

## Basic Hadoop Commands :-

1) `hadoop fs -mkdir 20761A0585`

put : copyFromLoc

2) `hadoop fs -ls`

get : copyToLocal

Found 1 item

`drwxr-xr-x - cloudera cloudera 0 2023-01-06 01:03 20761A0585`

gedit :- It is used to create a file

→ `gedit fi.txt`

→ It created a file called fi.

→ copyFromLocal command can be used instead of put command. But since it is case sensitive most use put command

→ To view the fi.txt use

`cat fi.txt`

→ To copy the fi file to 20761A0585 directory use

`hadoop fs -put fi.txt 20761A0585`

→ To check whether fi is copied to 20761A0585 directory use

`hadoop fs -cat 20761A0585/fi.txt`

## Commands In Hadoop :-

1) # Print the Hadoop Version

`hadoop version`

O/P: Hadoop 2.6.0

2) List the contents of the root directory in HDFS

`hadoop fs -ls`

O/P: `drwxr-xr-x - cloudera cloudera 0 2023-01-06 01:20 20761A0585`

3) Report the amount of space used and available on currently mounted filesystems

hadoop fs - df hdfs://

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

hadoop fs -count hdfs://

```
3) O/P:- Filesystem      Size   Used   Available   Use %
= 
hdfs://quickstart     5853152052  872656692  45347261350   1 %
cloudera:8020
```

4) O/P: 81 937 861286761 hdfs://

5) Run a DFS filesystem checking utility

hadoop fsck -l

O/P :-

6) Run a cluster balancing utility

## hadoop balancer

23/01/19 18:55:49 INFO balancer.Balancer: namenodes  
= [hd:fs://0.0.0.0:8022]

Jan 19, 2023 6:55:52 PM Balancing took 3.585 sec

7) Create a new directory named "hadoop" below the /user/training directory in HDFS. Since you're currently logged in with the "training" user ID, /user/training is your home directory in HDFS.

hadoop fs -mkdir /user/training/hadoop

User: hadoop fs -mkdir hadoop

- 1) add a sample text file from the local directory named "data" to the new directory you created in HDFS during the previous step.

[hadoop fs -put data/sample.txt /user/training/hadoop]

- 2) list the contents of this new directory in HDFS

[hadoop fs -ls /user/training/hadoop]

- 3) O/P: [user@host ~] \$ hadoop fs -ls /user/training/hadoop  
10) Add the entire local directory called "retail" to /user/training directory in HDFS

[hadoop fs -put data/retail /user/training/hadoop]

- 4) To check the file exists or not

[hadoop fs -ls]

- 12) See how much space this directory occupies in HDFS

[hadoop fs -du -s -h hadoop/retail]

- 13) Delete a file 'customers' from the 'retail' directory

[hadoop fs -rm hadoop/retail/customers]

- 14) Ensure this file is no longer in HDFS

[hadoop fs -ls hadoop/retail/customers]

O/P: No such file or directory

15) Delete all files from the "retail" directory using a wildcat

```
[hadoop fs - rm hadoop/retail/*]
```

16) To empty the trash

```
[hadoop fs - expunge]
```

17) finally, remove the entire retail directory and all of its contents in HDFS

```
[hadoop fs - rm -r hadoop/retail]
```

O/P: Deleted hadoop filename

18) list the hadoop directory Again

```
[hadoop fs - ls hadoop]
```

O/P: No such file or directory.

19) Add the purchases.txt file from the local directory named "/home/training/" to the hadoop directory you created

O/P:  
[hadoop fs - copyFromLocal /home/training/  
purchases.txt hadoop]

(or)

```
[hadoop fs - put f2.txt hadoop]
```

20) To view f2.txt use

```
Cat f1.txt
```

To view f2.txt in hadoop1 directory

```
[hadoop fs - cat hadoop1/f2.txt]
```

21) Add purchase.txt file to local directory named /home/training/ to the hadoop directory you created

hadoop fs -copyToLocal /home/training/  
purchase.txt hadoop

22) cp is used to copy the files between directories.

hadoop fs -cp /user/training/\*.txt /user/  
training hadoop

23) -get command can be used alternatively '  
-copyToLocal'.

hadoop fs -get hadoop /sample.txt /home/  
training /

24) Display last kilobyte of the file "purchase"  
to ~~stdout~~ stdout.

hadoop fs -tail hadoop /purchases.txt

o/p: Lakireddy Balireddy college of engineering

25) Default file permissions are 666 in HDFS  
use '-chmod' command to change permissions  
of a file

hadoop fs -ls hadoop /purchases.txt

Sudo -u hadoop fs -chmod 600 hadoop/  
purchases.txt

26) Default names of owner and group are training, training.

Use '-chown' to change over name and group name simultaneously

```
hadoop fs -ls hadoop/purchases.txt
```

```
sudo -u hdfs hadoop fs -chown root:root  
hadoop/purchases.txt
```

27) Default name of group is training

# use '-chgrp' command to change groupname

```
hadoop fs -ls hadoop/purchases.txt
```

```
sudo -u hdfs hadoop fs -chgrp training  
hadoop/purchases.txt
```

28) Move a directory from one location to other

```
hadoop fs -mv hadoop apache-hadoop
```

29) Default replication factor to a file is 3.

Use '-setrep' command to change replication factor of a file.

```
hadoop fs -setrep -w 2 apache-hadoop/sample.txt
```

30) Copy a directory from one mode in the cluster to another. Use '-distcp' command to copy,

- overwrite option to overwrite in an existing files.

- update command to synchronize both directories

hadoop fs - distcp hdfs://namenode A /apache  
hadoop hdfs://namenode

31) Command to make the name node leave safe mode

hadoop fs - expunge

Sudo -u hdfs hadfs dfsadmin - safemode leave

27/01/2023

## MapReducing Technique : WordCount:

→ Open Eclipse



Delete the WordCount which is previously available



① Now goto "File" → "New" → "Java Project"



Enter name as WordCount press "Next"



Click on "Add External Jars"



filesystem → usr → lib - hadoop



Select All the Jars in hadoop and click on OK.



Again click on "Add External Jars" and goto Client and add all Jar's . click OK.



Finish

- ② Now WordCount file is created. Dropdown the WordCount. Click src [Right click]

↓  
New

↓

Class

↓

name : WordCount

↓  
Finish

- ③ In the VM open Browser and type  
[www.apache.org](http://www.apache.org) WordCount

↓

click "apache 3.3.4 hadoop tutorial"

↓

Copy the Source Code

and paste it in the WordCount file  
which you have created.

- ④ To Run the WordCount file :

Right Click WordCount.java

↓

Exports

↓

Dropdown Java and Click on JAR file

↓

Select the path by clicking the Browse

↓

Finish

- ⑤ Now go to Terminal

↓

ls

↓

pwd (to know your present working directory)

↓  
cat > /home/cloudera/input.txt

↓  
write some content and to end click  
ctrl+z.

↓  
cat /home/cloudera/input.txt

↓  
ls (check input.txt)

↓  
hdfs dfs -ls /

↓  
hdfs dfs -mkdir /585

↓  
hdfs dfs -ls /

↓  
hdfs dfs -mkdir

↓  
hdfs dfs -put /home/cloudera/input.txt /

↓  
hdfs dfs -cat /585/input.txt

(input.txt content will be displayed)

↓

hadoop jar /home/cloudera/wordCount.jar wordCount  
- /585/input.txt /out3

↓

To check if your code is successfully Run or not

↓  
hdfs dfs -ls /out3

O/P:-

-rw-r--r-- 1 cloudera supergroup 0 2023-01-27 01:08 /out3/  
-rw-r--r-- 1 cloudera supergroup 24 2023-01-27 01:08 /  
out3/part-r-00000  
↓  
copy this

hdfs dfs -cat /out3/part-r-00000  
part

O/P:- bat 4

cat 4

mat 4

rat 4

## Matrix Program:-

03-02-2023

172.16.2.72:9999

- Open the Above link and download the New file.
- Copy the Code to a Notepad.
- Follow the Same steps As the WordCount Program.

→ Instead of WordCount write Matrix.

→ Step 1:- Add all the external jar files

Step 2:- Create a new class named Matrix

Step 3:- Copy the Code

Step 4:- Create a jar file.

Step 5:- Open terminal

↓

ls

↓

cat > /home/cloudera/input.txt

O/P:- M,0,0,1 N,0,0,1  
M,0,1,0 N,0,1,1  
M,1,0,0 N,1,0,1  
M,1,1,1 N,1,1,1

aint

1

↓

cat /home/cloudera/input1.txt

↓

hadoop fs -ls /585

↓

hdfs dfs -put /home/cloudera/input1.txt /585

↓

hadoop fs -ls

O/P

-rwxr-xr-x -cloudera cloudera 0 2023-01-20 01:56 1585

-rwxr-xr-x -cloudera cloudera 0 2023-01-20 01:55 /out4

↓

hadoop jar /home/cloudera/MatrixFactorization/585/

input1.txt /out4

↓

hdfs dfs -ls /out4

↓

dfs-cat /out4/part-r-00000

4

O/P

0,0,1.0

0,1,1.0

1,0,1.0

1,1,1.0

17/2/23

- **Sudo -i** : to move to root directory  
→ **Sudo -u hdfs** ~~dfsadmin -sa-fermode leave~~  
O/P: Safe mode is OFF

1585  
→ In Mysql (after doing mysql -uroot -pcloudera)  
to go to the database (retail-db) use,

**use retail-db**

→ Show tables;

→ **Select \* from departments;**

→ Sqoop Import --connect "jdbc:mysql://quickstart:  
3306/retail-db" --osname system name  
--table departments --target-dir  
/user/cloudera/dep1;

↓  
To move database to hdfs, use this command

3/3/2023

1) Open 2 terminals one for mysql and another  
for quickstart  
→ In mysql terminal [Import Mysql to HDFS]

mysql -u root -P

Pwd: cloudera

O/p: Welcome to MySQL Monitor.

→ Show databases;

Database
information-security
cm
nav
navms
test

→ Create database demo;

O/p: Query OK, 1 row affected

→ Use demo;

Database changed

→ CREATE TABLE Employee (Id int Not NULL, Name  
VARCHAR(20), DOB Date, Gender Char(1), Salary int,  
PRIMARY KEY (Id));

O/p: Query OK, 0 rows affected.

→ Insert into Employee values ((1, 'Alex', '1987-12-12',  
10000), (2, 'Julia', '1987-11-11', 'F', 20000), (3, 'John'  
'1987-10-10', 'M', 25000), (4, 'Khalishi', '1987-9-9',  
'F', 30000));

O/p: Query OK, 4 rows affected.

→ In quickstart Terminal

Sqoop import --connect jdbc:mysql://localhost/demo --username root --password cloudera --table Employee

→ hdfs dfs -cat Employee/\*

Output  
1, Alex, 1987-12-12, M, 10000  
2, Julia, 1987-11-11, F, 20000  
3, John, 1987-10-10, M, 25000  
4, Khalishi, 1987-09-09, F, 30000

→ In mysql terminal

insert into Employee values (5, 'Jonsnow', '1987-08-09', 'M', 35000);

insert into Employee values (6, 'Arya', '1993-07-07', 'F', 40000);

Output: Query OK 2 rows affected

→ In quickstart Terminal

Sqoop import --connect jdbc:mysql://localhost/demo --username root --password cloudera --table Employee --incremental append --check-column id --last-value 4

→ hdfs dfs -cat Employee/\*

Output  
1, Alex, 1987-12-12, M, 10000  
2, Julia, 1987-11-11, F, 20000  
3, John, 1987-10-10, M, 25000  
4, Khalishi, 1987-09-09, F, 30000  
5, Jonsnow, 1987-08-09, M, 35000  
6, Arya, 1993-07-07, F, 40000

→ In MySQL terminal

CREATE TABLE EmployeeNew (Id int NOT NULL, Name  
VARCHAR(20), DOB DATE, Gender char(1), Salary int,  
PRIMARY KEY (Id));

[Export HDFS to MySQL]

O/P: Query OK, 0 rows affected

→ In quickstart terminal

sqoop export --connect jdbc:mysql://localhost/  
demo --username root --password cloudera --export  
--dir /user/cloudera/employee --update-mode allowinsert  
--table EmployeeNew

→ In MySQL terminal

Select \* from EmployeeNew;

O/P	Id	Name	DOB	Gender	Salary
	1	Alex	1987-12-12	M	10000
	2	Julia	1987-11-11	F	20000
	3	John	1987-10-10	M	25000
	4	Khalishi	1987-09-09	F	30000
	5	Jonsnow	1987-08-09	M	35000
	6	Arya	1993-07-07	F	40000

17/3/2023

- Open 2 terminals One for 'hive' and another for quickstart.
- In quickstart terminal  
Create a txt file

cat > employee.txt

1201, Gopal, 45000, Technical Manager  
1202, Manisha, 45000, Proof reader  
1203, Kiran, 40000, Technical writer  
1024, Masthan, 40000, Teacher

- Open hive terminal

→ hive

→ O/P:- Logging initialized using configuration...

→ Create database Company1;

O/P:- Query OK, 1 row affected

→ Use company1

Database changed

→ hive> Create Table employee1 (eid int, name string, salary string, des string) > row format delimited fields terminated by ',';

O/P:- OK

→ LOAD DATA LOCAL INPATH '/home/cloudera / employee.txt' OVERWRITE INTO TABLE employee1;

O/P:- OK

→ Select \* from employee1;

1201	Gopal	45000	Technical Manager	
1202	Manisha	45000	Proof Reader	
1203	Kiran	40000	Technical writer	
1204	Masthan	40000	Teacher	

→ ALTER TABLE employee1 RENAME TO emp;

→ Show tables;

O/p:- emp  
employee

→ ALTER TABLE emp CHANGE name ename string;

→ Select \* from emp;

O/P:

1201	Gopal	45000	Technical manager
1202	Manisha	45000	Proof reader
1203	Kiran	40000	Technical writer
1204	Martha	40000	Teacher

→ desc emp;

O/P:

eid	int
ename	string
Salary	string
des	string

→ ALTER TABLE emp ADD COLUMN dept STRING COMMENT 'Department name';

→ desc emp;

O/P:

eid	int
ename	string
Salary	string
des	string
dept	string

→ drop table emp;

O/P: OK

→ CREATE TABLE employee1 (eid int, name string, Salary string, des string) row format delimited  
terminated by ',';

O/P: OK

→ LOAD DATA LOCAL INPATH '/home/cloudera/employee.txt'  
OVERWRITE INTO TABLE employee1;

O/P:- OK

→ Select \* from employee1;

O/P: 1201 Gopal 45000 Technical Manager  
1202 Manisha 45000 Proof reader.

→ SELECT \* FROM employee1 WHERE salary > 40000;

O/P: OK  
1201 Gopal 45000 Technical Manager  
1202 Manisha 45000 Proof reader.