Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

2020

Department of Information Technology

A.Y. 2024-2025

Class: T.Y. B. Tech-IT, Semester: VI

Subject: Big Data Lab

Experiment – 3: Study HDFS Commands

1. **Aim:** Study HDFS Commands**.**

2. **Objectives:** After study of this experiment, the student will be able to
    Implement the basics hands-on commands on hadoop distributed file system.
3. **Outcomes:**
    One can create and store file and directories in HDFS.
4. **Prerequisite:** Basics of Hadoop, Cloud concepts.

5. **Requirements:** PC, Internet and VMWare software, Cloudera.

6. **Theory:**
   Students need to write the theory of the following points.

1. Describe working and the core components of HDFS in detail.

HDFS (Hadoop Distributed File System) – Overview

HDFS is a distributed storage system designed for handling large datasets efficiently across multiple machines. It follows a **master-slave architecture** and is optimized for **high throughput and fault tolerance**.

1. How HDFS Works

- Files are **split into blocks** (default 128MB/256MB) and **replicated** across nodes.
- **Read Operation**: The client queries the **NameNode** for metadata, then retrieves blocks directly from **DataNodes**.
- **Write Operation**: The client writes data to DataNodes, and blocks are replicated.
- **Fault Tolerance**: Replication and periodic **heartbeats** ensure data reliability.

2. Core Components of HDFS

1. **NameNode (Master)**: Manages metadata (file structure, block locations).
2. **DataNodes (Slaves)**: Store actual data blocks and send heartbeats to the NameNode.
3. **Secondary NameNode**: Merges logs with metadata for efficient recovery.
4. **HDFS Client**: Interface for users to interact with HDFS.

3. Key Features

- **Fault Tolerance** (replication across nodes).
- **Scalability** (adds nodes dynamically).
- **Write-Once, Read-Many** (optimized for batch processing).
- **Data Locality** (moves computation closer to data).

Conclusion

HDFS is a **reliable, scalable, and high-performance storage system** for Big Data applications, ensuring **efficient storage and processing** across distributed environments.

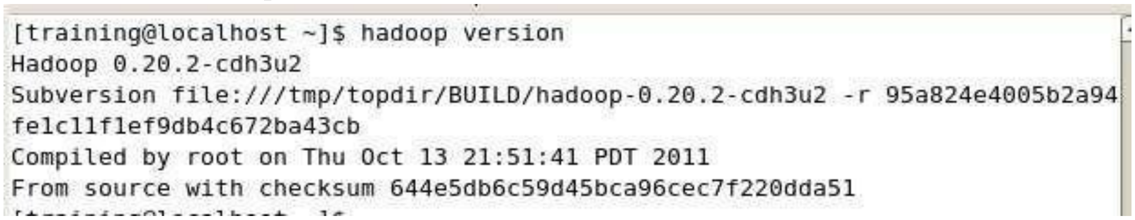
7. Laboratory Exercise

Procedure

1. **Download Cloudera and VMware**
2. Open VMWare Workstation > select Cloudera 1.5/1.6 > open virtual machine > in terminals perform the HDFS commands.
3. **Execute the following commands: (Refer links in references)**

<u>Commands:</u> Minimum 20 commands has to be executed.
Examples of which are as follows:

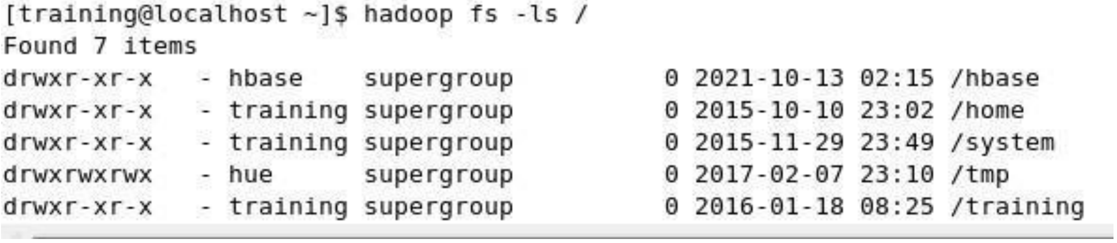
1. Print the Hadoop version:



```
[training@localhost ~]$ hadoop version
Hadoop 0.20.2-cdh3u2
Subversion file:///tmp/topdir/BUILD/hadoop-0.20.2-cdh3u2 -r 95a824e4005b2a94
fe1c11f1ef9db4c672ba43cb
Compiled by root on Thu Oct 13 21:51:41 PDT 2011
From source with checksum 644e5db6c59d45bca96cec7f220dda51
```

hadoop version

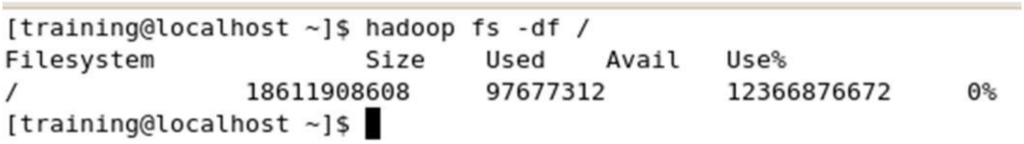
2. List the contents of the root directory in HDFS:



```
[training@localhost ~]$ hadoop fs -ls /
Found 7 items
drwxr-xr-x   - hbase    supergroup          0 2021-10-13 02:15 /hbase
drwxr-xr-x   - training supergroup          0 2015-10-10 23:02 /home
drwxr-xr-x   - training supergroup          0 2015-11-29 23:49 /system
drwxrwxrwx   - hue      supergroup          0 2017-02-07 23:10 /tmp
drwxr-xr-x   - training supergroup          0 2016-01-18 08:25 /training
```

hadoop fs -ls /

3. Report the amount of space used and available on the currently mounted file system:

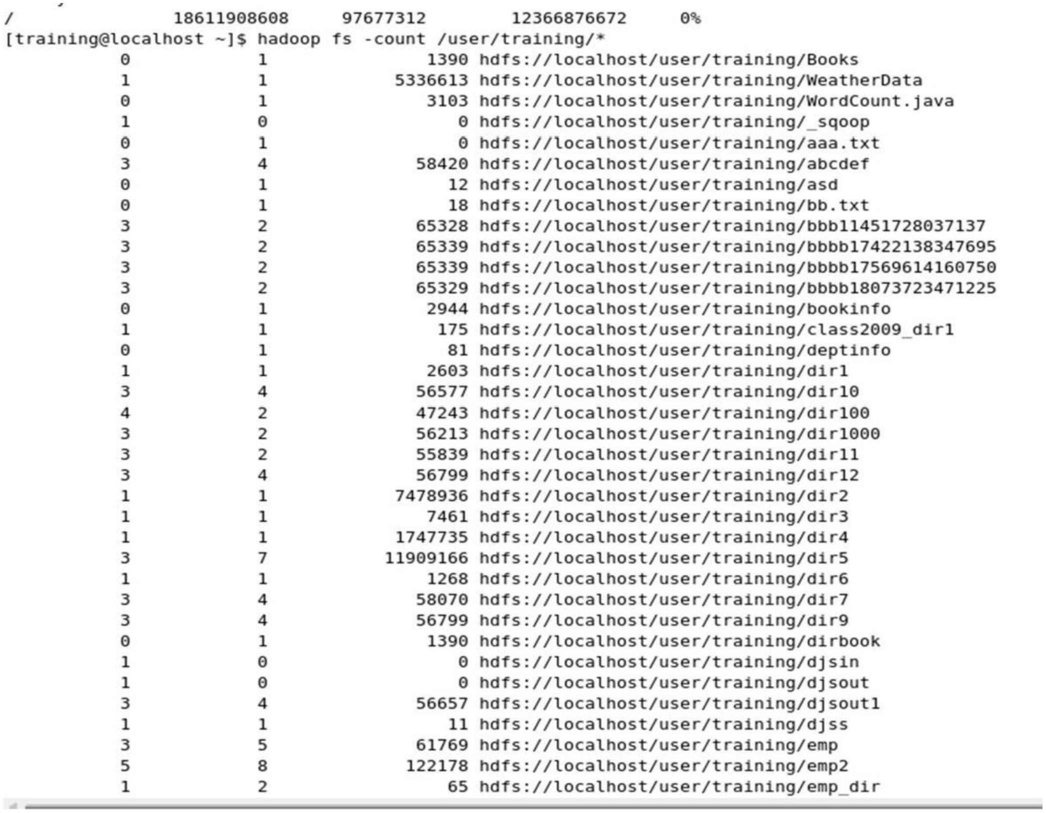


```
[training@localhost ~]$ hadoop fs -df /
Filesystem              Size     Used     Avail    Use%
/               18611908608     97677312         12366876672      0%
[training@localhost ~]$
```

hadoop

fs -df /

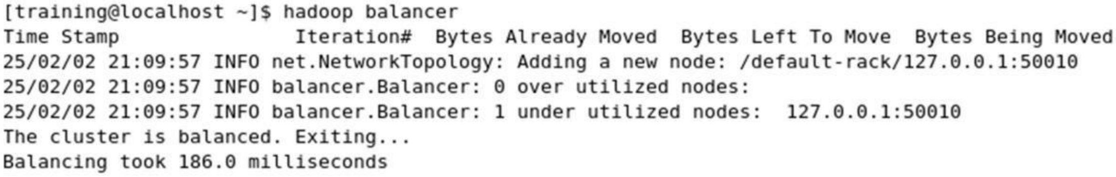
4. Count the number of directories, files, and bytes under the paths that match the specified file pattern:



```
/              18611908608      97677312        12366876672      0%
[training@localhost ~]$ hadoop fs -count /user/training/*
           0            1               1390 hdfs://localhost/user/training/Books
           1            1            5336613 hdfs://localhost/user/training/WeatherData
           0            1               3103 hdfs://localhost/user/training/WordCount.java
           1            0                  0 hdfs://localhost/user/training/_sqoop
           0            1                  0 hdfs://localhost/user/training/aaa.txt
           3            4              58420 hdfs://localhost/user/training/abcdef
           0            1                 12 hdfs://localhost/user/training/asd
           0            1                 18 hdfs://localhost/user/training/bb.txt
           3            2              65328 hdfs://localhost/user/training/bbb11451728037137
           3            2              65339 hdfs://localhost/user/training/bbbb17422138347695
           3            2              65339 hdfs://localhost/user/training/bbbb17569614160750
           3            2              65329 hdfs://localhost/user/training/bbbb18073723471225
           0            1               2944 hdfs://localhost/user/training/bookinfo
           1            1                175 hdfs://localhost/user/training/class2009_dir1
           0            1                 81 hdfs://localhost/user/training/deptinfo
           1            1               2603 hdfs://localhost/user/training/dir1
           3            4              56577 hdfs://localhost/user/training/dir10
           4            2              47243 hdfs://localhost/user/training/dir100
           3            2              56213 hdfs://localhost/user/training/dir1000
           3            2              55839 hdfs://localhost/user/training/dir11
           3            4              56799 hdfs://localhost/user/training/dir12
           1            1            7478936 hdfs://localhost/user/training/dir2
           1            1               7461 hdfs://localhost/user/training/dir3
           1            1            1747735 hdfs://localhost/user/training/dir4
           3            7           11909166 hdfs://localhost/user/training/dir5
           1            1               1268 hdfs://localhost/user/training/dir6
           3            4              58070 hdfs://localhost/user/training/dir7
           3            4              56799 hdfs://localhost/user/training/dir9
           0            1               1390 hdfs://localhost/user/training/dirbook
           1            0                  0 hdfs://localhost/user/training/djsin
           1            0                  0 hdfs://localhost/user/training/djsout
           3            4              56657 hdfs://localhost/user/training/djsout1
           1            1                 11 hdfs://localhost/user/training/djss
           3            5              61769 hdfs://localhost/user/training/emp
           5            8             122178 hdfs://localhost/user/training/emp2
           1            2                 65 hdfs://localhost/user/training/emp_dir
```

hadoop fs -count /user/training/*

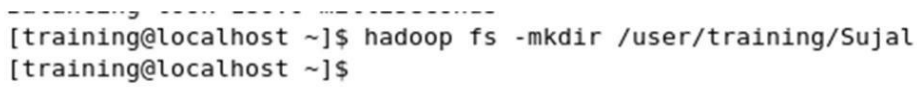
5. Run a cluster balancing utility:



```
[training@localhost ~]$ hadoop balancer
Time Stamp               Iteration#  Bytes Already Moved  Bytes Left To Move  Bytes Being Moved
25/02/02 21:09:57 INFO net.NetworkTopology: Adding a new node: /default-rack/127.0.0.1:50010
25/02/02 21:09:57 INFO balancer.Balancer: 0 over utilized nodes:
25/02/02 21:09:57 INFO balancer.Balancer: 1 under utilized nodes:  127.0.0.1:50010
The cluster is balanced. Exiting...
Balancing took 186.0 milliseconds
```

hadoop balancer

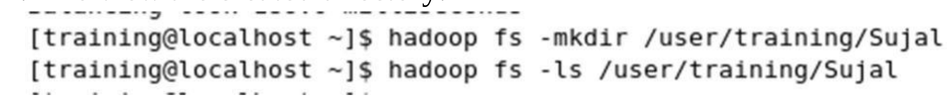
6. Create a new directory named "Abc":



```
[training@localhost ~]$ hadoop fs -mkdir /user/training/Sujal
[training@localhost ~]$
```

hadoop fs -mkdir /user/training/Abc

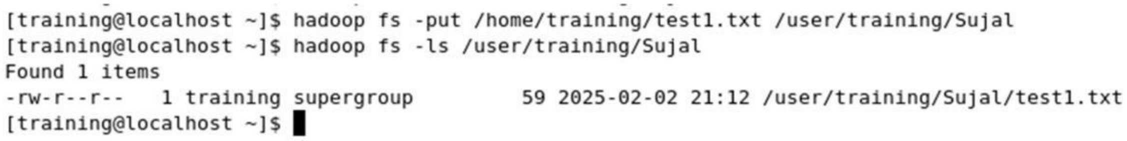
7. To view the created directory:



```
[training@localhost ~]$ hadoop fs -mkdir /user/training/Sujal
[training@localhost ~]$ hadoop fs -ls /user/training/Sujal
```

hadoop fs -ls /user/training/Abc

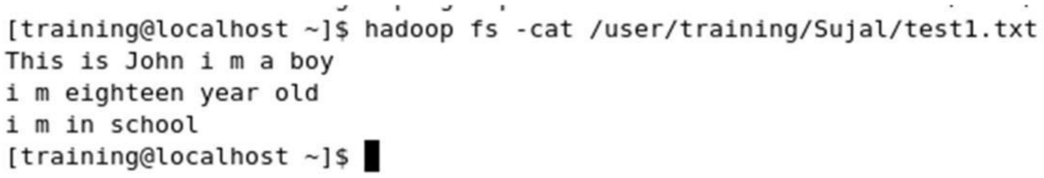
8. Adding a file to the directory that we have just created:



```
[training@localhost ~]$ hadoop fs -put /home/training/test1.txt /user/training/Sujal
[training@localhost ~]$ hadoop fs -ls /user/training/Sujal
Found 1 items
-rw-r--r--   1 training supergroup         59 2025-02-02 21:12 /user/training/Sujal/test1.txt
[training@localhost ~]$ 
```

hadoop fs -put /home/training/test1.txt /user/training/Abc/

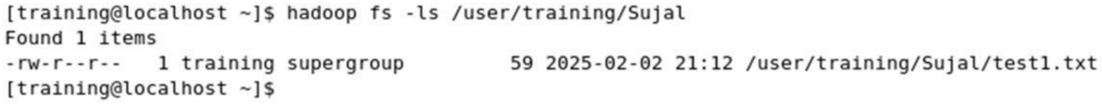
9. To view the contents of the test1.txt file:



```
[training@localhost ~]$ hadoop fs -cat /user/training/Sujal/test1.txt
This is John i m a boy
i m eighteen year old
i m in school
[training@localhost ~]$ 
```

hadoop fs -cat /user/training/Abc/test1.txt

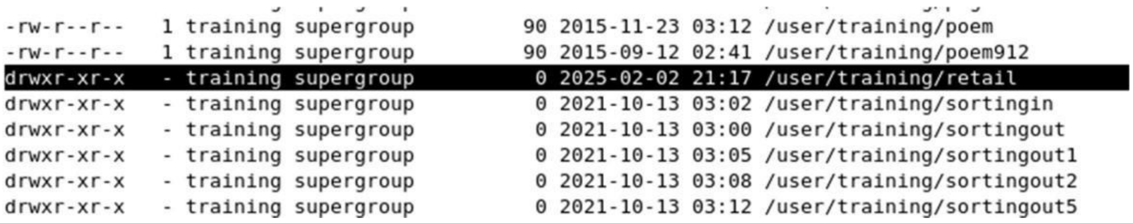
10. List the contents of the directory:



```
[training@localhost ~]$ hadoop fs -ls /user/training/Sujal
Found 1 items
-rw-r--r--   1 training supergroup         59 2025-02-02 21:12 /user/training/Sujal/test1.txt
[training@localhost ~]$
```

hadoop fs -ls /user/training/Abc

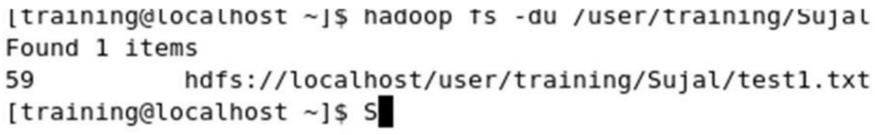
11. Add the entire local directory called "retail" to the /user/training directory in HDFS:



```
-rw-r--r--   1 training supergroup         90 2015-11-23 03:12 /user/training/poem
-rw-r--r--   1 training supergroup         90 2015-09-12 02:41 /user/training/poem912
drwxr-xr-x   - training supergroup          0 2025-02-02 21:17 /user/training/retail
drwxr-xr-x   - training supergroup          0 2021-10-13 03:02 /user/training/sortingin
drwxr-xr-x   - training supergroup          0 2021-10-13 03:00 /user/training/sortingout
drwxr-xr-x   - training supergroup          0 2021-10-13 03:05 /user/training/sortingout1
drwxr-xr-x   - training supergroup          0 2021-10-13 03:08 /user/training/sortingout2
drwxr-xr-x   - training supergroup          0 2021-10-13 03:12 /user/training/sortingout5
```

hadoop fs -put /home/training/retail /user/training/ 12. See how much

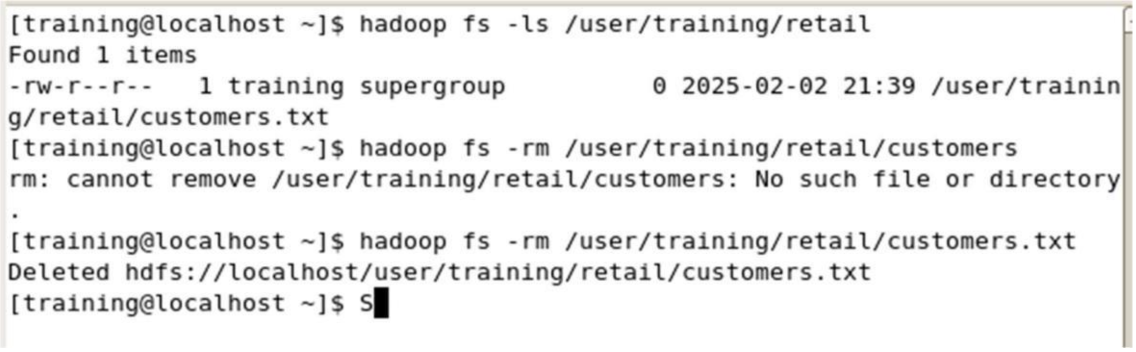
space this directory occupies in HDFS:



```
[training@localhost ~]$ hadoop fs -du /user/training/Sujal
Found 1 items
59          hdfs://localhost/user/training/Sujal/test1.txt
[training@localhost ~]$ S
```

hadoop fs -du -h /user/training/retail

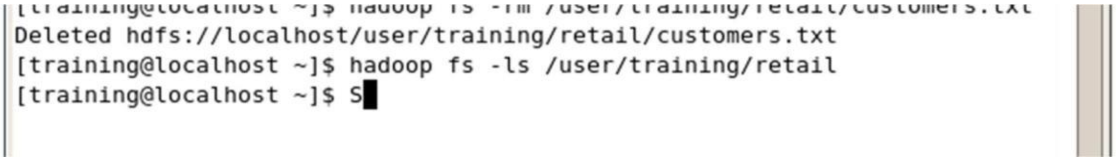
13. Delete a file ‘customers’ from the “retail” directory:



```
[training@localhost ~]$ hadoop fs -ls /user/training/retail
Found 1 items
-rw-r--r--   1 training supergroup          0 2025-02-02 21:39 /user/trainin
g/retail/customers.txt
[training@localhost ~]$ hadoop fs -rm /user/training/retail/customers
rm: cannot remove /user/training/retail/customers: No such file or directory
.
[training@localhost ~]$ hadoop fs -rm /user/training/retail/customers.txt
Deleted hdfs://localhost/user/training/retail/customers.txt
[training@localhost ~]$ S
```

hadoop fs -rm /user/training/retail/customers

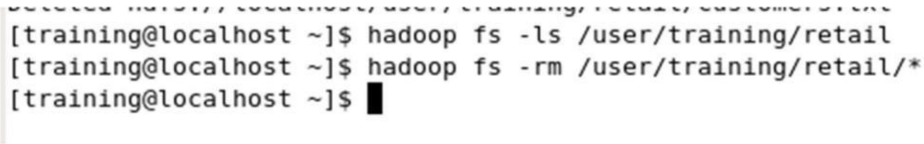
14. Ensure this file is no longer in HDFS:



```
[training@localhost ~]$ hadoop fs -rm /user/training/retail/customers.txt
Deleted hdfs://localhost/user/training/retail/customers.txt
[training@localhost ~]$ hadoop fs -ls /user/training/retail
[training@localhost ~]$ S
```

hadoop fs -ls /user/training/retail

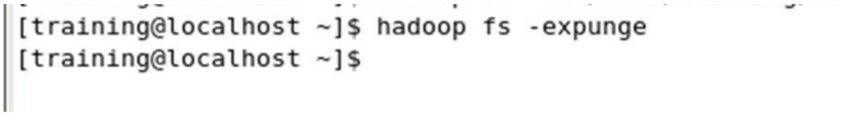
15. Delete all files from the “retail” directory using a wildcard:



```
Deleted hdfs://localhost/user/training/retail/customers.txt
[training@localhost ~]$ hadoop fs -ls /user/training/retail
[training@localhost ~]$ hadoop fs -rm /user/training/retail/*
[training@localhost ~]$ 
```

hadoop fs -rm /user/training/retail/* 16. To empty the

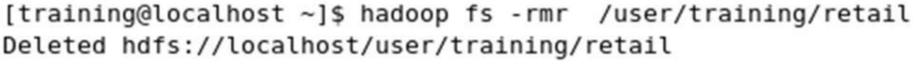
trash:



```
[training@localhost ~]$ hadoop fs -expunge
[training@localhost ~]$
```

hadoop fs -expunge

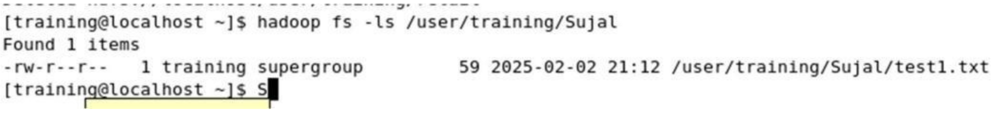
17. Finally, remove the entire retail directory and all of its contents in HDFS:



```
[training@localhost ~]$ hadoop fs -rmr  /user/training/retail
Deleted hdfs://localhost/user/training/retail
```

hadoop fs -rm -r /user/training/retail

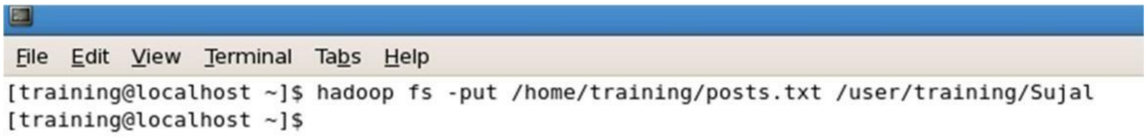
18. List the Abc directory again:



```
[training@localhost ~]$ hadoop fs -ls /user/training/Sujal
Found 1 items
-rw-r--r--   1 training supergroup         59 2025-02-02 21:12 /user/training/Sujal/test1.txt
[training@localhost ~]$ S
```

hadoop fs -ls /user/training/Abc

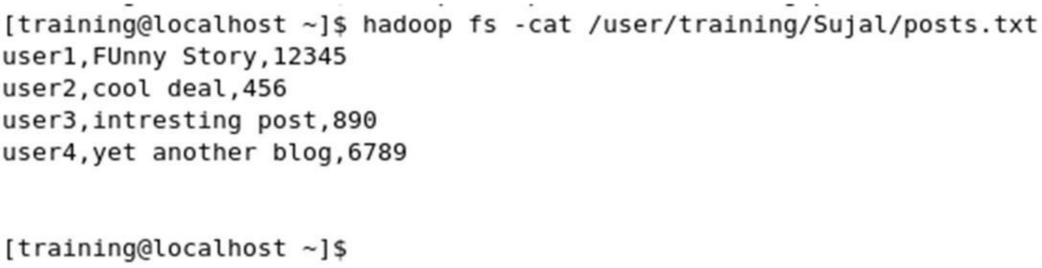
19. Add the posts.txt file from the local directory named “/home/training/” to the Hadoop directory you created in HDFS:



```
[training@localhost ~]$ hadoop fs -put /home/training/posts.txt /user/training/Sujal
[training@localhost ~]$
```

hadoop fs -put /home/training/posts.txt /user/training/Abc/

20. To view the contents of your text file posts.txt which is present in your Hadoop directory:



```
[training@localhost ~]$ hadoop fs -cat /user/training/Sujal/posts.txt
user1,FUnny Story,12345
user2,cool deal,456
user3,intresting post,890
user4,yet another blog,6789


[training@localhost ~]$
```

hadoop fs -cat /user/training/Abc/posts.txt

8. Result/Observation/Program code

Students have to implement the command with proper syntax for the above stated sample list and attached the output for the same.

9. Conclusion:

1. Write what was performed in the experiment.

Performed in the Experiment

- Executed **basic HDFS commands** such as file creation, deletion, and directory management.
- Uploaded and retrieved files using **HDFS shell commands**.
- Verified data replication and block storage across nodes.

2. Mention few applications of what was studied.

. Applications

- **Big Data Processing** (Hadoop ecosystem).
- **Data Storage & Management** (Cloud and enterprise systems).
- **Machine Learning & AI** (Handling large datasets).
- **Log & Event Analysis** (Storing logs in distributed environments).

3. Write the significance of the topic studied in the experiment.

. Significance

- **Efficient data storage** and retrieval in distributed systems.
- **Fault tolerance** ensures data reliability.
- **Scalability** for handling large datasets.
- **Foundation for Big Data applications** in analytics and AI.

10. References:

1. Alex Holmes “Hadoop in Practice”, Manning Press, Dreamtech Press.
2. https://www.tutorialspoint.com/hadoop/hadoop_command_reference.htm
3. https://riptutorial.com/hadoop/example/13394/hadoop-v1-commands