



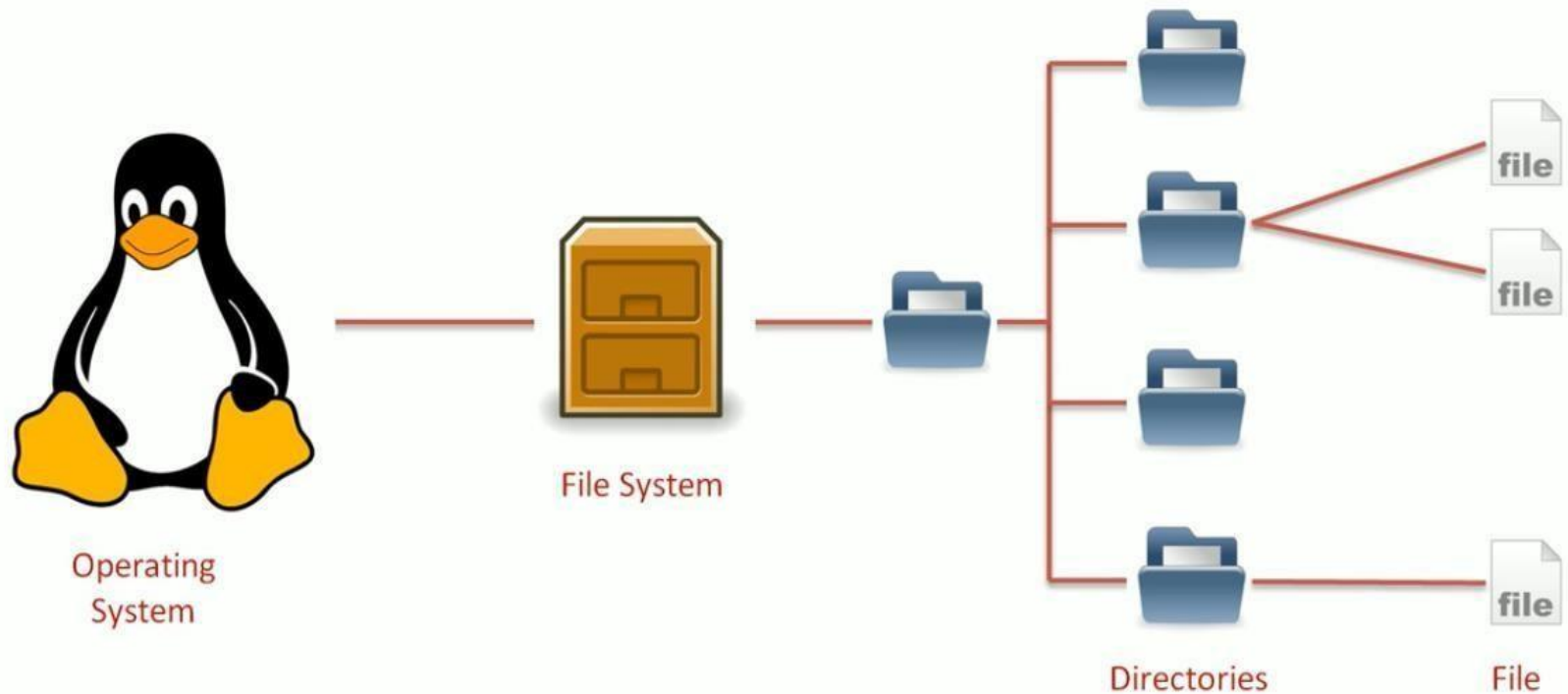
**Sohail Akhtar**  
**CS Department**  
**Bahria University, Islamabad Campus**

**Provided by:**  
**Dr Muhammad Rashid Hussain**

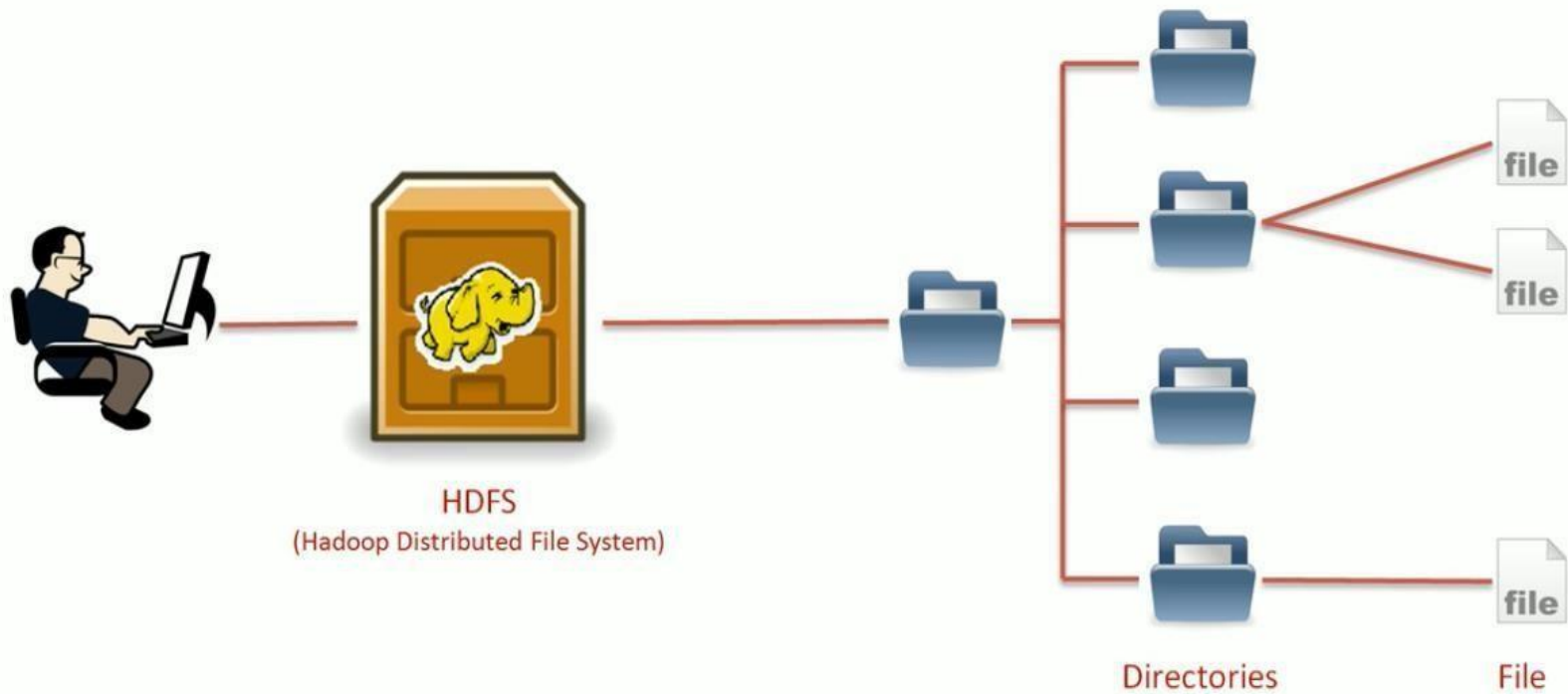
# Agenda

## Hadoop Architecture

# Typical File System

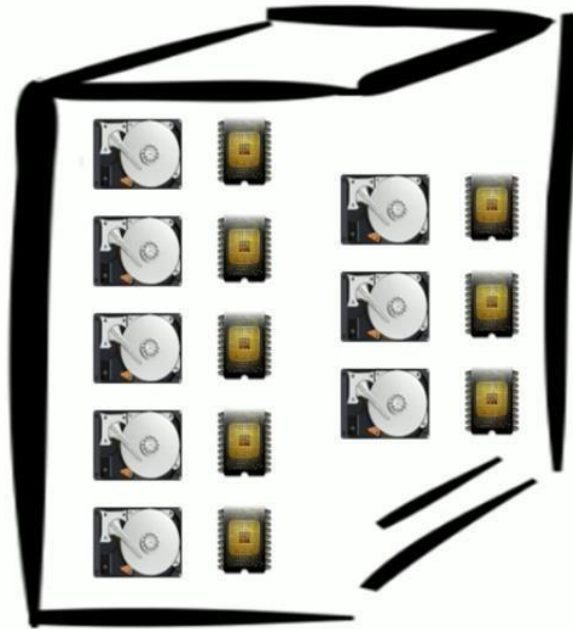


# Hadoop Distributed File System



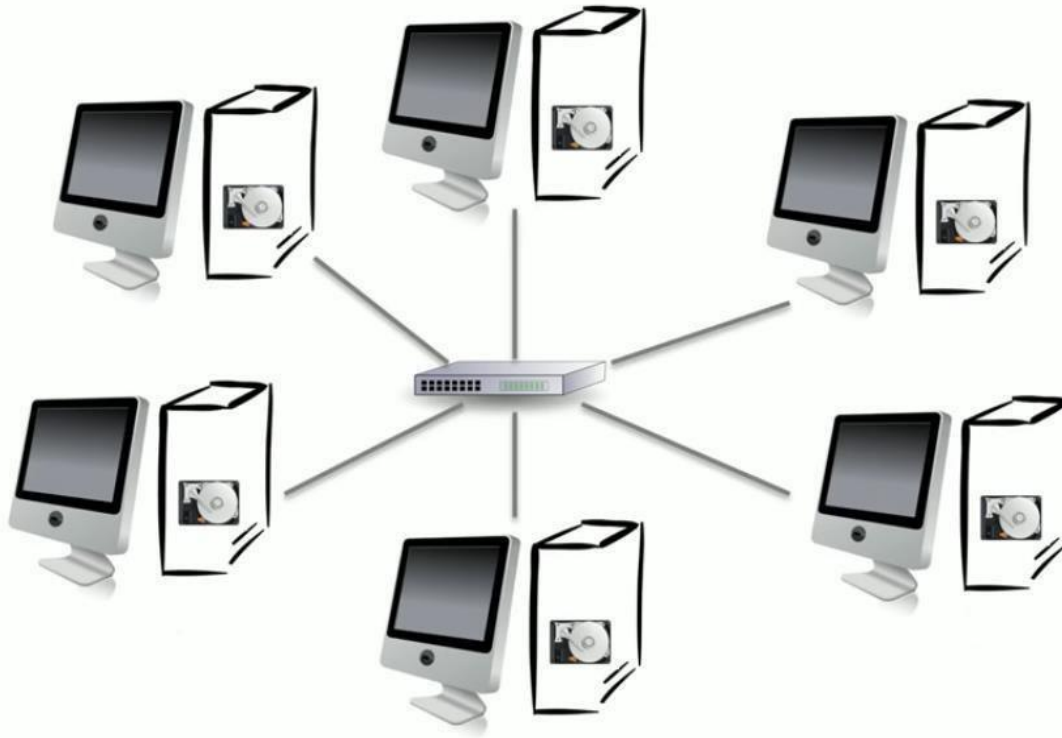
# Traditional Enhancement of Storage

## Vertical Scaling



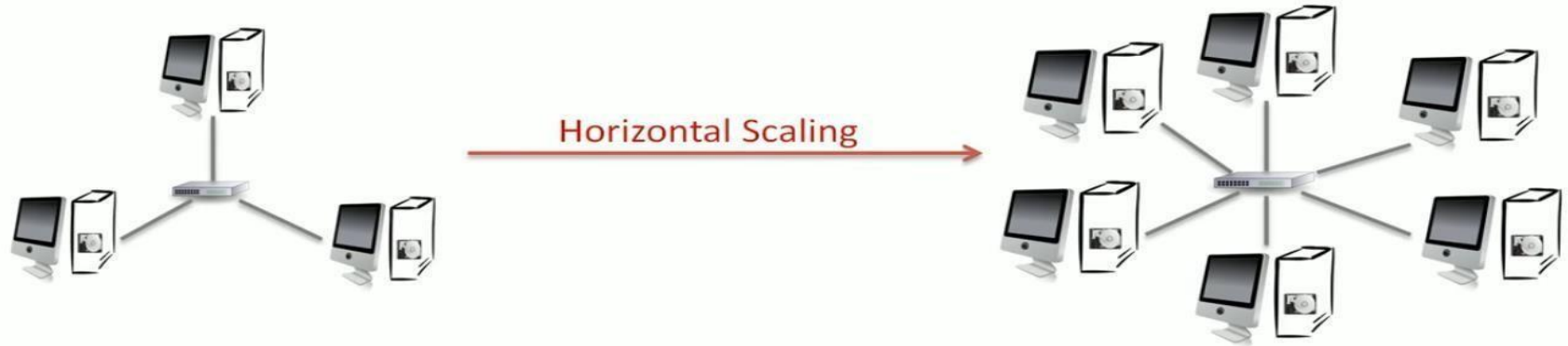
Big Data inflow is huge so not a cost effective solution anymore

# Increase Storage as you grow



Commodity Hardware with lot of up-scaling is solution

# Increase Storage as you grow

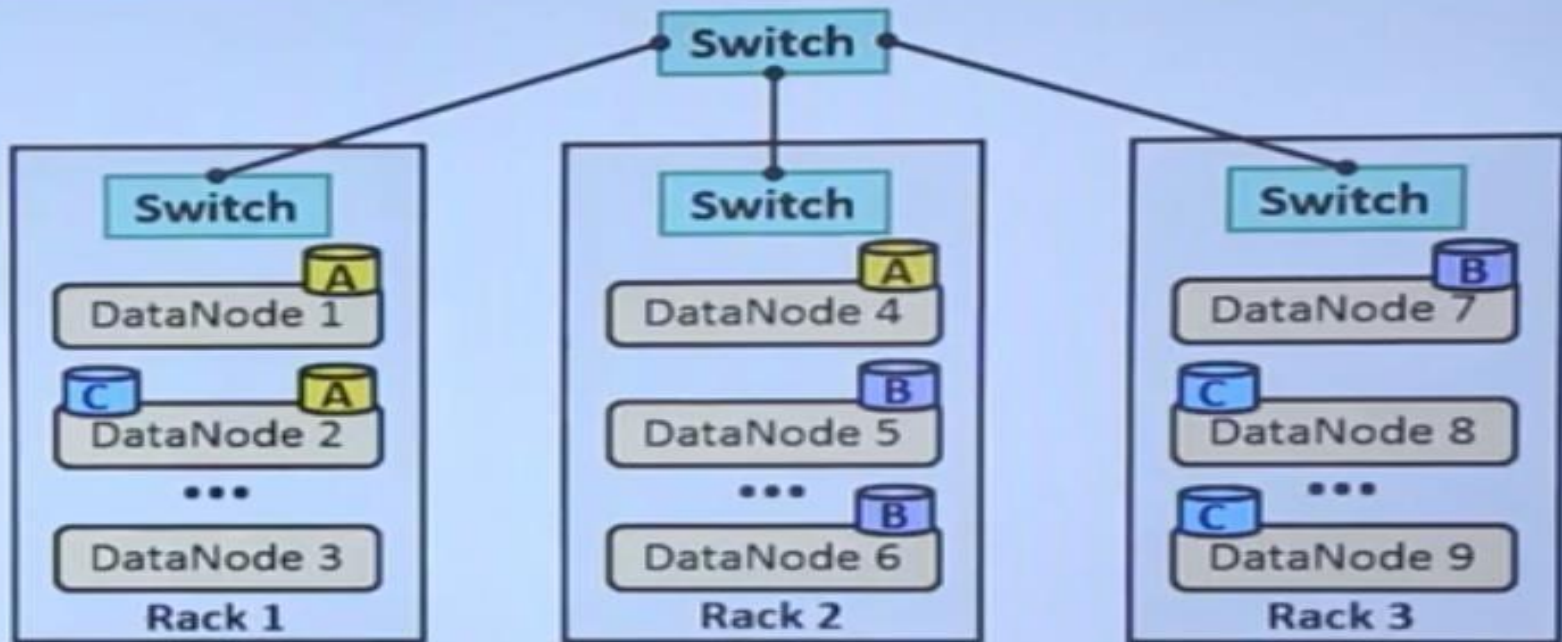


Commodity Hardware with lot of up-scaling is solution



# Hadoop

## HDFS Architecture



### ➤ What is HDFS Architecture?

- Hadoop Distributed File System (HDFS) is like Master-Worker architecture. The master is the NameNode and the workers are the low-cost commodity hardware. In the DataNodes, the actual data is stored. In this architecture, there is single NameNode and multiple DataNodes.



# Hadoop

## HDFS Architecture

### ➤ What is the task of NameNode?

- The NameNode is used to store the meta-data and another data related to DataNodes. The NameNode also responsible for:
- Managing the file-system namespace
- It controls the access of different clients into the data blocks.
- Periodically checks the availability of the DataNodes.
- It also cares about the replication factor of the data blocks.

# Hadoop

## HDFS Architecture

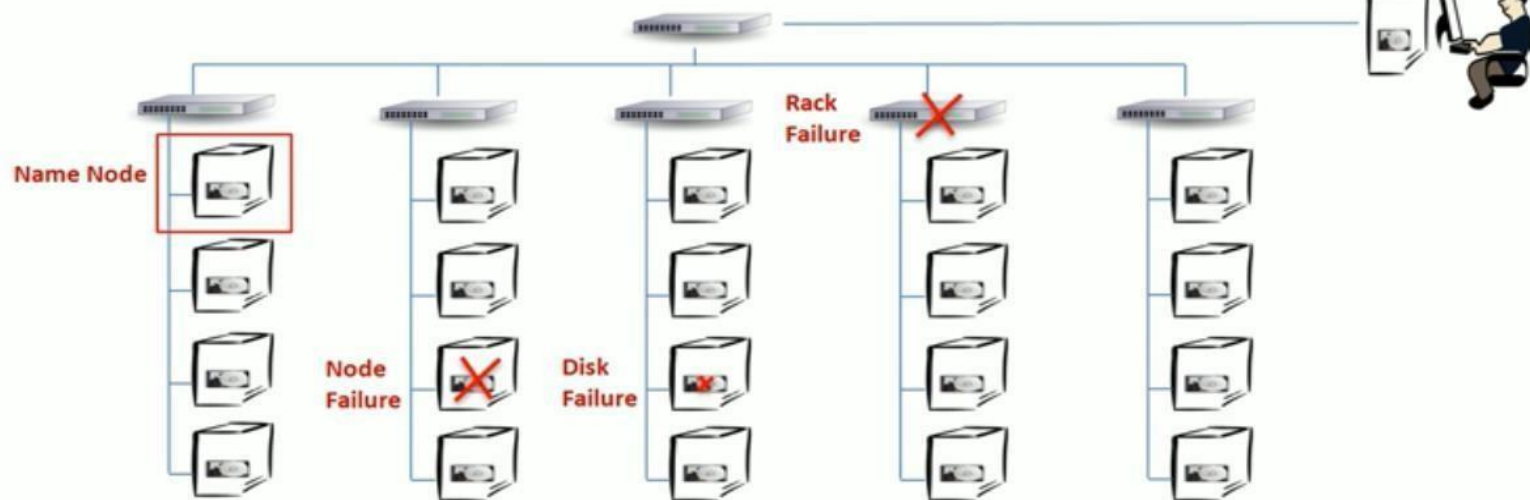
### ➤ What is the task of DataNodes?

- DataNodes are the main storages of data. Hadoop uses low-cost hardware to store data.
- DataNodes are responsible for storing, replication creating, deleting these type of jobs according to the instruction of NameNode.
- These DataNodes send the health report to the NameNode periodically. The default time is 3 seconds. So after every 3 seconds, these send the report to the NameNode.

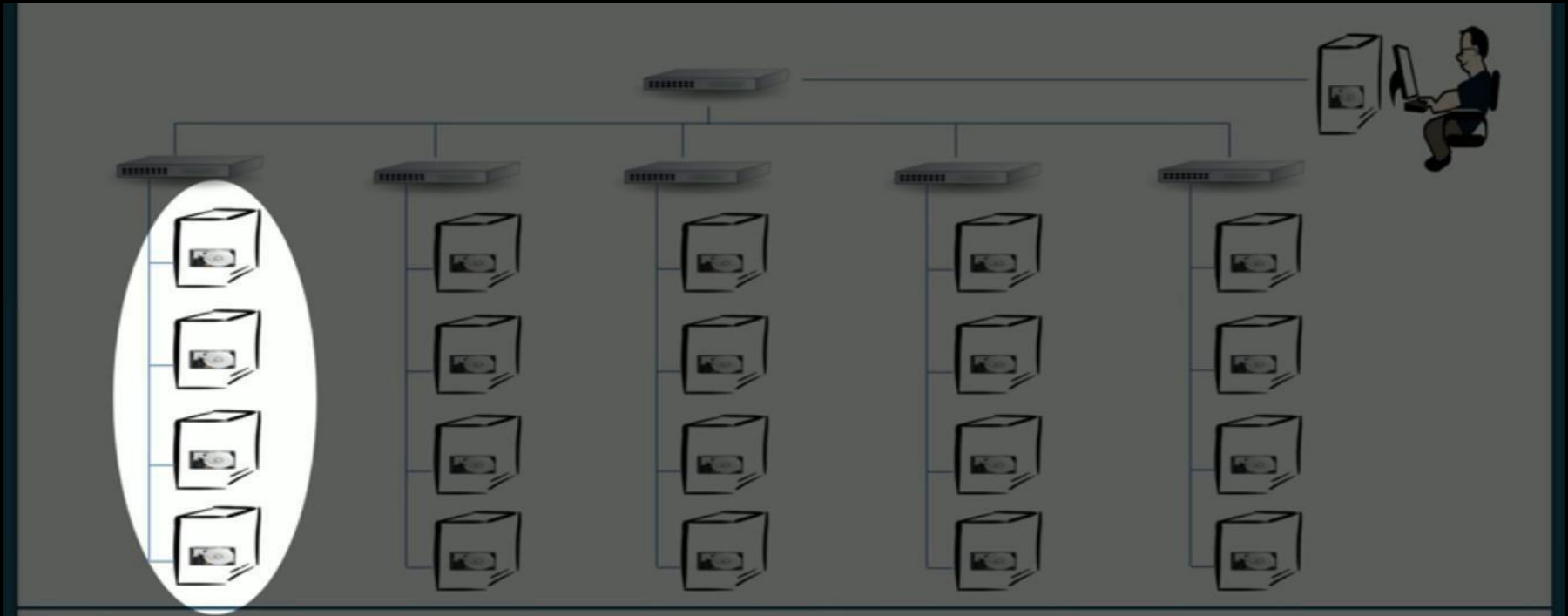
# HDFS Architecture



Hadoop Cluster



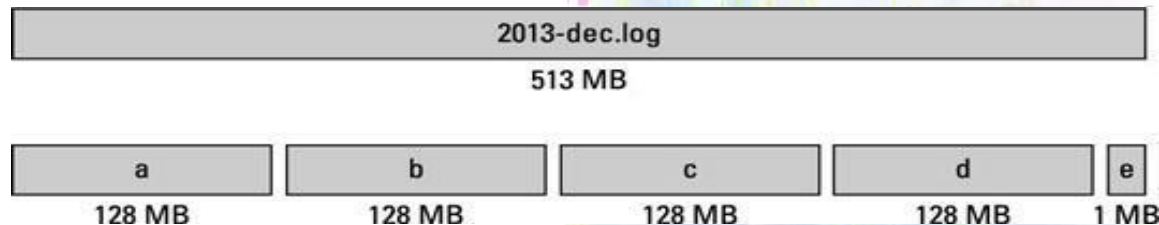
# HDFS Architecture



Rack Mounted Clusters in High Availability

# HDFS Blocks

- File is divided into blocks (default: 64MB) and duplicated in multiple places (default: 3)

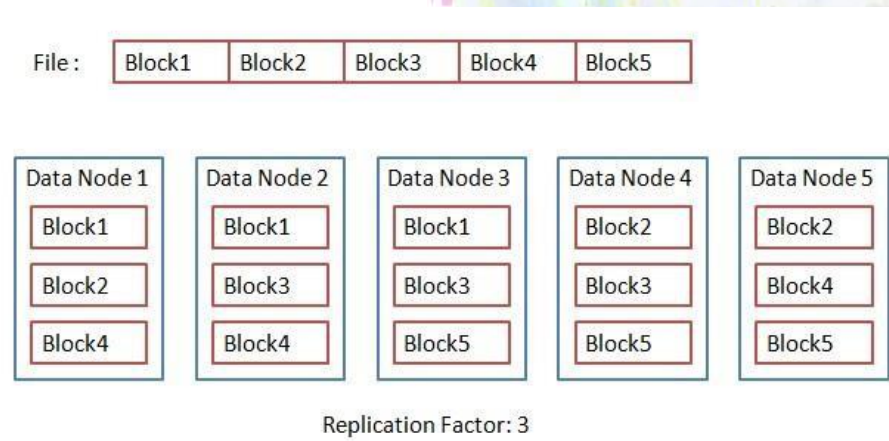


- Dividing into blocks is normal for a file system. E.g., the default block size in Linux is 4KB. The difference of HDFS is the scale.
- Hadoop was designed to operate at the petabyte scale.
- Every data block stored in HDFS has its own metadata and needs to be tracked by a central server.



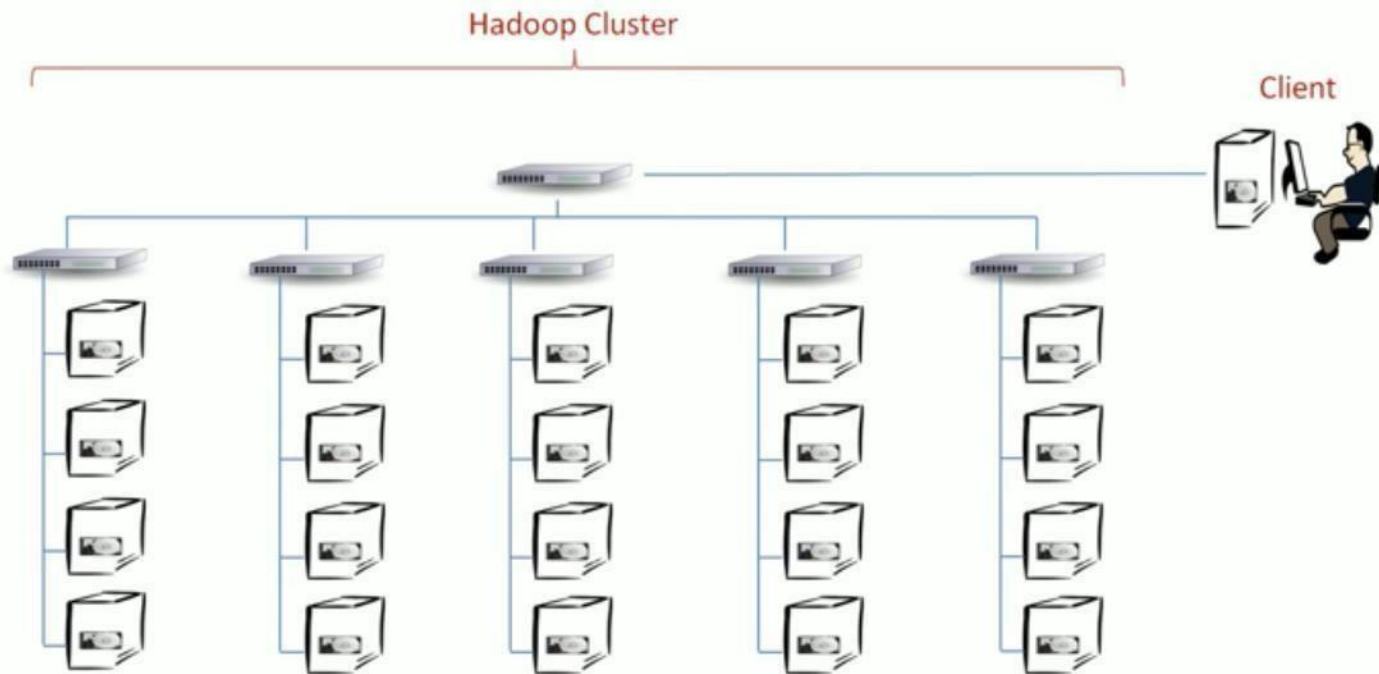
# HDFS Blocks

- Replication patterns of data blocks in HDFS



- When HDFS stores the replicas of the original blocks across the Hadoop cluster, it tries to ensure that the block replicas are stored in different failure points.

# Hadoop Cluster



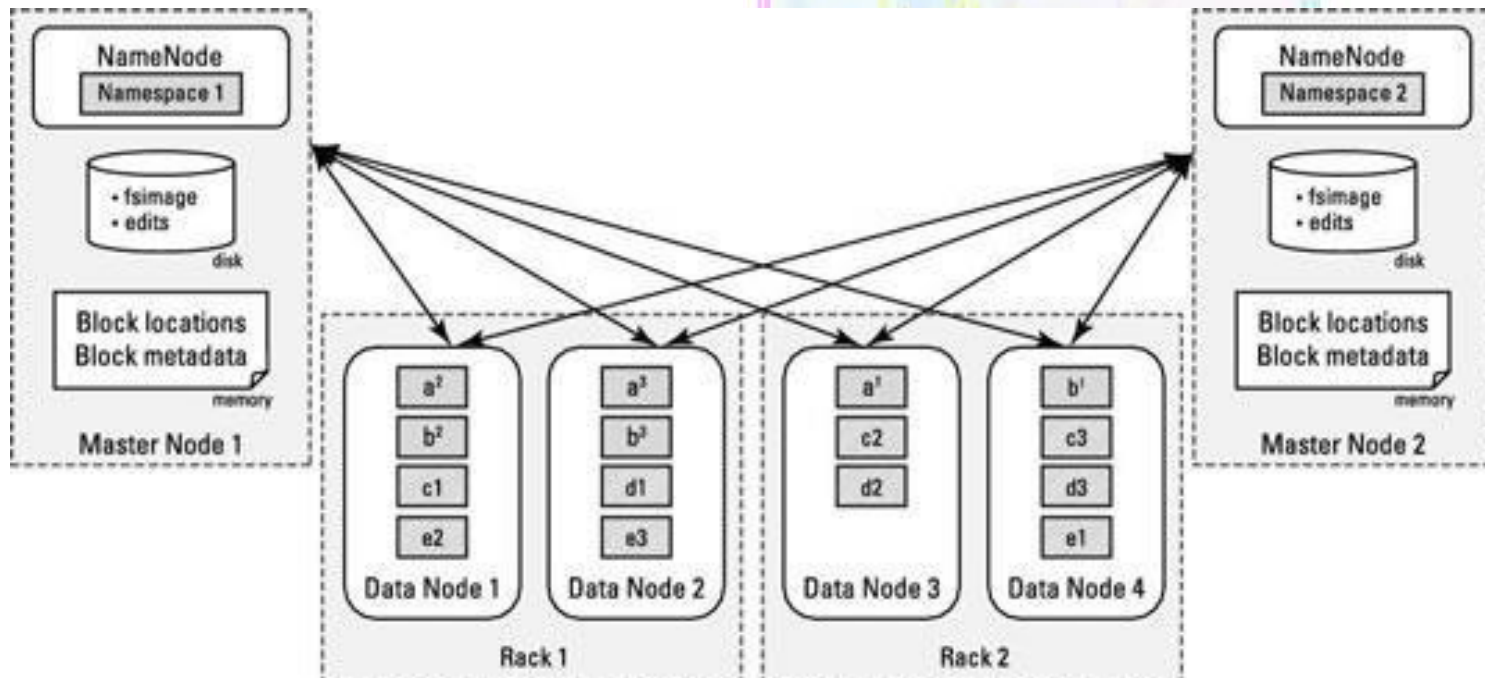
Master Slave Relationship



# Hadoop Architecture

**Name Node** – Manages file system namespace.  
**Data Node** – Manages data

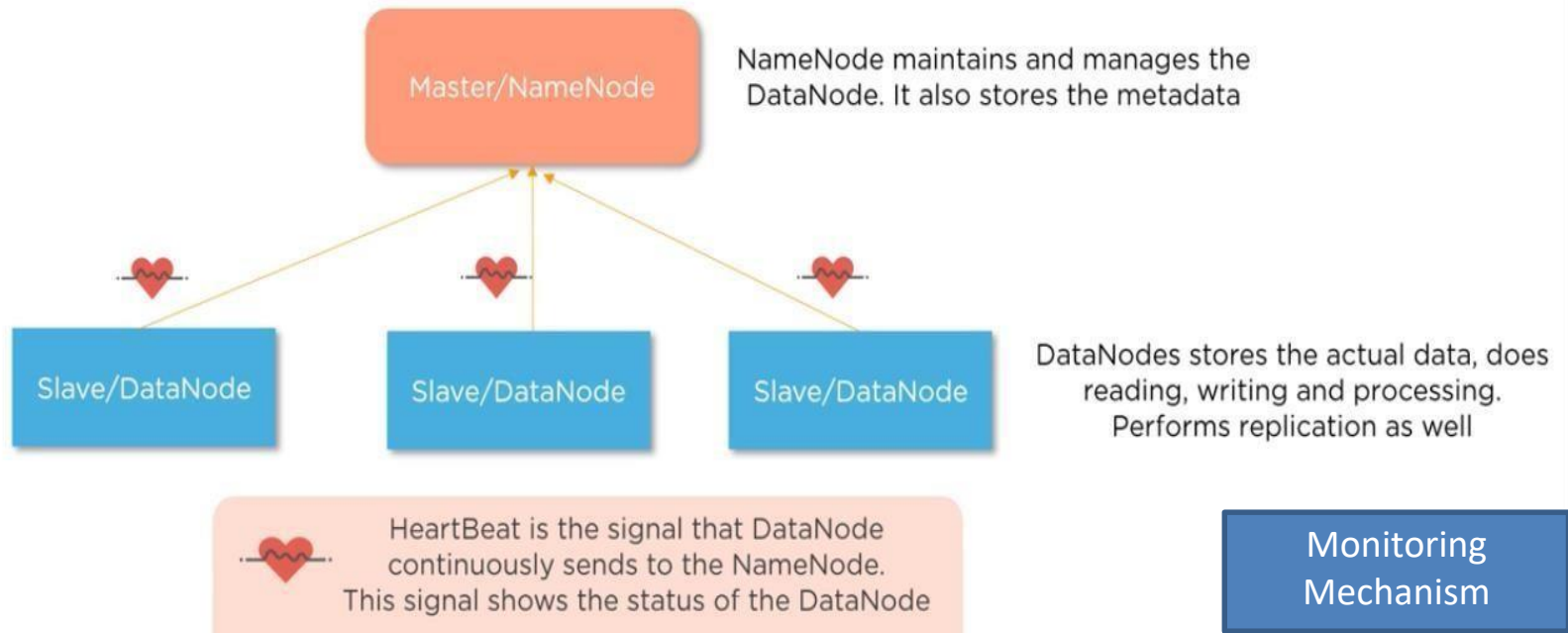
# HDFS Federation



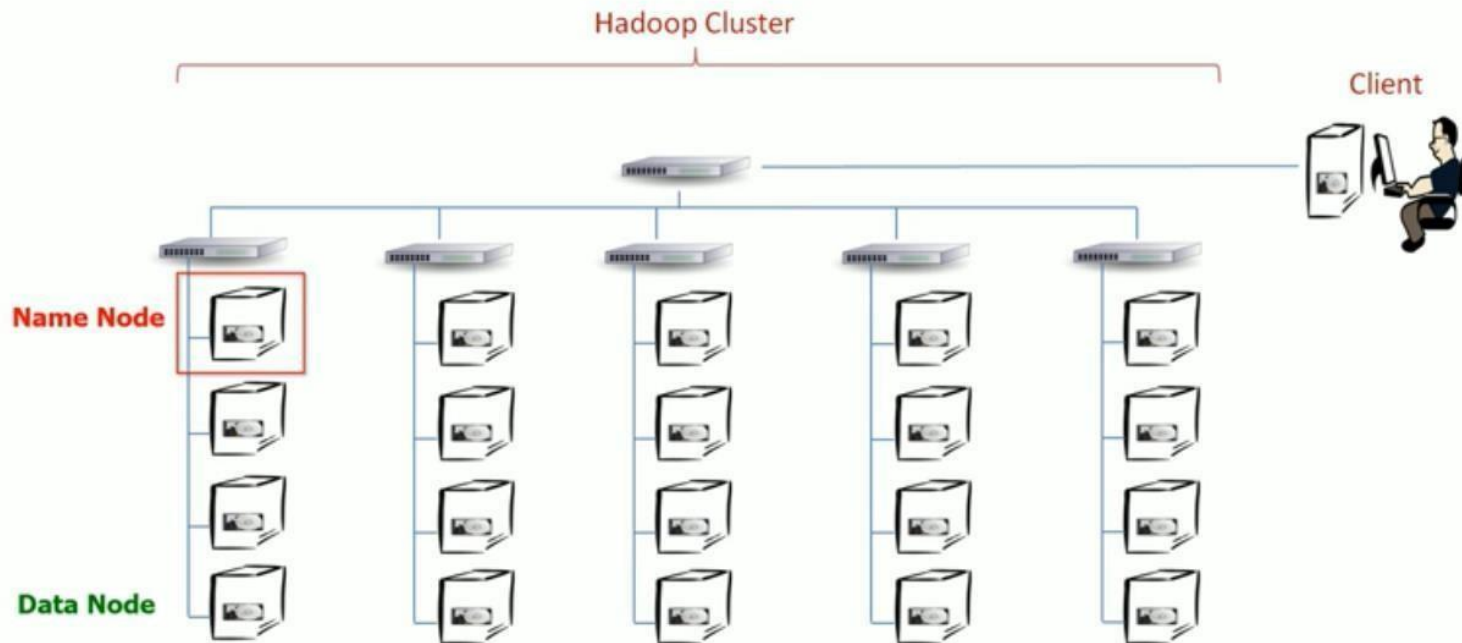
# HDFS Architecture

## What is HDFS?

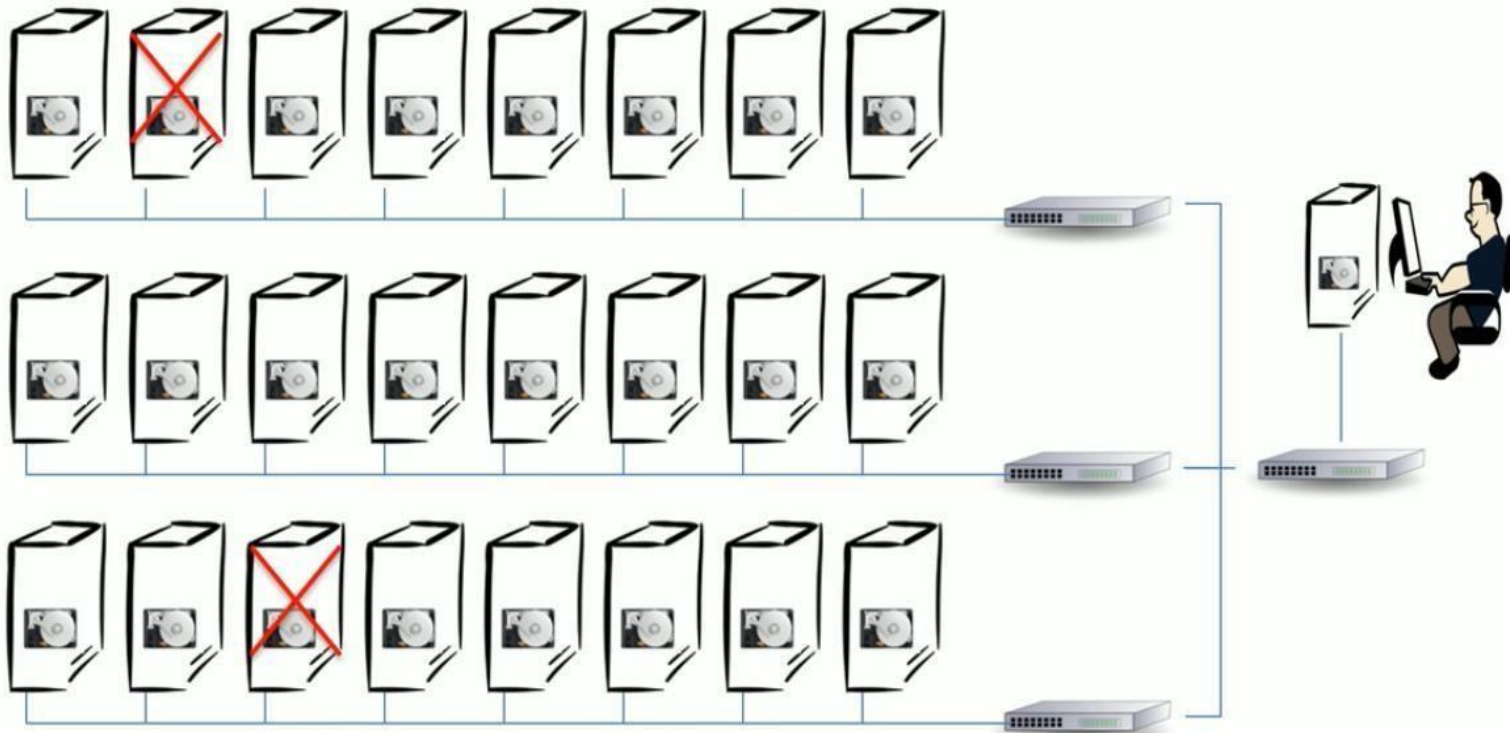
Master/slave nodes typically form the HDFS cluster



# Hadoop Cluster

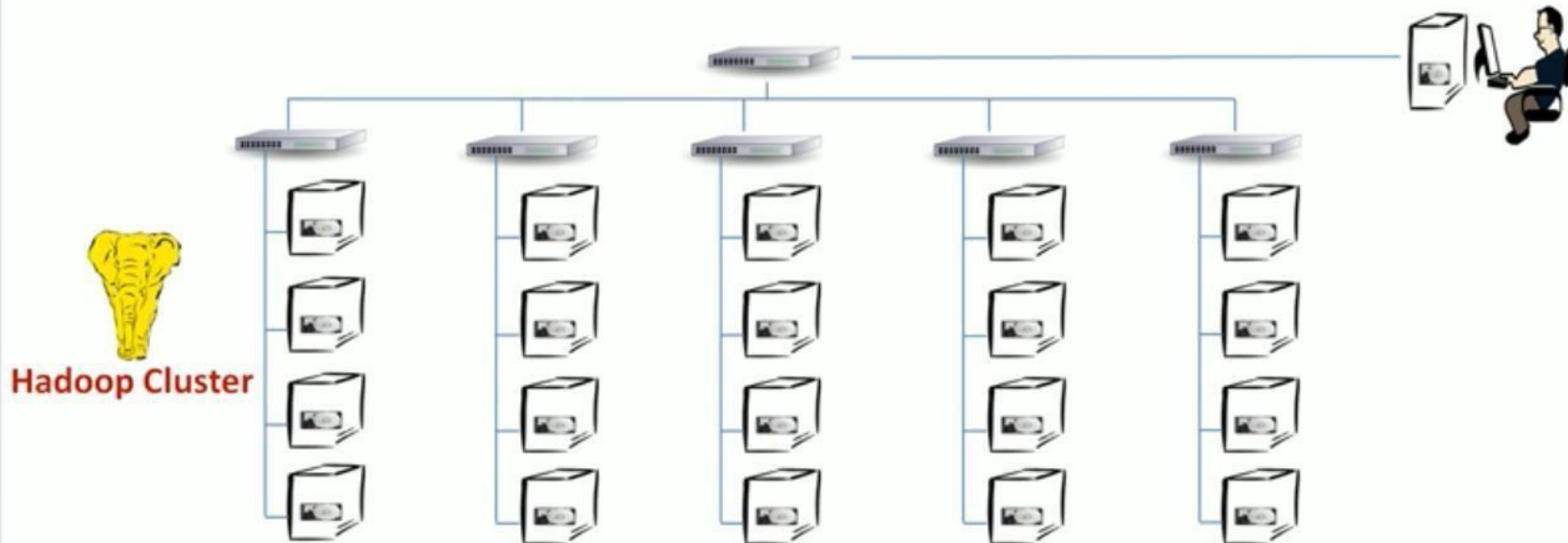


# Fault Tolerance



# Hadoop Architecture

## Fault Tolerance & High Availability

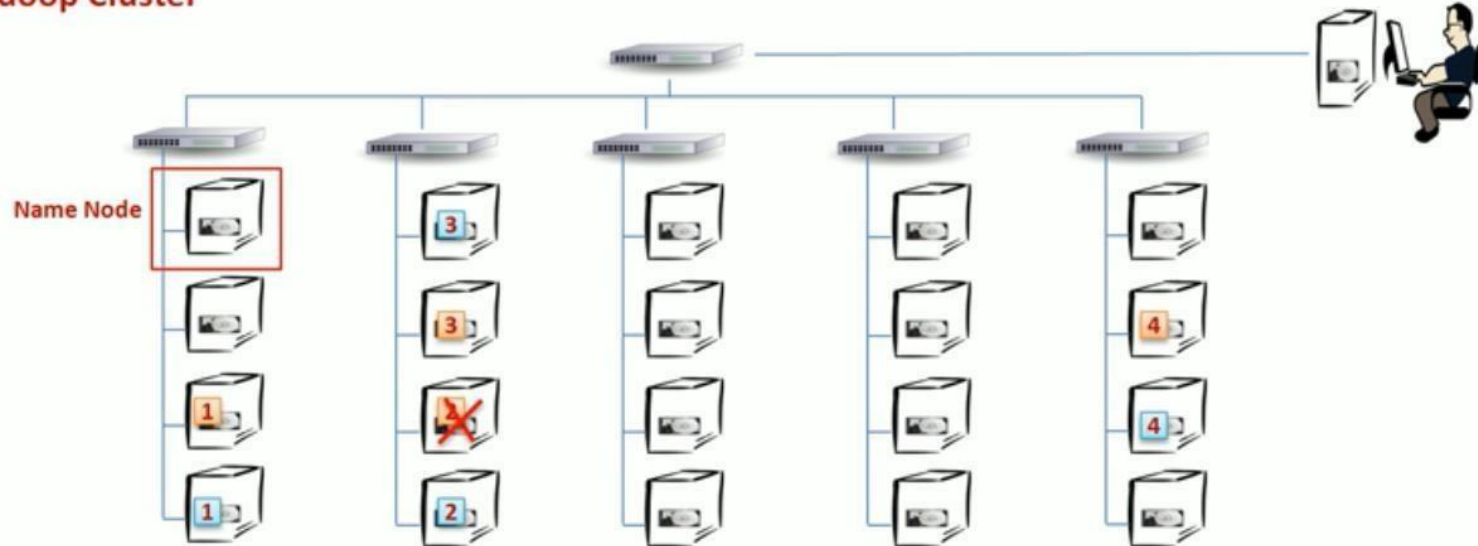


# Hadoop Architecture



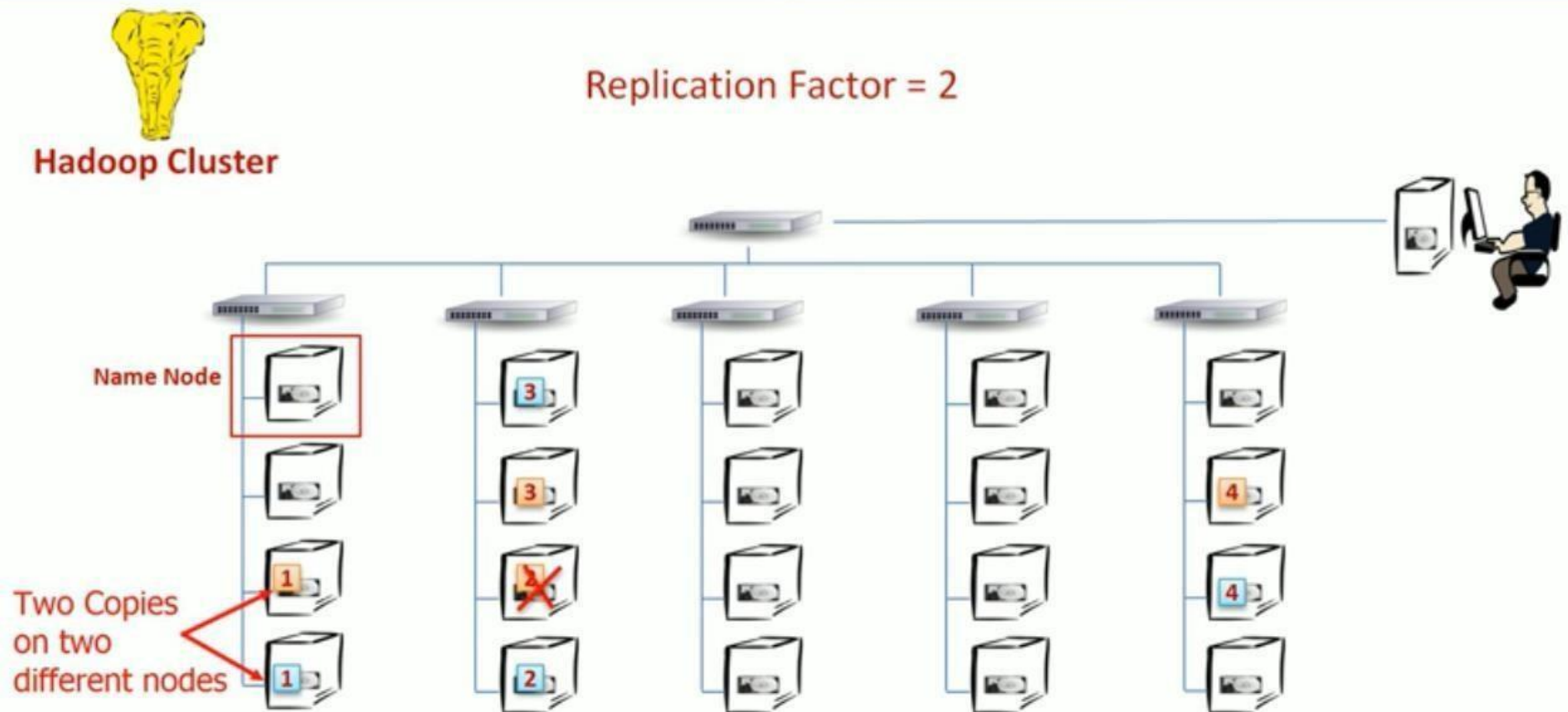
Hadoop Cluster

Replication Factor = 2

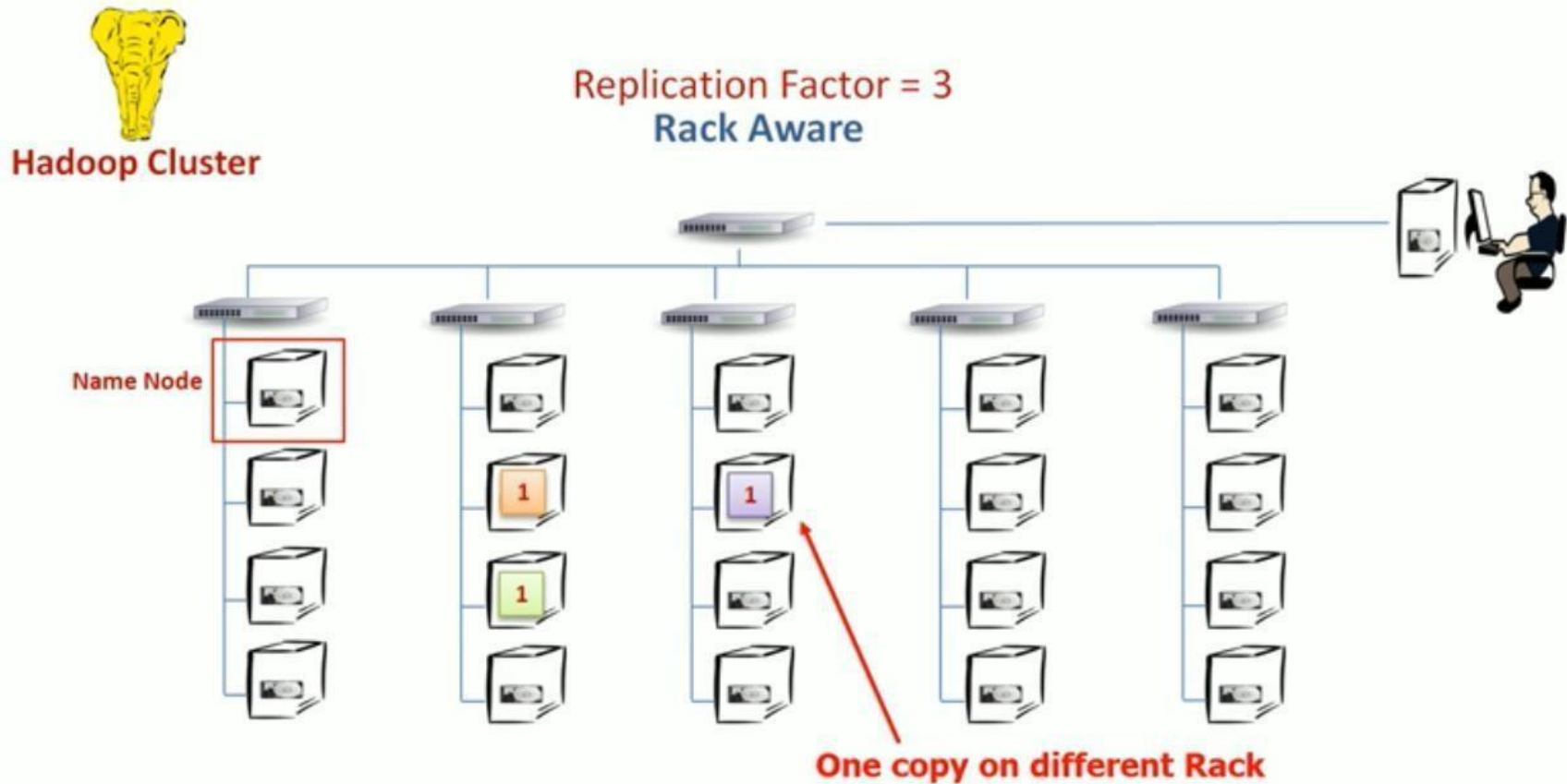




# HDFS Architecture



# HDFS Architecture – Rack Aware



# Throughput

## Performance Measurement

Minimize this.  
Lower is better.

- 1. Latency** - Time to get the first record
- 2. Throughput** - Number of records processed per unit of time.

Maximize this.  
Higher is better.

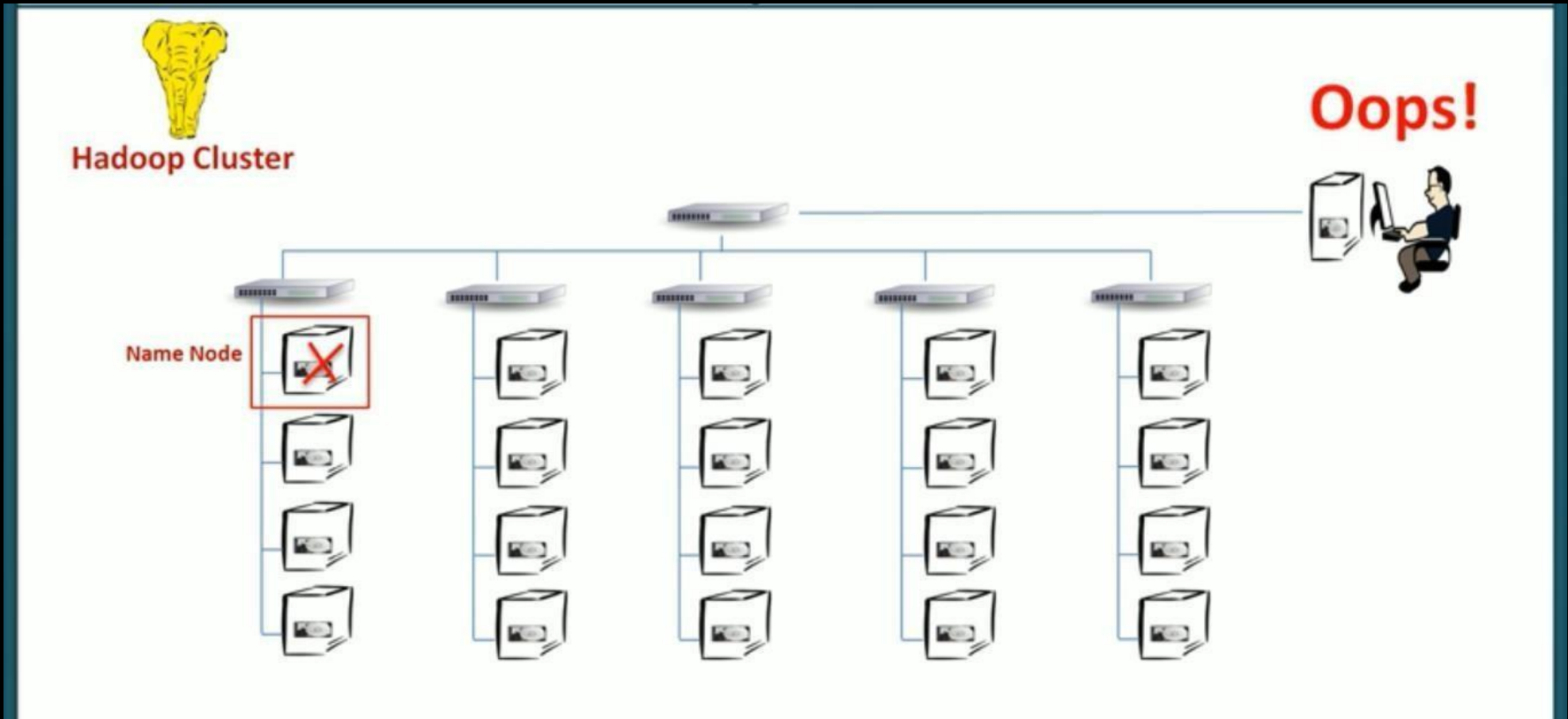
# HDFS Architecture



## High Availability

Percentage of Uptime of the system.

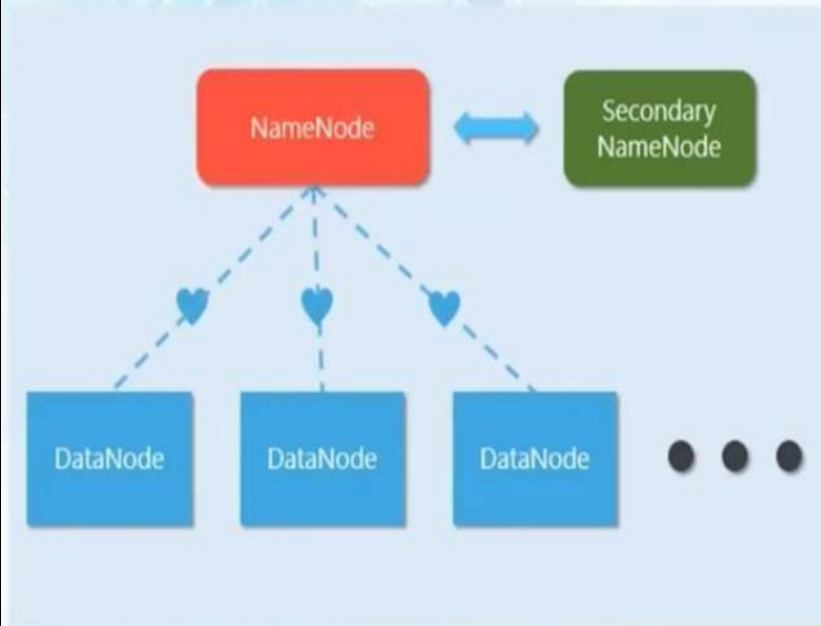
# HDFS Architecture



Name Node is critical and single point of failure

# Hadoop

## NameNode & DataNode



### NameNode:

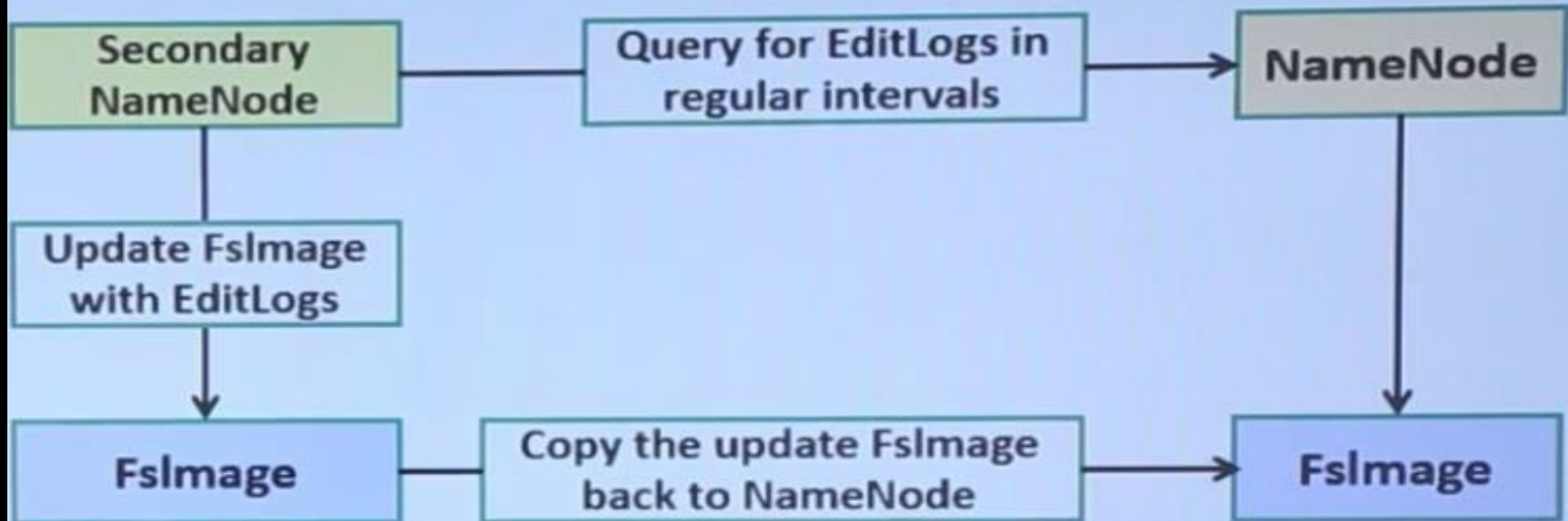
- Maintains and Manages DataNodes
- Records metadata i.e. information about data blocks e.g. location of blocks stored, the size of the files, permissions, hierarchy, etc.
- Receives heartbeat and block report from all the DataNodes

### DataNode:

- Slave daemons
- Stores actual data
- Serves read and write requests from the clients

# Hadoop – secondary NameNode

## HDFS Architecture



### ➤ What is the Secondary NameNode?

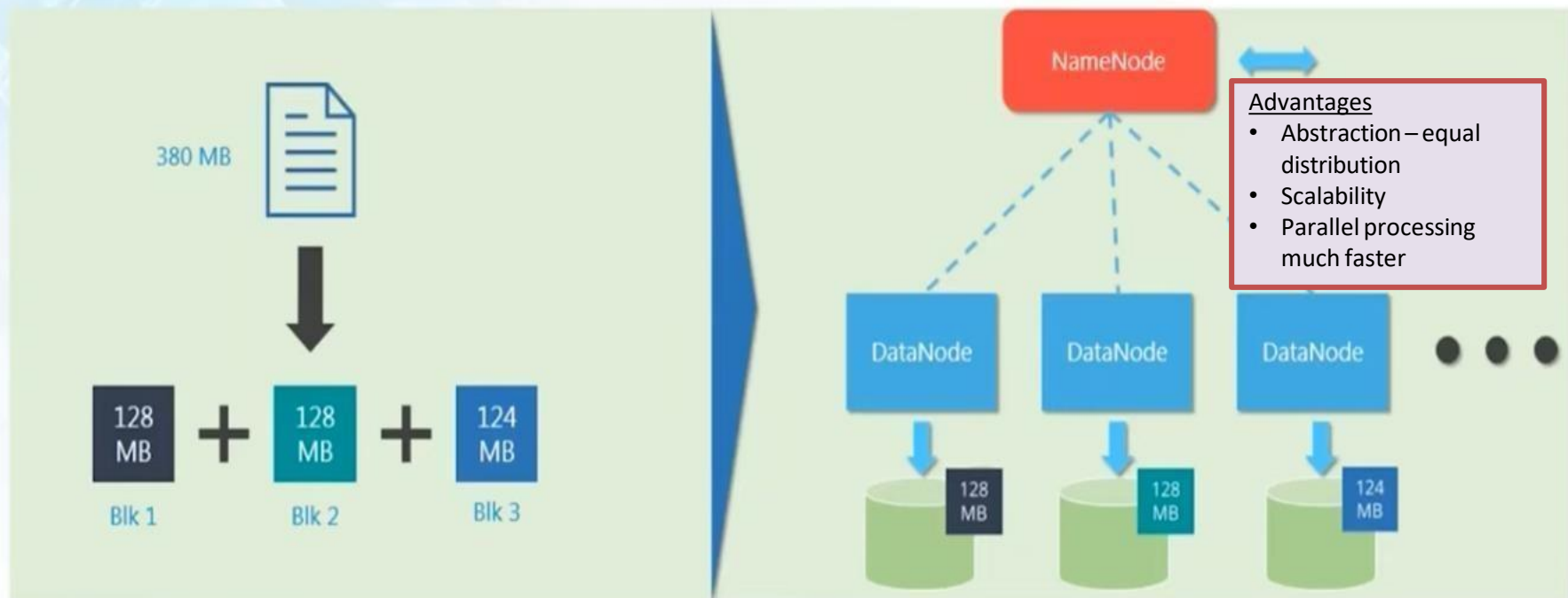
- **Secondary NameNode:** The Secondary NameNode is another specially dedicated node, which is used to take the checkpoints of the file-system. The Secondary NameNode is not the substitute of the Primary NameNode. It helps the NameNode but not replace for NameNode.



# Hadoop – Data Storage

## HDFS Data Blocks

- Each file is stored on HDFS as blocks
- The default size of each block is 128 MB in Apache Hadoop 2.x (64 MB in Apache Hadoop 1.x)

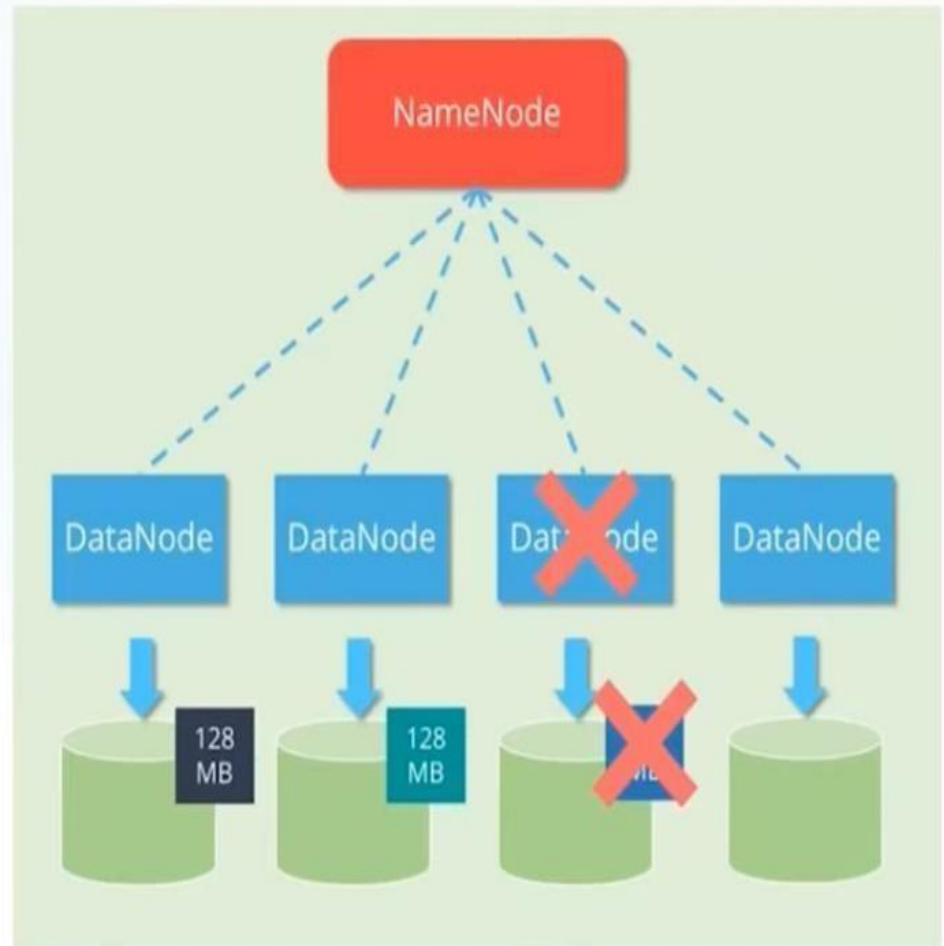
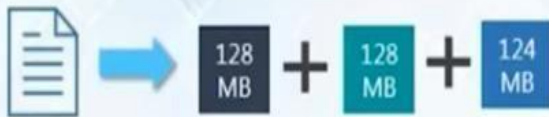


# Hadoop – Data Storage

## Fault Tolerance

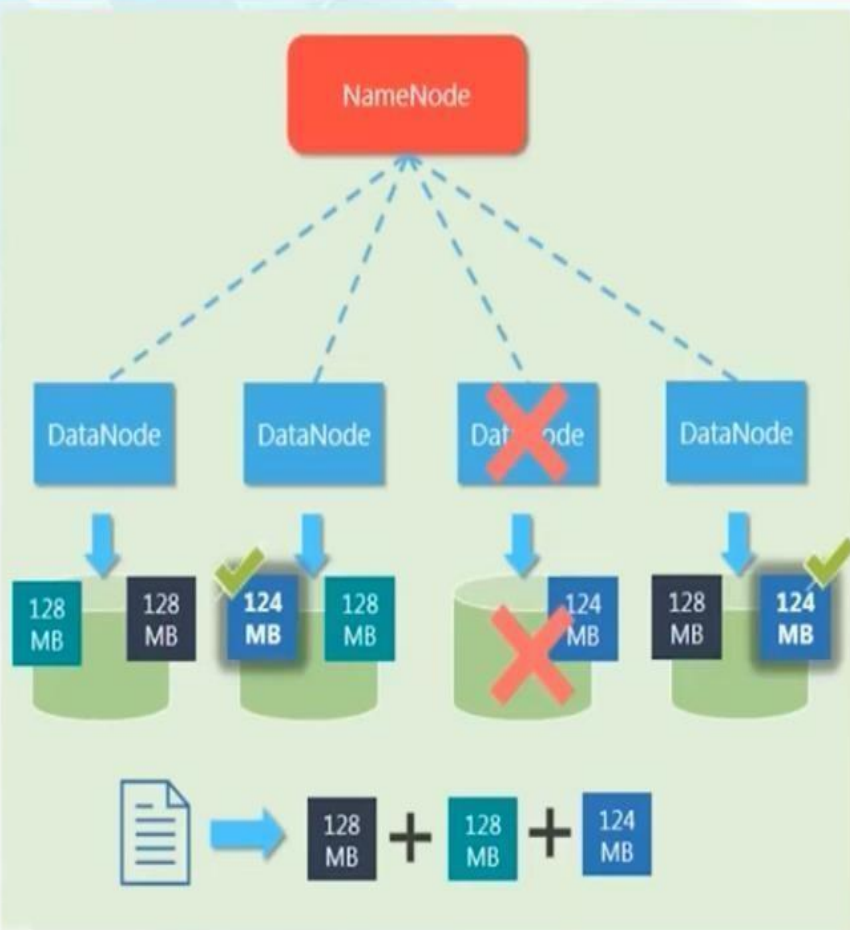
### Scenario:

One of the DataNodes crashed containing the data blocks



# Hadoop – Data Storage

## Fault Tolerance: Replication Factor

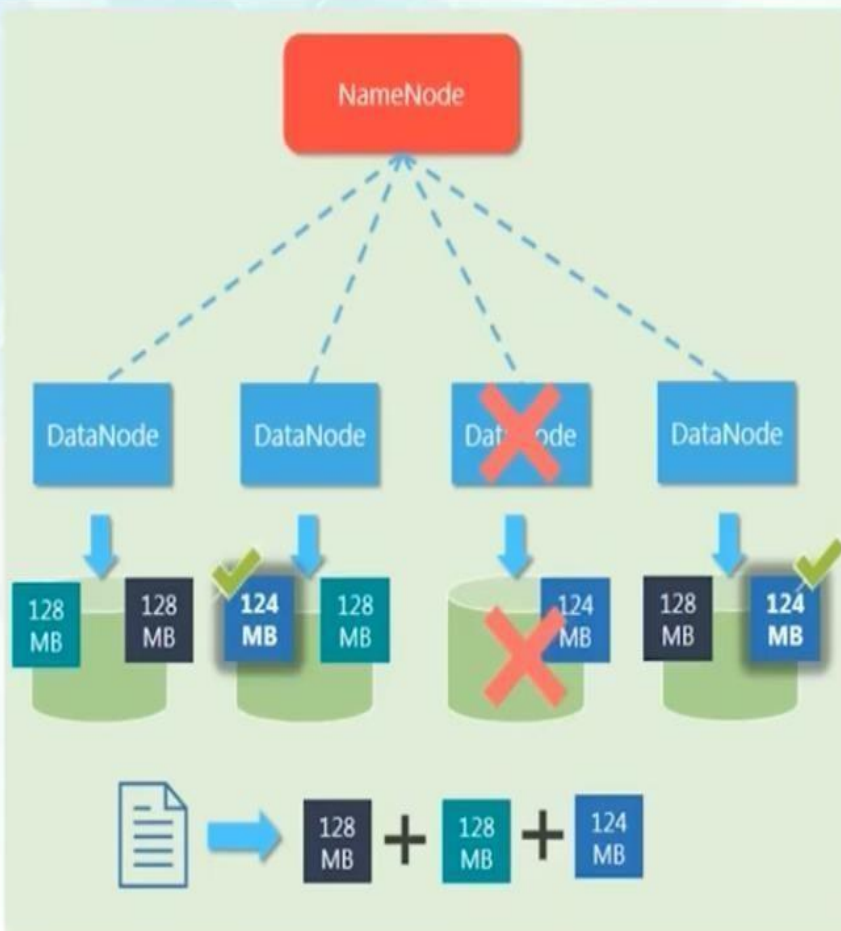


### Solution:

Each data blocks are replicated (thrice by default) and are distributed across different DataNodes

# Hadoop – Data Storage

## Fault Tolerance: Replication Factor



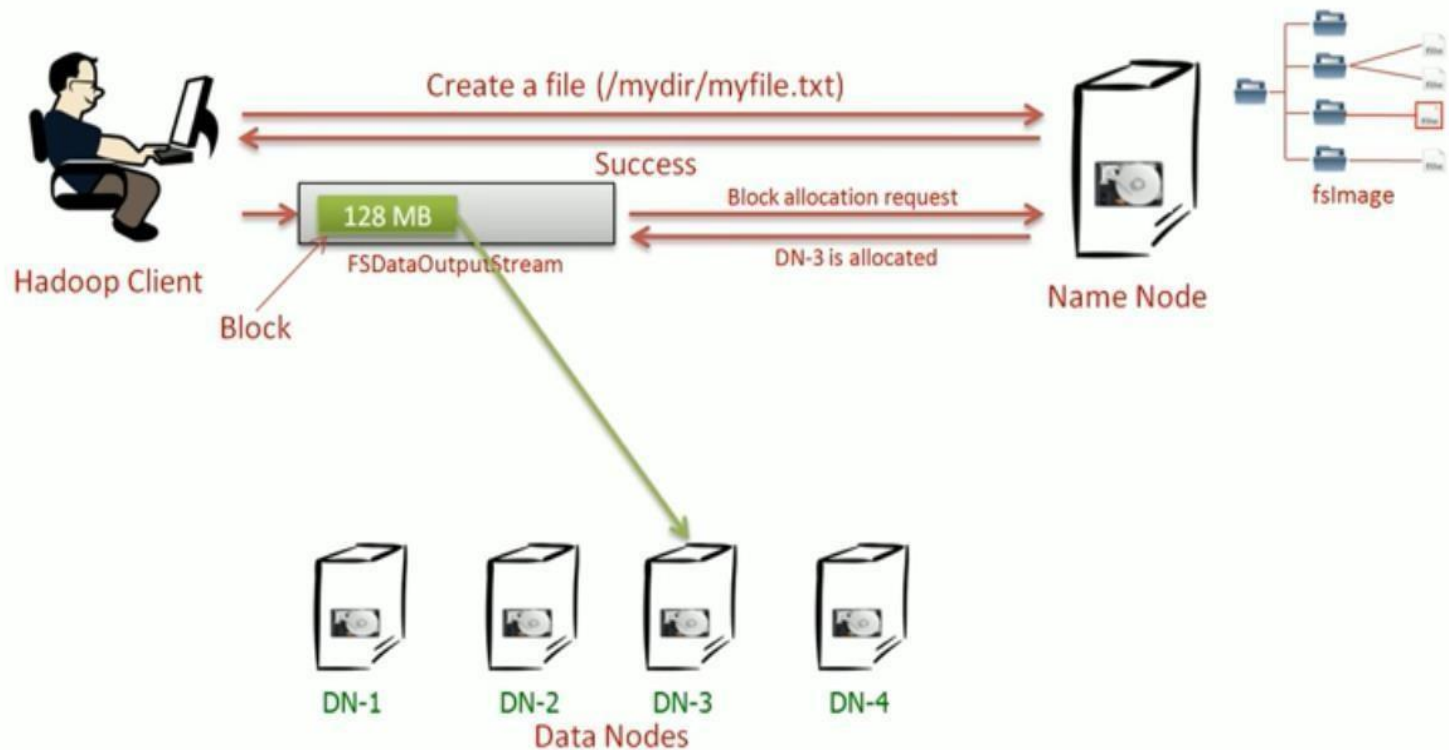
### Solution:

Each data blocks are replicated (thrice by default) and are distributed across different DataNodes



As it is said Never Put All Your Eggs in the Same Basket

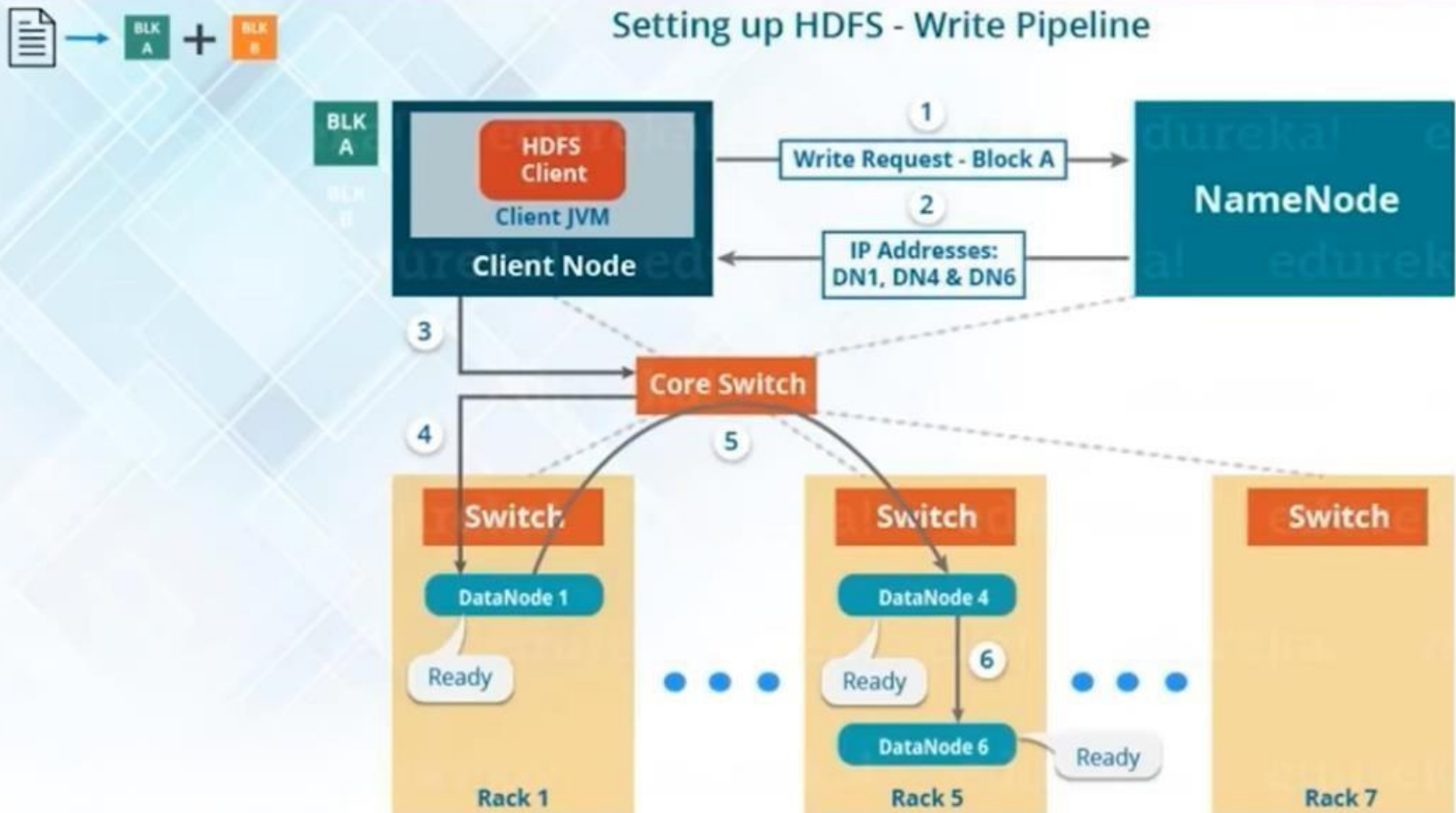
# Hadoop Architecture





# HDFS – Write Mechanism

## HDFS Write Mechanism – Pipeline Setup

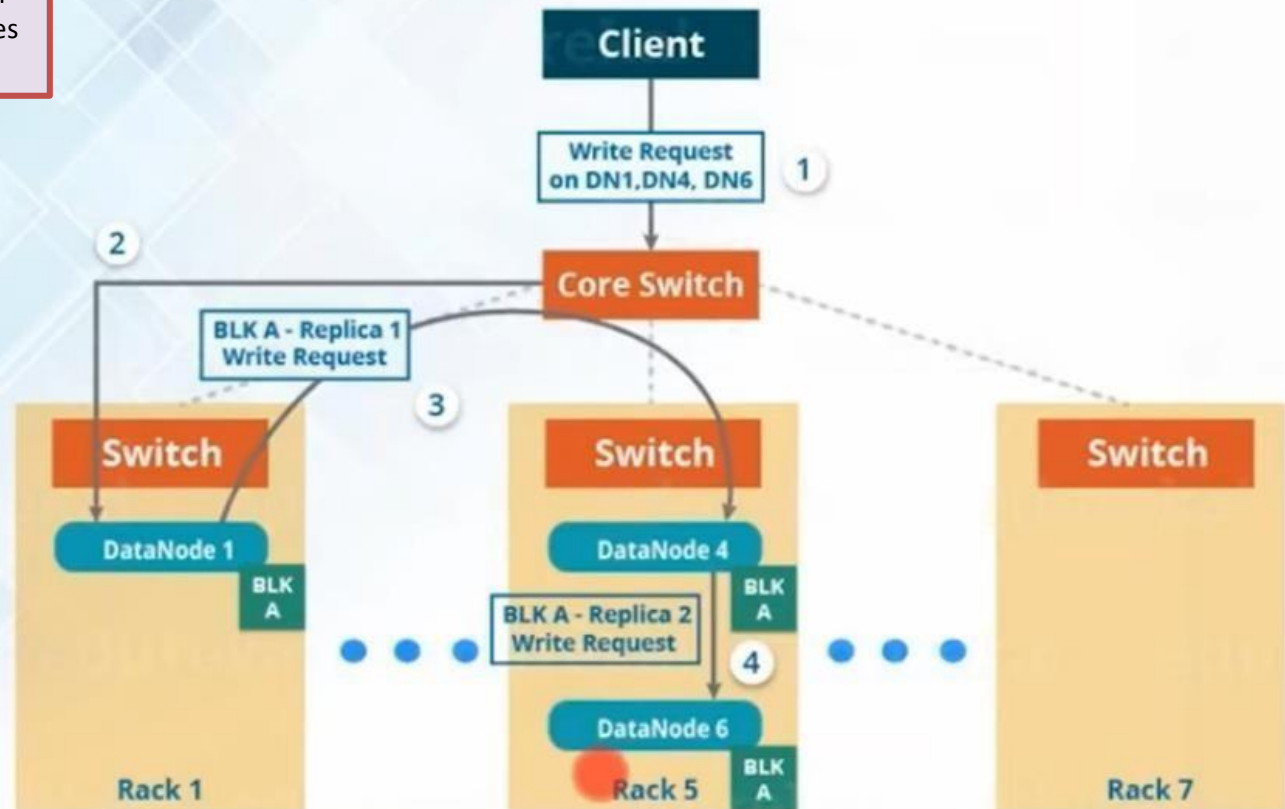


# HDFS – Write Mechanism

## HDFS Write Mechanism – Writing a Block

- Client Copies on DN-1 and then DN-1 contacts DN-2 for copying and in turn DN-2 does it on DN-6

HDFS - Write Pipeline

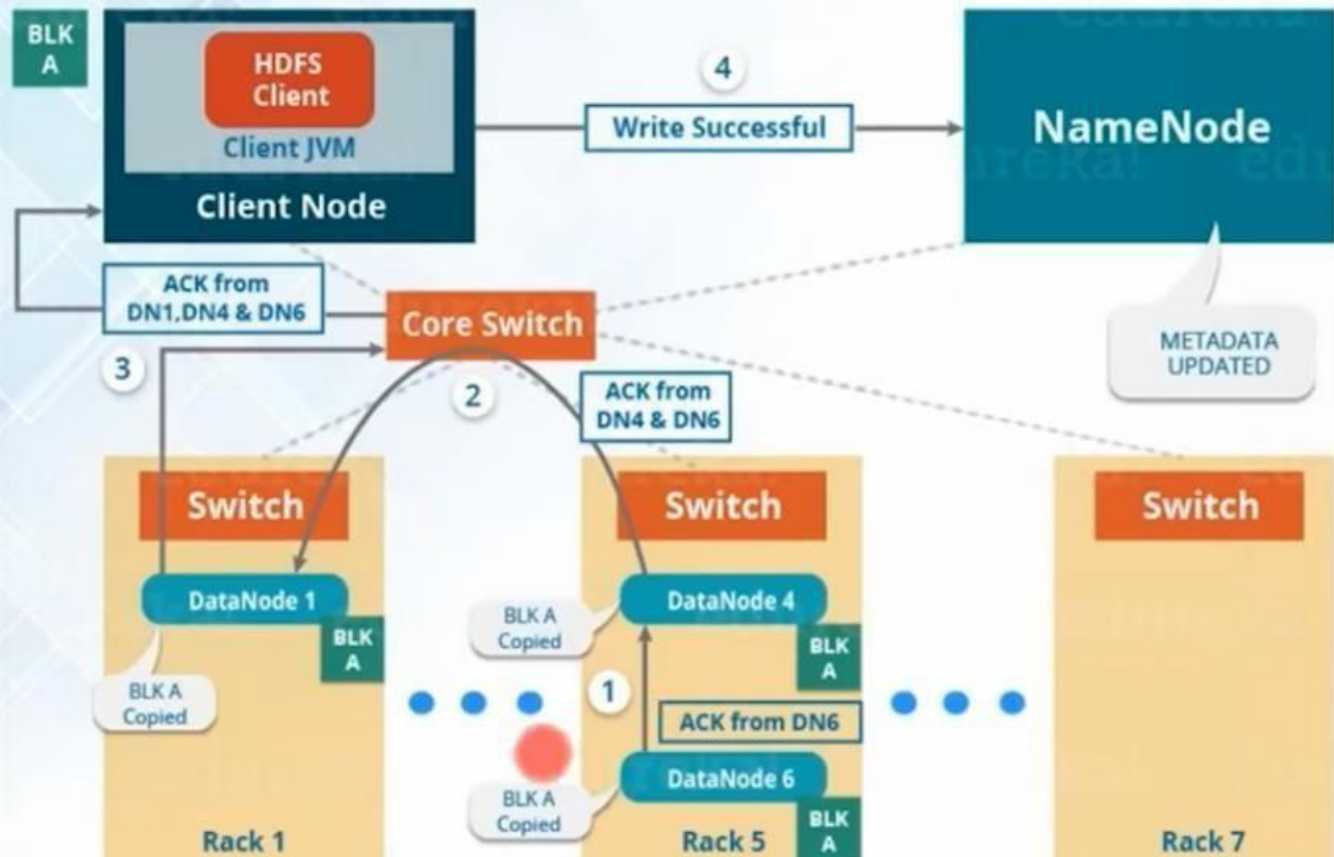




# HDFS – Write Mechanism

## HDFS Write Mechanism - Acknowledgement

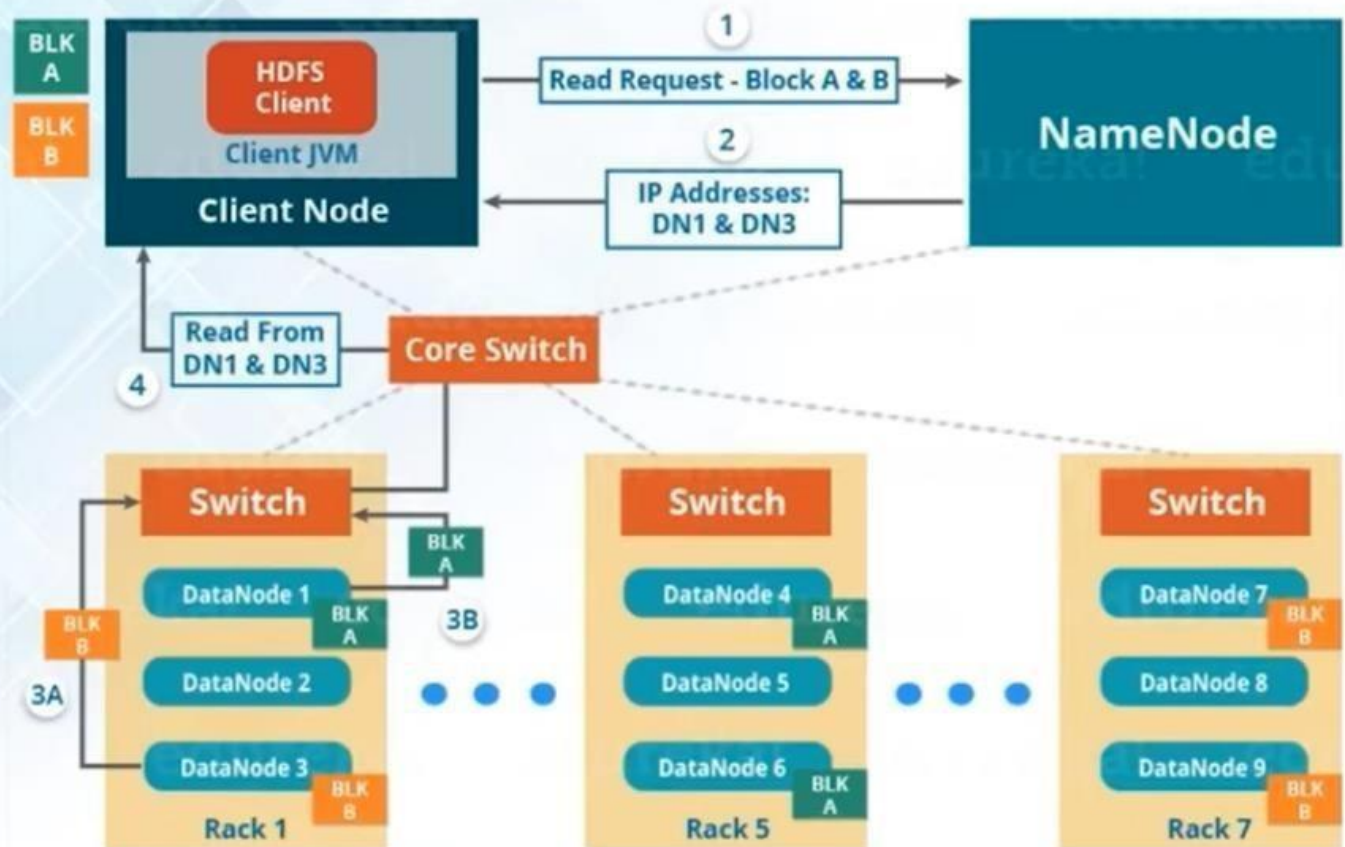
Acknowledgement in HDFS - Write



# HDFS – Read Mechanism

## HDFS Read Mechanism

HDFS - Read Architecture



# Control and Data Flow - Summary

- Read:
  - The client queries the NameNode with the file name, read range start offset, and the range length.
  - The NameNode returns the locations of the blocks of the specified file within the specified range.
  - The client then sends a request to one of the DataNodes, most likely the closest one.
- Write:
  - A client request to create a file does not reach the NameNode immediately.
  - The client caches the file data into a temporary local file.
  - Once the local file accumulates data worth over one block size, the client contacts the NameNode, which updates the file system namespace and returns the allocated data block location.
  - The client flushes the block from the local temporary file to the specified DataNode.
  - When a file is closed, the remaining last block data is transferred to the DataNodes.

# Control and Data Flow - Summary

- HDFS is designed such that clients never read and write file data through the NameNode.
- A client asks the NameNode which DataNodes it should contact using the class ClientProtocol through an RPC connection.
- Then the client communicates with a DataNode directly to transfer data using the DataTransferProtocol, which is a streaming protocol for performance reasons.

# HDFS Architecture



**How to protect against NN Failure?**

**Backup**



# HDFS Architecture

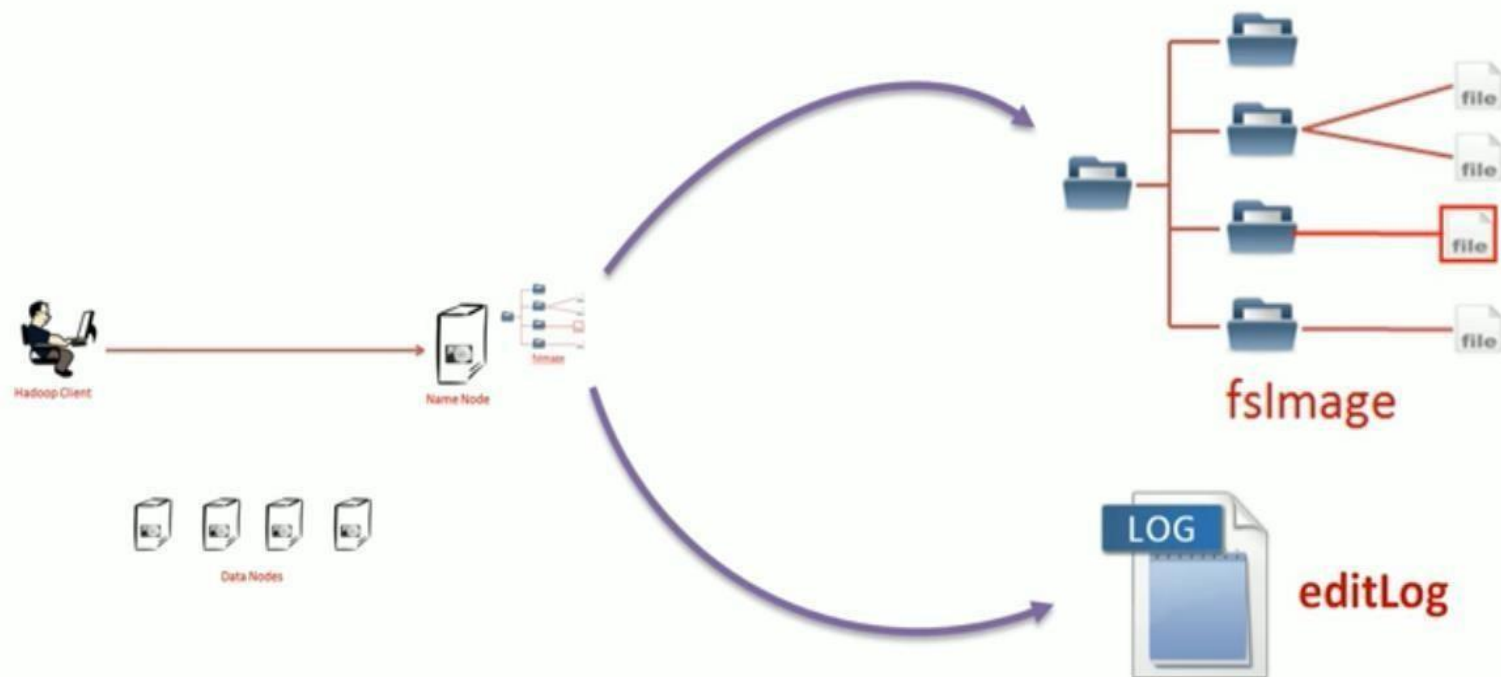


## How to protect against NN Failure?

### Backup following things

- HDFS Namespace information
- Standby Name node

# HDFS Architecture



Namespaces can be reconstructed using edit logs



# HDFS Architecture



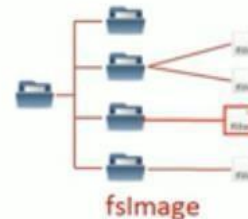
**Secondary Name Node**

# HDFS Architecture

## Secondary Name Node



- **fslmage**



In Memory and we loose it once NN reboots but can be re-built using edit logs

- **editLog**



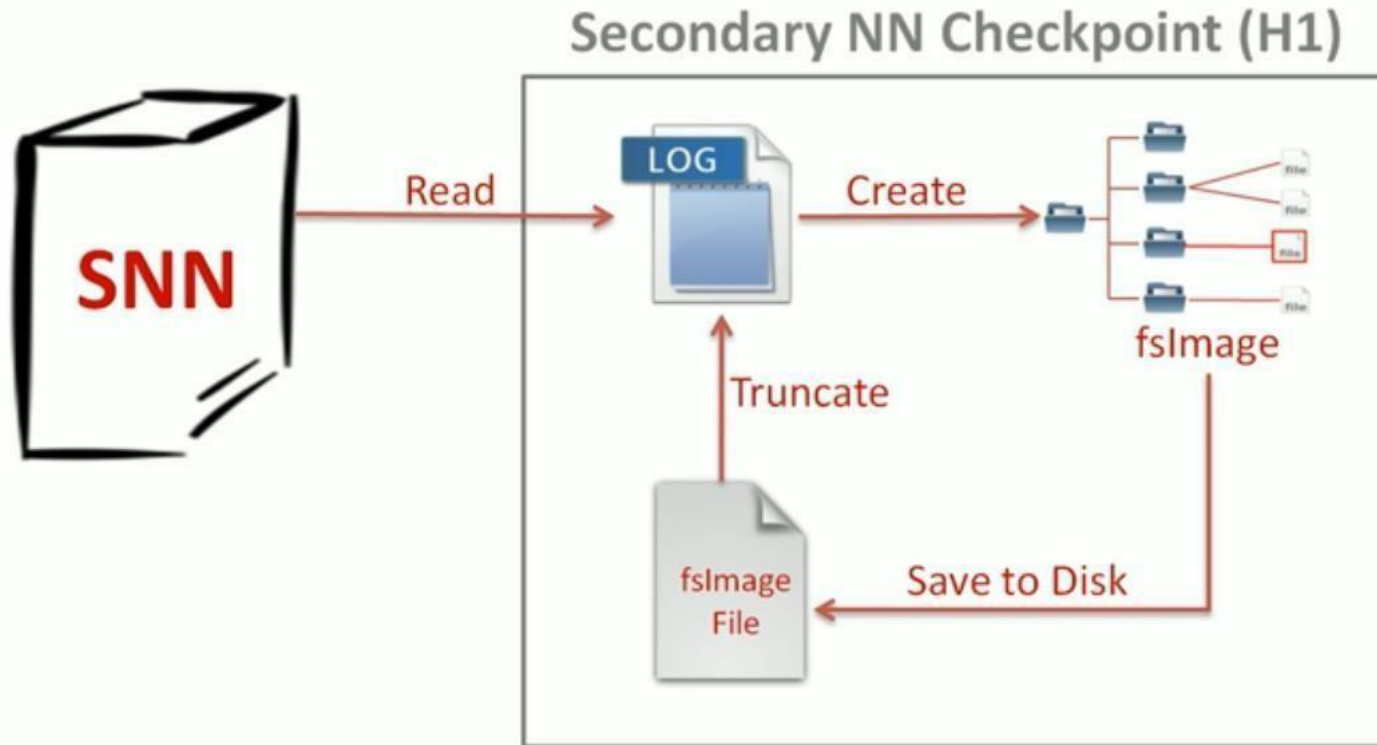
# HDFS Architecture

## What if you restart NN?

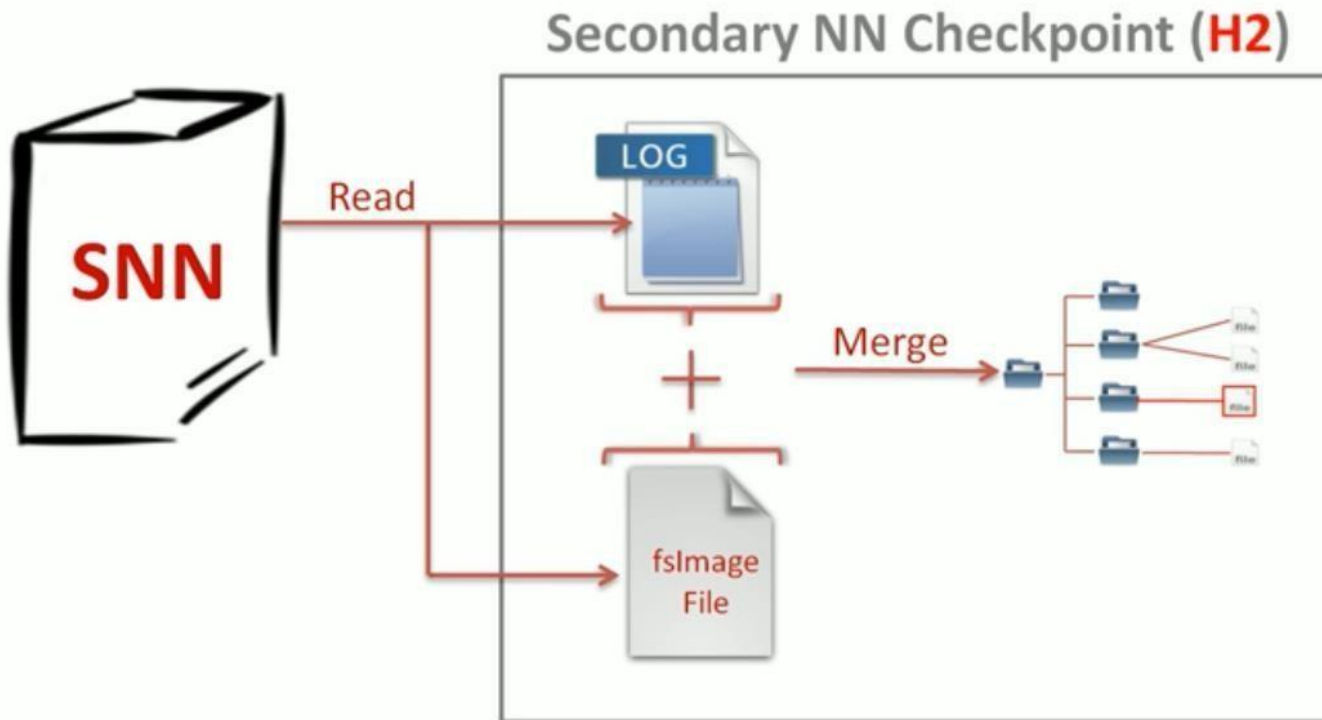
Time is critical in re-booting NN but edit file is huge so may consume time



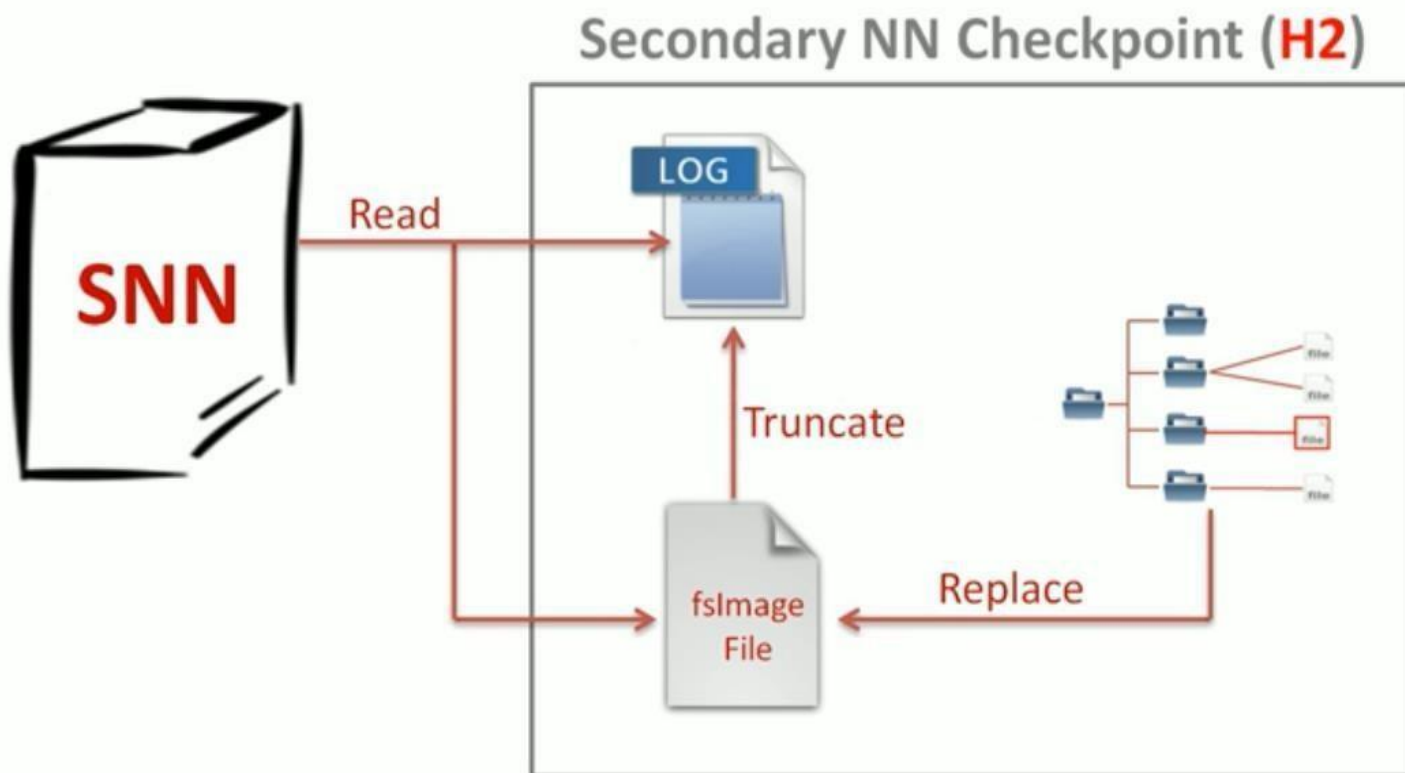
# HDFS Architecture Check Point - 1 (every hr)



## HDFS Architecture Check Point -2

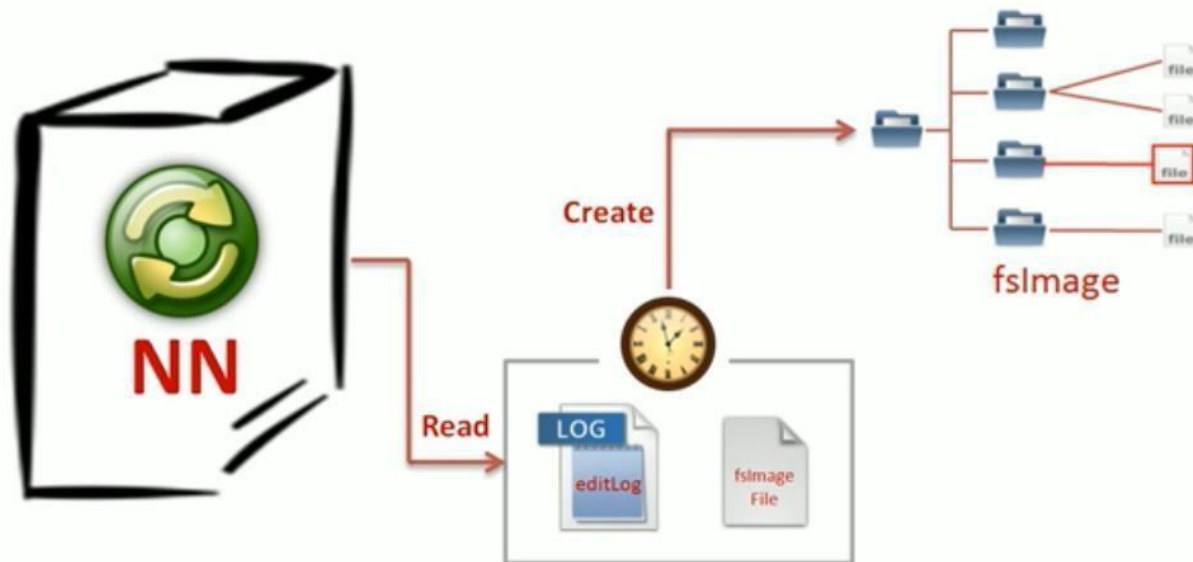


# HDFS Architecture Check Point -2



## HDFS Architecture Check Point -2

**What if you restart NN?**





# HDFS – Basic Commands

## Basic HDFS Commands


➤ **Some basic HDFS Commands:**

- HDFS commands are very similar with the UNIX commands. Some syntax and output format may differ for the commands.

Command	Description
ls	List all files with permissions and other details.
mkdir	Create a directory
rm	Remove File or Directory
put	Store file/folder from local disk to HDFS
cat	Concatenate text /display file content
get	Store file/folder from HDFS to local disk
count	Count number of directory, number of files and file size.

# Hadoop Architecture

## Summary

- 
- Master/Slave architecture
    - Single Name Node (Master)
    - One or more Data Nodes (Slaves)
  - NN manages FS namespace
  - All client interactions start with NN
  - DN stores file data as Blocks
  - DN sends heartbeat and block report to NN
  - File is broken into Blocks and stored on DN
  - NN maintains file to block mapping, location, order of blocks and other metadata.
  - Default block size is 128 MB
  - You can change block size for a file
  - Client directly interacts with DN for reading/writing blocks
  - Client buffers data locally to provide streaming read/write
  - NN and DN can be installed on single machine to create a single node cluster for learning

**THANK YOU**