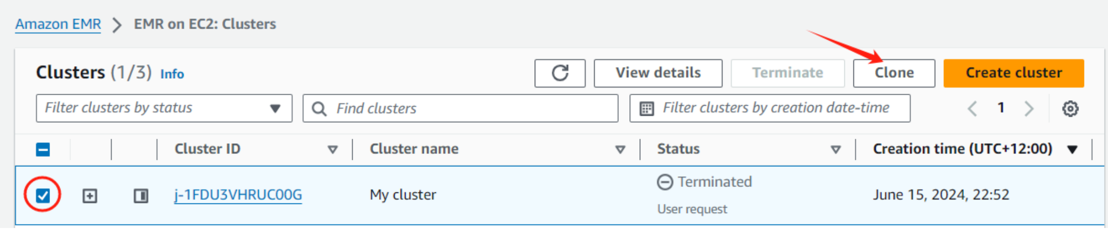
Lab 3 Hive QL

In this lab, you will learn different ways to operate Hive, including web interface and JDBC (optional). You also need to practise some basic HQL using Hue.

To begin the lab, please create an Amazon EMR instance or clone the EMR instance you set up during the last lab. If you forget, you can refer to the previous lab instructions.

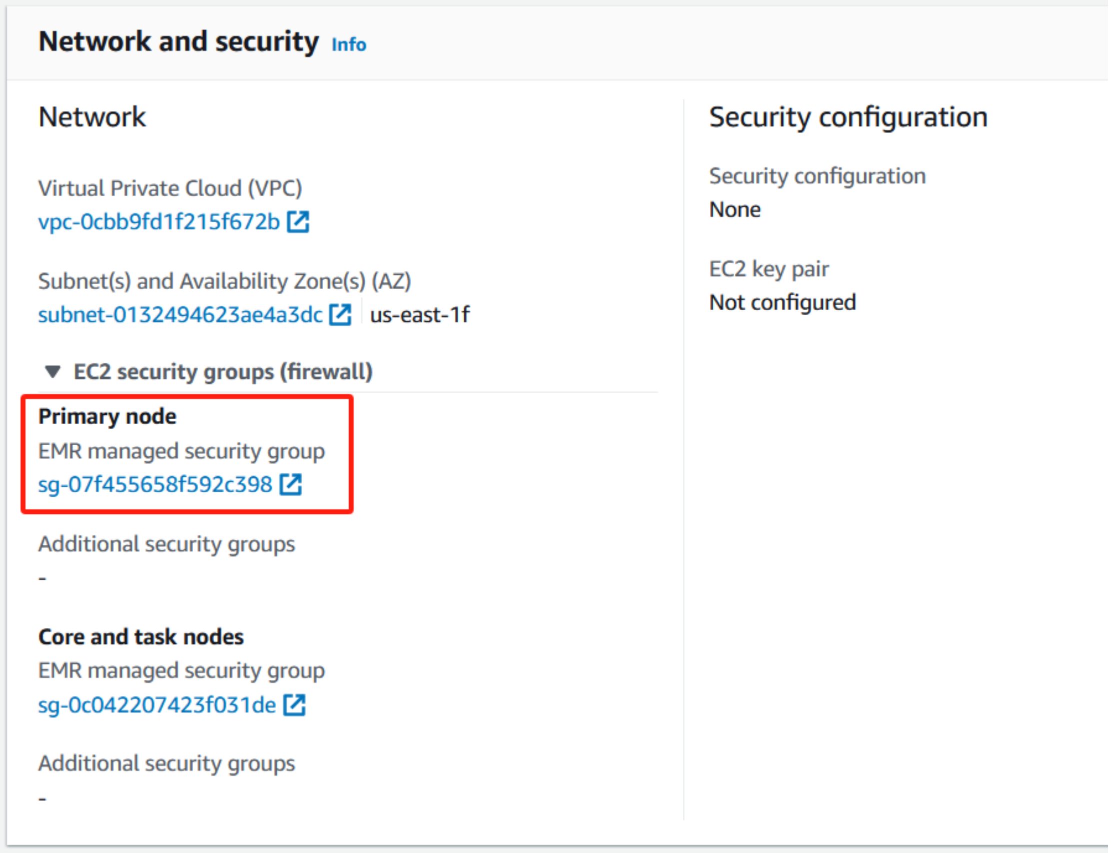
You may also **clone** the previous instance that you created for the previous lab, as shown below.


Amazon EMR > EMR on EC2: Clusters
Clusters (1/3) Info
View details
Terminate
Clone
Create cluster
Filter clusters by status
Find clusters
Filter clusters by creation date-time
1

	Cluster ID	Cluster name	Status	Creation time (UTC+12:00)
☑	j-1FDU3VHRUC00G	My cluster	Terminated User request	June 15, 2024, 22:52

Task 1 Preparation – Update Security Policy

When your EMR instance starts, expand the EC2 security groups and then choose the security group of the primary node.



Network and security Info

Network

Virtual Private Cloud (VPC)
vpc-0cbb9fd1f215f672b

Subnet(s) and Availability Zone(s) (AZ)
subnet-0132494623ae4a3dc | us-east-1f

▼ EC2 security groups (firewall)

Primary node
EMR managed security group
sg-07f455658f592c398

Additional security groups
-

Core and task nodes
EMR managed security group
sg-0c042207423f031de

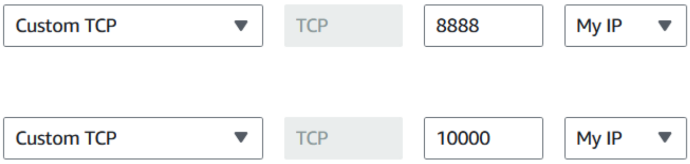
Additional security groups
-

Security configuration

Security configuration
None

EC2 key pair
Not configured

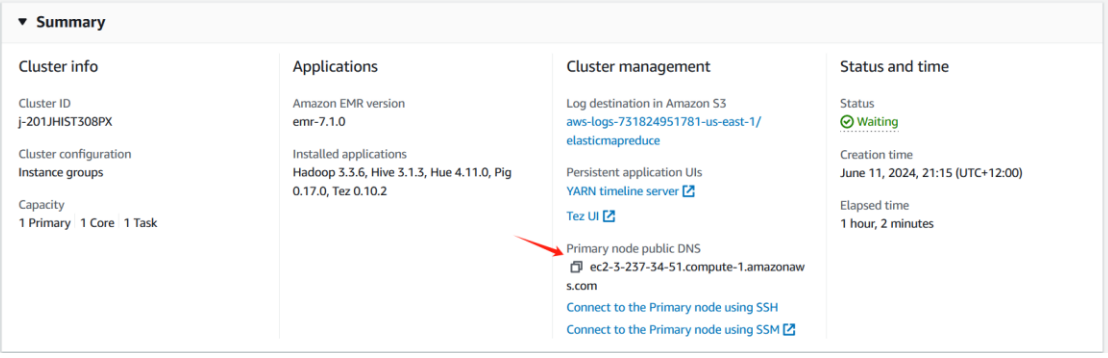
Select “Edit the inbound rules”, and then add two new rules, which allow your IP to access ports 8888 and 10000, being used by Hue and hive.server2.thrift, respectively. Then, save the rules.



Custom TCP ▼	TCP	8888	My IP ▼
Custom TCP ▼	TCP	10000	My IP ▼

Task 2 Hue and HQL

Find your cluster’s public DNS from the summary of the cluster. Copy the public DNS.



▼ Summary

Cluster info

Cluster ID
j-201JHIST308PX

Cluster configuration
Instance groups

Capacity
1 Primary | 1 Core | 1 Task

Applications

Amazon EMR version
emr-7.1.0

Installed applications
Hadoop 3.3.6, Hive 3.1.3, Hue 4.11.0, Pig 0.17.0, Tez 0.10.2

Cluster management

Log destination in Amazon S3
aws-logs-731824951781-us-east-1/elasticmapreduce

Persistent application UIs
YARN timeline server
Tez UI

Primary node public DNS
ec2-3-237-34-51.compute-1.amazonaws.com
Connect to the Primary node using SSH
Connect to the Primary node using SSM

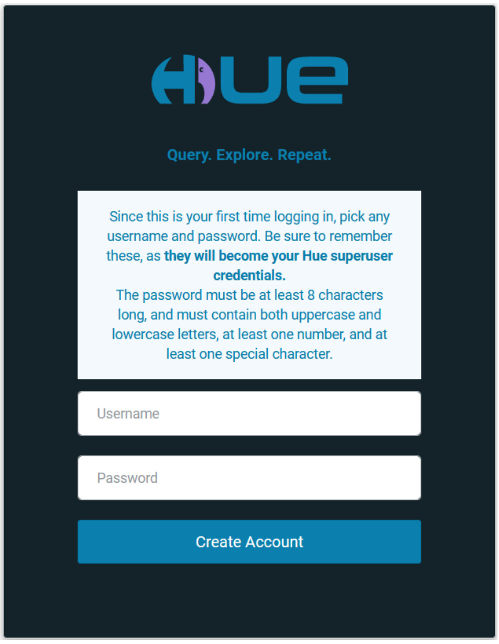
Status and time

Status
Waiting

Creation time
June 11, 2024, 21:15 (UTC+12:00)

Elapsed time
1 hour, 2 minutes

Open your browser (Chrome or Firefox), then access your EMR public DNS with the port number 8888. For example, http://ec2-3-237-34-51.compute-1.amazonaws.com:8888, you should be able to see the Hue web interface as below.



Query. Explore. Repeat.

Since this is your first time logging in, pick any username and password. Be sure to remember these, as **they will become your Hue superuser credentials.**
The password must be at least 8 characters long, and must contain both uppercase and lowercase letters, at least one number, and at least one special character.

Username

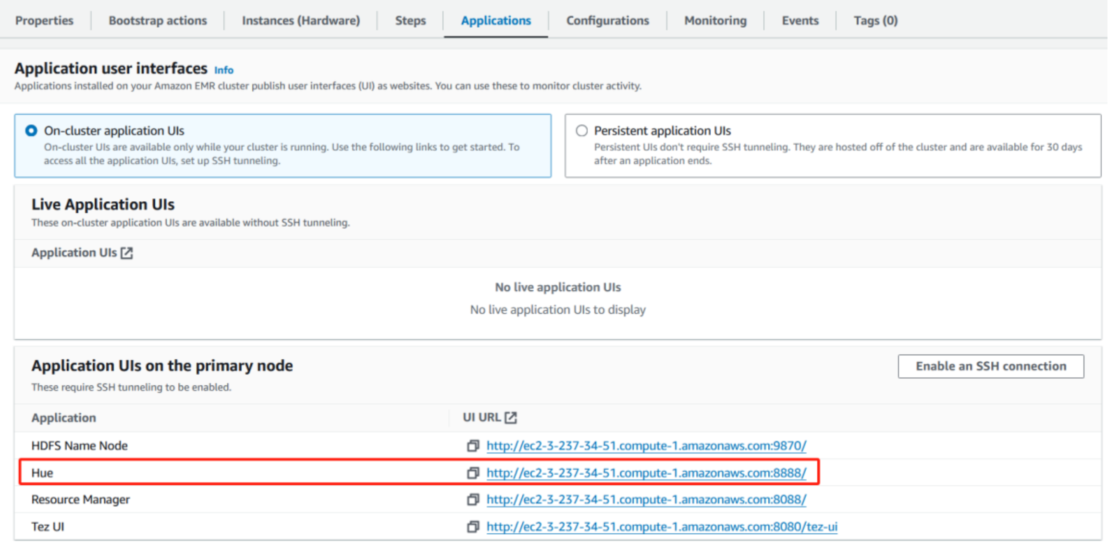
Password

Create Account

Please create the account using the username: **hadoop**

Don’t use other usernames. Then, provide a password that satisfies the password policy.

You can also find the Hue Link from the applications tab of your EMR cluster:



Properties | Bootstrap actions | Instances (Hardware) | Steps | Applications | Configurations | Monitoring | Events | Tags (0)

Application user interfaces Info

Applications installed on your Amazon EMR cluster publish user interfaces (UI) as websites. You can use these to monitor cluster activity.

● On-cluster application UIs
On-cluster UIs are available only while your cluster is running. Use the following links to get started. To access all the application UIs, set up SSH tunneling.

○ Persistent application UIs
Persistent UIs don't require SSH tunneling. They are hosted off of the cluster and are available for 30 days after an application ends.

Live Application UIs

These on-cluster application UIs are available without SSH tunneling.

Application UIs

No live application UIs

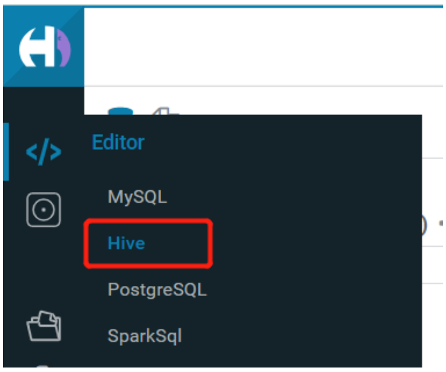
No live application UIs to display

Application UIs on the primary node

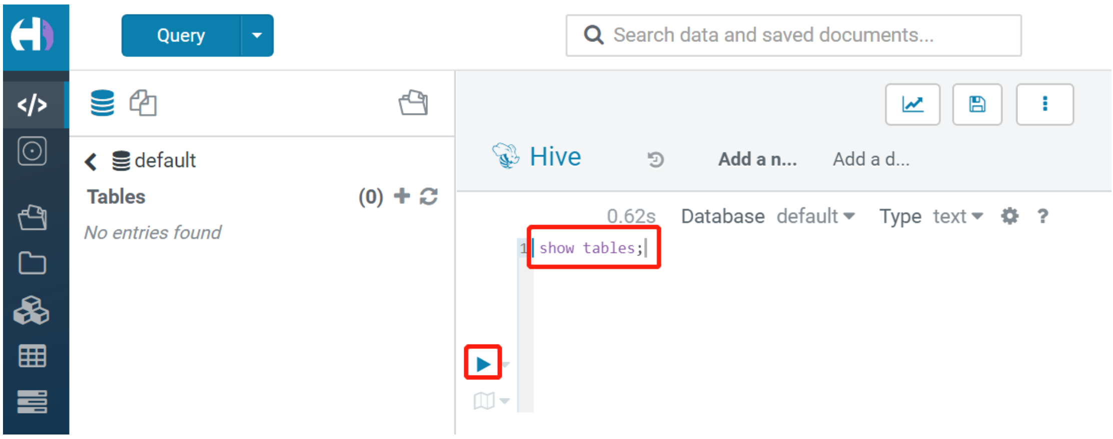
These require SSH tunneling to be enabled.

Application	UI URL
HDFS Name Node	http://ec2-3-237-34-51.compute-1.amazonaws.com:9870/
Hue	http://ec2-3-237-34-51.compute-1.amazonaws.com:8888/
Resource Manager	http://ec2-3-237-34-51.compute-1.amazonaws.com:8088/
Tez UI	http://ec2-3-237-34-51.compute-1.amazonaws.com:8080/tez-ui

Switch to HIVE:



Create an account by providing a username and password. Please pay attention to the password policy. After logging in to Hue, type “show tables” in the Hive Editor and running it. Check out the results.

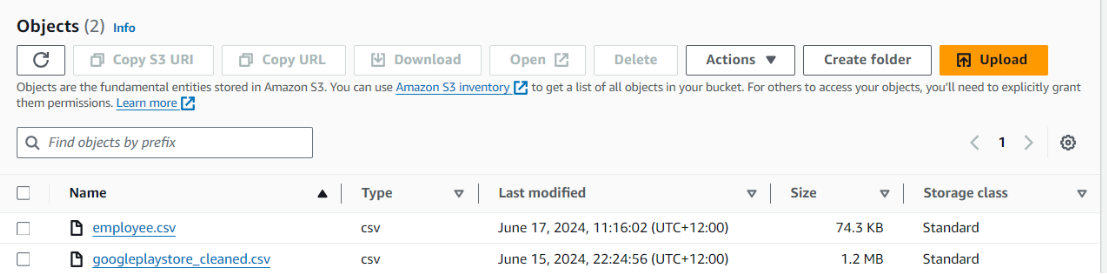


```
show tables;
```

Task 3 Employee Data Analysis with Hive

This task helps the students, especially those from non-IT backgrounds, to learn HQL. This task is developed based on the Kaggle dataset: https://www.kaggle.com/datasets/rhuebner/human-resources-data-set

Download the employee.csv from Canvas and upload it to your S3 storage. In this tutorial, the data file has been uploaded to *'s3://bigdatabucket2024/datasets/employee.csv'*. Your S3 path will be different.



Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name	Type	Last modified	Size	Storage class
□	employee.csv	csv	June 17, 2024, 11:16:02 (UTC+12:00)	74.3 KB	Standard
□	googleplaystore_cleaned.csv	csv	June 15, 2024, 22:24:56 (UTC+12:00)	1.2 MB	Standard

Employee Data

Column Name	Column Type	Description
Employee_Name	STRING	Name of the employee
EmpID	INT	Employee ID
MarriedID	INT	Indicator if the employee is married (0: No, 1: Yes)
MaritalStatusID	INT	ID representing the marital status
GenderID	INT	ID representing the gender
EmpStatusID	INT	ID representing the employment status
DeptID	INT	Department ID
PerfScoreID	INT	Performance score ID
FromDiversityJobFairID	INT	Indicator if hired from a diversity job fair (0: No, 1: Yes)
Salary	INT	Employee's salary
Termd	INT	Indicator if the employee is terminated (0: No, 1: Yes)
PositionID	INT	Position ID
Position	STRING	Job title of the employee
State	STRING	State where the employee works
Zip	STRING	Zip code of the work location
DOB	DATE	Date of birth of the employee
Sex	STRING	Gender of the employee
MaritalDesc	STRING	Description of the marital status
CitizenDesc	STRING	Citizenship status
HispanicLatino	STRING	Indicator if the employee is Hispanic or Latino (No/Yes)
RaceDesc	STRING	Description of the race
DateofHire	DATE	Date when the employee was hired
DateofTermination	DATE	Date when the employee was terminated
TermReason	STRING	Reason for termination
EmploymentStatus	STRING	Employment status (e.g., Active)
Department	STRING	Department where the employee works
ManagerName	STRING	Name of the manager
ManagerID	INT	Manager ID
RecruitmentSource	STRING	Source of recruitment
PerformanceScore	STRING	Performance score description
EngagementSurvey	FLOAT	Engagement survey score
EmpSatisfaction	INT	Employee satisfaction score
SpecialProjectsCount	INT	Number of special projects the employee is involved in

LastPerformanceReview_Date	DATE	Date of the last performance review
DaysLateLast30	INT	Number of days late in the last 30 days
Absences	INT	Number of absences

Create the employee_data table using the below script in HUE.

```
CREATE TABLE employee_data (
   Employee_Name STRING,
   EmpID INT,
   MarriedID INT,
   MaritalStatusID INT,
   GenderID INT,
   EmpStatusID INT,
   DeptID INT,
   PerfScoreID INT,
   FromDiversityJobFairID INT,
   Salary INT,
   Termd INT,
   PositionID INT,
   Position STRING,
   State STRING,
   Zip STRING,
   DOB DATE,
   Sex STRING,
   MaritalDesc STRING,
   CitizenDesc STRING,
   HispanicLatino STRING,
   RaceDesc STRING,
   DateofHire DATE,
   DateofTermination DATE,
   TermReason STRING,
   EmploymentStatus STRING,
   Department STRING,
   ManagerName STRING,
   ManagerID INT,
   RecruitmentSource STRING,
   PerformanceScore STRING,
   EngagementSurvey FLOAT,
   EmpSatisfaction INT,
   SpecialProjectsCount INT,
   LastPerformanceReview_Date DATE,
   DaysLateLast30 INT,
   Absences INT
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE
TBLPROPERTIES ("skip.header.line.count"="1");
```

Load the data from S3 storage to the Hive:

```
LOAD DATA INPATH 's3://bigdatabucket2024/datasets/employee.csv' INTO TABLE
employee_data;
```

Tips: You may need to change the S3 location to where your employee.csv located.

Querying Data

Select all data from the table

```
SELECT * FROM employee_data;
```

Select specific columns

```
SELECT Employee_Name, Position, Salary FROM employee_data;
```

Filter data with a WHERE clause and sort the data in a descending order using ORDER BY

```
SELECT Employee_Name, Position, Salary
FROM employee_data
WHERE salary > 60000
ORDER BY salary DESC;
```

Using Functions and Expressions

Calculate the average salary:

```
SELECT AVG(Salary) AS avg_salary FROM employee_data;
```

Calculate the average salary by position

```
SELECT position, AVG(Salary) AS avg_salary
FROM employee_data
GROUP BY position;
```

Tips: when using aggregate functions such as COUNT, SUM, or AVG, any non-aggregated columns, such as position, must be included in the GROUP BY clause.

Convert employee names to uppercase:

```
SELECT UPPER(Employee_Name) AS upper_name FROM employee_data;
```

Extract year from date of birth:

```
SELECT Employee_Name, YEAR(DOB) AS birth_year FROM employee_data;
```

Calculate the age of the employee:

```
SELECT
   Employee_Name,
   YEAR(DOB) AS birth_year,
   YEAR(CURRENT_DATE) - YEAR(DOB) AS age
FROM
   employee_data;
```

When you add “ORDER BY age” to the above clause, you will find some minus-age employees appear. This is caused by incorrect data processing, e.g., dob as 70-4-12 was converted to 2070, rather than 1970. Let’s fix it by adding IF THEN ELSE.

```
SELECT
   Employee_Name,
   YEAR(DOB) AS birth_year,
   CASE
      WHEN (YEAR(CURRENT_DATE) - YEAR(DOB)) < 0 THEN (YEAR(CURRENT_DATE) -
YEAR(DOB)) + 100
      ELSE (YEAR(CURRENT_DATE) - YEAR(DOB))
   END AS age
FROM
   employee_data
ORDER BY age;
```

Aggregating Data

Count the number of employees in each department:

```
SELECT Department, COUNT(*) AS num_employees
FROM employee_data
GROUP BY Department;
```

Find the maximum and minimum salary in each department:

```
SELECT Department, MAX(Salary) AS max_salary, MIN(Salary) AS min_salary
FROM employee_data
GROUP BY Department;
```

Understand the gender distribution within each department.

```
SELECT Department, Sex, COUNT(*) AS num_employees
FROM employee_data
GROUP BY Department, Sex;
```

Top 5 Highest Paid Positions

```
SELECT Position, MAX(Salary) AS max_salary
FROM employee_data
GROUP BY Position
ORDER BY max_salary DESC
```

```
LIMIT 5;
```

Calculate the average employee satisfaction score for each department.

```
SELECT Department, AVG(EmpSatisfaction) AS avg_satisfaction
FROM employee_data
GROUP BY Department;
```

Determine the total number of absences reported in each department.

```
SELECT Department, SUM(Absences) AS total_absences
FROM employee_data
GROUP BY Department;
```

Find out how performance scores are distributed across employees.

```
SELECT PerformanceScore, COUNT(*) AS num_employees
FROM employee_data
GROUP BY PerformanceScore;
```

Get the minimum, maximum, average, and standard deviation of engagement survey scores for each department.

```
SELECT Department,
    MIN(EngagementSurvey) AS min_engagement,
    MAX(EngagementSurvey) AS max_engagement,
    AVG(EngagementSurvey) AS avg_engagement,
    STDDEV(EngagementSurvey) AS stddev_engagement
FROM employee_data
GROUP BY Department;
```

Calculate the average tenure (in years) of employees by department.

```
SELECT Department,
    AVG(DATEDIFF(CURRENT_DATE, DateofHire) / 365.0) AS avg_tenure
FROM employee_data
GROUP BY Department;
```

Find out the number of Hispanic/Latino employees in each department.

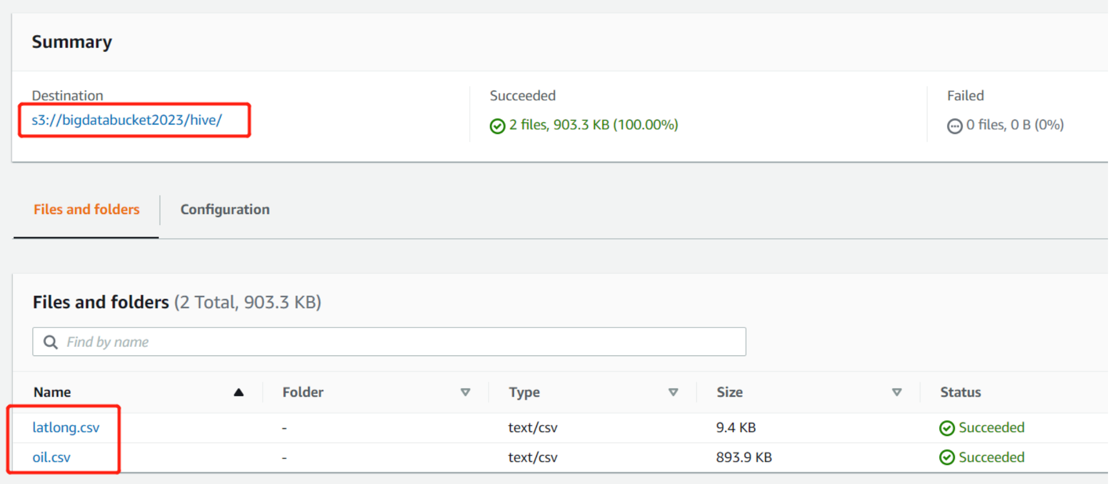
```
SELECT Department,
    SUM(CASE WHEN HispanicLatino = 'Yes' THEN 1 ELSE 0 END) AS num_hispanic_latino
FROM employee_data
GROUP BY Department;
```

Task 4 Analysing Oil Import Prices with Hive

This task is developed based on the “Practical Big Data Analytics – Chapter 4”. In this task, we will use Hive to analyse the import prices of oil in countries across the world from 1980-2016. The data is available from the OECD (Organization for Economic Co-operation and Development) website

Create a folder in under the S3 bucket you’ve created during the last lab. If you don’t have one, please go ahead and create a bucket with a folder, e.g., *hive* folder in below example.

Download two csv files from the Canvas and upload both to the folder.



Summary

Destination	Succeeded	Failed
s3://bigdatabucket2023/hive/	2 files, 903.3 KB (100.00%)	0 files, 0 B (0%)

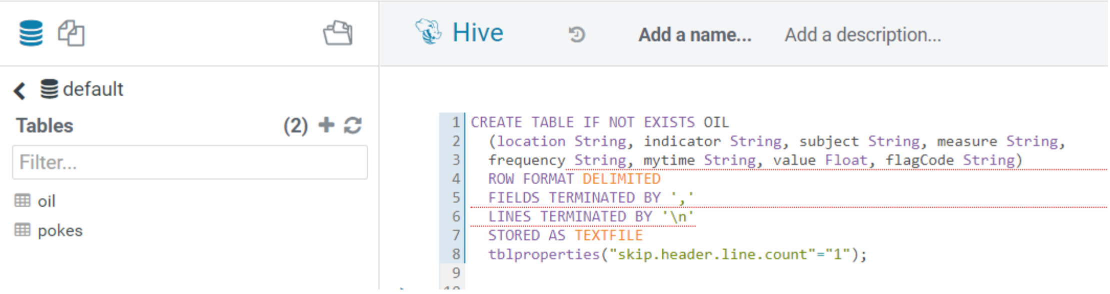
Files and folders (2 Total, 903.3 KB)

Name	Folder	Type	Size	Status
latlong.csv	-	text/csv	9.4 KB	Succeeded
oil.csv	-	text/csv	893.9 KB	Succeeded

Logon on Hue (Refer to Task 3) and follow the below steps:

# Hive QL Commands to create the table

```
CREATE TABLE IF NOT EXISTS OIL
        (location String, indicator String, subject String, measure String,
        frequency String, mytime String, value Float, flagCode String)
        ROW FORMAT DELIMITED
        FIELDS TERMINATED BY ','
        LINES TERMINATED BY '\n'
        STORED AS TEXTFILE
        tblproperties("skip.header.line.count"="1");
```



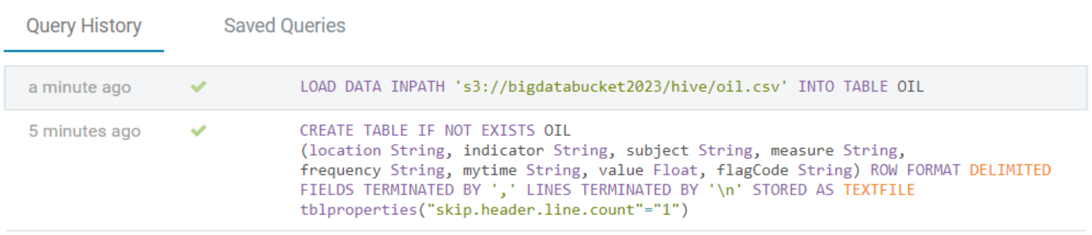
```
CREATE TABLE IF NOT EXISTS OIL
  (location String, indicator String, subject String, measure String,
  frequency String, mytime String, value Float, flagCode String)
  ROW FORMAT DELIMITED
  FIELDS TERMINATED BY ','
  LINES TERMINATED BY '\n'
  STORED AS TEXTFILE
  tblproperties("skip.header.line.count"="1");
```

# Load data from CSV to the table.

Please noted that the file path 's3://bigdatabucket2024/hive/oil.csv' could be different, dependent on the file path in your S3 folder.

```
LOAD DATA INPATH 's3://bigdatabucket2024/hive/oil.csv' INTO TABLE OIL;
```

You will see the results below:



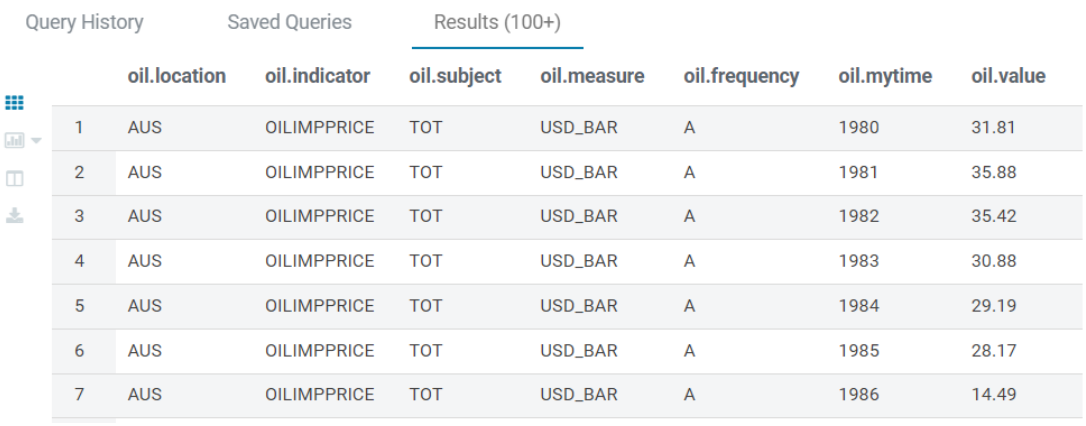
a minute ago ✓

```
LOAD DATA INPATH 's3://bigdatabucket2023/hive/oil.csv' INTO TABLE OIL
```

5 minutes ago ✓

```
CREATE TABLE IF NOT EXISTS OIL
(location String, indicator String, subject String, measure String,
frequency String, mytime String, value Float, flagCode String) ROW FORMAT DELIMITED
FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' STORED AS TEXTFILE
tblproperties("skip.header.line.count"="1")
```

Check the imported data using: SELECT * FROM oil;



	oil.location	oil.indicator	oil.subject	oil.measure	oil.frequency	oil.mytime	oil.value
1	AUS	OILIMPPRICE	TOT	USD_BAR	A	1980	31.81
2	AUS	OILIMPPRICE	TOT	USD_BAR	A	1981	35.88
3	AUS	OILIMPPRICE	TOT	USD_BAR	A	1982	35.42
4	AUS	OILIMPPRICE	TOT	USD_BAR	A	1983	30.88
5	AUS	OILIMPPRICE	TOT	USD_BAR	A	1984	29.19
6	AUS	OILIMPPRICE	TOT	USD_BAR	A	1985	28.17
7	AUS	OILIMPPRICE	TOT	USD_BAR	A	1986	14.49

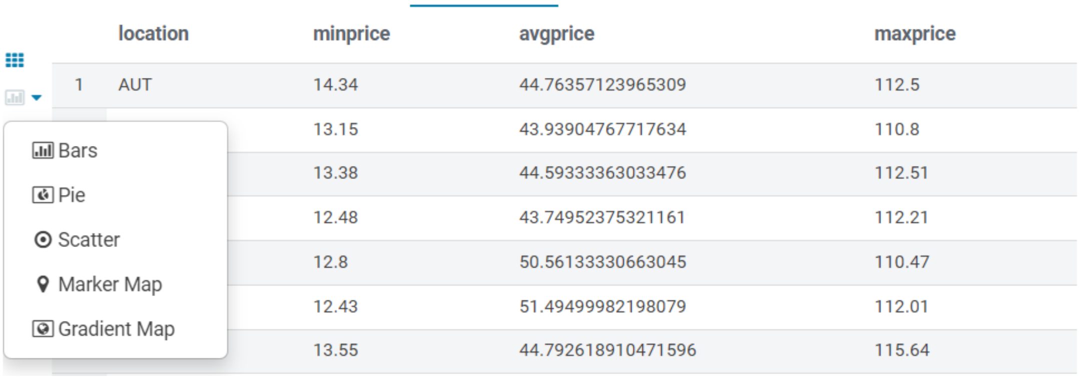
Run Query – To find the maximum, minimum, and average value of oil prices in each country from 1980-2015 (the date range of the dataset), we can use familiar SQL operators. The query would be as below:

```
SELECT LOCATION, MIN(value) as MINPRICE, AVG(value) as AVGPRICE,
MAX(value) as MAXPRICE
FROM OIL
WHERE FREQUENCY LIKE "A"
GROUP BY LOCATION;
```

The Hive Manual provides an in-depth look into these commands and the various ways data can be saved, queried, and retrieved.

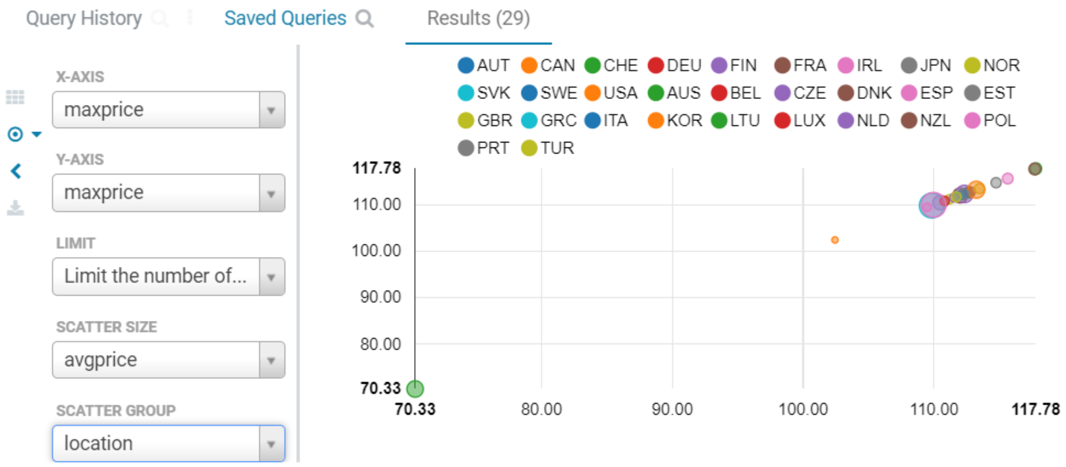
Hue also includes a set of useful features such as data visualization, data download, and others that allow users to perform ad hoc analysis on data.

To access the visualization feature, click on the visualization icon underneath the grid icon in the results section, as shown in the following screenshot:



	location	minprice	avgprice	maxprice
1	AUT	14.34	44.76357123965309	112.5
		13.15	43.93904767717634	110.8
		13.38	44.59333363033476	112.51
		12.48	43.74952375321161	112.21
		12.8	50.56133330663045	110.47
		12.43	51.49499982198079	112.01
		13.55	44.792618910471596	115.64

Select Scatter. In Hue, this type of chart, also known more generally as a scatterplot, allows users to create multivariate charts very easily. Different values for the x and y axes, as well as scatter size and grouping, can be selected, as shown in the following screenshot:



X-AXIS
maxprice

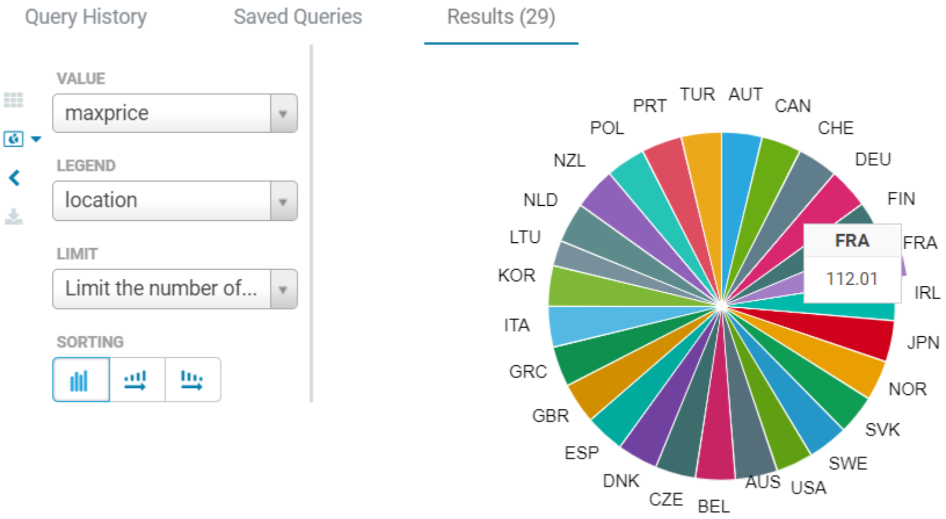
Y-AXIS
maxprice

LIMIT
Limit the number of...

SCATTER SIZE
avgprice

SCATTER GROUP
location

Next one is a simple pie chart that can be constructed by selecting Pie in the dropdown menu:



VALUE
maxprice

LEGEND
location

LIMIT
Limit the number of...

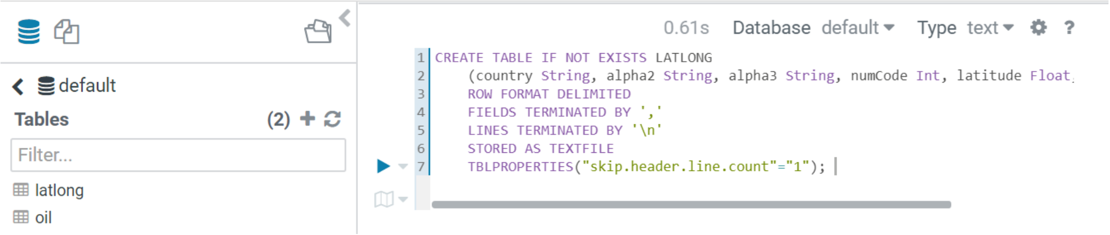
SORTING

Task 5 Joining Tables in Hive (Optional)

Hive supports advanced join functionalities. In this task, you will use join to query and visualize the data. It is based on the previous task of making sure that two CVS files are uploaded to the S3 bucket.

# Login to Hue and run Hive commands to create the table and load data

```
CREATE TABLE IF NOT EXISTS LATLONG
        (country String, alpha2 String, alpha3 String, numCode Int, latitude Float, 
longitude Float)
        ROW FORMAT DELIMITED
        FIELDS TERMINATED BY ','
        LINES TERMINATED BY '\n'
        STORED AS TEXTFILE
        TBLPROPERTIES("skip.header.line.count"="1");
```

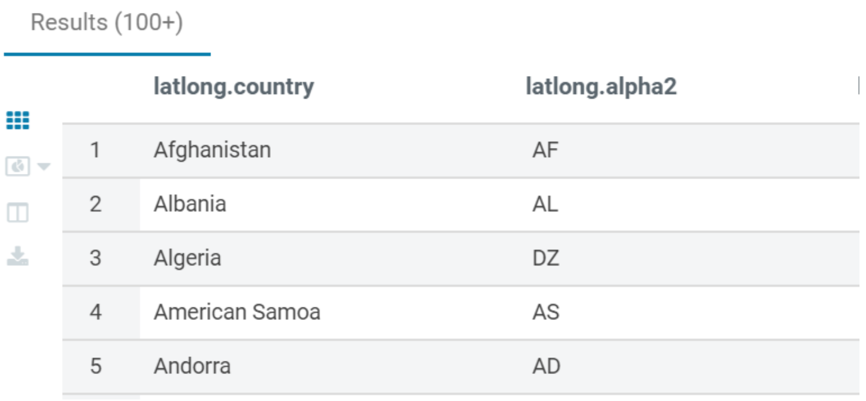


```
CREATE TABLE IF NOT EXISTS LATLONG
    (country String, alpha2 String, alpha3 String, numCode Int, latitude Float,
    ROW FORMAT DELIMITED
    FIELDS TERMINATED BY ','
    LINES TERMINATED BY '\n'
    STORED AS TEXTFILE
    TBLPROPERTIES("skip.header.line.count"="1");
```

# Load the data from S3.

```
LOAD DATA INPATH 's3://bigdatabucket2023/hive/latlong.csv' INTO TABLE LATLONG;
```

# Check the imported data by using: Select * from latlong;



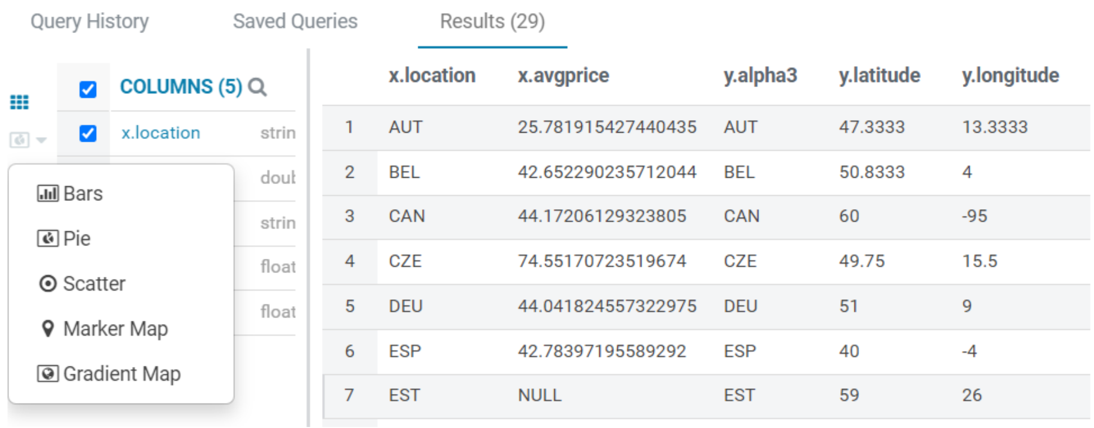
	latlong.country	latlong.alpha2
1	Afghanistan	AF
2	Albania	AL
3	Algeria	DZ
4	American Samoa	AS
5	Andorra	AD

# Join the oil data with the lat/log data:

```
SELECT DISTINCT * FROM
(SELECT location, avg(value) as AVGPRICE from oil GROUP BY location) x
LEFT JOIN
```

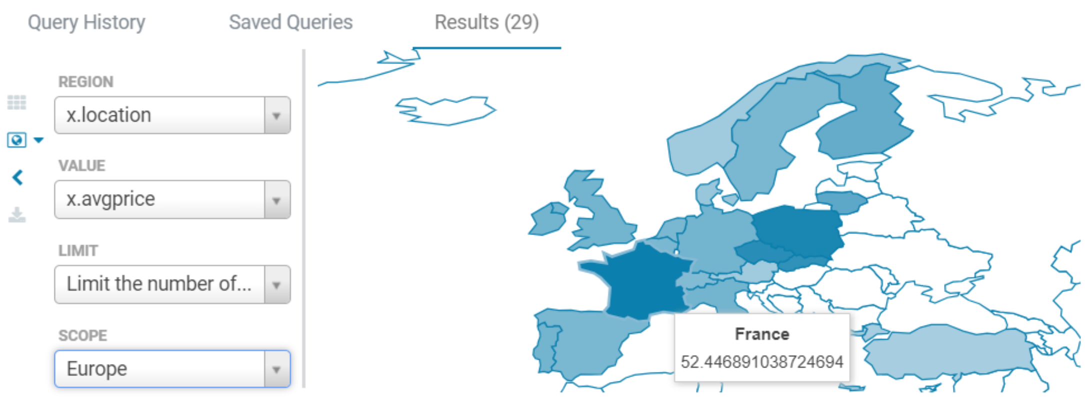
```
(SELECT TRIM(ALPHA3) AS alpha3, latitude, longitude from LATLONG) y
ON (x.location = y.alpha3);
```

We can now proceed with creating geospatial visualizations. It would be useful to bear in mind that these are preliminary visualizations in Hue that provide a very convenient means to view data.



	x.location	x.avgprice	y.alpha3	y.latitude	y.longitude
1	AUT	25.781915427440435	AUT	47.3333	13.3333
2	BEL	42.652290235712044	BEL	50.8333	4
3	CAN	44.17206129323805	CAN	60	-95
4	CZE	74.55170723519674	CZE	49.75	15.5
5	DEU	44.041824557322975	DEU	51	9
6	ESP	42.78397195589292	ESP	40	-4
7	EST	NULL	EST	59	26

Select “Gradient Map” from the drop-down menu and enter the appropriate values to create the chart, as shown in the following figure:



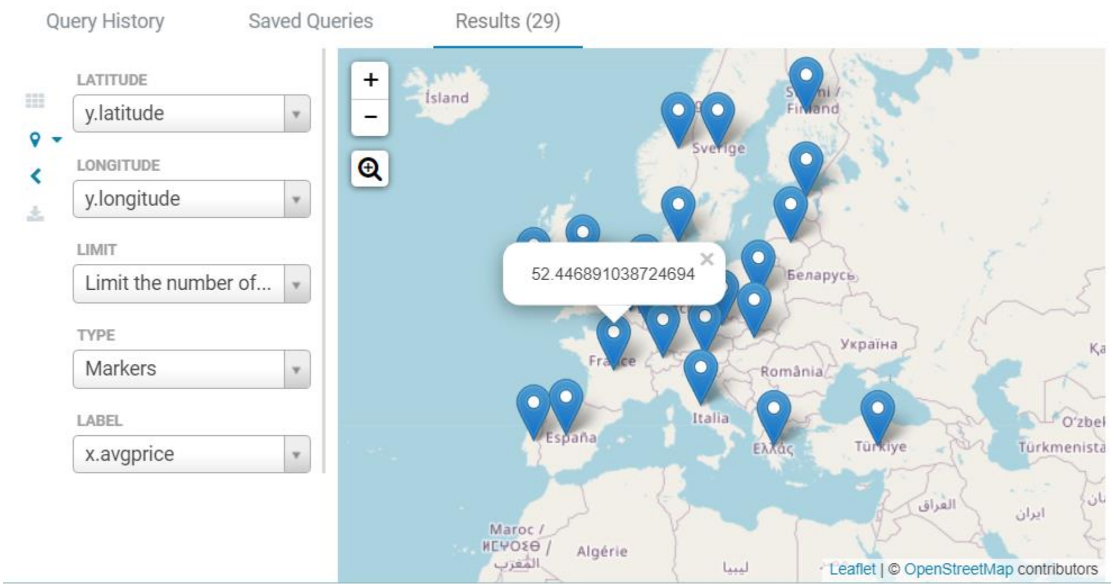
REGION
x.location

VALUE
x.avgprice

LIMIT
Limit the number of...

SCOPE
Europe

The next chart was developed using the Marker Map option in the drop-down menu. It uses the three-character country code to place markers and associated values on the respective regions, as shown in the following figure:



Query History
Saved Queries
Results (29)

LATITUDE
y.latitude

LONGITUDE
y.longitude

LIMIT
Limit the number of...

TYPE
Markers

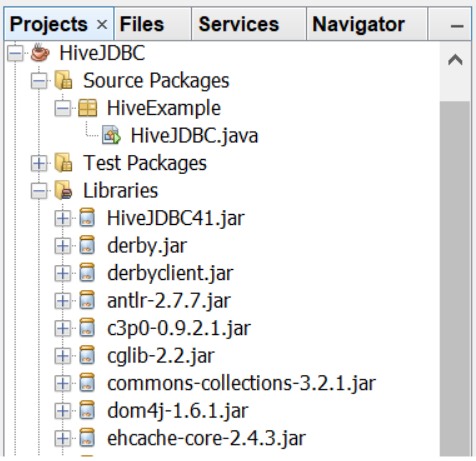
LABEL
x.avgprice

52.446891038724694

Task 6 Hive Thrift – JDBC (Optional)

Download HiveJDBC from the Canvas and unzip the file. Launch NetBeans and open this project. If you don't know how to install NetBeans, please refer to the instructions from the first lab.

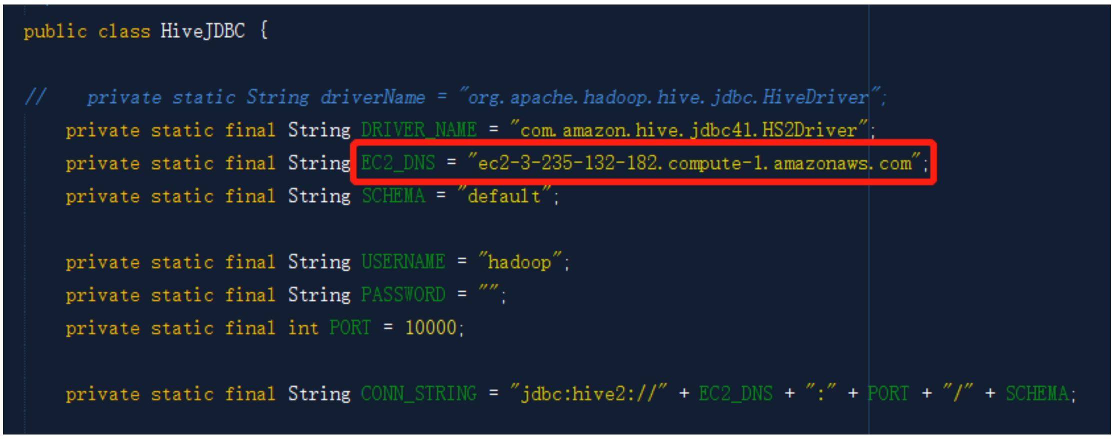
You will see the following figure when you expand the packages and libraries.



Projects × Files Services Navigator

```
HiveJDBC
  Source Packages
    HiveExample
      HiveJDBC.java
  Test Packages
  Libraries
    HiveJDBC41.jar
    derby.jar
    derbyclient.jar
    antlr-2.7.7.jar
    c3p0-0.9.2.1.jar
    cglib-2.2.jar
    commons-collections-3.2.1.jar
    dom4j-1.6.1.jar
    ehcache-core-2.4.3.jar
```

Open the HiveJDBC.java file and change the EC2_DNS to your EMR DNS.



```
public class HiveJDBC {

//    private static String driverName = "org.apache.hadoop.hive.jdbc.HiveDriver";
    private static final String DRIVER_NAME = "com.amazon.hive.jdbc41.HS2Driver";
    private static final String EC2_DNS = "ec2-3-235-132-182.compute-1.amazonaws.com";
    private static final String SCHEMA = "default";

    private static final String USERNAME = "hadoop";
    private static final String PASSWORD = "";
    private static final int PORT = 10000;

    private static final String CONN_STRING = "jdbc:hive2://" + EC2_DNS + ":" + PORT + "/" + SCHEMA;
```

You need to make sure that port 10000 is accessible by your IP. This step should be completed in the first task of this lab.

Right-click and choose run file. The query: select * from pokes will be sent to Hive. The results will be fetched and printed out to the console.

Please try other HQL.

Reference and Resources

- Dasgupta, N. (2018). Practical big data analytics: Hands-on techniques to implement enterprise analytics and machine learning using Hadoop, Spark, NoSQL and R. Packt Publishing Ltd.

- Getting Started with HQL, https://cwiki.apache.org/confluence/display/Hive/GettingStarted
- Connecting to the Hue Web User Interface, https://docs.aws.amazon.com/emr/latest/ReleaseGuide/accessing-hue.html
- View Web Interfaces Hosted on Amazon EMR Clusters, https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-web-interfaces.html

- EMR Bootstrap Actions, Latest Amazon JDBC/ODBC Drivers, http://awssupportdatasvcs.com/bootstrap-actions/Simba/latest/

- Starting Apache Hive, https://docs.cloudera.com/HDPDocuments/HDP3/HDP-3.1.5/starting-hive/content/hive_start_a_command_line_query_locally.html

- Use the Hive JDBC Driver, https://docs.aws.amazon.com/emr/latest/ReleaseGuide/HiveJDBCDriver.html