Hadoop Distributed File System

Agenda

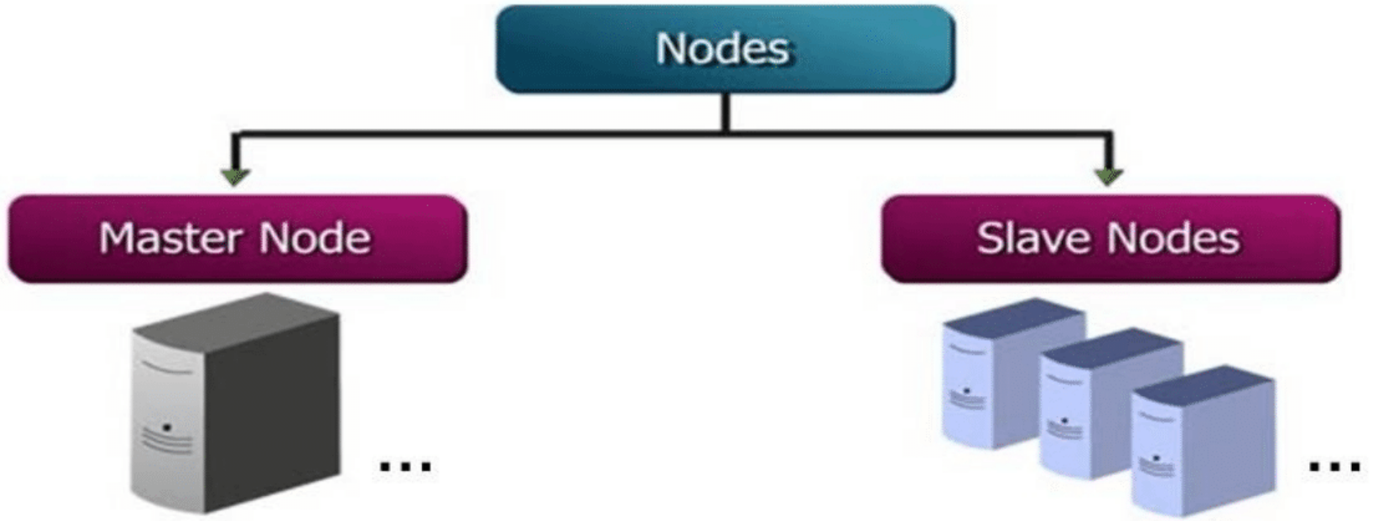
- ✓ Introduction to HDFS
- ✓ HDFS Nodes
- ✓ Daemons
- ✓ Data Storage Mechanism
- ✓ Basic Architecture
- ✓ Features and Characteristics

What is HDFS

- HDFS is a Filesystem designed for storing very large files running on cluster of commodity hardware (non-expensive, low-end hardware, used for daily purpose)
- It is designed on the principle of storage of less number of large files
- It provides fault-tolerant storage layer for Hadoop and its other components
- It stores data reliably even in case of hardware failure
- It is a distributed file system that provides high-throughput access to application data

HDFS Nodes

- There are two types of nodes, works in master – slaves pattern
- Hadoop cluster typically have namenode(s) (master(s)) and n number of datanodes (slaves) to form the HDFS cluster



HDFS Master

- HDFS cluster consists of master servers that manages the file system namespace and regulates access to files by clients.
- It manages all the slave nodes and assign work to slaves
- It executes file system namespace operations like opening, closing, and renaming files and directories.
- It should be deployed on reliable hardware

HDFS Slave

- HDFS cluster consists of number of slaves, which actually stores the data
- Actual worker nodes, which does actual task of read, write, process, etc.
- Responsible for serving read and write requests from the file system's clients.
- They also perform block creation, deletion, and replication upon instruction from the Master.
- It can be deployed on commodity Hardware

Two Daemons Run for HDFS (for storage)

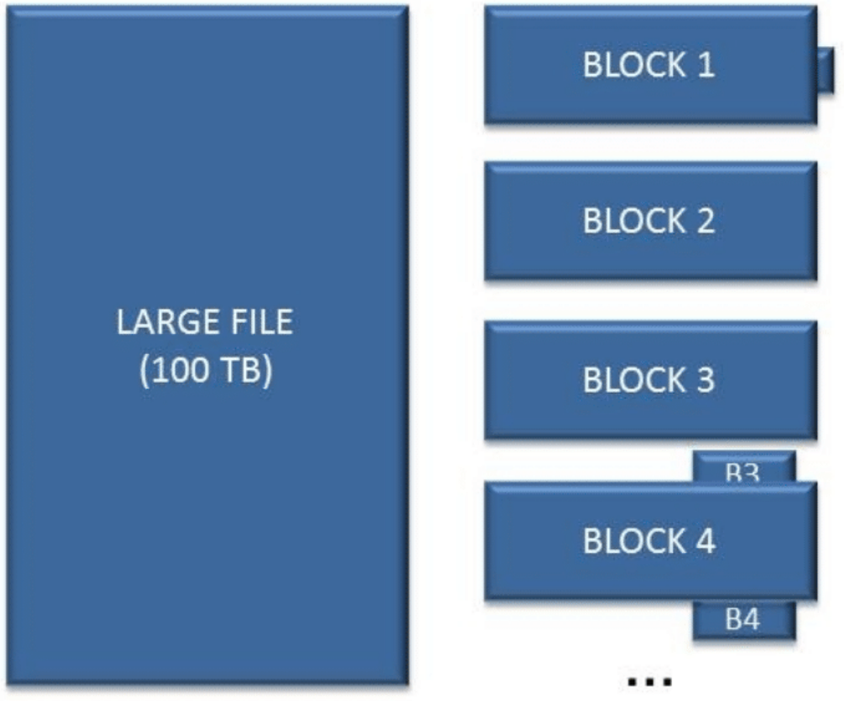
1. **NameNode**
    - This daemon run on Masters
    - Namenode stores all the metadata like filename, path, number of blocks, blockIds, block locations, number of replicas, slave related config, etc.
    - It keeps the meta-data in memory for fast retrieval.
2. **DataNode**
    - This daemon runs on all the slaves
    - Datanodes store the actual data

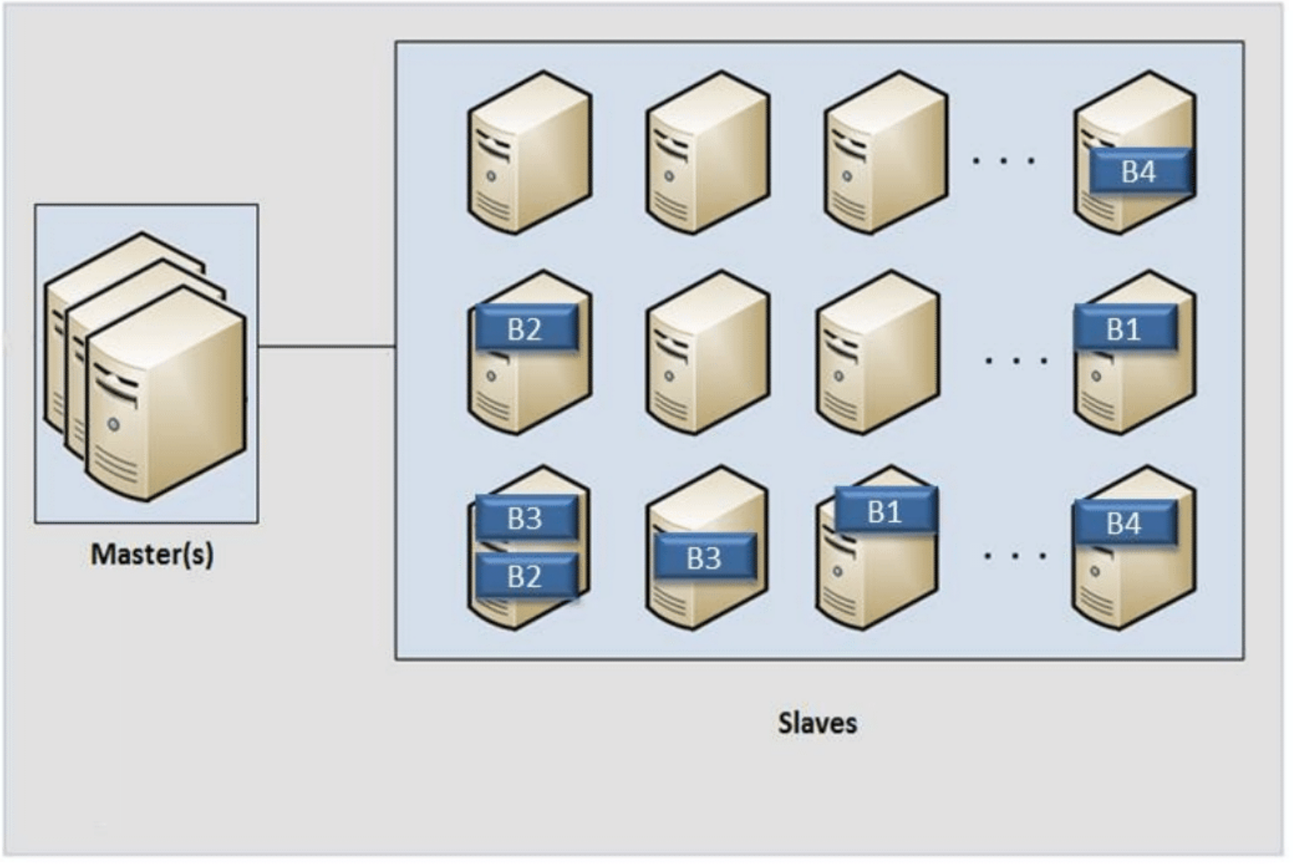
Data Storage in HDFS

- When a file is stored in HDFS, it is broken into small chunks (blocks)
  - The default block size is 64 MB
  - These blocks are stored on the multiple nodes in the cluster in distributed manner
  - Since data is stored distributedly, it provides a mechanism for Map-Reduce to process the data in parallel over the cluster

- HDFS stores multiple copies of data on different nodes
  - By default 3 copies of blocks are created
  - Replication of data provides fault tolerance, reliability, high-availability

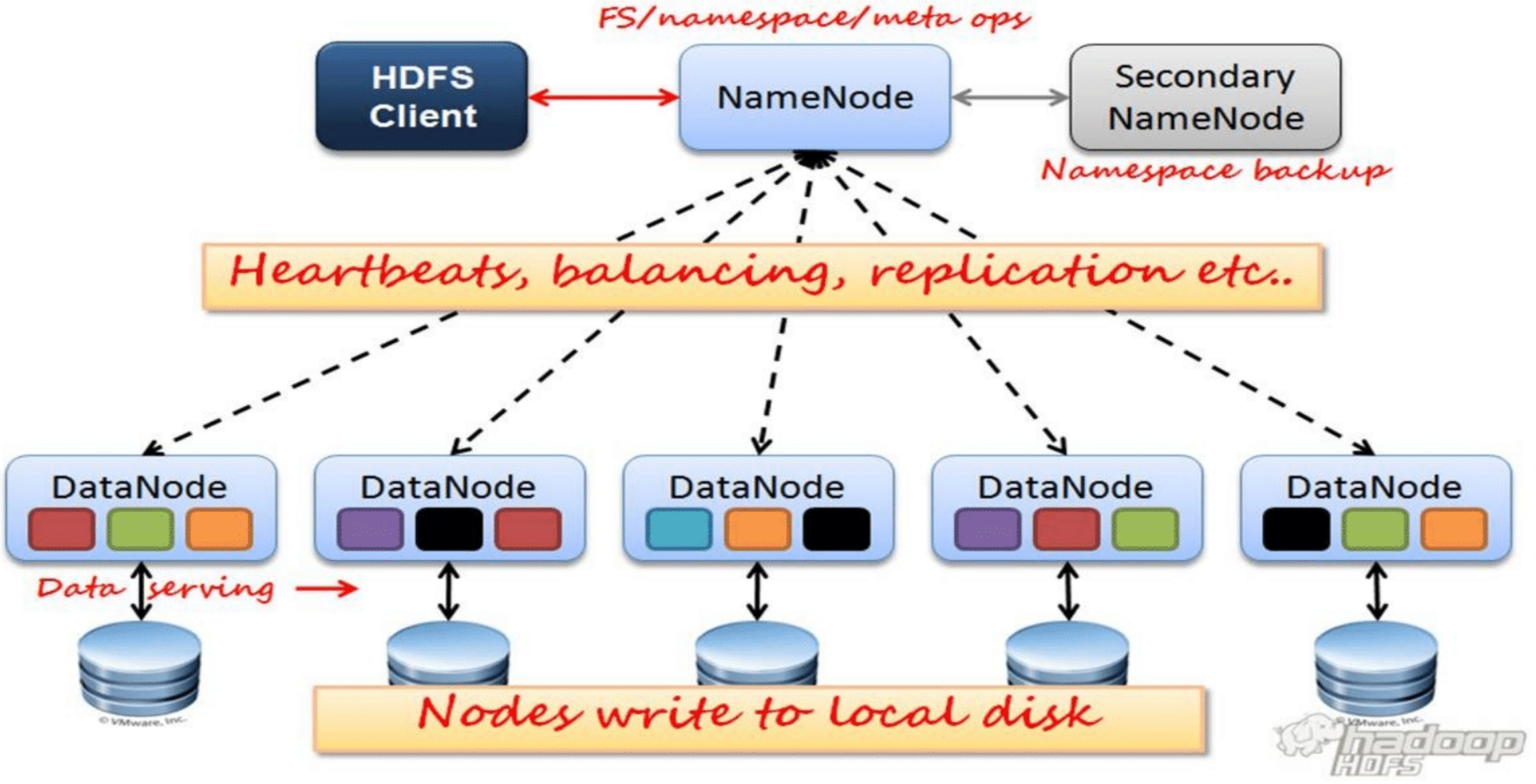
Data Storage in HDFS





Hadoop Cluster

HDFS Architecture



Features of HDFS

- Distributed Storage
- Blocks
- Replication
- High Availability
- Data Reliability
- Fault tolerant
- Scalability
- High throughput access to application

Distributed storage

- Data is distributed across the cluster of nodes
- Data is splitted into smaller pieces and stored across multiple nodes of the cluster
- In this way it provides a way to Map-Reduce to process subset of large data parallely on multiple nodes
- Distributed storage also provides fault-tolerance capability

Blocks

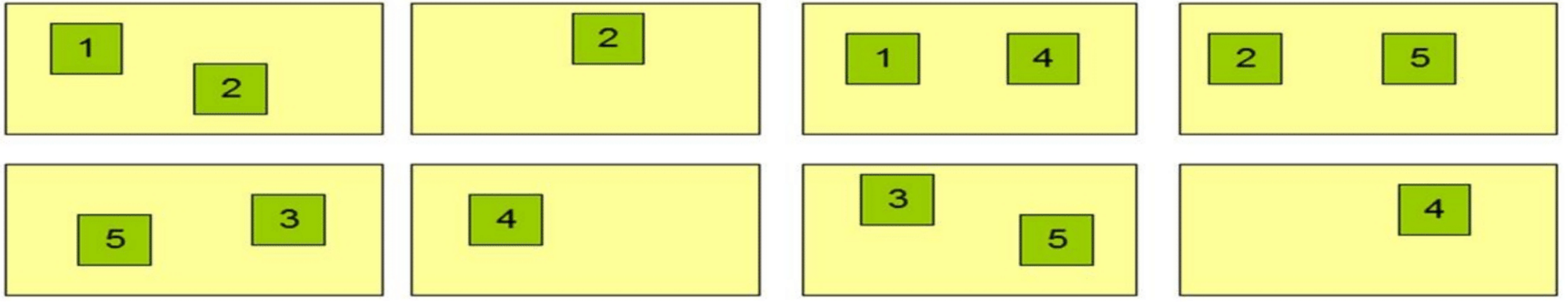
- Files are splitted into n number of blocks
- Unlike 1 KB block size of OS filesystem, HDFS's default block size is 64 MB (which can be increased according to the requirements)
  - It saves disk seek time,
  - Another advantage is in the case of processing (1 Mapper processes 1 block at a time)

Blocks

Block Replication

Namenode (Filename, numReplicas, block-ids, ...)
/users/sameerp/data/part-0, r:2, {1,3}, ...
/users/sameerp/data/part-1, r:3, {2,4,5}, ...

Datanodes



Replication

- In HDFS all the data blocks are replicated across the cluster of nodes
- Ideally one replica is present at one geographical location
- The default replication factor is 3, which can be changed according to the requirements.
- We can change replication factor to desired value by editing configuration files (hdfs-site.xml)

High Availability

- HDFS provides high availability of data by creating multiple replicas of data blocks
- If a network link or node or some hardware goes down data will be available from some other path
- If a hardware goes down data can be accessed from other nodes or other paths as we have replicated data (default 3 replicas)

Data Reliability

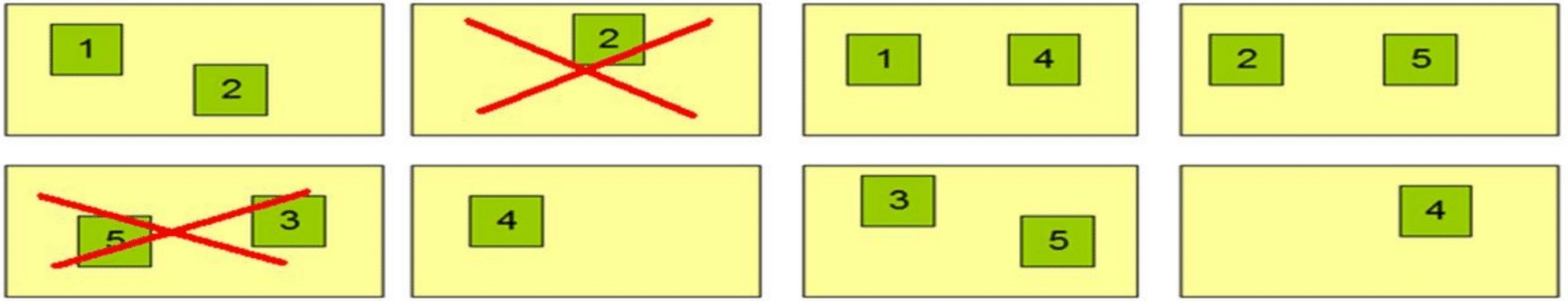
- Data is stored quite reliably in HDFS, this feature is related to High Availability.
- Since data is highly available due to replication, hence if a node or few nodes or some hardware crashes, the data will be stored reliably.
- We can again balance the cluster by making the replication factor to desirable value (by running few commands)
- In other words we can say Hadoop (HDFS) is fault tolerant

Data Reliability

Block Replication

Namenode (Filename, numReplicas, block-ids, ...)
/users/sameerp/data/part-0, r:2, {1,3}, ...
/users/sameerp/data/part-1, r:3, {2,4,5}, ...

Datanodes



Fault Tolerance

- HDFS provides fault tolerant storage layer for Hadoop and other components in the ecosystem
- Replication factor and distributed storage helps us to attain this feature of Hadoop
- If a hardware goes down data can be accessed from other nodes or other paths as we have replicated data (default 3 replicas)

Scalability

Two ways to scale the Filesystem

- Vertical Scaling
  - Add more disks on the nodes of the cluster
  - For this we need to edit the configuration files and add corresponding entries of newly added disks
- Horizontal Scaling
  - Another option is to add more nodes in the cluster
  - We can add n number of nodes in the cluster **on the fly** (without any downtime)

High throughput Access to Application Data

- HDFS provides high throughput access to application data.
- Throughput is the amount of work done in a unit time.
- In HDFS, work is divided and shared among different systems which execute the tasks assigned to them independently and in parallel.
- By reading data in parallel, we decrease the actual time to read data tremendously.

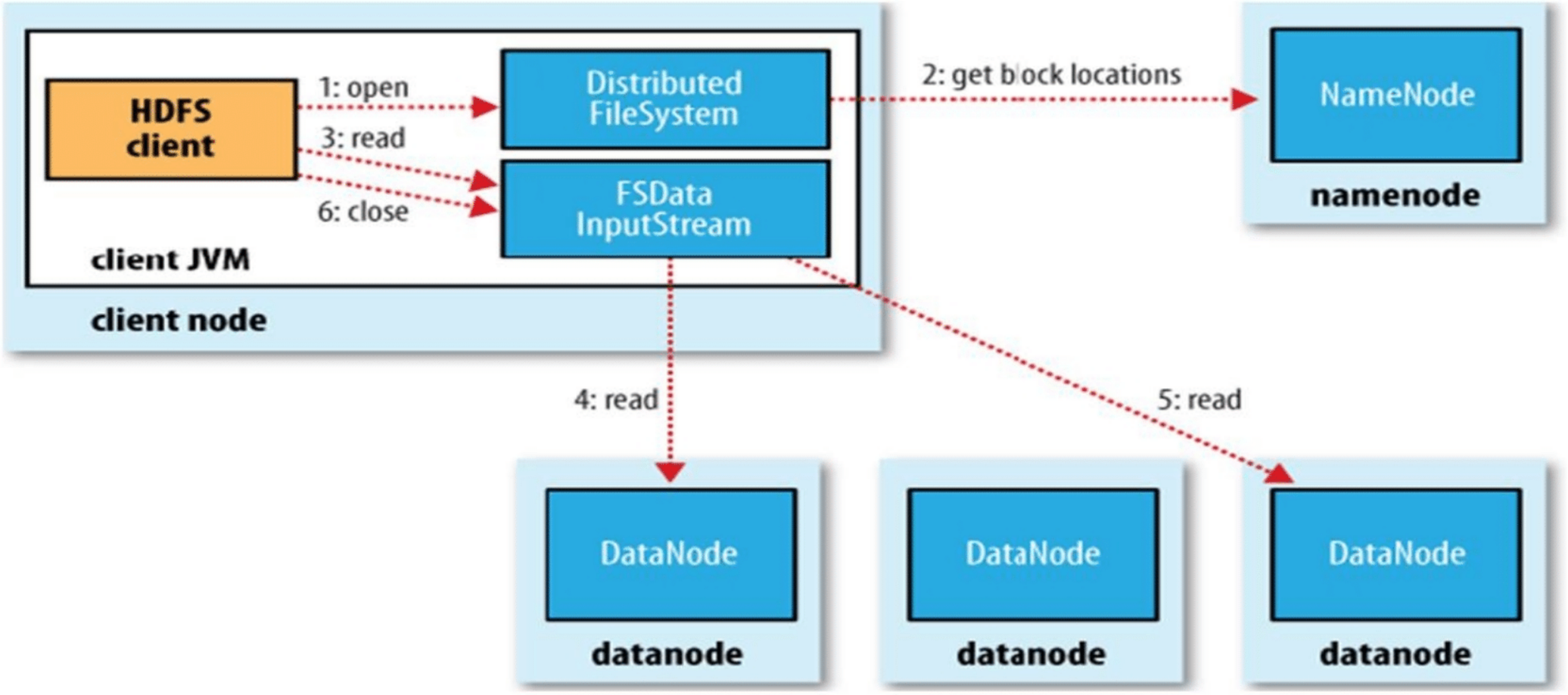
File Access & other Operations

- Programming
- CLI (Command Line Interface)
- We can perform almost all the operations which we can do on local filesystem
- HDFS provides different access rights rwx to ugo
- We can also browse the filesystem by browser (http://Master-IP:50070)

File Read operation

- To read a file from HDFS, client needs to interact with Namenode
- Namenode provides the address of the slaves where file is stored
- Client will interact with respective Datanodes to read the file
- Namenode also provide a token to the client which it shows to data node for authentication

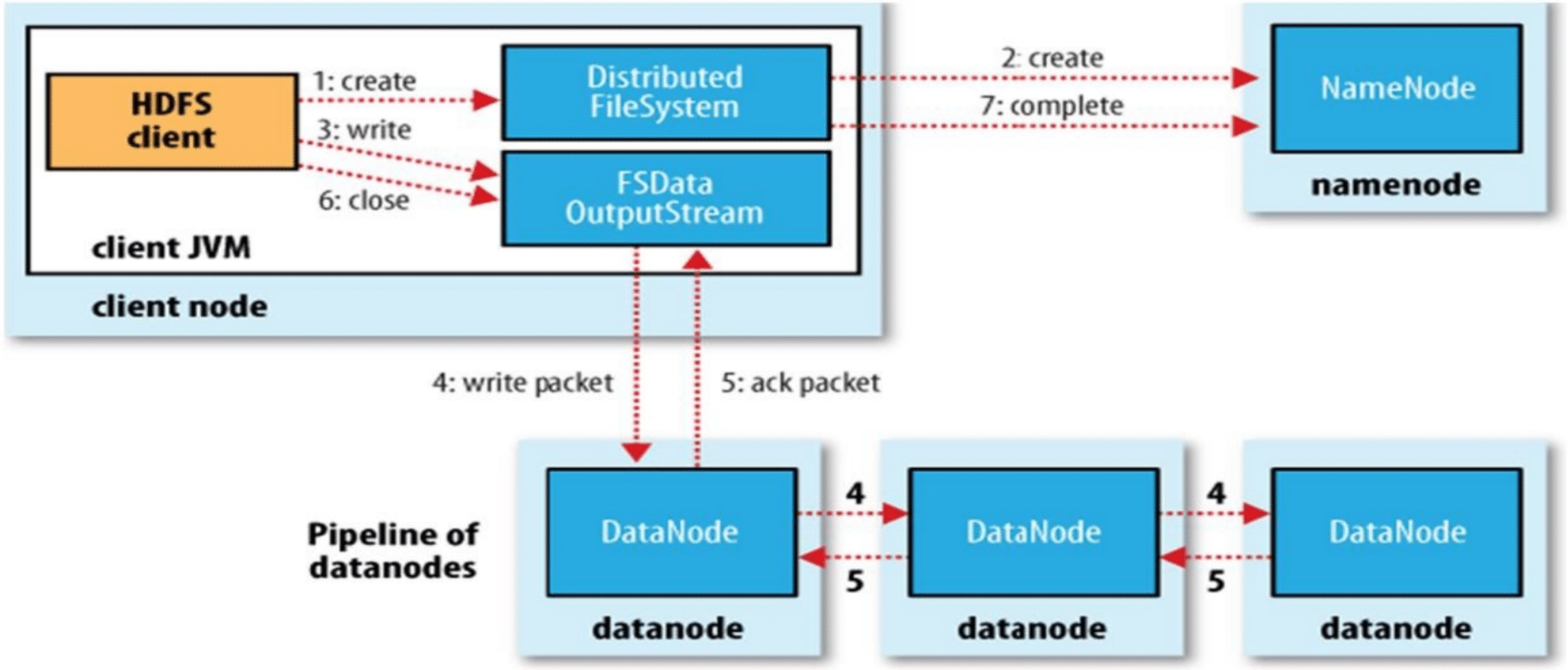
File Read operation



File Write operation

- To write a file into HDFS, client needs to interact with namenode.
- Namenode provides the address of the slave on which client will start writing the data
- As soon as client finishes writing the block, the slave starts copying the block to another slave which in tern copy the block to another slave (3 replicas by default)
- After required replicas are created then it will send the acknowledge to the client
- Authentication process (same as file read)

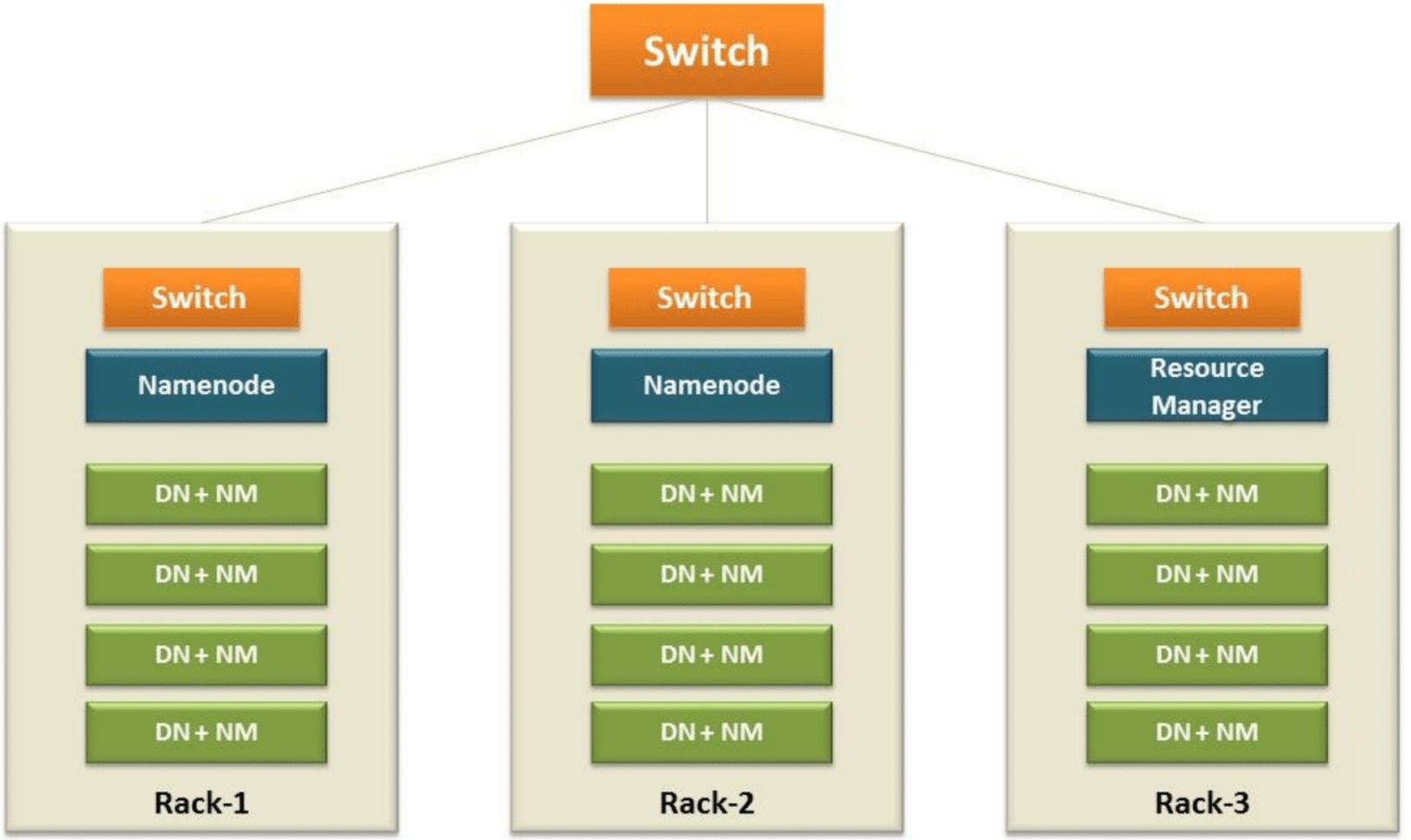
File Write operation



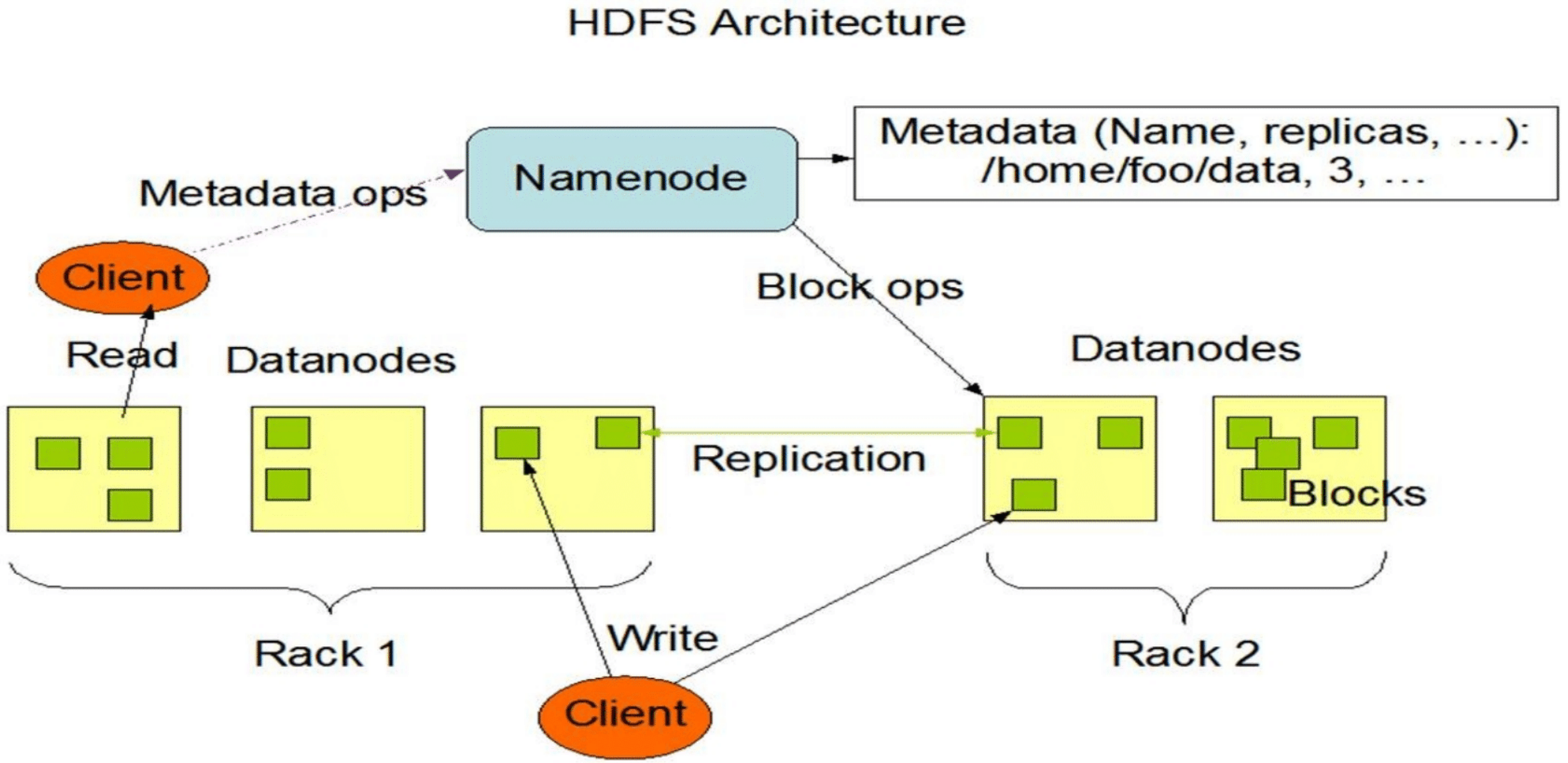
Rack Awareness

- Hadoop run on a cluster of computers which are commonly spread across many racks
- NameNode places replicas of block on multiple racks for improved fault tolerance
- NameNode tries to place replicas of block across multiple racks, so that if complete rack goes down then also system will be highly available
- Optimizing replica placement distinguishes HDFS from most other distributed file systems.
- The purpose of a rack-aware replica placement policy is to improve data reliability, availability, and network bandwidth utilization.

Rack Awareness



Advance HDFS Architecture



Thank You

DataFlair Web Services Pvt Ltd
+91-8451097879 / +91-7718877477
info@data-flair.com
http://data-flair.com
https://www.facebook.com/DataFlairWS