Guía (para consultar la explicación paso a paso):

https://medium.com/@dataakkadian/how-to-install-and-running-cloudera-docker-container-on-ubuntu-b7c77f147e03

1. Desinstalar cualquier versión antigua de Docker

```
$ sudo apt-get remove docker docker-engine docker.io
```

2. Actualizar lista de paquetes, activar uso de https en el repositorio y agregar llave de Docker a este..

```
$ sudo apt-get update
```

```
$ sudo apt-get install \
apt-transport-https \
ca-certificates \
curl \
software-properties-common
```

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

3. Setear el repositorio de Docker apuntando al estable

```
$ sudo add-apt-repository \
"deb [arch=amd64] https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) \
stable"
```

4. Instalar Docker

```
$ sudo apt-get update
```

```
$ sudo apt-get install docker-ce=5:18.09.0~3-0~ubuntu-xenial
```

5. Importar la imagen de cloudera-quickstart a Docker (Demora un poco porque son más de 4gb de descarga e instalación)

```
$ sudo docker pull cloudera/quickstart:latest
```

6. Se debe revisar la memoria de swap dado que normalmente viene en cero la máquina y Cloudera solicita espacio. Entonces para crear el mismo pueden seguir paso a paso la página: https://www.stackscale.com/es/blog/anadir-memoria-swap-servidor-ubuntu/, sólo dos cambios:
    a. Utilizar el que indica como "2 Método Rápido"
    b. Configurar el swap a 12G (**sudo fallocate –l 12G /swapfile)**, esto porque Cloudera pide 8G)

    Lo demás igual a la página

7. Ejecutar el contenedor de cloudera-quickstart (Esto está en modo interactivo, por lo tanto la consola muestra TODO lo que está ejecutando, paso a paso). En este punto de la página he aumentado la memoria y le coloque el nombre el clúster

```
$ sudo docker run --name quickstart.cloudera -m 8G --memory-reservation 4G --memory-swap 8G --hostname=quickstart.cloudera --privileged=true -t -i -v $(pwd):/zaid --publish-all=true -p8888 -p8088 cloudera/quickstart /usr/bin/docker-quickstart
```

El prompt debe quedar así:

```
[root@quickstart /]#
```

8. Para verificar que todo corre y los puertos que se levantaron para HUE y YARN, se debe levantar otra terminal (SIN CERRAR LA ANTERIOR) y conectar SSH (o salir de cloudera con control-P y control-Q), luego correr este comando:
    a. Listar los contenedores que están corriendo y ver la columna NAME
    `$ sudo docker ps`
    b. Obtener los puertos para la Web con el comando:
    `$ sudo docker inspect <nombre del contenedor>`

Acá se verán los puertos externos que es donde configura el hue y yarn, entonces:

I. Hue, en mi caso es el puerto 49153 (que aparece en la imagen en **"HostPort" del puerto 8888)**
II. yarn, en mi caso es el puerto 49154 (que aparece en la imagen en **"HostPort" del puerto 8088).**
III. **Importante es que se revise cual es "IPAddress" que está un poco mas abajo que indica la IP interna que trabaja Cloudera y que deben colocar en los sqoop. En mi caso es 172.17.0.2.**

Nota: Si lo anterior, se realizó en otra terminal, volver a la terminal con Cloudera, si se realizó en la misma terminal deben ejecutar el comando sudo Docker attach <nombre del contenedor> que volverá a Cloudera.

```
[root@quickstart /]#
```

Nota: con lo anterior funciona ya el sistema muy bien.

Parte 2: Ejercicios:

1. Crear datos en MySql

```
$ mysql -uroot -pcloudera
Mysql> Create database midbmy;
```

```
Mysql> CREATE TABLE mitablai (id MEDIUMINT NOT NULL
AUTO_INCREMENT, nombre CHAR(30) NOT NULL, genero CHAR(30), year
char(4), veces_vista INTEGER(10), PRIMARY KEY (id));

Mysql> CREATE TABLE mitablae (id MEDIUMINT NOT NULL
AUTO_INCREMENT, nombre CHAR(30) NOT NULL, genero CHAR(30), year
char(4), veces_vista INTEGER(10), PRIMARY KEY (id));

Mysql> INSERT INTO mitablai (nombre, genero, year, veces_vista)
VALUES ("Star Wars", "Acción", "2019", 12000000), (“IT",
"Terror", "2019", 530000), ("Dark Fenix", "Acción", "2018",
1000000), ("Avenger", "Acción", "2019", 9500000), ("Toy Story
4", "Infantil", "2019", 5600000), ("Increibles 2", "Infantil",
"2018", 550000), ("Titanic", "Drama", "1997", 10500000);

Mysql> grant all privileges on *.* to 'root'@'localhost'
IDENTIFIED BY 'cloudera' WITH GRANT OPTION;

Mysql> grant all privileges on *.* to 'root' IDENTIFIED BY
'cloudera' WITH GRANT OPTION;

Mysql> grant all privileges on *.* to 'cloudera' IDENTIFIED BY
'cloudera' WITH GRANT OPTION;

Mysql> exit;
```

2. Probar SQOOP:
    a. Antes de ejecutar sqoop, setear ACCUMULO_HOME dentro del contenedor:

```
# mkdir /var/lib/accumulo
# ACCUMULO_HOME='/var/lib/accumulo'
# export ACCUMULO_HOME
```

    b. Pruebas con Sqoop. Ingesta. Estos comandos funcionan, por favor probar (verificar que el "**IPAddress" es 172.17.0.2, sino cambiar en la instrucción**)

b.1.- Sin target

```
# sqoop import \
  -m 1 \
  --connect jdbc:mysql://172.17.0.2:3306/midbmy \
  --table=mitabla \
  --username=root \
  --password=cloudera \
  --fields-terminated-by=',' \
  --lines-terminated-by '\n'
```

b.2.- Con target

```
# sqoop import \
-m 1 \
--connect jdbc:mysql://172.17.0.2:3306/midbmy\
--table=mitabla\
--username=root \
--password=cloudera \
--target-dir=/prueba \
--fields-terminated-by=',' \
--lines-terminated-by '\n'
```

b.3.- Hacia HIVE (Primero crear la BD mibdduochive en HIVE, mediante comando pueden entrar a Hive sólo con la instrucción hive dentro del clúster de Cloudera o entrar por hue (create database mibdhive;))

```
# sqoop import \
-m 1 \
--connect jdbc:mysql://172.17.0.2:3306/midbmy \
--table=mitablamy \
--username=root \
--password=cloudera \
--target-dir=/prueba_hive \
--where "veces_vista > 1000000" \
--compress \
--compression-codec org.apache.hadoop.io.compress.SnappyCodec \
--hive-import \
--hive-database mibdhive \
--create-hive-table \
--hive-table mitabla_hive
```

b.4.- Pueden mirar en hue la tabla creada en hive y trabajar con ella.

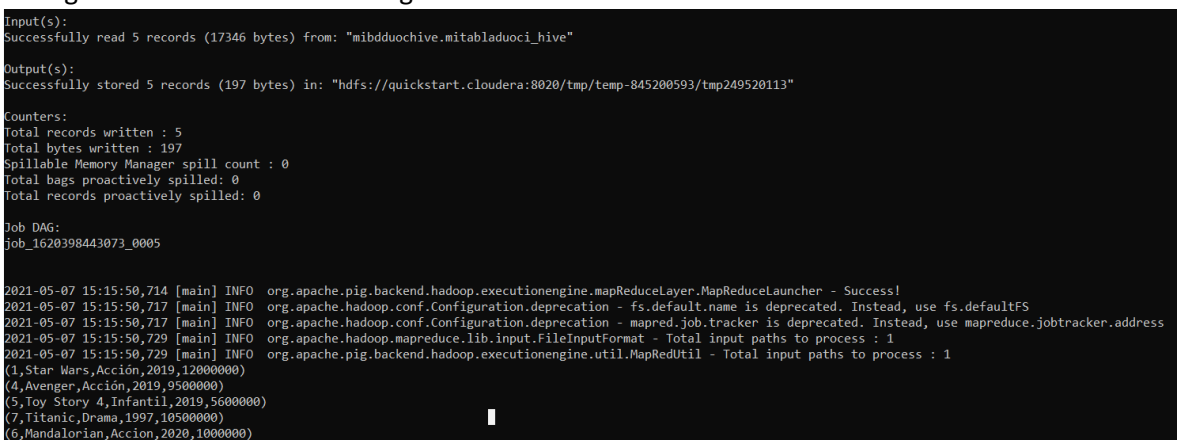
b.5.- Como adicional pueden hacer que se vea la tabla en impala, para ello deben:

I. Ingresar a impala por comando con impala-shell o por hue seleccionando impala.
II. Dentro de impala ejecutar "**invalidate metadata;",** si se desea solamente una tabla podemos realizar **invalidate metadata <bd_hive>.<tabla_hive>**
III. Dentro de impala pueden trabajar ingresando datos y lo que ingresen los pueden ver por hive. En mi caso ingresé un dato más a la tabla con:
insert into mitabla_hive values (6, 'Mandalorian', 'Accion', '2020', 1000000)

b.6.- Como otro adicional podrían ver la data de hive en pig, de la siguiente manera:

I. Ingresar a pig por comando desde el prompt
[root@quickstart /]# pig -useHCatalog;
II. Dentro de pig ejecutar:
grunt> Tabla_pig = LOAD 'mibdhive.mitabla_hive' USING org.apache.hive.hcatalog.pig.HCatLoader();
(donde: 'mibdhive.mitabla_hive' es <bd_hive>.<tabla_hive>)
grunt> dump Tabla_pig

III. Entregará un resultado como el siguiente:



```
Input(s):
Successfully read 5 records (17346 bytes) from: "mibdduochive.mitabladuoci_hive"

Output(s):
Successfully stored 5 records (197 bytes) in: "hdfs://quickstart.cloudera:8020/tmp/temp-845200593/tmp249520113"

Counters:
Total records written : 5
Total bytes written : 197
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_1620398443073_0005


2021-05-07 15:15:50,714 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2021-05-07 15:15:50,717 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
2021-05-07 15:15:50,717 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2021-05-07 15:15:50,729 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2021-05-07 15:15:50,729 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(1,Star Wars,Acción,2019,12000000)
(4,Avenger,Acción,2019,9500000)
(5,Toy Story 4,Infantil,2019,5600000)
(7,Titanic,Drama,1997,10500000)
(6,Mandalorian,Accion,2020,1000000)
```

b.7.- ya realizadas todas las acciones para ver los datos y mover los mismos podemos volver con algún dato adicional a mysql con el sqoop export, entonces:

I. Ingresar a hive ya sea por consola o por hue y ejecutar el comando:
describe formatted mitabla_hive;
Que muestra donde se encuentran los datos de hive en el clúster (importante ver "Location")

7	veces_vista	int	
8		NULL	NULL
9	# Detailed Table Information	NULL	NULL
10	Database:	mibdduochive	NULL
11	Owner:	root	NULL
12	CreateTime:	Fri May 07 15:02:14 UTC 2021	NULL
13	LastAccessTime:	UNKNOWN	NULL
14	Protect Mode:	None	NULL
15	Retention:	0	NULL
16	Location:	hdfs://quickstart.cloudera:8020/user/hive/warehouse/mibdduochive.db/mitabladuoci_hive	NULL
17	Table Type:	MANAGED_TABLE	NULL
18	Table Parameters:	NULL	NULL

Que en hdfs es la ruta: /user/hive/warehouse/mibdhive.db/mitabla_hive

II. Con ello ya podemos realizar el sqoop de export

```
sqoop export \
-m 1 \
--connect jdbc:mysql://localhost/midbmy \
--username=root \
--password=cloudera \
--table=mitablae \
--export-dir=/user/hive/warehouse/mibdduochive.db/mitablai_hive \
--input-fields-terminated-by '\0001'
```

III. Y verificar en mysql que se hayan ingresado los datos a la tabla que creamos inicialmente llamada **mitablae.** Entonces si se fijan se exportó la tabla con un dato más (que fue el ingresado en impala)

```
mysql> select * from mitabladuoce;
+----+-------------+----------+------+-------------+
| id | nombre      | genero   | year | veces_vista |
+----+-------------+----------+------+-------------+
|  1 | Star Wars   | Acción   | 2019 |    12000000 |
|  4 | Avenger     | Acción   | 2019 |     9500000 |
|  5 | Toy Story 4 | Infantil | 2019 |     5600000 |
|  6 | Mandalorian | Accion   | 2020 |     1000000 |
|  7 | Titanic     | Drama    | 1997 |    10500000 |
+----+-------------+----------+------+-------------+
5 rows in set (0.00 sec)
```

3. Probar PIG: (En el clúster de Cloudera se ingresa con el comando pig o en pueden hacerlo por hue)

   a. Crear el archivo "/tmp/book.txt" con texto dentro:

   ```
   # vi /tmp/book.txt

   Contenido:

   "Hace mucho mucho tiempo

   En una galaxia muy muy lejana

   Episodio X: El ascenso de los skywalker"
   ```

   b. Copiar archivo a HDFS usando put

   c. Ejecutar este script de PIG para contar las palabras (los comandos se ejecutan línea por línea):

```
# pig
Pig> input_lines = LOAD '/tmp/book.txt' AS (line:chararray);

-- extraer palabras de cada línea y ponerlas en un "saco" de
Pig,
-- después desanidar el saco para crear una palabra por línea
words = FOREACH input_lines GENERATE FLATTEN(TOKENIZE(line))
AS word;

-- filtrar las palabras que tengan sólo espacio en blanco
filtered_words = FILTER words BY word MATCHES '\\w+';
```

```
-- agrupar por palabra
word_groups = GROUP filtered_words BY word;

-- contar los valores en cada grupo
word_count = FOREACH word_groups GENERATE
COUNT(filtered_words) AS count, group AS word;

-- ordenar las palabras por conteo
ordered_word_count = ORDER word_count BY count DESC;

STORE ordered_word_count INTO '/tmp/book-word-count.txt';
```

d. Revisar salida de Pig con `dump ordered_word_count`