

<b>Name of Student: Pushkar Sane</b>		
<b>Roll Number: 45</b>		<b>Lab Assignment Number: 1</b>
<b>Title of Lab Assignment: To study HDFS architecture and implement Hadoop commands.</b>		
<b>DOP: 16-08-2024</b>		<b>DOS: 23-08-2024</b>
<b>CO Mapped:</b> CO1	<b>PO Mapped:</b> PO1, PO2, PO3, PO4, PO5, PO7, PSO1, PSO2	<b>Signature:</b>

**Practical No.1**

**Aim:** To study HDFS architecture and implement Hadoop commands.

**Theory:**

HDFS is the primary or major component of the Hadoop ecosystem which is responsible for storing large data sets of structured or unstructured data across various nodes and thereby maintaining the metadata in the form of log files.

1. Namenode:

- The central node responsible for managing the filesystem metadata (e.g., file names, permissions, and replication details).
- It does not store the actual data but keeps track of which DataNode stores which data blocks.

2. Datanodes:

- These are the nodes where actual data is stored.
- The data is split into blocks and distributed across multiple DataNodes to ensure redundancy and fault tolerance.

3. Client:

- The client interacts with the HDFS, performing read and write operations.
- Read Operation: The client requests metadata from the Namenode and then reads data directly from the relevant DataNodes.
- Write Operation: The client writes data to the DataNodes, and the data is replicated to ensure fault tolerance.

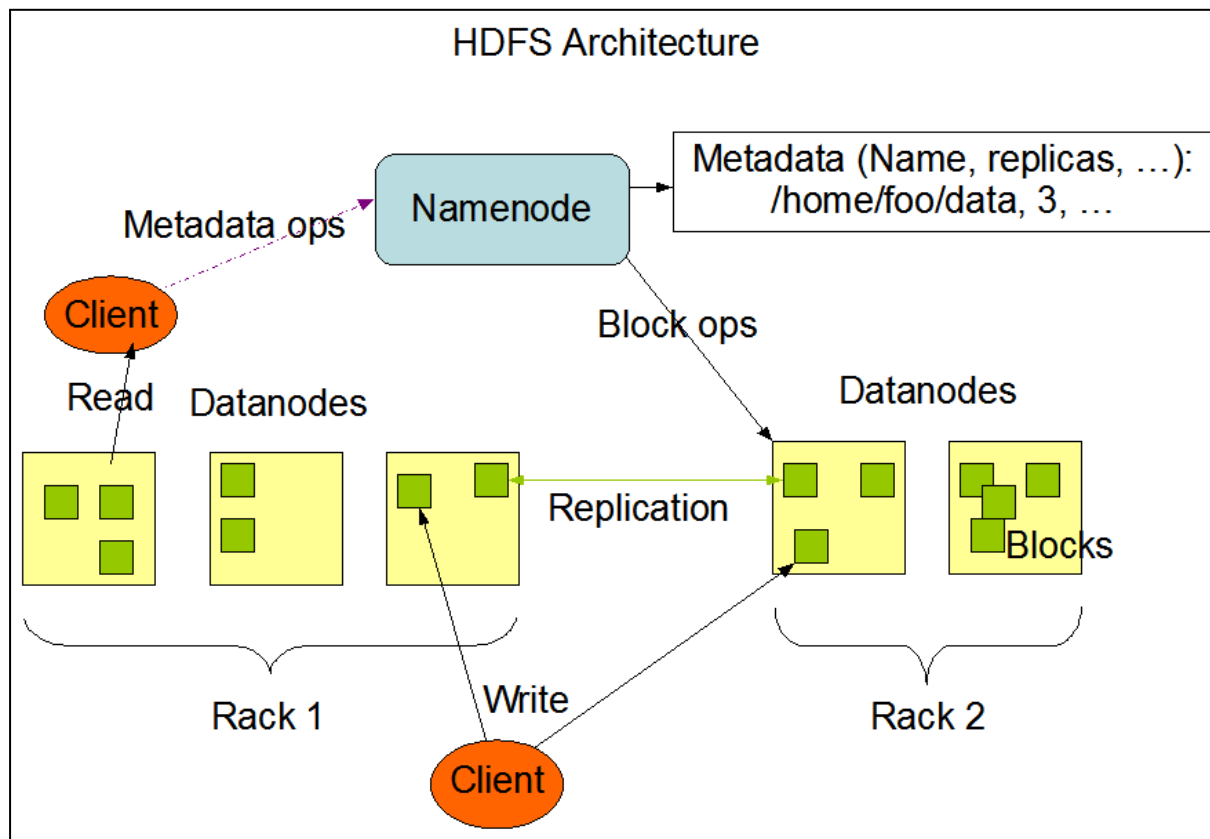
4. Replication:

- When the client writes data, it is replicated across multiple DataNodes (typically across different racks) to provide data redundancy and fault tolerance.
- The replication factor is determined by the Namenode.

5. Racks:

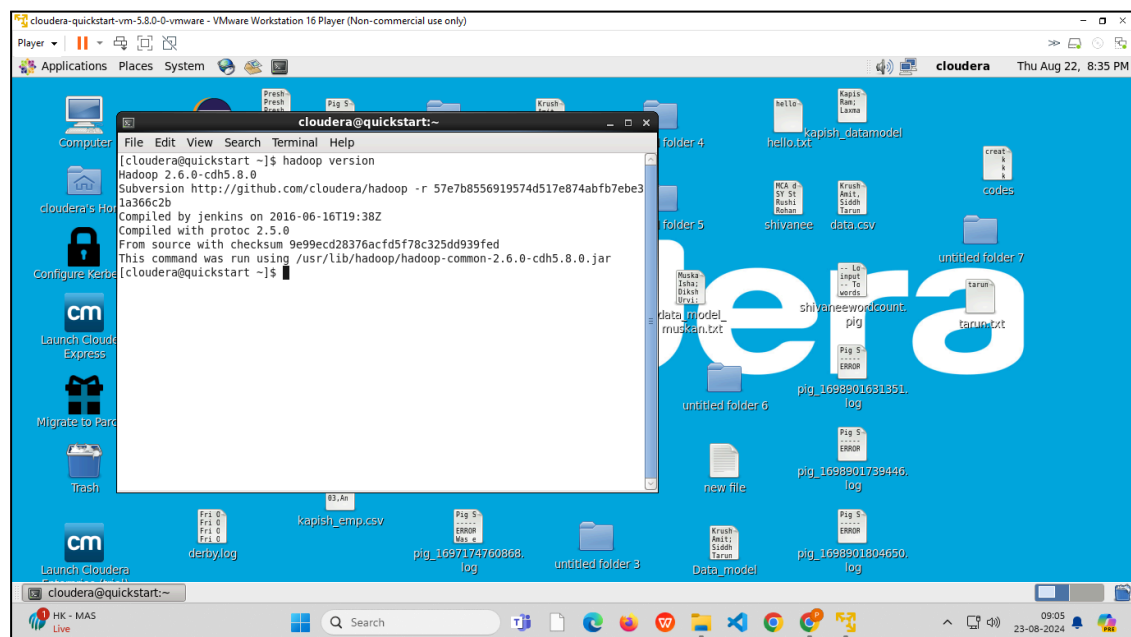
The system typically organizes DataNodes into racks. Data replication is done across racks to avoid data loss in case of a rack failure.

This architecture is designed to provide scalable, fault-tolerant storage in a distributed computing environment.

**Diagram of HDFS Architecture:****1. Hadoop Version**

Description: The Hadoop fs shell command version prints the Hadoop version.

Command: `hadoop version`



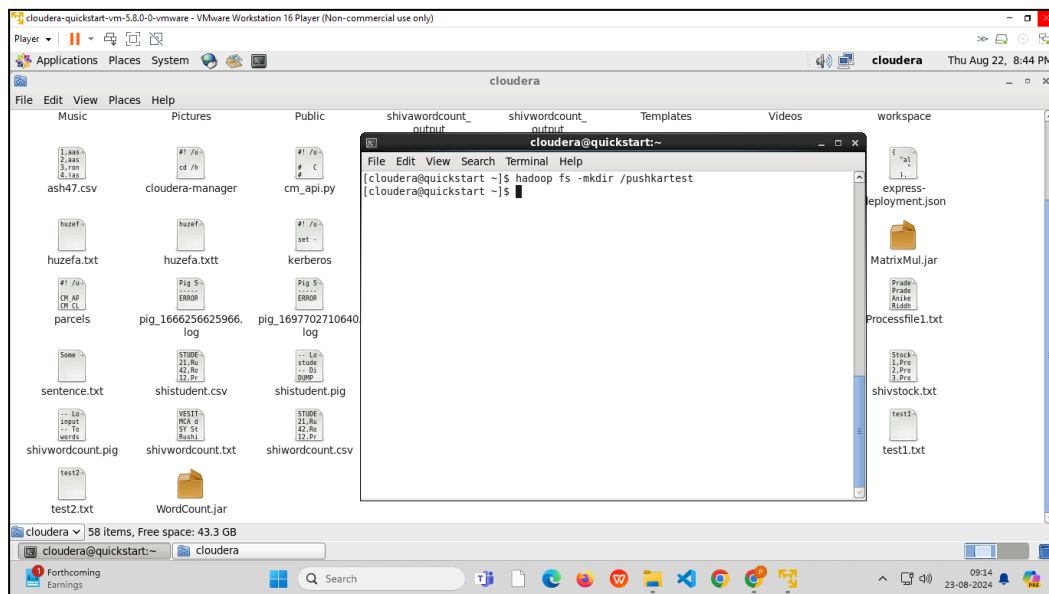
## 2. Make Directory

Description: This command creates the directory in HDFS if it does not already exist.

Note: If the directory already exists in HDFS, then we will get an error message that the file already exists.

Command:

`hadoop fs -mkdir /path/directory_name`



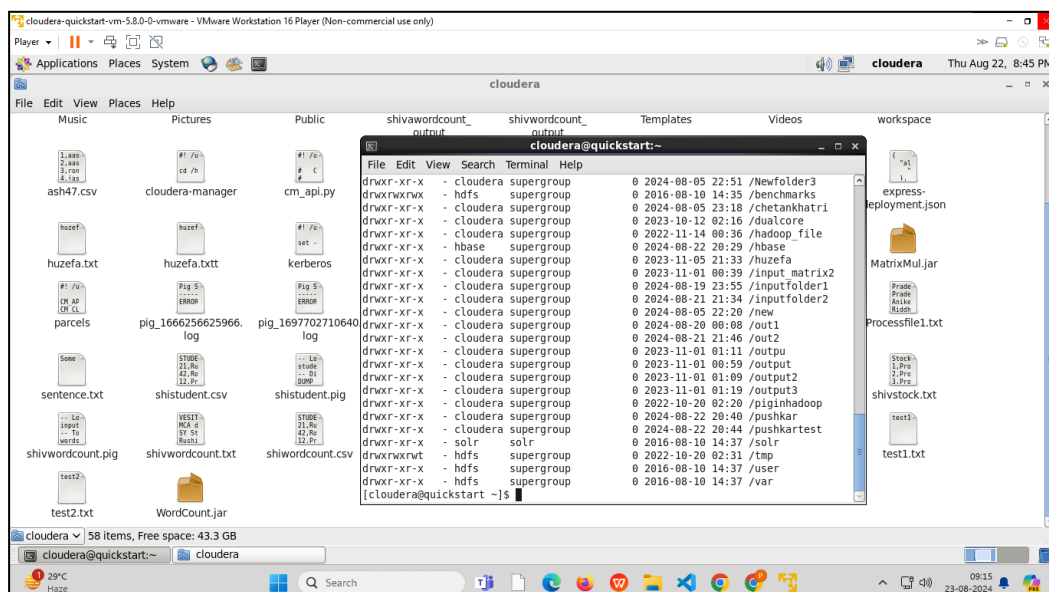
## 3. Listing Directories

Description:

Using the `ls` command, we can check for the directories in HDFS.

Command:

`hadoop fs -ls /`



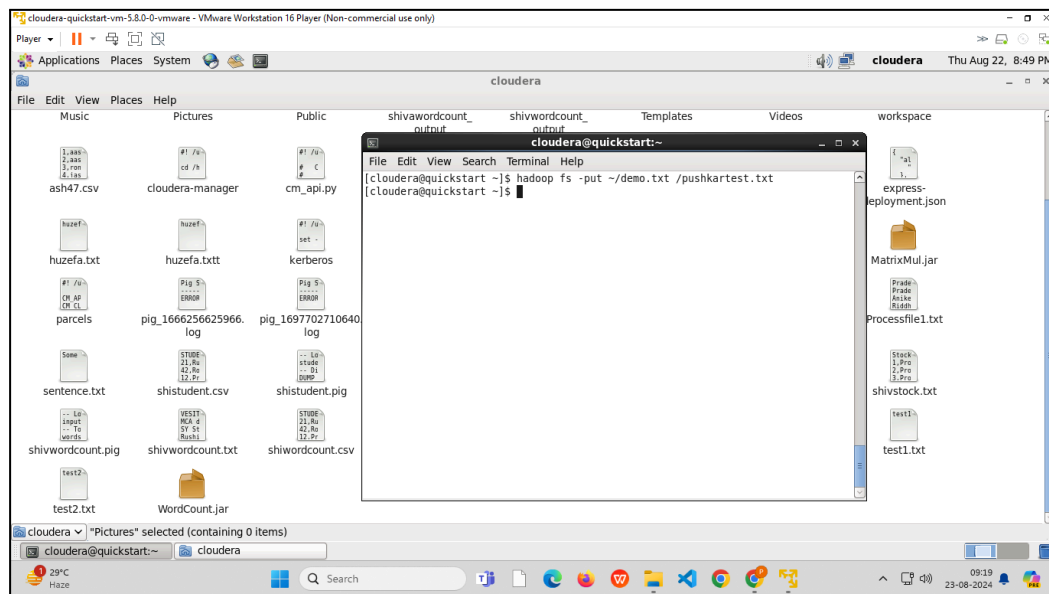
#### 4. copyFromLocal or put

Description:

The Hadoop fs shell command is similar to the copyFromLocal, which copies files or directory from the local filesystem to the destination in the Hadoop filesystem.

Command:

`hadoop fs -put ~/localfilepath /fileofdestination`



#### 5. count

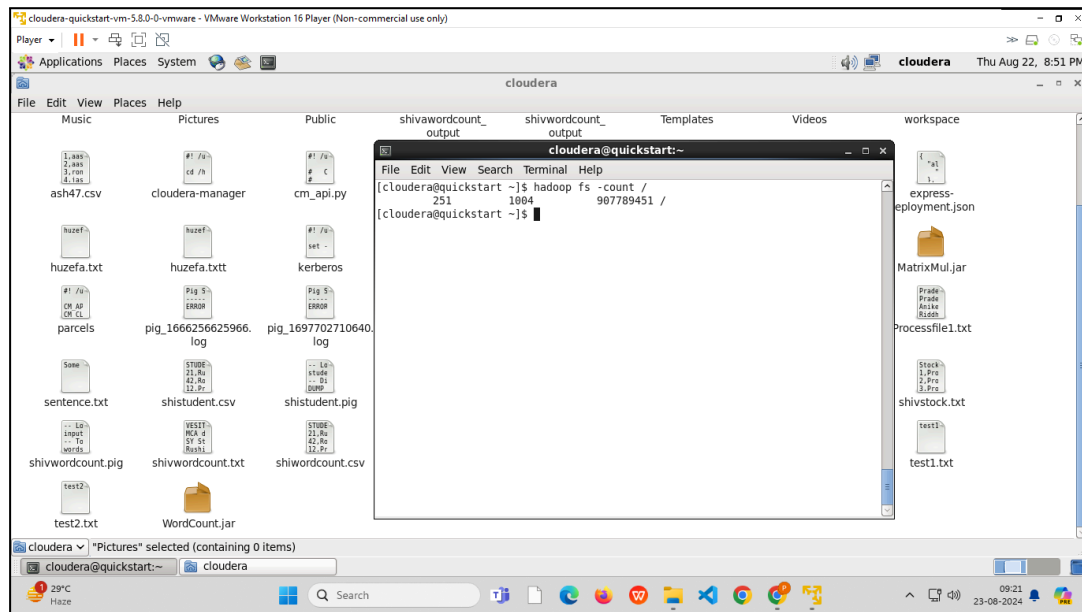
Description: The Hadoop fs shell command count counts the number of files, directories, and bytes under the paths that match the specified file pattern.

Options:

- q – shows quotas(quota is the hard limit on the number of names and amount of space used for individual directories)
- u – it limits output to show quotas and usage only
- h – shows size in a human-readable format
- v – shows header line

Command:

`hadoop fs -count /`

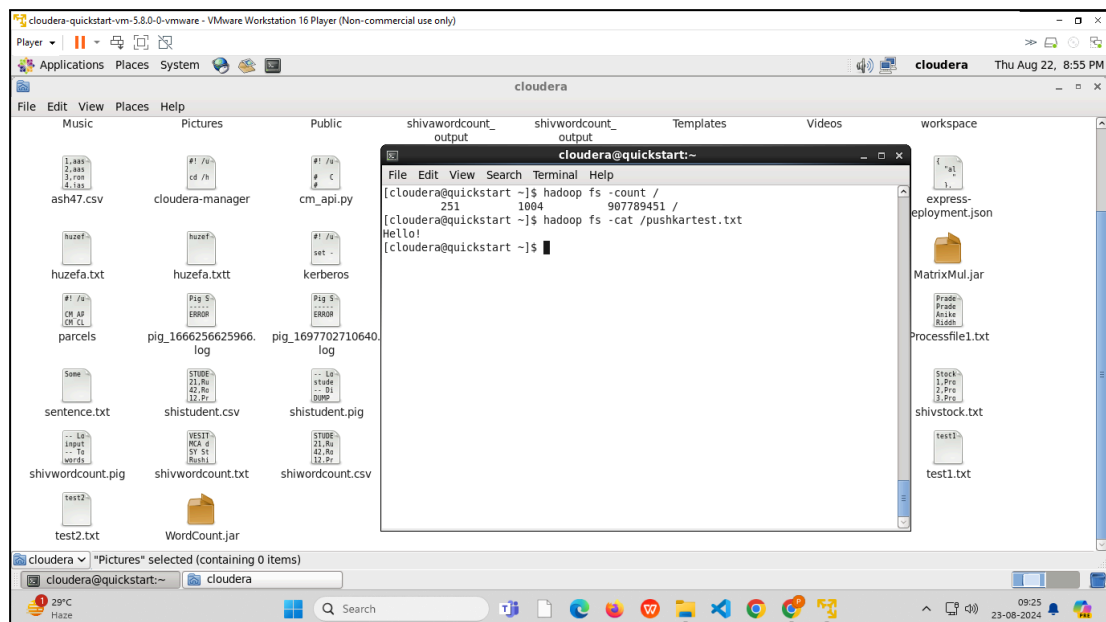


## 6. cat

Description: The cat command reads the file in HDFS and displays the content of the file on console or stdout.

Command:

`hadoop fs -cat /path`

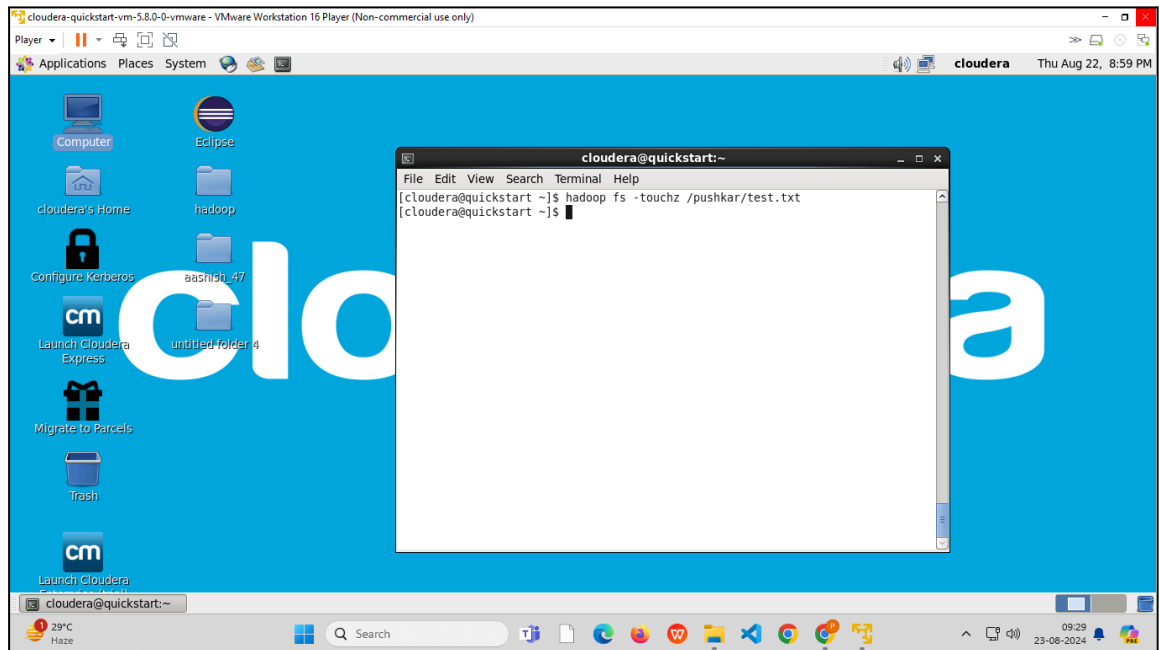


## 7. touchz

Description: touchz command creates a file in HDFS with file size equal to 0 byte. The directory is the name of the directory where we will create the file, and filename is the name of the new file we are going to create

Command:

`hadoop fs -touchz /directory/file`



## 8. stat:

Description:

The Hadoop fs shell command `stat` prints the statistics about the file or directory in the specified format.

Formats:

`%b` – file size in bytes

`%g` – group name of owner

`%n` – file name

`%o` – block size

`%r` – replication

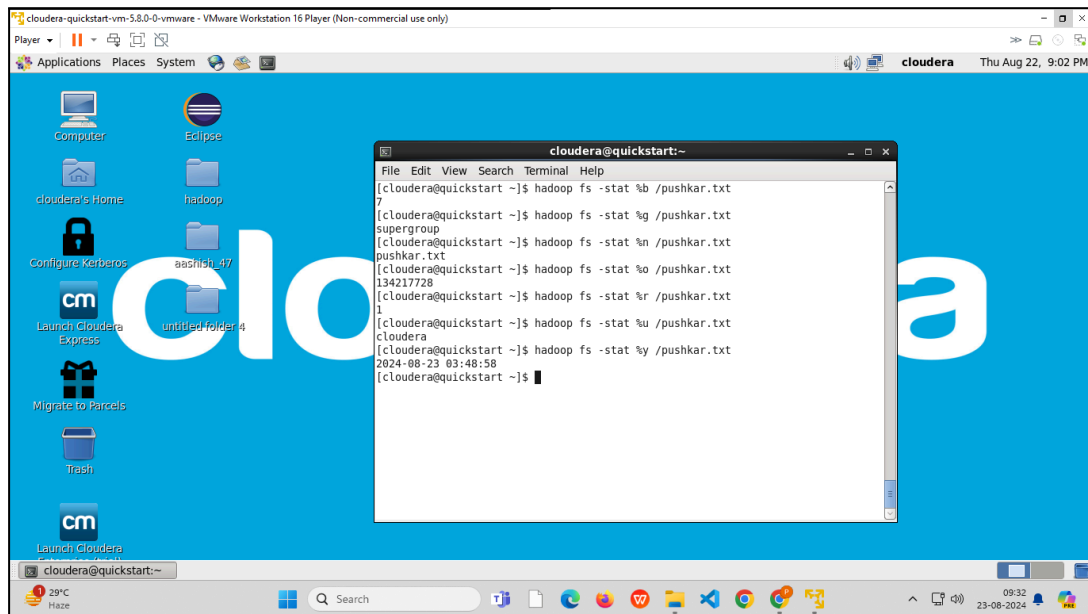
`%u` – user name of owner

`%y` – modification date

If the format is not specified then `%y` is used by default.

Command:

`hadoop fs -stat %format /path`



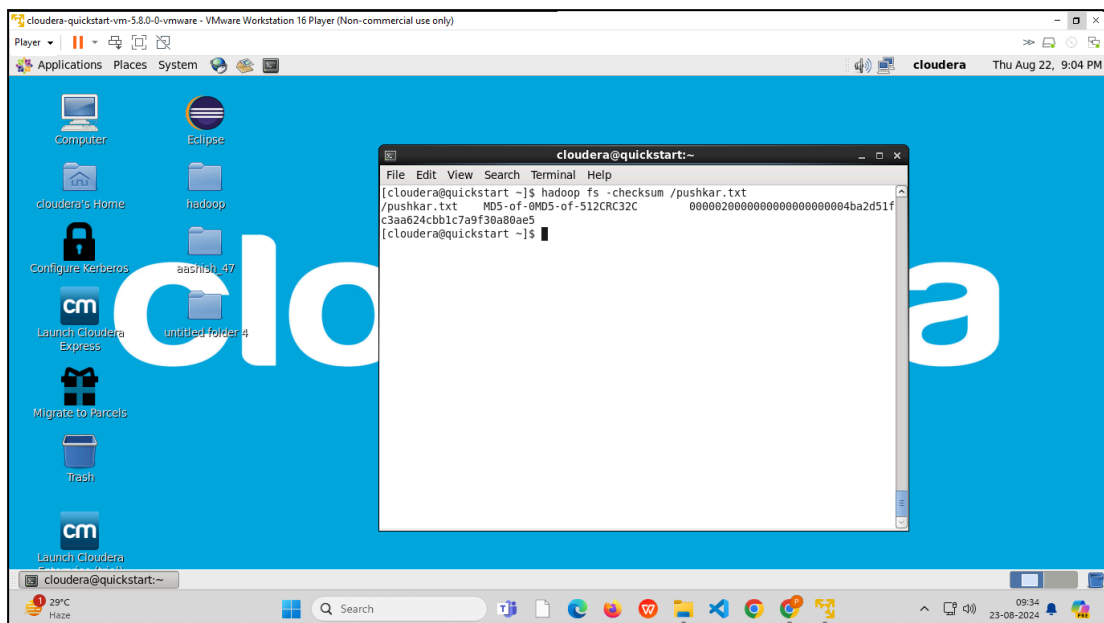
```
cloudera@quickstart:~$ hadoop fs -stat %b /pushkar.txt
7
[cloudera@quickstart ~]$ hadoop fs -stat %g /pushkar.txt
supergroup
[cloudera@quickstart ~]$ hadoop fs -stat %n /pushkar.txt
pushkar.txt
[cloudera@quickstart ~]$ hadoop fs -stat %o /pushkar.txt
134217728
[cloudera@quickstart ~]$ hadoop fs -stat %r /pushkar.txt
1
[cloudera@quickstart ~]$ hadoop fs -stat %u /pushkar.txt
cloudera
[cloudera@quickstart ~]$ hadoop fs -stat %y /pushkar.txt
2024-08-23 03:48:58
[cloudera@quickstart ~]$
```

## 9. checksum

Description: The Hadoop fs shell command checksum returns the checksum information of a file.

Command:

`hadoop fs -checksum /path`



```
cloudera@quickstart:~$ hadoop fs -checksum /pushkar.txt
/pushkar.txt MD5-of-0MD5-of-512CRC32C 0000020000000000000000004ba2d51f
c3aa624cbb1c7a9f30a80ae5
[cloudera@quickstart ~]$
```

## 10. usage

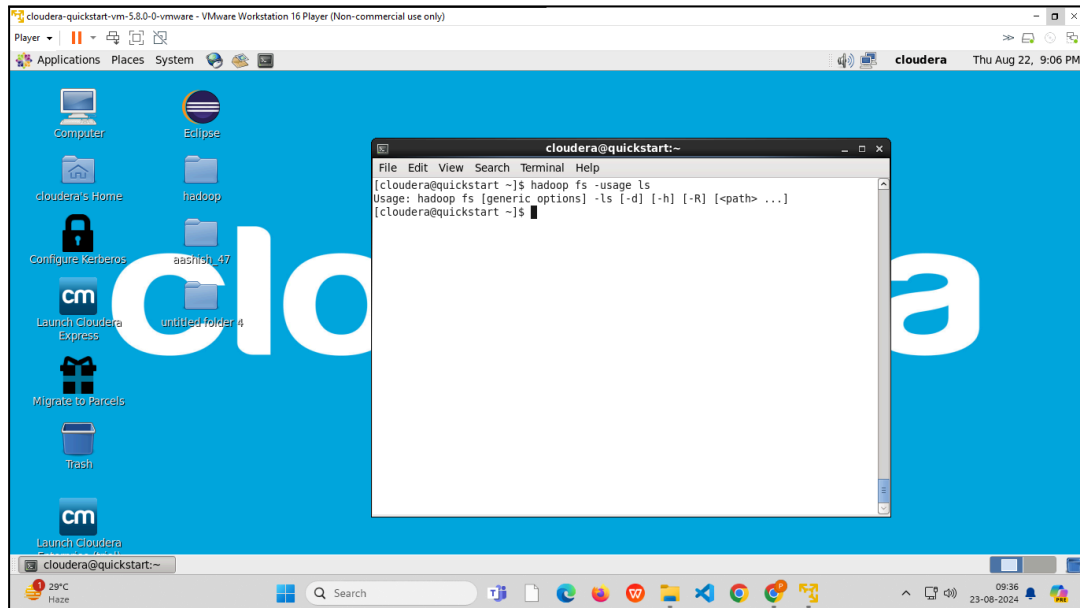
Description:

The Hadoop fs shell command usage returns the help for an individual command.



Command:

`hadoop fs -usage [command]`



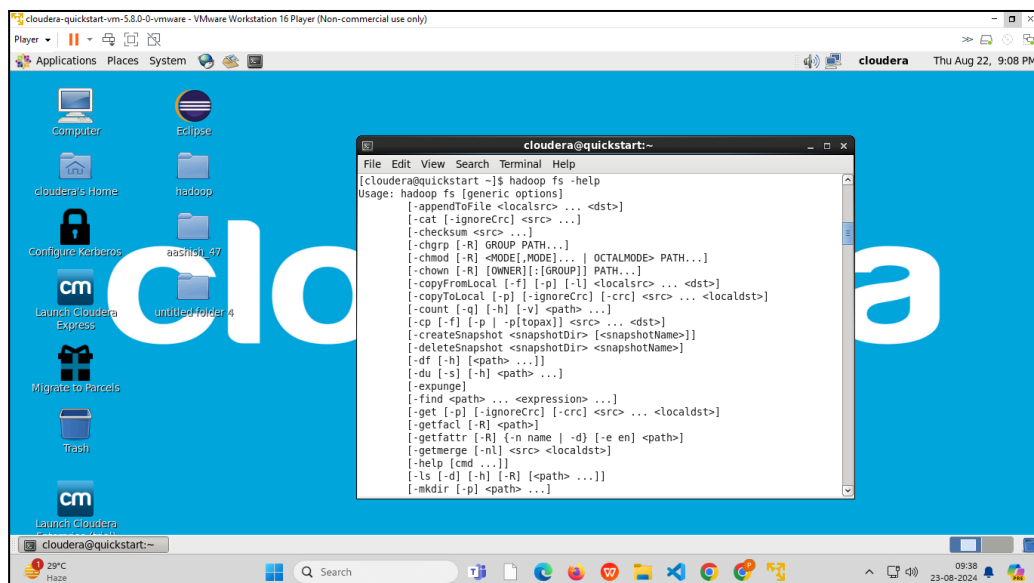
## 11. help

Description:

The Hadoop fs shell command help shows help for all the commands or the specified command.

Command:

`hadoop fs -help [command]`

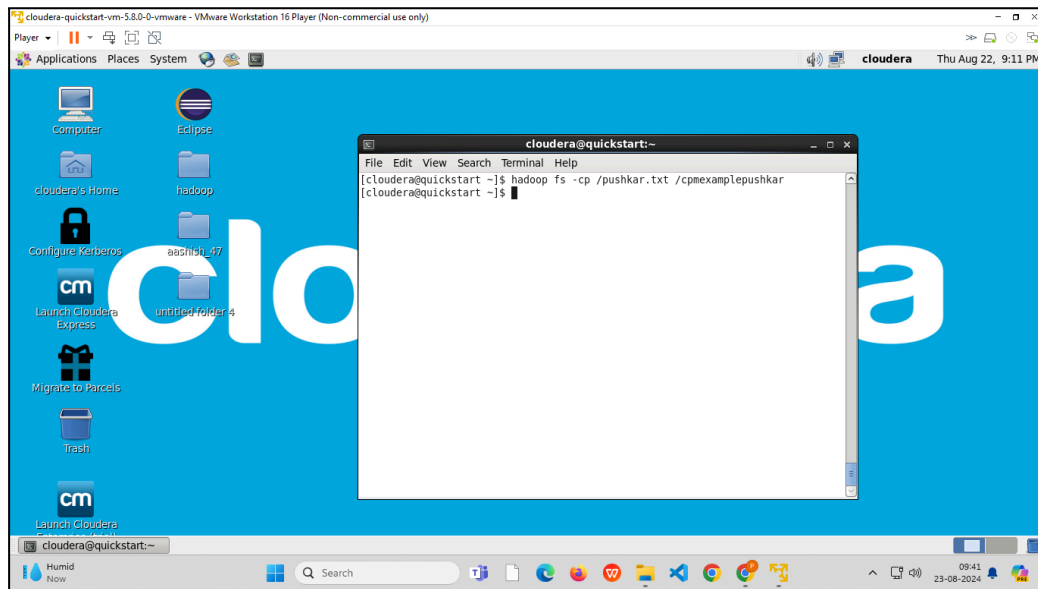


**12. cp**

Description: The cp command copies a file from one directory to another directory within the HDFS.

Command:

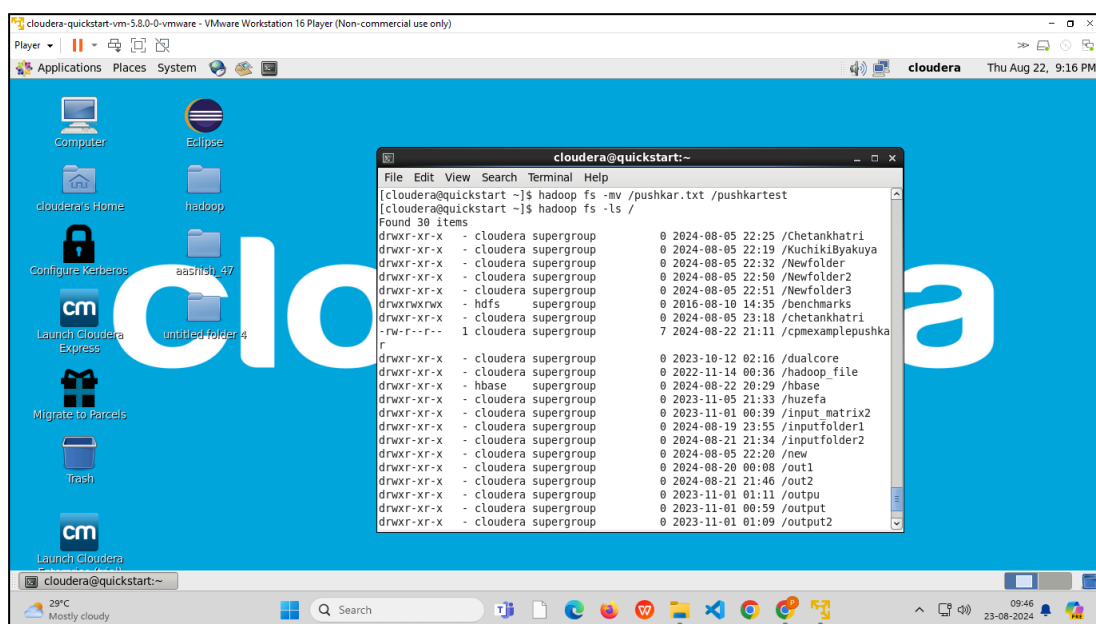
`hadoop fs -cp <src> <des>`

**13. mv**

Description: The HDFS mv command moves the files or directories from the source to a destination within HDFS.

Command:

`hadoop fs -mv <src> <des>`



**Conclusion:**

From this practical, I have successfully studied and implemented the hadoop commands.