Introduction to Hadoop Storage File System

Hadoop Storage File System is a distributed file system designed to store and manage large volumes of data effectively. It offers high-throughput access to application data and is suitable for applications that have large data sets.

m by mahesh reddy

Understanding HDFS Architecture

1

Data Nodes

These are the commodity hardware machines where actual data is stored.

2

Name Node

It is the centerpiece of an HDFS file system. It keeps the directory tree of all files in the file system, maintains and manages the data node and takes care of replication and fault tolerance.

3

Secondary Name Node

This is not a backup name node but is a helper node for the primary name node. It receives the information from the name node and does the necessary merging for FSimage and edit log, then sends it back to the name node.

HDFS Commands for File System Operations

Upload

Upload local files to HDFS.

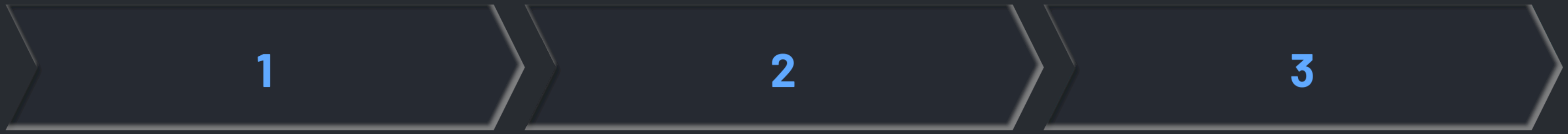
Download

Download files from HDFS to the local file system.

Delete

Delete files in HDFS.

Moving Data from Local Disk to HDFS



1

Packaging Data

Data is staged and packaged for transmission to HDFS.

2

Transfer to Hadoop

Transfer the data from the local disk to the Hadoop cluster.

3

Validation

Confirm the successful migration and validate the data in HDFS.

Getting Data from HDFS to Local Disk

1 **CopyToLocal**

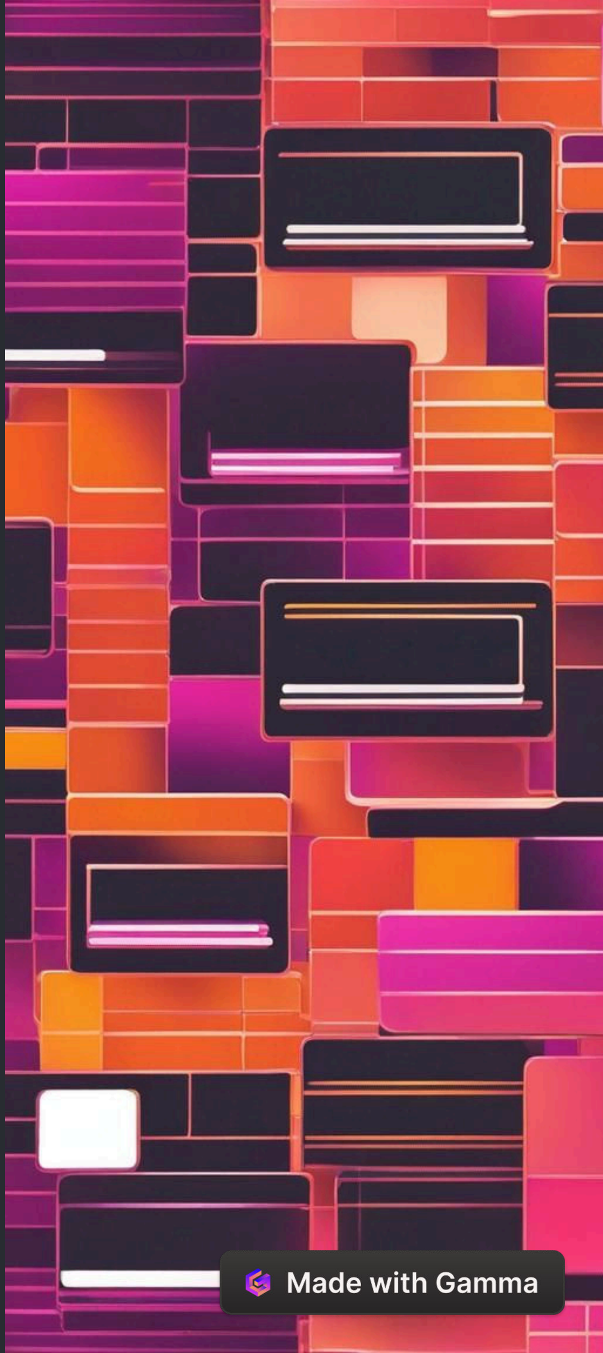
Transfer files from HDFS to the local file system.

2 **MoveToLocal**

Move files from HDFS to the local file system.

3 **GetMerge**

Merge HDFS files with checksum validation to the local file system.



Hadoop File Formats

Parquet

Columnar storage format with efficient data encoding and compression.

AVRO

Row-based storage format with support for schema evolution.

ORC

Optimized Row Columnar format with strong compression and indexing.

Hadoop Data Compression Techniques

3x Compression

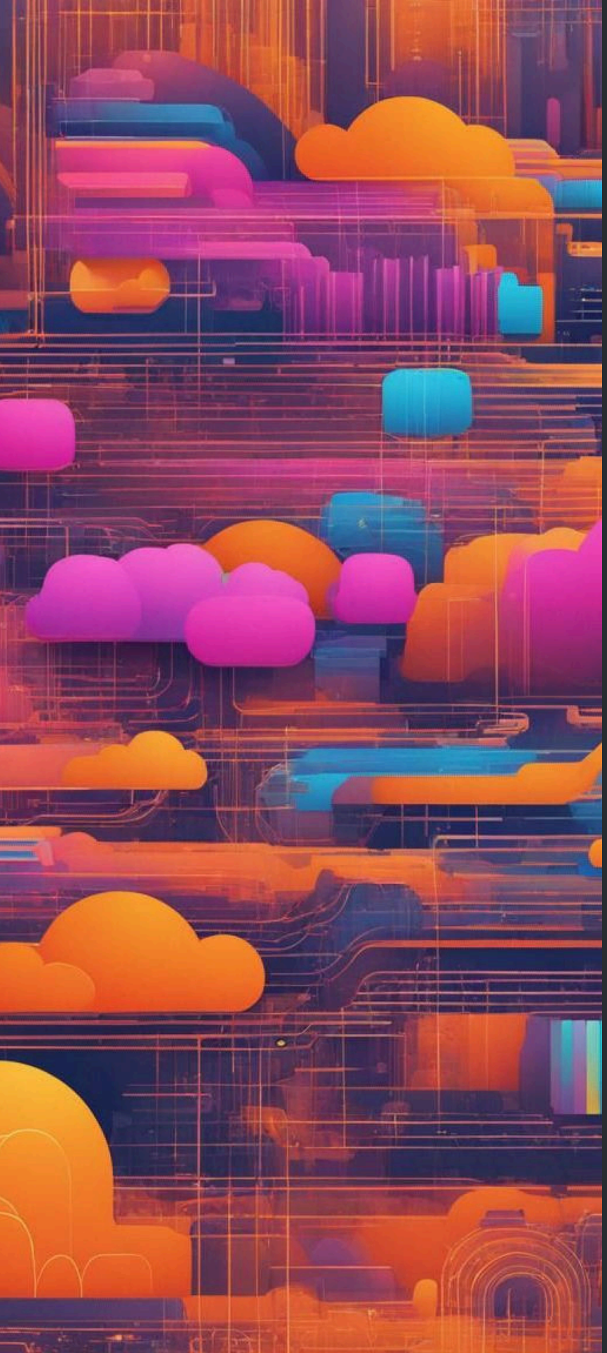
Compression Ratio

Efficiently compress data to one-third of its original size.

Snappy

Fast Compression

High-speed compression and decompression algorithm.



Conclusion and Best Practices

1

Best Practices

Implement data replication and backup strategies in HDFS for fault tolerance and resiliency.

2

Conclusion

Understanding HDFS and its operations is crucial for efficient big data management and processing.