Hbase Shell Commands

...

IMPORTANT

Copyright Infringement and Illegal Content Sharing Notice

Hbase

HDFS along with MapReduce follows the write once and read many times paradigm.

In other words, data in HDFS is written once, but it can be read an unlimited number of times.

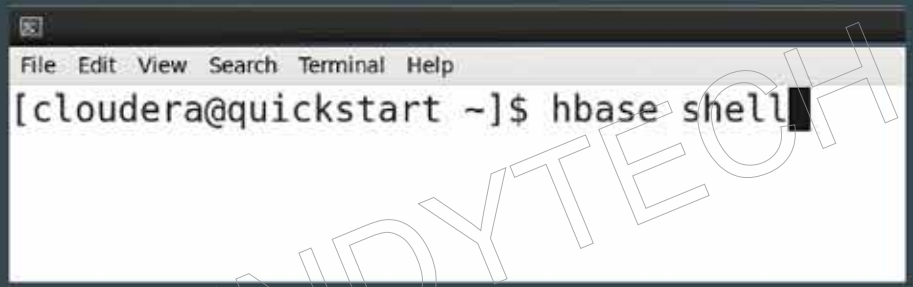
There is no provision for updating an existing data set or record in HDFS.

Even though HBase uses HDFS to store data, it supports update operations by maintaining multiple versions of the same data points.

Hbase Commands

Connect to hbase:

hbase shell



Hbase Commands

See list of tables in hbase

list

```
[cloudera@quickstart ~]$ hbase shell
2020-05-22 01:24:16,586 INFO  [main] Configuration.deprecation: hadoop.n
e.lib.available
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.2.0-cdh5.13.0, rUnknown, Wed Oct  4 11:16:18 PDT 2017

hbase(main):001:0> list
TABLE
0 row(s) in 0.1820 seconds

=> []
hbase(main):002:0>
```

Hbase Commands

Did you get an error, something like below

```
hbase(main):002:0> list
TABLE

ERROR: org.apache.hadoop.hbase.PleaseHoldException: Master is initializing
        at org.apache.hadoop.hbase.master.HMaster.checkInitialized(HMaster.java:23
        at org.apache.hadoop.hbase.master.MasterRpcServices.getTableNames(MasterRp
        at org.apache.hadoop.hbase.protobuf.generated.MasterProtos$MasterService$2
(MasterProtos.java:55650)
        at org.apache.hadoop.hbase.ipc.RpcServer.call(RpcServer.java:2191)
        at org.apache.hadoop.hbase.ipc.CallRunner.run(CallRunner.java:112)
        at org.apache.hadoop.hbase.ipc.RpcExecutor$Handler.run(RpcExecutor.java:18
        at org.apache.hadoop.hbase.ipc.RpcExecutor$Handler.run(RpcExecutor.java:16

List all tables in hbase. Optional regular expression parameter could
be used to filter the output. Examples:

  hbase> list
  hbase> list 'abc.*'
  hbase> list 'ns:abc.*'
  hbase> list 'ns:.*'


hbase(main):003:0>
```

Hbase Commands

Your hbase master and region server might be stopped.
Let's check that.

In terminal type the below command to see status of all services

```
[cloudera@quickstart ~]$ sudo service --status-all
```

```
HBase master daemon is running
hbase-regionserver is not running.
```

Note: Just check for hbase master and region server. We do not have to worry about other services

Hbase Commands

If any of these 2 is stopped we need to start it again. Else we do not have to do this.

sudo service hbase-master restart
sudo service hbase-regionserver restart

```
[cloudera@quickstart ~]$ sudo service hbase-master restart
stopping master.
Stopped HBase master daemon:                               [  OK  ]
starting master, logging to /var/log/hbase/hbase-hbase-master-quickstart.clouder
Started HBase master daemon (hbase-master):                [  OK  ]
[cloudera@quickstart ~]$ sudo service hbase-regionserver restart
Restarting Hadoop HBase regionserver daemon: Stopping Hadoop HBase regionserver
er to stop because no pid file /var/run/hbase/hbase-hbase-regionserver.pid
hbase-regionserver.
Starting Hadoop HBase regionserver daemon: starting regionserver, logging to /va
se-regionserver-quickstart.cloudera.out
hbase-regionserver.
[cloudera@quickstart ~]$
```

Hbase Commands

Exit from hbase and connect back

```
hbase(main):003:0> exit
```

```
[cloudera@quickstart ~]$ hbase shell
2020-05-22 01:24:16,586 INFO  [main] Configuration.deprecation: hadoop.n
e.lib.available
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.2.0-cdh5.13.0, rUnknown, Wed Oct  4 11:16:18 PDT 2017

hbase(main):001:0> list
TABLE
0 row(s) in 0.1820 seconds

=> []
hbase(main):002:0>
```

Hbase Commands

Create a table named students with 3 column families.

create 'students', 'personal_details', 'contact_details', 'marks'

```
hbase(main):001:0> create 'students', 'personal_details', 'contact_details', 'marks'
0 row(s) in 1.4170 seconds

=> Hbase::Table - students
hbase(main):002:0>
```

```
hbase(main):015:0> list
TABLE
students
1 row(s) in 0.0070 seconds

=> ["students"]
hbase(main):016:0>
```

Hbase Commands

Adding records using put

put 'students', 'student1', 'personal_details:name', 'Sumit'

put 'students', 'student1', 'personal_details:email', 'Sumit@gmail.com'

```
hbase(main):001:0> put 'students', 'student1', 'personal_details:name', 'Sumit'
0 row(s) in 0.2580 seconds

hbase(main):002:0> put 'students', 'student1', 'personal_details:email', 'Sumit@gmail.com'
0 row(s) in 0.0040 seconds

hbase(main):003:0>
```

Hbase Commands

See all records using scan

scan 'students'

```
hbase(main):004:0> scan 'students'
ROW                     COLUMN+CELL
 student1               column=personal_details:email, timestamp=1590137243652, value=Sumit@gmail.
                        com
 student1               column=personal_details:name, timestamp=1590137203018, value=Sumit
1 row(s) in 0.0120 seconds
```

Hbase Commands

Get record with row key student1

get ‘students’, ‘student1’

```
hbase(main):008:0> get 'students', 'student1'
COLUMN                          CELL
 personal_details:email         timestamp=1590137243652, value=Sumit@gmail.com
 personal_details:name          timestamp=1590137203018, value=Sumit
2 row(s) in 0.0170 seconds
```

Hbase Commands

Just display personal details for student 1

get ‘students’, ‘student1’, {column => ‘personal_details’}

```
hbase(main):009:0> get 'students', 'student1', {COLUMN => 'personal_details'}
COLUMN                          CELL
 personal_details:email         timestamp=1590137243652, value=Sumit@gmail.com
 personal_details:name          timestamp=1590137203018, value=Sumit
2 row(s) in 0.0060 seconds
```

Hbase Commands

Just display name of student1

get ‘students’, ‘student1’, {column => ‘personal_details:name’}

```
hbase(main):010:0> get 'students', 'student1', {COLUMN => 'personal_details:name'}
COLUMN                          CELL
 personal_details:name          timestamp=1590137203018, value=Sumit
1 row(s) in 0.0060 seconds
```

Hbase Commands

Delete email id column for student1

delete ‘students’, ‘student1’, ‘personal_details:email’

```
hbase(main):011:0> delete 'students', 'student1', 'personal_details:email'
0 row(s) in 0.0260 seconds
```

```
hbase(main):012:0> scan 'students'
ROW                    COLUMN+CELL
 student1              column=personal_details:name, timestamp=1590137203018, value=Sumit
1 row(s) in 0.0110 seconds
```

Hbase Commands

Describe a table

describe ‘students’

```
hbase(main):013:0> describe 'students'
Table students is ENABLED
students
COLUMN FAMILIES DESCRIPTION
{NAME => 'contact_details', DATA_BLOCK_ENCODING => 'NONE', BLOOMFILTER => 'ROW', REPLICATION_SCOPE =>
 '0', VERSIONS => '1', COMPRESSION => 'NONE', MIN_VERSIONS => '0', TTL => 'FOREVER', KEEP_DELETED_CEL
LS => 'FALSE', BLOCKSIZE => '65536', IN_MEMORY => 'false', BLOCKCACHE => 'true'}
{NAME => 'marks', DATA_BLOCK_ENCODING => 'NONE', BLOOMFILTER => 'ROW', REPLICATION_SCOPE => '0', VERS
IONS => '1', COMPRESSION => 'NONE', MIN_VERSIONS => '0', TTL => 'FOREVER', KEEP_DELETED_CELLS => 'FAL
SE', BLOCKSIZE => '65536', IN_MEMORY => 'false', BLOCKCACHE => 'true'}
{NAME => 'personal_details', DATA_BLOCK_ENCODING => 'NONE', BLOOMFILTER => 'ROW', REPLICATION_SCOPE =
> '0', VERSIONS => '1', COMPRESSION => 'NONE', MIN_VERSIONS => '0', TTL => 'FOREVER', KEEP_DELETED_CE
LLS => 'FALSE', BLOCKSIZE => '65536', IN_MEMORY => 'false', BLOCKCACHE => 'true'}
3 row(s) in 0.0850 seconds
```

Hbase Commands

Check if table exists or not

exists ‘students’

```
hbase(main):014:0> exists 'students'
Table students does exist
0 row(s) in 0.0160 seconds
```

Hbase Commands

Drop a table

Drop is used to delete a HBase table. But this operator cannot be applied directly to the table. Instead, the table is first disabled and then dropped.

drop ‘students’

```
hbase(main):015:0> drop 'students'

ERROR: Table students is enabled. Disable it first.

Drop the named table. Table must first be disabled:
  hbase> drop 't1'
  hbase> drop 'ns1:t1'
```

Hbase Commands

So first disable it and then drop as shown below

```
hbase(main):016:0> disable 'students'
0 row(s) in 2.3180 seconds

hbase(main):017:0> drop 'students'
0 row(s) in 1.2790 seconds

hbase(main):018:0> list
TABLE
0 row(s) in 0.0150 seconds

=> []
hbase(main):019:0>
```

Hbase Commands

create 'census', 'personal', 'professional'

(census is the table name and personal and professional are the column families)
Insert records using put

```
hbase(main):040:0> scan 'census'
ROW                           COLUMN+CELL
 1                            column=personal:gender, timestamp=1590138104722, value=male
 1                            column=personal:marital_status, timestamp=1590138104695, value=unmarried
 1                            column=personal:name, timestamp=1590138104661, value=Mike Jones
 1                            column=professional:education_level, timestamp=1590138104775, value=high s
                              chool
 1                            column=professional:employed, timestamp=1590138104745, value=yes
 1                            column=professional:field, timestamp=1590138104796, value=construction
 2                            column=personal:gender, timestamp=1590138105008, value=male
 2                            column=personal:marital_status, timestamp=1590138104991, value=divorced
 2                            column=personal:name, timestamp=1590138104975, value=Ben
 3                            column=personal:gender, timestamp=1590138104947, value=female
 3                            column=personal:marital_status, timestamp=1590138104871, value=married
 3                            column=personal:name, timestamp=1590138104845, value=Jill Tang
 3                            column=personal:spouse, timestamp=1590138104901, value=Jim Tang
 3                            column=professional:education_level, timestamp=1590138104921, value=post-g
                              rad
 4                            column=personal:gender, timestamp=1590138105062, value=female
 4                            column=personal:marital_status, timestamp=1590138105046, value=divorced
 4                            column=personal:name, timestamp=1590138105032, value=Maria
4 row(s) in 0.0360 seconds
```

Hbase Commands

```
hbase(main):041:0> get 'census', 1
COLUMN                          CELL
 personal:gender                timestamp=1590138104722, value=male
 personal:marital_status        timestamp=1590138104695, value=unmarried
 personal:name                  timestamp=1590138104661, value=Mike Jones
 professional:education_le      timestamp=1590138104775, value=high school
 vel
 professional:employed          timestamp=1590138104745, value=yes
 professional:field             timestamp=1590138104796, value=construction
6 row(s) in 0.0190 seconds
```

```
hbase(main):042:0> get 'census', 1, 'personal:name'
COLUMN                          CELL
 personal:name                  timestamp=1590138104661, value=Mike Jones
1 row(s) in 0.0070 seconds
```

```
hbase(main):043:0> get 'census', 1, 'personal:name', 'professional:education_level'
COLUMN                          CELL
 personal:name                  timestamp=1590138104661, value=Mike Jones
 professional:education_le      timestamp=1590138104775, value=high school
 vel
2 row(s) in 0.0100 seconds
```

Hbase Commands

```
hbase(main):044:0> scan 'census', {COLUMNS => ['personal:name']}
ROW                           COLUMN+CELL
 1                            column=personal:name, timestamp=1590138104661, value=Mike Jones
 2                            column=personal:name, timestamp=1590138104975, value=Ben
 3                            column=personal:name, timestamp=1590138104845, value=Jill Tang
 4                            column=personal:name, timestamp=1590138105032, value=Maria
4 row(s) in 0.0220 seconds
```

```
hbase(main):047:0> scan 'census', {COLUMNS => ['personal:name'], LIMIT => 1}
ROW                           COLUMN+CELL
 1                            column=personal:name, timestamp=1590138104661, value=Mike Jones
1 row(s) in 0.0090 seconds
```

```
hbase(main):048:0> scan 'census', {COLUMNS => ['personal:name'], LIMIT => 1, STARTROW => "2"}
ROW                           COLUMN+CELL
 2                            column=personal:name, timestamp=1590138104975, value=Ben
1 row(s) in 0.0100 seconds
```

Hbase Commands

```
hbase(main):049:0> scan 'census', {COLUMNS => ['personal:name'], LIMIT => 3, STARTROW => "2"}
ROW                           COLUMN+CELL
 2                            column=personal:name, timestamp=1590138104975, value=Ben
 3                            column=personal:name, timestamp=1590138104845, value=Jill Tang
 4                            column=personal:name, timestamp=1590138105032, value=Maria
3 row(s) in 0.0130 seconds
```

```
hbase(main):050:0> scan 'census', {COLUMNS => ['personal:name'], STARTROW => "2", STOPROW => "3"}
ROW                           COLUMN+CELL
 2                            column=personal:name, timestamp=1590138104975, value=Ben
1 row(s) in 0.0130 seconds
```

Hbase Commands

Let us now try to see the hbase directory in hdfs

```
[cloudera@quickstart ~]$ hadoop fs -ls /hbase/
Found 8 items
drwxr-xr-x   - hbase supergroup          0 2020-05-22 01:23 /hbase/.tmp
drwxr-xr-x   - hbase supergroup          0 2020-05-22 02:06 /hbase/MasterProcWALs
drwxr-xr-x   - hbase supergroup          0 2020-05-22 01:23 /hbase/WALs
drwxr-xr-x   - hbase supergroup          0 2020-05-22 02:15 /hbase/archive
drwxr-xr-x   - hbase supergroup          0 2020-05-22 01:23 /hbase/data
-rw-r--r--   1 hbase supergroup         42 2020-05-22 01:22 /hbase/hbase.id
-rw-r--r--   1 hbase supergroup          7 2020-05-22 01:22 /hbase/hbase.version
drwxr-xr-x   - hbase supergroup          0 2020-05-22 01:22 /hbase/oldWALs
[cloudera@quickstart ~]$
```

Hbase Commands

```
[cloudera@quickstart ~]$ hadoop fs -ls /hbase/data/default/census/
Found 3 items
drwxr-xr-x   - hbase supergroup          0 2020-05-22 02:54 /hbase/data/default/census/.tabledesc
drwxr-xr-x   - hbase supergroup          0 2020-05-22 02:54 /hbase/data/default/census/.tmp
drwxr-xr-x   - hbase supergroup          0 2020-05-22 02:54 /hbase/data/default/census/6c2bd5ca5327401689d9a3ef7f3de9f5
[cloudera@quickstart ~]$
```

```
[cloudera@quickstart ~]$ hadoop fs -ls /hbase/data/default/census/6c2bd5ca5327401689d9a3ef7f3de9f5
Found 4 items
-rw-r--r--   1 hbase supergroup         41 2020-05-22 02:54 /hbase/data/default/census/6c2bd5ca5327401689d9a3ef7f3de9f5/.regioninfo
drwxr-xr-x   - hbase supergroup          0 2020-05-22 02:54 /hbase/data/default/census/6c2bd5ca5327401689d9a3ef7f3de9f5/personal
drwxr-xr-x   - hbase supergroup          0 2020-05-22 02:54 /hbase/data/default/census/6c2bd5ca5327401689d9a3ef7f3de9f5/professional
drwxr-xr-x   - hbase supergroup          0 2020-05-22 02:54 /hbase/data/default/census/6c2bd5ca5327401689d9a3ef7f3de9f5/recovered.edits
[cloudera@quickstart ~]$
```

```
[cloudera@quickstart ~]$ hadoop fs -ls /hbase/data/default/census/6c2bd5ca5327401689d9a3ef7f3de9f5/personal
[cloudera@quickstart ~]$ hadoop fs -ls /hbase/data/default/census/6c2bd5ca5327401689d9a3ef7f3de9f5/professional
[cloudera@quickstart ~]$
```

Hbase Commands

We cannot see Hfiles yet.. Because flush has not happened.

We can try that by using disable table option.

Disable flushes the in memory changes to disk.

```
[cloudera@quickstart ~]$ hadoop fs -ls /hbase/data/default/census/d4be49ff53fee104451a320d55c06c7d/
Found 5 items
-rw-r--r--   1 hbase supergroup         41 2020-05-22 03:04 /hbase/data/default/census/d4be49ff53fee104451a320d55c06c7d/.regioninfo
drwxr-xr-x   - hbase supergroup          0 2020-05-22 03:04 /hbase/data/default/census/d4be49ff53fee104451a320d55c06c7d/.tmp
drwxr-xr-x   - hbase supergroup          0 2020-05-22 03:04 /hbase/data/default/census/d4be49ff53fee104451a320d55c06c7d/personal
drwxr-xr-x   - hbase supergroup          0 2020-05-22 03:04 /hbase/data/default/census/d4be49ff53fee104451a320d55c06c7d/professional
drwxr-xr-x   - hbase supergroup          0 2020-05-22 03:04 /hbase/data/default/census/d4be49ff53fee104451a320d55c06c7d/recovered.edits
[cloudera@quickstart ~]$ hadoop fs -ls /hbase/data/default/census/d4be49ff53fee104451a320d55c06c7d/personal
Found 1 items
-rw-r--r--   1 hbase supergroup       1578 2020-05-22 03:04 /hbase/data/default/census/d4be49ff53fee104451a320d55c06c7d/personal/3f4d403c51a84
cf0b7d166933bd114b2
[cloudera@quickstart ~]$
```

Hbase Commands

Hbase Commands

Get data based on a filter condition:

In HBase, fetching data based on a filtering condition is achieved using filters.

These filters are like Java methods which take two input parameters — a logical operator and a comparator.

The logical operator specifies the type of the test, i.e., equals, less than, etc. The comparator is the number/value against which you wish to compare your record.

Hbase Commands

Some commonly used filter functions are:

1. ValueFilter

2. QualifierFilter

3. FamilyFilter

Hbase Commands

ValueFilter

scan 'census', {FILTER => "ValueFilter(=,'binary:Maria')"}

```
hbase(main):008:0> scan 'census', {FILTER => "ValueFilter(=,'binary:Maria')"}
ROW                           COLUMN+CELL
 4                            column=personal:name, timestamp=1590141871708, value=Maria
1 row(s) in 0.0110 seconds

hbase(main):009:0> scan 'census', {FILTER => "ValueFilter(=,'binary:Mariaa')"}
ROW                           COLUMN+CELL
0 row(s) in 0.0210 seconds

hbase(main):010:0>
```

Hbase Commands

QualifierFilter

scan 'census', {FILTER => "QualifierFilter(=,'substring:Name')"}

```
hbase(main):015:0> scan 'census', {FILTER => "QualifierFilter(=,'substring:Name')"}
ROW                           COLUMN+CELL
 1                            column=personal:name, timestamp=1590141871328, value=Mike Jones
 2                            column=personal:name, timestamp=1590141871656, value=Ben
 3                            column=personal:name, timestamp=1590141871526, value=Jill Tang
 4                            column=personal:name, timestamp=1590141871708, value=Maria
4 row(s) in 0.0240 seconds
```

Hbase Commands

FamilyFilter

scan 'census', {FILTER => "FamilyFilter(=,'substring:professional')"}

```
hbase(main):017:0> scan 'census', {FILTER => "FamilyFilter(=,'substring:professional')"}
ROW                          COLUMN+CELL
 1                           column=professional:education_level, timestamp=1590141871448, value=high s
                             chool
 1                           column=professional:employed, timestamp=1590141871429, value=yes
 1                           column=professional:field, timestamp=1590141871482, value=construction
 3                           column=professional:education_level, timestamp=1590141871598, value=post-g
                             rad
2 row(s) in 0.0090 seconds
```

Hbase Commands

Count number of records.

Count 'census'

```
hbase(main):018:0> count 'census'
4 row(s) in 0.0290 seconds

=> 4
```

We have learnt Hbase Shell Commands

Happy Learning!!!

Google REVIEWS

5 Star Google Rated
Big Data Course

LEARN FROM THE EXPERT

9108179578

Call for more details

Follow US

Trainer	Mr. Sumit Mittal
Phone	9108179578
Email	trendytech.sumit@gmail.com
Website	https://trendytech.in/courses/big-data-online-training/
LinkedIn	https://www.linkedin.com/in/bigdatabysumit/
Twitter	@BigdataBySumit
Instagram	bigdatabysumit
Facebook	https://www.facebook.com/trendytech.in/
Youtube	TrendyTech