# Introduction to Hadoop Distributed File System (HDFS)

Hadoop Distributed File System (HDFS) is a distributed, scalable, and portable filesystem written in Java for the Hadoop framework. It provides high-throughput access to application data and is designed to be fault-tolerant and suitable for use on commodity hardware.

# Overview of HDFS and Storage Commands

## What is HDFS?

HDFS is the primary storage system used by Hadoop applications.

## Storage Operations

Learn basic HDFS commands for storage and data management.

### LIST FILES

| | |
|---|---|
| hdfs dfs -ls / | List all the files/directories for the given hdfs destination path |
| hdfs dfs -ls -d /hadoop | Directories are listed as plain files. In this case, this command will list the details of hadoop folder |
| hdfs dfs -ls -h /data | Format file sizes in a human-readable fashion (eg 64.0m instead of 67108864) |
| hdfs dfs -ls -R /hadoop | Recursively list all files in hadoop directory and all subdirectories in hadoop directory |
| hdfs dfs -ls /hadoop/dat* | List all the files matching the pattern. In this case, it will list all the files inside hadoop directory which starts with 'dat' |

### READ/WRITE FILES

| | |
|---|---|
| hdfs dfs -text /hadoop/derby.log | [illegible] |
| hdfs dfs -cat /hadoop/test | This command will display the content of the HDFS file test on your stdout |
| hdfs dfs -appendToFile /home/ubuntu/test1/hadoop/tex | Appends the content of a local file test1 to a h[illegible] file test2 |

### UPLOAD/DOWNLOAD FILES

| | |
|---|---|
| hdfs dfs -put /home/ubuntu/sample /hadoop | Copies the file from local file system to HDFS |
| hdfs dfs -put -f /home/ubuntu/sa /hadoop | [illegible] |
| hdfs dfs -put -l /home/ubuntu/sa /hadoop | [illegible] |
| hdfs dfs -put -p /home/ubuntu/sa /hadoop | [illegible] |
| hdfs dfs -get /newfile /home/ubu | [illegible] |
| hdfs dfs -get -p /newfile /home/u | [illegible] modification times, ownership and the mode |
| hdfs dfs -get /hadoop/*.txt /home/ubuntu/ | Copies all the files matching the pattern from local file system to HDFS |
| hdfs dfs -copyFromLocal /home/ubuntu/sample /hadoop | Works similarly to the put command, except that the source is restricted to a local file reference |
| hdfs dfs -copyToLocal /newfile /home/ubuntu/ | Works similarly to the put command, except that the destination is restricted to a local file reference |
| hdfs dfs -moveFromLocal /home/ubuntu/sample /hadoop | Works similarly to the put command, except that the source is deleted after it's copied |

### FILE MANAGEMENT

| | |
|---|---|
| hdfs dfs -cp /hadoop/file1 /hadoop1 | Copies file from source to destination on HDFS. In this case, copying file1 from |

| | |
|---|---|
| hdfs dfs -rm -r /hadoop<br>hdfs dfs -rm -R /hadoop<br>hdfs dfs -rmr /hadoop | Deletes the directory and any content under it recursively |
| hdfs dfs -rm -skipTrash /hadoop | The -skipTrash option will bypass trash, if enabled, and delete the specified file(s) immediately |
| hdfs dfs -rm -f /hadoop | If the file does not exist, do not display a diagnostic message or modify the exit status to reflect an error |
| hdfs dfs -rmdir /hadoop1 | Delete a directory |
| hdfs dfs -mkdir /hadoop2 | Create a directory in specified HDFS location |
| hdfs dfs -mkdir -f /hadoop2 | Create a directory in specified HDFS location. This command does not fail even if the directory already exists |
| hdfs dfs -touchz /hadoop3 | Creates a file of zero length at <path> with current time as the timestamp of that <path> |

### OWNERSHIP AND PERMISSIONS

| | |
|---|---|
| hdfs dfs -checksum /hadoop/file1 | Dump checksum information for files that match the file pattern <src> to stdout |
| hdfs dfs -chmod 755 /hadoop/file1 | Changes permissions of the file |
| hdfs dfs -chmod -R 755 /hadoop | Changes permissions of the files recursively |
| hdfs dfs -chown ubuntu:ubuntu /hadoop | Changes owner of the file. 1st ubuntu in the command is owner and 2nd one is group |
| hdfs dfs -chown -R ubuntu:ubuntu /hadoop | Changes owner of the files recursively |
| hdfs dfs -chgrp ubuntu /hadoop | Changes group association of the file |
| [illegible] | [illegible] |

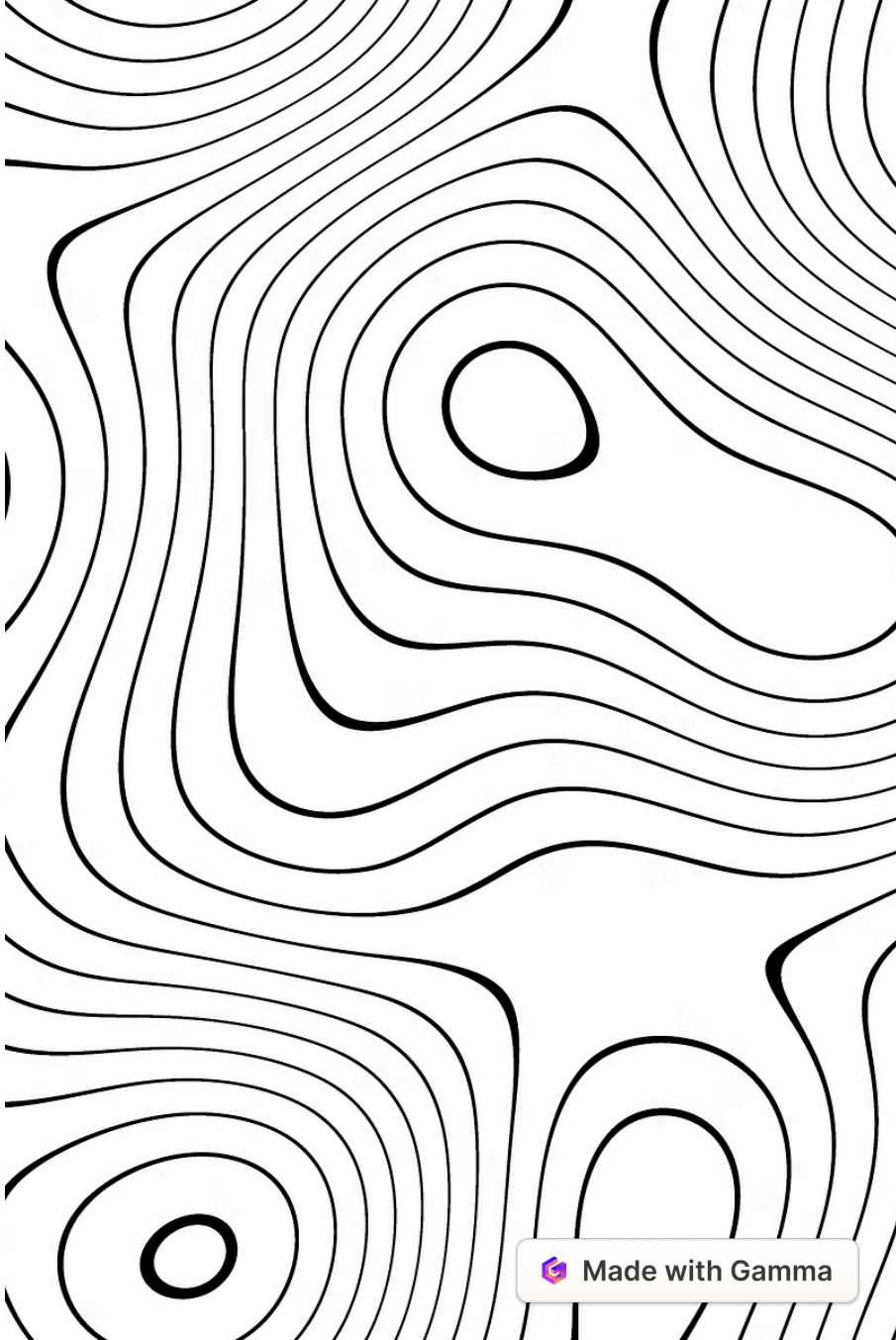
| | |
|---|---|
| [illegible] | [illegible] filesystem |
| [illegible] | [illegible] filesystem. -h parameter [illegible] fashion |
| [illegible] | [illegible] files that match the specified [illegible] |
| [illegible] | [illegible] file that matches the pattern, [illegible] |
| hdfs dfs -du -h /hadoop/file | Show the amount of space, in bytes, used by the files that match the specified file pattern. Formats the sizes of files in a human-readable fashion |

### ADMINISTRATION

| | |
|---|---|
| hdfs balancer -threshold 30 | Runs a cluster balancing utility. Percentage of disk capacity. This overwrites the default threshold |
| hadoop version | To check the version of Hadoop |
| hdfs fsck / | It checks the health of the Hadoop file system |
| hdfs dfsadmin -safemode leave | The command to turn off the safemode of NameNode |
| hdfs dfsadmin -refreshNodes | Re-read the hosts and exclude files to upd[illegible] allowed to connect to the Namenode and [illegible] decommissioned or recommissioned |
| hdfs namenode -format | Formats the NameNode |

# Creating HDFS Directory

- **Objective:** Create a directory structure in HDFS.
- **Command I:** hdfs dfs -mkdir -p /user/your_username/your_directory
- **Command II:** hdfs dfs -put /path/to/local/file /user/your_username/your_directory



```
    2 2022-03-22 23:58
66736 2022-03-22 23:57
47128 2022-03-22 23:57
88969 2022-03-22 23:58
 2539 2022-03-23 21:10
86359 2022-03-22 23:58


txt /user/cloudera/demo


66736 2022-03-22 23:57
47128 2022-03-22 23:57
88969 2022-03-22 23:58
86359 2022-03-22 23:58


    2 2022-03-22 23:58
 2539 2022-03-23 21:10
```

# Moving Files to HDFS

1

## Demonstrate

Explain the process of moving files from the local machine to HDFS.

2

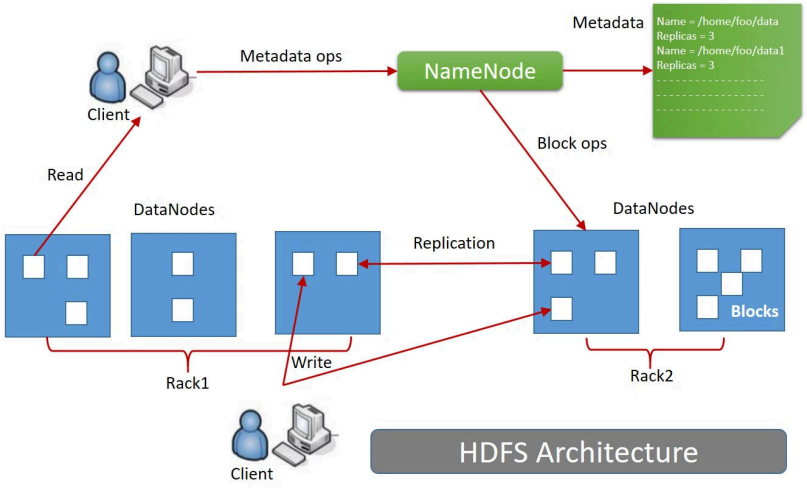
## Command Usage

Illustrate the use of the command: hdfs dfs -put /path/to/local/file /user/your_username/your_directory

# Exploring HDFS Data

Demonstrate how to explore and view data in Hadoop Distributed File System (HDFS) using the following commands:

- hdfs dfs -cat /user/your_username/your_directory/your_file
- hdfs dfs -ls /user/your_username/your_directory



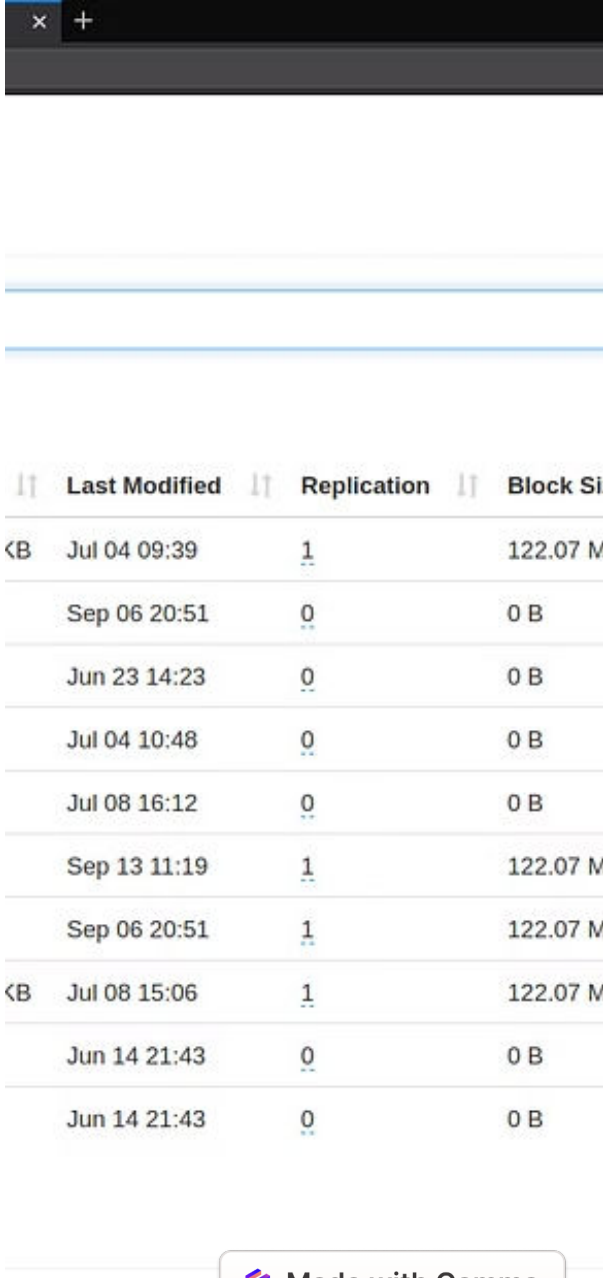
# Retrieving Data from HDFS

## Objective

Copy a file from HDFS to the local file system.

## Command I

Use `hdfs dfs -get /user/your_username/your_directory/your_file /path/to/local/directory` to copy the file to the local system.



# Lab Objectives

## Recap Objectives

Using HDFS commands to move data for processing.

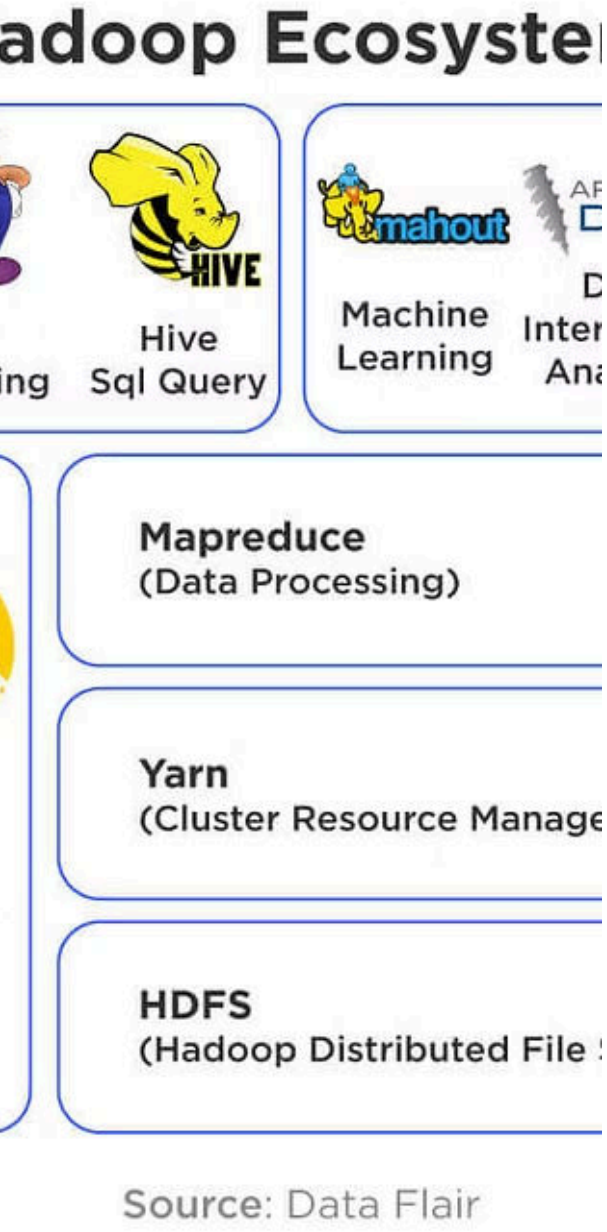
## Importance of Data Preparation

Emphasize the importance of preparing data in HDFS for subsequent processing.

# Executing HDFS Commands

Reinforce the importance of utilizing HDFS commands effectively for managing data.

Ensure that all files stored in HDFS can be easily accessed using the appropriate commands.




# Summary and Conclusion

1

## Recap Key Points

Review the main concepts of HDFS.

2

## Importance of Mastery

Highlight the significance of mastering HDFS commands.

3

## Efficient Data Management

Emphasize the role of Hadoop commands in efficient data management.

# Summary and Conclusion

Recap the key points covered in the presentation, emphasizing the importance of mastering HDFS commands for efficient data management in Hadoop. Understanding HDFS and its storage commands is fundamental for anyone working with big data. Efficient management and manipulation of data in HDFS are essential skills for data engineers and analysts.