cloudera

Cloudera Altus on Azure - Hands-on Workshop

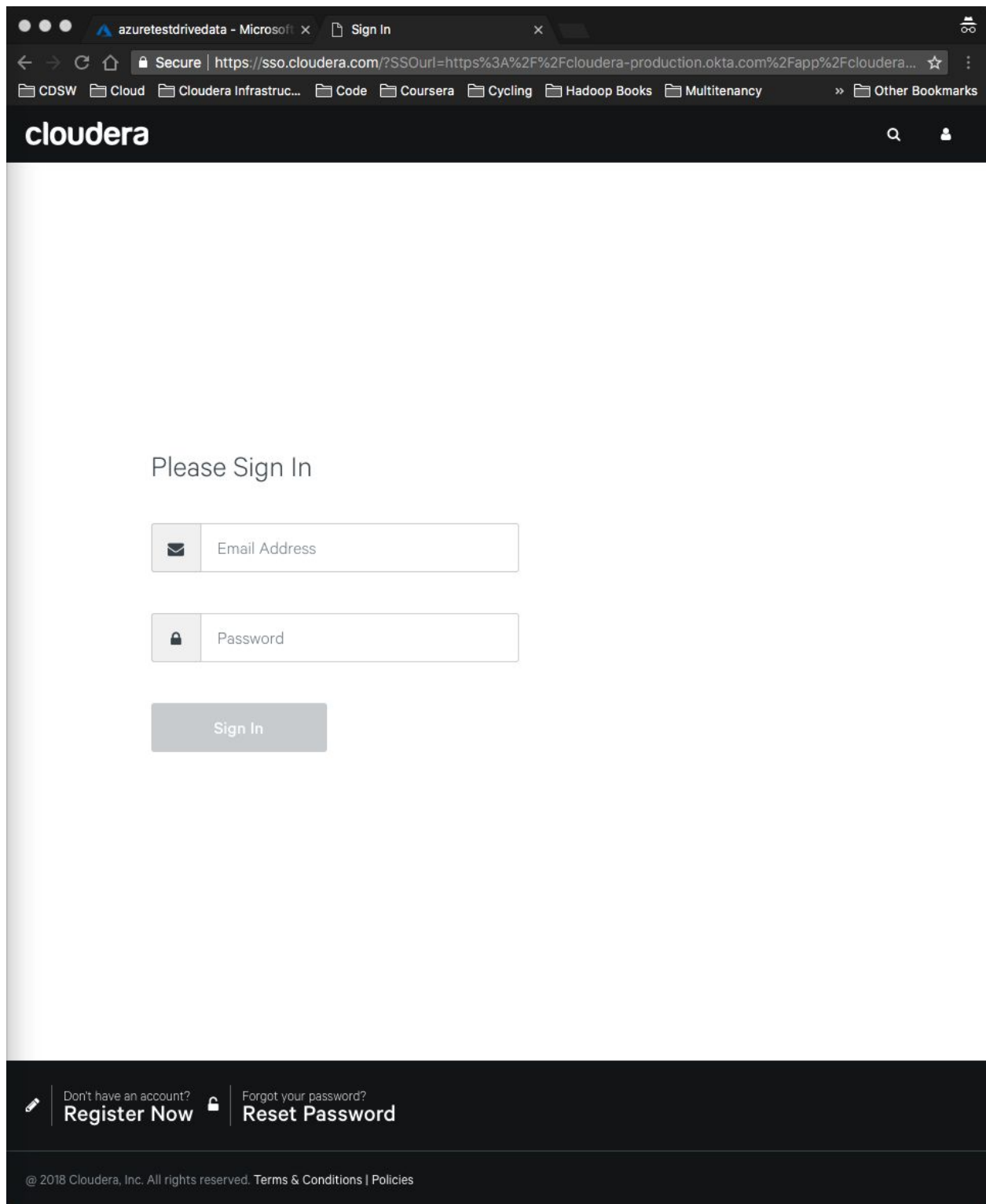Exercise Manual

Created by Cloudera

Summary

Welcome to the Altus on Azure Hands-on-Lab attendee guide. Below are a series of exercises designed to familiarize yourself with the Cloudera Altus platform offering. In addition to this, we will walk through a simple data engineering problem and how Altus empowers the end users to solve it and gain further insights into the data.

The data engineering problem consists of correlating structured data with unstructured data. Let's say you work for a sports retail company and you'd like to know the top selling products by state. That's easy enough. Just query the data stored in your relational database and you have your answer. Let's take it one step further and ask the question: Are the highest selling products also the most popular products on our website? How do we answer this question? At the end of this lab we will find out!

Exercise 1: Logging in to Altus

Now that we have had an overview of Altus, let's get our hands dirty. The first thing we'll need to do is log into the system. Navigate to the following URL in your browser, click **Sign In** at the top right:

http://altus.cloudera.com/

cloudera

Please Sign In

✉ Email Address

🔒 Password

Sign In

✎ Don't have an account?
Register Now   🔒 Forgot your password?
Reset Password

@ 2018 Cloudera, Inc. All rights reserved. **Terms & Conditions | Policies**

Cloudera **Altus**

Login with your username and password. Please note that this is a shared login which will be used by all lab attendees.
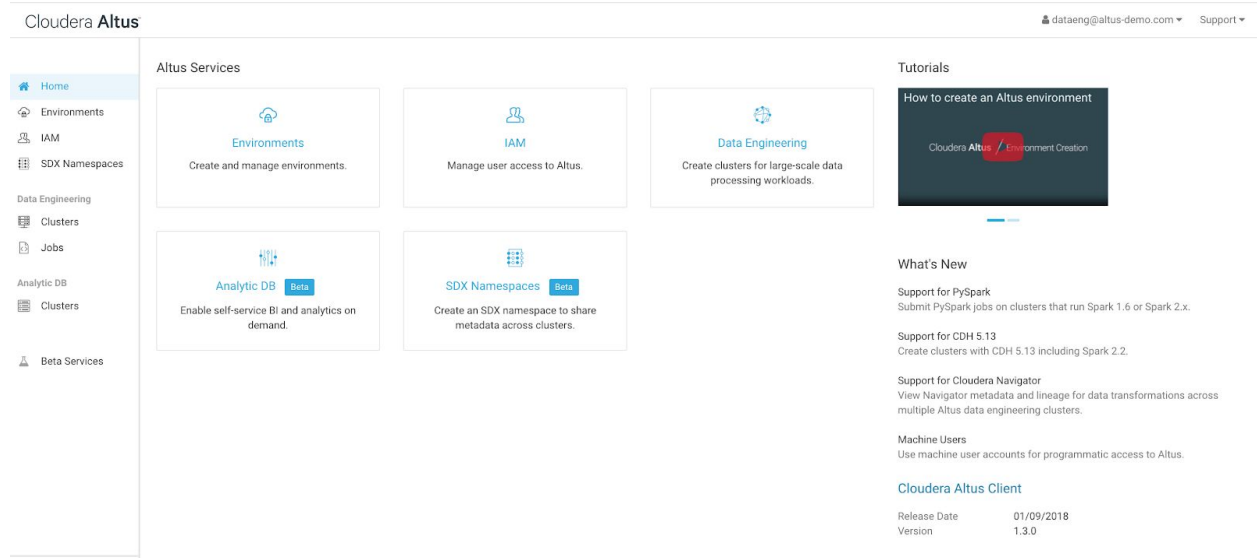Username: testdrive@altus-demo.com
Password: In the email you received when you registered.

Once you are logged in, we can get started. Your home page should look something similar to below.

The Altus organization you're currently logged in to is called the Altus-demo organization. This organization is used by Cloudera Engineers for demos as well as hands-on-labs, just like today's! Since this is a live demo environment, **please do NOT delete any clusters or resources.**

Exercise 2: Getting to know the Altus UI

Now that we have logged in, let's get familiar with the UI.

In the top-right hand corner of the page, click on Support -> Documentation
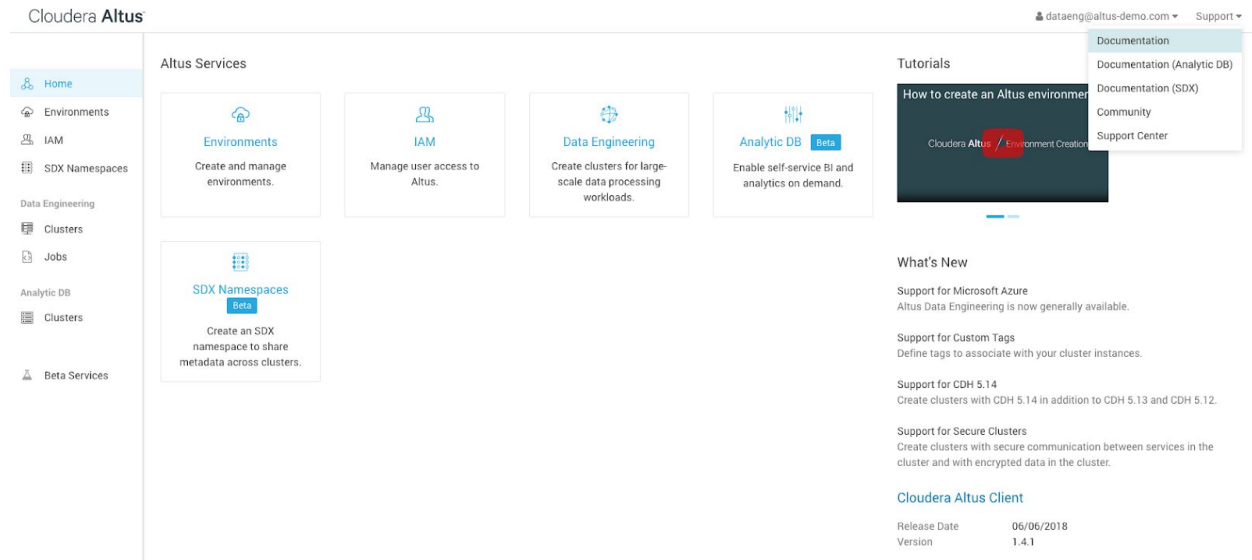
You can visit these docs at any time in case you have further questions about Altus components.

On the right-side of the page, you'll see the What's New section which details the latest features of Altus. Below that is the Latest Cloudera Altus CLI download link. Everything we're doing here today can be done through the CLI in a more programmatic fashion.

What's New

Support for Microsoft Azure
Altus Data Engineering is now generally available.

Support for Custom Tags
Define tags to associate with your cluster instances.

Support for CDH 5.14
Create clusters with CDH 5.14 in addition to CDH 5.13 and CDH 5.12.

Support for Secure Clusters
Create clusters with secure communication between services in the cluster and with encrypted data in the cluster.

Cloudera Altus Client

Release Date   06/06/2018
Version   1.4.1

Click on Environments in the left-hand pane. Then click on the Cloud button in the main pane and select Azure.

An Altus environment: Defines the resources in your Azure subscription that are used by Cloudera Altus to create clusters and jobs. In a production environment, an administrator can set up and assign separate Altus environments to different users and groups. **Please do not create any new environments for this lab**.

Click on the environment called "**AzureTestDrive2**" and view the environment details:

As you can see, Altus leverages existing resources in the customer's Azure subscription (resources such as vnets, network security groups etc. created by an Azure administrator). An Altus administrator then creates an Altus environment using those Azure resources.

Click on Clusters in the left-hand pane.

Cloudera Altus

Clusters

	Cloud All ▾	Environment All ▾	Status All ▾	Service Type All ▾	Search	Search clusters

- Home
- IAM
- Environments

Data Engineering
- **Clusters**
- Jobs

Analytic DB
- Clusters

Create Cluster

Cloud ⇕	Name ⇕	Status ⇕	Type ⇕	Workers ⇕	Creation Time ⇕
AWS	Altus_DG_RetailData	✓ Created	Hive on Spark	3	01/25/2018 3:56 PM EST
	AltusDE-Spark-Demo	✓ Created	Spark 2.x	3	01/22/2018 6:59 PM EST
AWS	engineering-training	✓ Created	Hive	5	01/09/2018 12:12 PM EST

Displaying 1 - 3

Here you will see a list of existing clusters you or other data engineers have created.

Click on Jobs in the left-hand pane.

Here you will see a list of previous jobs you have run. Since you are using a shared login, you will see past jobs that others have run as the testdrive@altus-demo.com user.

Cloudera Altus

Exercise 3: How to create a cluster

In this exercise, we will walk through the steps to create a data engineering Hive on Spark cluster. However, in the interest of time, we have created some clusters for you to use in the lab. This exercise will show you the create process but **please DO NOT CLICK CREATE** at the end of the exercise.

Click on Clusters in the left pane then Create Cluster in the main pane.

Fill out the information for building a cluster.

General Information:
Cluster Name: **leave blank (since we are not going to create a cluster at the end)**
Service Type: **Hive on Spark**
CDH Version: **CDH 5.15**
SDX Namespace: **altus-test-drive-namespace**
Environment: **AzureTestDrive3**

General Information

Cluster Name * ⓘ	Enter cluster name
Service Type * ⓘ	Spark 2.x ⬍
CDH Version * ⓘ	CDH 5.15 (Spark 2.2) ⬍
SDX Namespace ⓘ	altus-test-drive-namespace ⬍
Environment * ⓘ	☁ 🔒 AzureTestDrive (Region: eastus2) ▾

> ⓘ The environment you selected is configured for secure clusters. All clusters created using this environment will require authentication.

Cloudera **Altus**

Node Configuration:
Worker: **Set the Number of Nodes to 3** (instead of 5).

Node Configuration

Node Type	Instance Type ⓘ	Number of Nodes
Worker ⓘ	STANDARD_DS12_V2 28 GiB 4vCPU ▾	3
Master	STANDARD_DS12_V2 28 GiB 4vCPU	1
Cloudera Manager	STANDARD_DS12_V2 28 GiB 4vCPU	1

Credentials:
For the SSH Public Key, use any public key you have access to. If you don't currently have a public key, then click on "**Direct Input**" and copy and paste the below public key into the SSH Public Key box:

ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDbmWtQ9V8S41bBV83/VI8m+3DQwTxJYigroT30bzv7UYYZ7Guk+zW9DaHEeHdIOeO3RD
ovPzU2j0C2Zw6nGE2iRZ8KbPUAFQxUlkkhYetMPb+ILV6mB4y9Rf1nx//r8npFbuTnYcockIUZggMS+NRoQY+vFO3CATL1Lfrx0R9CCW
b4GfeAOaCpDHuPrthlf2s2QT//yxyf9++0S87ysqqC0Ox9csdmYMUgfPWWVzD4noOcu3nKySxxbNm0aAAHOd+H42+cnFjwxfgg/mr+8
PZOTot1LOqSNj5YXPORab/V9/+3pIO+AFA+08JESX75uy5AozoDKYXGaBBOZXGJxcfr poweradmin@altus-demo.com

Credentials

SSH Public Key * ⓘ   ○ File Upload   ● Direct Input

ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDbmWtQ9V8S41bBV83/VI8m+3DQwTxJYigroT3
0bzv7UYYZ7Guk+zW9DaHEeHdIOeO3RDovPzU2j0C2Zw6nGE2iRZ8KbPUAFQxUlkkhYetM
Pb+ILV6mB4y9Rf1nx
//r8npFbuTnYcockIUZggMS+NRoQY+vFO3CATL1Lfrx0R9CCWb4GfeAOaCpDHuPrthlf2s2Q
T
//yxyf9++0S87ysqqC0Ox9csdmYMUgfPWWVzD4noOcu3nKySxxbNm0aAAHOd+H42+cnFj
wxfgg/mr+8PZOTot1LOqSNj5YXPORab
/V9/+3pIO+AFA+08JESX75uy5AozoDKYXGaBBOZXGJxcfr poweradmin@altus-demo.com

Also pick a username and password for Cloudera Manager. You can use "**cloudera**" for both the username and password.

Credentials

SSH Public Key * ⓘ	⦿ File Upload ◯ Direct Input
	[ ] Choose File
Cloudera Manager ⓘ Username *	guest
Cloudera Manager ⓘ Password *	Type a password ⊕
Confirm Cloudera Manager Password *	Type the password again ⊕

Now click Cancel**. Please do NOT CLICK** "**Create Cluster**".

Cancel   **Create Cluster**

Cloudera **Altus**

Exercise 4: Preparing the object store

In this exercise, you will be familiarizing yourself with the data stored in ADLS. You will also created a subfolder (named after yourself) to host the output from an Altus job you will run.
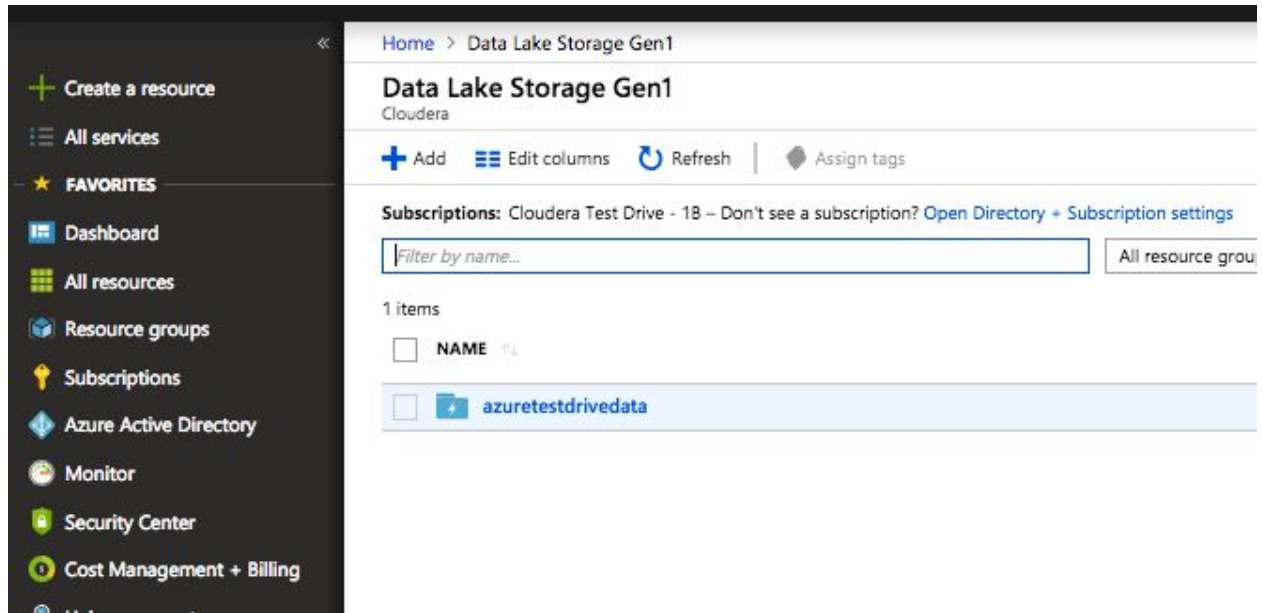
Once you've logged in to the Azure portal, go to the search bar at the top of the screen and type in "data lake storage"
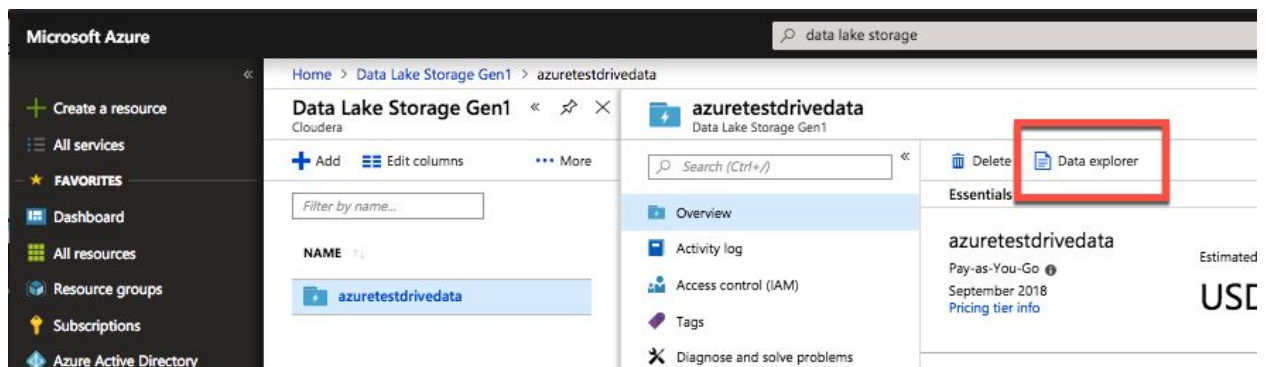

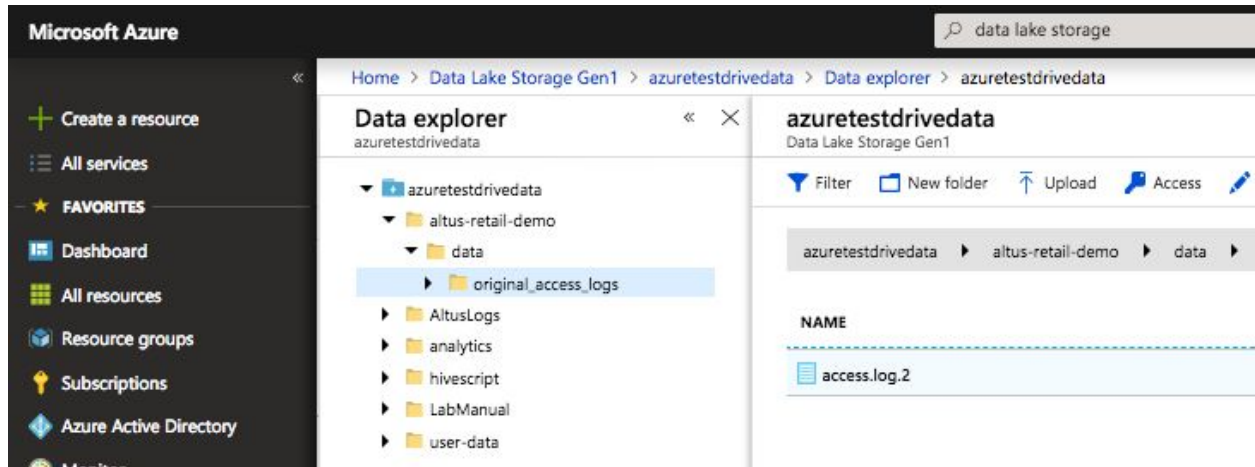Click on "Data Lake Storage Gen 1" under Services
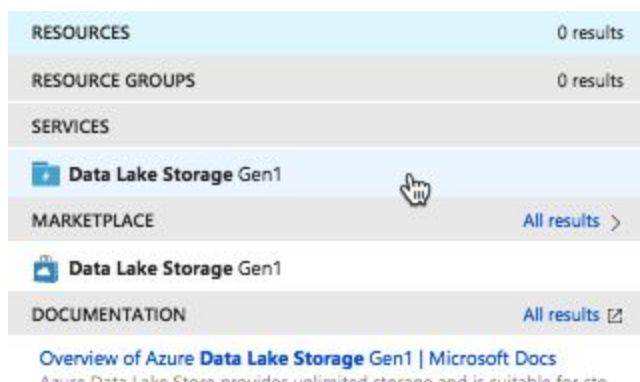
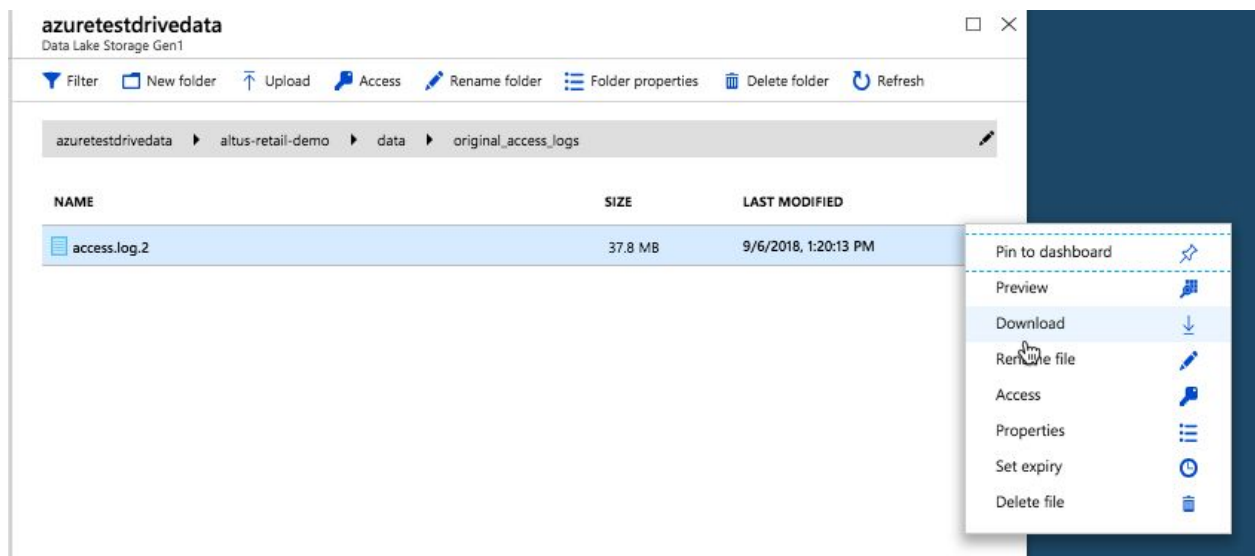Click on **azuretestdrivedata**

Click on "**Data Explorer**" which is in the right-most pane at the top:

Navigate to the **azuretestdrivedata** ADLS folder. Go into the
altus-retail-demo/data/original_access_logs folder.

Cloudera **Altus**

Inside the **original_access_logs** folder is **the access.log.2** file which contains weblog data. Download the file (by clicking on the three dots next to the filename) and open it up with your favourite text editor (notepad or text editor).

This file looks similar to below:

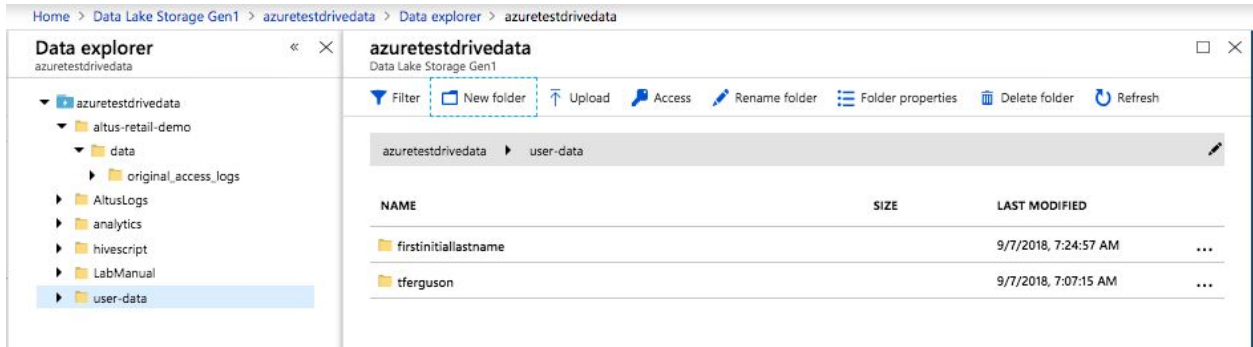Navigate back to the top-level **azuretestdrivedata** folder and go into **user-data** folder.
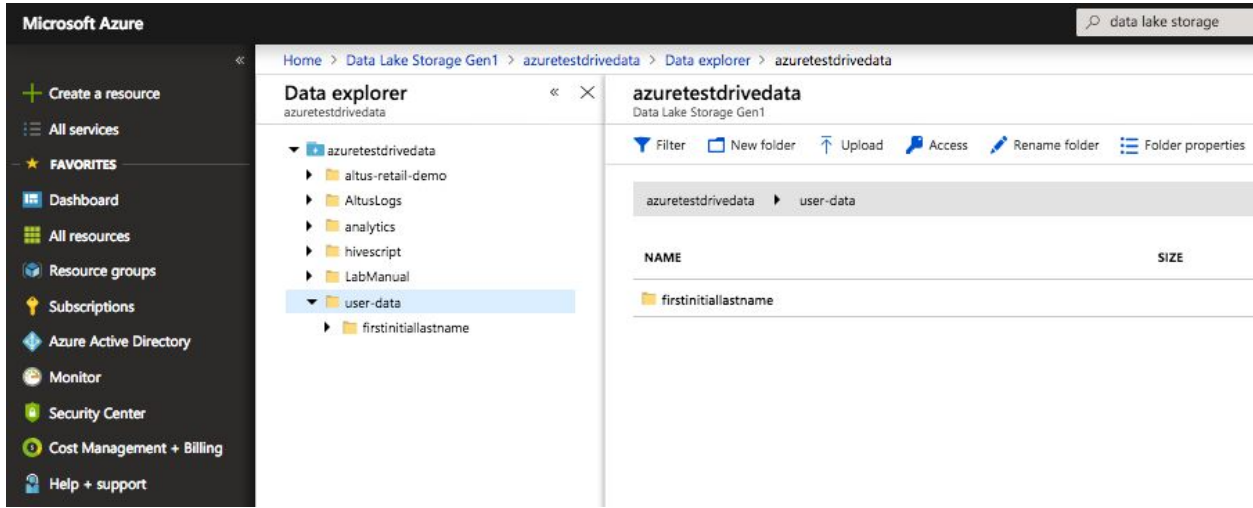
Create a sub-folder with your name. Click on "New Folder" and enter your short name, first Initial last name (all lowercase and no spaces). For example tferguson.

Navigate to the **hivescript** folder and download the **tokenize.sql** file to your laptop.
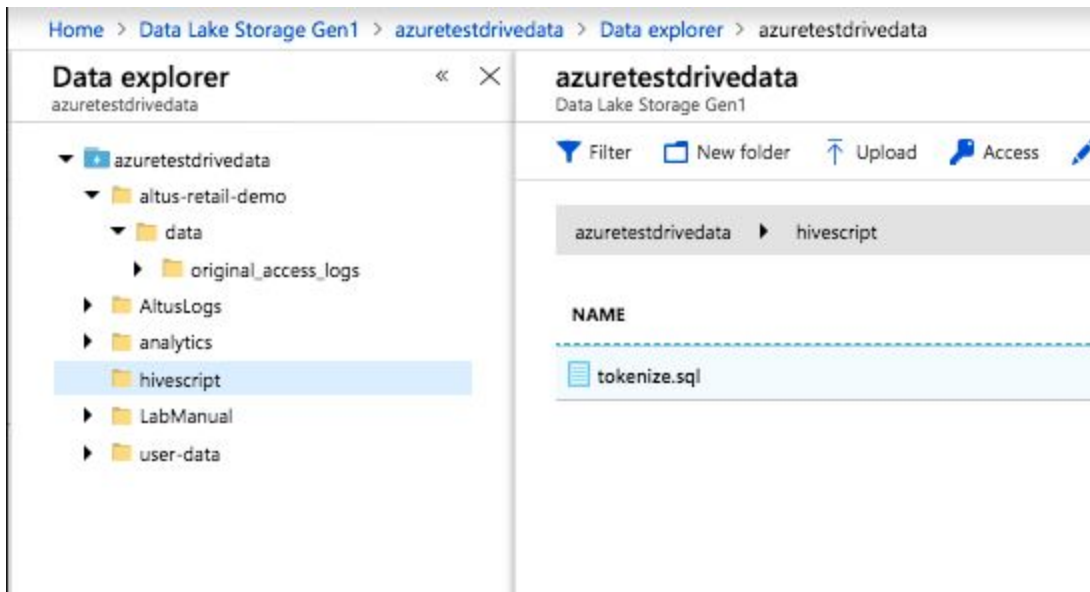
Open and review the tokenize.sql file. This is the Data Engineering query you are going to run.

```
-- Create your database
CREATE DATABASE IF NOT EXISTS ${YOURNAMEHERE};

use ${YOURNAMEHERE};

-- Create intermediate_access_logs table, drop if exists first
drop table if exists intermediate_access_logs;
CREATE EXTERNAL TABLE intermediate_access_logs (
    ip STRING,
    date STRING,
    method STRING,
    url STRING,
```

```
    http_version STRING,
    code1 STRING,
    code2 STRING,
    dash STRING,
    user_agent STRING)
ROW FORMAT SERDE 'org.apache.hadoop.hive.contrib.serde2.RegexSerDe'
WITH SERDEPROPERTIES (
    'input.regex' = '([^ ]*) - - \\[([^\\]]*)\\] "([^\ ]*) ([^\ ]*) ([^\ ]*)" (\\d*) (\\d*)
"([^"]*)" "([^"]*)"',
    'output.format.string' = "%1$$s %2$$s %3$$s %4$$s %5$$s %6$$s %7$$s %8$$s %9$$s")
LOCATION
'adl://azuretestdrivedata.azuredatalakestore.net/altus-retail-demo/data/original_access_logs
';

-- Create tokenized_access_logs table, drop if exists first
drop table if exists tokenized_access_logs;
CREATE EXTERNAL TABLE tokenized_access_logs (
    ip STRING,
    date STRING,
    method STRING,
    url STRING,
    http_version STRING,
    code1 STRING,
    code2 STRING,
    dash STRING,
    user_agent STRING)
stored as parquet
LOCATION
'adl://azuretestdrivedata.azuredatalakestore.net/user-data/${YOURNAMEHERE}/tokenized_access_
logs';

INSERT OVERWRITE TABLE tokenized_access_logs SELECT * FROM intermediate_access_logs;
```

Exercise 5: Running a Job.

In this exercise, you will use pre-created clusters to run your job. Ask the instructor for
guidelines on which clusters to use. The job you are about to run will take
semi-structured web log data and will transform it into a structured format for SQL
querying.

Click on **Jobs**, then **Submit Jobs**.

Submission: Single Job
Job type: Hive
Job Name: use your name
Script: Simply select 'direct input' and copy/pasted the tokenize.sql code
Hive Script Parameters: Add one parameter with variable name - YOURNAMEHERE and value with sub-folder name you previously created in ADLS. This name will also be a database name.
Job XML: None
Action on Failure: None
Cluster: Use the cluster name you got in the email.

Click on Submit Jobs. The job will take ~3 mins to complete. If your job fails, you can try cloning the job and rerunning it or contact the lab instructor if you need help debugging the problem.

Exercise 6: Workload Analytics

Click on your completed job. Here you can view details about the job you just ran.

Click on **Analytics**. It might take a few minutes to appear.

Click on **Execution Details** and then click on any one of the SQL queries. Then click on **View Query**.

Looks familiar? It's part of the SQL script that you ran.

create external table tokenize...

Query

```
CREATE EXTERNAL TABLE tokenized_access_logs (
  ip STRING,
  date STRING,
  method STRING,
  URL STRING,
  http_version STRING,
  code1 STRING,
  code2 STRING,
  dash STRING,
  user_agent STRING
) STORED AS parquet location '*****'
```

Click on Baseline to view more info about your job. Note that since this is your first time running a job, Altus Workload Analytics does not have enough info from previous runs to create the baseline.

Jobs
stefan-salandy   [Actions ▾]   Details   **Analytics**

Health Checks   Execution Details   **Baseline**   Trends   📅 1/28/2018 4:24 PM   ⧗ 1m 6s

All Stages / stefan-salandy ▾

Metric ⓘ	Current	Baseline ⓘ
Active Tasks	0	
Disk bytes Spilled	0 B	
Duration	1m 6s	
Executor Runtime	6s	
Failed Task attempts	0	

For example purposes we have included screenshots of previously run jobs. These jobs have Baseline and Trend analysis:

Health Checks   Execution Details   **Baseline**   Trends   📅 2/7/2018 9:47 AM   ⧗ 1m 40s   👤 Stefan Salandy

All Stages / stefan-salandy-retail-clickstream ▾

Metric ⓘ	Current	Baseline ⓘ	Difference	
Duration	1m 40s	1m 1s	+40s	+65%
Total memory-time taken by all map tasks	452.4K MiB·s	320.2K MiB·s	+132.2K MiB·s	+41%
Total time spent by all map tasks	1m 20s	57s	+23s	+41%
Total time spent by all maps in occupied slots	8m 2s	5m 41s	+2m 21s	+41%
Total vcore-time taken by all map tasks	1m 20s	57s	+23s	+41%
Executor Runtime	6s	9s	-3s	-33%
Total Task duration	8s	11s	-3s	-27%
GC time elapsed	< 1s	< 1s	+< 1s	+18%
Physical memory snapshot	357.5 MiB	371.9 MiB	-14.4 MiB	-4%

Cloudera **Altus**

Duration	Data Processed	Data Generated
1m 40s 47s 35s	157 KiB 5 KiB	199 B

End Time	Duration	Status	Health Issues
2/7/2018 9:48 AM	1m 40s	⊘ Completed	None
1/17/2018 10:58 AM	1m 14s	⊘ Completed	None
1/16/2018 9:25 PM	1m 37s	⊘ Completed	None
1/16/2018 1:42 PM	35s	⊘ Completed	None

Finally, if the Altus cluster has not yet been terminated, you can navigate to Cloudera Manager and view the queries that were run from the Hive server web UI. **For the purposes of this lab, we have not configured the network to allow access to Cloudera Manager for the attendees.** Instead, please review the below screenshot showing the query from the Hive server UI.

Home   Local logs   Metrics Dump   Hive Configuration   Stack Trace

| User Name	Query	Execution Engine	State	Opened (s)	Closed Timestamp	Latency (s)	Drilldown Link
hdfs	drop table if exists intermediate_access_logs	mr	FINISHED	2	Sun Feb 11 16:49:06 UTC 2018	2	Drilldown
hdfs	CREATE EXTERNAL TABLE intermediate_access_logs (ip STRING, date STRING, method STRING, url STRING, http_version STRING, code1 STRING, code2 STRING, dash STRING, user_agent STRING) ROW FORMAT SERDE 'org.apache.hadoop.hive.contrib.serde2.RegexSerDe' WITH SERDEPROPERTIES ('input.regex' = '([^]*) - - \\[([^\\]]*)\\] "([^\ ]*) ([^\ ]*) ([^\ ]*)" (\\d*) (\\d*) "([^"]*)" "([^"]*)"', 'output.format.string' = "%1$$s %2$$s %3$$s %4$$s %5$$s %6$$s %7$$s %8$$s %9$$s")	mr	FINISHED	3	Sun Feb 11 16:49:10 UTC 2018	3	Drilldown

## Exercise 7: Combine Structured and UnStructured data

-- YOU MIGHT NOT HAVE SUFFICIENT PERMISSIONS TO PERFORM THIS LAB.

Altus Data Warehouse (DW) clusters are used to run Analytics. A SQL Editor is built in and that allows you to run queries directly, or you can plug in a BI tool and use that if you prefer.
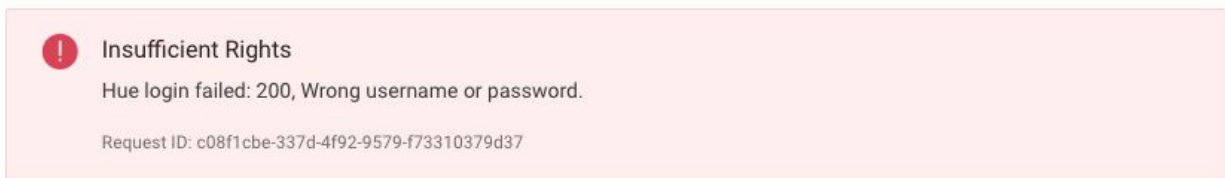
In this lab we'll go to the SQL Editor:
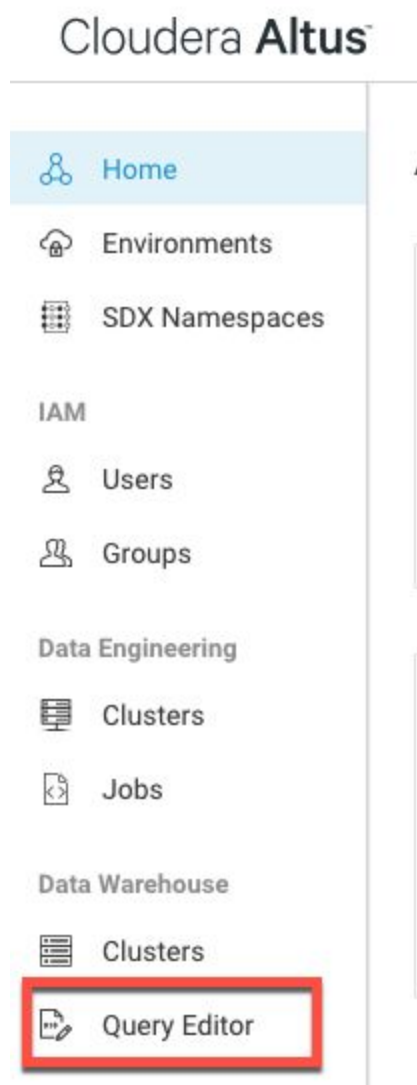
and choose the DW cluster we've created **test-drive-instructor-dw**:
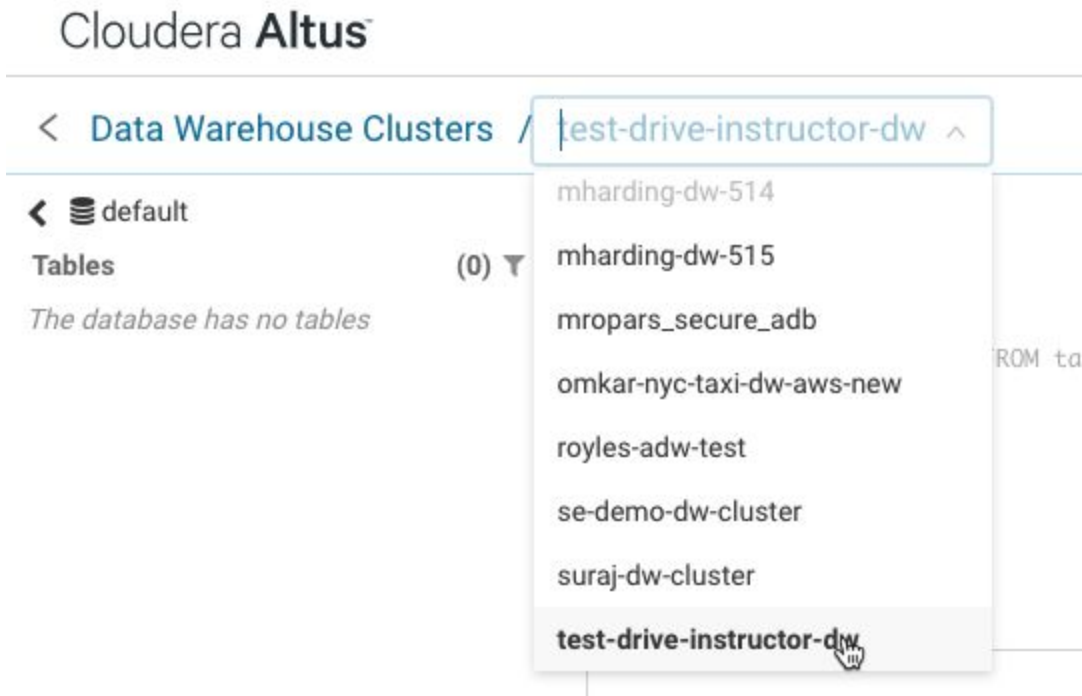
If you see this:

then just refresh your screen and you'll be into the SQL Editor.

So now we're ready to run a query. But what are we querying? If we go back to Azure you can see that the structured warehouse data (i.e. the data about customers, orders etc) is stored in the ADLS account and mapped to tables:

These tables are in the SDX namespace, altus-test-drive-namespace, under the **retaildb** already. You might have to invalidate the impala metadata to see the retaildb and the tables. (If you can see the **retaildb** and its tables you can skip this bit!):

Click the arrow to the right of **default:**

and you'll get to the 'top' of the database server, looking at all the possible databases:
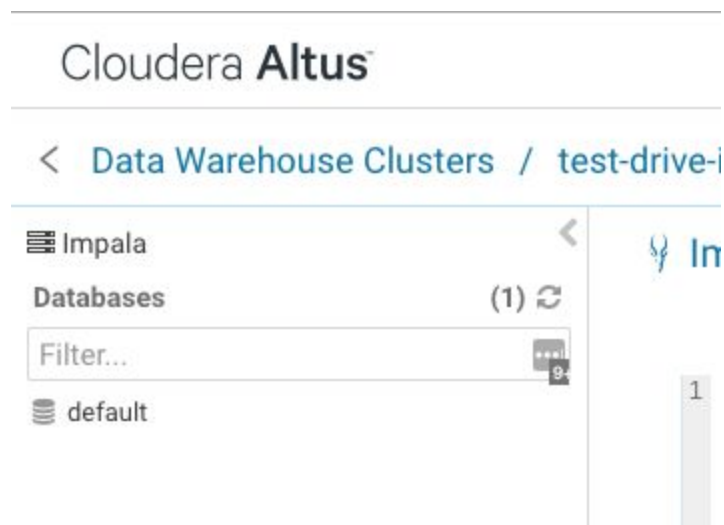
Right now you only see **default.** Select the little circle and choose 'invalidate metadata':

and you should now see the **retaildb**:
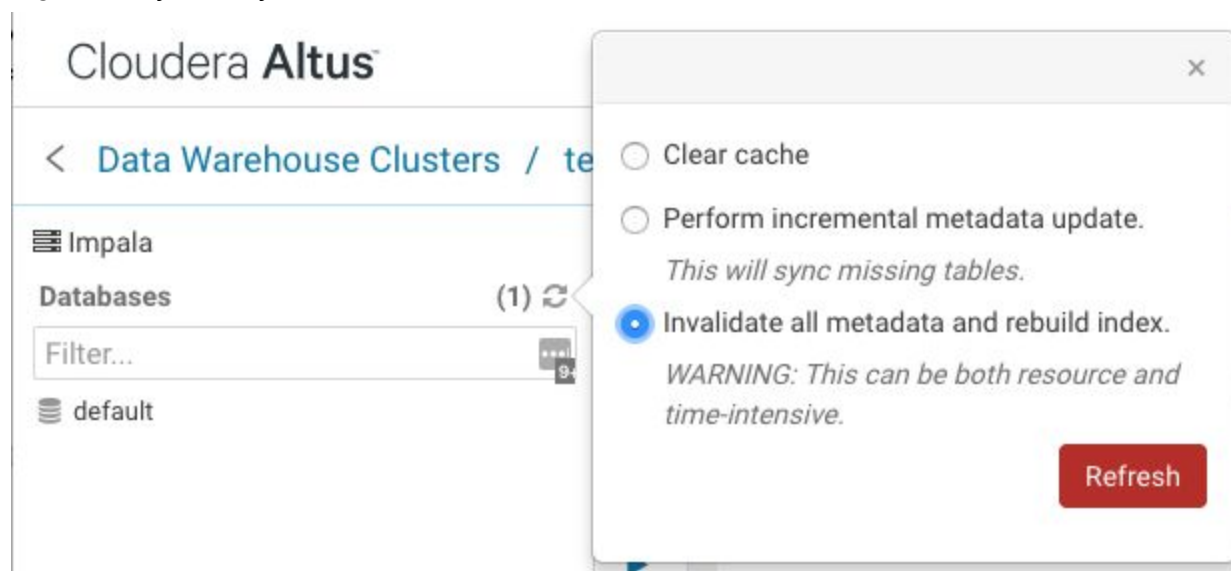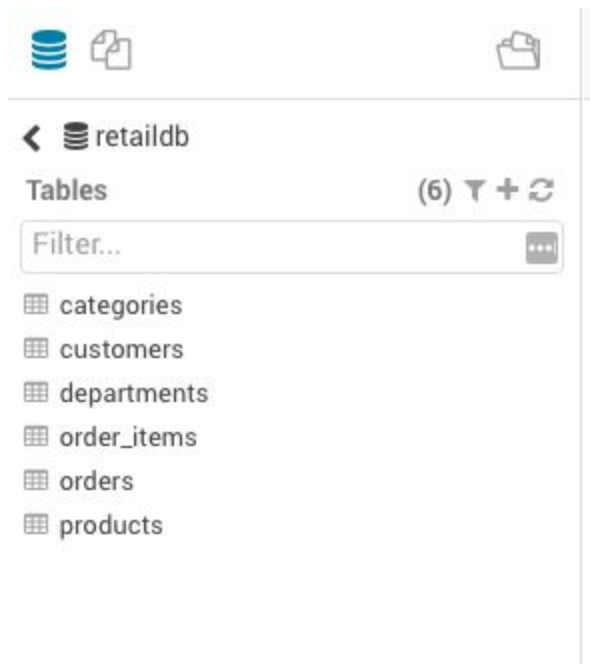
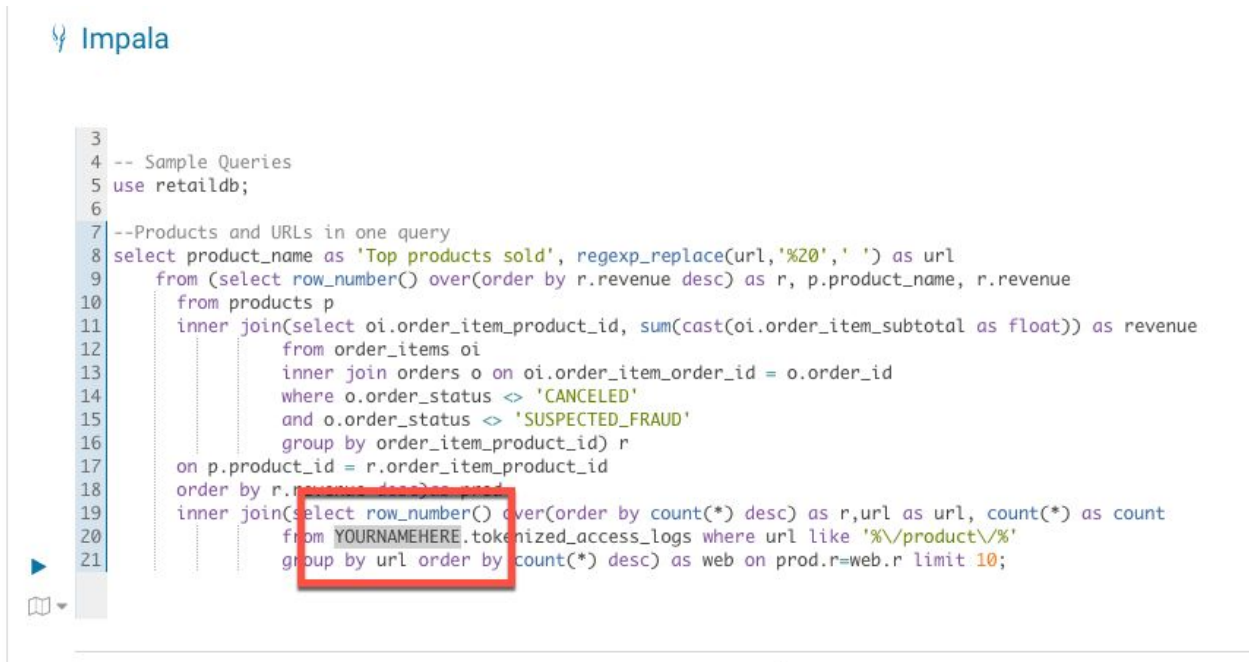Click on **retaildb** and you should see the tables:

Copy the following SQL query into the SQL Editor:

```
-- Refresh the metadata
invalidate metadata;

-- Sample Queries
use retaildb;

--Products and URLs in one query
select product_name as 'Top products sold', regexp_replace(url,'%20',' ') as url
    from (select row_number() over(order by r.revenue desc) as r, p.product_name, r.revenue
      from products p
      inner join(select oi.order_item_product_id, sum(cast(oi.order_item_subtotal as float))
as revenue
                 from order_items oi
                 inner join orders o on oi.order_item_order_id = o.order_id
                 where o.order_status <> 'CANCELED'
                 and o.order_status <> 'SUSPECTED_FRAUD'
                 group by order_item_product_id) r
      on p.product_id = r.order_item_product_id
      order by r.revenue desc)as prod
      inner join(select row_number() over(order by count(*) desc) as r,url as url, count(*)
as count
                 from YOURNAMEHERE.tokenized_access_logs where url like '%\/product\/%'
                 group by url order by count(*) desc) as web on prod.r=web.r limit 10;
```

Before you run it, change the **YOURNAMEHERE** string (2 lines from the bottom) to your first initial/last name, just like you did earlier:

Once you've done that, highlight ALL the text (Ctl-A on a Windows machine; Command A on a Mac, or drag with your mouse), and then click the little blue arrow to execute everything:

Ψ Impala

```
1  -- Refresh the metadata
2  invalidate metadata;
3
4  -- Sample Queries
5  use retaildb;
6
7  --Products and URLs in one query
8  select product_name as 'Top products sold', regexp_replace(url,'%20',' ') as url
9      from (select row_number() over(order by r.revenue desc) as r, p.product_name, r.revenue
10       from products p
11       inner join(select oi.order_item_product_id, sum(cast(oi.order_item_subtotal as float)) as revenue
12              from order_items oi
13              inner join orders o on oi.order_item_order_id = o.order_id
14              where o.order_status <> 'CANCELED'
15              and o.order_status <> 'SUSPECTED_FRAUD'
16              group by order_item_product_id) r
17       on p.product_id = r.order_item_product_id
18       order by r.revenue desc)as prod
19       inner join(select row_number() over(order by count(*) desc) as r,url as url, count(*) as count
20              from tferguson.tokenized_access_logs where url like '%\/product\/%'
21              group by url order by count(*) desc) as web on prod.r=web.r limit 10;
```

and you should see the results from querying the structured and unstructured data:

Query ▾

Search data and saved documents...

Impala   ↺   Add a name...   Add a description...

5.88s ≣ retaildb ▾ text ▾ ⚙ ?

≣ retaildb

Tables   (6) ▼ + ⟳

Filter...

- categories
- customers
- departments
- order_items
- orders
- products

```
1  -- Sample Queries
2  use retaildb;
3
4  --Products and URLs in one query
5  select product_name as 'Top products sold', regexp_replace(url,'%20',' ') as url
6      from (select row_number() over(order by r.revenue desc) as r, p.product_name, r.revenue
7          from products p
8          inner join(select oi.order_item_product_id, sum(cast(oi.order_item_subtotal as float)) as revenue
9                  from order_items oi
10                 inner join orders o on oi.order_item_order_id = o.order_id
11                 where o.order_status <> 'CANCELED'
12                 and o.order_status <> 'SUSPECTED_FRAUD'
13                 group by order_item_product_id) r
14     on p.product_id = r.order_item_product_id
15     order by r.revenue desc)as prod
16     inner join(select row_number() over(order by count(*) desc) as r,url as url, count(*) as count
17             from spabba.tokenized_access_logs where url like '%\/product\/%'
18             group by url order by count(*) desc) as web on prod.r=web.r limit 10;
19
20
```

2/2
▶

Query History 🔍 📅   Saved Queries 🔍   Results (10) 🔍 ⤢

	top products sold	url
1	Field & Stream Sportsman 16 Gun Fire Safe	/department/apparel/category/cleats/product/Perfect Fitness Perfect Rip Deck
2	Perfect Fitness Perfect Rip Deck	/department/apparel/category/featured shops/product/adidas Kids' RG III Mid Football Cleat
3	Diamondback Women's Serene Classic Comfort Bi	/department/golf/category/women's apparel/product/Nike Men's Dri-FIT Victory Golf Polo
4	Nike Men's Free 5.0+ Running Shoe	/department/apparel/category/men's footwear/product/Nike Men's CJ Elite 2 TD Football Cleat
5	Nike Men's Dri-FIT Victory Golf Polo	/department/fan shop/category/water sports/product/Pelican Sunstream 100 Kayak
6	Pelican Sunstream 100 Kayak	/department/fan shop/category/indoor/outdoor games/product/O'Brien Men's Neoprene Life Vest
7	O'Brien Men's Neoprene Life Vest	/department/fan shop/category/camping & hiking/product/Diamondback Women's Serene Classic (
8	Nike Men's CJ Elite 2 TD Football Cleat	/department/fan shop/category/fishing/product/Field & Stream Sportsman 16 Gun Fire Safe
9	Under Armour Girls' Toddler Spine Surge Runni	/department/footwear/category/cardio equipment/product/Nike Men's Free 5.0+ Running Shoe
10	adidas Youth Germany Black/Red Away Match Soc	/department/footwear/category/fitness accessories/product/Under Armour Hustle Storm Medium [

Conclusion: Visualizing the results.

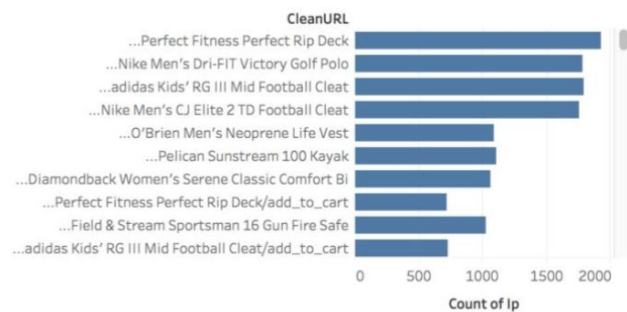Your instructor will show you a visualization of the results of your job using a BI Tool.
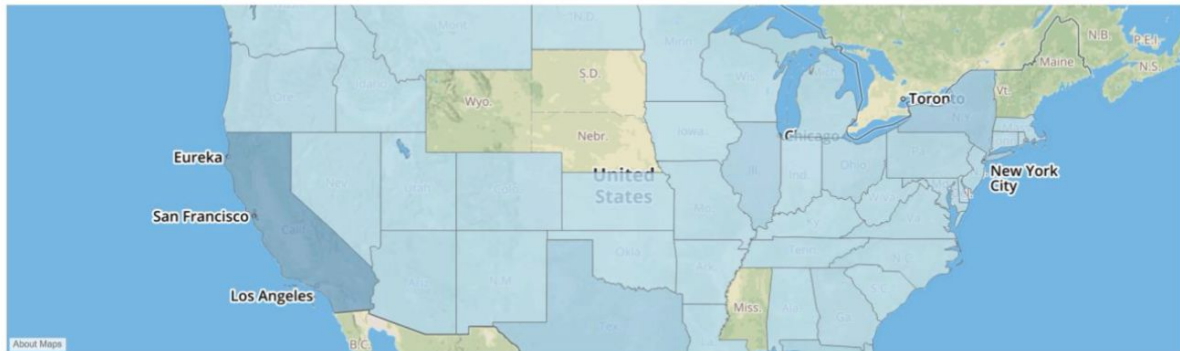
This brings us to the end of the lab. We've discussed the value Cloudera Altus on Azure brings to data engineers and business intelligence users. Cloudera Altus allows end users to be self-sufficient and easily correlate disparate data sources for faster insights. Thank you for your time and patience!