Table of Contents

Assignment 2 - HDFS and Map Reduce

Objective: Transfer a file to HDP and run few HDFS commands

Create a test file which is more than 500 MB (for example zip a movie) then upload it to HDP then run the command mentioned in the lecture (HDFS -Part 2 Lecture - Slide 48 -53)

Firstly, I took a file which I converted into Zip it was kept in my local file which I uploaded to HDP (figure 1).

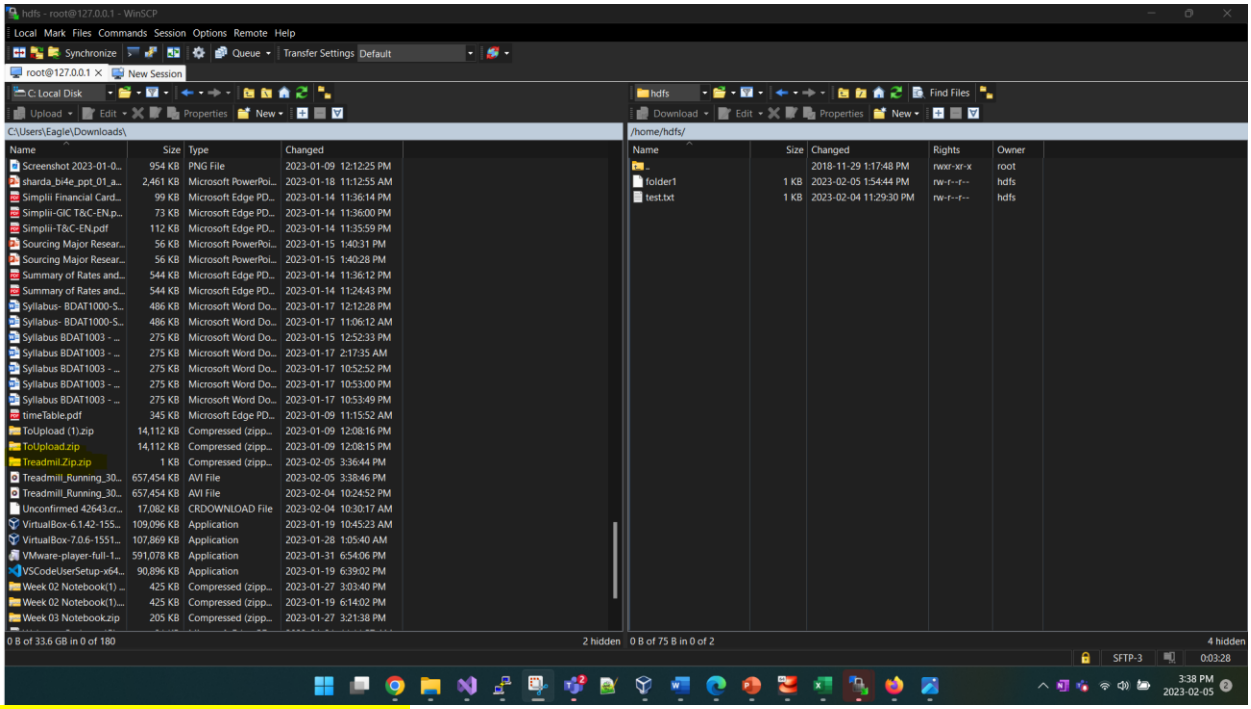


Figure 1Uploaded Zip file

Here, I made a directory of BDAT using ***hdfs dfs -mkdir {/name of directory}*** command after that I used ***ls command*** to check that I created a dictionary or not (figure 2).

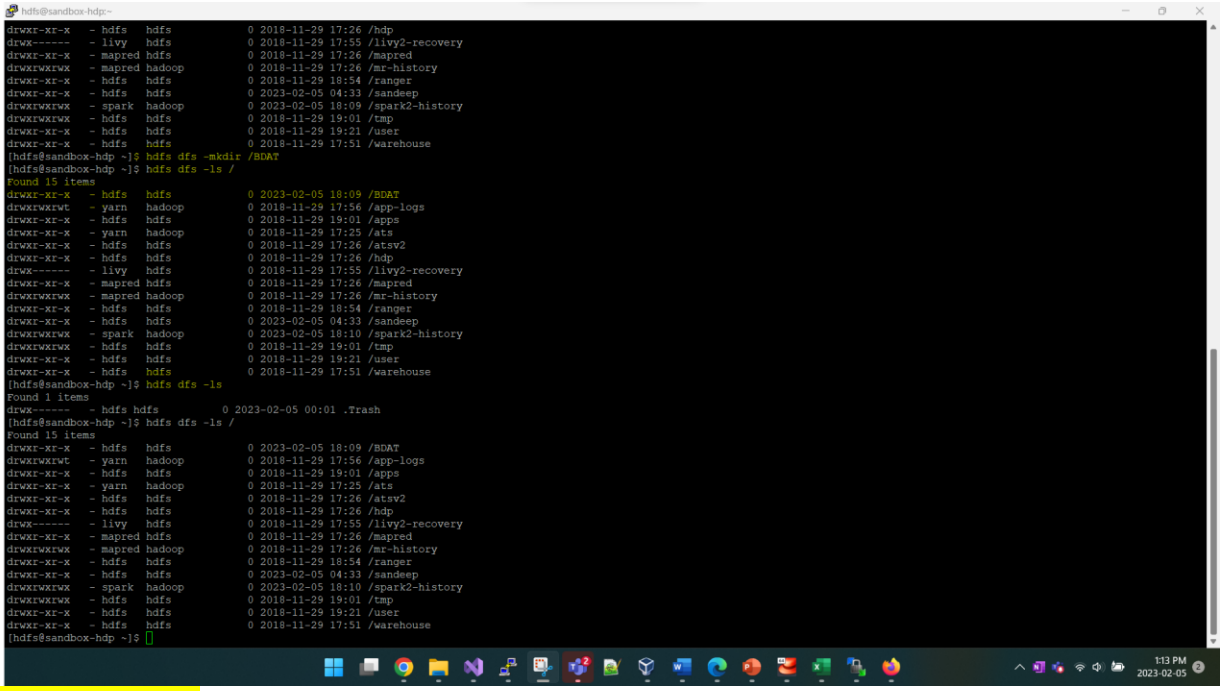


Figure 2 Made Dirc

In this third figure I ran the command ***hdfs dfs -copyFromLocal {File name } /BDAT{Dir}***
hdfs dfs -ls /BDAT for check the test file in the directory. (figure 3).

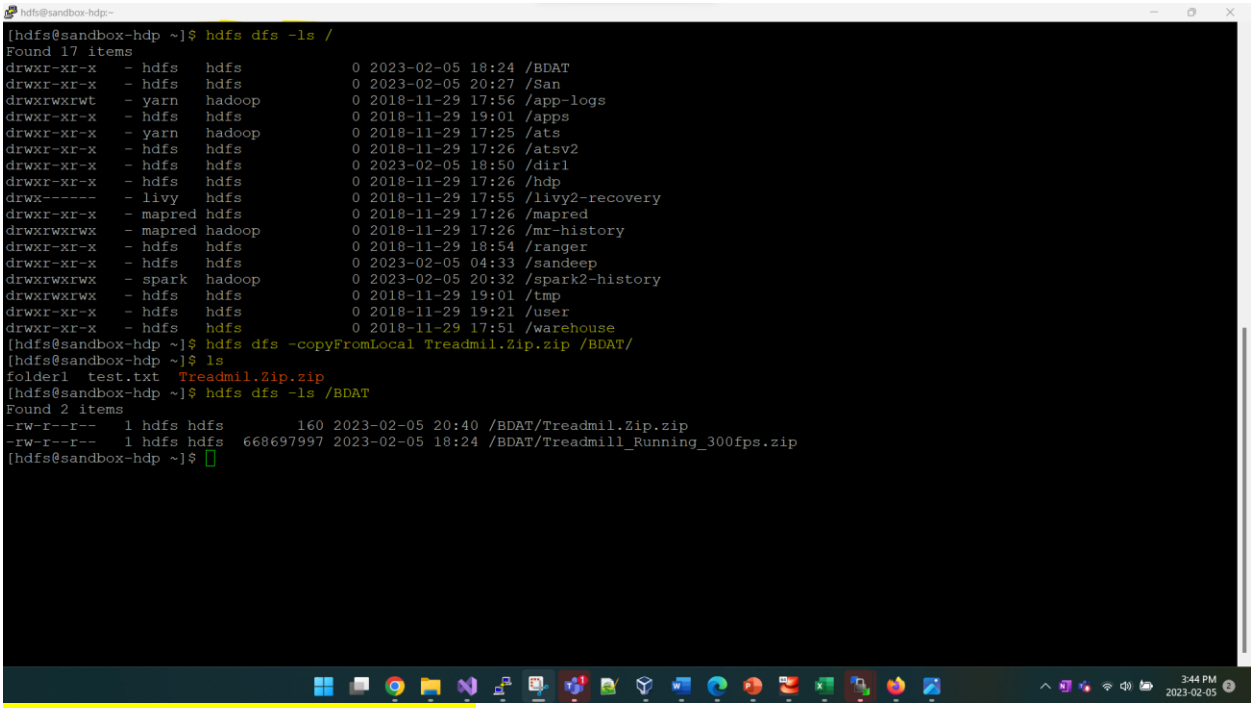


Figure 3 used copy from local

use the following link and explore the commands for fsck and dfs . Choose five commands from fsk and five commands from dfs. make sure you explain the command along with screen shots)

There are some five commands which I took from **dfs** such as **put, get, chmod, ls, cat**

in this figure I used chmod command to change the permission. ***Chmod 700 {file name}.***
My hello file was able to write and read after after changing the permsiion with dfs it can be executed (figure 4).

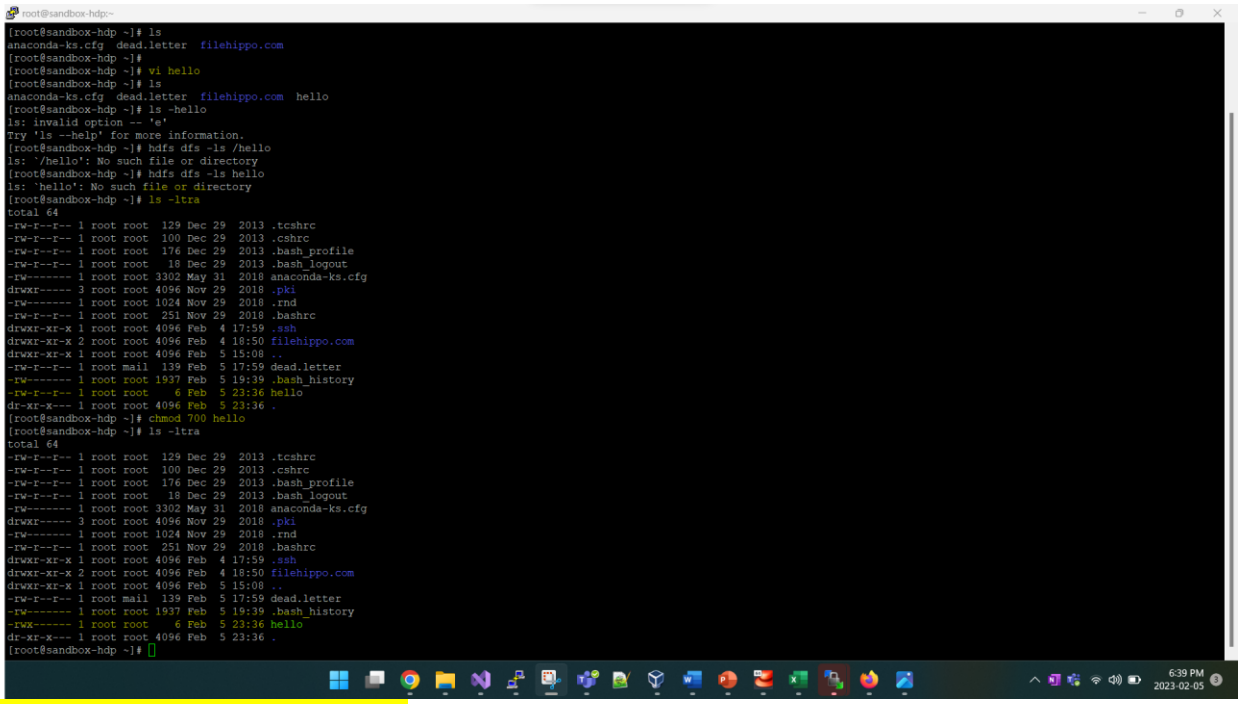


Figure 4 chmod command

I used **put command** to uploading file local system to **hdfs** system. I wanted to upload my folder 1 file into San directory so I gave **hdfs dfs -put {file name } /San**

San is a directory. Moreover I used get command which do same work (Figure 5)

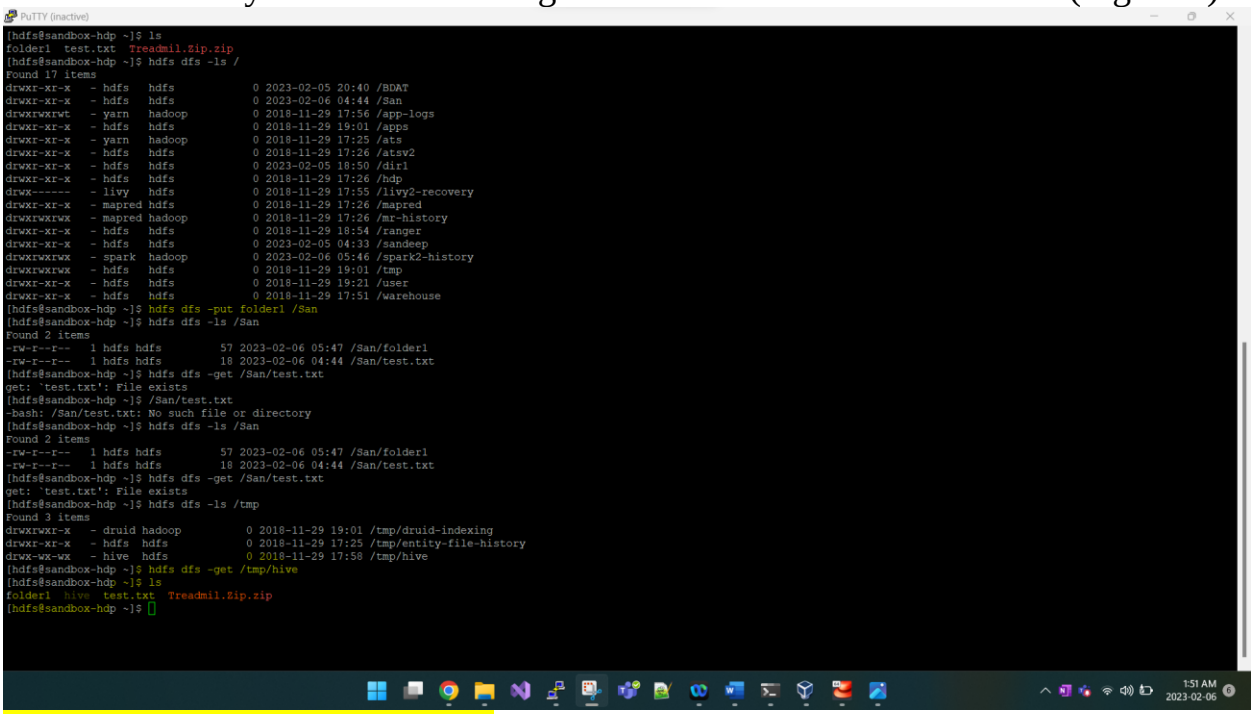
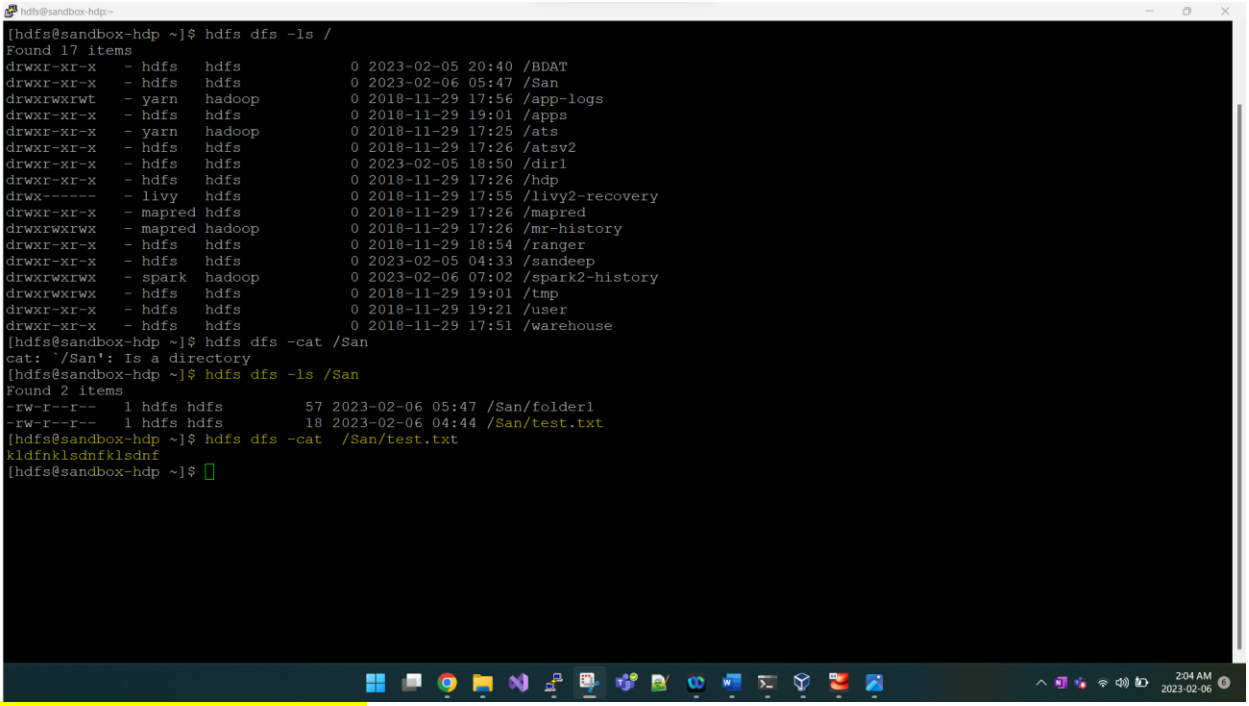


Figure 5 get and put command

I used ls command to check the content of San directory where I got 2 files. One in both of them I used cat command to check the file comment. For instance, **hdfs dfs -cat {file path and name}** (figure 6).



```
hdfs@sandbox-hdp:~
[hdfs@sandbox-hdp ~]$ hdfs dfs -ls /
Found 17 items
drwxr-xr-x   - hdfs   hdfs            0 2023-02-05 20:40 /BDAT
drwxr-xr-x   - hdfs   hdfs            0 2023-02-06 05:47 /San
drwxrwxrwt   - yarn   hadoop          0 2018-11-29 17:56 /app-logs
drwxr-xr-x   - hdfs   hdfs            0 2018-11-29 19:01 /apps
drwxr-xr-x   - yarn   hadoop          0 2018-11-29 17:25 /ats
drwxr-xr-x   - hdfs   hdfs            0 2018-11-29 17:26 /atsv2
drwxr-xr-x   - hdfs   hdfs            0 2023-02-05 18:50 /dir1
drwxr-xr-x   - hdfs   hdfs            0 2018-11-29 17:26 /hdp
drwx------   - livy   hdfs            0 2018-11-29 17:55 /livy2-recovery
drwxr-xr-x   - mapred hdfs            0 2018-11-29 17:26 /mapred
drwxrwxrwx   - mapred hadoop          0 2018-11-29 17:26 /mr-history
drwxr-xr-x   - hdfs   hdfs            0 2018-11-29 18:54 /ranger
drwxr-xr-x   - hdfs   hdfs            0 2023-02-05 04:33 /sandeep
drwxrwxrwx   - spark  hadoop          0 2023-02-06 07:02 /spark2-history
drwxrwxrwx   - hdfs   hdfs            0 2018-11-29 19:01 /tmp
drwxr-xr-x   - hdfs   hdfs            0 2018-11-29 19:21 /user
drwxr-xr-x   - hdfs   hdfs            0 2018-11-29 17:51 /warehouse
[hdfs@sandbox-hdp ~]$ hdfs dfs -cat /San
cat: `/San': Is a directory
[hdfs@sandbox-hdp ~]$ hdfs dfs -ls /San
Found 2 items
-rw-r--r--   1 hdfs hdfs         57 2023-02-06 05:47 /San/folder1
-rw-r--r--   1 hdfs hdfs         18 2023-02-06 04:44 /San/test.txt
[hdfs@sandbox-hdp ~]$ hdfs dfs -cat  /San/test.txt
kldfnklsdnfklsdnf
[hdfs@sandbox-hdp ~]$
```

Figure 6 Cat command

There are five commands which I took from **fsck** such as **location, delete, files, racks, blocks**. Here I am going to describe all commands below:

I used **fsck files command** which tell us all files information in figure 7 a)
The full command is hdfs fsck /San -files. Here San is a directory.

```
hdfs@sandbox-hdp:~
login as: root
root@127.0.0.1's password:
Last login: Mon Feb  6 21:33:09 2023 from 172.18.0.3
[root@sandbox-hdp ~]# sudo -iu hdfs
[hdfs@sandbox-hdp ~]$ hdfs fsck /San -files
Connecting to namenode via http://sandbox-hdp.hortonworks.com:50070/fsck?ugi=hdf
s&files=1&path=%2FSan
FSCK started by hdfs (auth:SIMPLE) from /172.18.0.2 for path /San at Mon Feb 06
22:27:42 UTC 2023
/San <dir>
/San/folder1 57 bytes, replicated: replication=1, 1 block(s):  OK
/San/test.txt 18 bytes, replicated: replication=1, 1 block(s):  OK

Status: HEALTHY
 Number of data-nodes:  1
 Number of racks:               1
 Total dirs:                    1
 Total symlinks:                0

Replicated Blocks:
 Total size:    75 B
 Total files:   2
 Total blocks (validated):      2 (avg. block size 37 B)
 Minimally replicated blocks:   2 (100.0 %)
 Over-replicated blocks:        0 (0.0 %)
 Under-replicated blocks:       0 (0.0 %)
 Mis-replicated blocks:         0 (0.0 %)
 Default replication factor:    1
 Average block replication:     1.0
 Missing blocks:                0
 Corrupt blocks:                0
 Missing replicas:              0 (0.0 %)

Erasure Coded Block Groups:
 Total size:    0 B
 Total files:   0
 Total block groups (validated):        0
 Minimally erasure-coded block groups:  0
 Over-erasure-coded block groups:       0
 Under-erasure-coded block groups:      0
 Unsatisfactory placement block groups: 0
 Average block group size:      0.0
 Missing block groups:          0
 Corrupt block groups:          0
 Missing internal blocks:       0
FSCK ended at Mon Feb 06 22:27:42 UTC 2023 in 2 milliseconds


The filesystem under path '/San' is HEALTHY
```

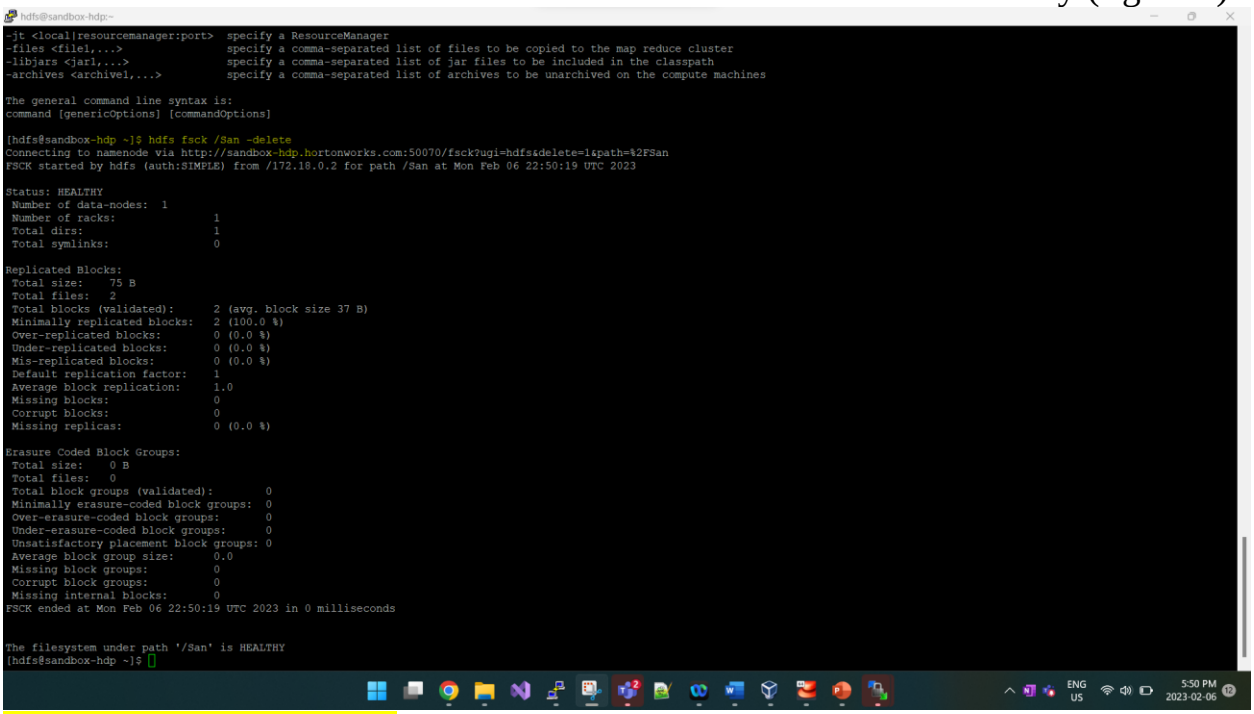
Figure 7(a) files command

I used block commands from **fsck** and checked **blocks** id in (figure 7)

```
hdfs@sandbox-hdp:~
The filesystem under path '/San' is HEALTHY
[hdfs@sandbox-hdp ~]$ hdfs fsck /San -files -blocks
Connecting to namenode via http://sandbox-hdp.hortonworks.com:50070/fsck?ugi=hdfs&files=1&blocks=1&path=%2FSan
FSCK started by hdfs (auth:SIMPLE) from /172.18.0.2 for path /San at Mon Feb 06 22:33:13 UTC 2023
/San <dir>
/San/folder1 57 bytes, replicated: replication=1, 1 block(s):  OK
0. BP-1419118625-172.17.0.2-1543512323726:blk_1073806549_65740 len=57 Live_repl=1

/San/test.txt 18 bytes, replicated: replication=1, 1 block(s):  OK
0. BP-1419118625-172.17.0.2-1543512323726:blk_1073804392_63579 len=18 Live_repl=1


Status: HEALTHY
 Number of data-nodes:  1
 Number of racks:               1
 Total dirs:                    1
 Total symlinks:                0

Replicated Blocks:
 Total size:    75 B
 Total files:   2
 Total blocks (validated):      2 (avg. block size 37 B)
 Minimally replicated blocks:   2 (100.0 %)
 Over-replicated blocks:        0 (0.0 %)
 Under-replicated blocks:       0 (0.0 %)
 Mis-replicated blocks:         0 (0.0 %)
 Default replication factor:    1
 Average block replication:     1.0
 Missing blocks:                0
 Corrupt blocks:                0
 Missing replicas:              0 (0.0 %)

Erasure Coded Block Groups:
 Total size:    0 B
 Total files:   0
 Total block groups (validated):        0
 Minimally erasure-coded block groups:  0
 Over-erasure-coded block groups:       0
 Under-erasure-coded block groups:      0
 Unsatisfactory placement block groups: 0
 Average block group size:      0.0
 Missing block groups:          0
 Corrupt block groups:          0
 Missing internal blocks:       0
FSCK ended at Mon Feb 06 22:33:13 UTC 2023 in 0 milliseconds


The filesystem under path '/San' is HEALTHY
[hdfs@sandbox-hdp ~]$
```

Figure 8 Block command

I used delete command from fsck to delete some files into San directory (figure 8)



```
hdfs@sandbox-hdp:~
-jt <local|resourcemanager:port>  specify a ResourceManager
-files <file1,...>                specify a comma-separated list of files to be copied to the map reduce cluster
-libjars <jar1,...>               specify a comma-separated list of jar files to be included in the classpath
-archives <archive1,...>          specify a comma-separated list of archives to be unarchived on the compute machines

The general command line syntax is:
command [genericOptions] [commandOptions]

[hdfs@sandbox-hdp ~]$ hdfs fsck /San -delete
Connecting to namenode via http://sandbox-hdp.hortonworks.com:50070/fsck?ugi=hdfs&delete=1&path=%2FSan
FSCK started by hdfs (auth:SIMPLE) from /172.18.0.2 for path /San at Mon Feb 06 22:50:19 UTC 2023

Status: HEALTHY
 Number of data-nodes:  1
 Number of racks:               1
 Total dirs:                    1
 Total symlinks:                0

Replicated Blocks:
 Total size:    75 B
 Total files:   2
 Total blocks (validated):      2 (avg. block size 37 B)
 Minimally replicated blocks:   2 (100.0 %)
 Over-replicated blocks:        0 (0.0 %)
 Under-replicated blocks:       0 (0.0 %)
 Mis-replicated blocks:         0 (0.0 %)
 Default replication factor:    1
 Average block replication:     1.0
 Missing blocks:                0
 Corrupt blocks:                0
 Missing replicas:              0 (0.0 %)

Erasure Coded Block Groups:
 Total size:    0 B
 Total files:   0
 Total block groups (validated):        0
 Minimally erasure-coded block groups:  0
 Over-erasure-coded block groups:       0
 Under-erasure-coded block groups:      0
 Unsatisfactory placement block groups: 0
 Average block group size:      0.0
 Missing block groups:          0
 Corrupt block groups:          0
 Missing internal blocks:       0
FSCK ended at Mon Feb 06 22:50:19 UTC 2023 in 0 milliseconds


The filesystem under path '/San' is HEALTHY
[hdfs@sandbox-hdp ~]$
```

Figure 9 delete command

In figure 9, I used **racks** and **location command** from **fsck** to check the location of the file and rack command is nothing, but it is in a location so just checked the **Datanodes**.

```
hdfs@sandbox-hdp:~
-D <property=value>                 define a value for a given property
-fs <file:///|hdfs://namenode:port> specify default filesystem URL to use, overrides 'fs.defaultFS' property from configurations.
-jt <local|resourcemanager:port>  specify a ResourceManager
-files <file1,...>                  specify a comma-separated list of files to be copied to the map reduce cluster
-libjars <jar1,...>                 specify a comma-separated list of jar files to be included in the classpath
-archives <archive1,...>            specify a comma-separated list of archives to be unarchived on the compute machines

The general command line syntax is:
command [genericOptions] [commandOptions]

[hdfs@sandbox-hdp ~]$ hdfs fsck /San -files -blocks -location -racks
fsck: Illegal option '-location'
Usage: hdfs fsck <path> [-list-corruptfileblocks | [-move | -delete | -openforwrite] [-files [-blocks [-locations | -racks | -replicaDetails | -upgradedomains]]]] [-includeSnapshots] [-show
progress] [-storagepolicies] [-maintenance] [-blockId <blk_Id>]
        <path>  start checking from this path
        -move   move corrupted files to /lost+found
        -delete delete corrupted files
        -files  print out files being checked
        -openforwrite   print out files opened for write
        -includeSnapshots       include snapshot data if the given path indicates a snapshottable directory or there are snapshottable directories under it
        -list-corruptfileblocks print out list of missing blocks and files they belong to
        -files -blocks  print out block report
        -files -blocks -locations       print out locations for every block
        -files -blocks -racks   print out network topology for data-node locations
        -files -blocks -replicaDetails  print out each replica details
        -files -blocks -upgradedomains  print out upgrade domains for every block
        -storagepolicies        print out storage policy summary for the blocks
        -maintenance    print out maintenance state node details
        -showprogress   show progress in output. Default is OFF (no progress)
        -blockId        print out which file this blockId belongs to, locations (nodes, racks) of this block, and other diagnostics info (under replicated, corrupted or not, etc)

Please Note:
        1. By default fsck ignores files opened for write, use -openforwrite to report such files. They are usually  tagged CORRUPT or HEALTHY depending on their block allocation status
        2. Option -includeSnapshots should not be used for comparing stats, should be used only for HEALTH check, as this may contain duplicates if the same file present in both original fs
 tree and inside snapshots.

Generic options supported are:
-conf <configuration file>          specify an application configuration file
-D <property=value>                 define a value for a given property
-fs <file:///|hdfs://namenode:port> specify default filesystem URL to use, overrides 'fs.defaultFS' property from configurations.
-jt <local|resourcemanager:port>  specify a ResourceManager
-files <file1,...>                  specify a comma-separated list of files to be copied to the map reduce cluster
-libjars <jar1,...>                 specify a comma-separated list of jar files to be included in the classpath
-archives <archive1,...>            specify a comma-separated list of archives to be unarchived on the compute machines

The general command line syntax is:
command [genericOptions] [commandOptions]

[hdfs@sandbox-hdp ~]$
```

Part 2 (A) Follow the slides (Week 7 – Hadoop Map Reduce Framework page 73-77). Copy salesjan2009 and jar file to your Hadoop cluster then run the following command….

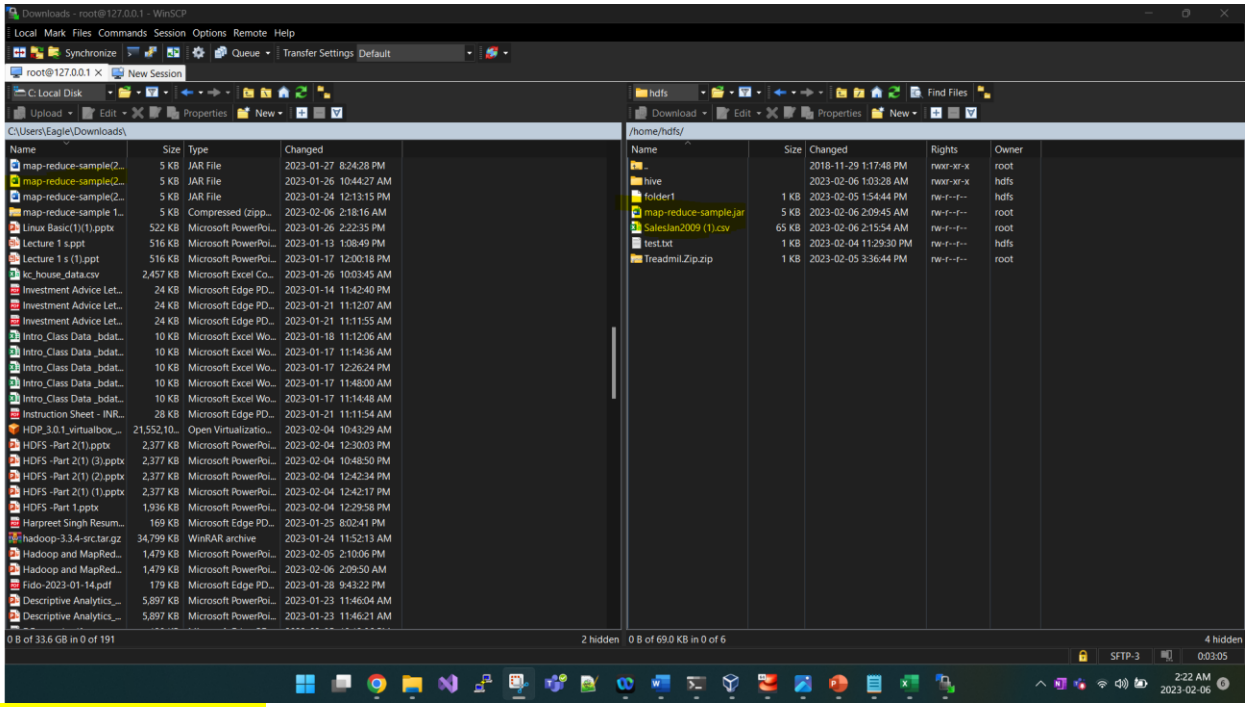


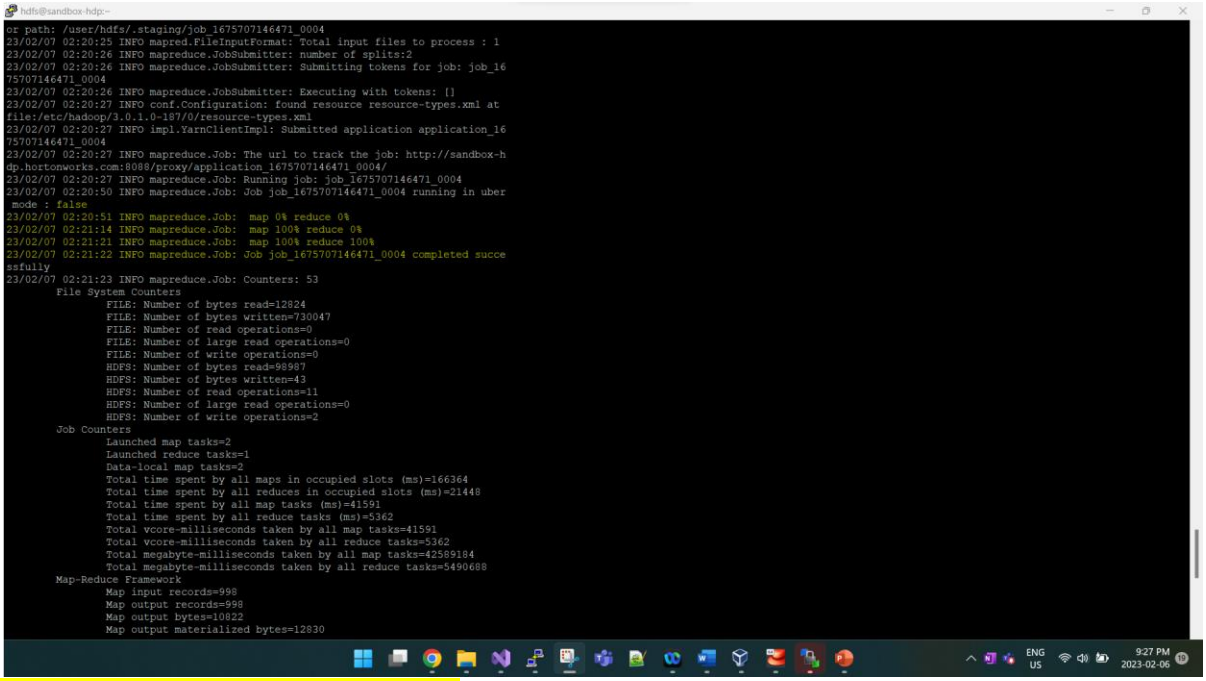
Figure 10 jar file

Firstly, I made a directory of abhishek then I transferred a jar file and map reduce. I used l -ls command after switching to hdfs user so then I used a command of which you told us to follow
It is **Hadoop jar map-reduce-sample.jar /abhishek /SalesJan2009.cav /abhishek /mp_output** then got the file info.

After getting all details I used to **hdfs dfs -ls /abhishek/** I got my files then used hdfs dfs -ls /abhishek/mp_output

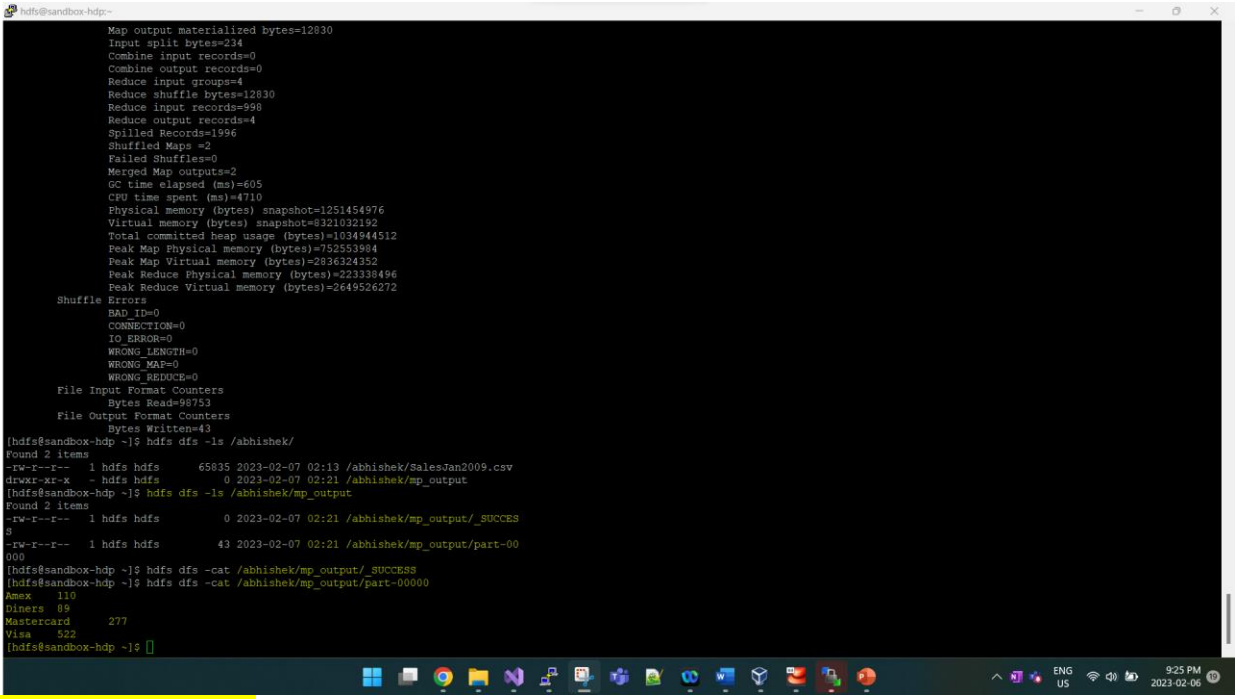
Used cat command to get the result **hdfs dfs -cat/abhishek/mp_output/_success hdfs dfs -cat/abhishek/mp_output/part-0000**

I got the result in figure 12.



```
hdfs@sandbox-hdp:~
or path: /user/hdfs/.staging/job_1675707146471_0004
23/02/07 02:20:25 INFO mapred.FileInputFormat: Total input files to process : 1
23/02/07 02:20:26 INFO mapreduce.JobSubmitter: number of splits:2
23/02/07 02:20:26 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_16
75707146471_0004
23/02/07 02:20:26 INFO mapreduce.JobSubmitter: Executing with tokens: []
23/02/07 02:20:27 INFO conf.Configuration: found resource resource-types.xml at
file:/etc/hadoop/3.0.1.0-187/0/resource-types.xml
23/02/07 02:20:27 INFO impl.YarnClientImpl: Submitted application application_16
75707146471_0004
23/02/07 02:20:27 INFO mapreduce.Job: The url to track the job: http://sandbox-h
dp.hortonworks.com:8088/proxy/application_1675707146471_0004/
23/02/07 02:20:27 INFO mapreduce.Job: Running job: job_1675707146471_0004
23/02/07 02:20:50 INFO mapreduce.Job: Job job_1675707146471_0004 running in uber
 mode : false
23/02/07 02:20:51 INFO mapreduce.Job:  map 0% reduce 0%
23/02/07 02:21:14 INFO mapreduce.Job:  map 100% reduce 0%
23/02/07 02:21:21 INFO mapreduce.Job:  map 100% reduce 100%
23/02/07 02:21:22 INFO mapreduce.Job: Job job_1675707146471_0004 completed succe
ssfully
23/02/07 02:21:23 INFO mapreduce.Job: Counters: 53
        File System Counters
                FILE: Number of bytes read=12824
                FILE: Number of bytes written=730047
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=98987
                HDFS: Number of bytes written=43
                HDFS: Number of read operations=11
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=2
                Launched reduce tasks=1
                Data-local map tasks=2
                Total time spent by all maps in occupied slots (ms)=166364
                Total time spent by all reduces in occupied slots (ms)=21448
                Total time spent by all map tasks (ms)=41591
                Total time spent by all reduce tasks (ms)=5362
                Total vcore-milliseconds taken by all map tasks=41591
                Total vcore-milliseconds taken by all reduce tasks=5362
                Total megabyte-milliseconds taken by all map tasks=42589184
                Total megabyte-milliseconds taken by all reduce tasks=5490688
        Map-Reduce Framework
                Map input records=998
                Map output records=998
                Map output bytes=10822
                Map output materialized bytes=12830
```

Figure 11 hadoop jar command



```
hdfs@sandbox-hdp:~
                Map output materialized bytes=12830
                Input split bytes=234
                Combine input records=0
                Combine output records=0
                Reduce input groups=4
                Reduce shuffle bytes=12830
                Reduce input records=998
                Reduce output records=4
                Spilled Records=1996
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=605
                CPU time spent (ms)=4710
                Physical memory (bytes) snapshot=1251454976
                Virtual memory (bytes) snapshot=8321032192
                Total committed heap usage (bytes)=1034944512
                Peak Map Physical memory (bytes)=752553984
                Peak Map Virtual memory (bytes)=2836324352
                Peak Reduce Physical memory (bytes)=223338496
                Peak Reduce Virtual memory (bytes)=2649526272
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=98753
        File Output Format Counters
                Bytes Written=43
[hdfs@sandbox-hdp ~]$ hdfs dfs -ls /abhishek/
Found 2 items
-rw-r--r--   1 hdfs hdfs      65835 2023-02-07 02:13 /abhishek/SalesJan2009.csv
drwxr-xr-x   - hdfs hdfs          0 2023-02-07 02:21 /abhishek/mp_output
[hdfs@sandbox-hdp ~]$ hdfs dfs -ls /abhishek/mp_output
Found 2 items
-rw-r--r--   1 hdfs hdfs          0 2023-02-07 02:21 /abhishek/mp_output/_SUCCES
S
-rw-r--r--   1 hdfs hdfs         43 2023-02-07 02:21 /abhishek/mp_output/part-00
000
[hdfs@sandbox-hdp ~]$ hdfs dfs -cat /abhishek/mp_output/_SUCCESS
[hdfs@sandbox-hdp ~]$ hdfs dfs -cat /abhishek/mp_output/part-00000
Amex    110
Diners  89
Mastercard      277
Visa    522
[hdfs@sandbox-hdp ~]$
```

Figure 12Output