Guia de instalación y uso de HBase

Prerrequisitos

- Sistema Operativo Ubuntu 18.04
- Tener instalado Java JDK (versión 8)

Instalación

Descargar de la página https://hbase.apache.org/downloads.html la opción bin. Se descargará un archivo tar.gz.

Descomprimir el archivo

```
$ tar xzvf hbase-2.2.4-bin.tar.gz
$ sudo mv hbase-2.2.4 /opt
```

Para situarse en el directorio de la instalación:

```
$ cd /opt/hbase-2.2.4
```

Configurar variable de entorno JAVA_HOME en sistema operativo

Situarse en el path raiz del usuario

```
$ cd
```

Abrir el archivo .profile para cambiar la variable de configuración

```
$ sudo gedit .profile
```

Aumentar la palabra export en la linea donde se esta configurando el JAVA_HOME (al final del archivo).

Crear directorio data

Situarse en la raiz del path del usuario.

```
$ cd
```

Crear el directorio data.

```
$ mkdir data
```

Configuración de un solo nodo en solitario

Configurar JAVA_HOME en HBase

Setear la variable de entorno JAVA_HOME. Para esto podemos modificar el script conf/hbase-env.sh para que este se encargue de configurar esta variable.

Abrir el archivo /opt/hbase-2.2.4/conf/hbase-env.sh

```
$ gedit /opt/hbase-2.2.4/conf/hbase-env.sh
```

Descomentar la línea #export JAVA_HOME=. Posteriormente aumentar la ruta donde se encuentra instalado el jdk.

```
...
export JAVA_HOME=$JAVA_HOME
...
```

Propiedades de la base de datos

Para poder setear las distintas propiedades de la base de datos, se deberá de modificar el archivo conf/hbase-site.xml.

En este caso, al ser una instalación para un servicio en solitario (stand alone) debemos configurar la ruta de los directorios donde queramos que la base de datos guarde los datos. Por esto modificaremos la propiedad **hbase,rootdir** y la propiedad **hbase.zookeeper.property.dataDir**.

Abrimos el archivo conf/hbase-site.xml

```
$ gedit /opt/hbase-2.2.4/conf/hbase-site.xml
```

Editamos el archivo agregando las propiedades que deseamos definir:

```
<configuration>
  <property>
    <name>hbase.rootdir</name>
    <value>file:///home/ulima/data/hbase</value>
  </property>
  <property>
    <name>hbase.zookeeper.property.dataDir</name>
    <value>/home/ulima/data/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).

      Disable this if you intend to run on LocalFileSystem, denoted by a
rootdir
      with the 'file://' scheme, but be mindful of the NOTE below.

      WARNING: Setting this to false blinds you to potential data loss and
      inconsistent system state in the event of process and/or node
failures. If
      HBase is complaining of an inability to use hsync or hflush it's
most
      likely not a false positive.
    </description>
  </property>
</configuration>
```

Utilización

Iniciar servicio

Para iniciar el servicio, desde una terminal ejecutaremos el siguiente comando:

```
$ /opt/hbase-2.2.4/bin/start-hbase.sh
```

Para verificar que se haya iniciado el servicio, puede ejecutar el siguiente comando:

```
$ jps
```

Si tiene un proceso llamado HMaster significa que el servicio está activo y ejecutándose.

Conectarse al servicio

Desde el terminal, deberá de ejecutar el siguiente comando:

```
$ /opt/hbase-2.2.4/bin/hbase shell
```

Si es que se muestra el shell interactivo de Hbase, significa que nos pudimos conectar con el servicio.

Para el servicio

Desde el terminar, deberá de ejcutar el siguiente comando:

```
$ /opt/hbase-2.2.4/bin/stop-hbase.sh
```

Sentencias DDL (Data Definition Language)

Crear una nueva tabla

```
hbase(main):001:0> create 'alumno', 'dp'
```

El primer es el nombre de la tabla y el segundo es el nombre de la **familia de columnas**. En este caso dp son los datos personales.

Visualizar información de una tabla.

```
hbase(main):002:0> describe 'alumno'
```

Desactivar / Activar una tabla

```
hbase(main):010:0> disable 'alumno'
hbase(main):011:0> enable 'alumno'
```

Eliminar una tabla

Para poder eliminar una tabla, primero hay que desactivarla.

```
hbase(main):012:0> disable 'alumno'
hbase(main):013:0> drop 'alumno'
```

Sentencias DML (Data Manipulation Language)

Insertando data.

```
hbase(main):004:0> put 'alumno', '20101212', 'dp:nombre', 'Billy Grados'

hbase(main):005:0> put 'alumno', '20101212', 'dp:edad', 32

hbase(main):006:0> put 'alumno', '20142323', 'dp:nombre', 'Juana Perez'

hbase(main):007:0> put 'alumno', '20142323', 'dp:edad', 24
```

Como se ve, se utiliza el comando put con 4 parámetros:

1. Nombre de la tabla
2. Identificador del registro
3. Nombre de la columna (campo)
4. Valor de la columna

Obteniendo data de una tabla

Toda la data de una tabla

```
hbase(main):008:0> scan 'alumno'
```

Data de cierto registro en específico.

```
hbase(main):009:0> get 'alumno', '20101212'
```

Como segundo parámetro se pone el identificador del registro.

Client API

Configuración

- Instalar librería happybase

```
$ conda install -c conda-forge happybase
```

Inicializar servicio de Thrift

```
$ /opt/hbase-2.2.4/bin/hbase-daemon.sh start thrift
```

Inicializar la base de datos HBase

```
$ /opt/hbase-2.2.4/bin/start-hbase.sh
```

Utilizando el API con python

El siguiente código en python permite registrar una nueva row en la la tabla alumno.

```
```

```
import happybase

connection = happybase.Connection('localhost')
table = connection.table('alumno')

table.put(b'20153434', {
    b'dp:nombre': b'Julian Perez',
    b'dp:ead': b'20'})
```

Para poder obtener (query) por datos:

```
row = table.row(b'20153434')
print(row[b'dp:nombre'])
```