HBase Basics

Recommended extra resources:

Apache HBase Reference Guide:
https://hbase.apache.org/book.html#getting_started

Agenda:

- HW6 overview
- HBase Local setup
- HBase Shell (CLI)
- HBase on EMR
- Java and Python examples

Local HBase setup:

Follow installation instructions for Standalone HBase from here:
https://hbase.apache.org/book.html#quickstart

Update .conf/**hbase-site.xml**

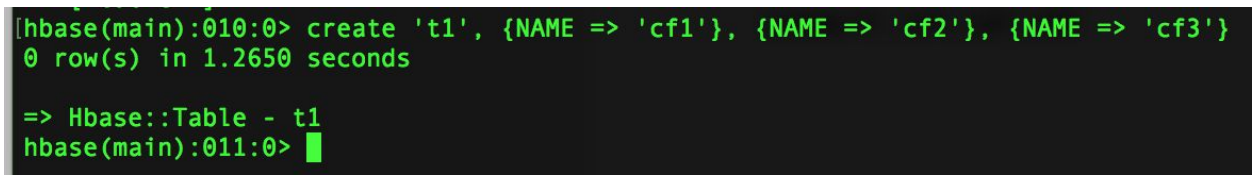
Start HBase:
./bin/start-hbase.sh

Stop HBase:
./bin/stop-hbase.sh

Start HBase shell (CLI):
./bin/hbase shell

Basic commands:

> create 't1', {NAME => 'cf1'}, {NAME => 'cf2'}, {NAME => 'cf3'}
Same as:
> create 't1', 'cf1', 'cf2', 'cf3'



> describe 't1'
> list

```
hbase(main):011:0> describe 't1'
Table t1 is ENABLED
t1
COLUMN FAMILIES DESCRIPTION
{NAME => 'cf1', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', COMPRESSION => 'NONE', MIN_VE
RSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}
{NAME => 'cf2', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', COMPRESSION => 'NONE', MIN_VE
RSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}
{NAME => 'cf3', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', COMPRESSION => 'NONE', MIN_VE
RSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}
3 row(s) in 0.0290 seconds

hbase(main):012:0> list
TABLE
t1
table1
2 row(s) in 0.0210 seconds

=> ["t1", "table1"]
```

> exists 't1'

```
hbase(main):017:0> exists 't1'
Table t1 does exist
0 row(s) in 0.0290 seconds
```

> put 't1', 'rowkey1', 'cf1:column1', 'v1'

> get 't1', 'rowkey1'

Note: watch the timestamp of the version:

```
hbase(main):014:0> put 't1', 'rowkey1', 'cf1:column1', 'v1'
0 row(s) in 0.0070 seconds

hbase(main):015:0> put 't1', 'rowkey1', 'cf1:column1', 'v2'
0 row(s) in 0.0030 seconds

hbase(main):016:0> get 't1', 'rowkey1'
COLUMN                                          CELL
 cf1:column1                                    timestamp=1539524393707, value=v2
1 row(s) in 0.0230 seconds

hbase(main):017:0> exists 't1'
Table t1 does exist
0 row(s) in 0.0290 seconds

hbase(main):018:0>
```

Working with versions:
alter 't1', { NAME => 'f1', VERSIONS => 3 }

```
hbase(main):018:0> alter 't1', NAME => 'cf1', VERSIONS => 3
Updating all regions with the new schema...
1/1 regions updated.
Done.
0 row(s) in 1.9340 seconds

hbase(main):019:0> put 't1', 'rowkey1', 'cf1:column1', 'v3.1'
0 row(s) in 0.0040 seconds

hbase(main):020:0> put 't1', 'rowkey1', 'cf1:column1', 'v3.2'
0 row(s) in 0.0030 seconds

hbase(main):021:0> put 't1', 'rowkey1', 'cf1:column1', 'v3.3'
0 row(s) in 0.0030 seconds

hbase(main):022:0> put 't1', 'rowkey1', 'cf1:column1', 'v3.4'
0 row(s) in 0.0040 seconds

hbase(main):023:0> put 't1', 'rowkey1', 'cf1:column1', 'v3.5'
0 row(s) in 0.0020 seconds

hbase(main):024:0> get 't1', 'rowkey1', {COLUMN => 'cf1:column1', VERSIONS => 2}
COLUMN                                  CELL
 cf1:column1                            timestamp=1539524926121, value=v3.5
 cf1:column1                            timestamp=1539524922646, value=v3.4
1 row(s) in 0.0250 seconds

hbase(main):025:0>
```

```
hbase(main):025:0> get 't1', 'rowkey1', {COLUMN => 'cf1:column1', VERSIONS => 5}
COLUMN                                  CELL
 cf1:column1                            timestamp=1539524926121, value=v3.5
 cf1:column1                            timestamp=1539524922646, value=v3.4
 cf1:column1                            timestamp=1539524917202, value=v3.3
1 row(s) in 0.0080 seconds

hbase(main):026:0>
```

Run commands from a script in HBase CLI:
./bin/hbase shell ./hbase_commands.txt

Create a namespace
create_namespace 'lab'

create_namespace, alter_namespace, describe_namespace, drop_namespace, list_namespace, list_namespace_tables

Run java spark job
/usr/lib/spark/spark-submit --class edu.harvard.e88.lab5.HoursCounterSparkJobParquet lab5-0.0.1-SNAPSHOT.jar input

To add hbase libraries into the spark classpath - update spark.driver.extraClassPath and spark.executor.extraClassPath in spark-defaults.conf

sudo vim /etc/spark/conf/spark-defaults.conf

:/usr/lib/hbase/*:/usr/lib/hadoop/hadoop-aws.jar:/usr/lib/hbase/lib/htrace-core-3.1.0-incubating.jar :/usr/lib/hbase/lib/metrics-core-2.2.0.jar

Run Python spark job
~/opt/spark/bin/spark-submit --packages com.hortonworks:shc-core:1.1.1-2.1-s_2.11 --repositories http://repo.hortonworks.com/content/repositories/releases/ HoursCounterSparkHbase.py

Python examples

Spark-HBase Connector

package from Hortonworks to interact with HBase from Spark.

Happybase

Happybase is a pure Python library to connect with HBase via its Thrift API.

```
conn = happybase.Connection(host = "172.31.18.179",table_prefix = "lab",
table_prefix_separator = ":")
table = conn.table("date_hour")
row = table.row(b'2018-09-15T00')
print(row[b'url:count'])

table.put(b'2018-09-15T22', {b'url:count': b'100000'})
row1 = table.row(b'2018-09-15T22')
print(row1[b'url:count'])
```