1.2. Quick Start

This guide describes setup of a standalone HBase instance that uses the local filesystem. It leads you through creating a table, inserting rows via the HBase shell, and then cleaning up and shutting down your standalone HBase instance. The below exercise should take no more than ten minutes (not including download time).

1.2.1. Download and unpack the latest stable release.

Choose a download site from this list of Apache Download Mirrors. Click on suggested top link. This will take you to a mirror of *HBase Releases*. Click on the folder named stable and then download the file that ends in .tar.gz to your local filesystem; e.g. hbase-0.91.0-SNAPSHOT.tar.gz.

Decompress and untar your download and then change into the unpacked directory.

```
$ tar xfz hbase-0.91.0-SNAPSHOT.tar.gz
$ cd hbase-0.91.0-SNAPSHOT
```

At this point, you are ready to start HBase. But before starting it, you might want to edit conf/hbase-site.xml and set the directory you want HBase to write to, hbase.rootdir.

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
 <property>
  <name>hbase.rootdir</name>
  <value>file:///DIRECTORY/hbase</value>
 </property>
</configuration>
```

Replace DIRECTORY in the above with a path to a directory where you want HBase to store its data. By default, hbase.rootdir is set to /tmp/hbase-${user.name} which means you'll lose all your data whenever your server reboots (Most operating systems clear /tmp on restart).

1.2.2. Start HBase

Now start HBase:

```
$ ./bin/start-hbase.sh
starting Master, logging to logs/hbase-user-master-example.org.out
```

You should now have a running standalone HBase instance. In standalone mode, HBase runs all daemons in the the one JVM; i.e. both the HBase and ZooKeeper daemons. HBase logs can be found in the logs subdirectory. Check them out especially if HBase had trouble starting.

Is java installed?

All of the above presumes a 1.6 version of Oracle java is installed on your machine and available on your path; i.e. when you type java, you see output that describes the options the java program takes (HBase requires java 6). If this is not the case, HBase will not start. Install java, edit conf/hbase-env.sh, uncommenting the JAVA_HOME line pointing it to your java install. Then, retry the steps above.

1.2.3. Shell Exercises

Connect to your running HBase via the shell.

```
$ ./bin/hbase shell
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version: 0.90.0, r1001068, Fri Sep 24 13:55:42 PDT 2010

hbase(main):001:0>
```

Type help and then <RETURN> to see a listing of shell commands and options. Browse at least the paragraphs at the end of the help emission for the gist of how variables and command arguments are entered into the HBase shell; in particular note how table names, rows, and columns, etc., must be quoted.

Create a table named test with a single column family named cf. Verify its creation by listing all tables and then insert some values.

```
hbase(main):003:0> create 'test', 'cf'
0 row(s) in 1.2200 seconds
hbase(main):003:0> list 'table'
```

```
test
1 row(s) in 0.0550 seconds
hbase(main):004:0> put 'test', 'row1', 'cf:a', 'value1'
0 row(s) in 0.0560 seconds
hbase(main):005:0> put 'test', 'row2', 'cf:b', 'value2'
0 row(s) in 0.0370 seconds
hbase(main):006:0> put 'test', 'row3', 'cf:c', 'value3'
0 row(s) in 0.0450 seconds
```

Above we inserted 3 values, one at a time. The first insert is at row1, column cf:a with a value of value1. Columns in HBase are comprised of a column family prefix -- cf in this example -- followed by a colon and then a column qualifier suffix (a in this case).

Verify the data insert.

Run a scan of the table by doing the following

```
hbase(main):007:0> scan 'test'
ROW      COLUMN+CELL
row1     column=cf:a, timestamp=1288380727188, value=value1
row2     column=cf:b, timestamp=1288380738440, value=value2
row3     column=cf:c, timestamp=1288380747365, value=value3
3 row(s) in 0.0590 seconds
```

Get a single row as follows

```
hbase(main):008:0> get 'test', 'row1'
COLUMN    CELL
cf:a      timestamp=1288380727188, value=value1
1 row(s) in 0.0400 seconds
```

Now, disable and drop your table. This will clean up all done above.

```
hbase(main):012:0> disable 'test'
0 row(s) in 1.0930 seconds
hbase(main):013:0> drop 'test'
0 row(s) in 0.0770 seconds
```

Exit the shell by typing exit.

```
hbase(main):014:0> exit
```

1.2.4. Stopping HBase

Stop your hbase instance by running the stop script.

```
$ ./bin/stop-hbase.sh
stopping hbase.....
```

1.2.5. Where to go next

The above described standalone setup is good for testing and experiments only. Next move on to Chapter 2, *Configuration* where we'll go into depth on the different HBase run modes, requirements and critical configurations needed setting up a distributed HBase deploy.

Prev Up Next
Chapter 1. Getting Started Home Chapter 2. Configuration