



Proyecto Hadoop 2_6



Guillermo Fora Goncer

ÍNDICE

Introducción.....	3
Creación de la máquina.....	3
Dataset.....	3
Descargar Dataset de Kaggle.....	3
Explorar archivo CSV.....	4
Subir CSV del Dataset.....	4
Subir CSV al HDFS.....	6
Manual de instancias CRUB HBase.....	6
Manual de instancias CRUB Hive.....	8
HIVE.....	9
Creación de la tabla en Hive con HUE.....	9
Consultas en Hive.....	15
Hbase.....	18
Creación de la tabla en Hbase con HUE.....	18
Consultas HBase.....	21

Introducción

Hemos visto que Hadoop es un ecosistema que consta de diferentes herramientas para almacenar, procesar y analizar datos Big Data (Petabytes: 10^{15} byte).

Hasta ahora hemos aprendido cómo instalar un clúster Hadoop en máquinas virtuales, así como en AWS y cómo lanzar tareas utilizando el motor MapReduce.

Para avanzar en el estudio de las herramientas del ecosistema Hadoop en este proyecto nos centraremos en Hive (SQL), HBase(NoSQL) y HUE (interfaz gráfica).

Lo que vamos a hacer es descargarnos desde Kaggle algunos ficheros, subirlos a Hadoop y utilizando Hive y HBase haremos consultas sobre sus datos, la herramienta HUE la podemos usar como interfaz.

Utilizaré este *tipo de formato para los comandos*.

Creación de la máquina

Proyecto_hadoopSe ha actualizado hace menos de un minuto

▼ Resumen

Información del clúster
ID del clúster
j-1WGSQWYR873J
Configuración del clúster
Grupos de instancias
Capacidad
1 Primary (Principal) | 3 Principal | 0 Tarea

Aplicaciones
Versión de Amazon EMR
emr-7.5.0
Aplicaciones instaladas
Hadoop 3.4.0, HBase 2.5.10, Hive 3.1.3, Hue 4.11.0

Administración de clústeres
Destino del registro en Amazon S3
[aws-logs-992382578743-us-east-1/elasticmapreduce](#)
IU de aplicación persistente
[Servidor de línea de tiempo de YARN](#)
[UI de Tez](#)
DNS público del nodo principal
[ec2-3-238-50-162.compute-1.amazonaws.com](#)
Conectarse al nodo principal mediante SSH
[Conectarse al nodo principal mediante SSM](#)

Dataset

Descargar Dataset de Kaggle

Para este proyecto he elegido este dataset (Películas de marvel)

<https://www.kaggle.com/datasets/sarthakbharad/marvel-movies-dat>

Explorar archivo CSV

Vamos a hacer una exploración del CSV del Dataset para que luego sea más fácil el proceso de creación de tablas.

Comprobamos que el .csv esta en ingles y tienes los siguientes nombres de columnas: index, Title, Director (1), Director (2), Release Date (DD-MM-YYYY), IMDb (scored out of 10), IMDB Metascore (scored out of 100), Rotten Tomatoes - Critics (scored out of 100%), Rotten Tomatoes - Audience (scored out of 100%), Letterboxd (scored out of 5), CinemaScore (grades A+ to F), Budget (in million \$), Domestic Gross (in million \$), Worldwide Gross (in million \$)

Esta es la descripción de las columnas:

- **Nombre de película, director 1, director 2 y fecha de lanzamiento**
- **IMBd** es un sitio web donde se puntúan las películas del 1 al 10.
- **IMBd Metascore** es la puntuación basada en reseñas de críticos de cine y otros factores. La puntuación de del 0 al 100
- **Rotten Tomatoes** es una plataforma que se especializa en críticas de cine. Tendremos dos columnas, una para críticos (critics) y otra de la audiencia (audience). La puntuación es sobre el 100%
- **Letterboxd** es una plataforma social donde los usuarios califican películas en una escala de 1 a 5.
- **CinemaScore** es un servicio que encarga encuestas a la audiencia que asiste al estreno de la película, obteniendo una calificación que varía de A+ (la mejor) a F (la peor).
- **Budget** es el presupuesto de la película en millones de dólares.
- **Domestic Gross** son ingresos obtenidos por la película solo en el mercado nacional (es decir, en el país de origen). Este valor también está en millones de dólares y muestra cuántos beneficios generó la película en su país de estreno.
- **Worldwide Gross** es la cantidad total de dinero recaudado a nivel mundial. Esto incluye las ganancias tanto a nivel nacional como internacional (otros países fuera del país de origen). También está en millones de dólares.

Limpiamos el CSV para que sea más fácil de manejar eliminando la columna index, dejamos la fecha en formato DD/MM/YY, quitamos paréntesis de los nombres de columnas, cambiamos los espacios por barra baja (_), cambiamos las comas de los números decimales por puntos y quitamos todas las comillas. Comprobamos que en el director_2 aparecen las columnas como nan, esto tendremos que cambiarlo por null para que Hbase no de error al importar el CSV

Subir CSV del Dataset

Una vez creada la máquina, vamos a subir el Dataset descargado del link mencionado arriba. **** Hay que tener en cuenta que cada vez que clones el EMR tendrás que cambiar el nombre de la máquina por el nuevo (Lo que pones después del hadoop@...) para poder subir los datos y realizar la conexión SSH****

```
scp -i ClavesHadoop.pem Marvel_Movies_Dataset.csv  
hadoop@ec2-3-238-50-162.compute-1.amazonaws.com:/home/hadoop
```

```
(base) iabd24@dm2ssd-H110M-S2H:~$ cd Descargas/
(base) iabd24@dm2ssd-H110M-S2H:~/Descargas$ scp -i ClavesHadoop.pem Marvel_Movie
s_Dataset.csv hadoop@ec2-3-238-50-162.compute-1.amazonaws.com:/home/hadoop
The authenticity of host 'ec2-3-238-50-162.compute-1.amazonaws.com (3.238.50.162
)' can't be established.
ED25519 key fingerprint is SHA256:9028ENFGwafgbtaHg5jYxT0I+urbvhZs5G1HqG0VdfQ.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-238-50-162.compute-1.amazonaws.com' (ED25519)
to the list of known hosts.
Marvel Movies Dataset.csv                                100% 3776    36.6KB/s   00:00
(base) iabd24@dm2ssd-H110M-S2H:~/Descargas$
```

```
ssh -i ClavesHadoop.pem hadoop@ec2-3-238-50-162.compute-1.amazonaws.com
```

```
(base) iabd24@dm2ssd-H110M-S2H:~/Descargas$ ssh -i ClavesHadoop.pem hadoop@ec2-3-238-50-162.compute-1.amazonaws.com

A newer release of "Amazon Linux" is available.
Version 2023.6.20241111:
Version 2023.6.20241121:
Run "/usr/bin/dnf check-release-update" for full release and version update info

#_
~\##### Amazon Linux 2023
~~~\#####\
~~~\###|
~~~\#/ https://aws.amazon.com/linux/amazon-linux-2023
~~~V~'-'>
~~~~
~~~.
~~~/_/m/'
```

Nos dirigimos al directorio donde hemos copiado el .csv

```
Last login: Wed Nov 27 18:51:32 2024

EEEEEEEEEEEEEEEEEEEE MMMMMMM      MMMMMMM RRRRRRRRRRRRRR
E::::::::::::::::::::E M::::::::M      M::::::::M R::::::::::::R
EE::::::::EEEEEEEE::::E M::::::::M      M::::::::M R::::RRRRR::::R
  E::::E      EEEEE M::::::::M      M::::::::M RR::::R      R::::R
  E::::E      M::::::::M:M      M::M::::M      R::R      R::::R
  E::::EEEEEEEEEE M::::M M::M M::M M::::M      R::RRRRR::::R
  E::::::::::::E M::::M M::M:M M::::M      R:::::::::RR
  E::::EEEEEEEEEE M::::M M::::M M::::M      R::RRRRR::::R
  E::::E      M::::M M::M M::::M      R::R      R::::R
  E::::E      EEEEE M::::M      MMM      M::::M      R::R      R::::R
EE::::::::EEEEEEEE::::E M::::M      M::::M      R::R      R::::R
E::::::::::::E M::::M      M::::M      RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM      MMMMMMM RRRRRRR      RRRRRR

[hadoop@ip-172-31-72-253 ~]$ cd /home/hadoop
[hadoop@ip-172-31-72-253 ~]$ ls
Marvel_Movies_Dataset.csv
[hadoop@ip-172-31-72-253 ~]$
```

Subir CSV al HDFS

Una vez subido a la máquina de AWS, ahora tenemos que subirlo al HDFS, para ello usamos el siguiente comando:

```
hdfs dfs -put /home/hadoop/Marvel_Movies_Dataset.csv
```

****Puedes escribir la ruta absoluta o la ruta relativa sin el /home/hadoop ya que se subirá a /user/hadoop****

Si queremos verificar que se ha subido correctamente usamos este comando:

```
hdfs dfs -ls
```

```
[hadoop@ip-172-31-77-249 ~]$ hdfs dfs -put /home/hadoop/Marvel_Movies_Dataset.csv
[hadoop@ip-172-31-77-249 ~]$ ^C
[hadoop@ip-172-31-77-249 ~]$ hdfs dfs -ls
Found 1 items
-rw-r--r--  1 hadoop hdfsadmin 3776 2024-11-28 16:20 Marvel_Movies_Dataset.csv
[hadoop@ip-172-31-77-249 ~]$
```

Manual de instancias CRUB HBase

Vamos a mencionar los comandos más usados en HBase.

start-hbase.sh → Iniciar el servidor

hbase shell → Acceder al shell

Operación	Comando	Descripción	Ejemplo
CREATE	<code>create</code>	Crea una tabla en HBase especificando las familias de columnas.	<code>create 'empleados', 'datos_personales', 'datos_laborales'</code>
INSERTAR	<code>put</code>	Inserta un valor en una fila y columna específica.	<code>put 'empleados', 'emp1', 'datos_personales:nombre', 'Juan Perez'</code>
LEER (fila específica)	<code>get</code>	Obtiene los datos de una fila completa o una columna específica.	<code>get 'empleados', 'emp1'</code>
LEER (toda la tabla)	<code>scan</code>	Escanea y muestra todas las filas de una tabla.	<code>scan 'empleados'</code>
ACTUALIZAR	<code>put</code>	Sobrescribe un valor existente en una columna.	<code>put 'empleados', 'emp1', 'datos_laborales:salario', '55000'</code>
ELIMINAR (valor específico)	<code>delete</code>	Elimina un valor de una columna específica.	<code>delete 'empleados', 'emp1', 'datos_laborales:puesto'</code>
ELIMINAR (toda la fila)	<code>deleteall</code>	Borra todos los datos asociados con una fila.	<code>deleteall 'empleados', 'emp1'</code>
DESACTIVAR TABLA	<code>disable</code>	Desactiva una tabla para operaciones.	<code>disable 'empleados'</code>
ELIMINAR TABLA	<code>drop</code>	Elimina una tabla previamente desactivada.	<code>drop 'empleados'</code>
DESCRIBIR TABLA	<code>describe</code>	Muestra el esquema de una tabla.	<code>describe 'empleados'</code>
CONTAR FILAS	<code>count</code>	Cuenta el número de filas en una tabla.	<code>count 'empleados'</code>

Manual de instancias CRUB Hive

hive → Iniciar shell

Operación	Comando	Descripción	Ejemplo
CREATE	<code>CREATE TABLE</code>	Crea una tabla especificando columnas, tipos de datos, y formato de almacenamiento.	<code>CREATE TABLE empleados (id INT, nombre STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';</code>
LOAD DATA	<code>LOAD DATA</code>	Carga datos desde un archivo local o HDFS a una tabla.	<code>LOAD DATA LOCAL INPATH '/ruta/empleados.csv' INTO TABLE empleados;</code>
SELECT	<code>SELECT</code>	Consulta datos de una tabla con opciones de filtrado, agrupación y orden.	<code>SELECT * FROM empleados WHERE salario > 50000;</code>
DESCRIBE	<code>DESCRIBE</code>	Muestra el esquema de una tabla (columnas y tipos de datos).	<code>DESCRIBE empleados;</code>
UPDATE	<code>UPDATE</code>	Actualiza valores en filas específicas de tablas transaccionales.	<code>UPDATE empleados_txn SET salario = 60000 WHERE id = 1;</code>
DELETE	<code>DELETE</code>	Elimina filas específicas de tablas transaccionales.	<code>DELETE FROM empleados_txn WHERE edad > 30;</code>
TRUNCATE	<code>TRUNCATE</code>	Vacía todos los datos de una tabla sin eliminarla.	<code>TRUNCATE TABLE empleados;</code>
DROP	<code>DROP TABLE</code>	Elimina una tabla y todos sus datos permanentemente.	<code>DROP TABLE empleados;</code>
SHOW DATABASES	<code>SHOW DATABASE S</code>	Lista todas las bases de datos disponibles.	<code>SHOW DATABASES;</code>

SHOW TABLES	SHOW TABLES	Lista todas las tablas de la base de datos actual.	SHOW TABLES;
INSERT OVERWRITE	INSERT OVERWRITE	Exporta los resultados de una consulta a un archivo o directorio.	INSERT OVERWRITE LOCAL DIRECTORY '/ruta/salida' SELECT * FROM empleados;
COUNT	SELECT COUNT(*)	Cuenta el número total de filas en una tabla.	SELECT COUNT(*) FROM empleados;

HIVE

Creación de la tabla en Hive con HUE

Accedemos a la interfaz de HUE, para ello vamos al EMR → Aplicaciones → Tonalidad

[Amazon EMR](#) > [EMR en EC2: Clústeres](#) > [Proyecto_hadoop](#)

Proyecto_hadoop
Se ha actualizado hace menos de un minuto
Terminar

▼ Resumen

Información del clúster

ID del clúster
j-DQL13TJS7P6B

Configuración del clúster
Grupos de Instancias

Capacidad
1 Primary (Principal) | 3 Principal | 0 Tarea

Aplicaciones

Versión de Amazon EMR
emr-7.5.0

Aplicaciones instaladas
HBase 2.5.10, Hadoop 3.4.0, Hive 3.1.3, Hue 4.11.0

Administración de clústeres

Destino del registro en Amazon S3
[aws-logs-992382578743-us-east-1/elasticmapreduce](#)

IU de aplicación persistente
[Servidor de línea de tiempo de YARN](#)
[UI de Tez](#)

DNS público del nodo principal
[ec2-34-239-173-184.compute-1.amazonaws.com](#)
[Conectarse al nodo principal mediante SSH](#)
[Conectarse al nodo principal mediante SSM](#)

Propiedades
Acciones de arranque
Instancias (hardware)
Pasos
Aplicaciones
Configuraciones

IU de la aplicación activas

Estas IU de aplicaciones en clúster están disponibles sin el túnel de SSH.

IU de la aplicación [\[?\]](#)

No hay ninguna IU de la aplicación activa

No hay ninguna IU de la aplicación activa que mostrar

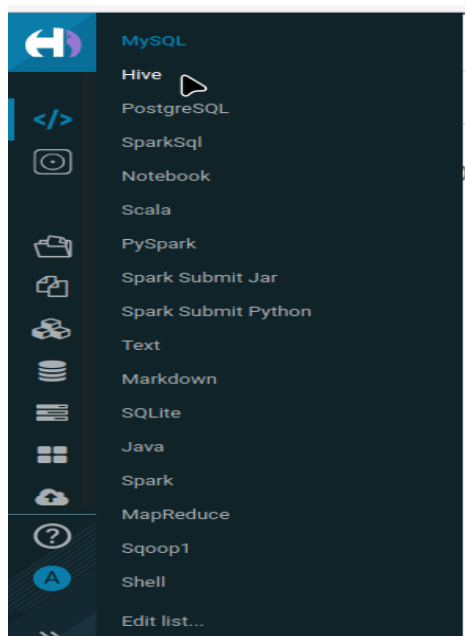
IU de la aplicación en el nodo principal

Estas requieren que el túnel de SSH esté habilitado.

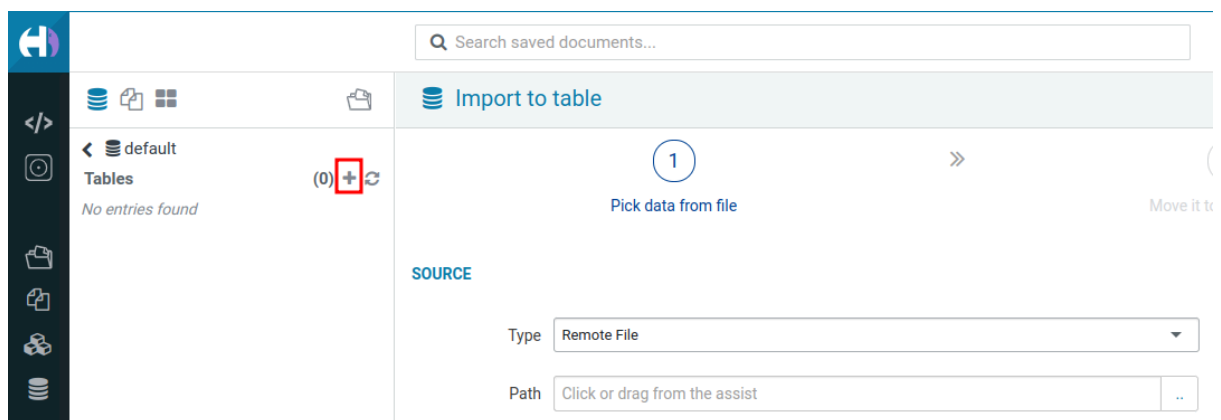
[Habilitar una conexión SSH](#)

Aplicación	URL de la IU [?]
Administrador de recursos	http://ec2-34-239-173-184.compute-1.amazonaws.com:8088/
HBase	http://ec2-34-239-173-184.compute-1.amazonaws.com:16010/
Nodo del nombre de HDFS	http://ec2-34-239-173-184.compute-1.amazonaws.com:9870/
Tonalidad	http://ec2-34-239-173-184.compute-1.amazonaws.com:8888/

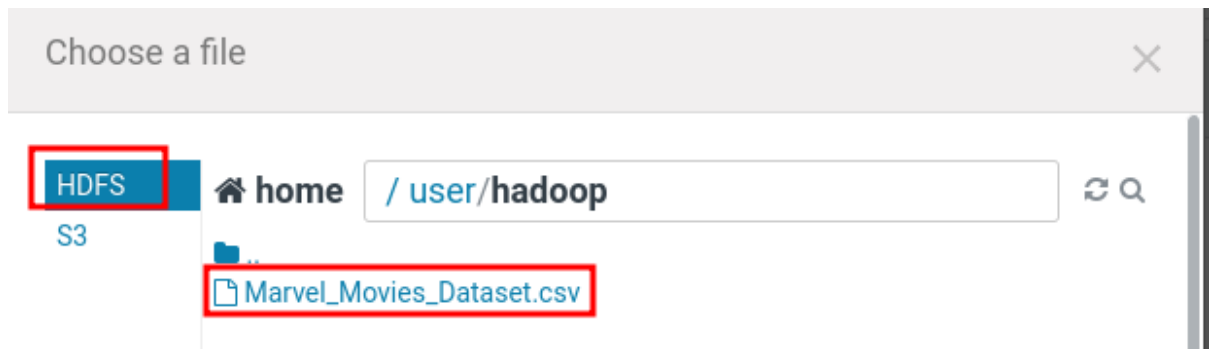
Quando accedemos a HUE tenemos que crear un usuario.
 Yo le pondré siempre admin → Admin.10
 Ahora nos dirigimos a la base de datos Hive.



Y le damos a crear tabla



Ahora seleccionamos la ruta de donde está el csv, que debería estar en el hdfs.



Ahora seleccionamos el tipo de archivo que es, los separadores que utiliza y le damos a next.

FORMAT

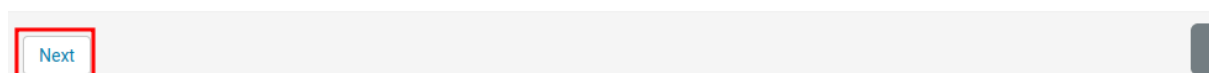
File Type: CSV File

Field Separator: Comma (,) Record Separator: New line Quote Character: Double Quote

☒ Has Header

PREVIEW

Title	Director ₁	Director ₂	ReleaseDate	IMDb	IMDB_Metascor	RottenTomatoes_Cr
Iron Man	Jon Favreau	nan	02/05/08	7,90	79	94
The Incredible Hulk	Louis Leterrier	nan	13/06/08	6,60	61	68
Iron Man 2	Jon Favreau	nan	07/05/10	6,90	57	72
Thor	Kenneth Branagh	nan	06/05/11	7,00	57	77
Captain America: The...	Joe Johnston	nan	22/07/11	6,90	66	80



Ahora seleccionamos como se va a llamar la tabla y el formato

Import to table

Pick data from file /user/hadoop/Marvel_Movies_Dataset.csv

Move it to table movies_marvel

DESTINATION

Name: movies_marvel

PROPERTIES

Format: Text

Extras

Partitions: + Add partition

Más abajo, podrás modificar el nombre y el formato de cada columna. (No puede haber espacios o guiones en los nombres de las columnas)

FIELDS

Name	Type	Value
Title	string	Iron Man
Director_1	string	Jon Favreau
Director_2	string	nan
ReleaseDate	date	02/05/08
IMDb	decimal	7,90
IMDB_Metascore	int	61
RottenTomatoes_Critics	int	68
RottenTomatoes_Audience	int	69

Name	Letterboxd	Type	decimal	4	2
	3,7				2,5
Name	CinemaScore	Type	string		A
					A-
Name	Budget	Type	int		140
					150
Name	Domestic_Gross	Type	decimal	4	2
	319				134,8
Name	Worldwide_Gross	Type	decimal	4	2
	585,8				265,5

Back Submit

Comprobamos que se ha creado correctamente

The screenshot shows the Hive console interface. On the left, the 'Tables' list under the 'default' database includes 'marvel_movies', which is highlighted with a red box. The table's schema is listed: title (string), director1 (string), director2 (string), releasedate (string), imdb (string), imdb_metascore (bigint), rottentomatoes_critics (bigint), rottentomatoes_audience (bigint), letterboxd (string), cinemascore (string), budget (bigint), domestic_gross (string), and worldwide_gross (string). The main console area shows a query: `SELECT * FROM 'default'. 'marvel_movies' LIMIT 100;` with a status of '0.19s default'. Below the query, the execution log shows 'INFO : Completed executing command' and 'INFO : OK'. At the bottom, the 'Results (34)' tab is active, displaying a table with columns 'marvel_movies.title', 'marvel_movies.director1', and 'marvel'. The results show three rows: 'Iron Man' by Jon Favreau, 'The Incredible Hulk' by Louis Leterrier, and 'Iron Man 2' by Jon Favreau.

	marvel_movies.title	marvel_movies.director1	marvel
1	Iron Man	Jon Favreau	nan
2	The Incredible Hulk	Louis Leterrier	nan
3	Iron Man 2	Jon Favreau	nan

De esta forma da un error de que no existe la base de datos y te crea una base de datos temporal.

Si quieres crearla sin que de errores puedes hacer de la siguiente forma:

Crea la tabla con la consola de Hive.

The screenshot shows the Hive console with a 'CREATE TABLE' statement highlighted in a red box. The statement is: `CREATE TABLE marvel_movies (title STRING, director_1 STRING, director_2 STRING, releasedate STRING, imdb FLOAT, imdb_metascore INT, rottentomatoes_critics FLOAT, rottentomatoes_audience FLOAT, letterboxd FLOAT, cinemascore STRING, budget FLOAT, domestic_gross FLOAT, worldwide_gross FLOAT);` The console shows the query was executed successfully in 0.17s. The left sidebar shows the 'marvel_movies' table has been created in the 'default' database.

Este es el código que tienes que poner:

```
CREATE TABLE marvel_movies (  
  title STRING,  
  director_1 STRING,  
  director_2 STRING,  
  releasedate STRING,  
  imdb FLOAT,  
  imdb_metascore INT,
```

```

    rottentomatoes_critics FLOAT,
    rottentomatoes_audience FLOAT,
    letterboxd FLOAT,
    cinemascor STRING,
    budget FLOAT,
    domestic_gross FLOAT,
    worldwide_gross FLOAT
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;

```

En el CSV tendrás que borrar el nombre de las columnas para importarlo ya que si están el nombre de las columnas dará error al importarlo.

```

LOAD DATA INPATH '/user/hadoop/Marvel_Movies_Dataset.csv' INTO TABLE
marvel_movies

```

Comprobamos que se ha creado correctamente y que se han insertado bien todas las filas.

The screenshot shows the Hive web interface. On the left, the 'Tables' section lists the 'marvel_movies' table with its schema: title (string), director_1 (string), director_2 (string), releasedate (string), imdb (float), imdb_metascore (int), rottentomatoes_critics (float), rottentomatoes_audience (float), letterboxd (float), cinemascor (string), budget (float), domestic_gross (float), and worldwide_gross (float). The main area shows a query 'SELECT * FROM marvel_movies' executed successfully, with results displayed in a table format. The results show 7 rows of movie data, including titles like 'Iron Man', 'The Incredible Hulk', 'Iron Man 2', 'Thor', 'Captain America: The First Avenger', 'The Avengers', and 'Iron Man 3'.

	marvel_movies.title	marvel_movies.director_1	marvel_movies.imdb
1	Iron Man	Jon Favreau	nan
2	The Incredible Hulk	Louis Leterrier	nan
3	Iron Man 2	Jon Favreau	nan
4	Thor	Kenneth Branagh	nan
5	Captain America: The First Avenger	Joe Johnston	nan
6	The Avengers	Joss Whedon	nan
7	Iron Man 3	Shane Black	nan

Consultas en Hive

Puede ir al apartado [Manual de instancias CRUB Hive](#) para ver como realizar las consultas.

*SELECT * FROM marvel_movies WHERE budget > 300*

1 | SELECT * FROM marvel_movies WHERE budget > 300 |

▶

📖

Query: SELECT * FROM marvel_movies WHERE budget > 300

INFO : Completed executing command(queryId=hive_20241204175855_1426f9ca-1539-4288-900a-9da5846966ad); Time taken: 0.0 seconds

INFO : OK

INFO : Concurrency mode is disabled, not creating a lock manager

Query History

Saved Queries

Results (5)

📊

📈

📖

📄

	marvel_movies.title	marvel_movies.director_1	marvel_movies.budget
1	Avengers: Age of Ultron	Joss Whedon	nan
2	Avengers: Infinity War	Anthony Russo	Joe Russo
3	Avengers: Endgame	Anthony Russo	Joe Russo
4	Doctor Strange in the Multiverse of Madness	Sam Raimi	nan
5	Ant-Man and the Wasp: Quantumania	Peyton Reed	nan

Query History

Saved Queries

Results (5)

📊

📈

📖

📄

	boxd	marvel_movies.cinemascore	marvel_movies.budget	marvel_movies
1	A		365	459
2	A		400	678.8
3	A+		400	858.4
4	B+		350	411.3
5	B		330	214.5

15

```
SELECT count(*) FROM marvel_movies where cinemascore = "A"
```

```
1 | SELECT count(*) FROM marvel_movies where cinemascore = "A"
```

9.16s default ▾ ⚙️

```
1 | SELECT count(*) FROM marvel_movies where cinemascroe = "A"
```

```
INFO : Compiling command(queryId=hive_20241204181610_5e4dd5a6-d316-4daa-b409-4863f623b7f5): SELECT count(*) FROM marvel_movies where cinemascore = "A"
INFO : Concurrency mode is disabled, not creating a lock manager
INFO : Semantic Analysis Completed (retrial = false)
```

Query History

Saved Queries

Results (1)

_c0

1	19
---	----

SHOW DATABASES

```
1 SHOW DATABASES
```

```
INFO : Starting task [stage=0.000] in serial mode
INFO : Completed executing command(queryId=hive_20241204182635_9d677f15232f); Time taken: 0.002 seconds
INFO : OK
INFO : Concurrency mode is disabled, not creating a lock manage
```

Query History

Saved Queries

Results (1)

database_name

1	default
---	---------

SHOW TABLES

```
1 SHOW TABLES
```

```
INFO : Starting task [stage=0.DDL] in serial mode
INFO : Completed executing command(queryId=hive_202412041830
60ebe4ad298a); Time taken: 0.013 seconds
INFO : OK
INFO : Concurrency mode is disabled, not creating a lock man
```

Query History

Saved Queries

Results (1)

tab_name



	tab_name
1	marvel_movies

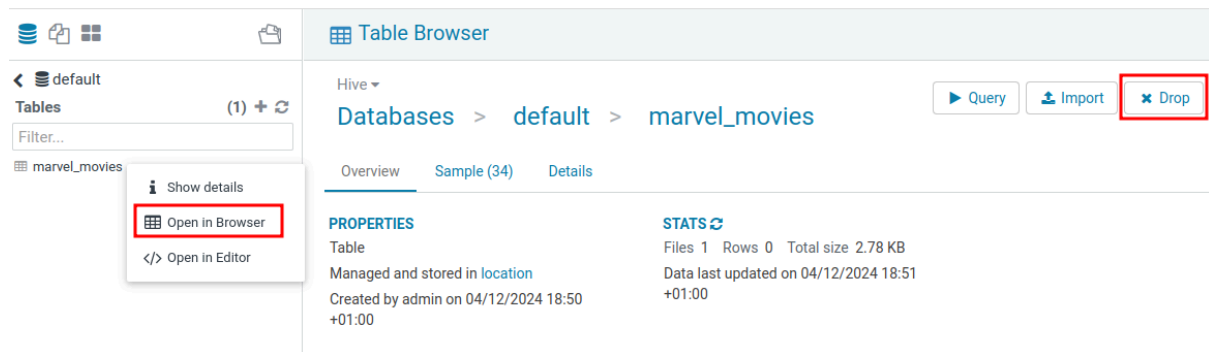
DESCRIBE marvel_movies

```
1 DESCRIBE marvel_movies
```

	col_name	data_type	comment
1	title	string	
2	director_1	string	
3	director_2	string	
4	releasedate	string	
5	imdb	float	
6	imdb_metascore	int	
7	rottentomatoes_critics	float	
8	rottentomatoes_audience	float	
9	letterboxd	float	
10	cinemascore	string	
11	budget	float	
12	domestic_gross	float	
13	worldwide_gross	float	

Los comandos delete, update y truncate no te deja realizarlos porque el formato de la tabla no es ORC y para cambiar que puedas usar ese formato tiene que cambiar la configuración del archivo hive-site.xml

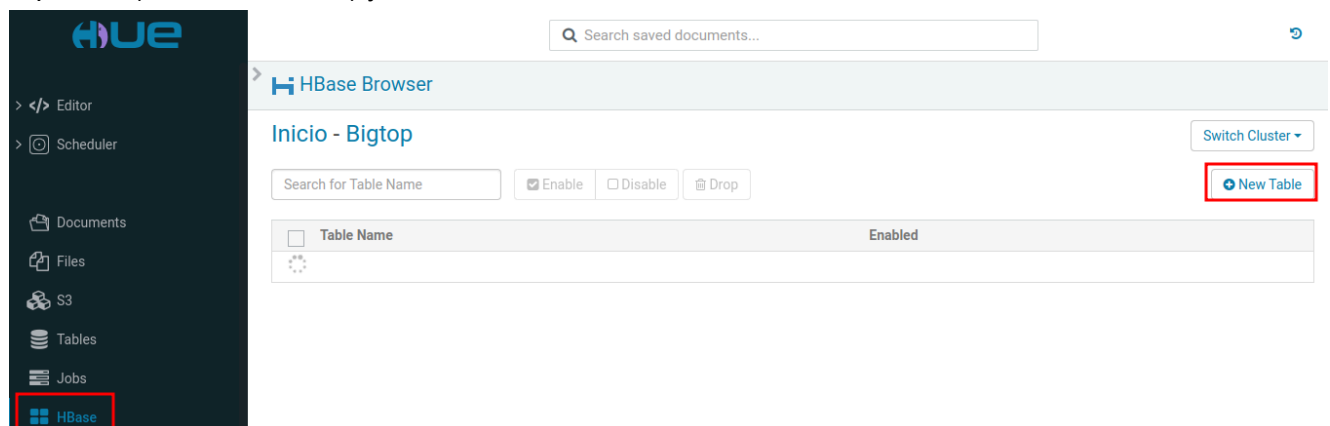
Para borrar la tabla tendrás que hacer click derecho en la tabla, seleccionar open in Browser y a la derecha aparece Drop.



Hbase

Creación de la tabla en Hbase con HUE

En la interfaz de HUE, seleccionamos Hbase que es el último icono de la barra de la izquierda (los 4 cuadrados) y seleccionamos New table a la derecha del todo.



Seleccionamos el nombre de la tabla y metemos el nombre de los grupos en los que queremos agrupar las columnas. Vamos a elegir como key_row el title.

- Info
 - director_1
 - director_2
 - releasedate
- Ratings
 - IMBd
 - IMBd_Metascore
 - RottenTomatoes_critics
 - Letterboxd
 - CinemaScore
- Financial
 - Budge
 - DomesticGross
 - WorldwideGross

Create New Table

×

Table Name:

Column Families:

✕ Info

+ Add a column property

✕ Ratings

+ Add a column property

✕ Financial

+ Add a column property

+ Add an additional column family

Cancel

Submit

Home - Bigtop / marvel_movies

row_key, row_prefix* +scan_len [col1, family:col2, fam3:, col_prefix* +3, fam: col2 to c

info ratings-

Filter Columns/Families

All Sort By ASC

No rows to display.

Fetched 10 entries starting from null in 0.223 seconds.

Drop Rows Bulk Upload New Row

Probamos a importar el CSV desde el HUE pero parece que no funciona así que lo haremos a través de la terminal. Nos conectamos por SSH al cluster de AWS como explicamos en el punto de [Subir CSV del dataset](#)

Para importar el CSV desde el HDFS usaremos el siguiente comando:

```
sudo hbase org.apache.hadoop.hbase.mapreduce.ImportTsv
-Dimporttsv.columns=HBASE_ROW_KEY,Info:title,Info:director_1,Info:director_2,Info:releasedate,Ratings:IMBd,Ratings:IMBd_Metacore,Ratings:RottenTomatoes_critics,Ratings:Letterboxd,Ratings:CinemaScore,Financial:Budge,Financial:DomesticGross,Financial:WorldwideGross -Dimporttsv.separator=',' marvel_movies
hdfs:///user/hadoop/Marvel_Movies_Dataset.csv
```

Comprobamos que se han importado bien los datos

Home - Bigtop / marvel_movies

row_key, row_prefix* +scan_len [col1, family:col2, fam3:, col_prefix* +3, fam: col2 to c

Financial: Info: Ratings-

Filter Columns/Families

All Sort By ASC

Financial: Budge	Financial: DomesticGross	Financial: WorldwideGross	Info: director_1	Info: director_2	Info: releasedate	Info: title	Ratings: CinemaSco
169	180.2	519.3	nan	12/6/2024, 2:28:59 PM	7.20	Peyton Reed	A
Ant-Man and the Wasp							
12/6/2024, 2:28:59 PM	216.6	622.7	nan	06/07/18	7.00	Peyton Reed	A-
Avengers: Age of Ultron							
365	459	1405	nan	01/05/15	7.30	Joss Whedon	A
Avengers: Infinity War							
400	678.8	2052	Joe Russo	27/04/18	12/6/2024, 2:28:59 PM	Anthony Russo	A

Consultas HBase

Puede ir al apartado [Manual de instancias CRUB HBase](#) para ver como realizar las consultas.

```
scan 'marvel_movies', {  
  COLUMNS => ['Financial:Budge'],  
  FILTER => "SingleColumnValueFilter('Financial', 'Budge', >=, 'binary:400')"  
}
```

```
hbase:016:0> scan 'marvel_movies', {  
hbase:017:1*   COLUMNS => ['Financial:Budge'],  
hbase:018:1*   FILTER => "SingleColumnValueFilter('Financial', 'Budge', >=, 'binary:400')"  
}  
ROW          COLUMN+CELL  
Avengers: Infinity W column=Financial:Budge, timestamp=2024-12-06T22:51:22.715,  
ar          value=400  
1 row(s)
```



```
count 'marvel_movies', {COLUMN => 'Info:title'}
```

```
hbase:023:0> count 'marvel_movies', {COLUMN => 'Info:title'}  
30 row(s)  
Took 0.0183 seconds  
=> 30  
hbase:024:0>
```

```
put 'marvel_movies', 'Black Panther', 'Info:director_2', 'Guillermo'
```

```
hbase:020:0> put 'marvel_movies', 'Black Panther', 'Info:director_2', 'Guillermo'  
Took 0.0542 seconds  
hbase:021:0>
```

Black Panther

Financial: Info: Ratings:  

Financial: Budge	Financial: DomesticGross	Financial: WorldwideGross	Info: director_1	Info: director_2
200	700.4	1350	null	Guillermo

describe 'marvel_movies'

```
hbase:021:0> describe 'marvel_movies'
Table marvel_movies is ENABLED
marvel_movies, {TABLE_ATTRIBUTES => {METADATA => {'hbase.store.file-tracker.impl' => 'DEFAULT'}}}
COLUMN FAMILIES DESCRIPTION
{NAME => 'Financial', INDEX_BLOCK_ENCODING => 'NONE', VERSIONS => '3', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', MIN_VERSIONS => '0', REPLICATION_SCOPE => '0', BLOOMFILTER => 'NONE', IN_MEMORY => 'false', COMPRESSION => 'NONE', BLOCKCACHE => 'false', BLOCKSIZE => '65536 B (64KB)'}
{NAME => 'Info', INDEX_BLOCK_ENCODING => 'NONE', VERSIONS => '3', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', MIN_VERSIONS => '0', REPLICATION_SCOPE => '0', BLOOMFILTER => 'NONE', IN_MEMORY => 'false', COMPRESSION => 'NONE', BLOCKCACHE => 'false', BLOCKSIZE => '65536 B (64KB)'}
{NAME => 'Ratings', INDEX_BLOCK_ENCODING => 'NONE', VERSIONS => '3', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', MIN_VERSIONS => '0', REPLICATION_SCOPE => '0', BLOOMFILTER => 'NONE', IN_MEMORY => 'false', COMPRESSION => 'NONE', BLOCKCACHE => 'false', BLOCKSIZE => '65536 B (64KB)'}
3 row(s)
Quota is disabled
Took 0.1129 seconds
hbase:022:0>
```

deleteall 'marvel_movies', 'Black Panther'

```
hbase:022:0> deleteall 'marvel_movies', 'Black Panther'
Took 0.0133 seconds
hbase:023:0>
```

Black Panther					
Financial: Info: Ratings: Filter Columns/Families					
Black Panther: Wakanda Forever					
Financial: Budget	Financial: DomesticGross	Financial: WorldwideGross	Info: director_1	Info: director_2	Info: releasedate
250	453.8	859.2	null	11/11/22	6.70

disable 'marvel_movies'

```
hbase:024:0> disable 'marvel_movies'
Took 0.6657 seconds
hbase:025:0>
```

Search saved documents...

Api Error: org.apache.hadoop.hbase.TableNotEnabledException: marvel_movies is disabled.

HBase Browser

Home - Bigtop / marvel_movies

Switch Cluster

row_key, row_prefix" + scan_len [col1, family:col2, fam3:, col_prefix" +3, fam: col2 to c

Financial: Info: Ratings: Filter Columns/Families

All Sort By ASC

drop 'marvel_movies'

```
hbase:025:0> drop 'marvel_movies'  
Took 0.3318 seconds  
hbase:026:0>
```

Home - Bigtop

☒ Enable☐ Disable

Table Name

No data available in table