APACHE IMPALA - LAB

Group Members

Taha Rushain – 05811
Muhammad Siraj - 10255
Maria Zafar – 19842
M Sarmad Ali - 19828

Table of Contents

Overview 2

Selling points 2

Components of a Impala Architecture 2

Demo 3

Installation 3

Projec Setup 6

Execution 7

Apache Impala – Lab

Overview

To overcome the slowness of Hive Queries, Cloudera offers a separate tool and that tool is what we call Impala. Impala queries run very faster than Hive Queries even after they are more or less same as Hive Queries. Impala is not built on MapReduce, Impala has its own execution engine.

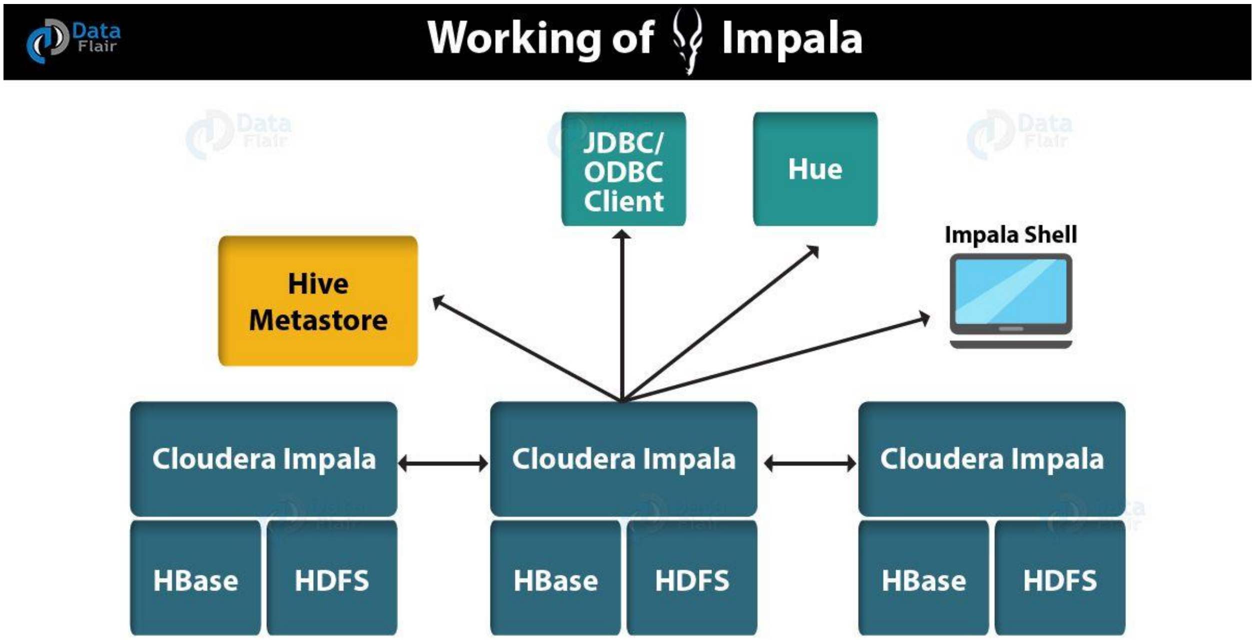
Selling points

- High-performance, low-latency SQL queries
- To share databases and tables between both Impala and hive it integrates very well with the Hive Metastore.
- It is Compatible with HiveQL Syntax
- We can easily integrate with Hbase database system and Amazon Simple Storage System.

We can perform interactive, ad-hoc and batch queries together in the Hadoop system, by using Impala's MPP (M-P-P) style execution along with other Hadoop processing MapReduce frameworks.

For querying analytic data it offers new possibilities. In addition, to query this type of data we can use exploratory data analysis and data discovery techniques.

Components of a Impala Architecture

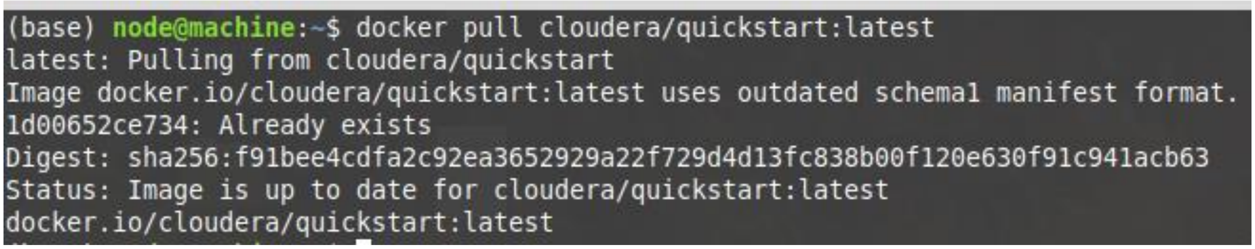


Demo

Installation

Step 1: Install Cloudera QuickStart Container using the following command

docker pull cloudera/quickstart:latest



```
(base) node@machine:~$ docker pull cloudera/quickstart:latest
latest: Pulling from cloudera/quickstart
Image docker.io/cloudera/quickstart:latest uses outdated schema1 manifest format.
1d00652ce734: Already exists
Digest: sha256:f91bee4cdfa2c92ea3652929a22f729d4d13fc838b00f120e630f91c941acb63
Status: Image is up to date for cloudera/quickstart:latest
docker.io/cloudera/quickstart:latest
```

Step 2: Download the dataset from Kaggle

We are using UK house pricing dataset that contains information on all registered property sales in England and Wales that are sold for full market value which can be downloaded from the following link

https://www.kaggle.com/hm-land-registry/uk-housing-prices-paid?select=price_paid_records.csv

Unzip the dataset and place the zip file along with the code files in a folder that would be used inside the container, in my case it is /home/node/Desktop/BDA

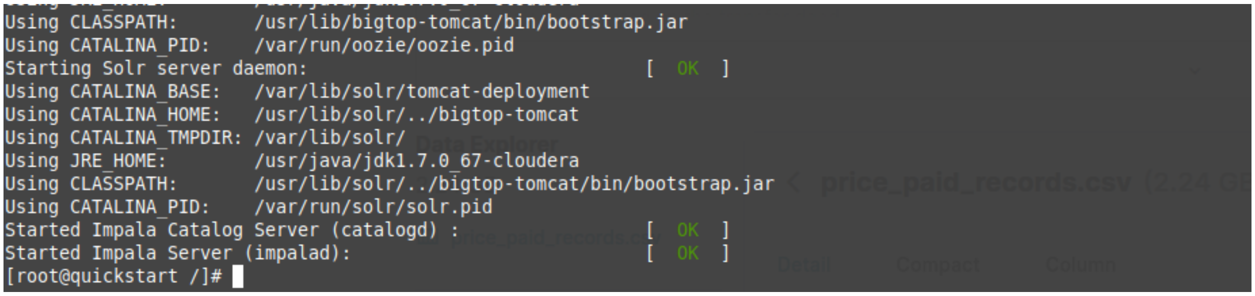
Step 3: Start the Docker image by executing the following command (Change the '/home/node/Desktop/BDA' to your folder location)

docker run --hostname=quickstart.cloudera --privileged=true -t -v /home/node/Desktop/BDA:/user/cloudera/shared -i -p 8888:8888 -p 7180:7180 cloudera/quickstart /usr/bin/docker-quickstart

Here

- /home/node/Desktop/BDA:/user/cloudera/shared: a mapping from the dataset folder on my Desktop to a shared folder within the cloudera docker (can now use the dataset within docker)
- 8888:8888: port 8888 of container is mapped to 8888 of my hosted OS (can now access hue on my Ubuntu through localhost:8888)
- 7180:7180: mapping for Cloudera manager, required to start the docker
- /usr/bin/docker-quickstart: starts the Cloudera docker
- --privileged=true: This is required for HBase, MySQL-backed Hive metastore, Hue, Oozie, Sentry, and Cloudera Manager
- -t : provide a bash shell after starting
- -i : we want to use the terminal

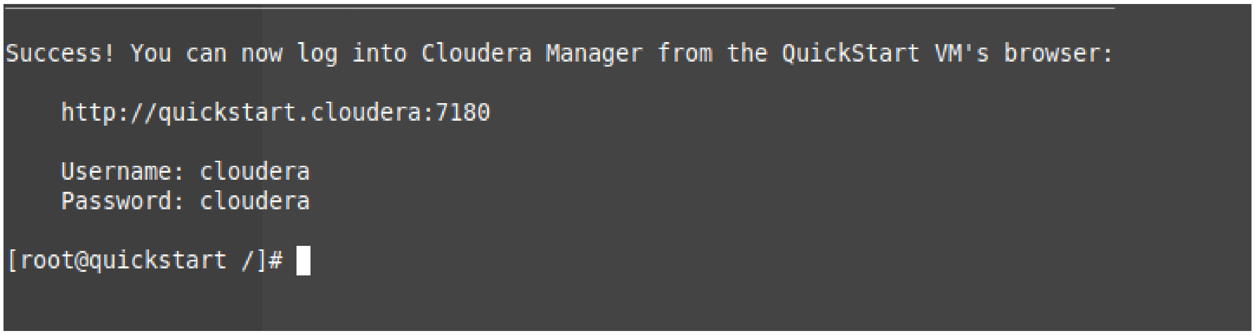
After the image is completely loaded, you will have a prompt inside the docker instance like this



```
Using CLASSPATH:       /usr/lib/bigtop-tomcat/bin/bootstrap.jar
Using CATALINA_PID:    /var/run/oozie/oozie.pid
Starting Solr server daemon:                               [  OK  ]
Using CATALINA_BASE:   /var/lib/solr/tomcat-deployment
Using CATALINA_HOME:   /usr/lib/solr/../bigtop-tomcat
Using CATALINA_TMPDIR: /var/lib/solr/
Using JRE_HOME:        /usr/java/jdk1.7.0_67-cloudera
Using CLASSPATH:       /usr/lib/solr/../bigtop-tomcat/bin/bootstrap.jar
Using CATALINA_PID:    /var/run/solr/solr.pid
Started Impala Catalog Server (catalogd) :                 [  OK  ]
Started Impala Server (impalad):                           [  OK  ]
[root@quickstart /]#
```

Step 4: Execute the following command to start cloudera manager

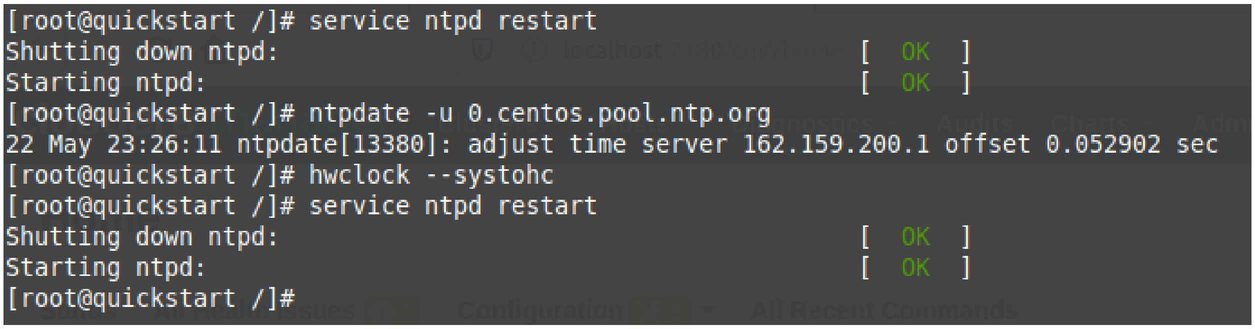
/home/cloudera/cloudera-manager –express



```
Success! You can now log into Cloudera Manager from the QuickStart VM's browser:

    http://quickstart.cloudera:7180

    Username: cloudera
    Password: cloudera

[root@quickstart /]#
```

Step 5: Execute the following commands (one by one) to sync your Docker time with NTP

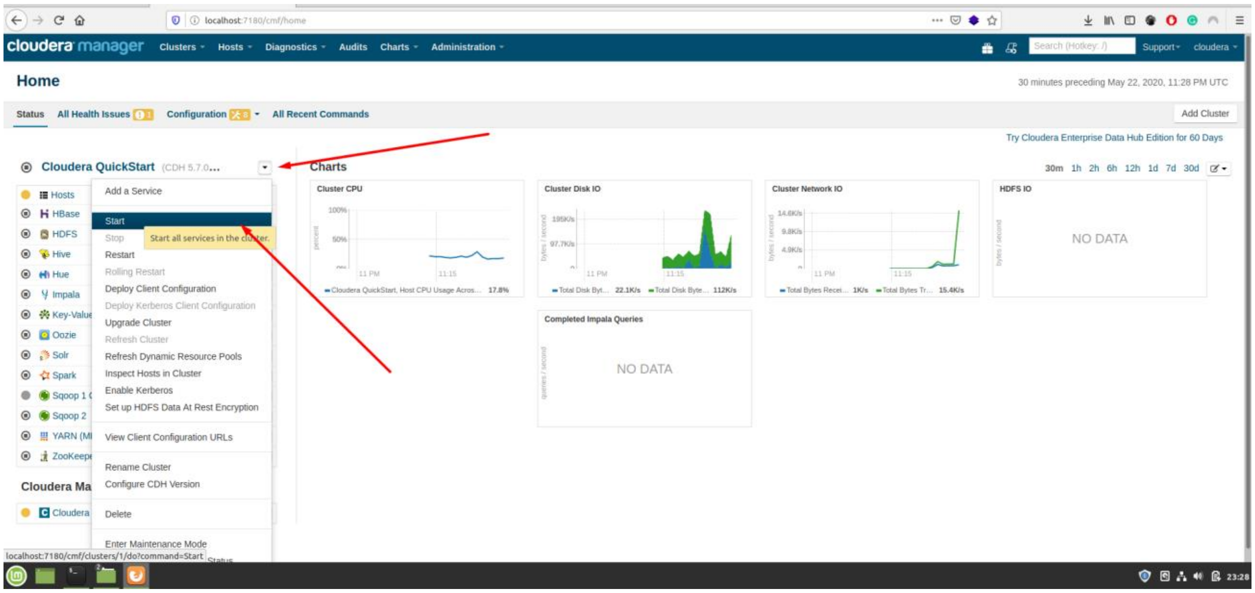
service ntpd restart
ntpdate -u 0.centos.pool.ntp.org
hwclock –systohc
service ntpd restart

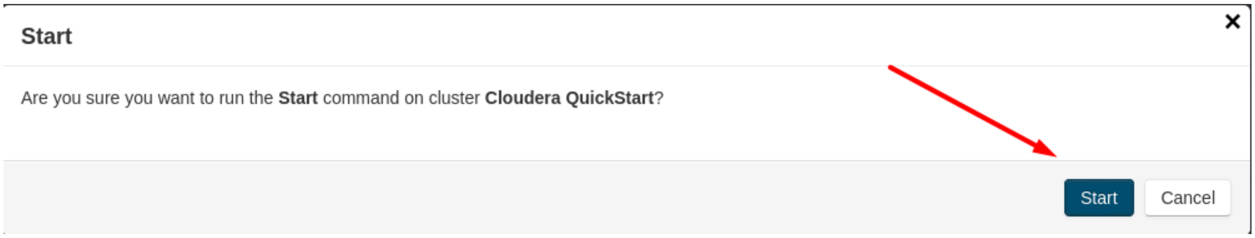


```
[root@quickstart /]# service ntpd restart
Shutting down ntpd:                                        [  OK  ]
Starting ntpd:                                             [  OK  ]
[root@quickstart /]# ntpdate -u 0.centos.pool.ntp.org
22 May 23:26:11 ntpdate[13380]: adjust time server 162.159.200.1 offset 0.052902 sec
[root@quickstart /]# hwclock --systohc
[root@quickstart /]# service ntpd restart
Shutting down ntpd:                                        [  OK  ]
Starting ntpd:                                             [  OK  ]
[root@quickstart /]#
```

Step 6: Open cloudera manager at the following url with cloudera as username and password http://localhost:7180/

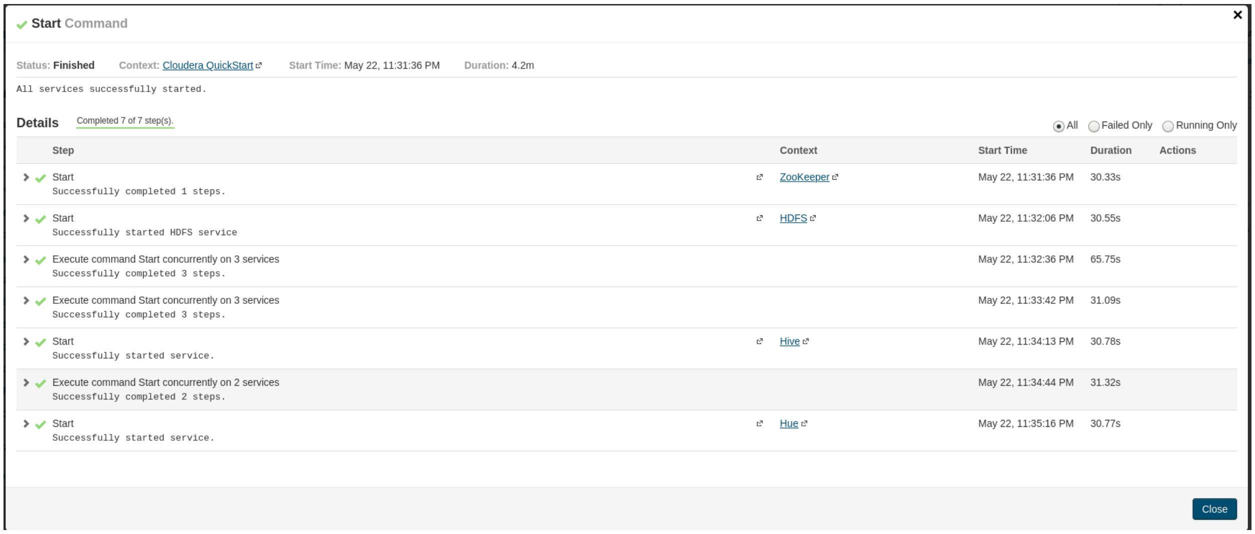
Step 7: Press the down arrow button next to 'Cloudera Quickstart' title, press Start to start all cloudera service and select start on the prompt as well, as in the given screenshot





Start

Are you sure you want to run the **Start** command on cluster **Cloudera QuickStart**?



Start Command

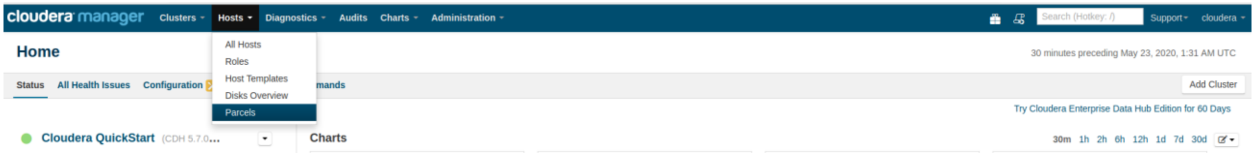
Status: **Finished** Context: Cloudera QuickStart Start Time: May 22, 11:31:36 PM Duration: 4.2m

```
All services successfully started.
```

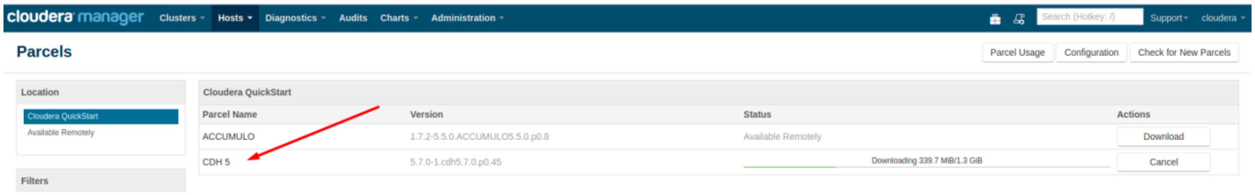
Details Completed 7 of 7 step(s).

Step	Context	Start Time	Duration	Actions
✓ Start Successfully completed 1 steps.	ZooKeeper	May 22, 11:31:36 PM	30.33s	
✓ Start Successfully started HDFS service	HDFS	May 22, 11:32:06 PM	30.55s	
✓ Execute command Start concurrently on 3 services Successfully completed 3 steps.		May 22, 11:32:36 PM	65.75s	
✓ Execute command Start concurrently on 3 services Successfully completed 3 steps.		May 22, 11:33:42 PM	31.09s	
✓ Start Successfully started service.	Hive	May 22, 11:34:13 PM	30.78s	
✓ Execute command Start concurrently on 2 services Successfully completed 2 steps.		May 22, 11:34:44 PM	31.32s	
✓ Start Successfully started service.	Hue	May 22, 11:35:16 PM	30.77s	

Step 8: After that, we need to install CDH5 parcel, go the following Hosts > Parcels in the top menu



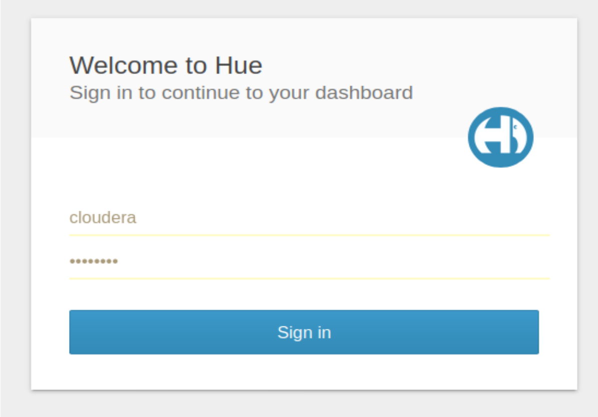
Step 9: Download the CDH 5 parcel by clicking the Download button in the second row.



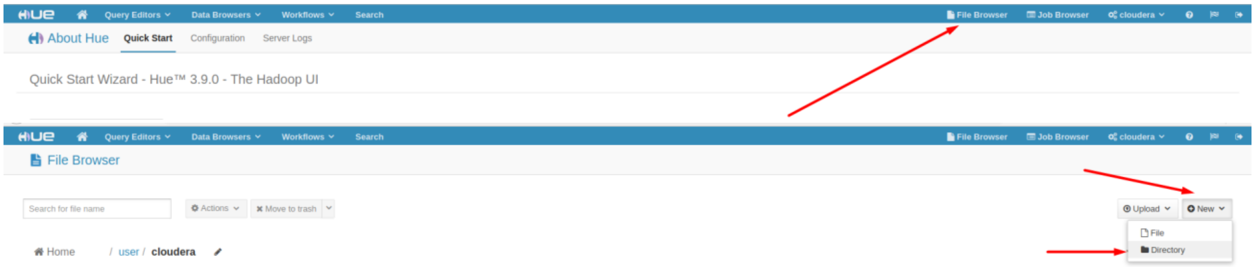
Parcel Name	Version	Status	Actions
ACCUMULO	1.7.2-5.5.0.ACCUMULO5.5.0.p0.8	Available Remotely	Download
CDH 5	5.7.0-1.cdh5.7.0.p0.45	Downloading 339.7 MiB/1.3 GiB	Cancel

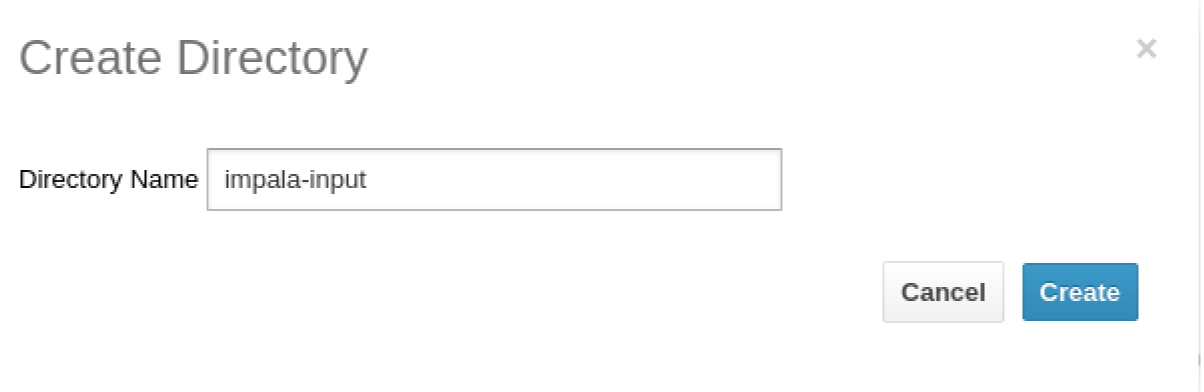
Projec Setup

Step 10: We would use Hue to upload to interface with HDFS and upload our dataset. Open the following url and use username/password ‘cloudera’
http://localhost:8888/



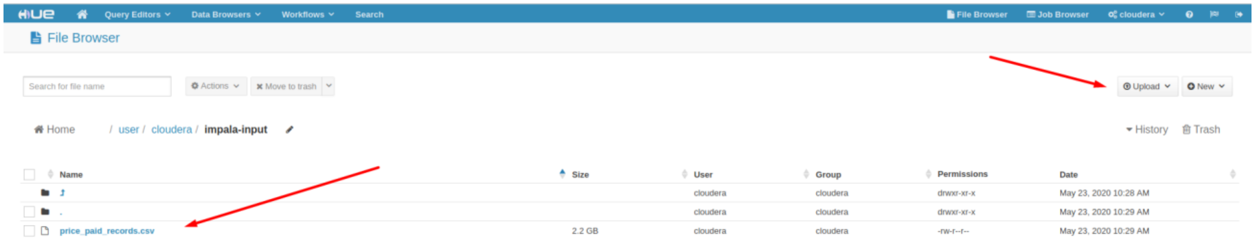
Step 11: Open File Browser, create a folder ‘impala-input’





Directory Name: impala-input

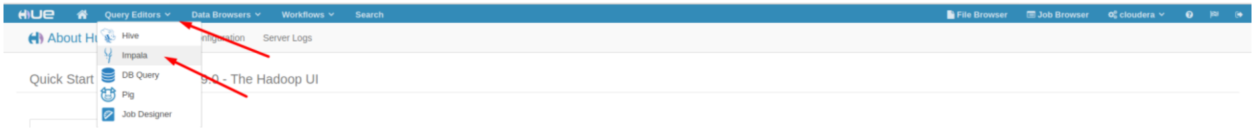
Step 12: Now open the 'impala-input' folder in Hue and upload out dataset.



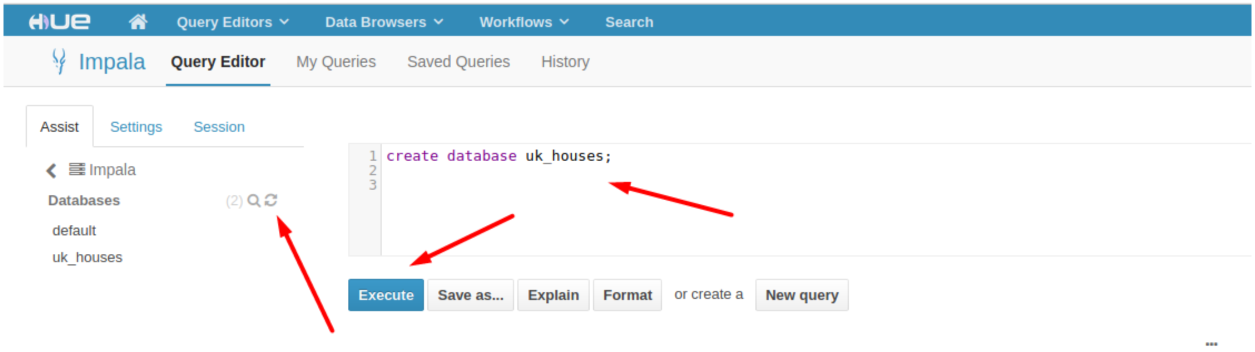
Name	Size	User	Group	Permissions	Date
↥		cloudera	cloudera	drwxr-xr-x	May 23, 2020 10:28 AM
.		cloudera	cloudera	drwxr-xr-x	May 23, 2020 10:29 AM
price_paid_records.csv	2.2 GB	cloudera	cloudera	-rw-r--r--	May 23, 2020 10:29 AM

Execution

Step 13: Open Impala editor in Hue

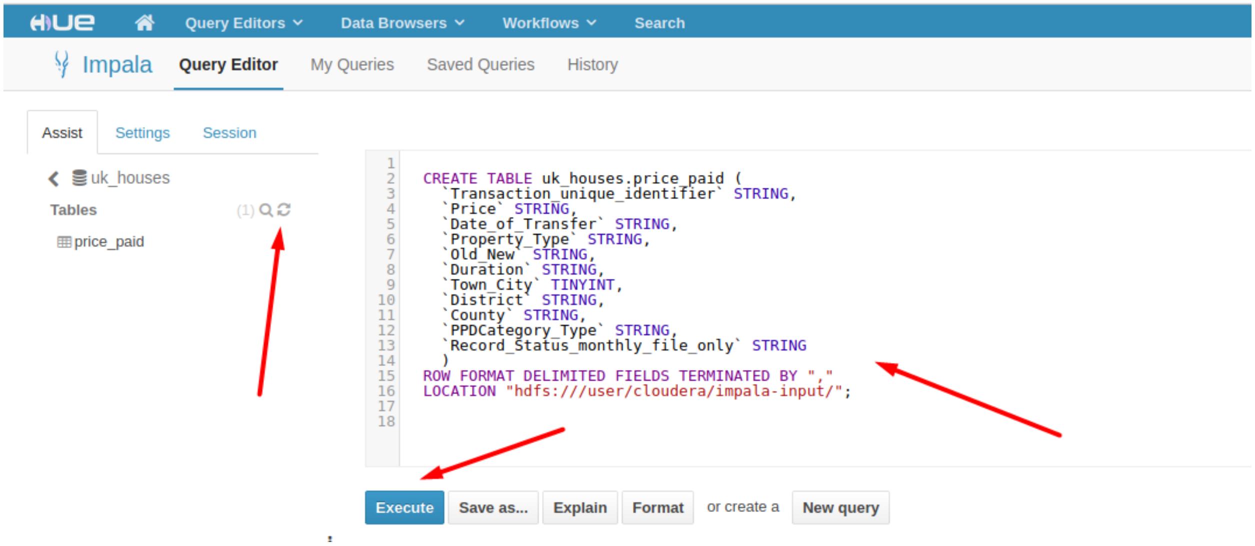


Step 14: Execute the following query to create DB and refresh



```
create database uk_houses;
```

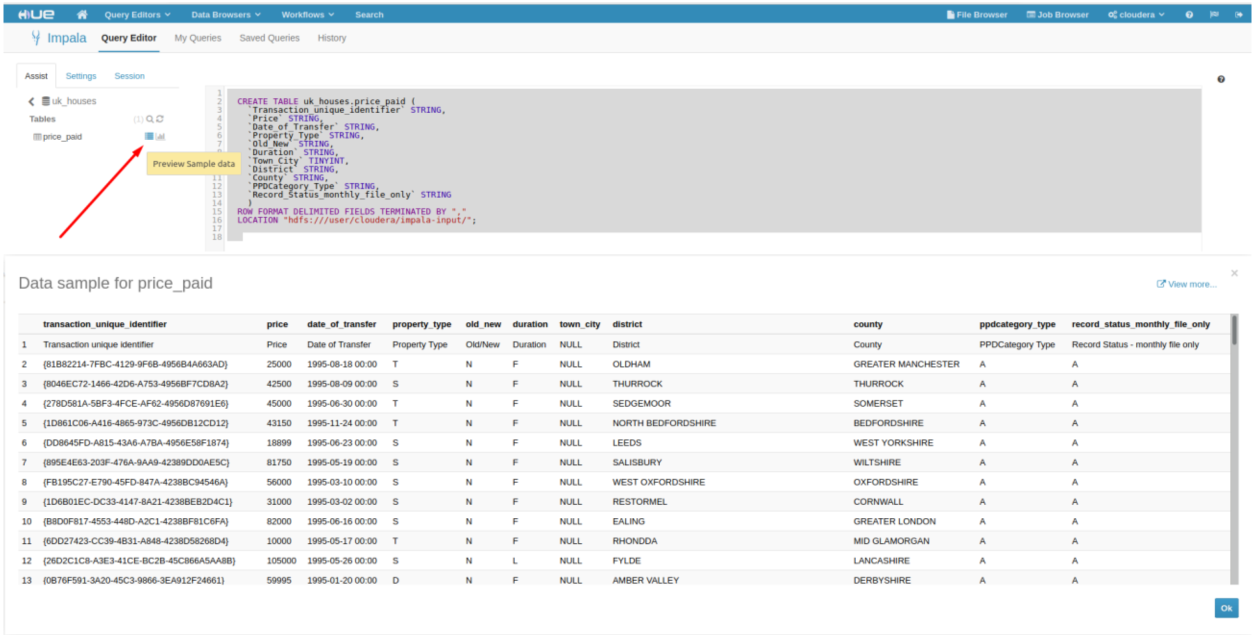
Step 15: Execute the following query to create the table for our dataset



```
CREATE TABLE uk_houses.price_paid (
   `Transaction_unique_identifier` STRING,
   `Price` STRING,
   `Date_of_Transfer` STRING,
   `Property_Type` STRING,
   `Old_New` STRING,
   `Duration` STRING,
   `Town_City` TINYINT,
   `District` STRING,
   `County` STRING,
   `PPDCategory_Type` STRING,
   `Record_Status_monthly_file_only` STRING
   )
ROW FORMAT DELIMITED FIELDS TERMINATED BY ","
LOCATION "hdfs:///user/cloudera/impala-input/";
```

```
CREATE TABLE uk_houses.price_paid (
  `Transaction_unique_identifier` STRING,
  `Price` STRING,
  `Date_of_Transfer` STRING,
  `Property_Type` STRING,
  `Old_New` STRING,
  `Duration` STRING,
  `Town_City` TINYINT,
  `District` STRING,
  `County` STRING,
  `PPDCategory_Type` STRING,
  `Record_Status_monthly_file_only` STRING
  )
ROW FORMAT DELIMITED FIELDS TERMINATED BY ","
LOCATION "hdfs:///user/cloudera/impala-input/";
```

Step 16: Check the table sample

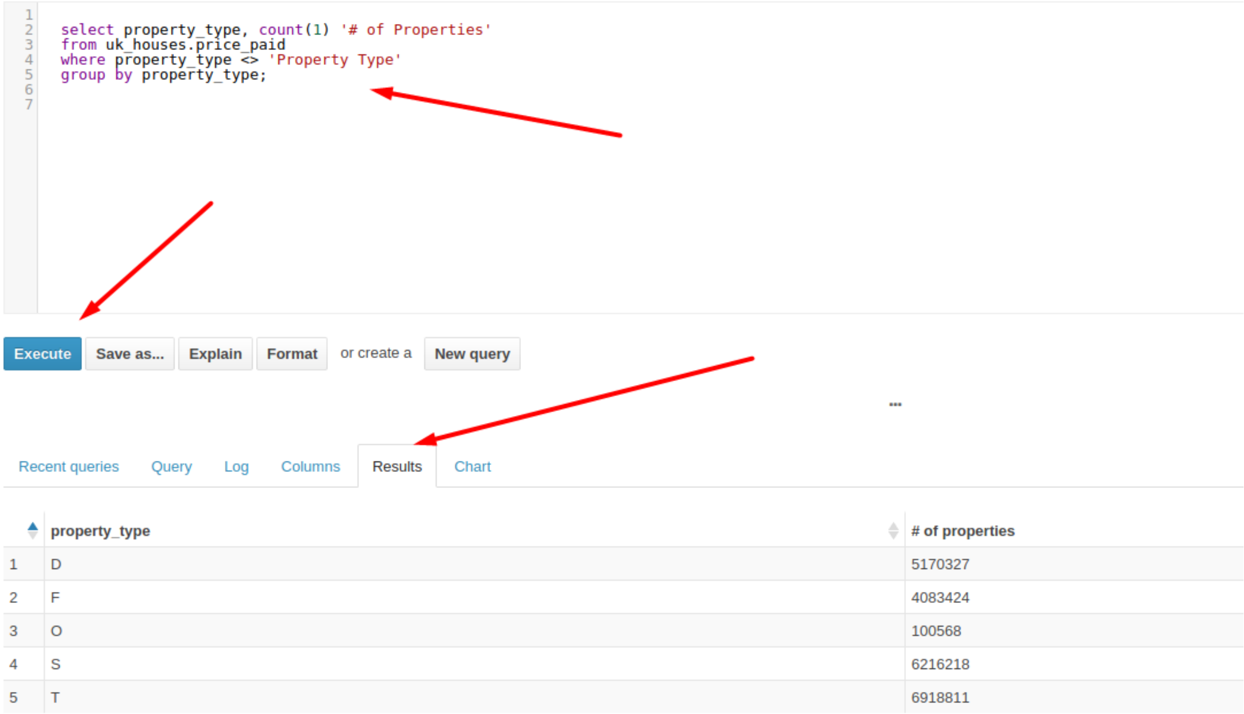


```
CREATE TABLE uk_houses.price_paid (
  `Transaction_unique_identifier` STRING,
  `Price` STRING,
  `Date_of_Transfer` STRING,
  `Property_Type` STRING,
  `Old_New` STRING,
  `Duration` STRING,
  `Town_City` TINYINT,
  `District` STRING,
  `County` STRING,
  `PPDCategory_Type` STRING,
  `Record_Status_monthly_file_only` STRING
  )
ROW FORMAT DELIMITED FIELDS TERMINATED BY ","
LOCATION "hdfs:///user/cloudera/impala-input/";
```

Data sample for price_paid

	transaction_unique_identifier	price	date_of_transfer	property_type	old_new	duration	town_city	district	county	ppdcategory_type	record_status_monthly_file_only
1	Transaction unique identifier	Price	Date of Transfer	Property Type	Old/New	Duration	NULL	District	County	PPDCategory Type	Record Status - monthly file only
2	{81B82214-7FBC-4129-9F6B-4956B4A663AD}	25000	1995-08-18 00:00	T	N	F	NULL	OLDHAM	GREATER MANCHESTER	A	A
3	{8046EC72-1466-42D6-A753-4956BF7CD8A2}	42500	1995-08-09 00:00	S	N	F	NULL	THURROCK	THURROCK	A	A
4	{278D581A-5BF3-4FCE-AF62-4956D87691E6}	45000	1995-06-30 00:00	T	N	F	NULL	SEDGEMOOR	SOMERSET	A	A
5	{1D861C06-A416-4865-973C-4956DB12CD12}	43150	1995-11-24 00:00	T	N	F	NULL	NORTH BEDFORDSHIRE	BEDFORDSHIRE	A	A
6	{DD8645FD-A815-43A6-A7BA-4956E58F1874}	18899	1995-06-23 00:00	S	N	F	NULL	LEEDS	WEST YORKSHIRE	A	A
7	{895E4E63-203F-476A-9AA9-42389DD0AE5C}	81750	1995-05-19 00:00	S	N	F	NULL	SALISBURY	WILTSHIRE	A	A
8	{FB195C27-E790-45FD-847A-4238BC94546A}	56000	1995-03-10 00:00	S	N	F	NULL	WEST OXFORDSHIRE	OXFORDSHIRE	A	A
9	{1D6B01EC-DC33-4147-8A21-4238BEB2D4C1}	31000	1995-03-02 00:00	S	N	F	NULL	RESTORMEL	CORNWALL	A	A
10	{B8D0F817-4553-448D-A2C1-4238BF81C6FA}	82000	1995-06-16 00:00	S	N	F	NULL	EALING	GREATER LONDON	A	A
11	{6DD27423-CC39-4B31-A848-4238D58268D4}	10000	1995-05-17 00:00	T	N	F	NULL	RHONDDA	MID GLAMORGAN	A	A
12	{26D2C1C8-A3E3-41CE-BC2B-45C866A5AA8B}	105000	1995-05-26 00:00	S	N	L	NULL	FYLDE	LANCASHIRE	A	A
13	{0B76F591-3A20-45C3-9866-3EA912F24661}	59995	1995-01-20 00:00	D	N	F	NULL	AMBER VALLEY	DERBYSHIRE	A	A

Step 17: Execute the following query to find the number of properties by Property Type



```
select property_type, count(1) '# of Properties'
from uk_houses.price_paid
where property_type <> 'Property Type'
group by property_type;
```

	property_type	# of properties
1	D	5170327
2	F	4083424
3	O	100568
4	S	6216218
5	T	6918811