

# CST8921 – Cloud Industry Trends

## Lab 4 Report

### Title

Analyzing Data with Azure Databricks: A Hands-On Exploration.

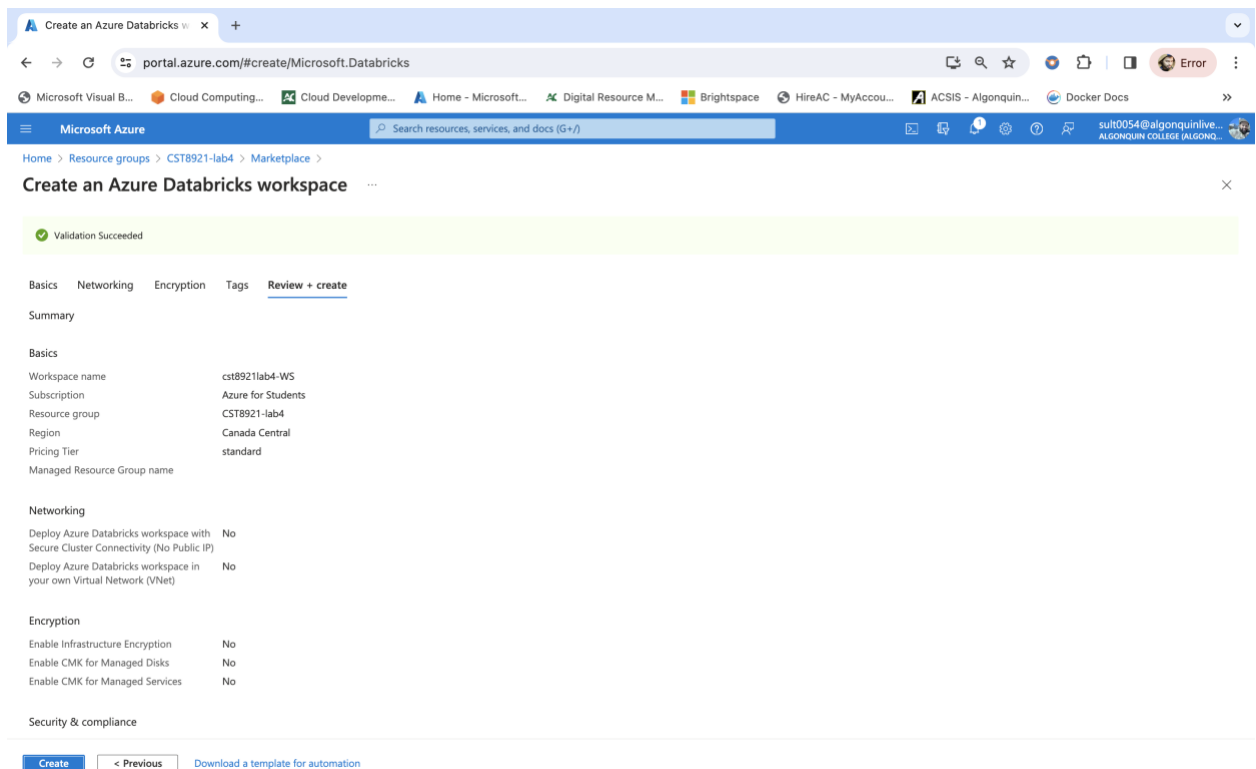
### Introduction

Embark on a journey to master Azure Databricks, an Apache Spark-powered analytics platform. In this lab we will provision a workspace, analyze data using Spark, explore Delta Lake functionality, and execute Databricks notebooks seamlessly from Azure Data Factory.

### Steps

#### Part 1: Explore databricks notebooks

1. Provision azure databricks workspace environment.



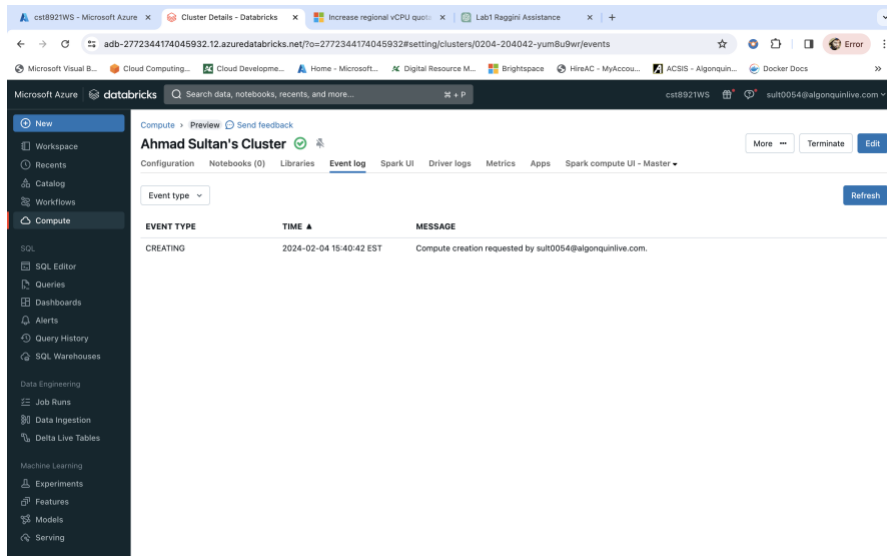
The screenshot shows the Azure portal interface for creating a new Azure Databricks workspace. The browser address bar indicates the URL is `portal.azure.com/#create/Microsoft.Databricks`. The page title is "Create an Azure Databricks workspace". A green banner at the top indicates "Validation Succeeded". Below this, there are tabs for "Basics", "Networking", "Encryption", "Tags", and "Review + create". The "Review + create" tab is selected, showing a summary of the workspace configuration.

Section	Property	Value
Basics	Workspace name	cst8921lab4-WS
	Subscription	Azure for Students
	Resource group	CST8921-lab4
	Region	Canada Central
	Pricing Tier	standard
Networking	Deploy Azure Databricks workspace with Secure Cluster Connectivity (No Public IP)	No
	Deploy Azure Databricks workspace in your own Virtual Network (VNet)	No
	Encryption	
Encryption	Enable Infrastructure Encryption	No
	Enable CMK for Managed Disks	No
	Enable CMK for Managed Services	No
Security & compliance		

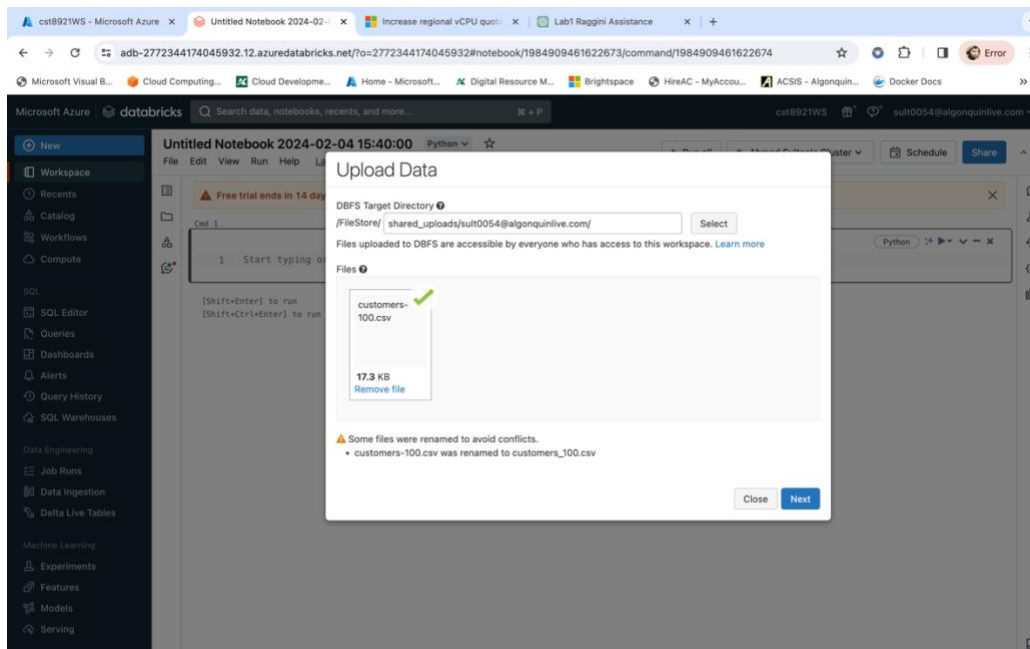
At the bottom of the page, there is a "Create" button, a "< Previous" button, and a link to "Download a template for automation".

## 2. Create a single node cluster in the workspace

Note : Azure Databricks is a distributed processing platform that uses Apache Spark clusters to process data in parallel on multiple nodes. Each cluster consists of a driver node to coordinate the work, and worker nodes to perform processing tasks.



## 3. Use Spark to analyze data file – create a notebook to explore data. Download sample file from this link: <https://www.datablist.com/learn/csv/download-sample-csv-files>. Upload the file downloaded to DBFS directory in the workspace.



6. In the Access files from notebooks pane, select the sample PySpark code and copy it to the clipboard. You will use it to load the data from the file into a DataFrame. Then select Done. While exploring notebook load data in dataframe, change the file name from products to the file name you have uploaded in dbfs. Please change the below code as per your configuration in databricks notebook. **`df1=spark.read.format("csv").option("header", "true").load("dbfs:/FileStore/shared_uploads/user@outlook.com/products.csv")`**

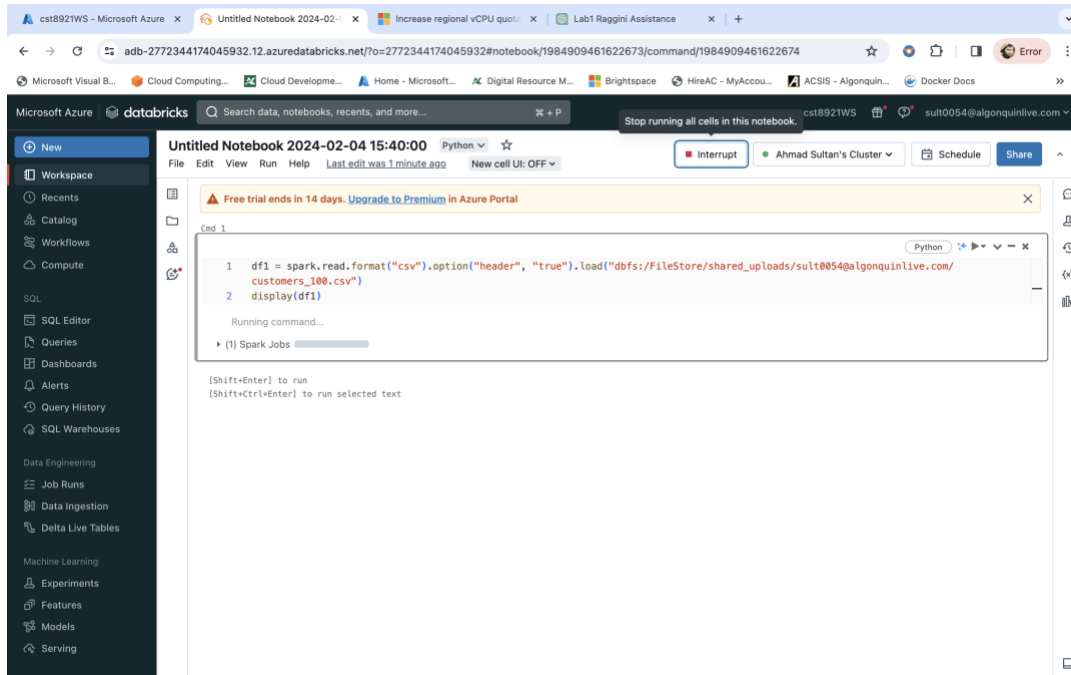
The screenshot displays the Databricks web interface. The browser's address bar shows the URL: `adb-2772344174045932.12.azuredatabricks.net/?o=2772344174045932#notebook/1984909461622673/command/1984909461622674`. The notebook is titled "Untitled Notebook 2024-02-04 15:40:00" and is running on "Ahmad Sultan's Cluster". A sidebar on the left contains navigation options like Workspace, Recents, Catalog, Workflows, Compute, SQL, and Machine Learning. The main editor area shows a code cell with the following PySpark code:

```
1 df1 = spark.read.format("csv").option("header", "true").load("dbfs:/FileStore/shared_uploads/sult0054@algonquinlive.com/
  customers_100.csv")
2 display(df1)
```

Below the code cell, instructions indicate that pressing `[Shift+Enter]` will run the entire cell, while `[Shift+Ctrl+Enter]` will run only the selected text. A yellow banner at the top of the code editor area states: "Free trial ends in 14 days. Upgrade to Premium in Azure Portal".

8. Explore the data using `display(df1)` command and visualize the results in notebook.

`display(df1)`



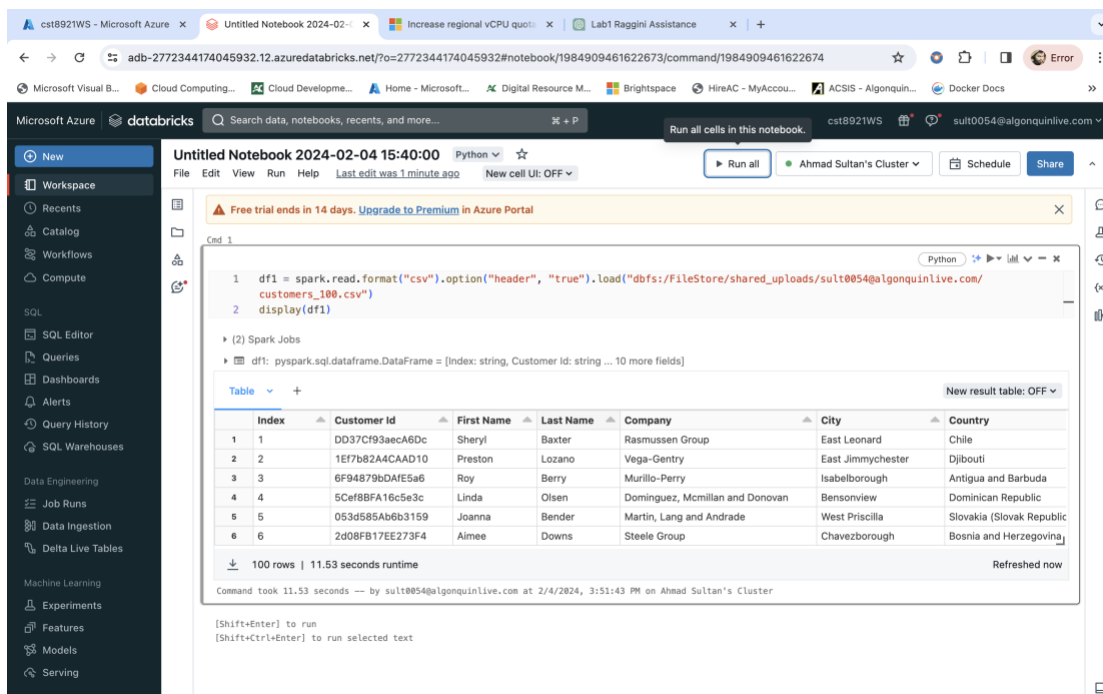
Untitled Notebook 2024-02-04 15:40:00 Python

```
1 df1 = spark.read.format("csv").option("header", "true").load("dbfs:/FileStore/shared_uploads/sult0054@algonquinlive.com/customers_100.csv")
2 display(df1)
```

Running command...

(1) Spark Jobs

[Shift+Enter] to run  
[Shift+Ctrl+Enter] to run selected text



Untitled Notebook 2024-02-04 15:40:00 Python

```
1 df1 = spark.read.format("csv").option("header", "true").load("dbfs:/FileStore/shared_uploads/sult0054@algonquinlive.com/customers_100.csv")
2 display(df1)
```

(2) Spark Jobs

df1: pyspark.sql.dataframe.DataFrame = [Index: string, Customer Id: string ... 10 more fields]

Index	Customer Id	First Name	Last Name	Company	City	Country
1	DD37Cf93aecA6Dc	Sheryl	Baxter	Rasmussen Group	East Leonard	Chile
2	1EF7b82A4CAAD10	Preston	Lozano	Vega-Gentry	East Jimmychester	Djibouti
3	6F94879bDAfE5a6	Roy	Berry	Murillo-Perry	Isabelborough	Antigua and Barbuda
4	5Cef8BFA16c5e3c	Linda	Olsen	Dominguez, Mcmillan and Donovan	Bensonview	Dominican Republic
5	053d585Ab6b3159	Joanna	Bender	Martin, Lang and Andrade	West Priscilla	Slovakia (Slovak Republic)
6	2d08FB17EE273F4	Almee	Downs	Steele Group	Chavezborough	Bosnia and Herzegovina

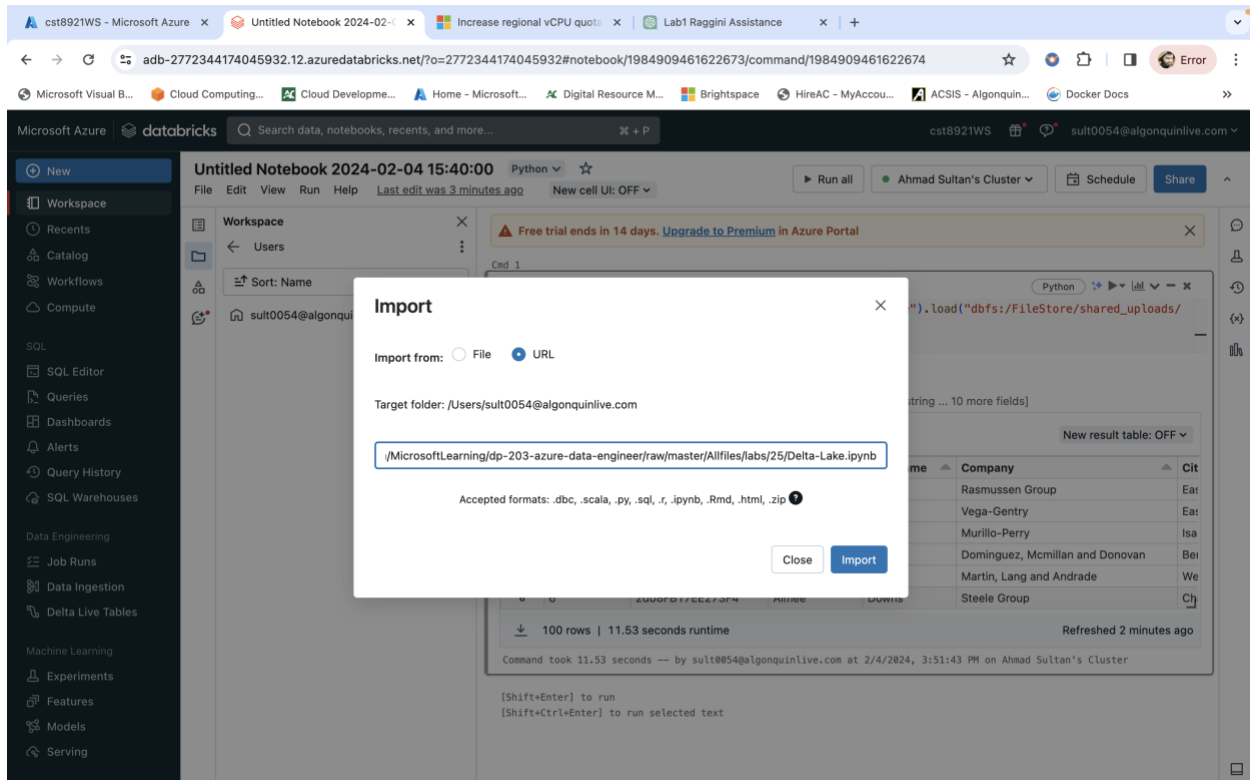
100 rows | 11.53 seconds runtime

Command took 11.53 seconds — by sult0054@algonquinlive.com at 2/4/2024, 3:51:43 PM on Ahmad Sultan's Cluster

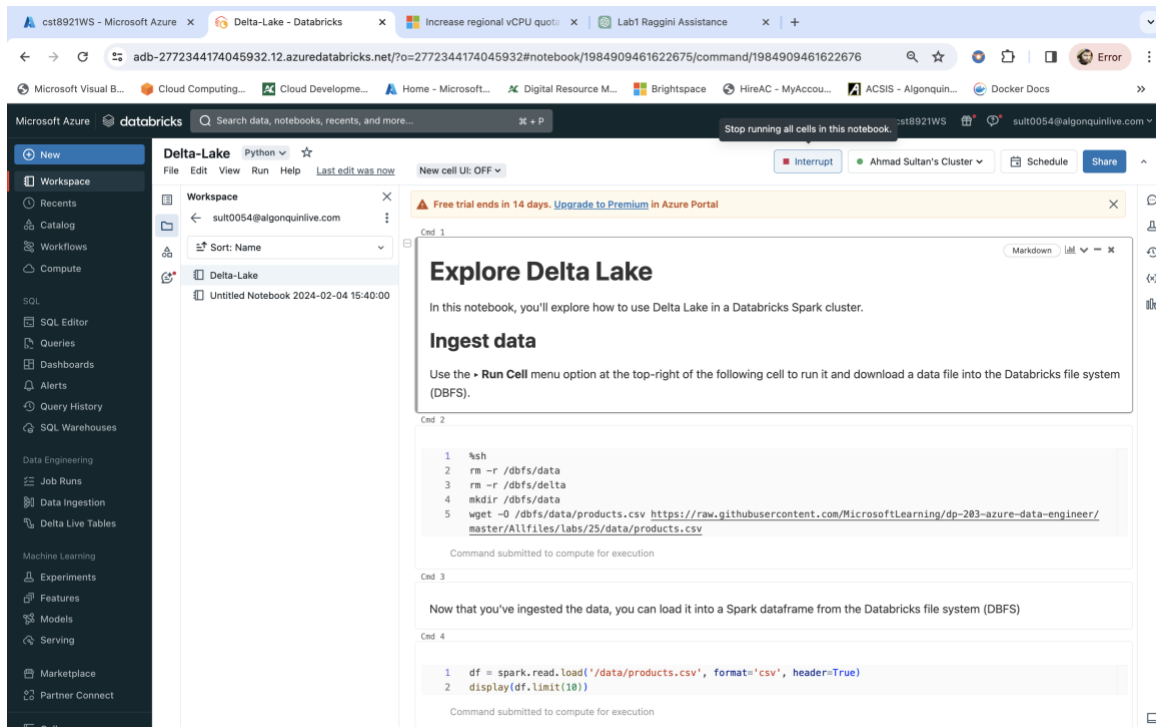
[Shift+Enter] to run  
[Shift+Ctrl+Enter] to run selected text

## Part 2: Use Delta Lake in Azure Databricks

1. In the Azure Databricks workspace portal for your workspace, in the sidebar on the left, select Workspace. Then select the Home folder.
2. At the top of the page, in the **:** menu next to your user name, select Import. Then in the Import dialog box, select URL and import the notebook from <https://github.com/MicrosoftLearning/dp-203-azure-data-engineer/raw/master/Allfiles/labs/25/Delta-Lake.ipynb>



2. Connect the notebook to your cluster, and follow the instructions it contains; running the cells it contains to explore delta lake functionality.



**Explore Delta Lake**

In this notebook, you'll explore how to use Delta Lake in a Databricks Spark cluster.

**Ingest data**

Use the **Run Cell** menu option at the top-right of the following cell to run it and download a data file into the Databricks file system (DBFS).

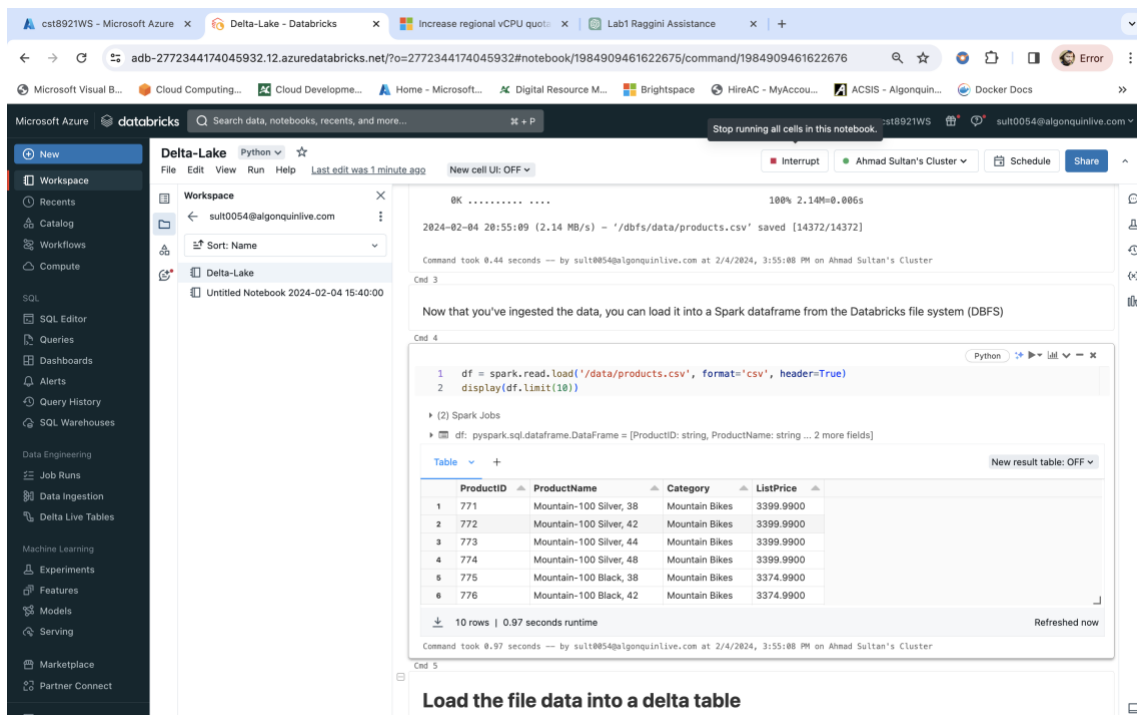
```
1 %sh
2 rm -r /dbfs/data
3 rm -r /dbfs/delta
4 mkdir /dbfs/data
5 wget -O /dbfs/data/products.csv https://raw.githubusercontent.com/MicrosoftLearning/dp-283-azure-data-engineer/master/Allfiles/labs/25/data/products.csv
```

Command submitted to compute for execution

Now that you've ingested the data, you can load it into a Spark dataframe from the Databricks file system (DBFS)

```
1 df = spark.read.load('/data/products.csv', format='csv', header=True)
2 display(df.limit(10))
```

Command submitted to compute for execution



0K ..... 100% 2.14M=0.006s

2024-02-04 20:55:09 (2.14 MB/s) - '/dbfs/data/products.csv' saved [14372/14372]

Command took 0.44 seconds --- by sult0054@algonquinlive.com at 2/4/2024, 3:55:08 PM on Ahmad Sultan's Cluster

Now that you've ingested the data, you can load it into a Spark dataframe from the Databricks file system (DBFS)

```
1 df = spark.read.load('/data/products.csv', format='csv', header=True)
2 display(df.limit(10))
```

Python

(2) Spark Jobs

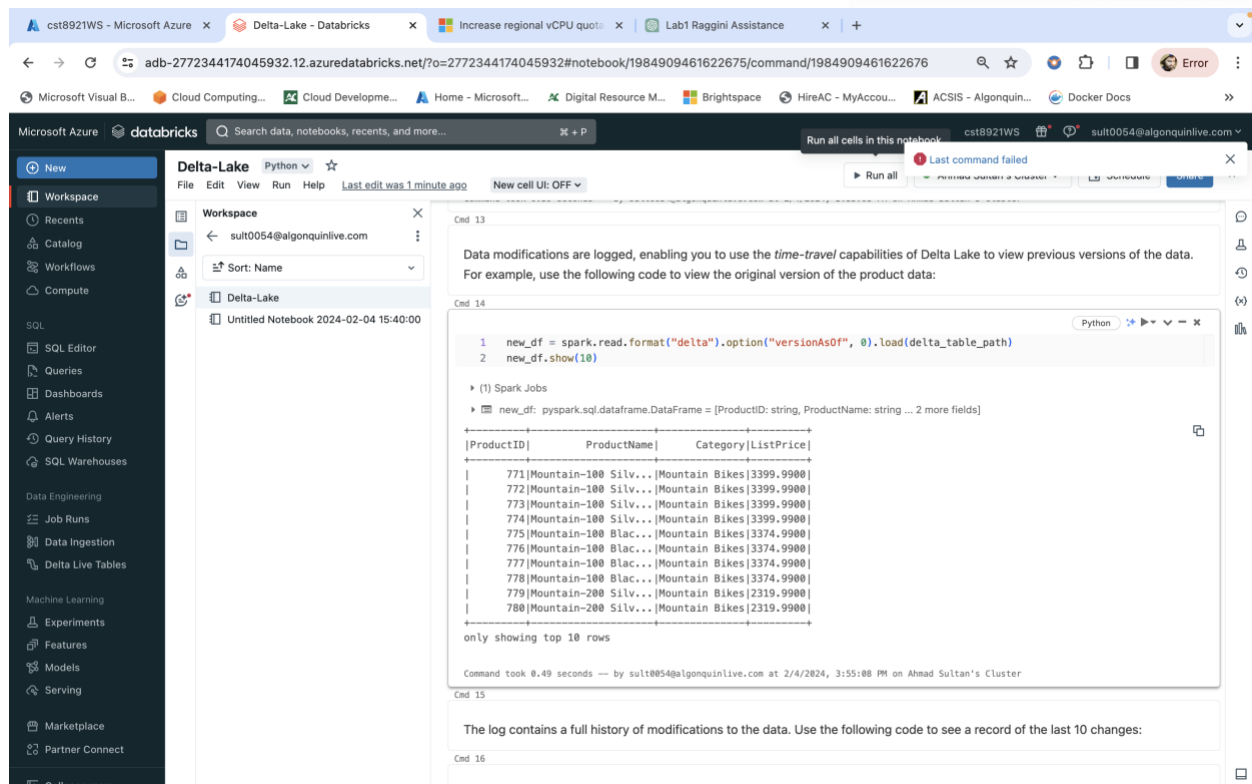
df: pyspark.sql.dataframe.DataFrame = [ProductID: string, ProductName: string ... 2 more fields]

ProductID	ProductName	Category	ListPrice
1 771	Mountain-100 Silver, 38	Mountain Bikes	3399.9900
2 772	Mountain-100 Silver, 42	Mountain Bikes	3399.9900
3 773	Mountain-100 Silver, 44	Mountain Bikes	3399.9900
4 774	Mountain-100 Silver, 48	Mountain Bikes	3399.9900
5 775	Mountain-100 Black, 38	Mountain Bikes	3374.9900
6 776	Mountain-100 Black, 42	Mountain Bikes	3374.9900

10 rows | 0.97 seconds runtime

Command took 0.97 seconds --- by sult0054@algonquinlive.com at 2/4/2024, 3:55:08 PM on Ahmad Sultan's Cluster

**Load the file data into a delta table**



The screenshot shows a Databricks notebook titled "Delta-Lake" with a Python environment. The notebook content includes a command to read data from a Delta table and display the top 10 rows. The output shows a table with columns: ProductID, ProductName, Category, and ListPrice. The data includes products like Mountain Bikes and Mountain Bikes with various specifications and prices.

```

1 new_df = spark.read.format("delta").option("versionAsOf", 0).load(delta_table_path)
2 new_df.show(10)

```

ProductID	ProductName	Category	ListPrice
771	Mountain-100 Silv...	Mountain Bikes	3399.9900
772	Mountain-100 Silv...	Mountain Bikes	3399.9900
773	Mountain-100 Silv...	Mountain Bikes	3399.9900
774	Mountain-100 Silv...	Mountain Bikes	3399.9900
775	Mountain-100 Blac...	Mountain Bikes	3374.9900
776	Mountain-100 Blac...	Mountain Bikes	3374.9900
777	Mountain-100 Blac...	Mountain Bikes	3374.9900
778	Mountain-100 Blac...	Mountain Bikes	3374.9900
779	Mountain-200 Silv...	Mountain Bikes	2319.9900
780	Mountain-200 Silv...	Mountain Bikes	2319.9900

only showing top 10 rows

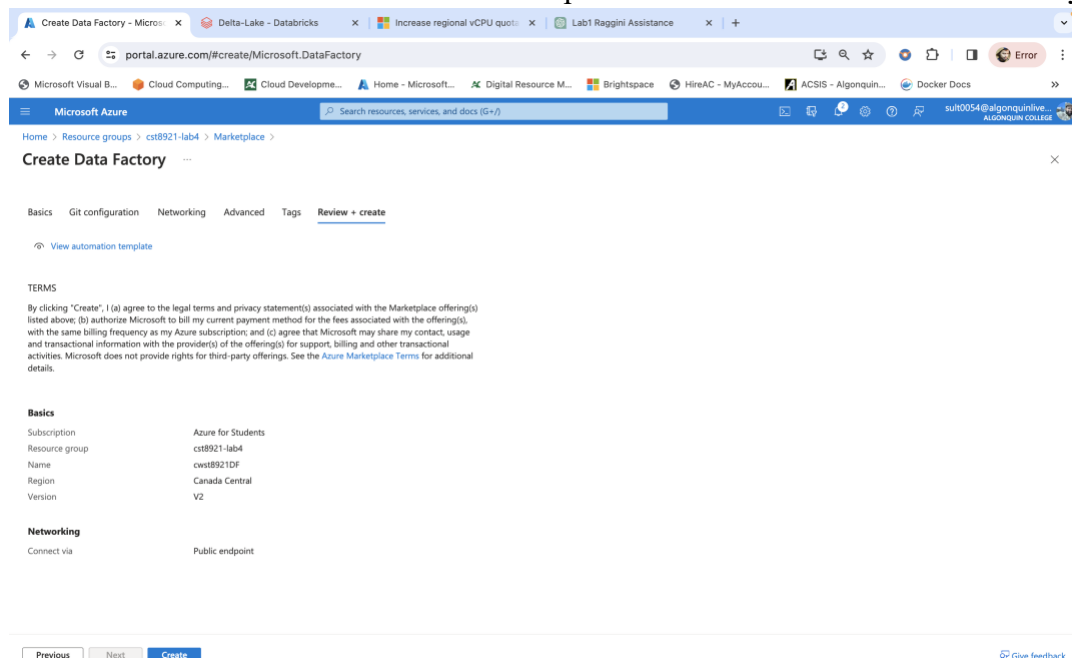
Command took 0.49 seconds --- by sult0054@algonquinlive.com at 2/4/2024, 3:55:08 PM on Ahmad Sultan's Cluster

Cmd 15: The log contains a full history of modifications to the data. Use the following code to see a record of the last 10 changes:

Cmd 16:

## Part 3: Execute Databricks notebook from Azure data factory

### 1. Use the notebook created in the part 1 to execute from azure data factory



The screenshot shows the "Create Data Factory" wizard in the Microsoft Azure portal. The "Review + create" tab is selected, showing the terms and conditions, and the basic configuration details.

**Basics**

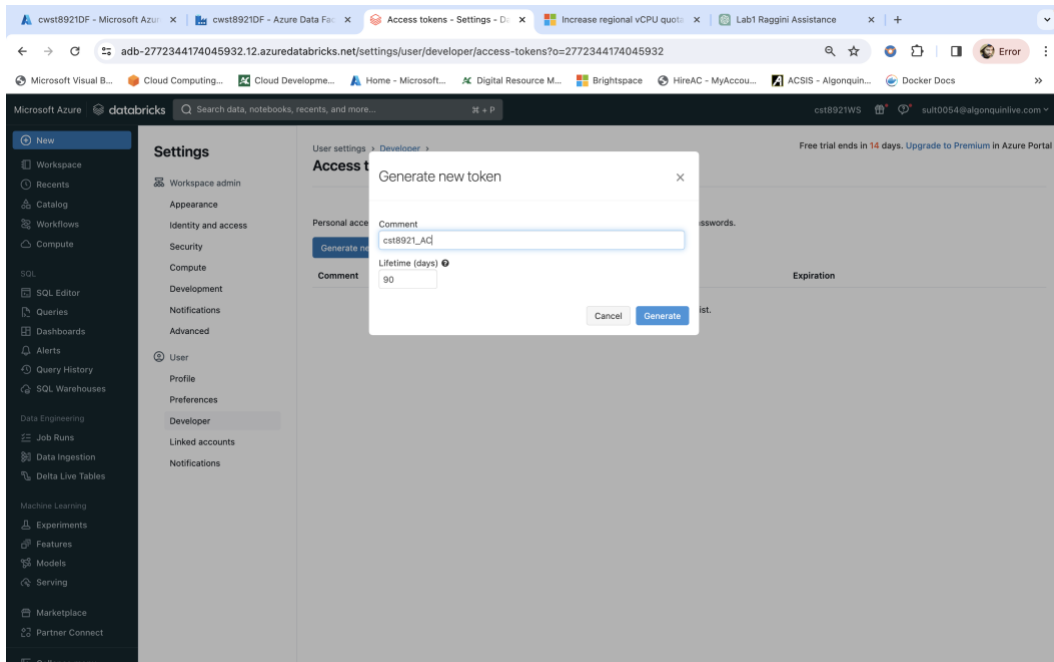
- Subscription: Azure for Students
- Resource group: cst8921-lab4
- Name: cst8921df
- Region: Canada Central
- Version: V2

**Networking**

- Connect via: Public endpoint

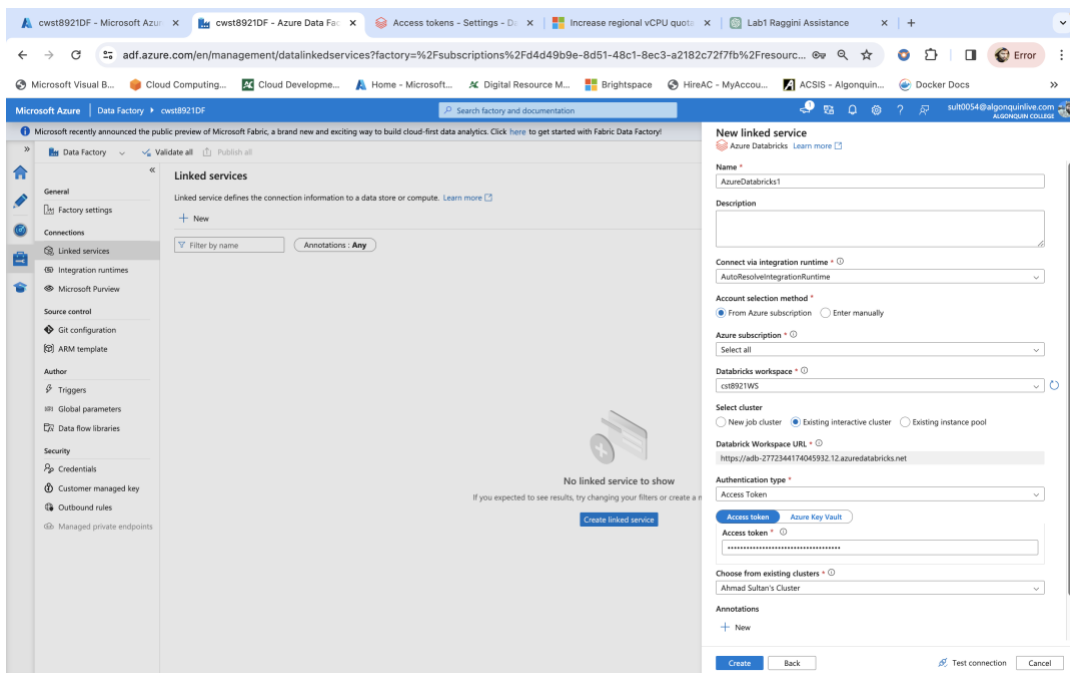
At the bottom, there are buttons for "Previous", "Next", and "Create".

## 2. generate access token from databricks notebook



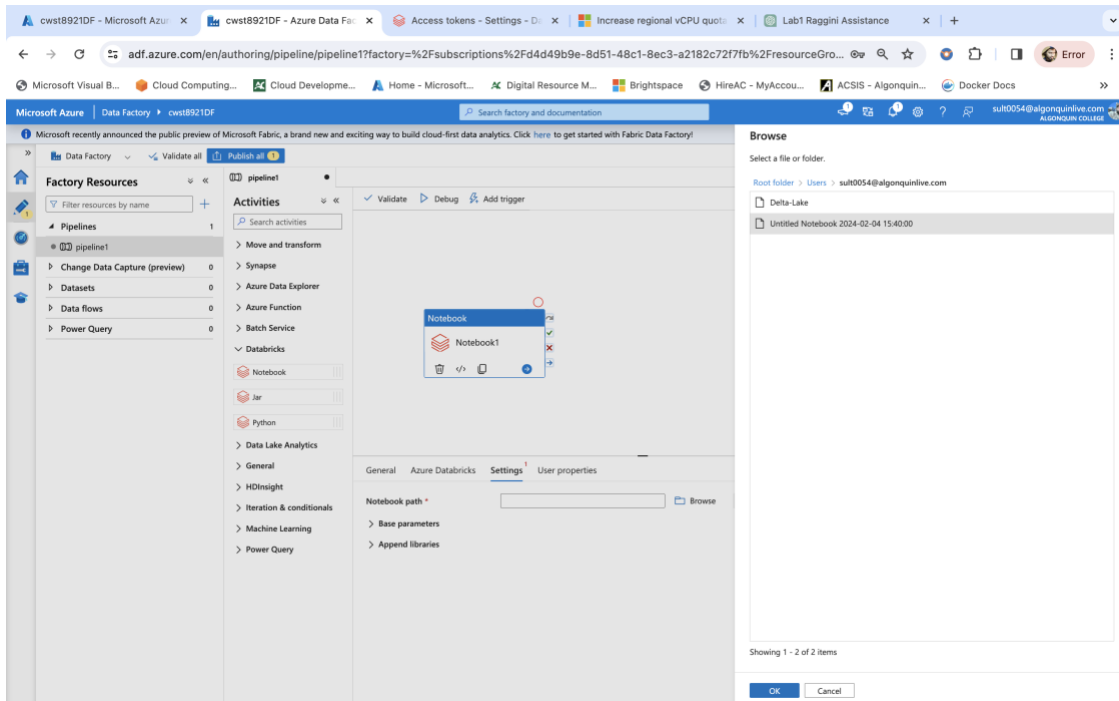
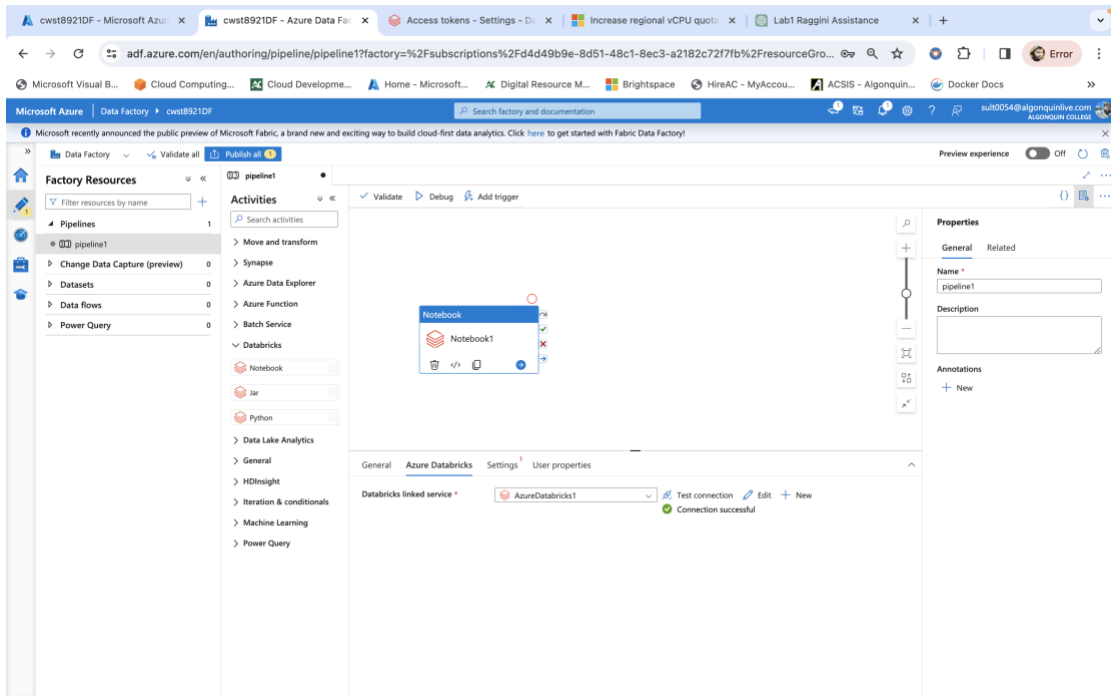
dapib8464e08757e053a4b332057ce1aa74b

## 3. create azure data factory instance and create a linked service to enable access to databricks workspace.

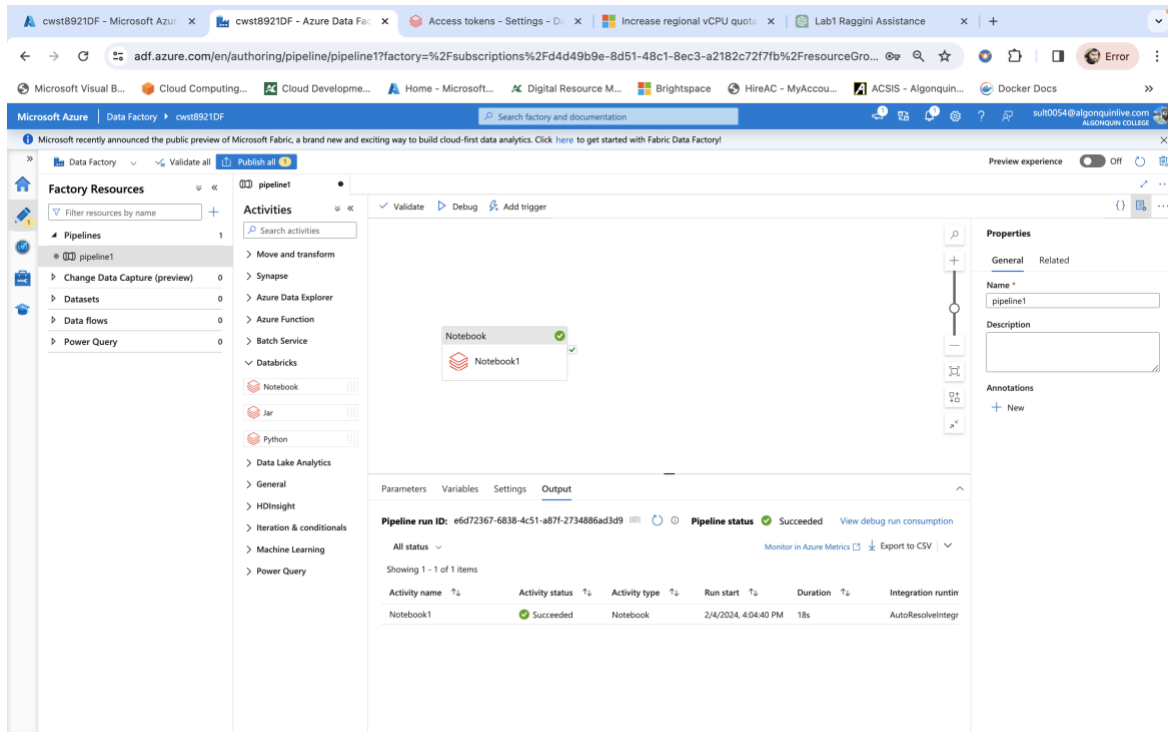




## 4. Create a pipeline to run the notebook from data factory.

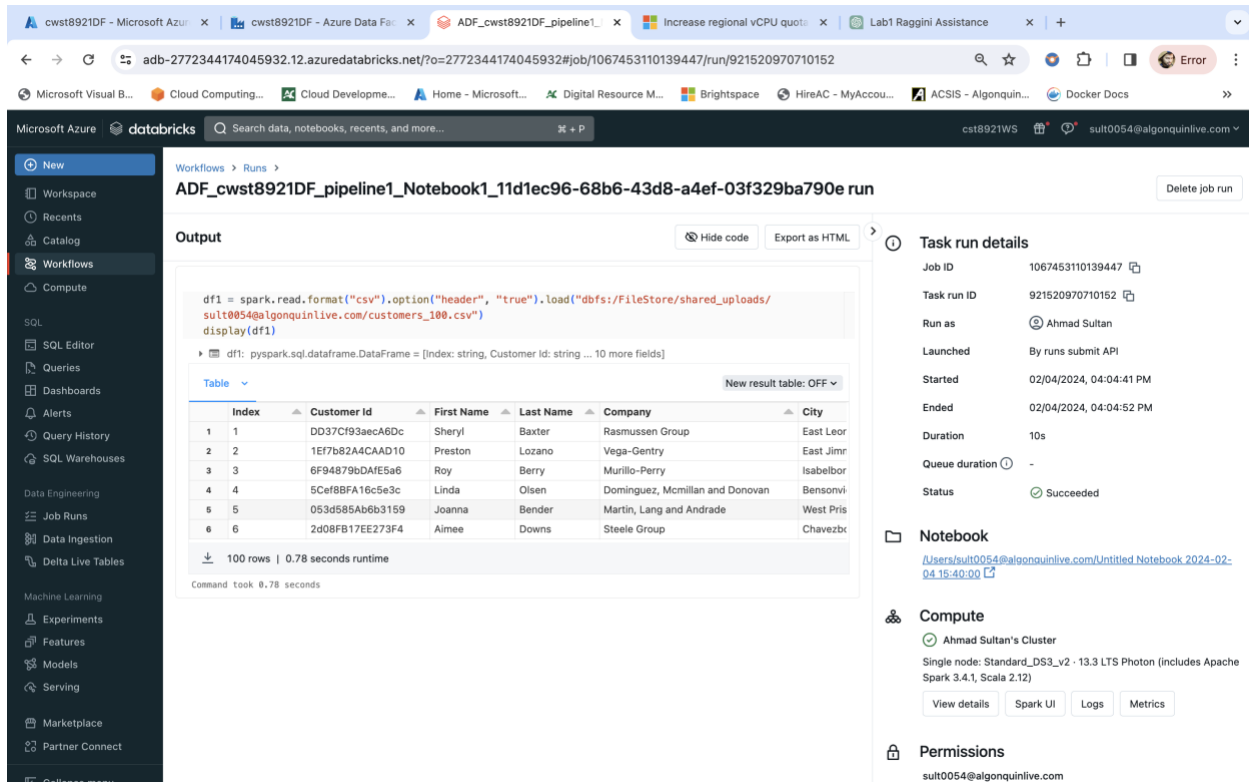


5. Execute the pipeline and monitor the run to see the status of pipeline if it is successful or not.



The screenshot shows the Microsoft Azure Data Factory portal. The left sidebar displays 'Factory Resources' with a list of pipelines, including 'pipeline1'. The main area shows the 'Activities' tab for 'pipeline1', which contains a 'Notebook' activity. The 'Output' tab is selected, showing the 'Pipeline run ID' as 'ef672367-6838-4c51-a87f-2734886ad3d9'. The 'Pipeline status' is 'Succeeded'. Below this, a table lists the activities:

Activity name	Activity status	Activity type	Run start	Duration	Integration runtime
Notebook1	Succeeded	Notebook	2/4/2024, 4:04:40 PM	18s	AutoResolveIntegr



The screenshot shows the Databricks workspace. The left sidebar displays 'Workflows' and 'Runs'. The main area shows the 'Task run details' for a notebook named 'ADF\_cwst8921DF\_pipeline1\_Notebook1\_11d1ec96-68b6-43d8-a4ef-03f329ba790e run'. The 'Output' tab is selected, showing the code executed in the notebook:

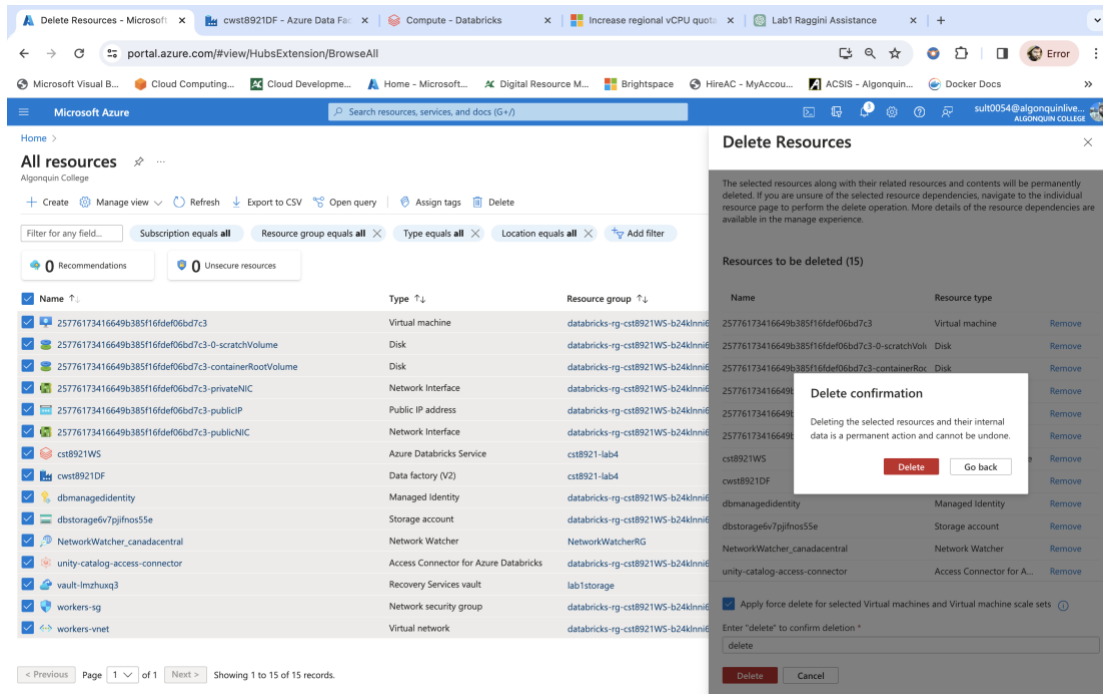
```
df1 = spark.read.format("csv").option("header", "true").load("dbfs:/FileStore/shared_uploads/sult0054@algonquinlive.com/customers_100.csv")
display(df1)
```

Below the code, a table displays the results of the query:

Index	Customer Id	First Name	Last Name	Company	City
1	DD37Cf93aecA6Dc	Sheryl	Baxter	Rasmussen Group	East Leor
2	1E7b82A4CAAD10	Preston	Lozano	Vega-Gentry	East Jimr
3	6F94879bDAfE5a6	Roy	Berry	Murillo-Perry	Isabelbor
4	5Cef8BFA16c5e3c	Linda	Olsen	Dominguez, Mcmillan and Donovan	Bensonvi
5	053d585Ab6b3159	Joanna	Bender	Martin, Lang and Andrade	West Pris
6	2d08FB17EE273F4	Aimee	Downs	Steele Group	Chavezbc

The 'Task run details' section on the right shows the 'Job ID' as '1067453110139447', the 'Task run ID' as '921520970710152', and the 'Status' as 'Succeeded'. The 'Notebook' section shows the path '/Users/sult0054@algonquinlive.com/Untitled Notebook 2024-02-04 15:40:00'.

## 6. Delete all the resources created in the lab.



The screenshot shows the Microsoft Azure portal interface. On the left, the 'All resources' page is visible, listing various resources created in the lab. On the right, the 'Delete Resources' dialog box is open, displaying a list of 15 resources to be deleted. A 'Delete confirmation' modal is also present, asking for confirmation to delete the selected resources.

Name	Type	Resource group	Action
25776173416649b385f16def06bd7c3	Virtual machine	databricks-rg-cst8921WS-b24kinn	Remove
25776173416649b385f16def06bd7c3-0-scratchVolume	Disk	databricks-rg-cst8921WS-b24kinn	Remove
25776173416649b385f16def06bd7c3-containerRootVolume	Disk	databricks-rg-cst8921WS-b24kinn	Remove
25776173416649b385f16def06bd7c3-privateNIC	Network Interface	databricks-rg-cst8921WS-b24kinn	Remove
25776173416649b385f16def06bd7c3-publicIP	Public IP address	databricks-rg-cst8921WS-b24kinn	Remove
25776173416649b385f16def06bd7c3-publicNIC	Network Interface	databricks-rg-cst8921WS-b24kinn	Remove
cst8921WS	Azure Databricks Service	cst8921-lab4	Remove
cst8921WS	Data factory (v2)	cst8921-lab4	Remove
dbmanagedidentity	Managed Identity	databricks-rg-cst8921WS-b24kinn	Remove
dbstorage6v7jgfhos5se	Storage account	databricks-rg-cst8921WS-b24kinn	Remove
NetworkWatcher_canadacentral	Network Watcher	NetworkWatcherRG	Remove
unity-catalog-access-connector	Access Connector for Azure Databricks	databricks-rg-cst8921WS-b24kinn	Remove
vault-lmzhuzq3	Recovery Services vault	lab1storage	Remove
workers-sg	Network security group	databricks-rg-cst8921WS-b24kinn	Remove
workers-vnet	Virtual network	databricks-rg-cst8921WS-b24kinn	Remove

## Results

1. Gain hands-on experience in provisioning Azure Databricks, creating clusters, and analyzing data using Spark and PySpark.
2. Explore Delta Lake capabilities, understanding how it enhances data reliability and performance.
3. Learn to integrate Azure Data Factory with Databricks, executing notebooks through pipelines and monitoring runs for efficient data processing.