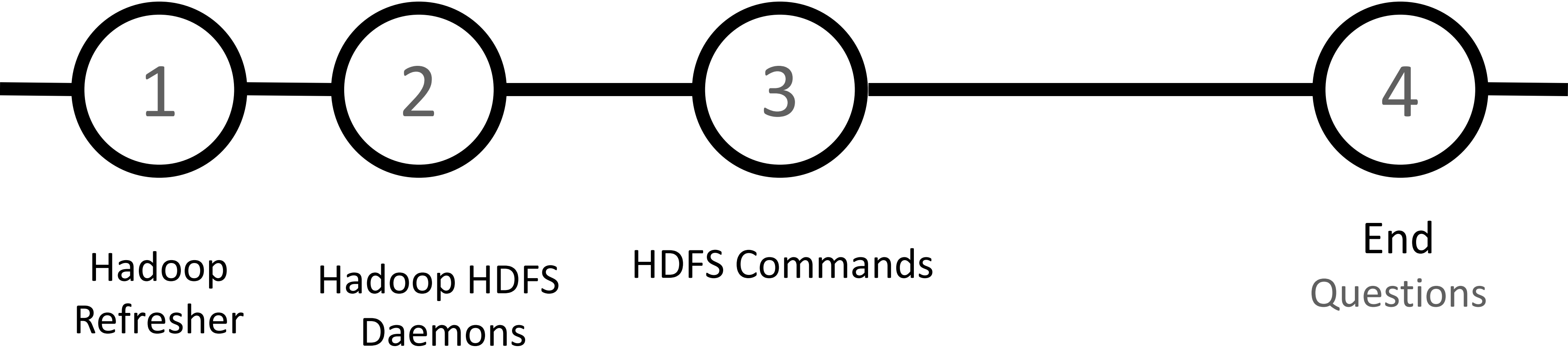
hadoop

Part 2

AGENDA



1 Hadoop Refresher

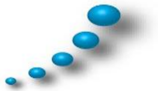
2 Hadoop HDFS Daemons

3 HDFS Commands

4 End Questions

Hadoop Refresher

Why Hadoop



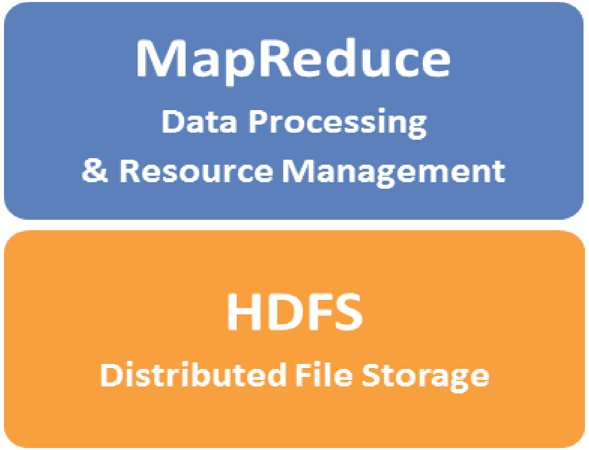
Scalability (petabytes of data, thousands of machines)

Flexibility in accepting all data formats (no schema)

Efficient and simple fault-tolerant mechanism

Commodity inexpensive hardware

Hadoop v1.0



MapReduce

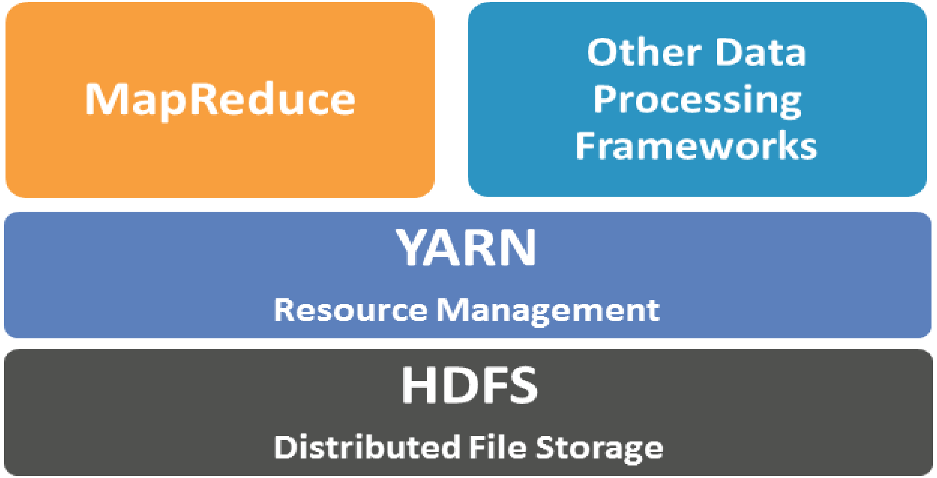
Data Processing

& Resource Management

HDFS

Distributed File Storage

Hadoop v2.0



MapReduce

Other Data Processing Frameworks

YARN

Resource Management

HDFS

Distributed File Storage

HDFS

Hadoop Distributed File Systm

Storage Layer

HDFS

- HDFS is responsible for **storing the large files** on a cluster of **commodity hardware**.

- Hadoop works in **master-slave** fashion.

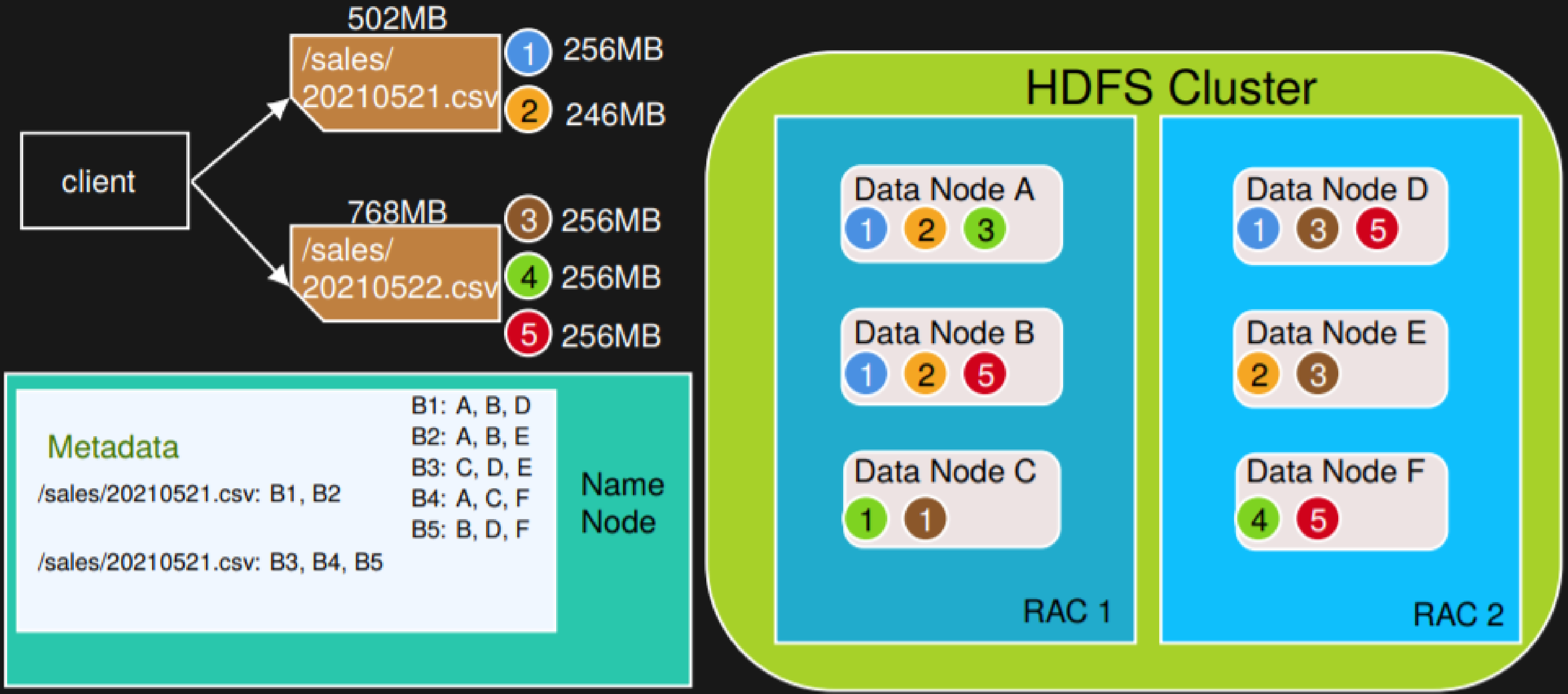
- HDFS also has two types of nodes that work in the same manner. These are the **NameNode(s)** and the **DataNodes**.

- Data is split into **blocks** with configurable block size, for example, 64MB, 128MB, and 512MB. Editing the configuration files **(hdfs-site.xml).**

- Each data block **is replicated** and distributed across the cluster data node. This replication is configurable, and by default, **(3) replica** (folds).

- Each block is stored in **three different nodes**. It is recommended to have **two nodes in the same rack** and **the third one in a different rack**. (**Rack Awareness**)

HDFS



Metadata

/sales/20210521.csv: B1, B2

/sales/20210521.csv: B3, B4, B5

B1: A, B, D
B2: A, B, E
B3: C, D, E
B4: A, C, F
B5: B, D, F

Figure: HDFS

NameNode (Management Node)

- A **NameNode** keeps **track of the location of the blocks** and which blocks make up these files. These details known as **metadata** like **filename**, the **number of blocks**, **number of replicas**, **a location of blocks**, **block IDs**, etc.
- This metadata is available in **memory** in the master for faster retrieval of data. In the local disk, a copy of the metadata is available for **persistence**. So NameNode memory **should be high as per the requirement**.
- NameNodes daemon must be always running.
- Hadoop cluster contains at least two NameNodes Active/Standby nodes.
- HDFS files are **write one**, so we can’t do any random writes.

DataNode

- There are **n** number of slaves.
- It manages **storage** of data.
- These slave nodes are **the actual worker** nodes that do the tasks.
- serve **read and write** requests from the file system’s clients.
- They perform block creation, deletion, and replication upon **instruction from the NameNode.**
- Once a block is written on a DataNode, it **replicates** it to other DataNode, and the process continues until creating the required number of replicas.
- DataNode sends a **heartbeat** message to NameNode **periodically** to indicate that it is **alive**

Secondary NameNode (**Not a backup**)

- Name node stores information in **two** files.
- **fsimage**: It’s a snapshot of the file system, stores information like modification time access time, permission, replication.
- **Edit logs**: It stores details of all the activities/transactions being performed on the HDFS..
- When the namenode is in the active state the edit logs size **grows continuously** as the edit logs can only be applied to the fsimage at the time of namenode **restart**, to get the latest state of the HDFS. If edit logs grows significantly and namenode tries to apply it on fsimage at the time of namenode restart, the process can **take very long**, here secondary node come into the play.
- Secondary namenode keeps the **checkpoint on the namenode**, It reads the edit logs from the namenode continuously after a specific interval and applies it to the fsimage copy of secondary namenode.
- Secondary namenode is **a helper node** and can’t **replace the namenode.**

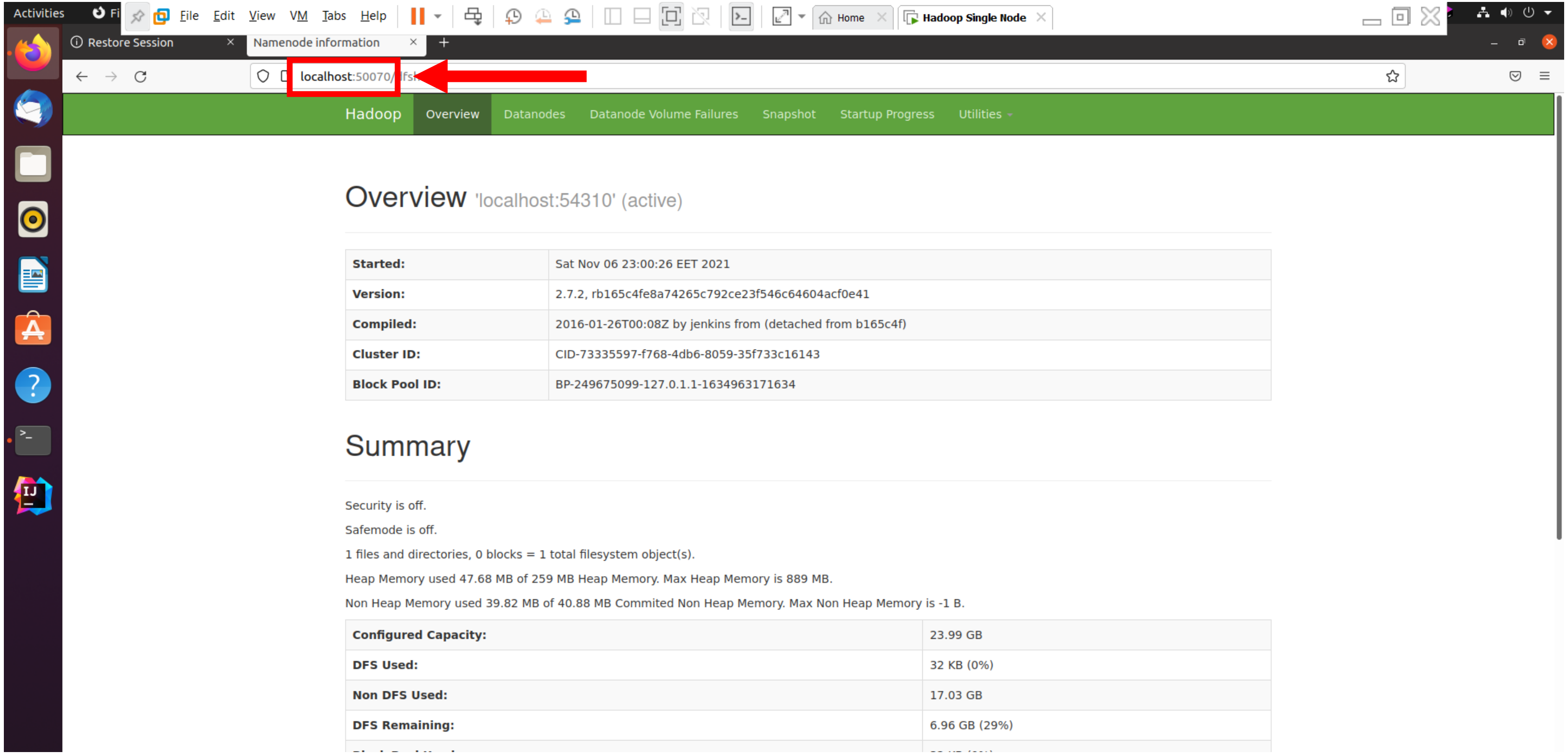
Hadoop HDFS Operations

- We need to interact with the file system either by **programming** or by the **command-line interface**.

- Hadoop Distributed File System has many similarities with the **Linux** like **create a directory, copy the file, change permissions**, etc.

- It also provides different access rights like **read, write, and execute to users, groups**, and others..

- We can browse the file system here by the browser **: http://localhost:50070**



Overview 'localhost:54310' (active)

Started:	Sat Nov 06 23:00:26 EET 2021
Version:	2.7.2, rb165c4fe8a74265c792ce23f546c64604acf0e41
Compiled:	2016-01-26T00:08Z by jenkins from (detached from b165c4f)
Cluster ID:	CID-73335597-f768-4db6-8059-35f733c16143
Block Pool ID:	BP-249675099-127.0.1.1-1634963171634

Summary

Security is off.

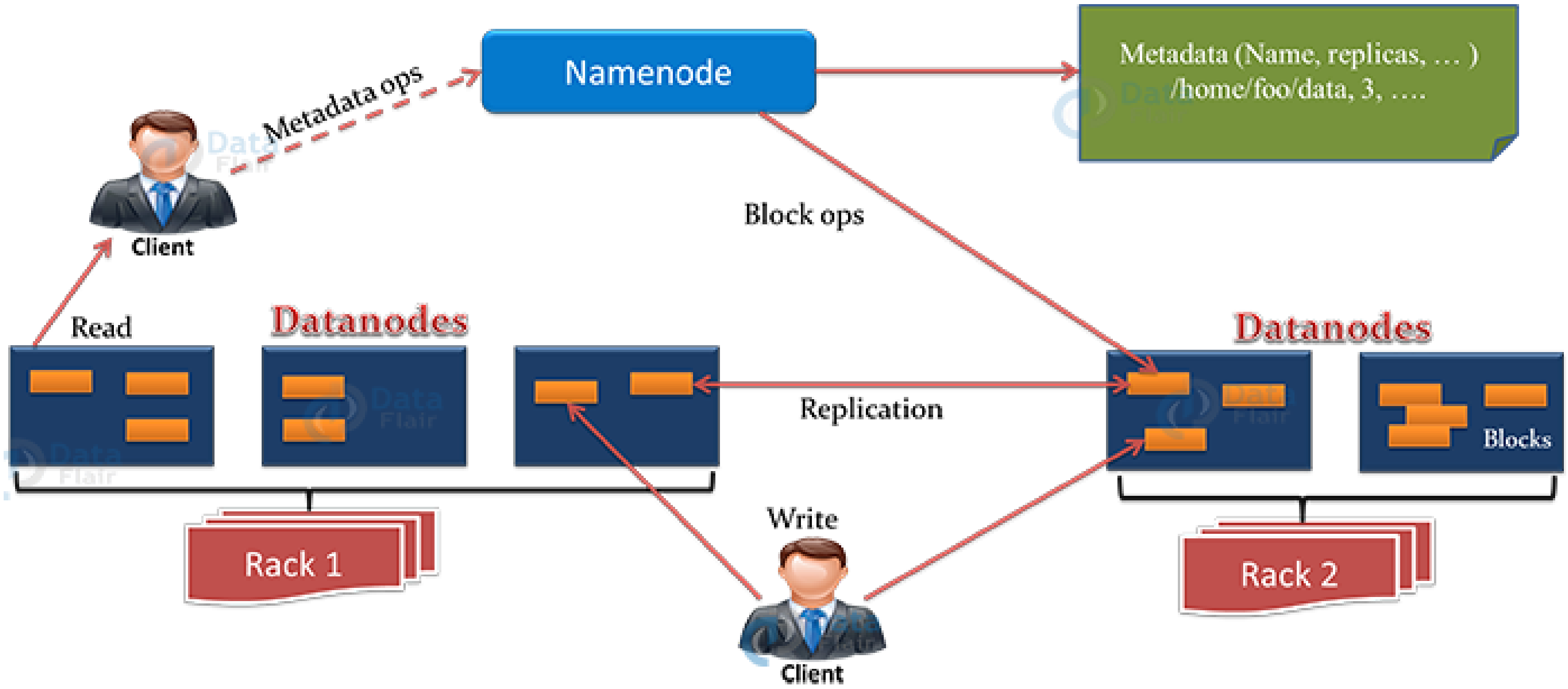
Safemode is off.

1 files and directories, 0 blocks = 1 total filesystem object(s).

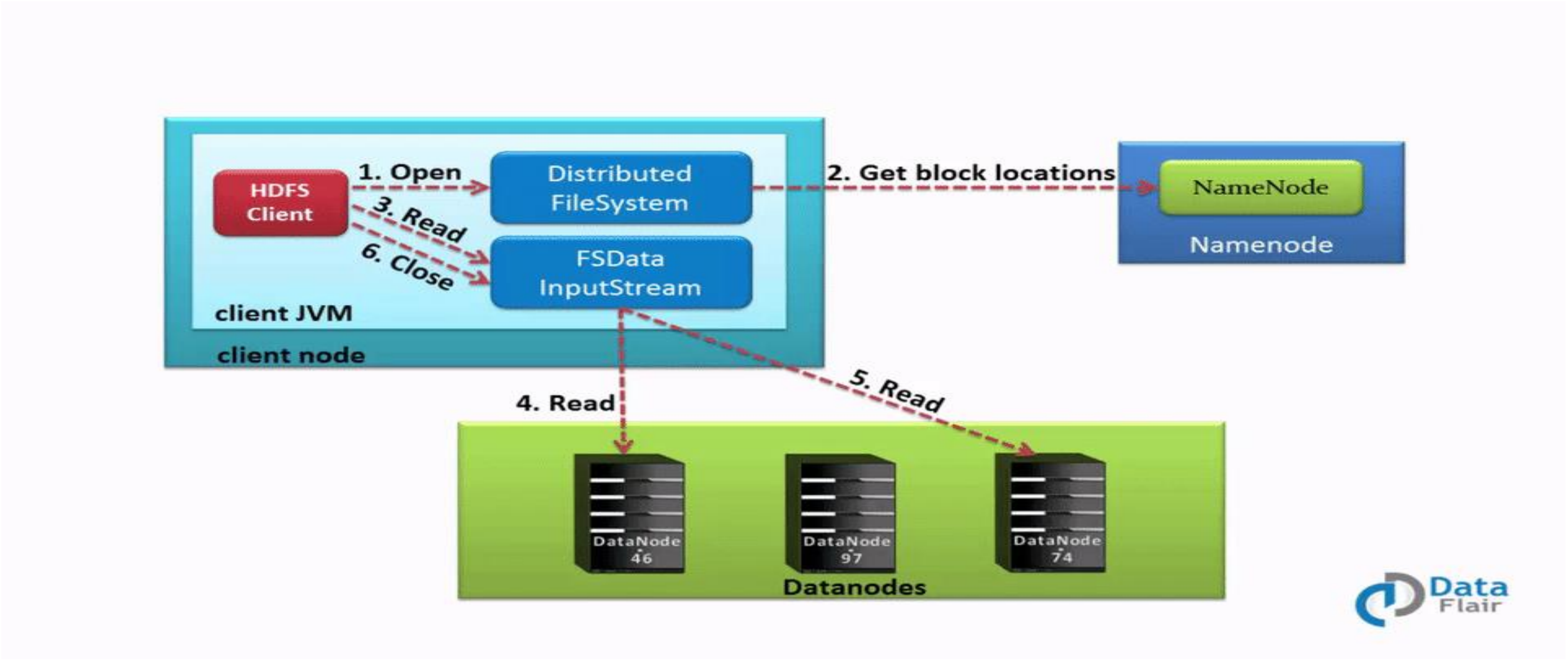
Heap Memory used 47.68 MB of 259 MB Heap Memory. Max Heap Memory is 889 MB.

Non Heap Memory used 39.82 MB of 40.88 MB Commited Non Heap Memory. Max Non Heap Memory is -1 B.

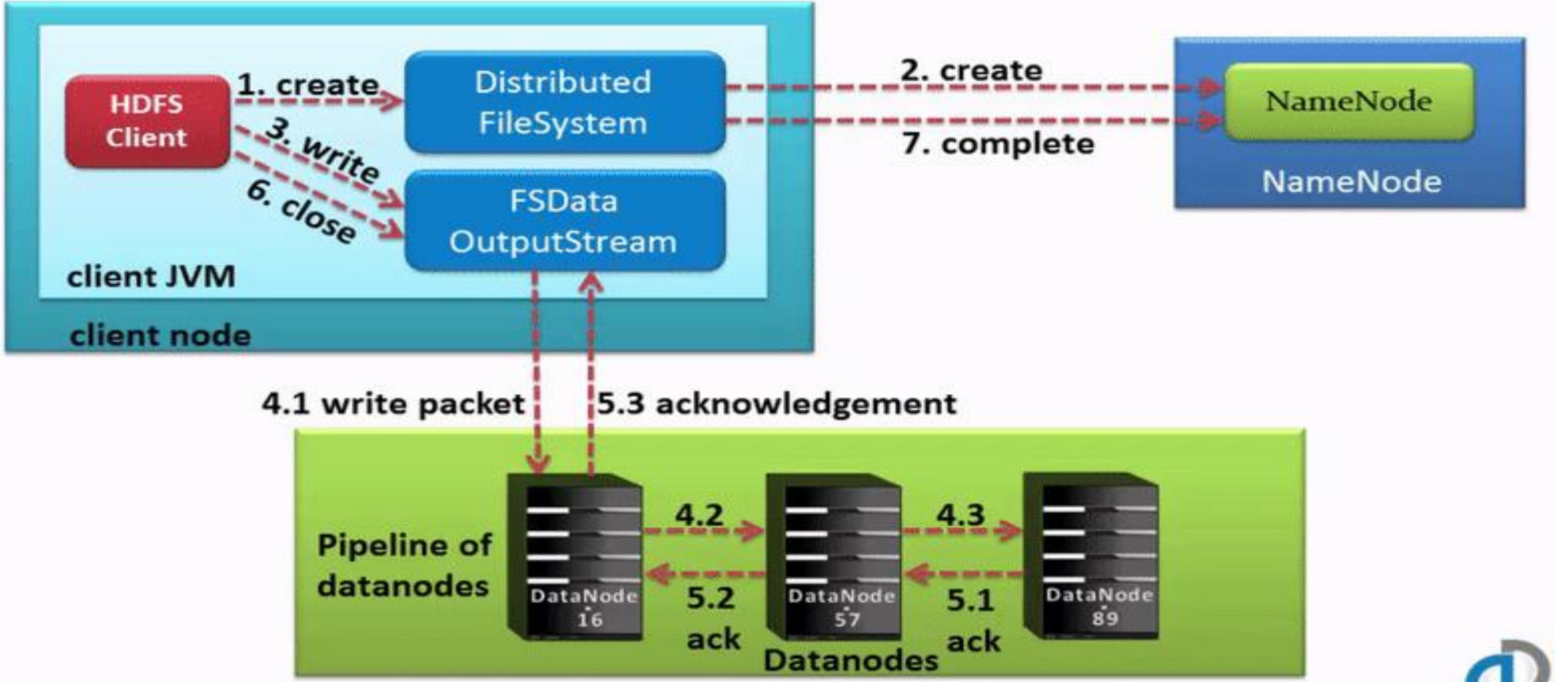
Configured Capacity:	23.99 GB
DFS Used:	32 KB (0%)
Non DFS Used:	17.03 GB
DFS Remaining:	6.96 GB (29%)


Namenode
Metadata (Name, replicas, …)
/home/foo/data, 3, ….
Metadata ops
Client
Block ops
Read
Datanodes
Datanodes
Replication
Blocks
Rack 1
Write
Client
Rack 2
Data Flair

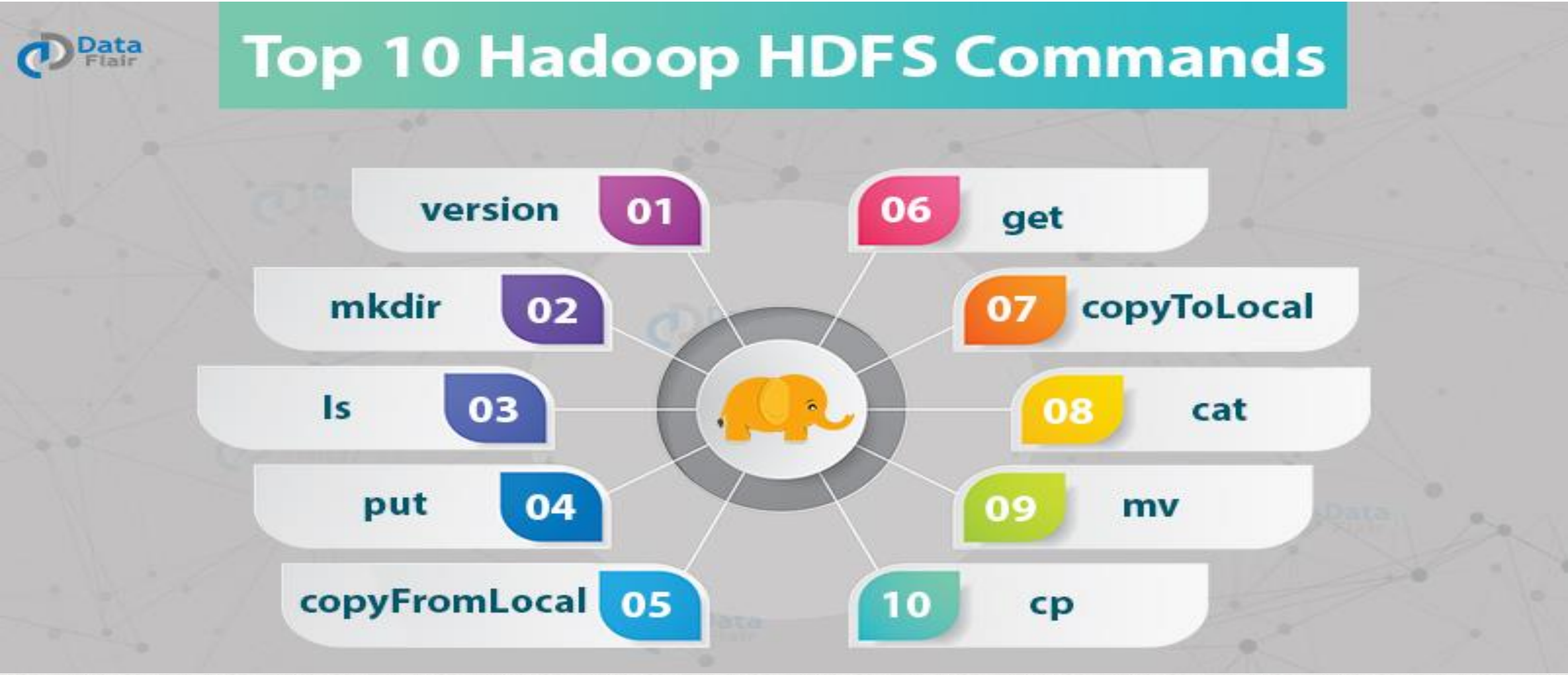
HDFS Read Operation



HDFS Write Operation



Hadoop HDFS Commands



Hadoop HDFS Commands

- hadoop version
- hadoop fs –mkdir /path/directory_name
- hadoop fs -ls /path , hadoop fs -ls –R /path
- haoop fs -put <localsrc> <dest>
- hadoop fs -copyFromLocal <localsrc> <hdfs destination>
- hadoop fs -get <src> <localdest>
- hadoop fs -copyToLocal <hdfs source> <localdst>
- hadoop fs –cat /path_to_file_in_hdfs
- hadoop fs -mv <src> <dest>
- hadoop fs -cp <src> <dest>

References

- https://data-flair.training/blogs/top-hadoop-hdfs-commands-tutorial/
- https://gability.com/en/courses/hadoop/03-hdfs.pdf
- https://data-flair.training/blogs/hadoop-hdfs-tutorial/
- https://data-flair.training/forums/topic/what-is-secondary-namenode-in-hadoop/

?

QUESTIONS

THANK YOU!