

Prácticas BigData

HIVE

1. Instalar, crear conexión, bases de datos y tablas

- Descomprimir el software en la carpeta /opt/hadoop y denominamos al directorio hive. Debe quedar /opt/hadoop/hive
- Poner lo siguiente en el .bashrc o en el fichero de arranque

```
export HIVE_HOME=/opt/hadoop/hive
export PATH=$PATH:$HIVE_HOME/bin:$HIVE_HOME/conf
```

- Accedemos al directorio conf y copiamos los ficheros de templates para convertirlos en xml

```
cp hive-default.xml.template hive-site.xml
cp hive-env.sh.template hive-env.sh
cp hive-exec-log4j.properties.template hive-execlog4j.properties
cp hive-log4j.properties.template hive-log4j.properties
```

- Modificamos el hive.env.sh y configuramos los siguientes parámetros

```
export HADOOP_HOME=/opt/hadoop
export HIVE_CONF_DIR=/opt/hadoop/hive/conf
```

- Creamos dentro de HDFS la estructura de HIVE. Por defecto está apuntando a estos directorios

```
$ hdfs dfs -mkdir /tmp
$ hdfs dfs -mkdir -p /user/hive/warehouse
$ hdfs dfs -chmod g+w /tmp
$ hdfs dfs -chmod g+w /user/hive/warehouse
```

- Accedemos a hive-site.xml y ponemos estos valores de configuración al principio del fichero

```
<property>
  <name>system:java.io.tmpdir</name>
  <value>/tmp/hive/java</value>
</property>
<property>
  <name>system:user.name</name>
  <value>${user.name}</value>
</property>
```

- Creamos un directorio donde guardar la metastore

```
mkdir bbdd
cd bbdd
```

- Creamos la Base de Datos y su configuración

schematool -dbType derby -initSchema

SLF4J: Class path contains multiple SLF4J bindings.

SLF4J: Found binding in [jar:file:/opt/hadoop/hive/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]

SLF4J: Found binding in [jar:file:/opt/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]

SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.

SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]

Metastore connection URL:

jdbc:derby;;databaseName=metastore_db;create=true

Metastore Connection Driver : org.apache.derby.jdbc.EmbeddedDriver

Metastore connection User: APP

Starting metastore schema initialization to 2.1.0

Initialization script hive-schema-2.1.0.derby.sql

Initialization script completed

schemaTool completed

- Arrancamos el cliente hive

```
hive
```

- Comprobar las bases de datos

```
show databases;
```

- Crear la Base de datos

```
create database ejemplo;
```

- Comprobar en HDFS que existe

Browse Directory

/user/hive/warehouse

Go!

Show 25 entries

Search:

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxrwxr-x	hadoop	supergroup	0 B	Jan 08 21:00	0	0 B	ejemplo.db

Showing 1 to 1 of 1 entries

Previous Next

Hadoop, 2017.

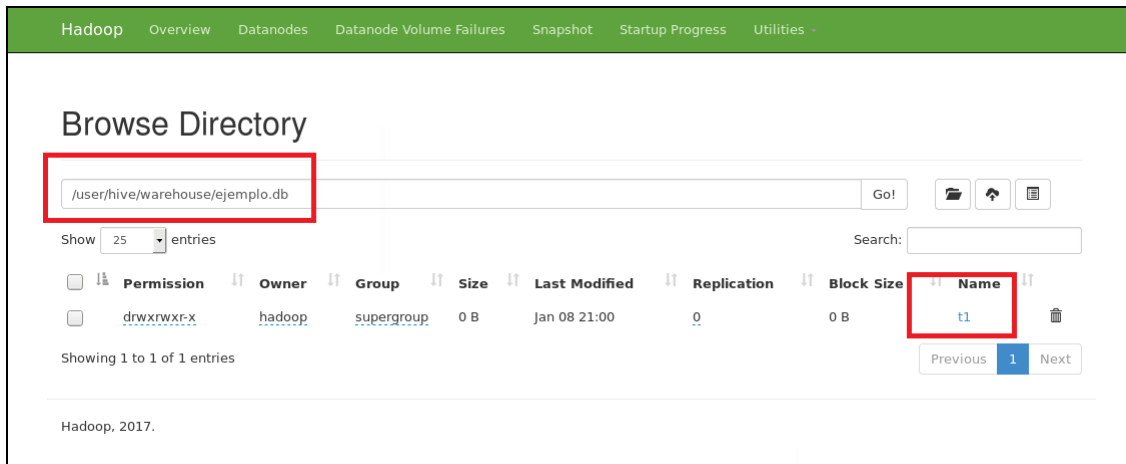
- Acceder a la Base de datos

```
use ejemplo;
```

- Crear una pequeña tabla

```
create table if not exists t1
(
  name string
);
```

- Comprobar en hdfs que existe
-



The screenshot shows the Hadoop Browse Directory interface. The address bar contains the path `/user/hive/warehouse/ejemplo.db`. Below the address bar, there is a table with the following columns: Permission, Owner, Group, Size, Last Modified, Replication, Block Size, and Name. The table contains one entry with the name `t1`. The Name column is highlighted with a red box.

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxrwxr-x	hadoop	supergroup	0 B	Jan 08 21:00	0	0 B	t1

- Insertar alguna fila en la tabla y comprobar el proceso mapreduce

```
Insert into t1 values ('mi nombre');
```

- Comprobamos que la ha guardado

```
select * from t1;
```

- Como práctica adicional, visualizar, el contenido de la tabla a través del comando siguiente. Debemos ver el fichero que ha creado

```
h Hdfs dfs -cat /xxxxxxx
```