured data	Daily 2 TB at-least
Transactions ----->	Structured data	~ GBs
Catalog ----->	Semi-Structured	~ 200GB
Social Media ----->	Semi and Unstructured	~ 3-4 TB
Logistics----->	Combined	~ GBs
Internal ----->	Combined	~ GBs

Problems faced :

Volume : This much huge amount of data

Variety : Various kind of data

Velocity : Speed of the data generated is very fast

Veracity : Quality of data ex. Temperature data, only numbers are not enough, Celsius/Kelvin

Value : Importance of data

## Walmart Case Study

### Beer diaper Problem

They hired a consultant which studied that on weekend people bought beer and diapers on a large scale, they studied this and started offers like, buy beer and diaper get 20% off, the sales boomed.

<-- This is called value data

Google : around 2000-2001 : started Distributed Computing

Scale-up : Vertical Scaling  
Scale-out : Horizontal Scaling

Problem with Vertical scaling :

- a. hardware limits (ex. Machine has a limit on RAM )
- b. Machine is a single point of failure

To cope-up with this : Distributed Computing

Google created a **framework** which can be installed on these machines that allowed them to talk with each other.

1. Google File System (GFS)

- It's like metadata of the files.
- Formatting the hard-disk means creating a file system.
- GFS is a distributed file system

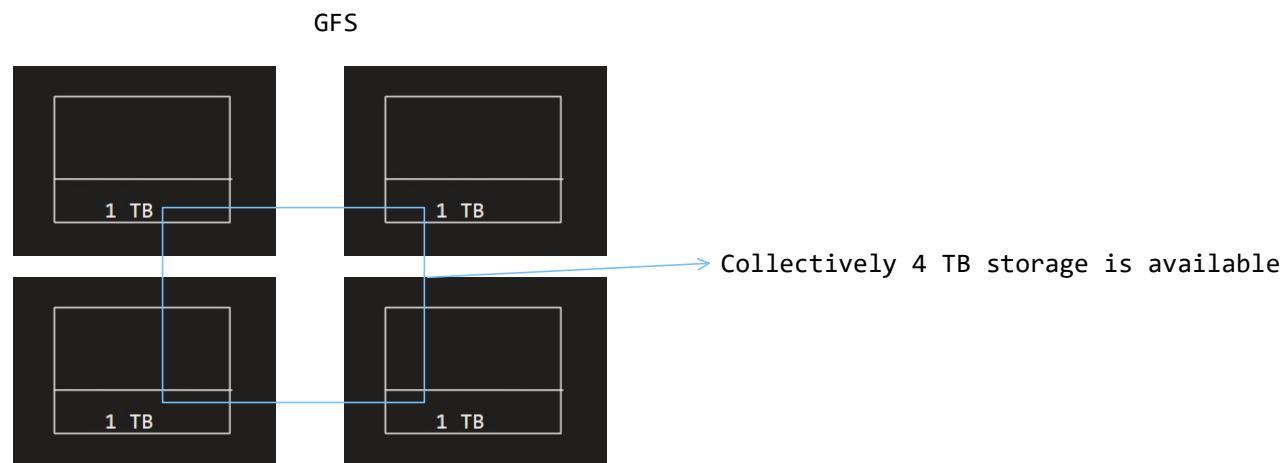


Fig: Cluster

2. MapReduce

- MapReduce is just a programming Framework. **Not** created by Google
- Implemented MapReduce in that cluster.

Google published a paper on this in 2003-4

**Doug and Mike** : their startup failed , they saw that paper published by google, got an idea, they came up with Hadoop. In 2005  
It's same as GFS.

Yahoo hired them, after some time yahoo realized Hadoop is very powerful then they made it open-source.  
It became Apache Hadoop in 1st April 2006

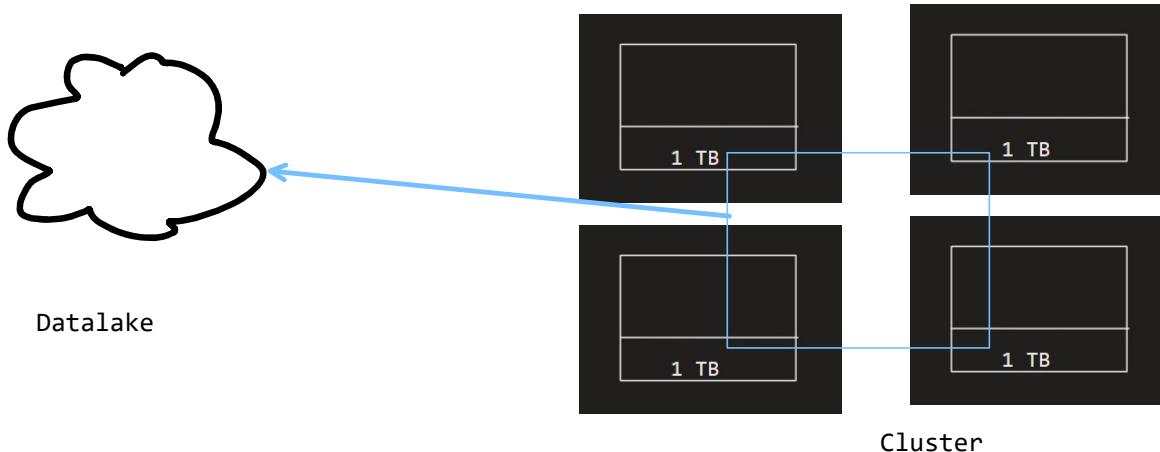
**Hadoop** is an open-source software framework for storing and processing very large datasets across clusters of computers

In starting days hadoop clusters were on premise.  
~ 2011-12 When cloud computing evolved, people started using hadoop in EC2 instances

Hadoop :      Storage      +      Resource management      +      Processing  
                  HDFS                           YARN                           MapReduce

Problem with hadoop :

When your first machine is full, then add another one then add another one ... continuously..  
Then datalake was being used, ex., aws S3.



#### Hadoop Features :

1. Open Source
2. Stores data in distributed manner.
3. Scalable.
4. Use commodity hardware
5. Fault Tolerance.
6. Availability
7. Cost Effective

#### 3 Gens of Hadoop :

1. Hadoop 1 ---> Obsolete
2. Hadoop 2 ---> Obsolete
3. Hadoop 3 ---> currently Being used

No company uses open source s/w

Because if that s/w fails who will be responsible ?

They use paid versions so they get **technical support**.

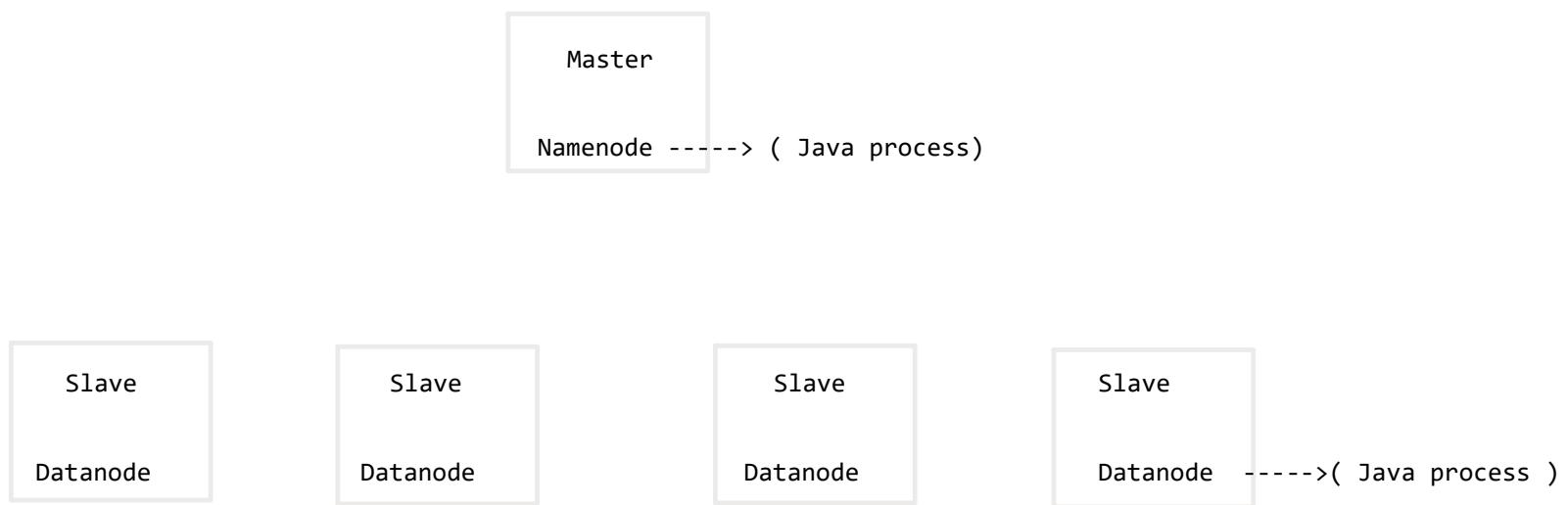
## Commercial Distributions of Hadoop :

1. Cloudera
    - i. They sell us their product which has open-source hadoop in it.
    - ii. But they take the responsibility of their product.
  2. Hortonworks ( merged with cloudera in 2018)
  3. AWS EMR
  4. Azure HDInsight
  5. GCP Dataproc
  6. IBM InfoSphere BigInsights
- 

### 1. Storage : HDFS

1. Master less architecture ex. Cassandra
2. Master Slave architecture

Hadoop was developed in java language.



Every 3 seconds, slave sends a heartbeat to master.

To store the 384 MB file in this, we have to divide the file into blocks of 128 MB  
We get 3 blocks.

Name node gives locations where the data can be copied.

Files will be distributed among the datanodes.

To prevent data loss, Replication is then done. By default 3 times

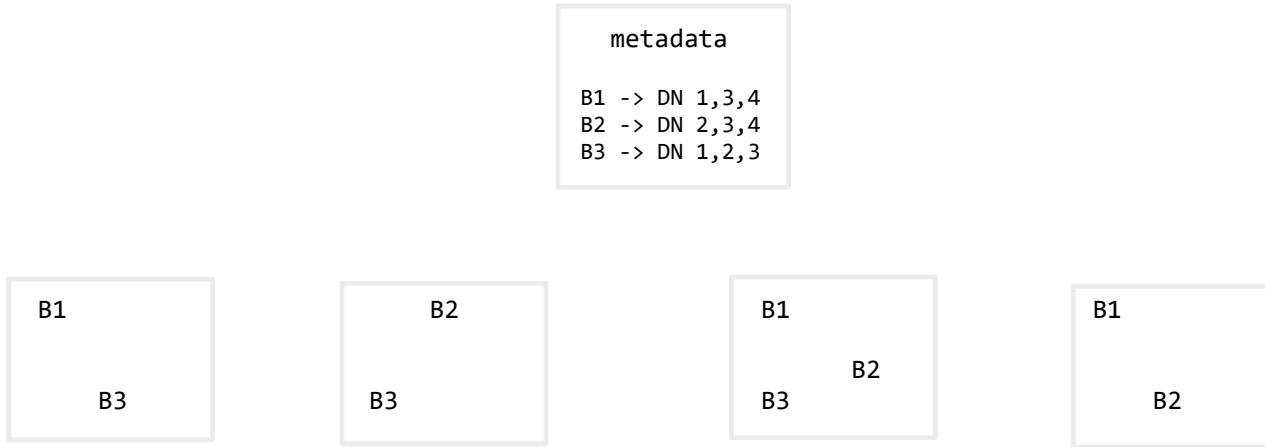
File size = 384 MB  
Block size = 128 MB  
No. of blocks = 3

**Master Node (NameNode) :**

- Stores metadata: filenames, block locations, replication
- Maintains FileSystem namespace
- No actual data stored here

**Slave Nodes (DataNodes) :**

- Store actual HDFS blocks
- Send heartbeats & block reports to NameNode



Namenode --> stores metadata ↓

Ex. Sales.csv is divided into 3 blocks..

```

B1 -> DN 1,3,4
B2 -> DN 2,3,4
B3 -> DN 1,2,3

```

If master goes down, one of the DataNodes become master.

If a DataNode is dead, now we have only 2 copied of Blocks that were in that DN,  
Now the blocks are copied in other DNS  
IF that DN comes back online, copies are reintroduced into that node, and the extra copies will be deleted

AWS EMR

AWS EMR on EC2

Steps :

1. Create cluster
2. Give a name
3. CEMR release: 5.30.0 version ( EMR release)

4. application bundle : Core hadoop.
5. Select : Hue, Tez (faster version of MapReduce), Gangila, Hadoop, Hive .....(Tez is Hindi word, means fast)
6. Cluster Configuration : uniform instance group
7. Primary : m4.large ( 2 VCPU and 8 GB RAM )
8. Core : m4.large
9. Task instance : Remove it, we don't need that.
10. Cluster scaling and provisioning: Set cluster size manually
11. Update instance(s) size to 3
12. Cluster termination and node replacement : select manually terminate the cluster
13. Cluster logs : turn off
14. Choose the keypair
15. IAM roles : EMR\_DefaultRole
16. EC2 instance profile for Amazon EMR : EMR\_EC2\_DefaultRole

3 WhatsApp SSSPICY! Play on CrazyGames Launch AWS Academy Learner Lab Properties > omsai-cluster > EMR

us-east-1.console.aws.amazon.com/emr/home?region=us-east-1#/clusterDetails/j-3NOAYQ6P18F5C

YouTube LeetCode\_To\_GitHub... ChatGPT Colab oalladwar1 - Replit Survivor.io Calculator Survivor.io Project Crack Interview Not...

aws Search [Alt+S] United States (N. Virginia) Account ID: 5614-0682-6670 vclabs/user4548958=omsaialladwar@gmail.com

Amazon EMR > EMR on EC2: Clusters > omsai-cluster

Your cluster "omsai-cluster" has been successfully created.

## omsai-cluster

Updated less than a minute ago

Terminate Clone in AWS CLI Clone

This EMR release reaches End of Support on May-01-2026 and will no longer be eligible for technical support. AWS strongly recommends that you run your workloads on the latest Amazon EMR release to receive security-critical updates and fixes. You can also use new Spark upgrade agent to modernize your existing applications on version 5.40 or higher to latest EMR version. To learn more, see [EMR Standard Support policy](#) and [Spark Upgrades](#).

### Summary

#### Cluster info

Cluster ID: j-3NOAYQ6P18F5C

Cluster ARN: arn:aws:elasticmapreduce:us-east-1:561406826670:cluster/j-3NOAYQ6P18F5C

Cluster configuration: Instance groups

Capacity: 1 Primary | 3 Core | 0 Task

#### Applications

Amazon EMR version: emr-5.30.0

Installed applications: Ganglia 3.7.2, Hadoop 2.8.5, Hive 2.3.6, Hue 4.6.0, Tez 0.9.2

#### Cluster management

Log destination in Amazon S3: Logging not configured

Primary node public DNS: ec2-3-234-212-90.compute-1.amazonaws.com

Connect to the Primary node using SSM

#### Status and time

Status: Starting

Creation time: November 17, 2025, 15:03 (UTC+05:30)

Elapsed time: 1 minute, 1 second

Properties Bootstrap actions Instances (Hardware) Steps Applications Configurations Monitoring Events Tags (0)

Cluster load [Info](#)

CloudShell Feedback

Cluster termination and node replacement [Info](#)

Edit

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

The screenshot shows the AWS EMR cluster properties page for the cluster 'omsai-cluster'. The 'Properties' tab is selected. The 'Network and security' section is highlighted with a blue oval. It contains the following details:

- Network**:
  - Virtual Private Cloud (VPC): [vpc-0757215f712452a0a](#)
  - Subnet(s) and Availability Zone(s) (AZ): [subnet-0523c6d349e6f6859](#) | us-east-1f
  - EC2 security groups (firewall)
- Security configuration**: Security configuration None
- EC2 key pair**: [traya-key-us-east-1](#)

Other sections visible include 'Cluster logs', 'Cluster termination and node replacement' (with an 'Edit' button), and 'Permissions'.

• GO to -- > EC2 Security groups :  
open sg of primary node ... ex., [sg-0b44738a530ddd834](#) - ElasticMapReduce-master  
Edit inbound rules :  
All traffic , anywhere Ipv4

SAVE RULES.

EMR serverless ? ---> problem is : we cannot expect a particular capacity.  
EMR on EC2 -----> we're using this.  
EMR on EKS -----> on virtual containers

To create new cluster again, just clone the existing (terminated can also work), remember to delete the inbound rule of all traffic from anywhere ipv4 first, then create a new one.

Steps to work :

1. Open MobaXterm
2. Session :SSH connection
3. Remote Host : Primary node public DNS ( encircled in screenshot )
4. Username : hadoop
5. Advanced SSH Settings : Use private key : upload key-pair .pem file

On aws emr page, click on applications :

Click on UI URL of Hue.  
To upload and download the data on Hadoop, we use Hue

Username : hadoop (compulsory)  
Password : Demo@123

In vertical menu, choose files  
Click on new and create a folder  
Upload a csv file into it.

HDFS is similar to S3, HDFS the first datalake in the world.

Go back to applications on aws emr :  
choose Ganglia (for monitoring) URI: it's an open source tool.

Go back to applications on aws emr :  
Click on namenode URI : summary of our namenode.  
Go to Utilities from horizontal menu.  
Select "user" from column "Name"  
Select "hadoop" from column "Name"  
Here we can see our file info, ex. Replications, etc..