HBase Practice Session

Realtime and Big Data Analytics - Fall 2018

If you're not using Dumbo, you can check that HMaster and HRegionServer (and QuorumPeerMain) are running.
(Note: Dumbo admins don't give us sudo permission.)

```
[cloudera@quickstart ~]$ sudo jps

      5250 QuorumPeerMain
      7946 Bootstrap
      7556 HRegionServer
      5326 DataNode
      6625 RESTServer
      7418 HistoryServer
      5429 JournalNode
      7386 Bootstrap
      5841 JobHistoryServer
      8145 Bootstrap
      6858 RunJar
      9277 Jps
      8181
      6187 ResourceManager
      6495 HMaster
      5927 NodeManager
      6941 RunJar
      5655 SecondaryNameNode
      6767 ThriftServer
      5518 NameNode
```

Start the HBase shell.

```
[cloudera@quickstart ~]$ hbase shell
```

Display all the HBase tables - you will see any tables created in the default HBase 'namespace'. A namespace in HBase is similar to a database in Hive or Impala.

```
list

      TABLE
      …
```

HBase Practice Session

Realtime and Big Data Analytics - Fall 2018

Before creating a table, first create your own HBase namespace - name it your netID:

```
create_namespace 'put_your_netid_here'
```

List all of the namespaces, verify that yours is listed:

```
list_namespace
```

View the HBase help information.

```
help
```

```
HBase Shell, version 1.2.0-cdh5.8.0, rUnknown, Thu Jun 16 12:46:57 PDT 2016
Type 'help "COMMAND"', (e.g. 'help "get"' -- the quotes are necessary) for help
on a specific command.
Commands are grouped. Type 'help "COMMAND_GROUP"', (e.g. 'help "general"') for
help on a command group.

COMMAND GROUPS:
  Group name: general
  Commands: status, table_help, version, whoami

  Group name: ddl
  Commands: alter, alter_async, alter_status, create, describe, disable,
disable_all, drop, drop_all, enable, enable_all, exists, get_table,
is_disabled, is_enabled, list, locate_region, show_filters

  Group name: namespace
  Commands: alter_namespace, create_namespace, describe_namespace,
drop_namespace, list_namespace, list_namespace_tables

  Group name: dml
  Commands: append, count, delete, deleteall, get, get_counter, get_splits,
incr, put, scan, truncate, truncate_preserve

  Group name: tools
  Commands: assign, balance_switch, balancer, balancer_enabled,
catalogjanitor_enabled, catalogjanitor_run, catalogjanitor_switch,
close_region, compact, compact_mob, compact_rs, flush, major_compact,
major_compact_mob, merge_region, move, normalize, normalizer_enabled,
normalizer_switch, split, trace, unassign, wal_roll, zk_dump

  Group name: replication
  Commands: add_peer, append_peer_tableCFs, disable_peer,
disable_table_replication, enable_peer, enable_table_replication, list_peers,
list_replicated_tables, remove_peer, remove_peer_tableCFs, set_peer_tableCFs,
show_peer_tableCFs
…
```

HBase Practice Session

Realtime and Big Data Analytics - Fall 2018

Display the status of the HBase cluster.

status

```
1 active master, 0 backup masters, 1 servers, 0 dead, 3.0000 average
load
```

Display the version of the HBase software.

version

```
1.2.0-cdh5.8.0, rUnknown, Thu Jun 16 12:46:57 PDT 2016
```

View the username of the logged-in user.

whoami

```
cloudera (auth:SIMPLE)
    groups: cloudera, default
```

Create an HBase table named `test1` in the namespace you created above and add a column family named `cf1`.

create 'yourNetId:test1', 'cf1'

```
0 row(s) in 1.3310 seconds

=> Hbase::Table - test1
```

List the tables in your namespace, verify that your new table is listed.

list_namespace_tables 'yourNetId'

```
TABLE
test1
1 row(s) in 0.0150 seconds
```

View the attributes for each column family defined for the table.

```
describe 'yourNetId:test1'
```

```
Table test1 is ENABLED
test1
COLUMN FAMILIES DESCRIPTION
{NAME => 'cf1', DATA_BLOCK_ENCODING => 'NONE', BLOOMFILTER => 'ROW',
REPLICATION
_SCOPE => '0', VERSIONS => '1', COMPRESSION => 'NONE', MIN_VERSIONS =>
'0', TTL
=> 'FOREVER', KEEP_DELETED_CELLS => 'FALSE', BLOCKSIZE => '65536',
IN_MEMORY =>
'false', BLOCKCACHE => 'true'}
1 row(s) in 0.2330 seconds
```

Change the number of versions stored to 3.

```
alter 'yourNetId:test1',NAME=>'cf1', VERSIONS =>3
```

```
Updating all regions with the new schema...
1/1 regions updated.
Done.
0 row(s) in 1.9600 seconds
```

View the attributes for each column family defined for the table.

```
describe 'yourNetId:test1'
```

```
Table test1 is ENABLED
test1
COLUMN FAMILIES DESCRIPTION
{NAME => 'cf1', DATA_BLOCK_ENCODING => 'NONE', BLOOMFILTER => 'ROW',
REPLICATION
_SCOPE => '0', VERSIONS => '3', COMPRESSION => 'NONE', MIN_VERSIONS =>
'0', TTL
=> 'FOREVER', KEEP_DELETED_CELLS => 'FALSE', BLOCKSIZE => '65536',
IN_MEMORY =>
'false', BLOCKCACHE => 'true'}
1 row(s) in 0.2330 seconds
```

HBase Practice Session

Realtime and Big Data Analytics - Fall 2018

Count the number of rows in table `test1`.

```
count 'yourNetId:test1'

      0 row(s) in 0.0570 seconds

      => 0
```

Add a row with rowkey = '1000', column family `cf1`, column name `col1` and value 'my_value' to table `test1`.

```
put 'yourNetId:test1', '1000', 'cf1:col1', 'my_value'

      0 row(s) in 0.0240 seconds
```

Display all rows in table `test1`. (Currently, there is just one row.)

```
scan 'yourNetId:test1'

      ROW                   COLUMN+CELL
       1000                 column=cf1:col1, timestamp=1491517139078, value=my_value
      1 row(s) in 0.0160 seconds
```

Add another row, set rowkey = '2000', column family `cf1`, column name `col1` and value 'my_value2' to table `test1`.

```
put 'yourNetId:test1', '2000', 'cf1:col1', 'my_value2'
      0 row(s) in 0.0060 seconds
```

Display all rows in table `test1`. (Currently, there are two rows.)

```
scan 'yourNetId:test1'

      ROW                   COLUMN+CELL
       1000                 column=cf1:col1, timestamp=1491517139078, value=my_value
       2000                 column=cf1:col1, timestamp=1491517173050, value=my_value2
      2 row(s) in 0.0140 seconds
```

HBase Practice Session

Realtime and Big Data Analytics - Fall 2018

Add another row with rowkey = '2000', column family `cf1`, column name `col1` and value 'my_value3' to table `test1`.

```
put 'yourNetId:test1', '2000', 'cf1:col1', 'my_value3'

      0 row(s) in 0.0050 seconds
```

Display the most recently stored values (`my_value3`, not `my_value2` is displayed).

```
scan 'yourNetId:test1'

      ROW                   COLUMN+CELL
       1000                 column=cf1:col1, timestamp=1491517139078, value=my_value
       2000                 column=cf1:col1, timestamp=1491517188889, value=my_value3
      2 row(s) in 0.0100 seconds
```

Display all stored values (both `my_value3`, and `my_value2` are displayed).

```
scan 'yourNetId:test1', VERSIONS => 3

     ROW                    COLUMN+CELL
      1000                  column=cf1:col1, timestamp=1491517139078, value=my_value
      2000                  column=cf1:col1, timestamp=1491517188889, value=my_value3
      2000                  column=cf1:col1, timestamp=1491517173050, value=my_value2
     2 row(s) in 0.0180 seconds
```

Display most recent value for rowkey = `1000`.

```
get 'yourNetId:test1', '1000'

      COLUMN                CELL
       cf1:col1             timestamp=1491517139078, value=my_value
      1 row(s) in 0.0140 seconds
```

Display most recent value for rowkey = `2000`.

```
get 'yourNetId:test1', '2000'

      COLUMN                CELL
       cf1:col1             timestamp=1491517139078, value=my_value3
      1 row(s) in 0.0050 seconds
```

HBase Practice Session

Realtime and Big Data Analytics - Fall 2018

Display data for the specified rowkey and column.

```
get 'yourNetId:test1', '2000', COLUMN=>'cf1:col1'

      COLUMN                CELL
       cf1:col1             timestamp=1491517188889, value=my_value3
      1 row(s) in 0.0070 seconds
```

Display all versions of data for specific rowkey and column.

```
get 'yourNetId:test1', '2000', {COLUMN=>'cf1:col1', VERSIONS=>3}

      COLUMN                CELL
       cf1:col1             timestamp=1491517188889, value=my_value3
       cf1:col1             timestamp=1491517173050, value=my_value2
      2 row(s) in 0.0190 seconds
```

Add rows with rowkey = '2000', column `cf1:col1` and values 'my_value4' and 'my_value5' to table `test1`.

```
put 'yourNetId:test1', '2000', 'cf1:col1', 'my_value4'
      0 row(s) in 0.0060 seconds


put 'yourNetId:test1', '2000', 'cf1:col1', 'my_value5'
      0 row(s) in 0.0060 seconds
```

Show up to 5 versions of data stored for rowkey = '2000', column `cf1:col1`.
Why are only 3 versions shown?

```
get 'yourNetId:test1', '2000', {COLUMN=>'cf1:col1', VERSIONS=>5}

      COLUMN                CELL
       cf1:col1             timestamp=1499896152488, value=my_value5
       cf1:col1             timestamp=1499896135981, value=my_value4
       cf1:col1             timestamp=1499895938208, value=my_value3
      3 row(s) in 0.0060 seconds
```

Change the number of versions stored to 2.

```
alter 'yourNetId:test1',NAME=>'cf1', VERSIONS =>2
```

```
Updating all regions with the new schema...
1/1 regions updated.
Done.
0 row(s) in 1.9180 seconds
```

View the attributes for each column family defined for the table.

```
describe 'yourNetId:test1'
```

```
Table test1 is ENABLED
test1
COLUMN FAMILIES DESCRIPTION
{NAME => 'cf1', DATA_BLOCK_ENCODING => 'NONE', BLOOMFILTER => 'ROW',
REPLICATION
_SCOPE => '0', VERSIONS => '2', COMPRESSION => 'NONE', MIN_VERSIONS =>
'0', TTL
=> 'FOREVER', KEEP_DELETED_CELLS => 'FALSE', BLOCKSIZE => '65536',
IN_MEMORY =>
'false', BLOCKCACHE => 'true'}
1 row(s) in 0.0390 seconds
```

Display up to 5 versions, note that only 2 are shown - why is that?

```
get 'yourNetId:test1', '2000', {COLUMN=>'cf1:col1', VERSIONS=>5}
```

```
COLUMN                CELL
 cf1:col1             timestamp=1499896152488, value=my_value5
 cf1:col1             timestamp=1499896135981, value=my_value4
2 row(s) in 0.0100 seconds
```