Big Data Hadoop Certification

Introduction to Big Data and Apache Hadoop

Agenda

01 What is Big Data?

02 Categorizing Data as Big Data

03 What is Hadoop?

04 Why Hadoop is a solution?

05 Hadoop Ecosystem

06 What is HDFS?

07 HDFS: Why Another File System?

08 HDFS Architecture

09 Quiz

Copyright Intellipaat. All rights reserved.

What is Big Data?

Copyright Intellipaat. All rights reserved.

What is Data?

Data is the individual unit of information that is defined as figures or facts and stored for later processing

Data Types:

Structured Data

Semi-structured Data

Unstructured Data

Copyright Intellipaat. All rights reserved.

What is Big Data?

Big Data is a collection of extremely large datasets that may be analyzed computationally to reveal patterns, trends, and associations, especially relating to human behavior and interactions.

It captures voices of the flight crew, recordings of microphones and earphones, and the performance information of the aircraft.

Black Box Data

Social media such as Facebook and Twitter hold information and the views posted by millions of people across the globe.

Social Media Data

Similar Examples:

Power Grid Data

Transport Data

Stock Exchange Data

Search Engine Data

Copyright Intellipaat. All rights reserved.

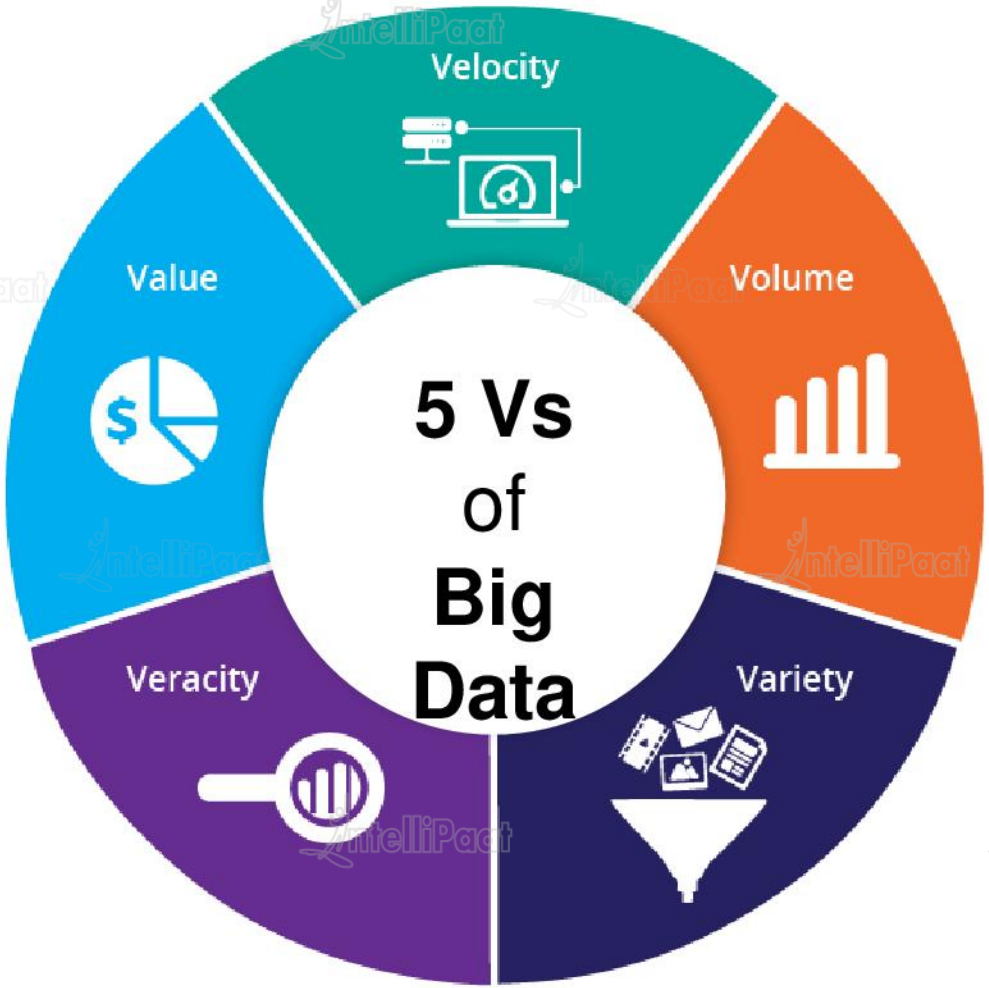
Categorizing Data as Big Data

5 Vs of Big Data

Copyright Intellipaat. All rights reserved.

5 Vs of Big Data

To make sense of the huge amount of data, it is often broken down based on the five Vs: Velocity, Volume, Value, Variety, and Veracity

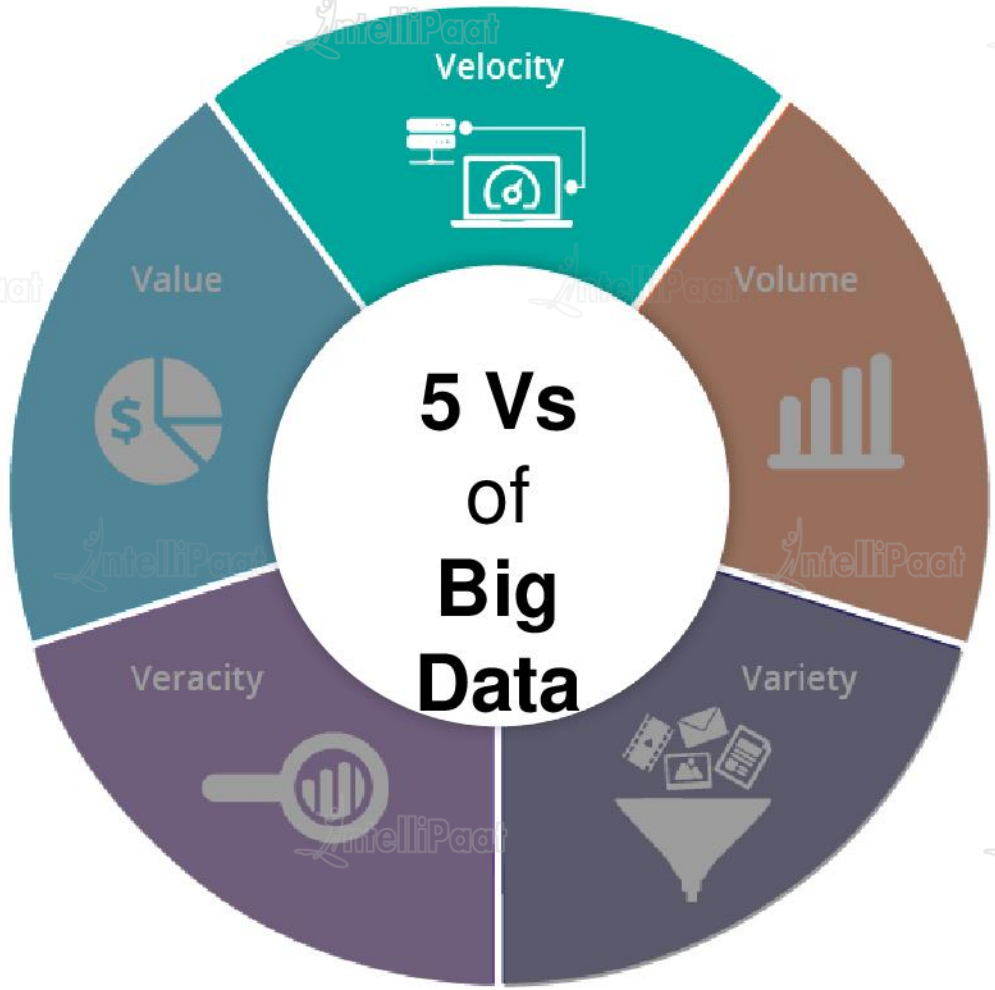


Copyright Intellipaat. All rights reserved.

5 Vs of Big Data: Velocity

- Velocity refers to the high speed at which data flows in from

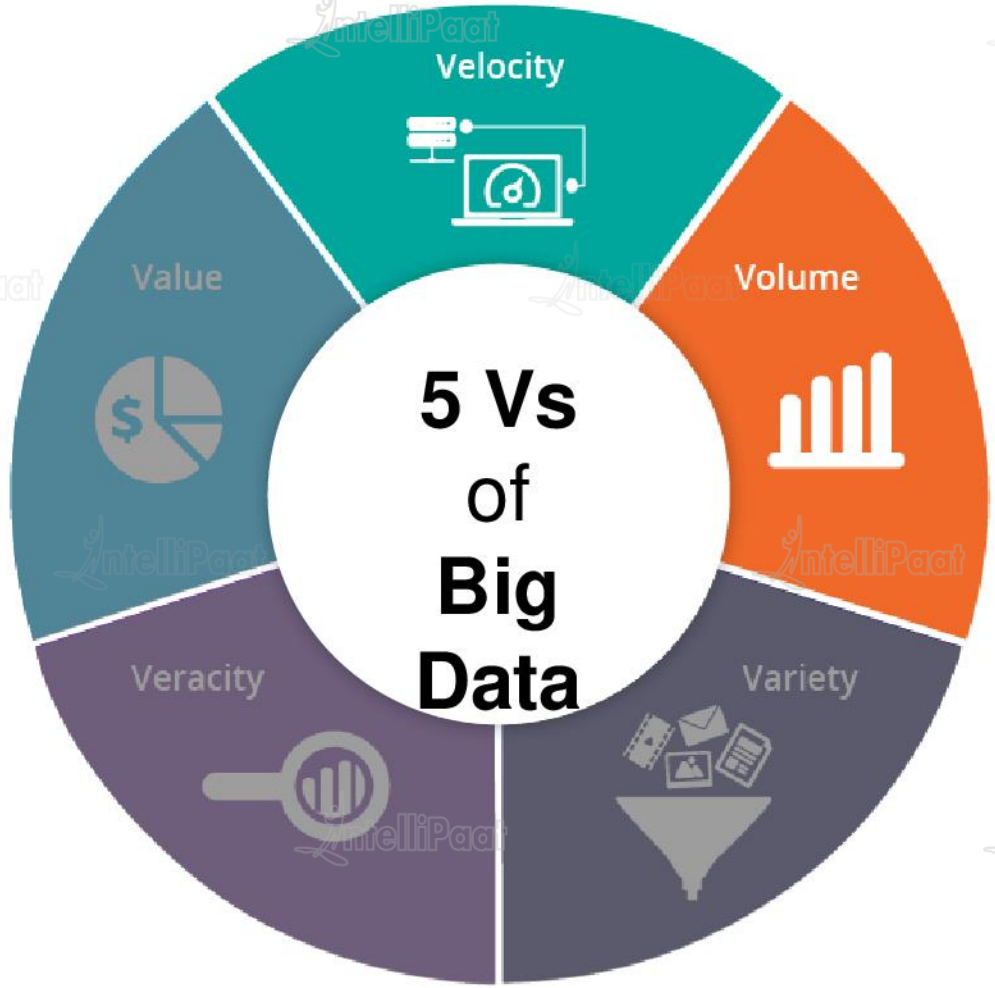
  multiple sources such as machines, networks, social media, mobile phones, etc.
- There is a massive and continuous flow of data
- **Example**: On Google, more than **3.5 billion searches are made per day**



Copyright Intellipaat. All rights reserved.

5 Vs of Big Data: Volume

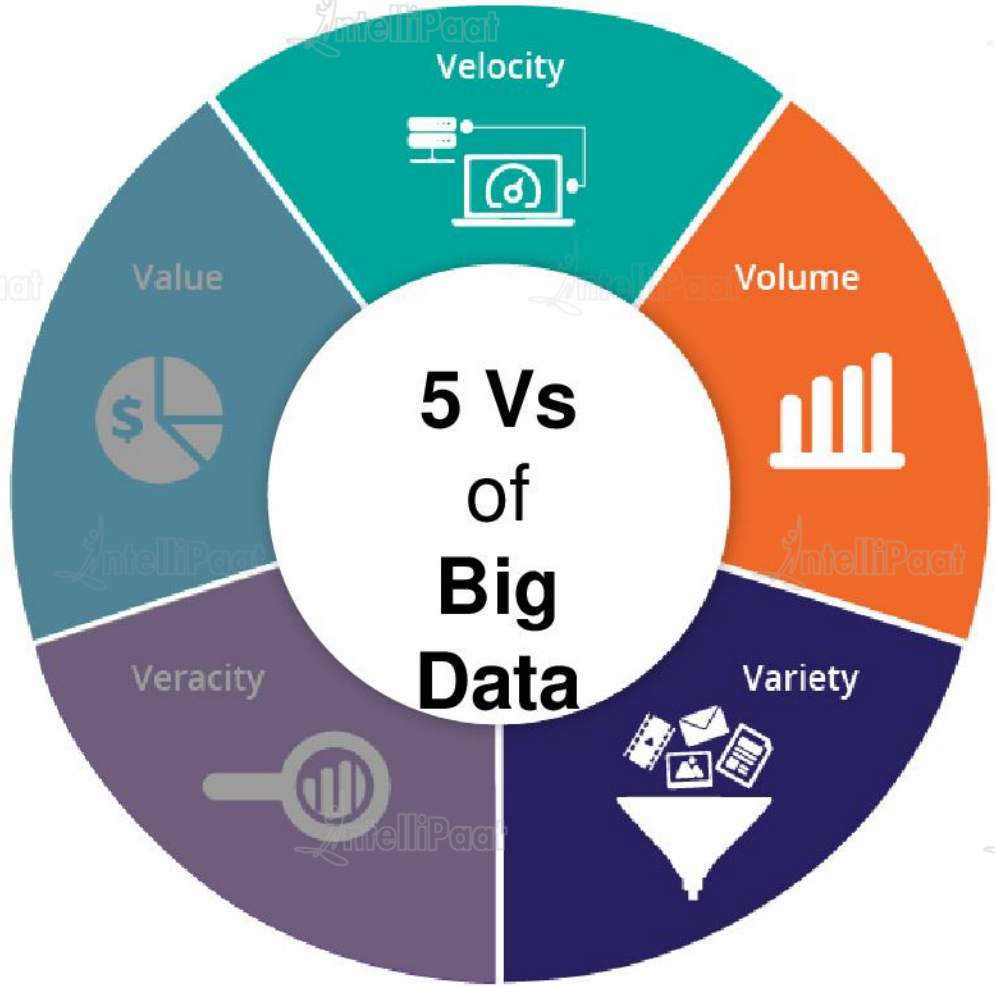
- Volume means the huge amount of data!
- To determine volume, the size of the data plays a crucial role
- When dealing with Big Data, it is necessary to consider the 'volume' of it
- **Example:** In the year 2016, the estimated global mobile traffic was **6.2 exabytes (6.2 billion GB) per month**



Copyright Intellipaat. All rights reserved.

5 Vs of Big Data: Variety

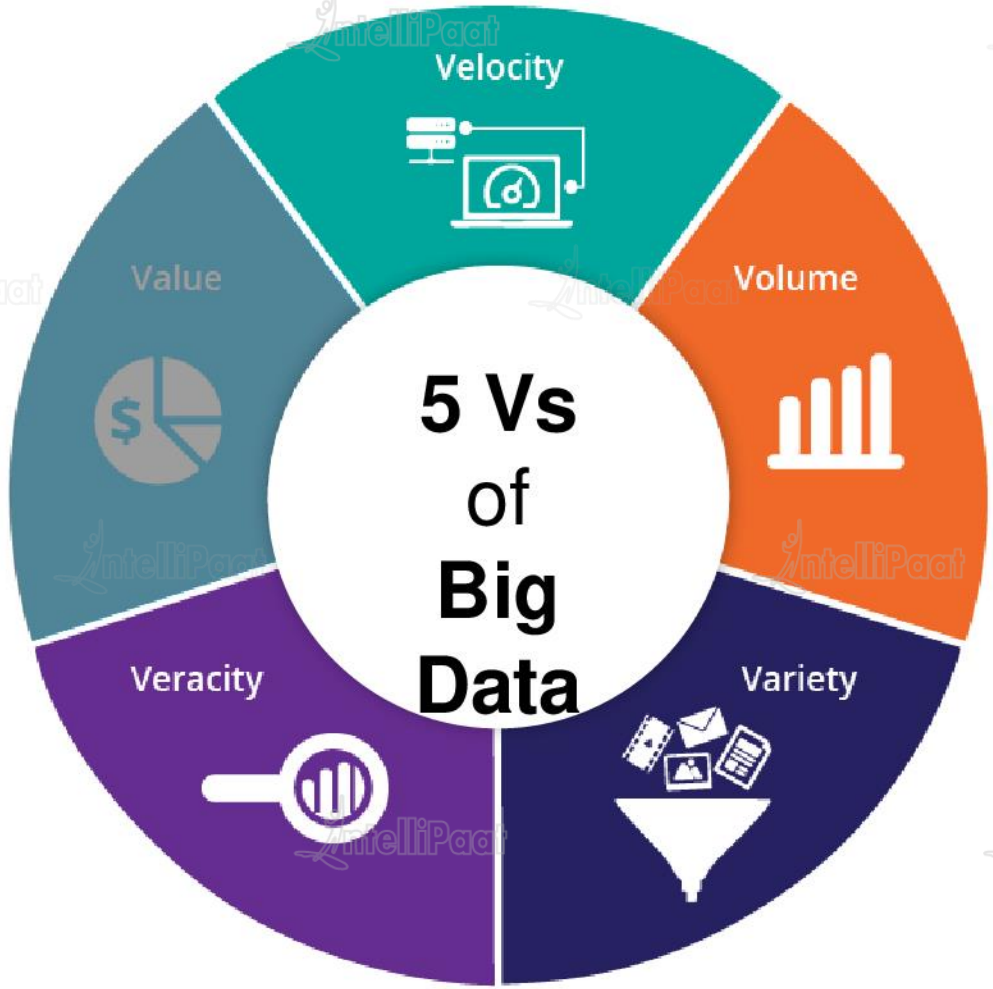
- It refers to the nature of data, i.e., whether the data is
  structured, semi-structured, or
  unstructured
- It also refers to the heterogeneous
  sources
- Variety is basically the arrival of data from various new
  sources
  both inside and outside of an
  enterprise



Copyright Intellipaat. All rights reserved.

5 Vs of Big Data: Veracity

- It refers to the inconsistencies and uncertainties in data
- Data can sometimes get messy; hence, quality and accuracy of the data are difficult to control
- Data in bulk can create confusion; whereas, less amount of data can convey half or incomplete information

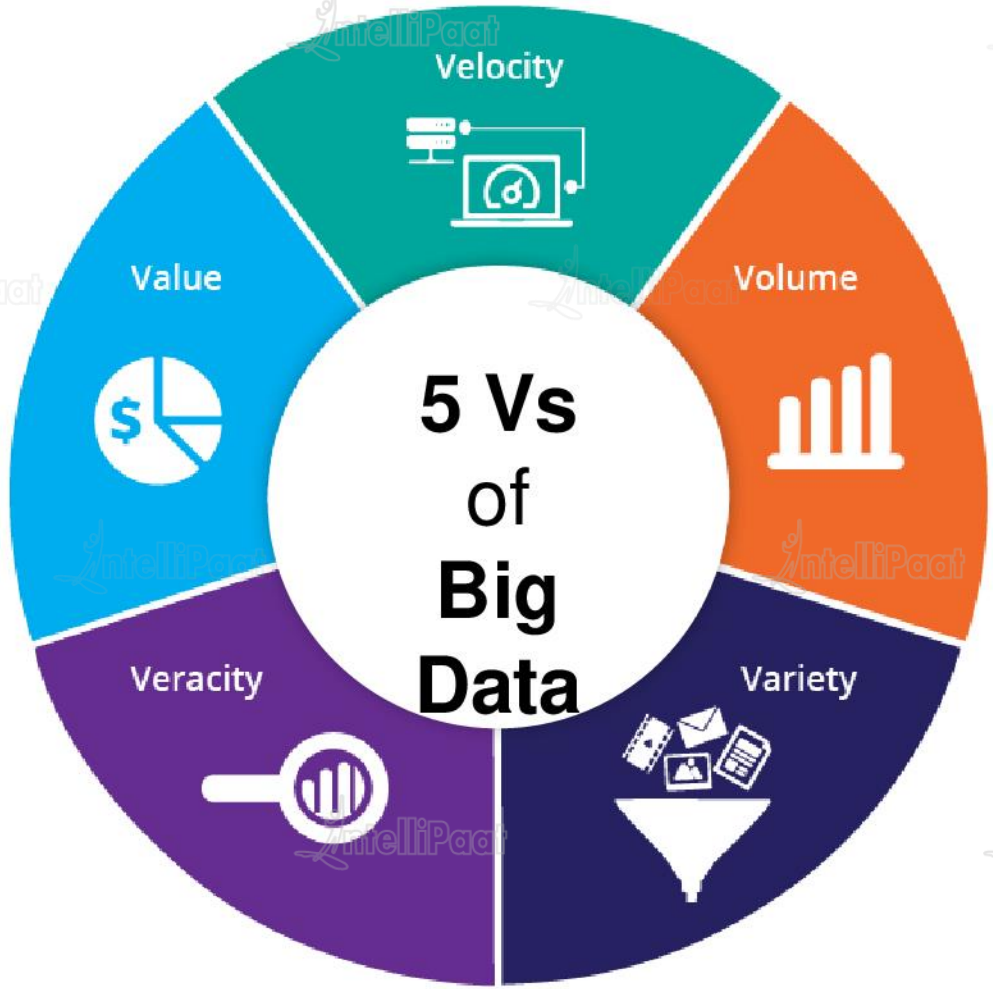


Copyright Intellipaat. All rights reserved.

5 Vs of Big Data: Value

- The bulk of data having no value is of no good to the company,

  unless we turn it into something useful
- Data (just by itself) is of no use or importance, but it needs to be
  converted into something valuable to extract information
- Hence, we can state that ‘value’ is the most important ‘V’ of all the
  5V
  s



Copyright Intellipaat. All rights reserved.

Problems with BigData

The good news is that Big Data is here!

The bad news is that we are struggling to **store** and **analyze** it

Problem 1: Hardware Failure

Although the storage capacities of hard drives have increased massively over the years, access speeds—the rate at which data can be read from the drives—have not kept up to the mark!

Problem 2: Combining Data

Most analyses require to combine data in some way; data read from one disk may need to be combined with the data from any of the other 99 disks—a challenging task for distributed systems!

Copyright Intellipaat. All rights reserved.

Problems with BigData

Solution for Hardware Failure

While we start using many pieces of hardware, the chance for one to fail is fairly high, and this might result in data loss.

A common way of avoiding data loss is through replication: creating multiple copies of the data so that, in the event of failure, there is another copy available.

This is what Hadoop does.

Hadoop's file system, **Hadoop Distributed File System (HDFS)**, creates multiple copies of the data

Solution for Combining Data

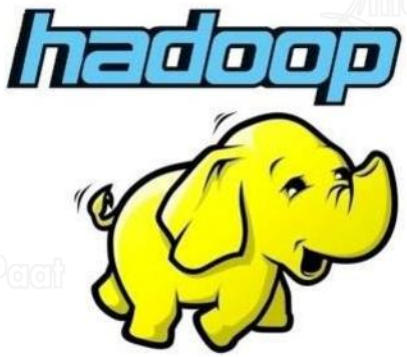
- **MapReduce** provides a programming model that abstracts the problem from the disk, reads and writes, and transforms it into a computation—map and reduce— over sets of keys and values
- MapReduce has a built-in reliability, an interface where the 'mixing' of map and reduce occurs

Copyright Intellipaat. All rights reserved.

What is Hadoop?

Copyright Intellipaat. All rights reserved.

What is Hadoop?



- Apache Hadoop is a **platform** used for storing Big Data across a spectrum of devices
- This is done to help us process **Big Data** in **parallel**
- It is made up of a distributed file system called **HDFS** and a computation layer called **MapReduce,** which is a processing paradigm
- Hadoop is an **open-source**, batch data processing system for dealing with enormous amounts of data

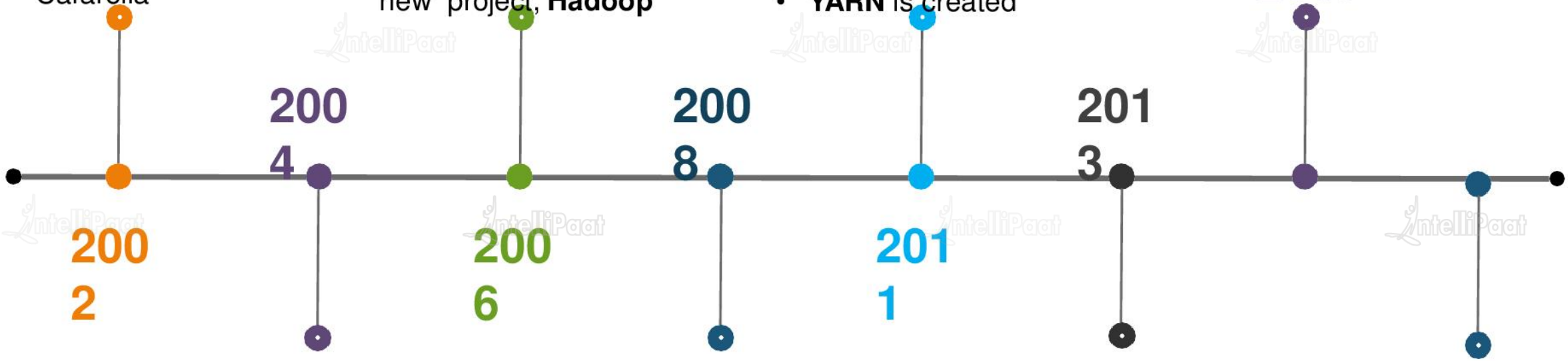
Goals/Requirements

Abstracting and facilitating the storage and processing of large and/or rapidly growing datasets

- Structured and non-structured data
- Simple programming models

Copyright Intellipaat. All rights reserved.

History of Hadoop



- Apache Nutch is created by Doug Cutting and Mike Cafarella

- Nutch Distributed File System **(NDFS)** is established

- Doug Cutting joins Yahoo
- **NFDS** and **MapReduce** are pulled out of Nutch to be clubbed under a new project, **Hadoop**

- Nutch gets divided and Hadoop is born
- Yahoo releases Hadoop as open-source project to ASF
- Hadoop-based startup, **Cloudera** is incorporated

- Many sub-projects, namely, **HBase** and **Hive**, are established
- **Hortonworks** is stablished
- **YARN** is created

- Greenplum releases Hadoop distribution

- Apache releases new Hadoop version, Hadoop3.x

- Apache releases Hadoop 3.2

Copyright Intellipaat. All rights reserved.

Use Cases of Hadoop

Using **"Know Me Program"**, complete information of travelers was retrieved, such as, when they are planning to travel, where they are planning to travel, etc., leading to the **increase in sales** and hence achieving **a hike in revenue.**

Uber

oop, Uber was able to achieve **100+ petabytes of data within a minute.**

d Uber in increasing their network in several countries and hence provide

service with zero or negligible latency.

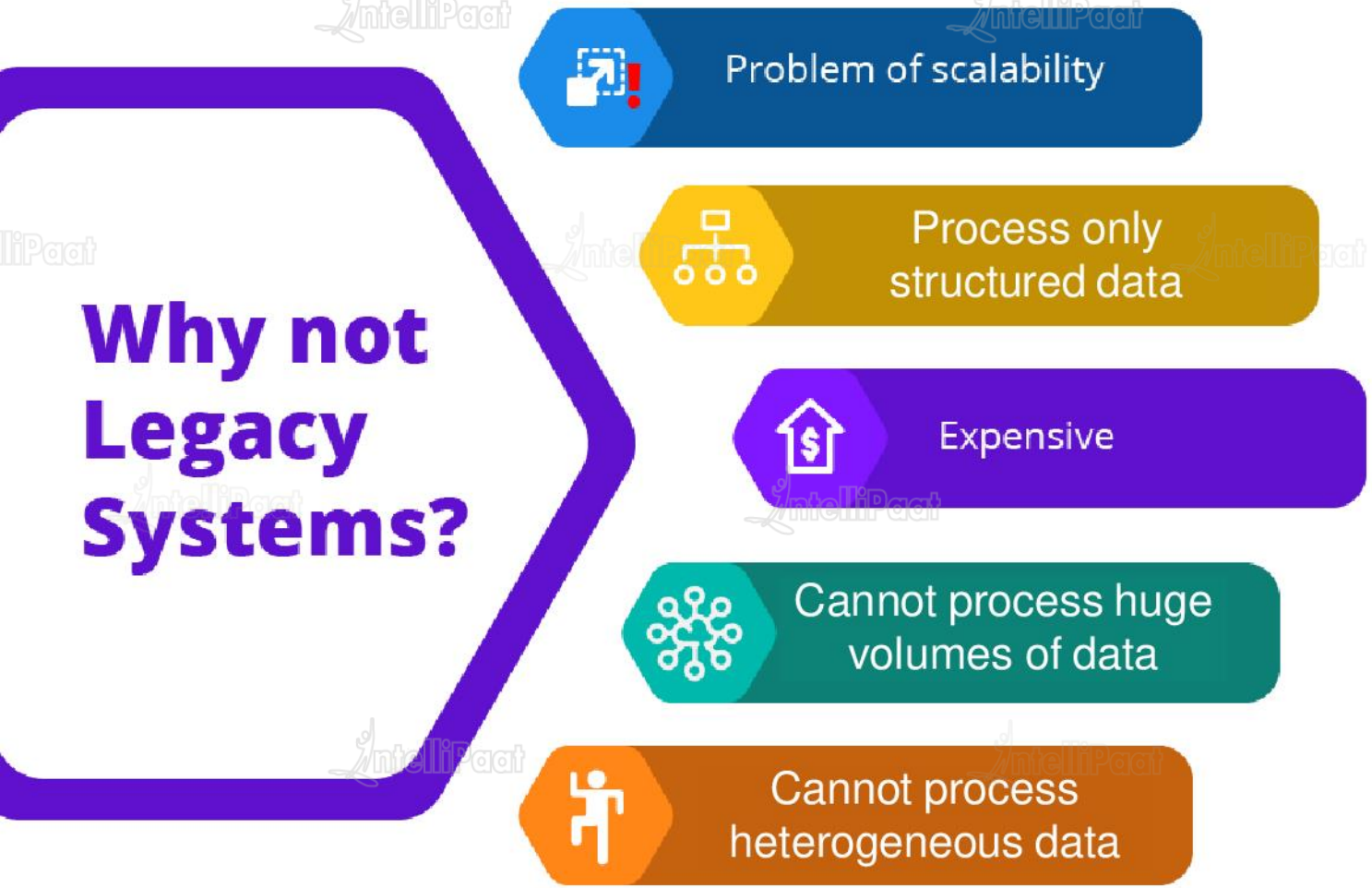
Amazon Web Services uses the open-source Apache Hadoop distributed computing technology **to make it easier to access large amounts of computing power to run data-intensive tasks**.

Copyright Intellipaat. All rights reserved.

Why Hadoop is asolution?

Copyright Intellipaat. All rights reserved.

Why not Legacy Systems?



Copyright Intellipaat. All rights reserved.

Why Hadoop?

Copyright Intellipaat. All rights reserved.

Why Hadoop?

Flexible

- A major challenge for any company or organization is the handling of

  structured and **unstructured data** available in Big Data

- Hadoop is very **flexible** and handles structured and

  unstructured data efficiently

- Hadoop can also handle **encoded data** and process it

  according to company needs

Copyright Intellipaat. All rights reserved.

Why Hadoop?

Scalable

- Additional nodes can be added when the maximum size of the **storage nodes** is reached, making the Hadoop framework easily scalable across the spectrum
- The nodes are **independent** of each other, so adding a new node to the cluster will not be difficult

Copyright Intellipaat. All rights reserved.

Why Hadoop?

Fault Tolerant

- Whenever data is stored in HDFS, it gets **replicated three times** at **multiple locations** in the cluster
- So, even if one or two of the systems collapse, the file will still be available on the backup system so that it can be easily retrieved
- Hadoop is **bulletproof** in terms of fault tolerance

Copyright Intellipaat. All rights reserved.

Why Hadoop?

High Speed Data Processing

- Hadoop works on the concept of **parallel processing**
- It can perform batch processes **10 times faster** than on a singlethreaded server or on the mainframe
- Hadoop is a comparatively **cheaper** and **cost-effective** way of handling Big Data than the other frameworks

In the 2014, a team from Databricks used a 207-node Spark cluster to sort 100 TB of data in 1,406 seconds, i.e., 4.27 TB in 1 minute!

Copyright Intellipaat. All rights reserved.

Hadoop vs RDBMS

Used for structured, semi- Structured, and unstructured data

Mainly used for structured data

Used for large datasets (TBs and PBs)

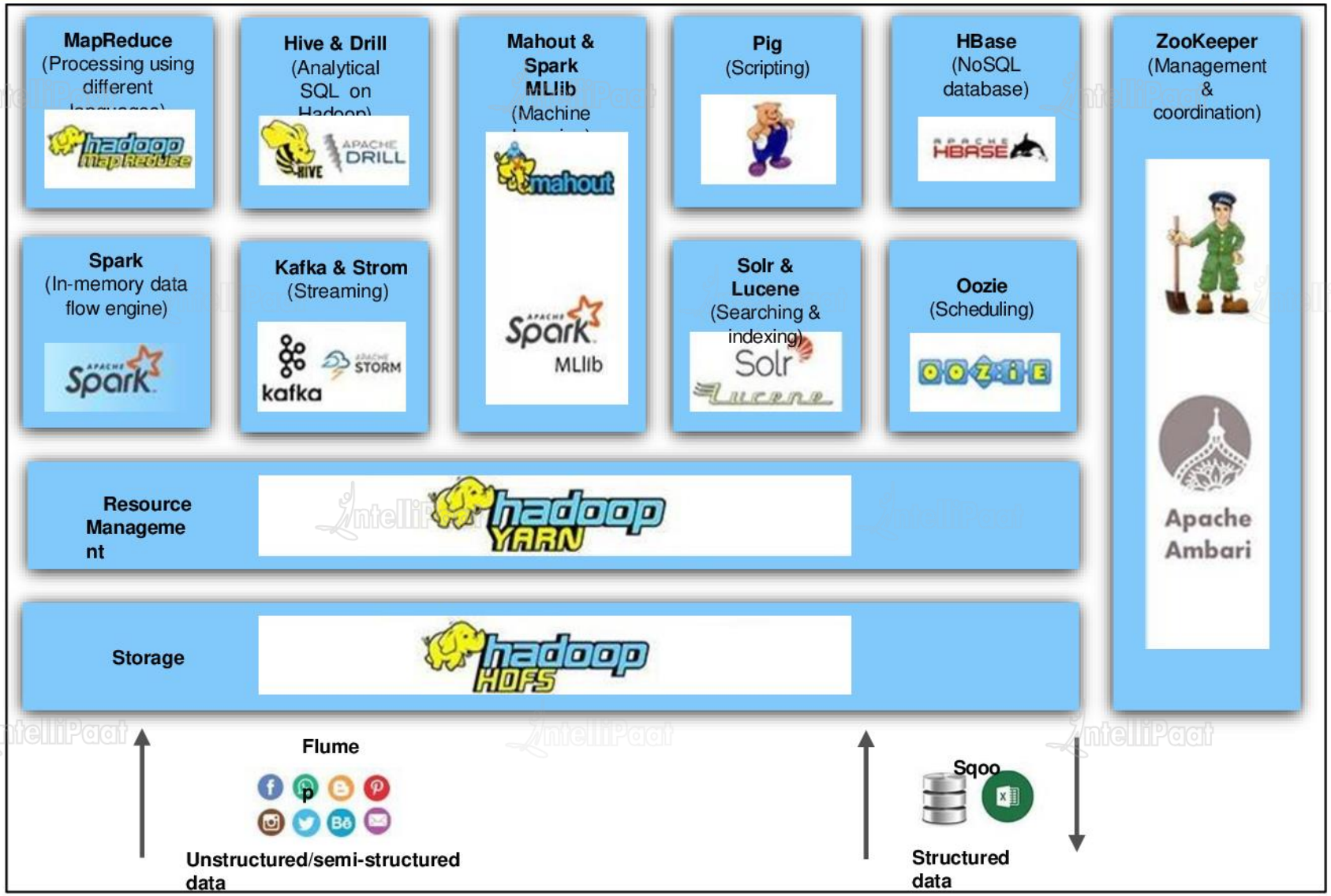
Used for average sized data (GBs)

Reads and writes data quickly

Only reads data quickly

Copyright Intellipaat. All rights reserved.

Hadoop Ecosystem



Copyright Intellipaat. All rights reserved.

Core Components: HDFS

- Hadoop Distributed File System (HDFS) is a storage system that runs on **Java** programming language, and it is used as the primary storage device in Hadoop applications

- HDFS consists of two components, **NameNode** and **DataNode**

- These are used to store large data across **multiple nodes** on the Hadoop cluster

Copyright Intellipaat. All rights reserved.

Core Components: MapReduce

- **MapReduce** acts as a **core component** in Hadoop Ecosystem as it facilitates the logic of processing
- It is a **software framework**, which enables us in writing applications that process large datasets using **distributed** and **parallel**

  algorithms in a Hadoop environment
- The parallel processing feature of MapReduce plays a **crucial role** in Hadoop ecosystem. It helps in performing Big

  Data analysis using **multiple machines** in the same cluster
- **Map function:** It converts one set of data into another, where individual elements are broken down into tuples

  (key/value pairs)
- **Reduce function:** It takes data from the Map function as an input, and it aggregates and summarizes the results

  produced by the Map function

Copyright Intellipaat. All rights reserved.

Core Components: YARN

- YARN supports a **variety** of processing engines and applications
- It distributes its duties across **multiple components**
- **Dynamically**, it allocates pools of resources to other applications on the go

- YARN supports **multiple scheduling methods** based on the queue format for submitting the processing jobs
- The default scheduler works on **FIFO**

Copyright Intellipaat. All rights reserved.

Hadoop 2.x vs Hadoop 3.x

Hadoop 2.x	Hadoop 3.x
 Consumes 200% of storage in HDFS	 Consumes just 50% of storage in HDFS
Scalability is limited (Up to 10,000 nodes in a cluster)	Scalability has improved (Over 10,000 nodes in a cluster)
Supports DFS, FTP, and Amazon S3 file system	All features plus Microsoft Azure Data Lake File system are supported
Uses HDFS balancer for data balancing	Uses intra-data node balancer for data balancing

Copyright Intellipaat. All rights reserved.

What is HDFS?

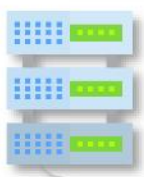
Copyright Intellipaat. All rights reserved.

What is HDFS?

HDFS is a file system designed for storing very **large files** with **streaming data access**
patterns, running on clusters of **commodity hardware**

'**Very large files**,' in this context, means the files that are of hundreds
of gigabytes, terabytes, or petabytes in size.

Streaming Data Access: Each analysis of the dataset involves a large portion of the dataset, so the time to read the whole dataset is more important than the latency in reading the first record.



Commodity Hardware: Hadoop is designed to run on clusters of commodity hardware, where the chance of node failure is high. HDFS is designed to carry on working without a noticeable interruption in the face of such a failure.

Copyright Intellipaat. All rights reserved.

HDFS Design Factors

Factors involved in the design of HDFS:

1. High throughput
2. Handling large datasets (TBs and PBs)
3. Hundreds/thousands of nodes
4. Portability across heterogeneous hardware/software

Copyright Intellipaat. All rights reserved.

Approach to Meet HDFS Design Goals

How was the design of HDFS altered to meet its design goals?

Simplified Coherency Model: Write once read many

times

Move Computation Close to Data: Not moving data around and hence improving efficiency

Data Replication: Helps in handling hardware failure

Copyright Intellipaat. All rights reserved.

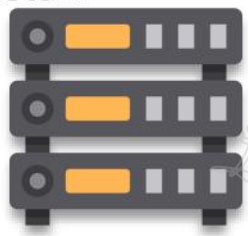
HDFS: Why have another file system?

Copyright Intellipaat. All rights reserved.

Need forHDFS

- HDFS **divides** Big Data into **several chunks** and **processes each chunk of data separately** on separate machines
- Since all the machines run on commodity hardware, it is very cheap

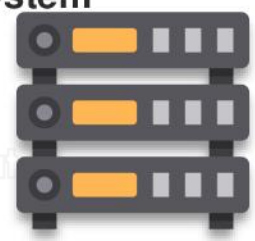
Centralized File System



1 Server
4 I/O Channels (100 MB/s)

Reads **1 TB** data in **43 minutes**

Distributed File System



X
10

10 Servers
4 I/O Channels (100 MB/s)

Reads **1 TB** data in **4.3 minutes**!

Copyright Intellipaat. All rights reserved.

How did HDFS solve the problem of Big Data?

- HDFS uses the **MapReduce** method for accessing data, which

  is **very fast**

- HDFS follows the **data coherency** model, in which the data is

  synchronized across the server. It is very simple to

  implement and **highly robust and scalable**

- HDFS is compatible with any kind of **commodity hardware**

  and operating system processors

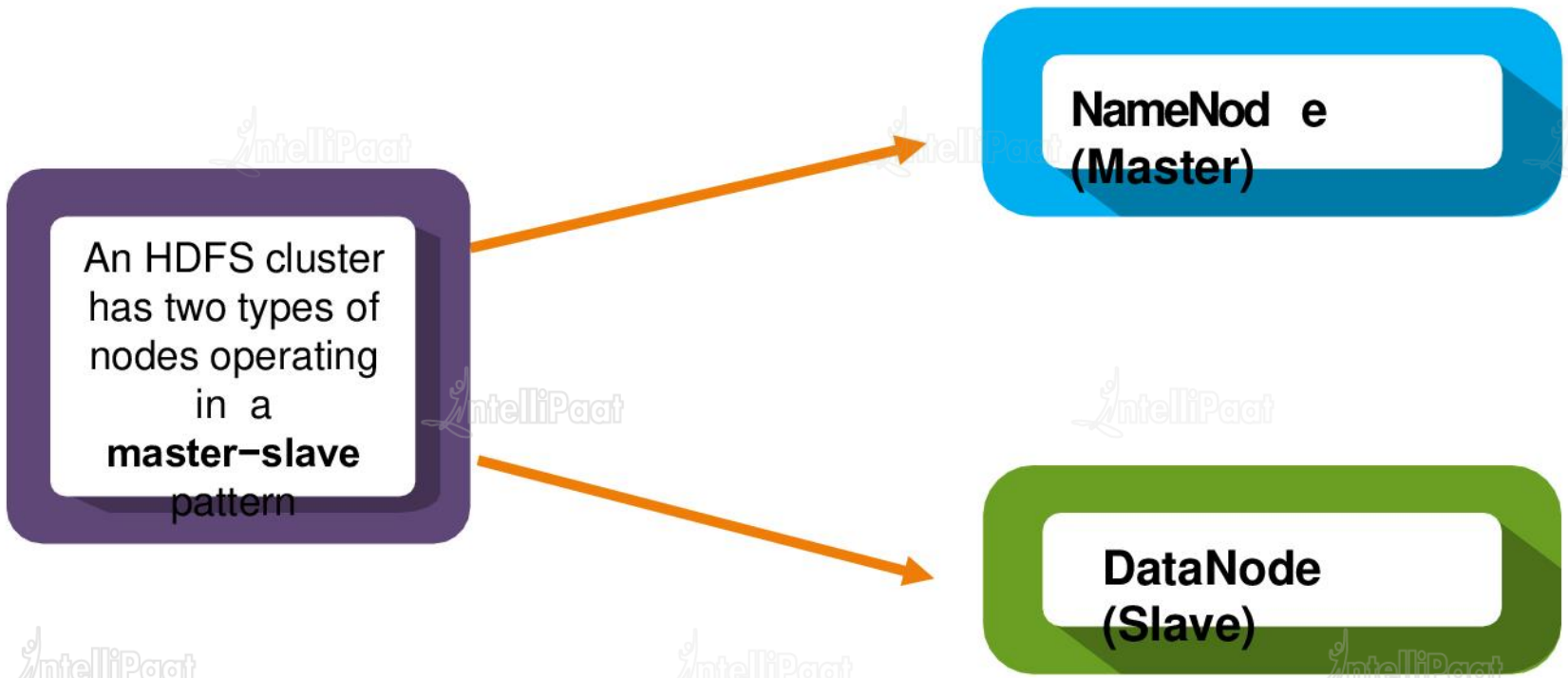
- As data is saved in multiple locations, HDFS is **fault**

Copyright Intellipaat. All rights reserved.

HDFS Components

Copyright Intellipaat. All rights reserved.

HDFS Components: Nodes



Copyright Intellipaat. All rights reserved.

HDFS Components: NameNode

- The NameNode **manages** the file system namespace and **maintains** the file system tree and the metadata for all files and directories in the tree
- This information is stored persistently on the local disk in the form of two files: **namespace image** and **edit log**
- The NameNode also knows the DataNodes on which all blocks for a given file are located. It **doesn't store block locations** persistently, as this is retrieved from DataNodes whenever the requirement pops up

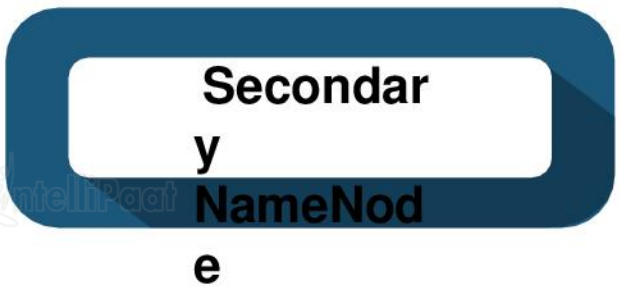
Copyright Intellipaat. All rights reserved.

HDFS Components: DataNode

- DataNodes are responsible for storing the actual data in HDFS
- They are the **workhorses** of the file system
- It **stores** and **retrieves** blocks whenever required
- It **reports back** to the NameNode periodically with the list of blocks it is storing
- When a DataNode starts up, it notifies the NameNode, along with the list of blocks it is responsible for
- The NameNode arranges the **replication of blocks** managed by the DataNode that is not available

Copyright Intellipaat. All rights reserved.

HDFS Components: Secondary NameNode



- Secondary NameNode doesn't act as a 'secondary' NameNode
- Its main role is to periodically **merge** the namespace image with the edit log to prevent the edit log from becoming too large
- It runs on a separate physical machine as it requires plenty of CPU and memory for performing the merge
- It keeps a copy of the namespace image that can be used when the NameNode fails
- In case a NameNode fails, the Secondary NameNode will copy the NameNode's metadata files to itself and will run as the new primary NameNode

Copyright Intellipaat. All rights reserved.

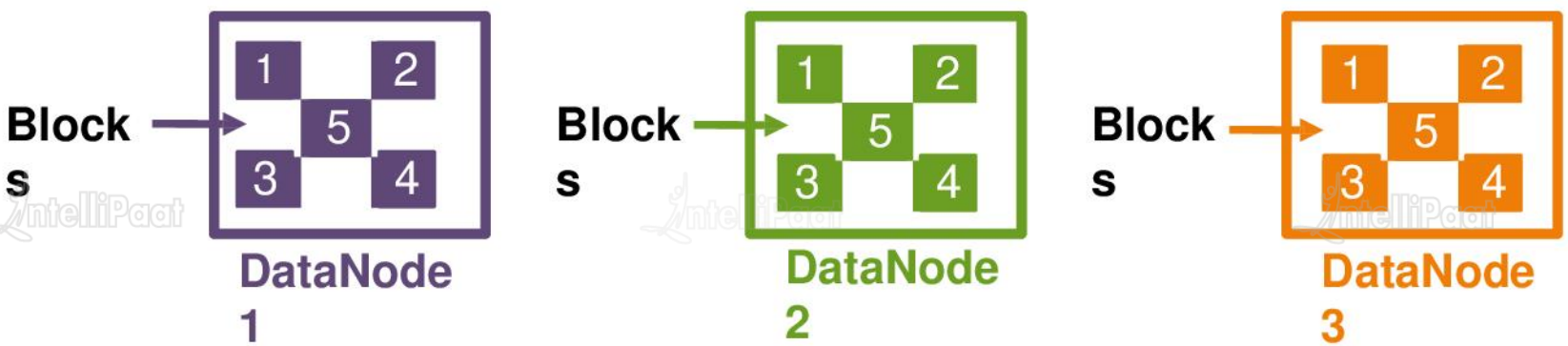
HDFS Components: Blocks

- The **smallest** unit writable by a disk or file system
- Everything a file system does is composed of operations done

  on blocks
- Blocks are stored on DataNodes

Default block size in HDFS: 128 MB

Can the block size of HDFS be increased

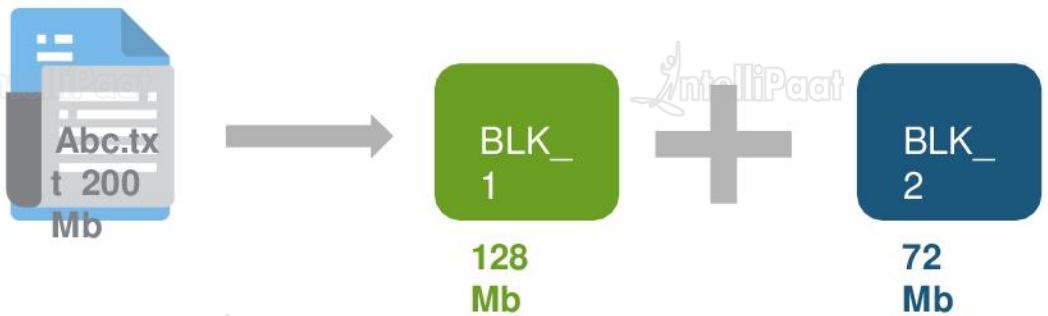


Copyright Intellipaat. All rights reserved.

HDFS Components: Blocks

Data is **stored** in HDFS in **several blocks**, with a maximum size **128 MB**

This is how it is:



Why is a Block in HDFS so large?

- To minimize the cost of seeks
- Transferring large files made up of multiple blocks operates at the disk transfer rate

Copyright Intellipaat. All rights reserved.

HDFS Components: Blocks

Block Replication

- To avoid **fault tolerance**, blocks are replicated inside

  multiple DataNodes present in the cluster
- To insure against corrupted blocks and disk and

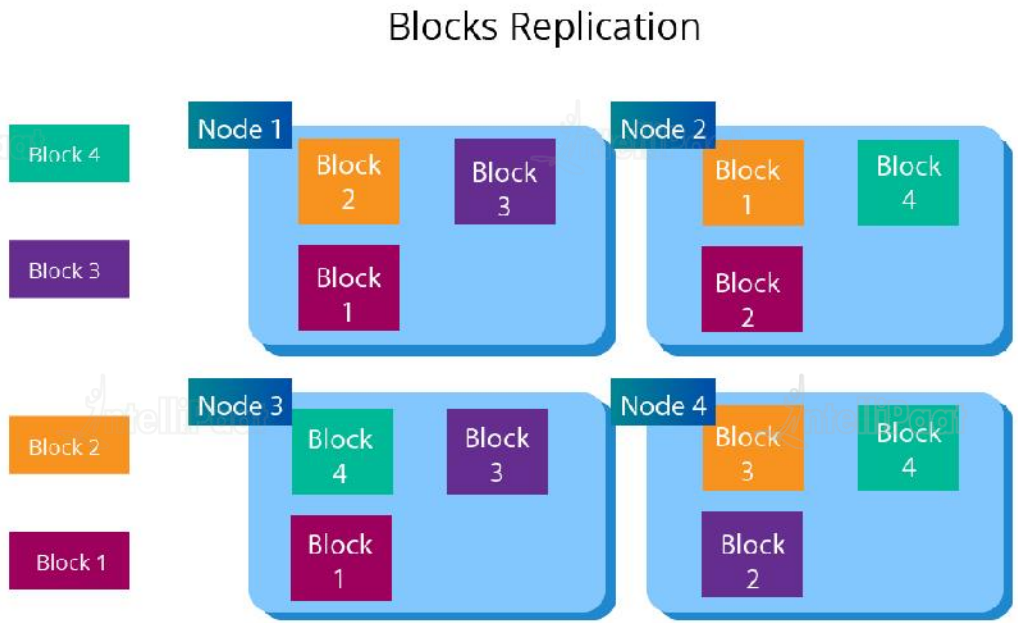
  machine failure, each block is replicated to a small

  number of

  physically separated machines (typically **three**)
- If a block is unavailable (due to loss of connection),

  a copy can be read from another location in a way

  that is transparent to the client

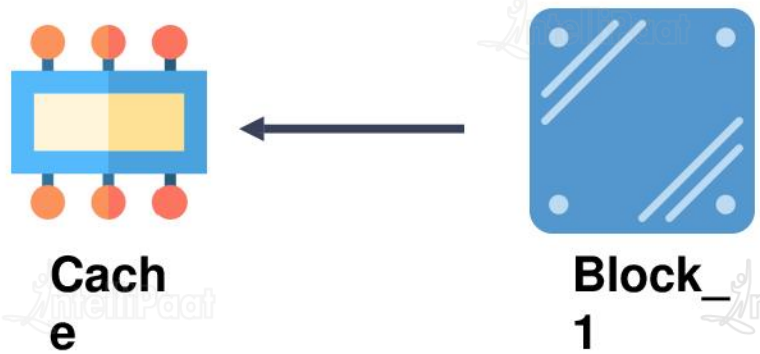


Copyright Intellipaat. All rights reserved.

HDFS Components: Blocks

Block Caching

- For frequently accessing files, blocks are explicitly cached in a DataNode's memory
- Hadoop frameworks take advantage of cached blocks by running tasks on the DataNode where a block is cached, for increased read performance

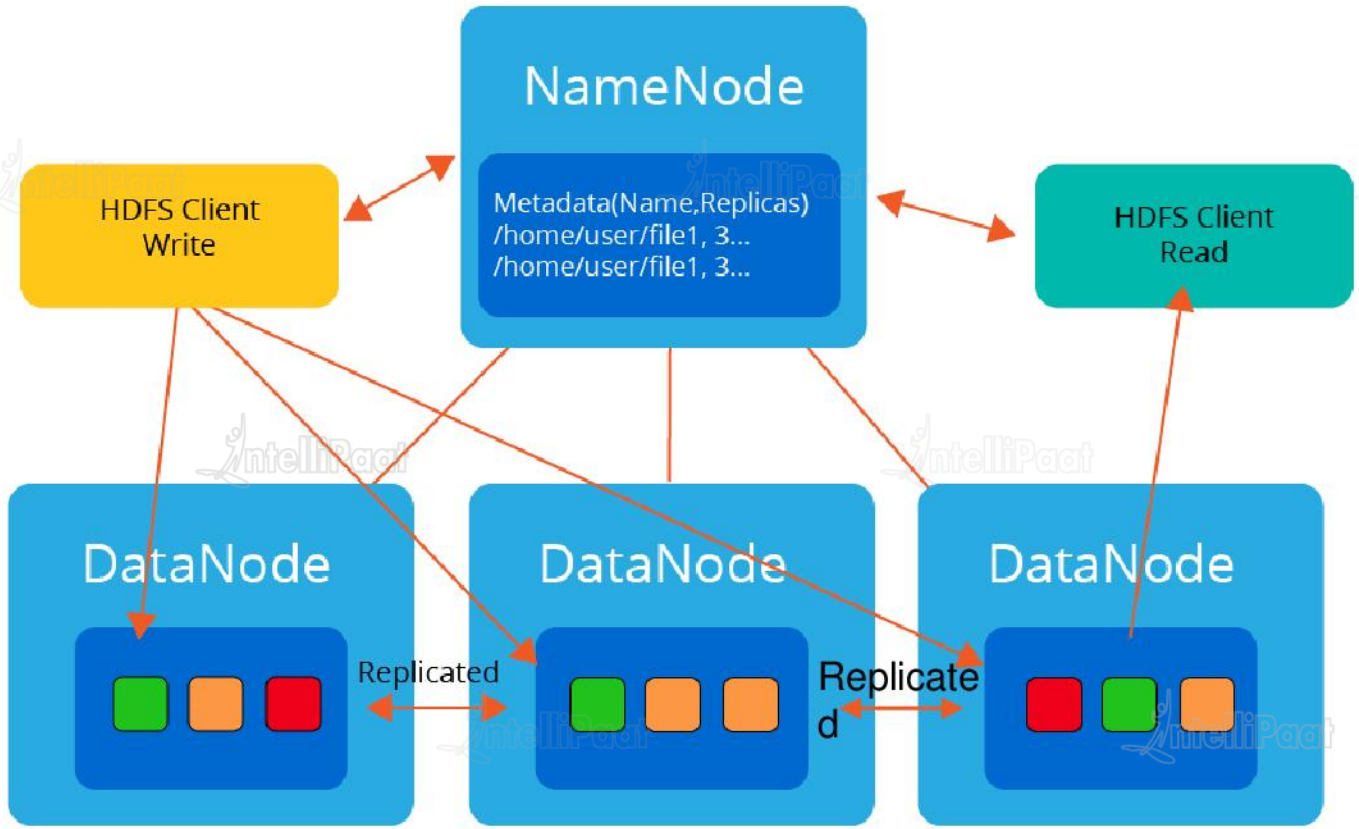


Copyright Intellipaat. All rights reserved.

HDFS Architecture

Copyright Intellipaat. All rights reserved.

HDFS Architecture

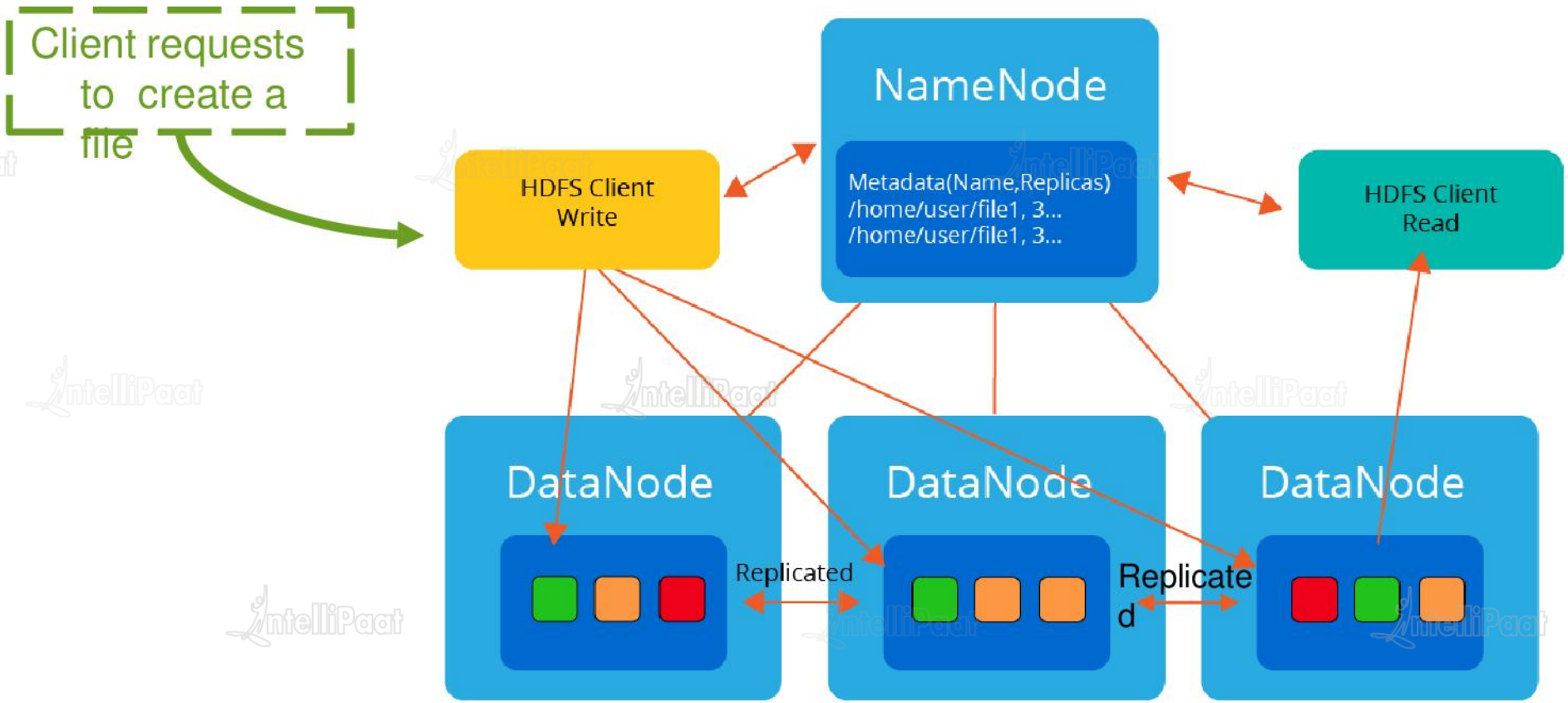


Copyright Intellipaat. All rights reserved.

Writing Data in HDFS

Copyright Intellipaat. All rights reserved.

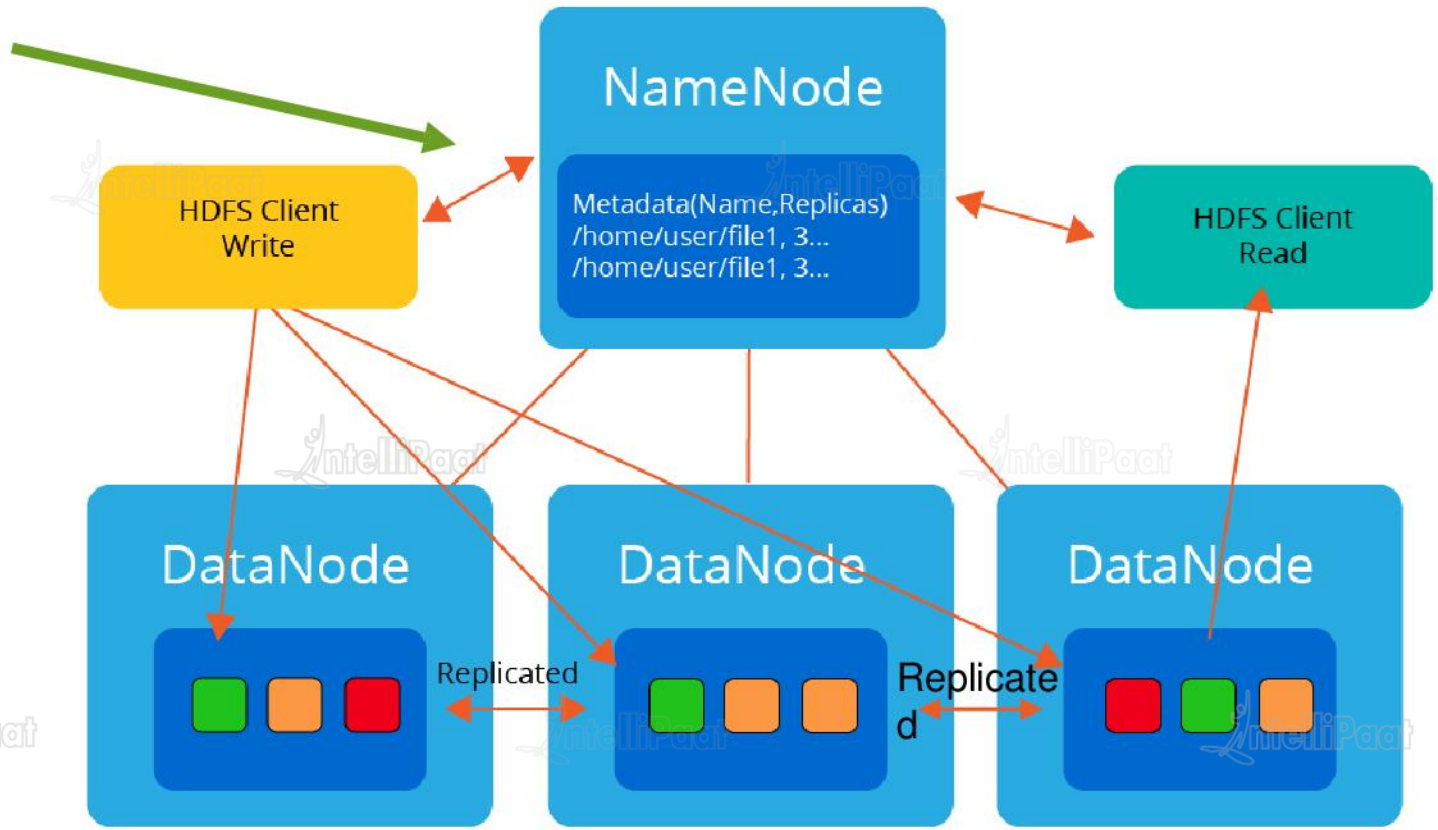
How is data written in HDFS?



Copyright Intellipaat. All rights reserved.

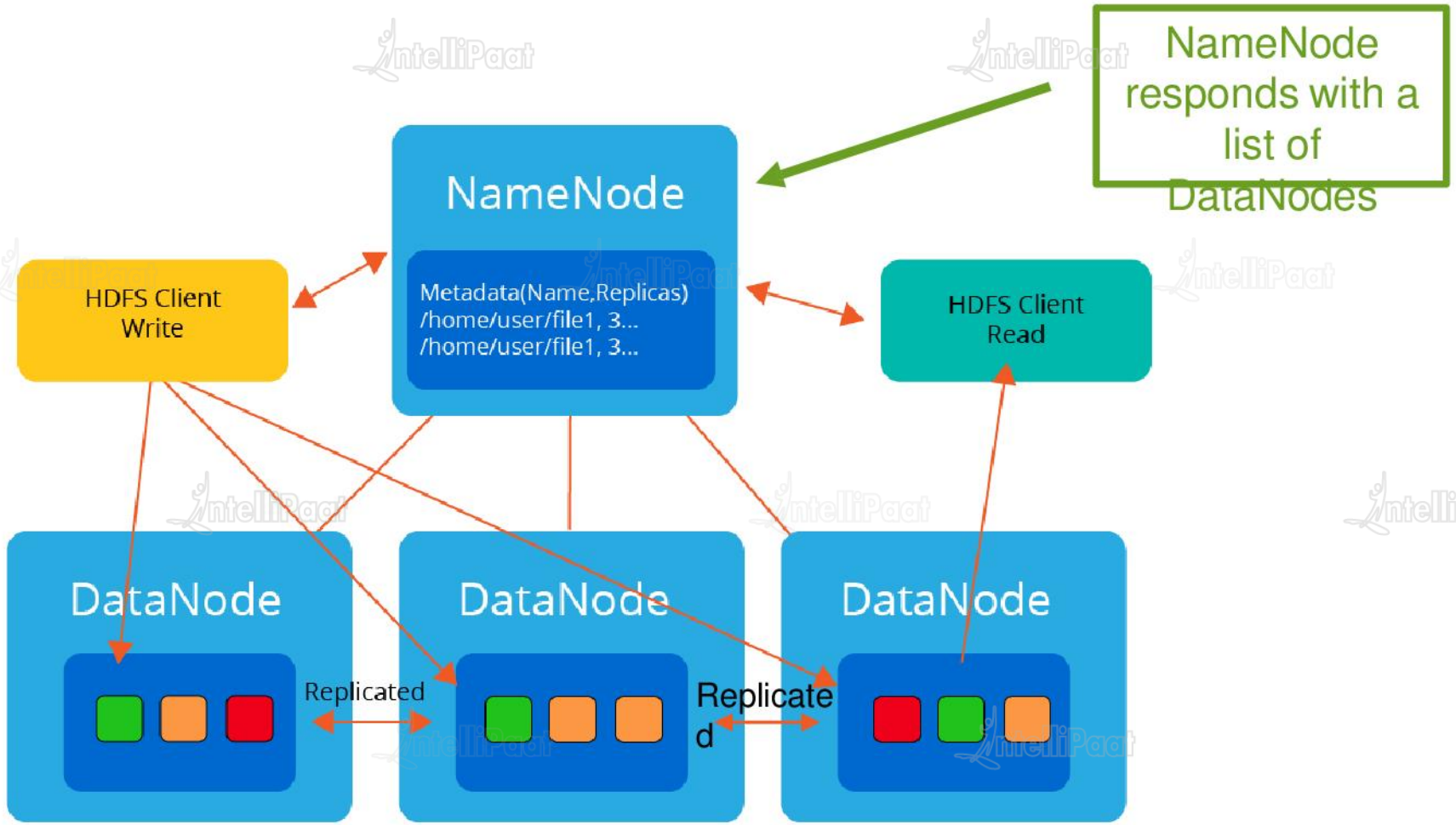
How is data written in HDFS?

NameNode is contacted once a block of data is accumulated



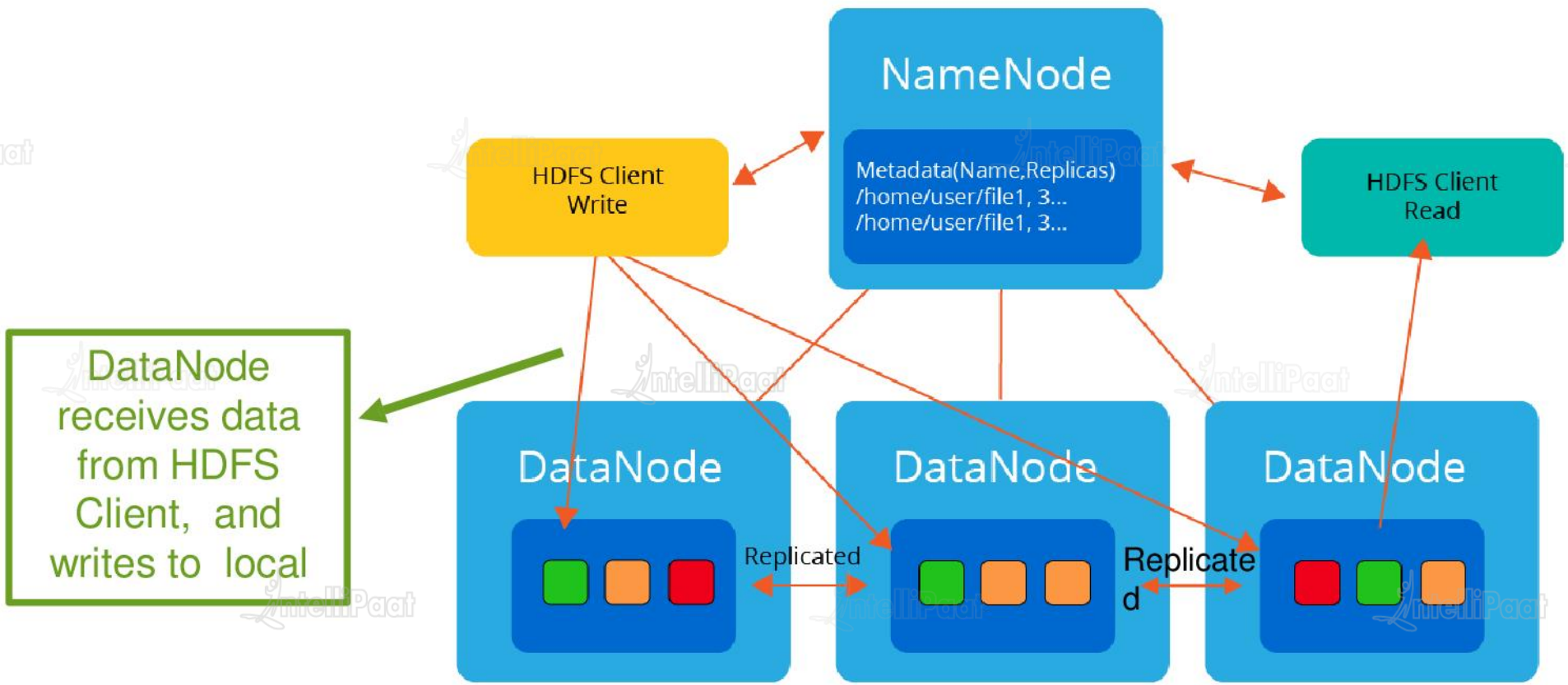
Copyright Intellipaat. All rights reserved.

How is data written in HDFS?



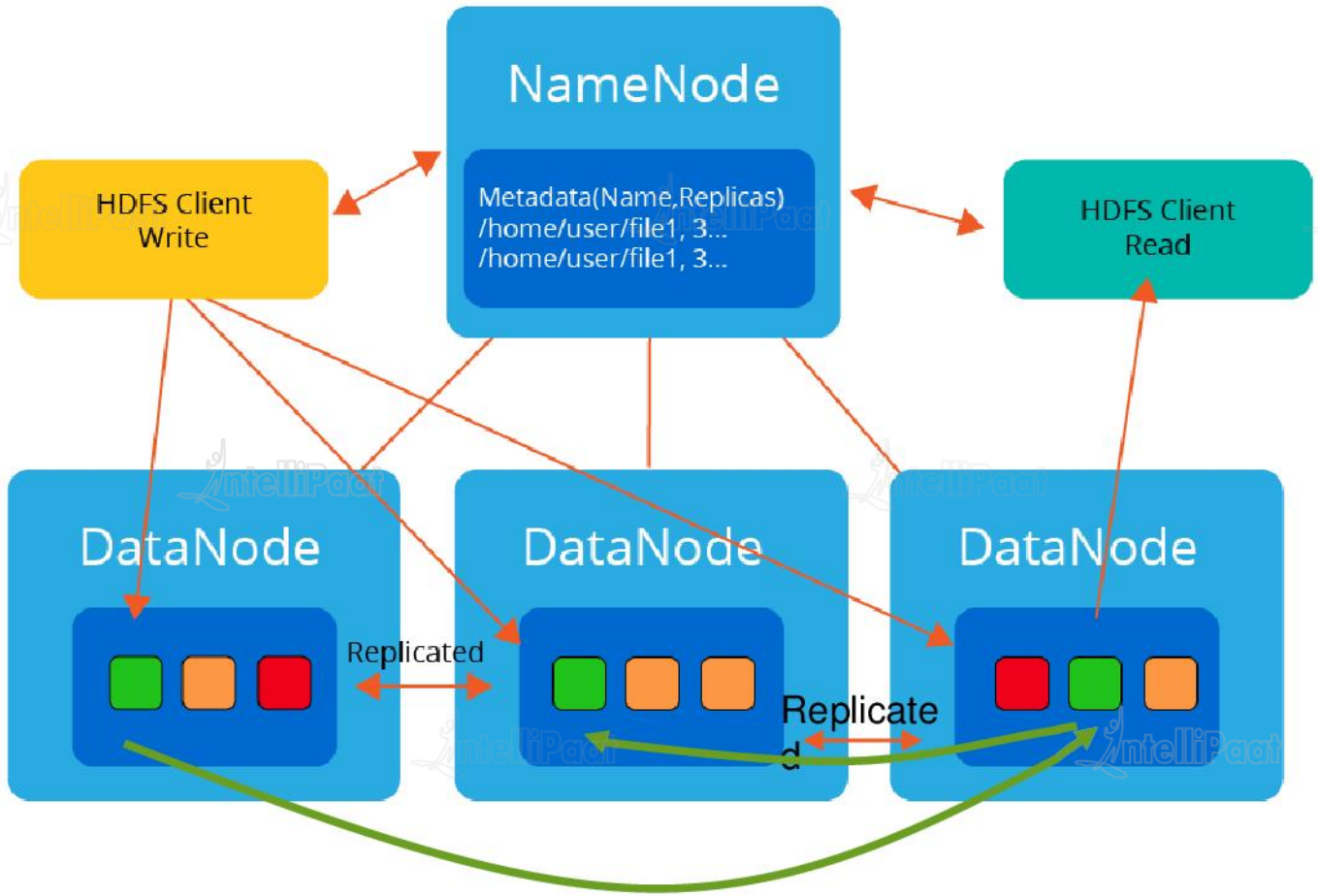
Copyright Intellipaat. All rights reserved.

How is data written in HDFS?



Copyright Intellipaat. All rights reserved.

How is data written in HDFS?

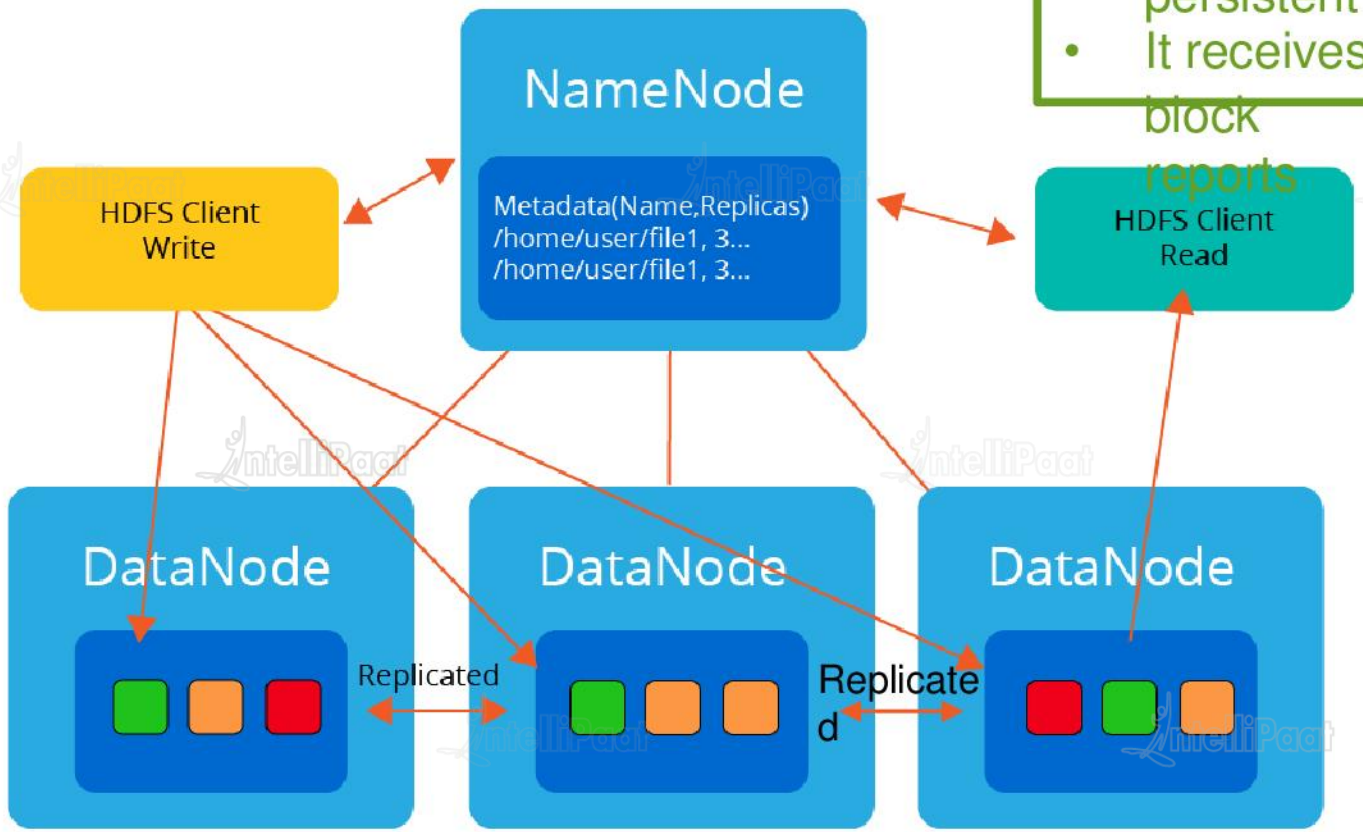


DataNode forwards the file to a second DataNode for replication and then to a third

Copyright Intellipaat. All rights reserved.

How is data written in HDFS?

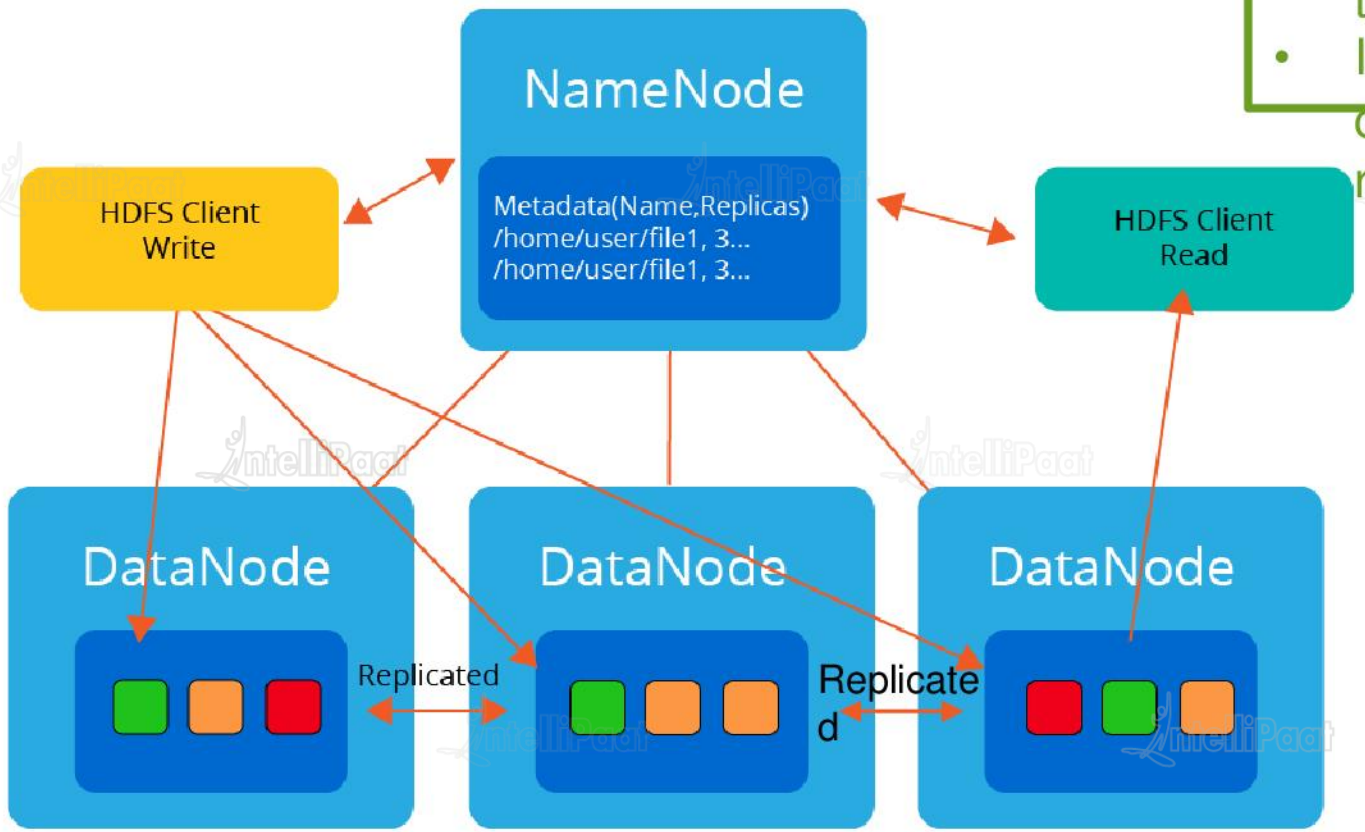
- NameNode commits the file creation into persistent store
- It receives heartbeat and block reports



Copyright Intellipaat. All rights reserved.

How is data written in HDFS?

- Client gets the DataNode list from the NameNode
- It reads from the replica closest to the render

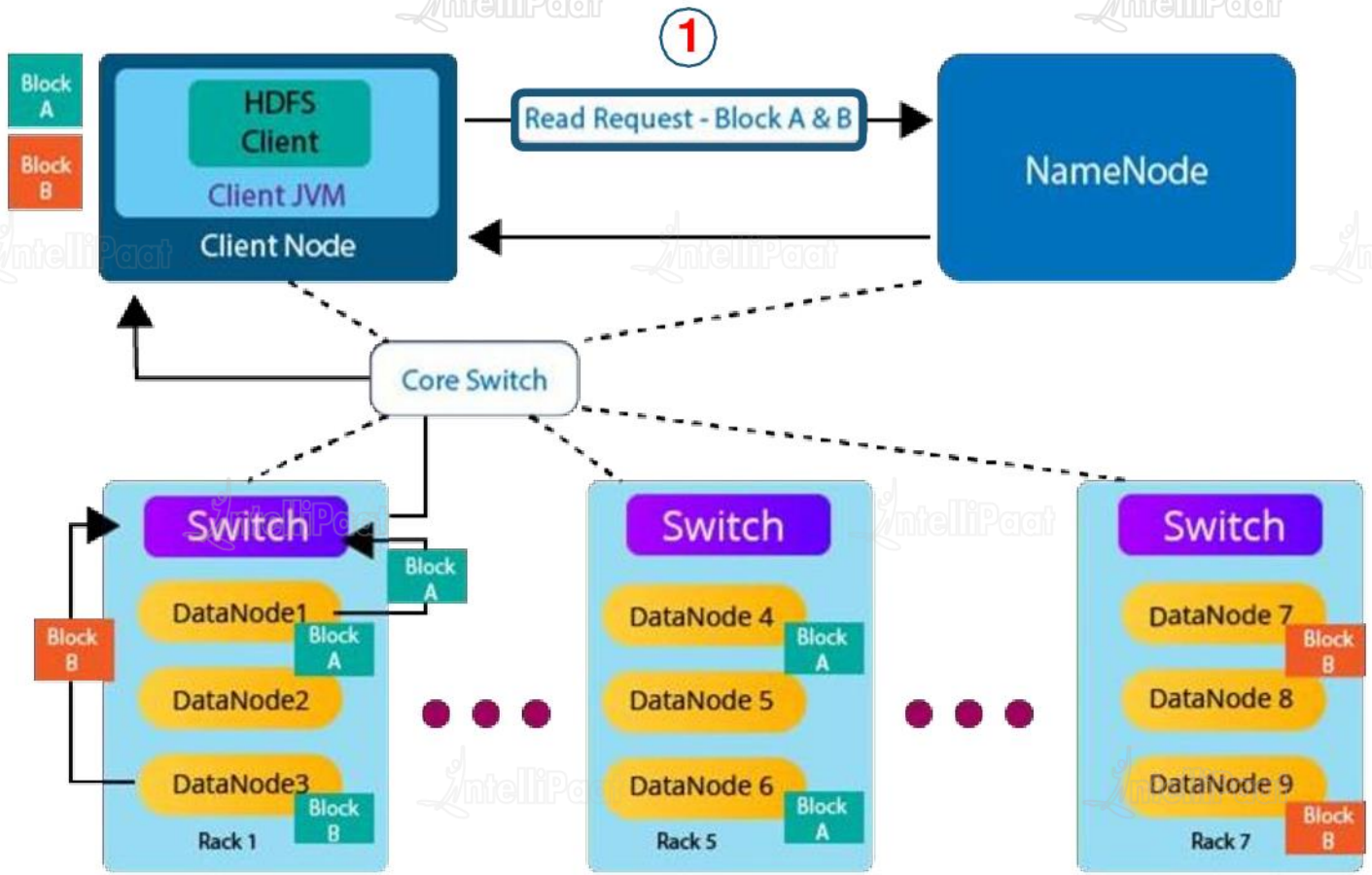


Copyright Intellipaat. All rights reserved.

Reading Data in HDFS

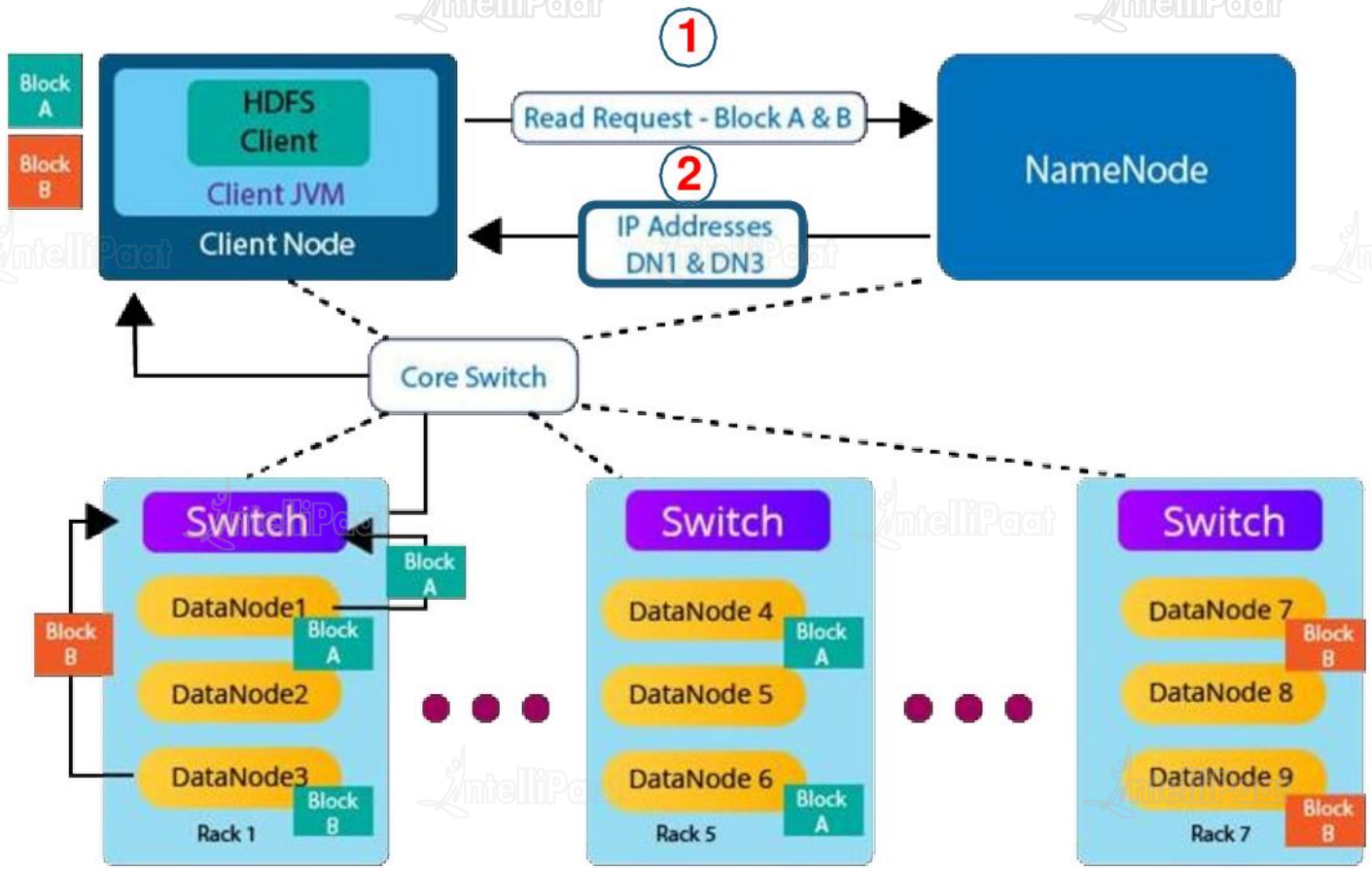
Copyright Intellipaat. All rights reserved.

How is data read in HDFS?



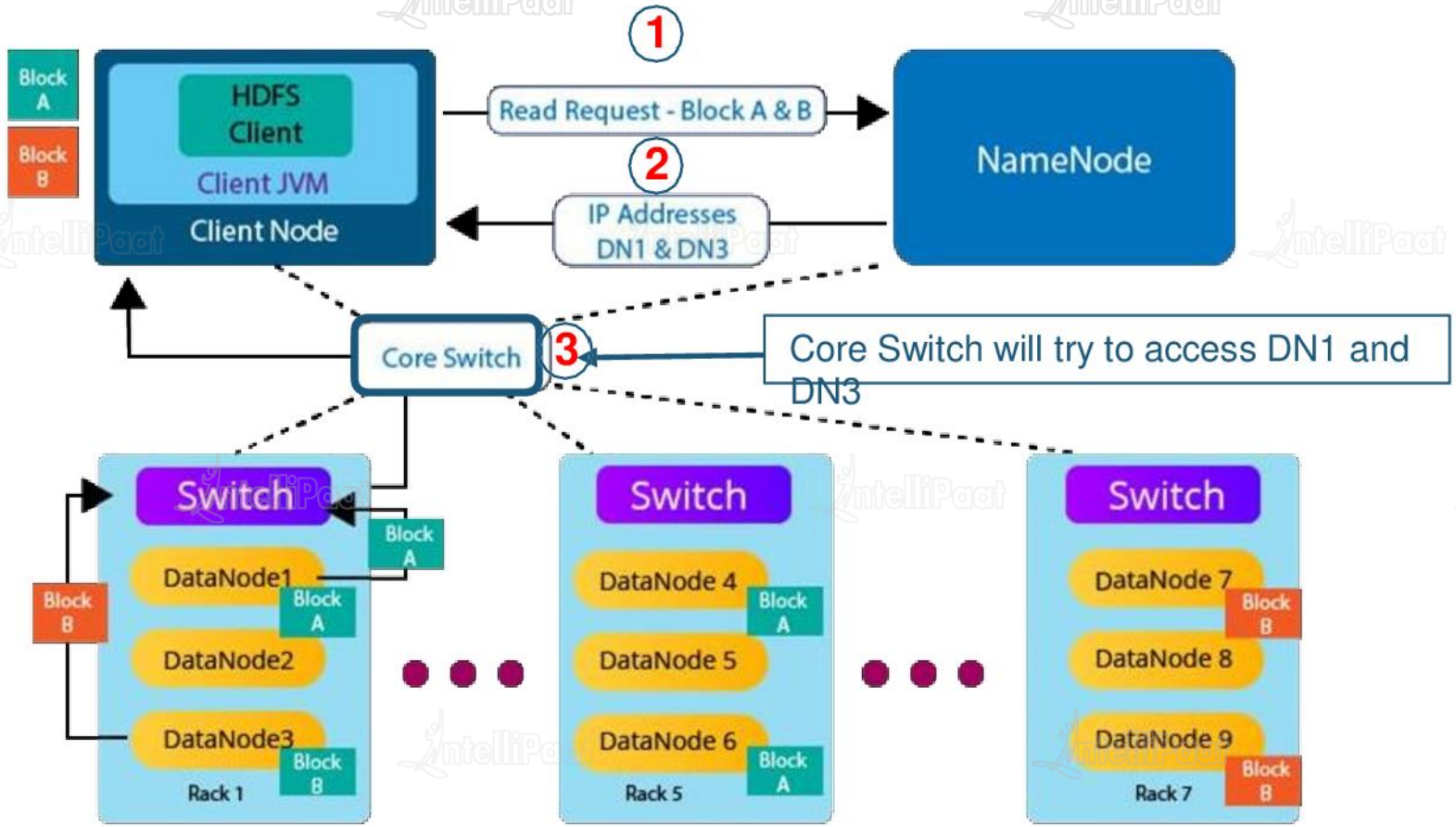
Copyright Intellipaat. All rights reserved.

How is data read in HDFS?



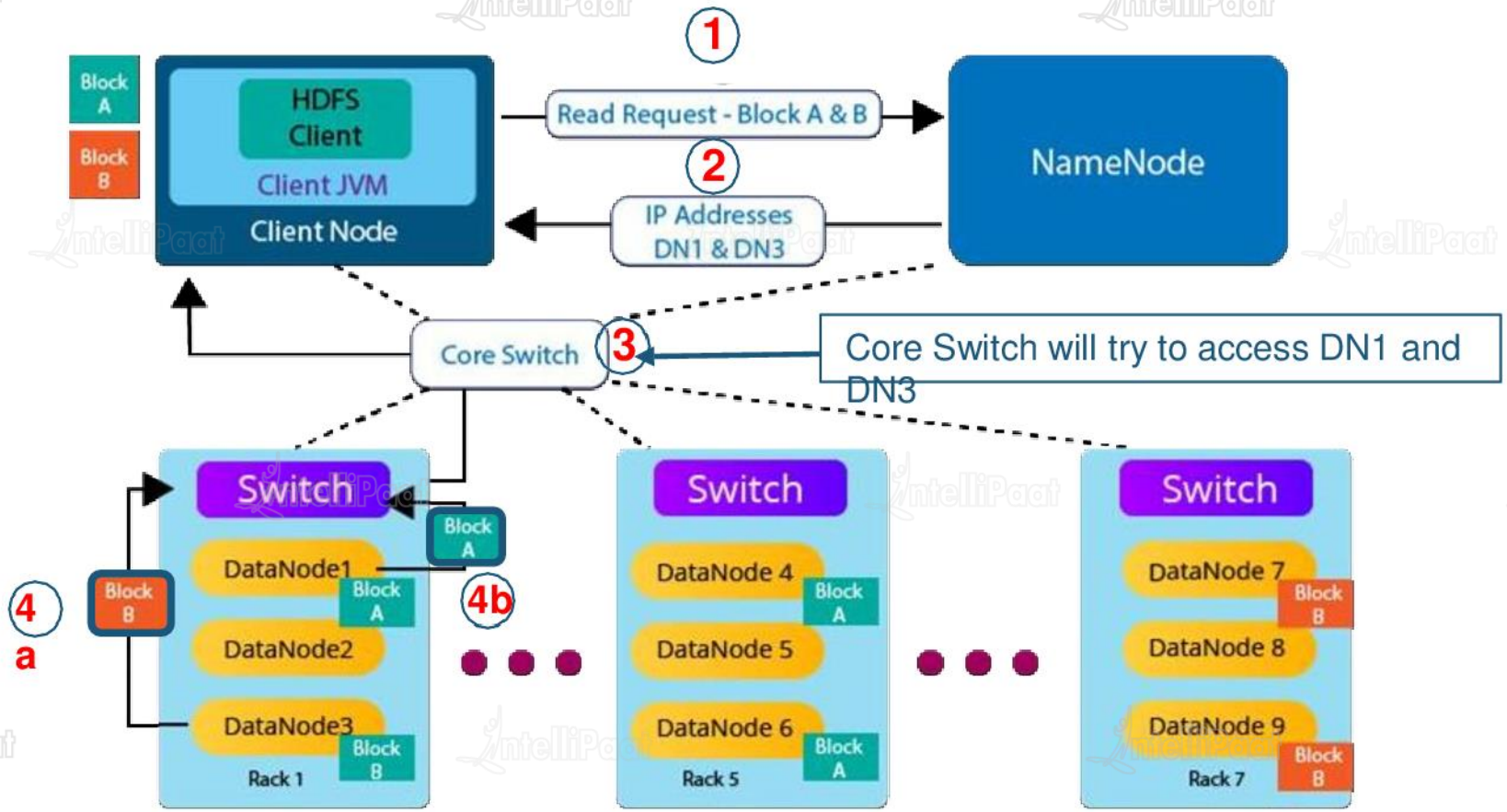
Copyright Intellipaat. All rights reserved.

How is data read in HDFS?



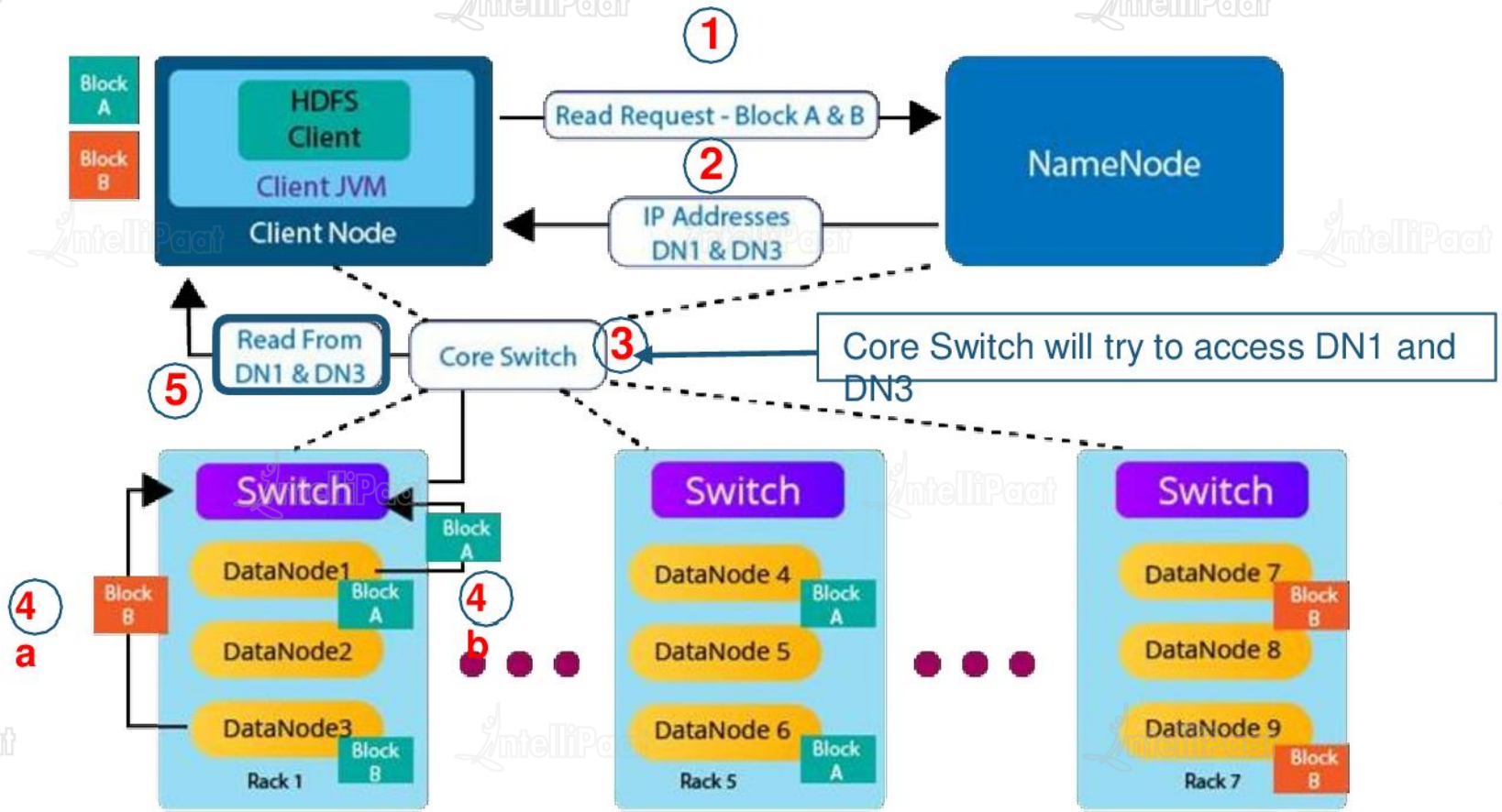
Copyright Intellipaat. All rights reserved.

How is data read in HDFS?



Copyright Intellipaat. All rights reserved.

How is data read in HDFS?



Copyright Intellipaat. All rights reserved.

Hands-On

Copyright Intellipaat. All rights reserved.

Hands-On: Basic HDFS Commands

- Create a directory in HDFS
- Display files in HDFS
- Upload files from the local directory to HDFS
- Download files from HDFS to the local directory
- List all the HDFS commands

Copyright Intellipaat. All rights reserved.

Quiz

Copyright Intellipaat. All rights reserved.

Question 1

What are the problems faced by Big Data which the

A Scalabilit y

B Hardware failure and combining data

C Network failure and combining data

D None of the above

Copyright Intellipaat. All rights reserved.

Answer 1

What are the problems faced by Big Data which the

A	Scalabilit y
B	Hardware failure and combining data ✓
C	Network failure and combining data
D	None of the above

Copyright Intellipaat. All rights reserved.

Question 2

Apache Hadoop is used for _____.

A. Storing data

B. Analyzing data

C. Storing and analyzing data

D. None of the above

Copyright Intellipaat. All rights reserved.

Answer 2

Apache Hadoop is used for _____.

A	Storing data
B	Analyzing data
C	Storing and analyzing data ✓
D	None of the above

Copyright Intellipaat. All rights reserved.

Question 3

Which command would you use for creating a

A	hdfs fsck
B	hdfs dfs –mkdir /user/test
C	hdfs dfs –ls /
D	None of the above

Copyright Intellipaat. All rights reserved.

Answer 3

Which command would you use for creating a

A	hdfs fsck
B	hdfs dfs –mkdir /user/test ✓
C	hdfs dfs –ls /
D	None of the above

Copyright Intellipaat. All rights reserved.

Question 4

The need for data replication can arise in various

A. Replication Factor is changed

B. DataNode goes down

C. Data Blocks get corrupted

D. All of the mentioned

Copyright Intellipaat. All rights reserved.

Answer 4

The need for data replication can arise in various

A	Replication Factor is changed
B	DataNode goes down
C	Data Blocks get corrupted
D	All of the mentioned ✓

Copyright Intellipaat. All rights reserved.

Question 5

HDFS is implemented in _____programmin

A	C++
B	JAVA
C	Scala
D	All of the mentioned

Copyright Intellipaat. All rights reserved.

Answer 5

HDFS is implemented in _____programmin

A	C+
B	JAV A
C	Scal a
D	All of the mentioned

Copyright Intellipaat. All rights reserved.

India: +91-7847955955

US: 1-800-216-8930 (TOLL FREE)

sales@intellipaat.com

24/7 Chat with Our Course Advisor

Copyright Intellipaat. All rights reserved.