

Taller HBase

Instalación

- i. Valide que la variable `JAVA_HOME` esté definida:

```
$ java -version
java version "1.8.0_121"
Java(TM) SE Runtime Environment (build 1.8.0_121-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.121-b13, mixed mode)
```

- i. Descargue la última versión estable de [HBase](#) del sitio oficial (a Mayo de 2019 la versión estable es 2.1.4)
- ii. Ingrese a la carpeta donde descargó el archivo y descomprimalo:

```
$ cd /home/mls/Descargas
$ tar xzvf hbase-2.1.4-bin.tar.gz
```

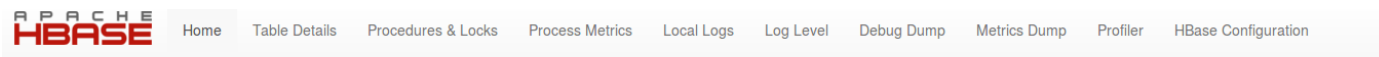
- i. Edite el archivo `/home/mls/Descargas/hbase-2.1.4/conf/hbase-site.xml`, el archivo debe quedar así en la sección de configuración:

```
...
<configuration>
  <property>
    <name>hbase.rootdir</name>
    <value>file:///home/mls/hbase</value>
  </property>
  <property>
    <name>hbase.zookeeper.property.dataDir</name>
    <value>/home/mls/zookeeper</value>
  </property>
  <property>
    <name>hbase.unsafe.stream.capability.enforce</name>
    <value>false</value>
    <description>
      Controls whether HBase will check for stream capabilities (hflush/hsync).
    </description>
  </property>
</configuration>
```

- i. Abra una terminal y sitúese en la carpeta de HBase(
`/home/mls/Descargas/hbase-2.1.4`). Luego inicie el servicio de HBase:

```
$ ./bin/start-hbase.sh
running master, logging to /home/mlds/Descargas/hbase-2.1.4/bin/../logs/hbase-ml
```

- i. Valide que el servicio funciona correctamente ingresando a <http://localhost:16010/>. Debe aparecer una interfaz similar:



Master Lisi

Region Servers

Base Stats	Memory	Requests	Storefiles	Compactions	Replications
ServerName	Start time	Last contact	Version	Requests Per Second	Num. Regions
lisi.16020.1559083912047	Tue May 28 17:51:52 COT 2019	2 s	2.1.4	0	0
Total:1				0	0

Rackin Masters

Consola de HBase

- i. Para ejecutar los comandos básicos debe ingresar primero a la consola de HBase. Puede usar la misma terminal del ejemplo anterior.

```
$ ./bin/hbase shell
2019-05-27 16:09:04,669 WARN [main] util.NativeCodeLoader: Unable to load native
HBase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
For Reference, please visit: http://hbase.apache.org/2.0/book.html#shell
Version 2.1.4, r5b7722f8551bca783adb36a920ca77e417ca99d1, Tue Mar 19 19:05:06 UTC
Took 0.0023 seconds
hbase(main):001:0>
```

Comandos básicos

create para crear tablas

El siguiente comando crea la tabla `test` con un `ColumnFamily` de nombre `cf`

```
hbase(main):001:0> create 'test', 'cf'
0 row(s) in 0.4170 seconds
=> Hbase::Table - test
```

list para listar las tablas

Con el comando `list` se pueden ver todas las tablas creadas.

```
hbase(main):002:0> list
TABLE
test
1 row(s) in 0.0180 seconds
=> ["test"]
```

put para insertar datos

Los siguientes comandos usan `put` para insertar 3 registros en la tabla `test`. Cada uno con las Keys `row1`, `row2` y `row3` respectivamente. También, asigna los valores `a=value1`, `b=value2`, y `c=value3`

```
hbase(main):003:0> put 'test', 'row1', 'cf:a', 'value1'
0 row(s) in 0.0850 seconds
hbase(main):004:0> put 'test', 'row2', 'cf:b', 'value2'
0 row(s) in 0.0110 seconds
hbase(main):005:0> put 'test', 'row3', 'cf:c', 'value3'
0 row(s) in 0.0100 seconds
```

scan para leer los datos

El siguiente comando lista los datos de la tabla `test`

```
hbase(main):006:0> scan 'test'
ROW                                COLUMN+CELL
 row1                             column=cf:a, timestamp=1421762485768, value=
 row2                             column=cf:b, timestamp=1421762491785, value=
 row3                             column=cf:c, timestamp=1421762496210, value=
3 row(s) in 0.0230 seconds
```

get para recuperar un registro

El siguiente comando obtiene el registro de la tabla `test` cuyo Key sea `row1`

```
hbase(main):007:0> get 'test', 'row1'
COLUMN                                CELL
 cf:a                                timestamp=1421762485768, value=value1
1 row(s) in 0.0350 seconds
```

put para actualizar datos

El siguiente comando actualiza la columna `cf:a` del registro con Key `row1` en la tabla `test`

```
hbase(main):008:0> put 'test', 'row1', 'cf:a', 'new Value'
0 row(s) in 0.0040 seconds
```

Haga el `get` para validar que el cambio se realizó.

```
hbase(main):009:0> get 'test', 'row1'
COLUMN                                CELL
cf:a                                timestamp=1488921801701, value=new Value
1 row(s) in 0.0050 seconds
```


`delete` para borrar datos de una columna específica

El siguiente comando borra el valor la columna `cf:c` del registro `row3`.

```
hbase(main):010:0> delete 'test', 'row3', 'cf:c'
0 row(s) in 0.0060 seconds
```

Haga el `get` para validar que el cambio se realizó.

```
hbase(main):011:0> scan 'test'
ROW                                COLUMN+CELL
row1                                column=cf:a, timestamp=1488921801701, value=new
row2                                column=cf:b, timestamp=1488921775047, value=valu
2 row(s) in 0.0050 seconds
```



Como `row3` ahora no tiene ninguna celda, no aparece en los resultados.

`deleteall` para borrar todos datos

`delete` solo borra una columna a la vez de cada registro. Para borrar todas las columnas:

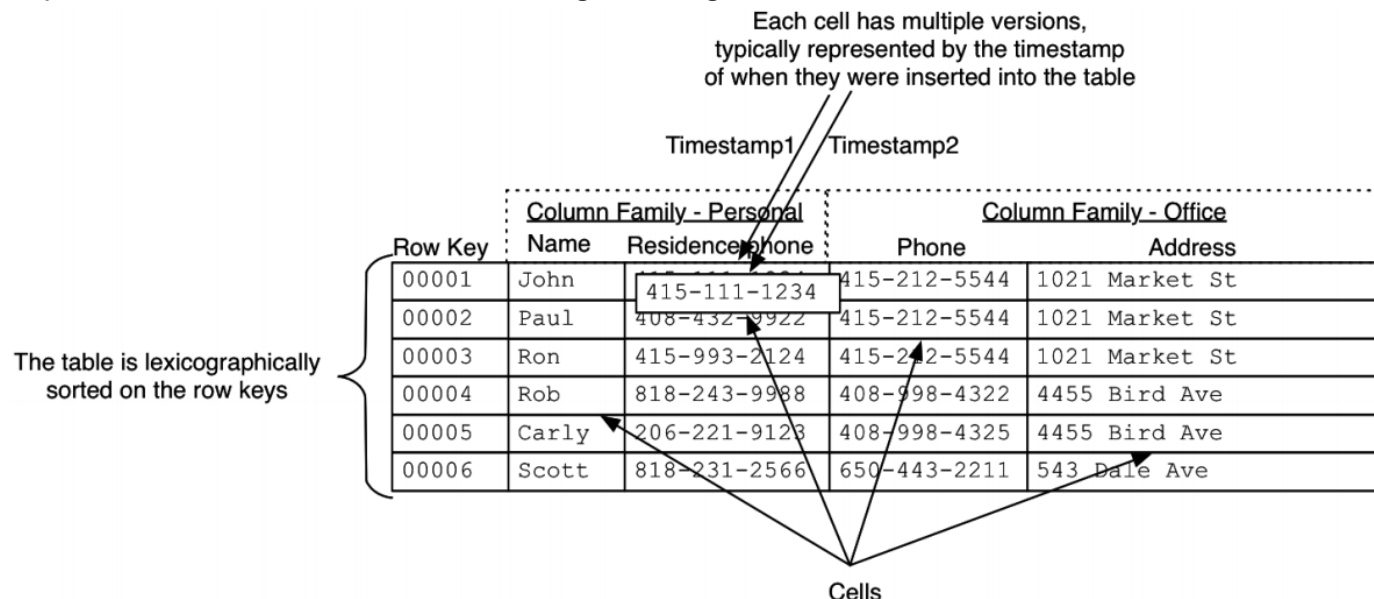
```
hbase(main):012:0> deleteall 'test', 'row2'
0 row(s) in 0.0020 seconds
```

Otros comandos

Visite este [blog](#) para ver una lista más detallada de los comandos de HBase shell.

Demo Agendas

Implementar el modelo ilustrado en la siguiente figura:



Modelo de datos

```

> create 'Agenda', 'Personal', 'Oficina'
> list
> put 'Agenda', '1', 'Personal:Nombre', 'Juan'
> scan 'Agenda'
> put 'Agenda', '1', 'Personal:Telefono', '312 123 45 67'
> scan 'Agenda'
> put 'Agenda', '1', 'Oficina:Telefono', '312 555 66 887'
> put 'Agenda', '1', 'Oficina:Direccion', 'Calle Falsa 123'
> put 'Agenda', '1', 'Personal:Telefono', '312 123 45 97', 1
> get 'Agenda', '1', {COLUMN => 'Personal:Telefono'}
> get 'Agenda', '1', {COLUMN => 'Personal:Telefono', TIMESTAMP => 1}
> get 'Agenda', '1'
> get 'Agenda', '1', ['Personal:Telefono', 'Oficina:Telefono']
> put 'Agenda', '2', 'Personal:Nombre', 'Maria'
> put 'Agenda', '2', 'Personal:Telefono', '320 987 65 43', 1
> put 'Agenda', '2', 'Personal:Telefono', '313 356 12 87', 3
> put 'Agenda', '2', 'Personal:Telefono', '321 987 65 21', 2
> get 'Agenda', '2', {COLUMN => 'Personal:Telefono', TIMERANGE => [1,3]}
> get 'Agenda', '2', {COLUMN => 'Personal:Telefono', TIMERANGE => [2,4]}
> get 'Agenda', '2', {COLUMN => 'Personal:Telefono', VERSIONS => 4}
> put 'Agenda', '2', 'Oficina:Telefono', '555 123 87 45'
> put 'Agenda', '2', 'Oficina:Direccion', 'Area 51'
> scan 'Agenda'
> delete 'Agenda', '2', 'Personal:Telefono', 1
> get 'Agenda', '2', {COLUMN => 'Personal:Telefono', VERSIONS => 3}

```

Taller entregable

Repita el taller de Cassandra de Twitter usando HBase.

Entregables

- Diagrama del modelo de datos (Tabla indicando sus dimensiones: filas, familia de columnas y columnas).
- Script de creación del modelo de datos.
- Explicación de cómo se realizarían las lecturas y escrituras en términos de operaciones put, update, get, scan y delete.

Comprima los archivos en un zip y use el siguiente link para cargar el desarrollo del taller:

[Dropbox request](#)

Deadline: 9 de Julio de 2019