**Hadoop Architecture**

Hadoop has three main components:

HDFS

YARN  
MAP REDUCE

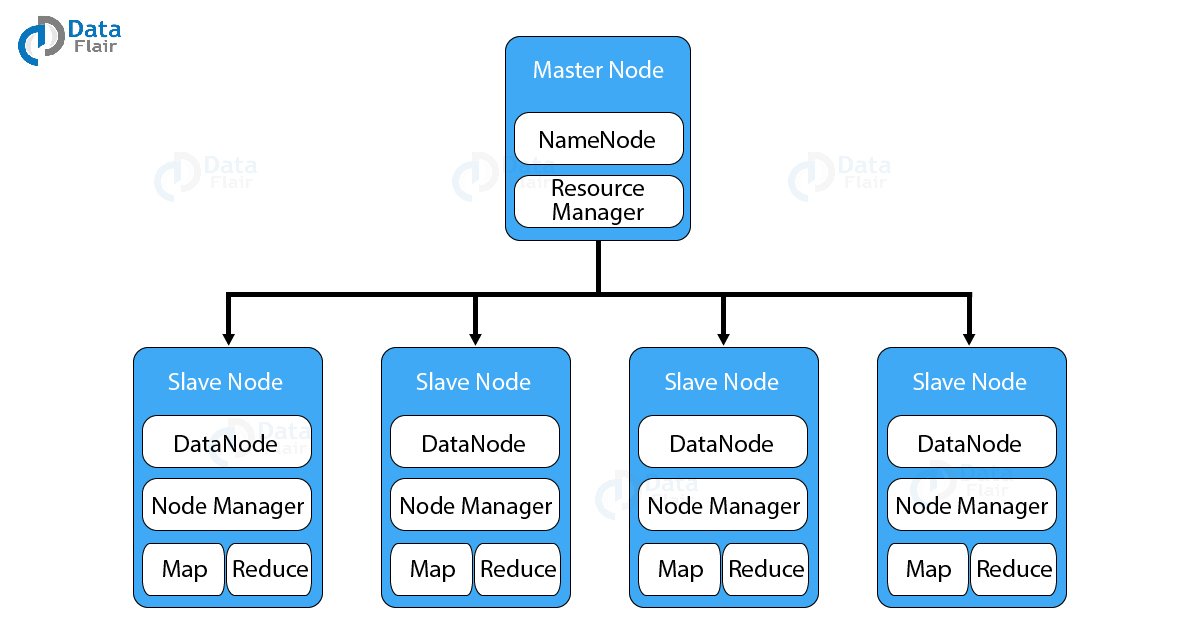
**HDFS**

🡪Hadoop Distributed File System

🡪Storage layer of Hadoop

🡪HDFS splits the data unit into small units called block and stores them in a distributed manner. It has two daemons, one is **Name Node** and other is **Data Node.**

HDFS has a **Master-slave architecture**. The daemon called Name Node runs on the master server. It is responsible for Namespace management and regulates file access by the client. Data Node daemon runs on slave nodes. It is responsible for storing actual business data. Internally, a file gets split into a number of data blocks and stored on a group of slave machines. Name node manages modifications to file system namespace. These are actions like the opening, closing and renaming files or directories. Name node also keeps track of mapping of blocks to Data nodes. This Data nodes serves read/write request from the file system’s client. Data node also creates, deletes and replicates blocks on demand from Name node.



**Java is the native language** of HDFS. Hence one can deploy Data node and Name ode on machines having Java installed. In a typical deployment, there is one dedicated machine running Name node. And all the other nodes in the cluster run Data node. The Name node contains metadata like the location of blocks on the Data nodes. And arbitrates resources among various competing Data nodes.

**Blocks**

The default block size in Hadoop is 64 MB or 128 Mb

Lets understand this with an example:

Suppose u r having a file name abc.txt of size 650 MB.

SO the Name Node will split the data unit into small data unit called block of each size up to 128 Mb(taking 128 MB as default size in this case considering having a huge Hadoop cluster)

So 650 MB will now get split into different block ie

Block1 Block 2 Block 3 Block 4 Block 5 Block 6

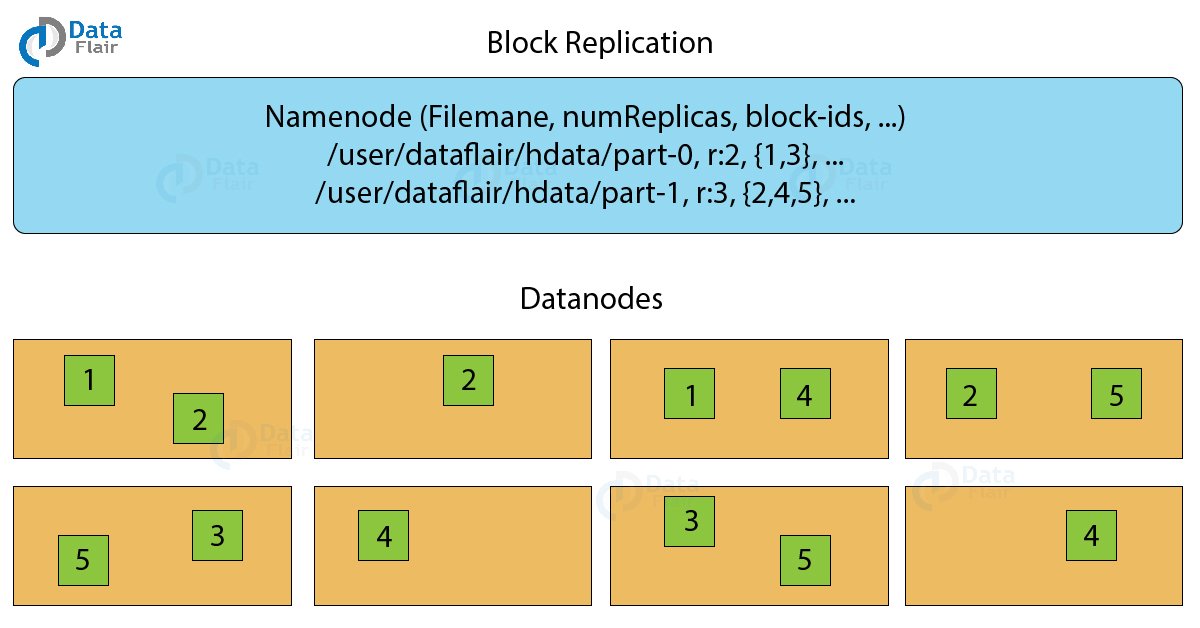
128 Mb 128Mb 128 Mb 128 Mb 128 Mb 10 Mb

What if the block size is 4kb? so for one reason would be data retrieve will me much faster, right?

The reason why we can’t have 4kb block size is because this will create huge metadata which will overload the Name Node. Hence we have to choose our HDFS block size judiciously.

**Replication Management**

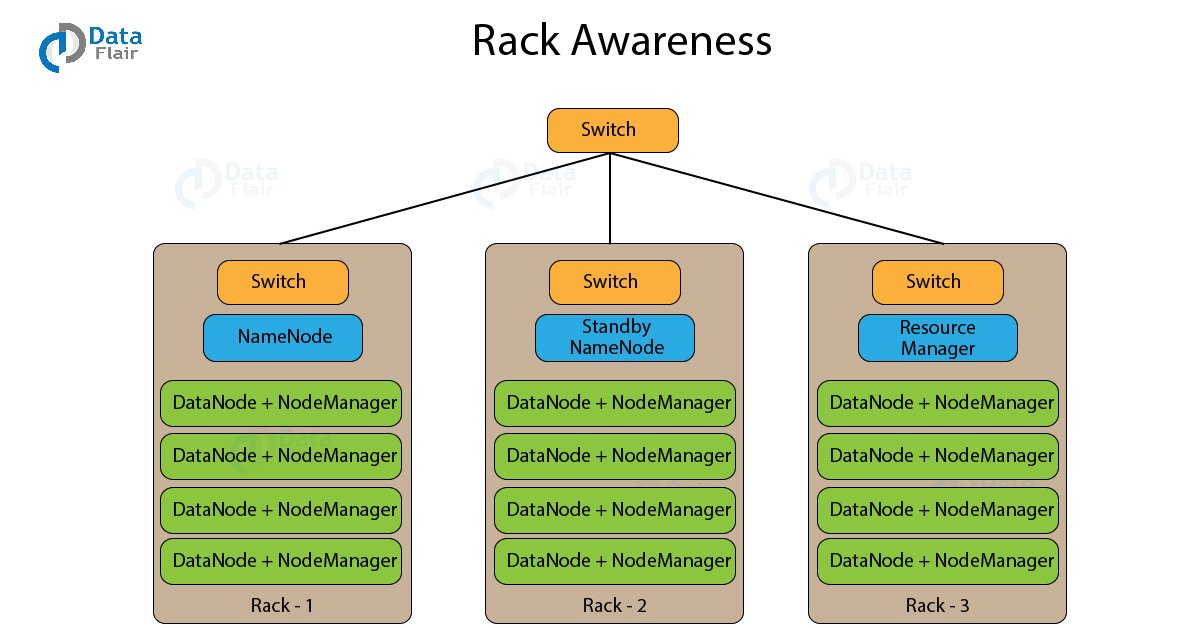
To provide **fault tolerance HDFS**uses a replication technique. In that, it makes copies of the blocks and stores in on different Data nodes. Replication factor decides how many copies of the blocks get stored. It is 3 by default but we can configure to any value.



To maintain the replication factor Name node collects block report from every Data node. Whenever a block is under-replicated or over-replicated the Name node adds or deletes the replicas accordingly.

Rack Awareness

The **Rack** is the collection of around 40-50 Data nodes connected using the same network switch. If the network goes down, the whole rack will be unavailable. A large Hadoop cluster is deployed in multiple racks.



A rack contains many Data node machines and there are several such racks in the production. HDFS follows rack awareness technique to place the replicas of the blocks in a distributed fashion. This rack awareness algorithm provides for low latency and fault tolerance. Suppose the replication factor configured is 3. Now rack awareness algorithm will place the first block on a local rack. It will keep the other two blocks on a different rack. It does not store more than two blocks in the same rack if possible.