

# Data Ingestion and Analysis with Azure Databricks

## Sumário

1	Version .....	3
2	Introduction .....	4
3	Data Ingestion .....	6
3.1	Introduction .....	6
3.2	Initial Storage on Amazon S3 .....	6
4	Layers of Service Provisioning and Configuration in Azure.....	7
4.1	Resource Group.....	7
4.2	Storage Account .....	8
4.3	Container.....	9
4.4	Key Vault .....	10
5	Data Transfer to Azure Data Lake Storage (ADLS) .....	13
6	Data Pipeline Execution .....	18
7	Creating Azure Databricks Workspace and Cluster.....	19
8	Data Extraction Layer .....	22
9	Data Transformation and Analysis Layer .....	25
10	Data Storage Layer .....	32
11	Data Query Layer.....	38
12	Conclusion .....	40
13	Reference .....	41

## 1 Version

This document was created by Cleber Zumba de Souza and can be freely distributed, as long as the source is mentioned.

Version	Action	Data
1.0	Document creation	2024/10/05

## 2 Introduction

In this project, I developed a data engineering solution integrating AWS, Azure, and Databricks technologies. The main goal was to create an efficient and scalable data pipeline that spans from data ingestion to transformation and analysis, ensuring secure storage and robust processing. The data processed in this pipeline is related to emergency calls from the San Francisco Fire Department.

### *Context and Objectives*

In modern data environments, it is essential to integrate different cloud platforms and tools to optimize the data workflow. This project was designed to:

- **Data Ingestion:** Read data from an S3 bucket on AWS.
- **Centralized Storage:** Store the data in a Data Lake in Azure Data Lake Storage (ADLS).
- **Processing and Analysis:** Use Databricks to transform, analyze, and store the transformed data.

### *Data Overview*

The data used in this pipeline are records of emergency calls answered by the San Francisco Fire Department. This data contains critical information such as the nature of the call, the neighborhood where it occurred, and response times. Analyzing this data is vital to improving emergency services and public safety.

### *Overview of Technologies Used*

1. **AWS S3:** Amazon's cloud storage service where raw data is initially stored.
2. **Azure Resource Group:** Groups and manages all project-related resources in Azure.
3. **Azure Storage Account:** Provides secure and scalable storage for data in Azure.
4. **Azure Data Factory:** Data integration service used to orchestrate and automate the movement of data between AWS S3 and ADLS services.
5. **Azure Data Lake Storage (ADLS):** Centralized storage solution in Azure that allows efficient and secure storage of data.
6. **Databricks:** Apache Spark-based data processing and analytics platform used to transform, analyze, and store data.

### *Project Workflow*

#### **1. Data Ingestion with Azure Data Factory:**

- A data pipeline in Azure Data Factory reads the data file stored in the AWS S3 bucket.
- The data is then transferred and stored in the Data Lake in Azure Data Lake Storage (ADLS).

#### **2. Processing and Analysis with Databricks:**

- Databricks integrates with ADLS to access the stored data.
- Using the processing power of Apache Spark, the data is transformed and analyzed as per the project needs.
- The transformed data is then stored back in ADLS or other destinations as needed.

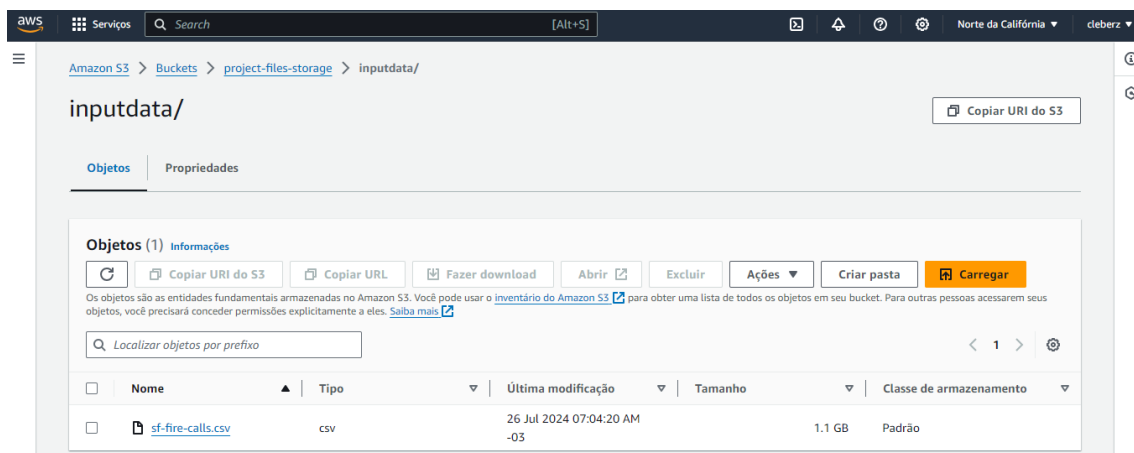
## 3 Data Ingestion

### 3.1 Introduction

In this chapter, I detailed the data ingestion process, from reading the file stored in the Amazon S3 bucket to transferring the data to Azure Data Lake Storage (ADLS). This is the first step in the data pipeline and is crucial to ensuring that the raw data is available for subsequent processing and analysis.

### 3.2 Initial Storage on Amazon S3

The San Francisco Fire Department's call data is initially stored in an Amazon S3 bucket. The image below shows the storage structure of the sf-fire-calls.csv file in the S3 bucket:

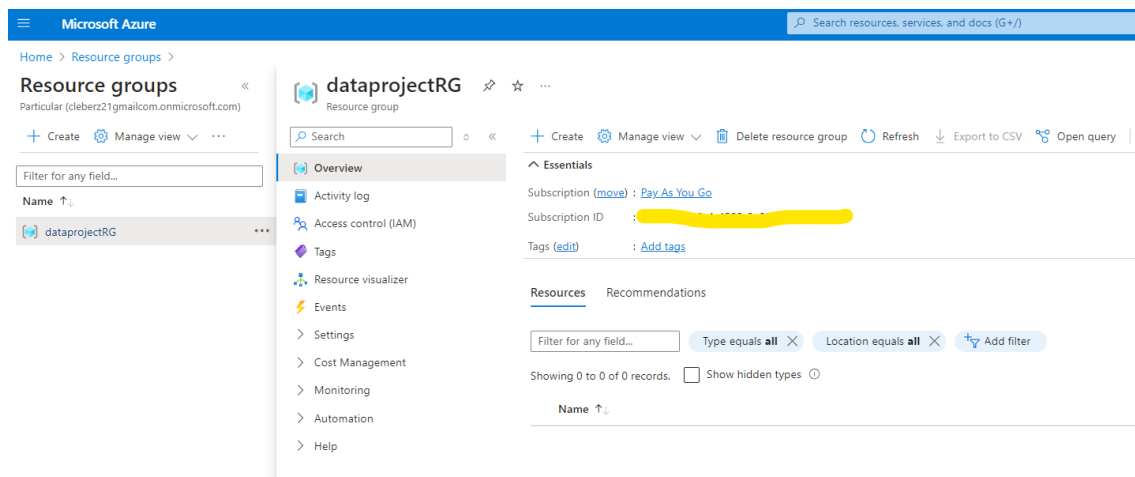


- **Bucket:** project-files-storage
- **Directory:** inputdata
- **File:** sf-fire-calls.csv
- **Size:** 1.1 GB
- **Last Modified:** July 26, 2024, 07:04 AM

## 4 Layers of Service Provisioning and Configuration in Azure

Before performing the data transfer, it was necessary to provision and configure the services in Azure. The steps include creating a Resource Group, a Storage Account, a Container, and a Key Vault.

### 4.1 Resource Group



A Resource Group is created to group and manage all project-related resources in Azure. This makes it easier to organize and manage resources.

## 4.2 Storage Account

The screenshot displays the Microsoft Azure portal interface. At the top, the header shows 'Microsoft Azure' and a search bar. Below the header, the breadcrumb trail indicates the path: 'Home > dataprojectst 1721846214976 | Overview >'. The main content area is titled 'dataprojectst' and 'Storage account'. On the left, a navigation pane lists various services under 'Data storage' and 'Security + networking'. The main pane shows the 'Overview' tab for the storage account. It includes a search bar, action buttons like 'Upload', 'Open in Explorer', 'Delete', 'Move', 'Refresh', 'Open in mobile', 'CLI / PS', and 'Feedback'. Below these, the 'Essentials' section lists key properties: Resource group (dataprojectRG), Location (canadacentral), Primary/Secondary Location (Primary: Canada Central, Secondary: Canada East), Subscription (Pay As You Go), Subscription ID (redacted), and Disk state (Primary: Available, Secondary: Available). The 'Tags' section shows 'Add tags'. The 'Properties' tab is active, displaying 'Data Lake Storage' settings: Hierarchical namespace (Enabled), Default access tier (Hot), Blob anonymous access (Disabled), Blob soft delete (Enabled (7 days)), Container soft delete (Enabled (7 days)), Versioning (Disabled), Change feed (Disabled), NFS v3 (Disabled), SFTP (Disabled), and Storage tasks assignments (None).

Property	Value
Resource group	dataprojectRG
Location	canadacentral
Primary/Secondary Location	Primary: Canada Central, Secondary: Canada East
Subscription	Pay As You Go
Subscription ID	[Redacted]
Disk state	Primary: Available, Secondary: Available
Tags	Add tags

Property	Value
Hierarchical namespace	Enabled
Default access tier	Hot
Blob anonymous access	Disabled
Blob soft delete	Enabled (7 days)
Container soft delete	Enabled (7 days)
Versioning	Disabled
Change feed	Disabled
NFS v3	Disabled
SFTP	Disabled
Storage tasks assignments	None

A Storage Account is created to provide secure and scalable storage for your data. This storage account is essential for storing data transferred from S3.



## 4.3 Container

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes the 'Microsoft Azure' logo and a search bar. The breadcrumb trail indicates the current location: Home > dataprojectst.1721846214976 | Overview > dataprojectst. The main heading is 'dataprojectst | Containers', with 'Storage account' noted below. A left-hand navigation pane lists various services, with 'Containers' selected under the 'Data storage' category. The main content area displays a table of containers. At the top of this area is a search bar labeled 'Search containers by prefix'. Below it, a table lists two containers: '\$logs' and 'inputdataset'. Each row includes a checkbox for selection, the container name, and the last modified timestamp. The '\$logs' container was last modified on 7/24/2024 at 3:40:46 PM, and the 'inputdataset' container was last modified on 7/24/2024 at 3:42:39 PM. Above the table, there are action buttons: '+ Container', 'Change access level', 'Restore containers', 'Refresh', 'Delete', and 'Give feedback'.

	Name	Last modified
<input type="checkbox"/>	\$logs	7/24/2024, 3:40:46 PM
<input type="checkbox"/>	inputdataset	7/24/2024, 3:42:39 PM

Within the Storage Account, a container was created to store the data in the desired format.

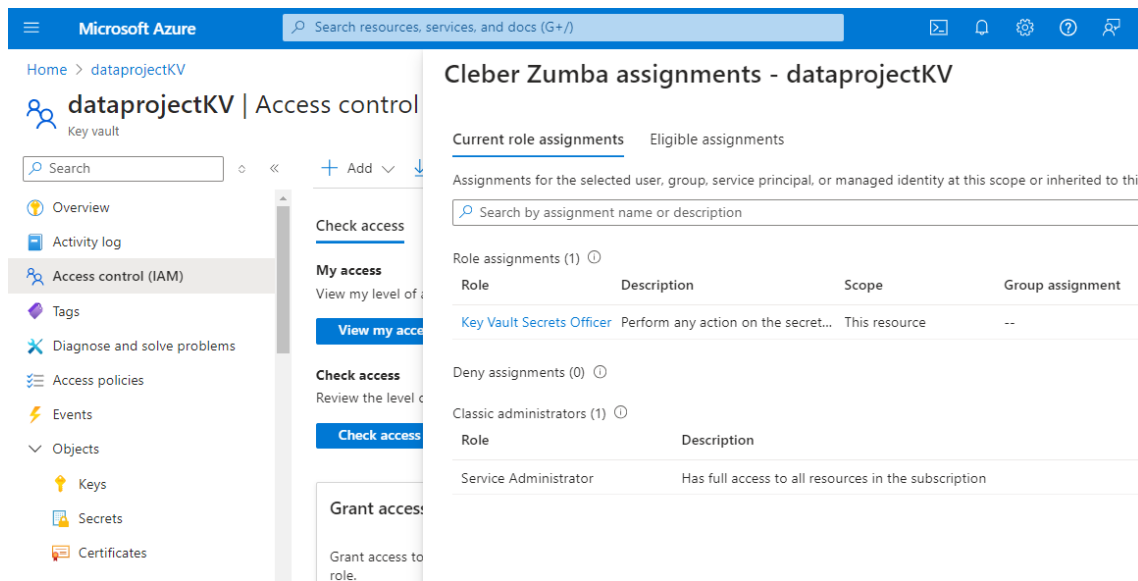
## 4.4 Key Vault

The screenshot displays the Microsoft Azure portal interface for a Key Vault named 'dataprojectKV'. The left-hand navigation pane is expanded, showing the 'Overview' section. The main content area displays the following information:

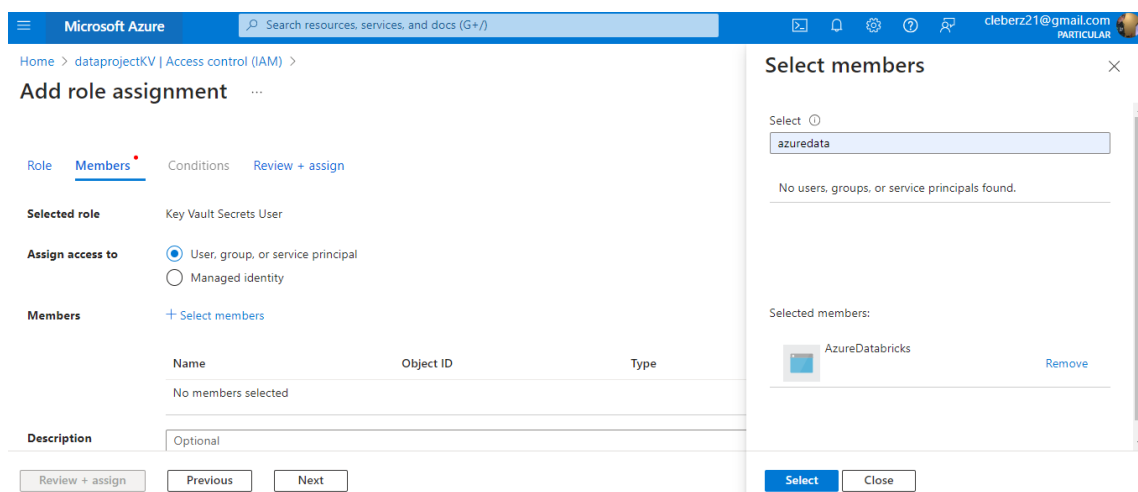
- Location:** Canada Central
- Subscription (move):** [Pay As You Go](#)
- Subscription ID:** [Redacted]
- Tags (edit):** [Add tags](#)

Below this information, there are tabs for 'Get started', 'Properties', 'Monitoring', 'Tools + SDKs', and 'Tutorials'. The 'Get started' tab is currently selected.

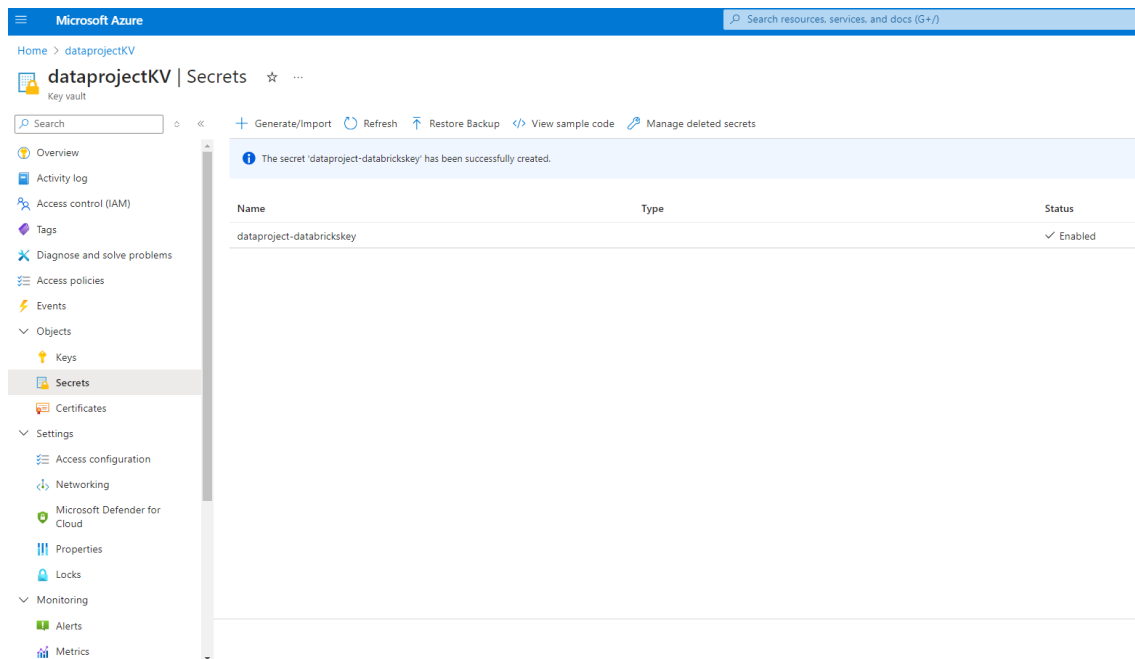
Key Vault Creation



Adding the Key Vault Secrets Officer role



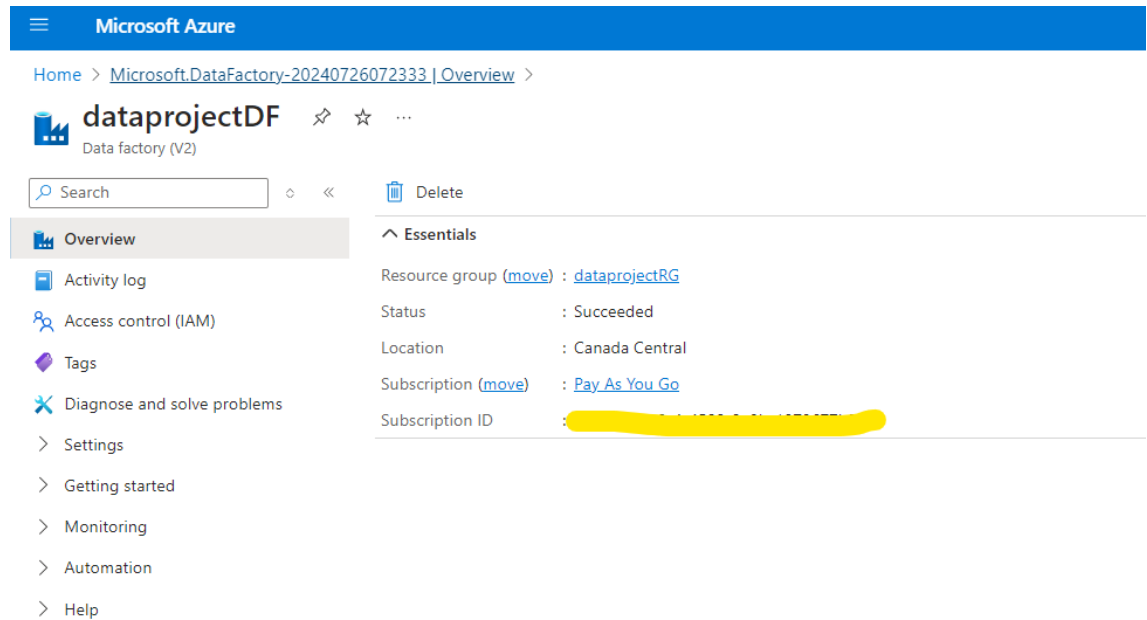
Adding Azure Databricks member to Key Vault Secrets Officer role



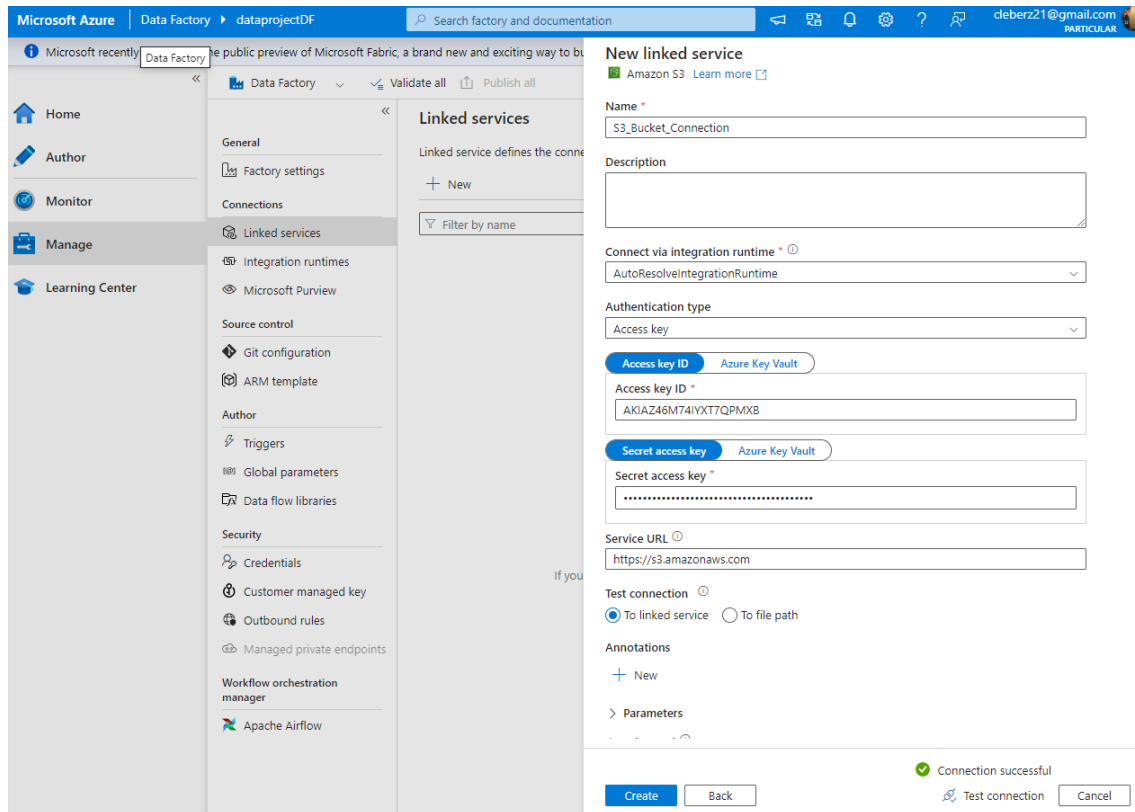
Creating a Secret Key Vault dataprojectKV

## 5 Data Transfer to Azure Data Lake Storage (ADLS)

Using Azure Data Factory, I set up a data pipeline to read the file stored in the Amazon S3 bucket and transfer it to Azure Data Lake Storage (ADLS). This process involves:



### Creating a Data Factory



**Setting up a Linked Service in Azure Data Factory to connect to S3 bucket.**

# Data Engineering with Microsoft Azure Databricks

The screenshot shows the 'New linked service' configuration page in the Microsoft Azure Data Factory portal. The left sidebar contains navigation options: Home, Author, Monitor, Manage, and Learning Center. The 'Manage' section is expanded, showing 'Linked services' as the selected option. The main panel displays the configuration for a new linked service named 'ADLS\_Storage\_Connection'. The configuration includes a description field, a dropdown for 'Connect via integration runtime' set to 'AutoResolveIntegrationRuntime', an 'Authentication type' dropdown set to 'Account key', and an 'Account selection method' set to 'From Azure subscription'. The 'Azure subscription' dropdown is set to 'Pay As You Go (9fe27680-c6c4-4588-8e0b-1079677b5052)'. The 'Storage account name' dropdown is set to 'dataprojectst'. Below these fields, there are options for 'Test connection' (To linked service or To file path) and 'Annotations' (New, Parameters, Advanced). At the bottom, there are buttons for 'Create', 'Back', 'Test connection', and 'Cancel'. A green checkmark indicates 'Connection successful'.

Microsoft Azure | Data Factory | dataprojectDF

Microsoft recently announced the public preview of Microsoft Fabric, a brand new and exciting way to build cloud-first data analytics. Click here to get started with Fabric Data Factory!

Home Author Monitor Manage Learning Center

General Factory settings Connections Linked services Integration runtimes Microsoft Purview Source control Git configuration ARM template Author Triggers Global parameters Data flow libraries Security Credentials Customer managed key Outbound rules Managed private endpoints Workflow orchestration manager Apache Airflow

Linked services

Linked service defines the connection information to a data store or compute. Learn more

+ New

Filter by name

Showing 1 - 1 of 1 items

Name

S3\_Bucket\_Connection

Name \* ADLS\_Storage\_Connection

Description

Connect via integration runtime \* AutoResolveIntegrationRuntime

Authentication type Account key

Account selection method From Azure subscription Enter manually

Azure subscription Pay As You Go (9fe27680-c6c4-4588-8e0b-1079677b5052)

Storage account name \* dataprojectst

Test connection To linked service To file path

Annotations + New > Parameters > Advanced

Create Back Test connection Cancel

Connection successful

## Configuring a Linked Service in Azure Data Factory to connect to ADLS.

The screenshot shows the 'Linked services' list page in the Microsoft Azure Data Factory portal. The left sidebar contains navigation options: Home, Author, Monitor, Manage, and Learning Center. The 'Manage' section is expanded, showing 'Linked services' as the selected option. The main panel displays a list of linked services. The list has columns for Name, Type, Related, and Annotations. Two linked services are listed: 'ADLS\_Storage\_Connection' (Azure Data Lake Storage Gen2) and 'S3\_Bucket\_Connection' (Amazon S3). Both have 0 related items and 0 annotations.

Microsoft Azure | Data Factory | dataprojectDF

Microsoft recently announced the public preview of Microsoft Fabric, a brand new and exciting way to build cloud-first data analytics. Click here to get started with Fabric Data Factory!

Home Author Monitor Manage Learning Center

General Factory settings Connections Linked services Integration runtimes Microsoft Purview Source control Git configuration ARM template Author Triggers Global parameters Data flow libraries Security Credentials Customer managed key Outbound rules Managed private endpoints Workflow orchestration manager Apache Airflow

Linked services

Linked service defines the connection information to a data store or compute. Learn more

+ New

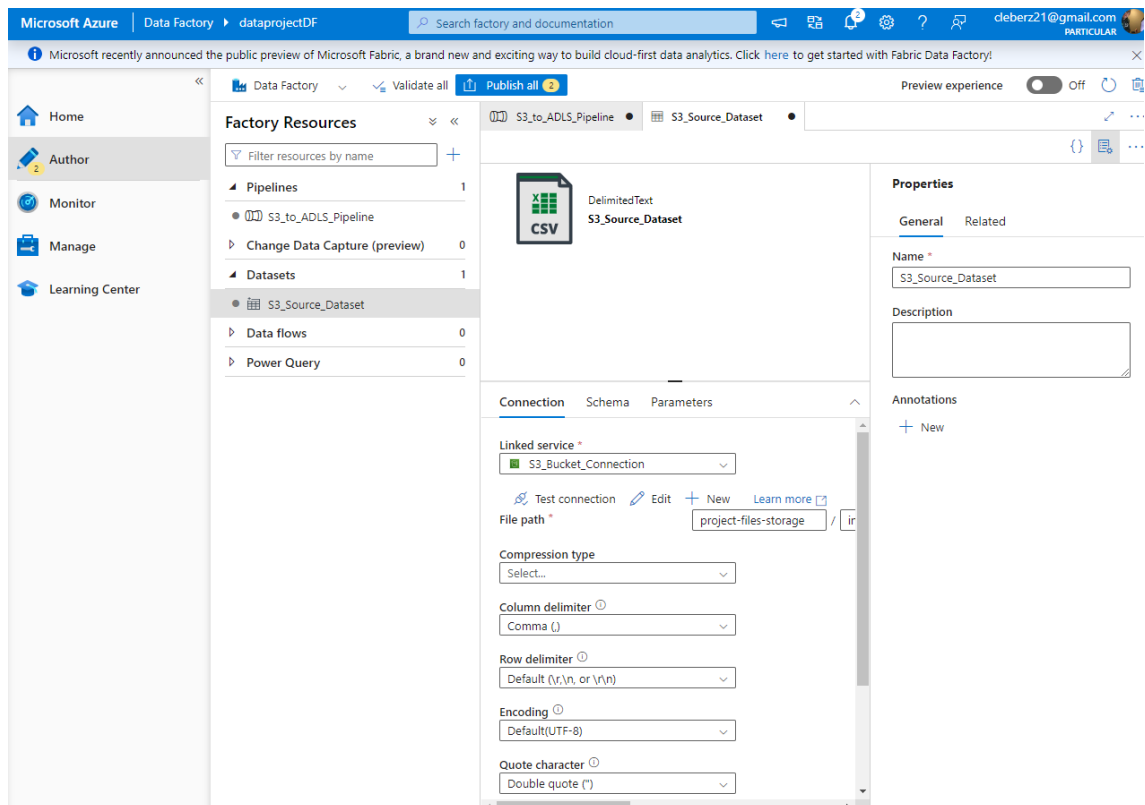
Filter by name Annotations: Any

Showing 1 - 2 of 2 items

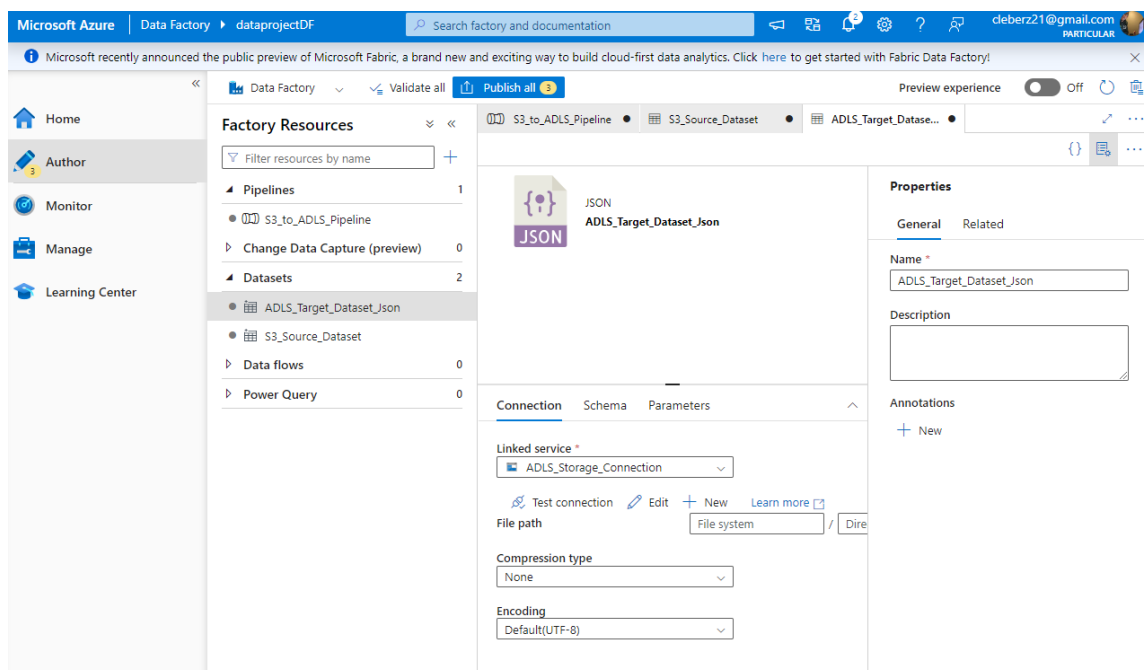
Name	Type	Related	Annotations
ADLS_Storage_Connection	Azure Data Lake Storage Gen2	0	
S3_Bucket_Connection	Amazon S3	0	

Linked Service created

# Data Engineering with Microsoft Azure Databricks

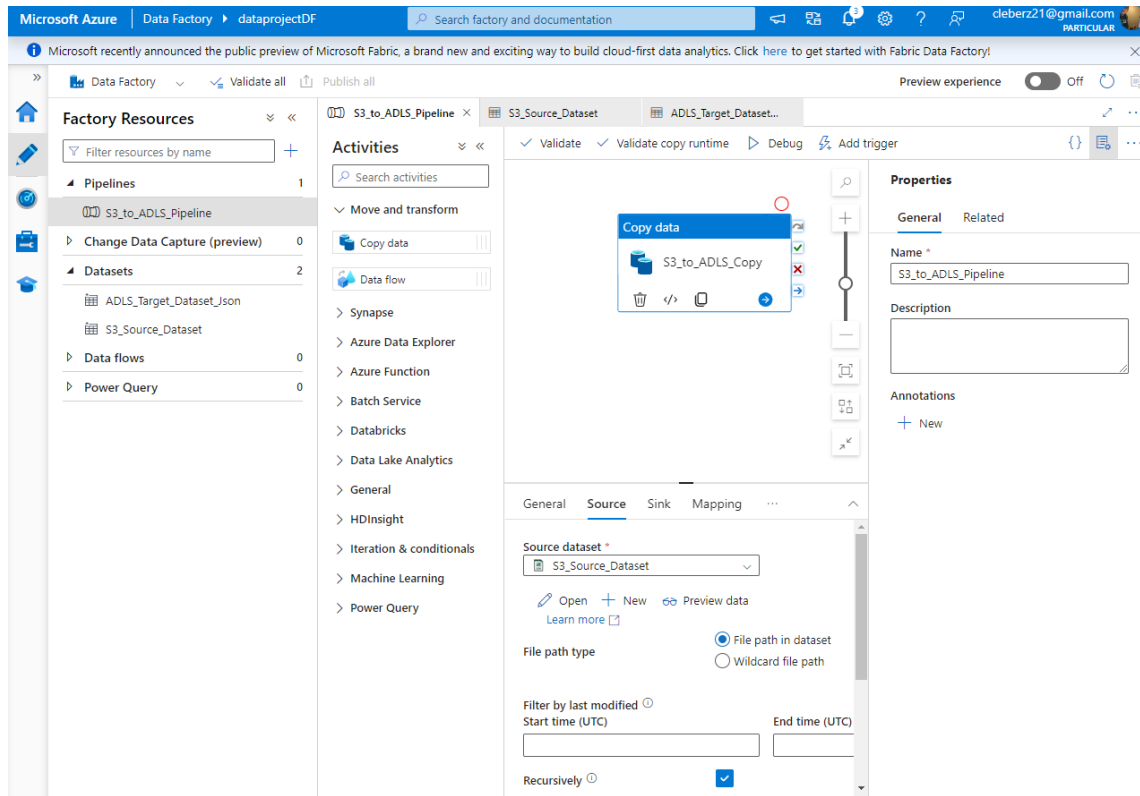


**Creation of a Dataset that represents the source of the data**



**Creation of a Dataset that represents the destination of the data.**

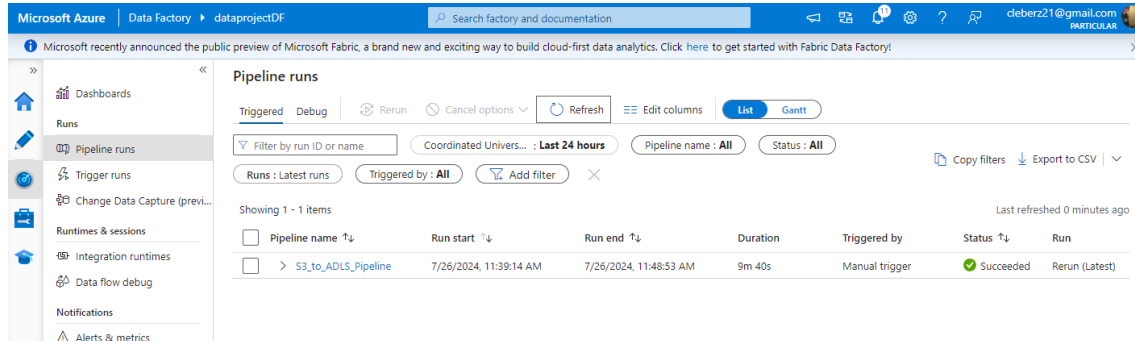




**Development of a Pipeline that copies data from S3 to ADLS.**

## 6 Data Pipeline Execution

The data pipeline has been configured to perform the data transfer efficiently and securely, ensuring that the `sf-fire-calls.csv` file is made available in the ADLS Data Lake, in Json format, for subsequent processing in Databricks.



Microsoft Azure | Data Factory | dataprojectDF

Microsoft recently announced the public preview of Microsoft Fabric, a brand new and exciting way to build cloud-first data analytics. Click [here](#) to get started with Fabric Data Factory!

### Pipeline runs

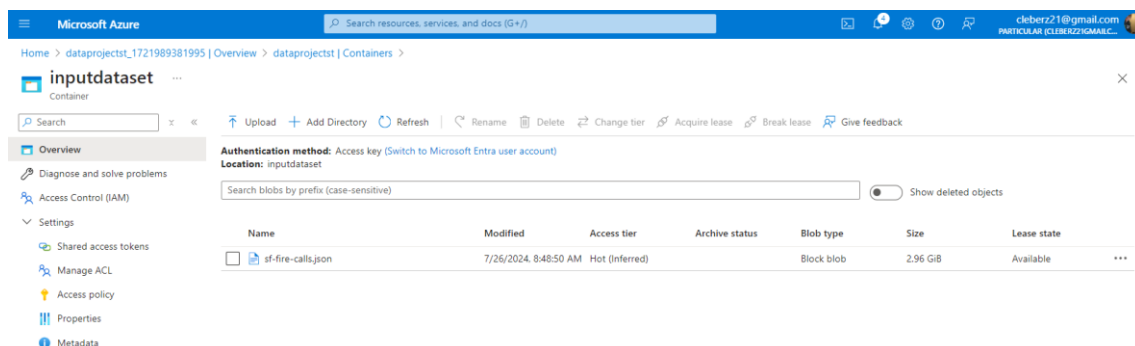
Triggered Debug Rerun Cancel options Refresh Edit columns List Gantt

Filter by run ID or name Coordinated Univers... : Last 24 hours Pipeline name : All Status : All

Runs : Latest runs Triggered by : All Add filter

Showing 1 - 1 items Last refreshed 0 minutes ago

Pipeline name	Run start	Run end	Duration	Triggered by	Status	Run
> S3_to_ADLS_Pipeline	7/26/2024, 11:39:14 AM	7/26/2024, 11:48:53 AM	9m 40s	Manual trigger	Succeeded	Rerun (Latest)



Microsoft Azure

Home > dataprojectst\_1721989381995 | Overview > dataprojectst | Containers >

### inputdataset

Container

Search Upload Add Directory Refresh Rename Delete Change tier Acquire lease Break lease Give feedback

Authentication method: Access key (Switch to Microsoft Entra user account)

Location: inputdataset

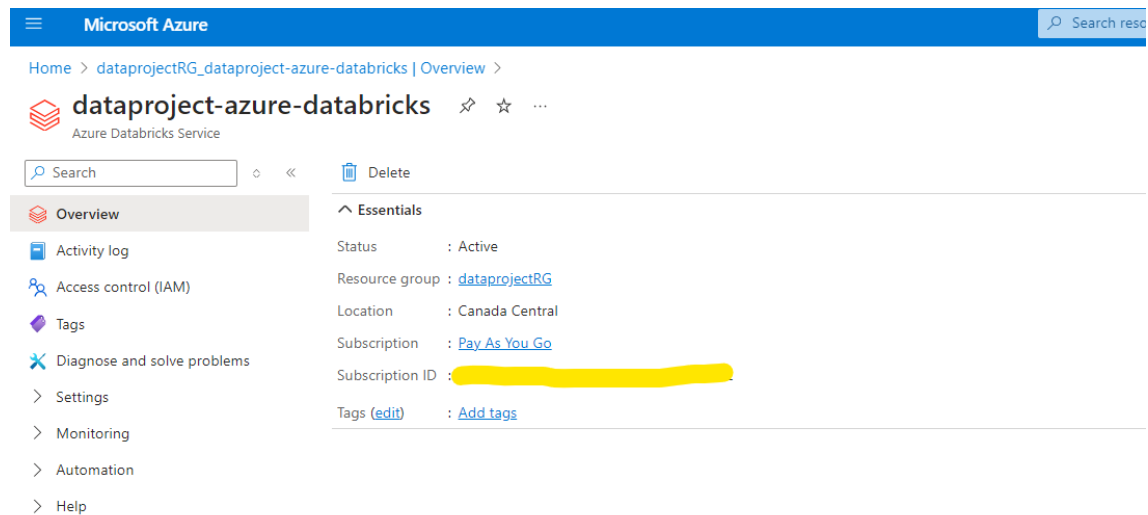
Search blobs by prefix (case-sensitive) Show deleted objects

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
sf-fire-calls.json	7/26/2024, 8:48:50 AM	Hot (Inferred)		Block blob	2.96 GiB	Available

Data transfer completed

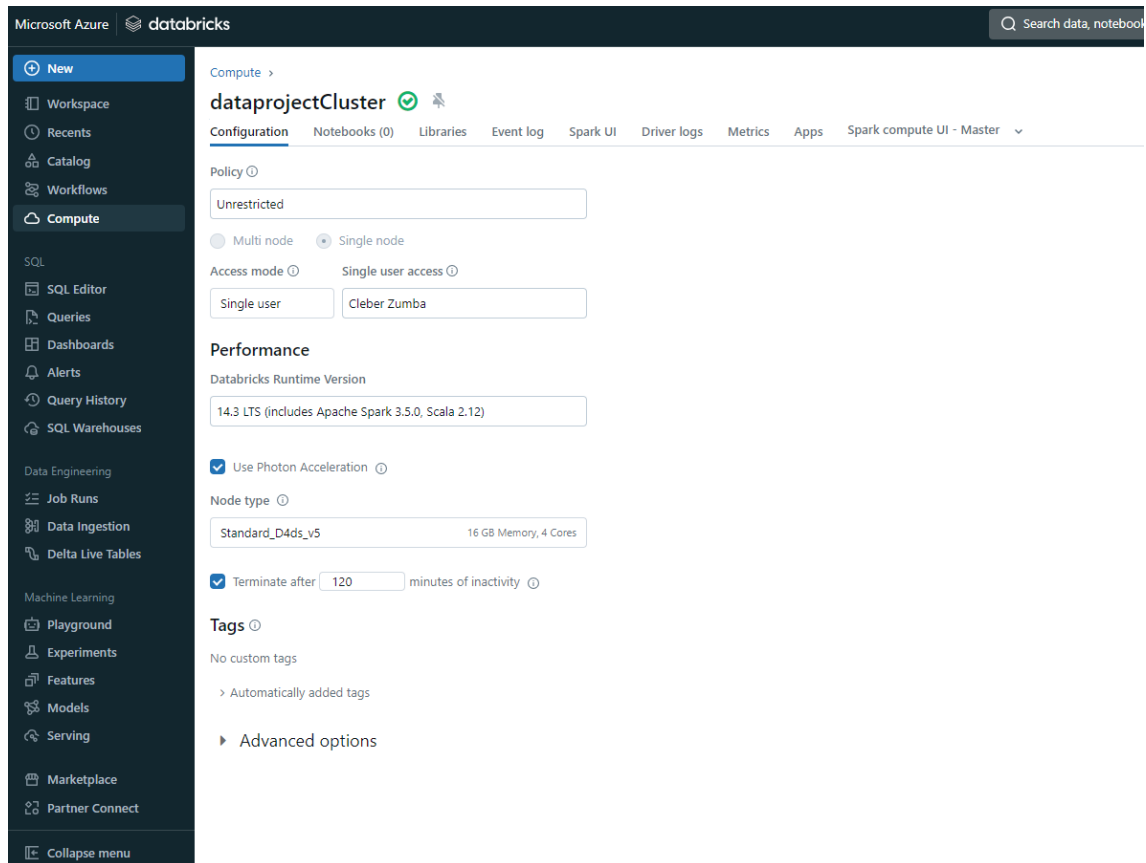
## 7 Creating Azure Databricks Workspace and Cluster

Azure Databricks has been provisioned to perform data transformation and analysis. This includes creating a workspace, cluster, and secret scope.



The screenshot displays the Microsoft Azure portal interface for the 'dataproject-azure-databricks' service. The top navigation bar shows 'Microsoft Azure' and a search bar. Below the navigation bar, the breadcrumb path is 'Home > dataprojectRG\_dataproject-azure-databricks | Overview >'. The main heading is 'dataproject-azure-databricks' with the subtitle 'Azure Databricks Service'. A search bar and a 'Delete' button are visible. The left sidebar contains a list of navigation options: Overview (selected), Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings, Monitoring, Automation, and Help. The main content area is titled 'Essentials' and lists the following details: Status: Active, Resource group: dataprojectRG, Location: Canada Central, Subscription: Pay As You Go, Subscription ID: [redacted], and Tags (edit): Add tags.

Essentials	
Status	: Active
Resource group	: <a href="#">dataprojectRG</a>
Location	: Canada Central
Subscription	: <a href="#">Pay As You Go</a>
Subscription ID	: [redacted]
Tags ( <a href="#">edit</a> )	: <a href="#">Add tags</a>



## Creating a Cluster in Databricks

Microsoft Azure | databricks

HomePage / Create Secret Scope

### Create Secret Scope

[Cancel](#) [Create](#)

A store for secrets that is identified by a name and backed by a specific store type. [Learn more](#)

Scope Name <sup>?</sup>

dataproject-datbricks-scope

Manage Principal <sup>?</sup>

All workspace users

Azure Key Vault <sup>?</sup>

DNS Name

https://dataprojectkv.vault.azure.net/

Resource ID

/subscriptions/[redacted]/resourceGroups/dataproje

Creating a Secret Scope in Databricks

```
Windows PowerShell
PS C:\Users\DELL> databricks secrets list-scopes
Scope
-----
dataproject-datbricks-scope
Backend
-----
AZURE_KEYVAULT
KeyVault URL
-----
https://dataprojectkv.vault.azure.net/
PS C:\Users\DELL>
```

List Scope

## 8 Data Extraction Layer

**San Francisco Fire Calls ETL and Analysis**

**SUMMARY**

Fire Calls-For-Service includes all fire units' responses to 911 calls from the city's Computer-Aided Dispatch ("CAD") system. This includes responses to Medical Incidents requiring EMS staff. Each record includes the call number, incident number, address, unit identifier, call type, and disposition. All relevant time intervals are also included. Because this dataset is based on responses, and since most calls involve multiple units, there are multiple records for each call number. Addresses are associated with an intersection or call box, not a specific address.

**HOW TO USE THIS DATASET**

This dataset is based on responses, and since most calls involve multiple units, there are multiple records for each call number. The most common call types are Medical Incidents, Alarms, Structure Fires, and Traffic Collisions.

**ETL Process**

Source Systems → Extract → Transform → Load → Destination

**San Francisco Fire Calls**

- This pipeline uses the San Francisco Fire Department's call event dataset and demonstrates:
  - End-to-end Data Engineering pipeline covers the extraction, transformation and loading (ETL) steps of large volumes of data, using PySpark for transformation and Spark SQL for queries. Caching techniques were implemented to optimize query performance, and data analysis was conducted to gain insights.
  - How to answer questions by analyzing data using Spark SQL
- Benefits of the Techniques Used:
  - Partitioning: Improves data reading and writing by dividing data into smaller, more manageable partitions.
  - Spark Settings: Tweaks like `spark.sql.shuffle.partitions` and `spark.sql.autoBroadcastJoinThreshold` help optimize shuffle and join operations.
  - Parquet Format: Parquet format storage improves reading and writing performance due to its columnar nature and support compression.
  - Cache: Caching frequently used `DataFrames` reduces subsequent data reading time.
  - Integrated Analysis: Analysis can be performed directly in Databricks, with integrated visualizations for easy interpretation of the results.
  - Using Databricks and Spark allows the pipeline to easily scale to large volumes of data.

Microsoft Azure

data

bricks

Search data, notebooks, recents, and more...

CTRL + P

datapipeline-azure-databricks

New

Workspace

Recents

Catalog

Workflows

Compute

SQL

SQL Editor

Queries

Dashboards

Alerts

Query History

SQL Warehouses

Data Engineering

Job Runs

Data Ingestion

Delta Live Tables

Machine Learning

Playground

Experiments

Features

Models

Serving

Marketplace

Partner Connect

Collapse menu

San Francisco Fire Calls

Python

☆

File Edit View Run Help

Last edit was 3 minutes ago

Provide feedback

Run all

datapipelineCluster

Schedule

Share

1. Data Sourcing/Extraction

Dataset has been downloaded from [San Francisco Fire Department Calls](#)

To load your data into DBFS, please refer to [Databricks Guide > Importing Data](#).

Auto-fix error in Chat: ON

04:57 PM (22s)

3

```
dbutils.fs.unmount('/mnt/inputdata')
dbutils.fs.mount(
  source = 'wasbs://inputdataset@datapipelinecluster.blob.core.windows.net',
  mount_point = '/mnt/inputdata',
  extra_configs=
    {'fs.azure.account.key.datapipelinecluster.blob.core.windows.net':dbutils.secrets.get('datapipeline-databricks-scope',
    'datapipeline-databricksnewkey')})

/mnt/inputdata has been unmounted.
True
```

2 minutes ago (<1s)

4

Python

✖

🔍

🔗

⋮

%fs ls dbfs:/mnt/inputdata

Table

+

🔍

🔗

🗑

	path	name	size	modificationTime
1	dbfs/mnt/inputdata/sf-fire-calls.csv	sf-fire-calls.csv	1137925359	1721851374000

1 row | 0.35 seconds runtime

Refreshed 2 minutes ago

23

# Data Engineering with Microsoft Azure Databricks

Microsoft Azure | databricks | Search data, notebooks, recents, and more... | CTRL + P | dataproject-azure-databricks

New | Workspace | Recents | Catalog | Workflows | Compute | SQL | SQL Editor | Queries | Dashboards | Alerts | Query History | SQL Warehouses | Data Engineering | Job Runs | Data Ingestion | Delta Live Tables | Machine Learning | Playground | Experiments | Features | Models | Serving | Marketplace | Partner Connect

San Francisco Fire Calls | Python | Last edit was 3 minutes ago | Provide feedback | Run all | dataprojectCluster | Schedule | Share

```
# Importando pacotes necessários
from pyspark.sql.types import *
from pyspark.sql.functions import *

# Definindo o schema
fire_schema = StructType([StructField('CallNumber', IntegerType(), True),
                           StructField('UnitID', StringType(), True),
                           StructField('IncidentNumber', IntegerType(), True),
                           StructField('CallType', StringType(), True),
                           StructField('CallDate', StringType(), True),
                           StructField('WatchDate', StringType(), True),
                           StructField('CallFinalDisposition', StringType(), True),
                           StructField('AvailableDtTm', StringType(), True),
                           StructField('Address', StringType(), True),
                           StructField('City', StringType(), True),
                           StructField('Zipcode', IntegerType(), True),
                           StructField('Battalion', StringType(), True),
                           StructField('StationArea', StringType(), True),
                           StructField('Box', StringType(), True),
                           StructField('OriginalPriority', StringType(), True),
                           StructField('Priority', StringType(), True),
                           StructField('FinalPriority', IntegerType(), True),
                           StructField('ALSUnit', BooleanType(), True),
                           StructField('CallTypeGroup', StringType(), True),
                           StructField('NumAlarms', IntegerType(), True),
                           StructField('UnitType', StringType(), True),
                           StructField('UnitSequenceInCallDispatch', IntegerType(), True),
                           StructField('FirePreventionDistrict', StringType(), True),
                           StructField('SupervisorDistrict', StringType(), True),
                           StructField('Neighborhood', StringType(), True),
                           StructField('Location', StringType(), True),
                           StructField('RowID', StringType(), True),
                           StructField('Delay', FloatType(), True)])
```

Microsoft Azure | databricks | Search data, notebooks, recents, and more... | CTRL + P | dataproject-azure-databricks

New | Workspace | Recents | Catalog | Workflows | Compute | SQL | SQL Editor | Queries | Dashboards | Alerts | Query History | SQL Warehouses | Data Engineering | Job Runs | Data Ingestion | Delta Live Tables | Machine Learning | Playground | Experiments | Features | Models | Serving | Marketplace | Partner Connect

San Francisco Fire Calls | Python | Last edit was 4 minutes ago | Provide feedback | Run all | dataprojectCluster | Schedule | Share

```
fire_df = spark.read.csv("/mnt/inputdata/sf-fire-calls.csv",
                        schema=fire_schema,
                        header=True,
                        ignoreLeadingWhiteSpace=True,
                        ignoreTrailingWhiteSpace=True)
```

fire\_df: pyspark.sql.dataframe.DataFrame = [CallNumber: integer, UnitID: string ... 26 more fields]

```
display(fire_df.limit(10))
```

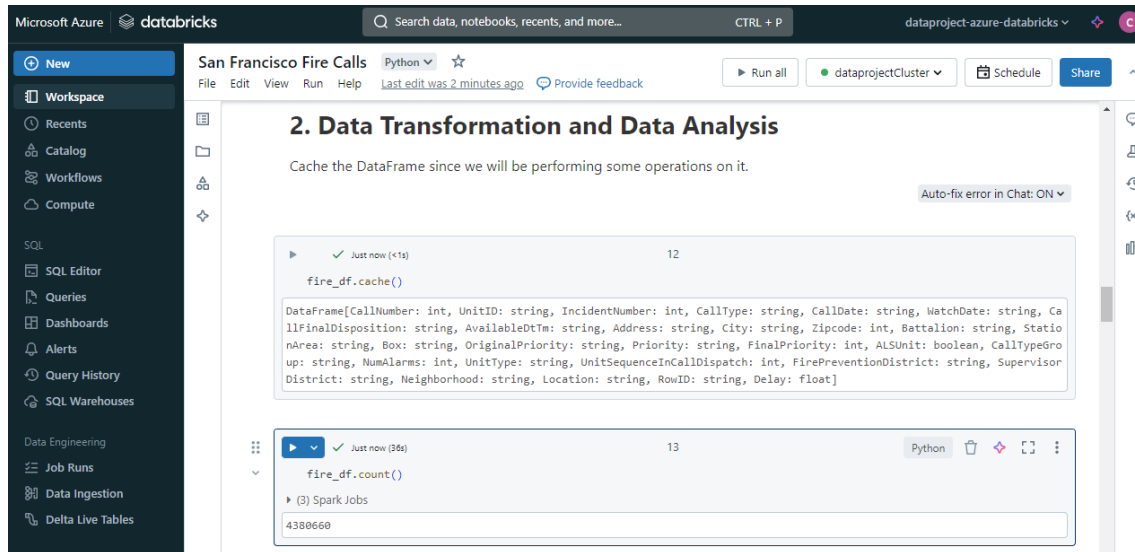
(1) Spark Jobs

	CallNumber	UnitID	IncidentNumber	CallType	CallDate	WatchDate	Ca
1	20110014	M29	2003234	Medical Incident	01/11/2002	01/10/2002	Other
2	20110015	M08	2003233	Medical Incident	01/11/2002	01/10/2002	Other
3	20110016	B02	2003235	Structure Fire	01/11/2002	01/10/2002	Other
4	20110016	B04	2003235	Structure Fire	01/11/2002	01/10/2002	Other
5	20110016	D2	2003235	Structure Fire	01/11/2002	01/10/2002	Other
6	20110016	E03	2003235	Structure Fire	01/11/2002	01/10/2002	Other
7	20110016	E38	2003235	Structure Fire	01/11/2002	01/10/2002	Other
8	20110016	E41	2003235	Structure Fire	01/11/2002	01/10/2002	Other
9	20110016	M03	2003235	Structure Fire	01/11/2002	01/10/2002	Other

10 rows | 0.66 seconds runtime | Refreshed now



## 9 Data Transformation and Analysis Layer



Microsoft Azure databricks

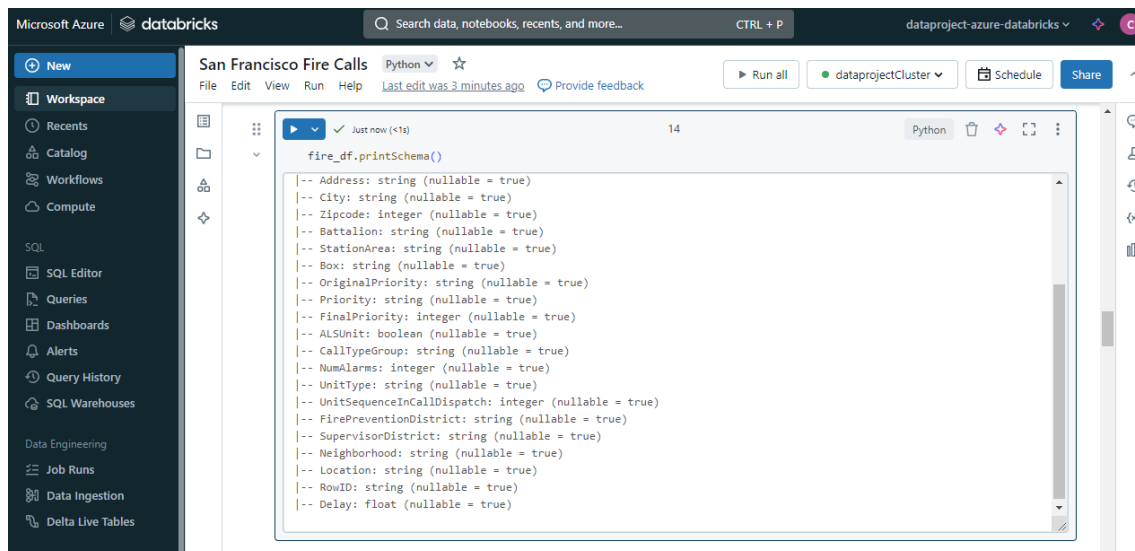
San Francisco Fire Calls Python

2. Data Transformation and Data Analysis

Cache the DataFrame since we will be performing some operations on it.

```
fire_df.cache()
```

DataFrame[CallNumber: int, UnitID: string, IncidentNumber: int, CallType: string, CallDate: string, WatchDate: string, CallFinalDisposition: string, AvailableDateTime: string, Address: string, City: string, Zipcode: int, Battalion: string, StationArea: string, Box: string, OriginalPriority: string, Priority: string, FinalPriority: int, ALSUnit: boolean, CallTypeGroup: string, NumAlarms: int, UnitType: string, UnitSequenceInCallDispatch: int, FirePreventionDistrict: string, SupervisorDistrict: string, Neighborhood: string, Location: string, RowID: string, Delay: float]



Microsoft Azure databricks

San Francisco Fire Calls Python

```
fire_df.printSchema()
```

```
-- Address: string (nullable = true)
-- City: string (nullable = true)
-- Zipcode: integer (nullable = true)
-- Battalion: string (nullable = true)
-- StationArea: string (nullable = true)
-- Box: string (nullable = true)
-- OriginalPriority: string (nullable = true)
-- Priority: string (nullable = true)
-- FinalPriority: integer (nullable = true)
-- ALSUnit: boolean (nullable = true)
-- CallTypeGroup: string (nullable = true)
-- NumAlarms: integer (nullable = true)
-- UnitType: string (nullable = true)
-- UnitSequenceInCallDispatch: integer (nullable = true)
-- FirePreventionDistrict: string (nullable = true)
-- SupervisorDistrict: string (nullable = true)
-- Neighborhood: string (nullable = true)
-- Location: string (nullable = true)
-- RowID: string (nullable = true)
-- Delay: float (nullable = true)
```

# Data Engineering with Microsoft Azure Databricks

Microsoft Azure | databricks

Search data, notebooks, recents, and more... CTRL + P

workspaceproject-azure-databricks

New Workspace Recents Catalog Workflows Compute SQL SQL Editor Queries Dashboards Alerts Query History SQL Warehouses Data Engineering Job Runs Data Ingestion Delta Live Tables Machine Learning Playground Experiments

San Francisco Fire Calls Python ☆

File Edit View Run Help Last edit was 5 minutes ago Provide feedback

Run all dataprojectCluster Schedule Share

Filter out "Medical Incident" call types

`filter()` and `where()` methods on the DataFrame are similar.

Auto-fix error in Chat: ON

```
few_fire_df = (fire_df.select("IncidentNumber", "AvailableDtTm", "CallType")
                    .where(col("CallType") != "Medical Incident"))

few_fire_df.show(5, truncate=False)
```

(1) Spark Jobs

few\_fire\_df: pyspark.sql.dataframe.DataFrame = [IncidentNumber: integer, AvailableDtTm: string ... 1 more field]

IncidentNumber	AvailableDtTm	CallType
2003234	01/11/2002 01:58:43 AM	Medical Incident
2003233	01/11/2002 02:10:17 AM	Medical Incident
2003236	01/11/2002 02:27:14 AM	Medical Incident
2003238	01/11/2002 02:28:30 AM	Medical Incident
2003240	01/11/2002 02:17:23 AM	Medical Incident

only showing top 5 rows

Microsoft Azure | databricks

Search data, notebooks, recents, and more... CTRL + P

workspaceproject-azure-databricks

New Workspace Recents Catalog Workflows Compute SQL SQL Editor Queries Dashboards Alerts Query History SQL Warehouses Data Engineering Job Runs Data Ingestion Delta Live Tables Machine Learning Playground Experiments

San Francisco Fire Calls Python ☆

File Edit View Run Help Last edit was 5 minutes ago Provide feedback

Run all dataprojectCluster Schedule Share

1) How many distinct types of calls were made to the Fire Department?

To be sure, let's not count "null" strings in that column.

Auto-fix error in Chat: ON

```
fire_df.select("CallType").where(col("CallType").isNotNull()).distinct().count()
```

(3) Spark Jobs

32

Microsoft Azure | databricks

Search data, notebooks, recents, and more... CTRL + P

workspaceproject-azure-databricks

New Workspace Recents Catalog Workflows Compute SQL SQL Editor Queries Dashboards Alerts Query History SQL Warehouses Data Engineering Job Runs Data Ingestion Delta Live Tables Machine Learning Playground Experiments

San Francisco Fire Calls Python ☆

File Edit View Run Help Last edit was 6 minutes ago Provide feedback

Run all dataprojectCluster Schedule Share

2) What are distinct types of calls were made to the Fire Department?

These are all the distinct type of call to the SF Fire Department

Auto-fix error in Chat: ON

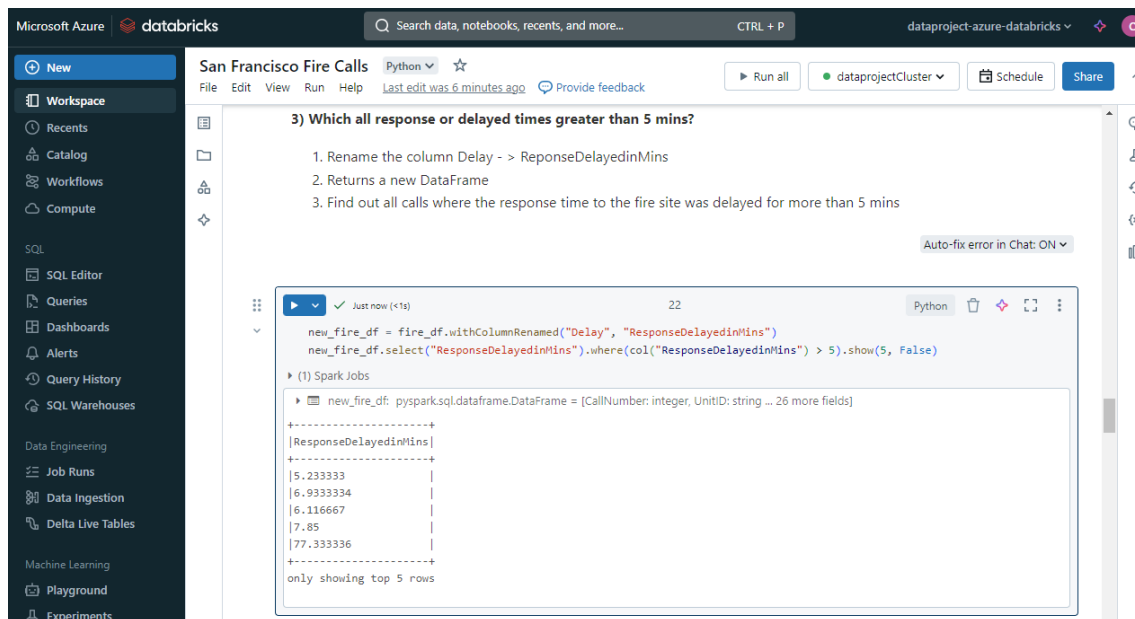
```
fire_df.select("CallType").where(col("CallType").isNotNull()).distinct().show(10, False)
```

(2) Spark Jobs

CallType
Elevator / Escalator Rescue
Alarms
Odor (Strange / Unknown)
Citizen Assist / Service Call
HazMat
Oil Spill
Vehicle Fire
Other
Outside Fire
Gas Leak (Natural and LP Gases)

only showing top 10 rows

# Data Engineering with Microsoft Azure Databricks



Microsoft Azure | databricks

Search data, notebooks, recents, and more... CTRL + P

dataport-azure-databricks

New

Workspace

Recents

Catalog

Workflows

Compute

SQL

SQL Editor

Queries

Dashboards

Alerts

Query History

SQL Warehouses

Data Engineering

Job Runs

Data Ingestion

Delta Live Tables

Machine Learning

Playground

Experiments

San Francisco Fire Calls Python

File Edit View Run Help Last edit was 6 minutes ago Provide feedback

Run all dataportCluster Schedule Share

3) Which all response or delayed times greater than 5 mins?

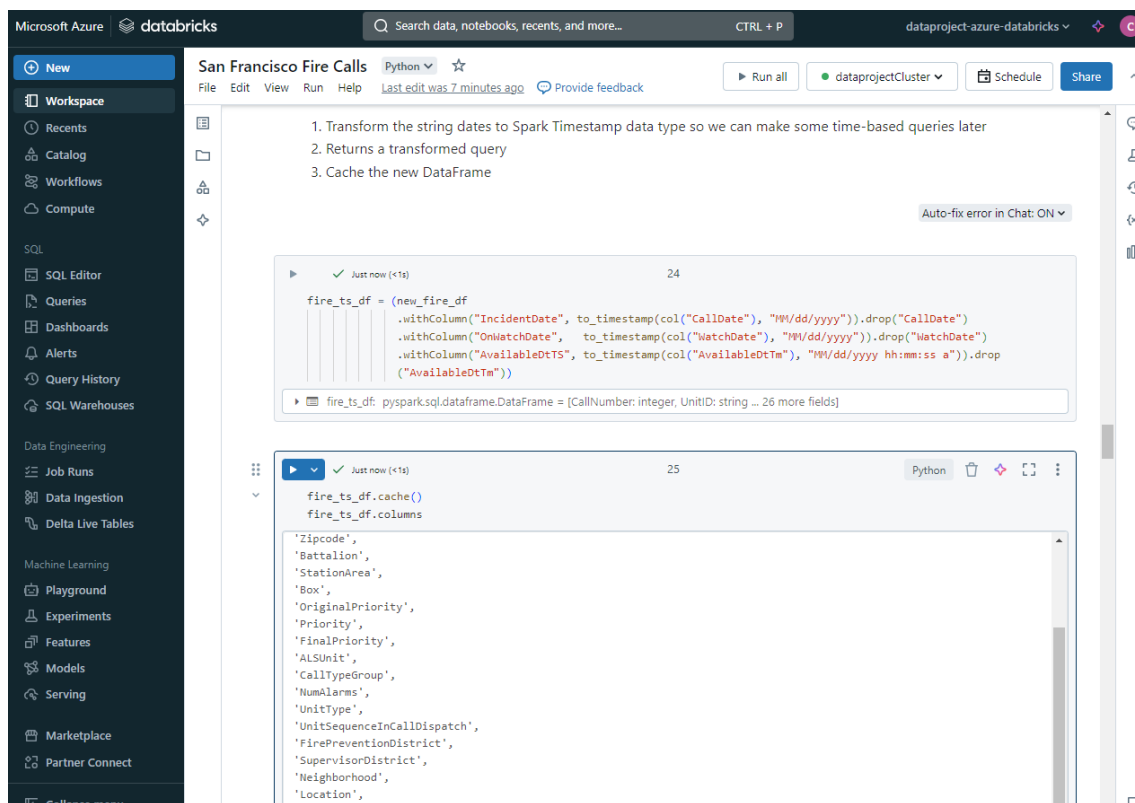
1. Rename the column Delay -> ReponseDelayedInMins
2. Returns a new DataFrame
3. Find out all calls where the response time to the fire site was delayed for more than 5 mins

Auto-fix error in Chat: ON

```
new_fire_df = fire_df.withColumnRenamed("Delay", "ResponseDelayedInMins")
new_fire_df.select("ResponseDelayedInMins").where(col("ResponseDelayedInMins") > 5).show(5, False)
```

(1) Spark Jobs

```
new_fire_df: pyspark.sql.dataframe.DataFrame = [CallNumber: integer, UnitID: string ... 26 more fields]
+-----+
|ResponseDelayedInMins|
+-----+
|5.233333|
|6.9333334|
|6.116667|
|7.85|
|77.333336|
+-----+
only showing top 5 rows
```



Microsoft Azure | databricks

Search data, notebooks, recents, and more... CTRL + P

dataport-azure-databricks

New

Workspace

Recents

Catalog

Workflows

Compute

SQL

SQL Editor

Queries

Dashboards

Alerts

Query History

SQL Warehouses

Data Engineering

Job Runs

Data Ingestion

Delta Live Tables

Machine Learning

Playground

Experiments

Features

Models

Serving

Marketplace

Partner Connect

San Francisco Fire Calls Python

File Edit View Run Help Last edit was 7 minutes ago Provide feedback

Run all dataportCluster Schedule Share

1. Transform the string dates to Spark Timestamp data type so we can make some time-based queries later
2. Returns a transformed query
3. Cache the new DataFrame

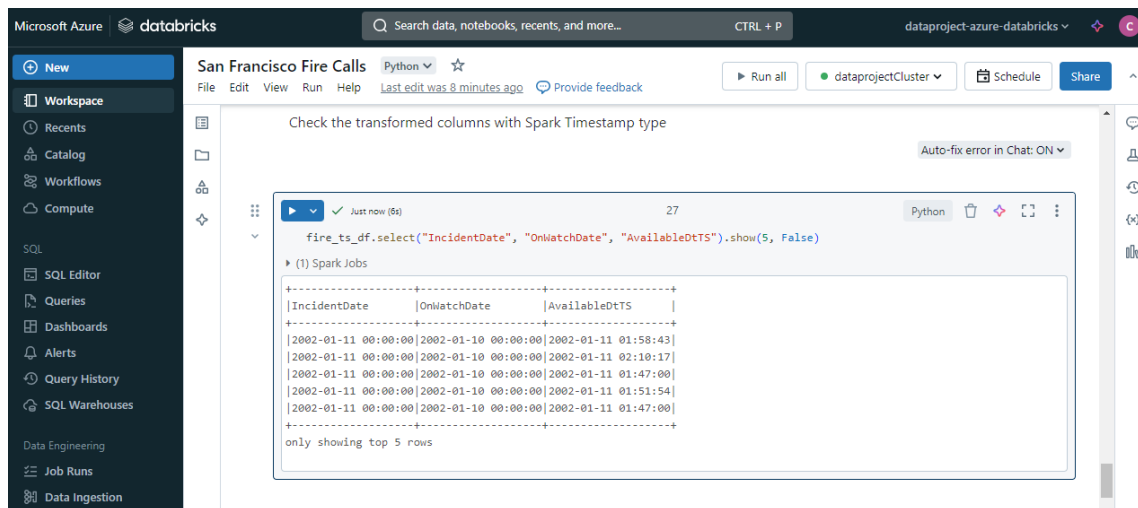
Auto-fix error in Chat: ON

```
fire_ts_df = (new_fire_df
              .withColumn("IncidentDate", to_timestamp(col("CallDate"), "MM/dd/yyyy")).drop("CallDate")
              .withColumn("OnWatchDate", to_timestamp(col("WatchDate"), "MM/dd/yyyy")).drop("WatchDate")
              .withColumn("AvailableDT", to_timestamp(col("AvailableDTm"), "MM/dd/yyyy hh:mm:ss a")).drop(
                "AvailableDTm"))
```

```
fire_ts_df: pyspark.sql.dataframe.DataFrame = [CallNumber: integer, UnitID: string ... 26 more fields]
```

```
fire_ts_df.cache()
fire_ts_df.columns
```

```
'Zipcode',
'Battalion',
'StationArea',
'Box',
'OriginalPriority',
'Priority',
'FinalPriority',
'ALUnit',
'CallTypeGroup',
'NumAlarms',
'UnitType',
'UnitSequenceInCallDispatch',
'FirePreventionDistrict',
'SupervisorDistrict',
'Neighborhood',
'Location',
```



Microsoft Azure databricks

San Francisco Fire Calls Python

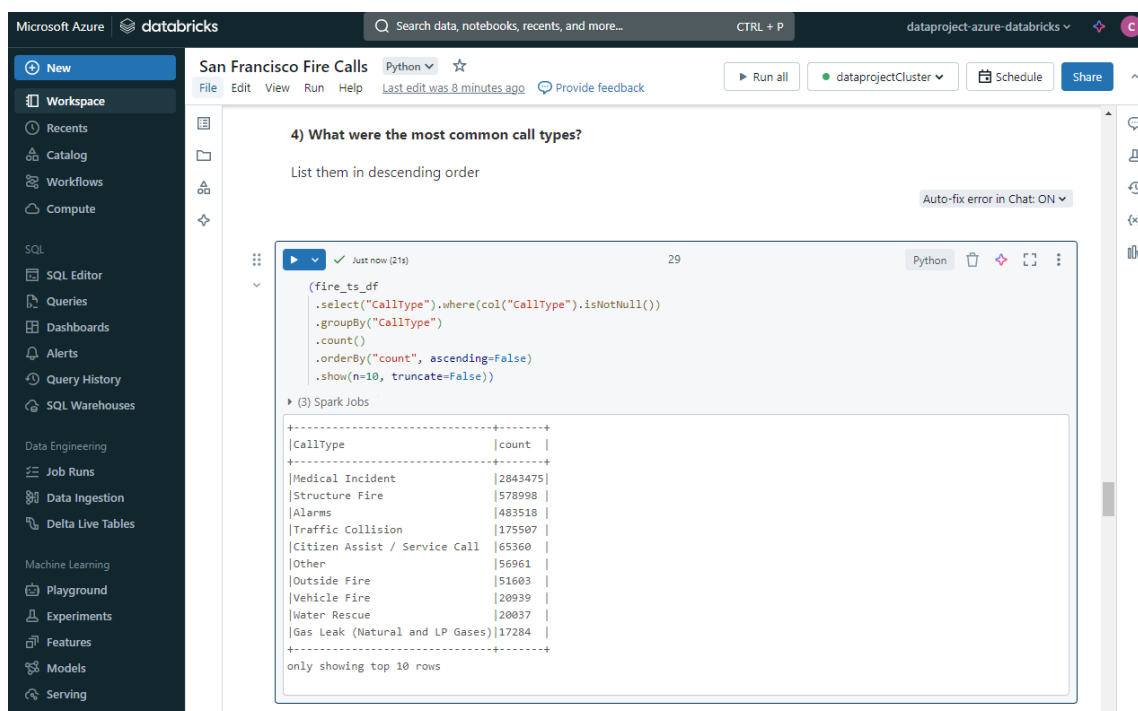
Check the transformed columns with Spark Timestamp type

```
fire_ts_df.select("IncidentDate", "OnWatchDate", "AvailableDtTS").show(5, False)
```

(1) Spark Jobs

IncidentDate	OnWatchDate	AvailableDtTS
2002-01-11 00:00:00	2002-01-10 00:00:00	2002-01-11 01:58:43
2002-01-11 00:00:00	2002-01-10 00:00:00	2002-01-11 02:10:17
2002-01-11 00:00:00	2002-01-10 00:00:00	2002-01-11 01:47:00
2002-01-11 00:00:00	2002-01-10 00:00:00	2002-01-11 01:51:54
2002-01-11 00:00:00	2002-01-10 00:00:00	2002-01-11 01:47:00

only showing top 5 rows



Microsoft Azure databricks

San Francisco Fire Calls Python

4) What were the most common call types?

List them in descending order

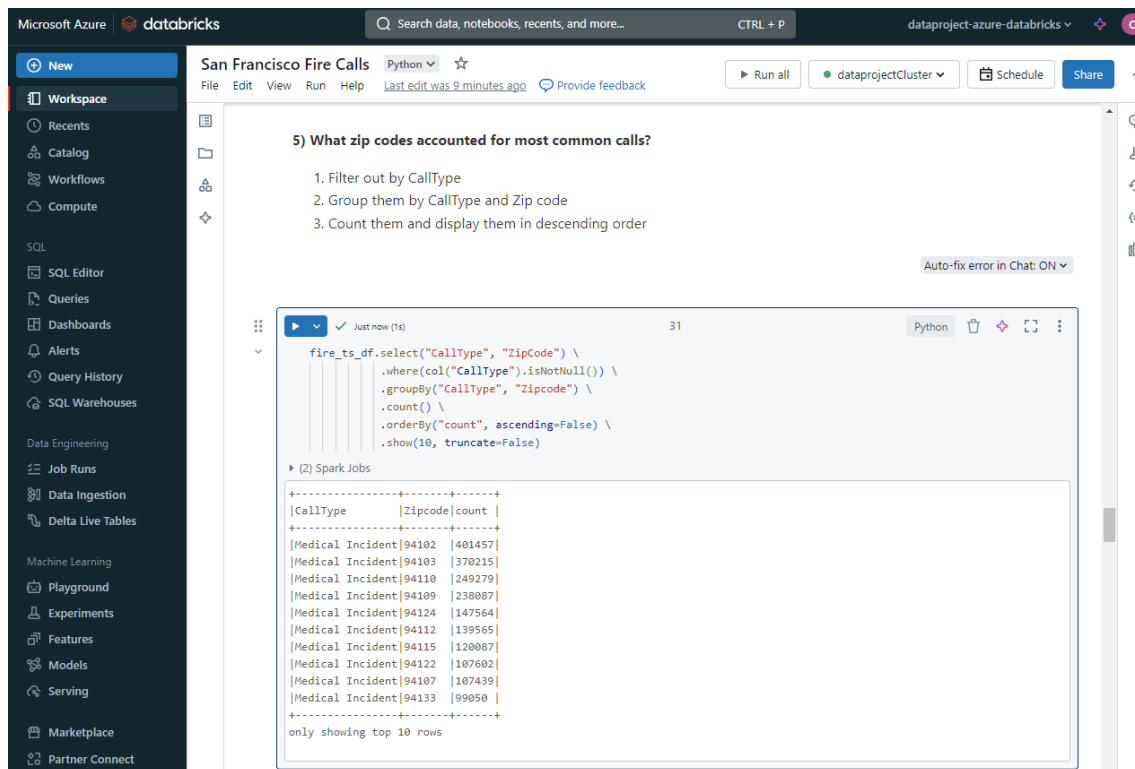
```
(fire_ts_df.select("CallType").where(col("CallType").isNotNull()).groupBy("CallType").count().orderBy("count", ascending=False).show(n=10, truncate=False))
```

(3) Spark Jobs

CallType	count
Medical Incident	2843475
Structure Fire	578998
Alarms	483518
Traffic Collision	175507
Citizen Assist / Service Call	65360
Other	56961
Outside Fire	51603
Vehicle Fire	20939
Water Rescue	20037
Gas Leak (Natural and LP Gases)	17284

only showing top 10 rows

# Data Engineering with Microsoft Azure Databricks



Microsoft Azure | databricks

Search data, notebooks, recents, and more... CTRL + P dataproject-azure-databricks

New Workspace Recents Catalog Workflows Compute SQL SQL Editor Queries Dashboards Alerts Query History SQL Warehouses Data Engineering Job Runs Data Ingestion Delta Live Tables Machine Learning Playground Experiments Features Models Serving Marketplace Partner Connect

### San Francisco Fire Calls

File Edit View Run Help Last edit was 9 minutes ago Provide feedback

Run all dataprojectCluster Schedule Share

5) What zip codes accounted for most common calls?

1. Filter out by CallType
2. Group them by CallType and Zip code
3. Count them and display them in descending order

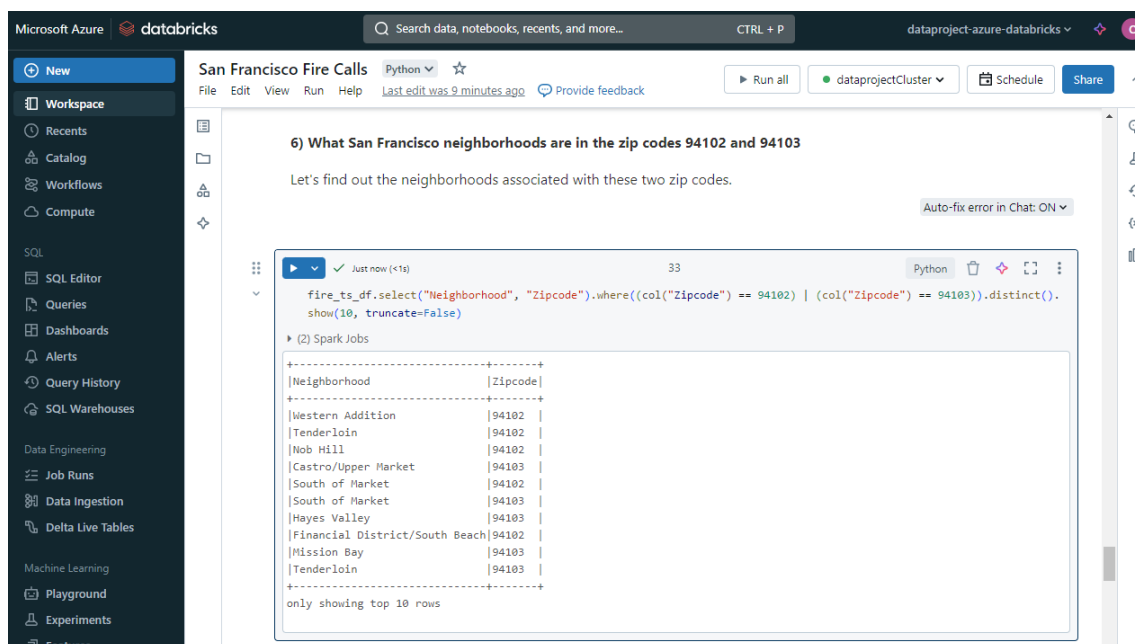
Auto-fix error in Chat: ON

```
fire_ts_df.select("CallType", "ZipCode") \
    .where(col("CallType").isNotNull()) \
    .groupBy("CallType", "Zipcode") \
    .count() \
    .orderBy("count", ascending=False) \
    .show(10, truncate=False)
```

(2) Spark Jobs

CallType	Zipcode	count
Medical Incident	94102	401457
Medical Incident	94103	370215
Medical Incident	94110	249279
Medical Incident	94109	238087
Medical Incident	94124	147564
Medical Incident	94112	139565
Medical Incident	94115	120087
Medical Incident	94122	107602
Medical Incident	94107	107439
Medical Incident	94133	99050

only showing top 10 rows



Microsoft Azure | databricks

Search data, notebooks, recents, and more... CTRL + P dataproject-azure-databricks

New Workspace Recents Catalog Workflows Compute SQL SQL Editor Queries Dashboards Alerts Query History SQL Warehouses Data Engineering Job Runs Data Ingestion Delta Live Tables Machine Learning Playground Experiments Features

### San Francisco Fire Calls

File Edit View Run Help Last edit was 9 minutes ago Provide feedback

Run all dataprojectCluster Schedule Share

6) What San Francisco neighborhoods are in the zip codes 94102 and 94103

Let's find out the neighborhoods associated with these two zip codes.

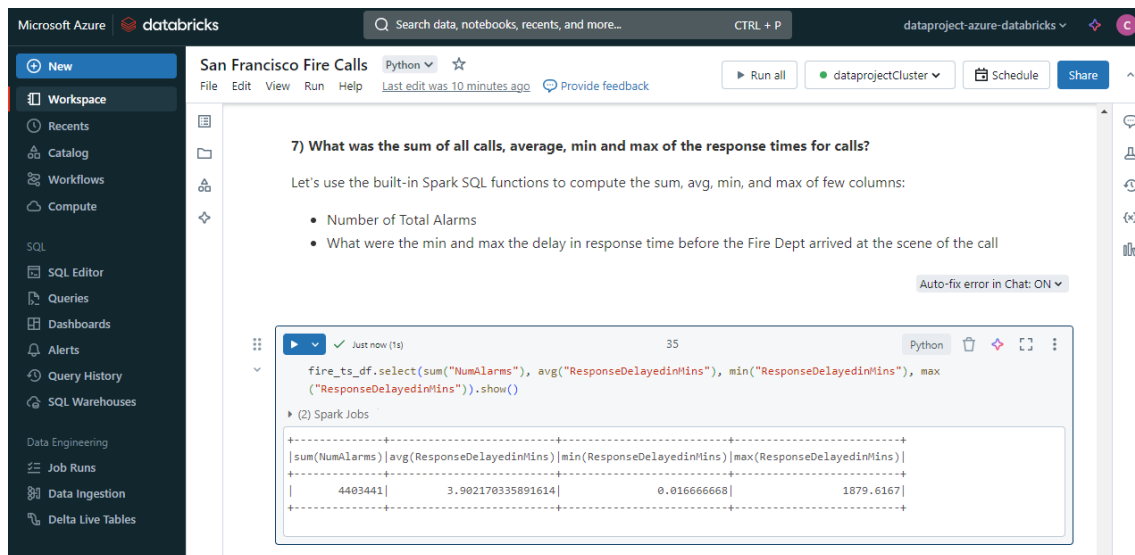
Auto-fix error in Chat: ON

```
fire_ts_df.select("Neighborhood", "Zipcode").where((col("Zipcode") == 94102) | (col("Zipcode") == 94103)).distinct().show(10, truncate=False)
```

(2) Spark Jobs

Neighborhood	Zipcode
Western Addition	94102
Tenderloin	94102
Nob Hill	94102
Castro/Upper Market	94103
South of Market	94103
South of Market	94103
Hayes Valley	94103
Financial District/South Beach	94102
Mission Bay	94103
Tenderloin	94103

only showing top 10 rows



Microsoft Azure databricks

Search data, notebooks, recents, and more... CTRL + P dataproject-azure-databricks

New Workspace Recents Catalog Workflows Compute SQL SQL Editor Queries Dashboards Alerts Query History SQL Warehouses Data Engineering Job Runs Data Ingestion Delta Live Tables

### San Francisco Fire Calls

File Edit View Run Help Last edit was 10 minutes ago Provide feedback

Run all dataprojectCluster Schedule Share

7) What was the sum of all calls, average, min and max of the response times for calls?

Let's use the built-in Spark SQL functions to compute the sum, avg, min, and max of few columns:

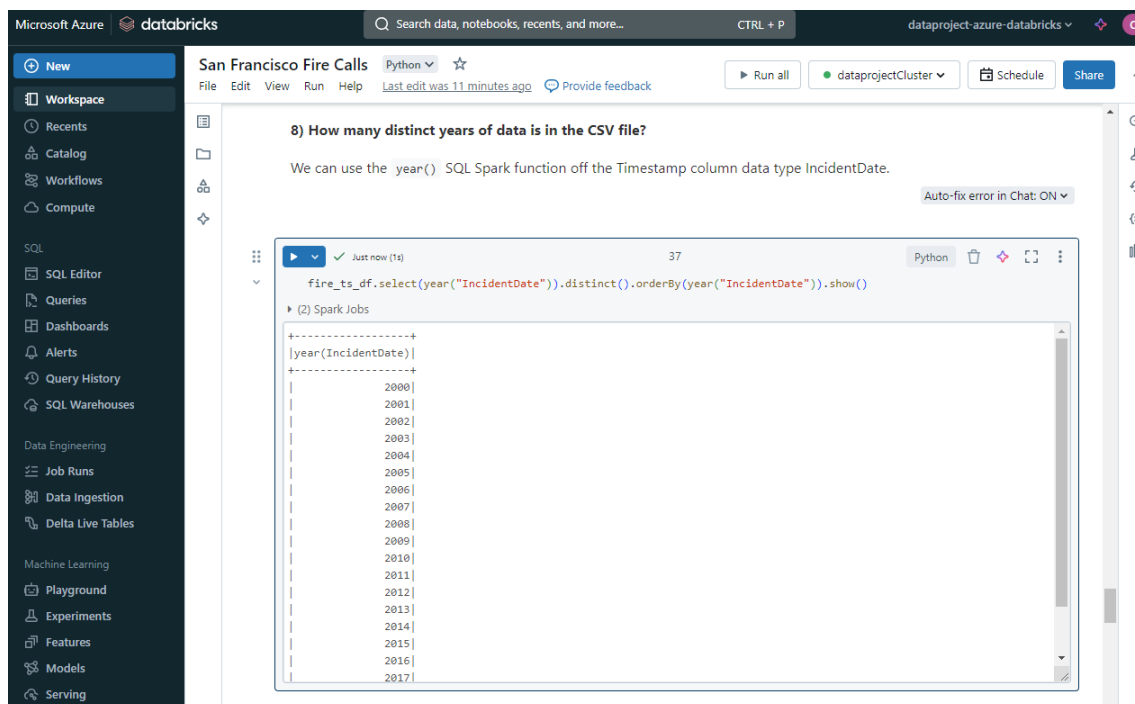
- Number of Total Alarms
- What were the min and max the delay in response time before the Fire Dept arrived at the scene of the call

Auto-fix error in Chat: ON

```
fire_ts_df.select(sum("NumAlarms"), avg("ResponseDelayedInMins"), min("ResponseDelayedInMins"), max("ResponseDelayedInMins")).show()
```

(2) Spark Jobs

sum(NumAlarms)	avg(ResponseDelayedInMins)	min(ResponseDelayedInMins)	max(ResponseDelayedInMins)
4403441	3.902170335891614	0.0166666668	1879.6167



Microsoft Azure databricks

Search data, notebooks, recents, and more... CTRL + P dataproject-azure-databricks

New Workspace Recents Catalog Workflows Compute SQL SQL Editor Queries Dashboards Alerts Query History SQL Warehouses Data Engineering Job Runs Data Ingestion Delta Live Tables Machine Learning Playground Experiments Features Models Serving

### San Francisco Fire Calls

File Edit View Run Help Last edit was 11 minutes ago Provide feedback

Run all dataprojectCluster Schedule Share

8) How many distinct years of data is in the CSV file?

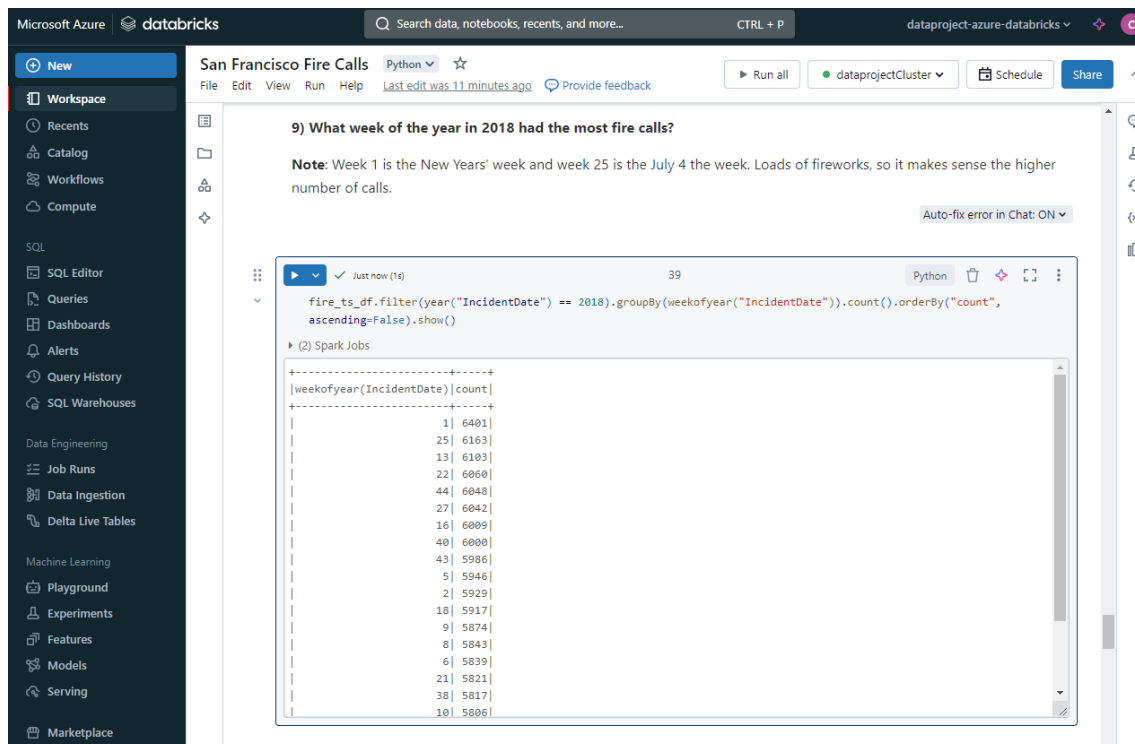
We can use the `year()` SQL Spark function off the Timestamp column data type IncidentDate.

Auto-fix error in Chat: ON

```
fire_ts_df.select(year("IncidentDate")).distinct().orderBy(year("IncidentDate")).show()
```

(2) Spark Jobs

year(IncidentDate)
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017



Microsoft Azure | databricks

Search data, notebooks, recents, and more... CTRL + P dataproject-azure-databricks

New Workspace Recents Catalog Workflows Compute SQL SQL Editor Queries Dashboards Alerts Query History SQL Warehouses Data Engineering Job Runs Data Ingestion Delta Live Tables Machine Learning Playground Experiments Features Models Serving Marketplace

### San Francisco Fire Calls

Python Last edit was 11 minutes ago Provide feedback Run all dataprojectCluster Schedule Share

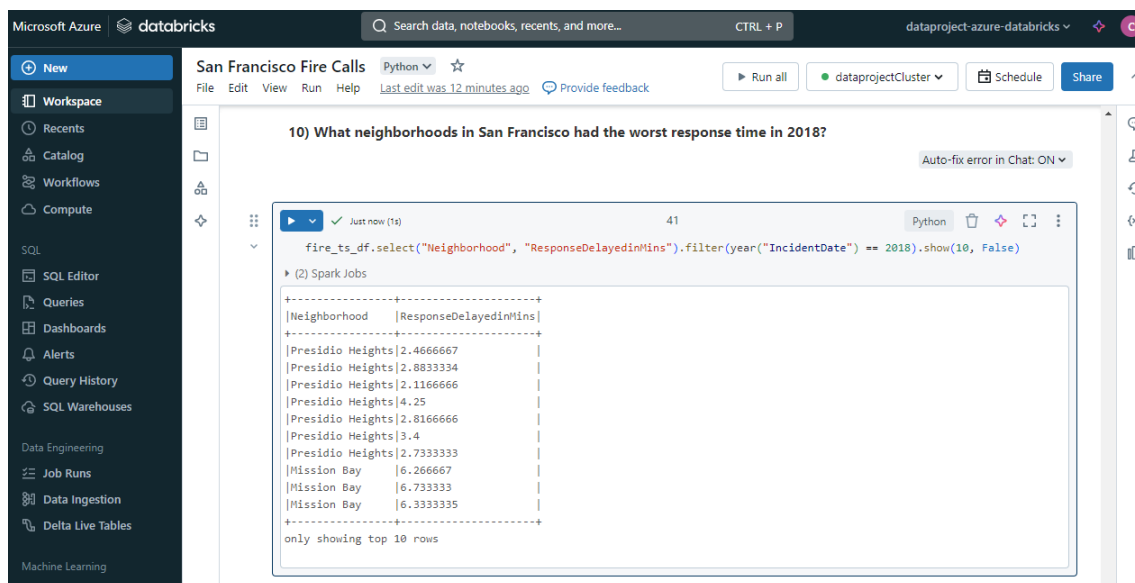
9) What week of the year in 2018 had the most fire calls?

Note: Week 1 is the New Years' week and week 25 is the July 4 the week. Loads of fireworks, so it makes sense the higher number of calls.

```
fire_ts_df.filter(year("IncidentDate") == 2018).groupBy(weekofyear("IncidentDate")).count().orderBy("count", ascending=False).show()
```

(2) Spark Jobs

weekofyear(IncidentDate)	count
1	6401
25	6163
13	6103
22	6060
44	6048
27	6042
16	6009
40	6000
43	5986
5	5946
2	5929
18	5917
9	5874
8	5843
6	5839
21	5821
38	5817
10	5806



Microsoft Azure | databricks

Search data, notebooks, recents, and more... CTRL + P dataproject-azure-databricks

New Workspace Recents Catalog Workflows Compute SQL SQL Editor Queries Dashboards Alerts Query History SQL Warehouses Data Engineering Job Runs Data Ingestion Delta Live Tables Machine Learning Playground Experiments Features Models Serving Marketplace

### San Francisco Fire Calls

Python Last edit was 12 minutes ago Provide feedback Run all dataprojectCluster Schedule Share

10) What neighborhoods in San Francisco had the worst response time in 2018?

```
fire_ts_df.select("Neighborhood", "ResponseDelayedinMins").filter(year("IncidentDate") == 2018).show(10, False)
```

(2) Spark Jobs

Neighborhood	ResponseDelayedinMins
Presidio Heights	2.4666667
Presidio Heights	2.8833334
Presidio Heights	2.1166666
Presidio Heights	4.25
Presidio Heights	2.8166666
Presidio Heights	3.4
Presidio Heights	2.7333333
Mission Bay	6.2666667
Mission Bay	6.7333333
Mission Bay	6.3333335

only showing top 10 rows

## 10 Data Storage Layer

**San Francisco Fire Calls** Python

File Edit View Run Help Last edit was now Provide feedback

Run all dataprojectCluster Schedule Share

### 3. Data Loading

Load the transformed data to persistent storage, so that it's query-able across notebooks and clusters

```
fire_ts_df.write.format("parquet").mode("overwrite").partitionBy("Neighborhood").save("/mnt/inputdataset/parquet/sf-fire-calls.parquet")
```

(1) Spark Jobs

Just now (<1s) 44

%fs ls /mnt/inputdataset/parquet/sf-fire-calls.parquet

Table	path	name	size	modification
1	> dbfs/mnt/inputdataset/parquet/sf-fire-calls.parquet/Neig...	Neighborhood=Bayview Hunters Point/	0	
2	> dbfs/mnt/inputdataset/parquet/sf-fire-calls.parquet/Neig...	Neighborhood=Bernal Heights/	0	
3	> dbfs/mnt/inputdataset/parquet/sf-fire-calls.parquet/Neig...	Neighborhood=Castro%2FUpper Market/	0	
4	> dbfs/mnt/inputdataset/parquet/sf-fire-calls.parquet/Neig...	Neighborhood=Chinatown/	0	
5	> dbfs/mnt/inputdataset/parquet/sf-fire-calls.parquet/Neig...	Neighborhood=Excelsior/	0	
6	> dbfs/mnt/inputdataset/parquet/sf-fire-calls.parquet/Neig...	Neighborhood=Financial District%2FSouth Beach/	0	
7	> dbfs/mnt/inputdataset/parquet/sf-fire-calls.parquet/Neig...	Neighborhood=Glen Park/	0	
8	> dbfs/mnt/inputdataset/parquet/sf-fire-calls.parquet/Neig...	Neighborhood=Golden Gate Park/	0	
9	> dbfs/mnt/inputdataset/parquet/sf-fire-calls.parquet/Neig...	Neighborhood=Haight Ashbury/	0	
10	> dbfs/mnt/inputdataset/parquet/sf-fire-calls.parquet/Neig...	Neighborhood=Hayes Valley/	0	
11	> dbfs/mnt/inputdataset/parquet/sf-fire-calls.parquet/Neig...	Neighborhood=Inner Richmond/	0	
12	> dbfs/mnt/inputdataset/parquet/sf-fire-calls.parquet/Neig...	Neighborhood=Inner Sunset/	0	
13	> dbfs/mnt/inputdataset/parquet/sf-fire-calls.parquet/Neig...	Neighborhood=Japantown/	0	
14	> dbfs/mnt/inputdataset/parquet/sf-fire-calls.parquet/Neig...	Neighborhood=Lakeshore/	0	

Saving data in parquet format to the Databricks file system.

Microsoft Azure Search resources, services, and docs (G+)

Home > dataprojectst | Containers >

**inputdataset** Container

Search Upload Add Directory Refresh Rename Delete Change tier Acquire lease Break lease

Overview Diagnose and solve problems Access Control (IAM) Settings

Authentication method: Access key (Switch to Microsoft Entra user account)  
Location: inputdataset

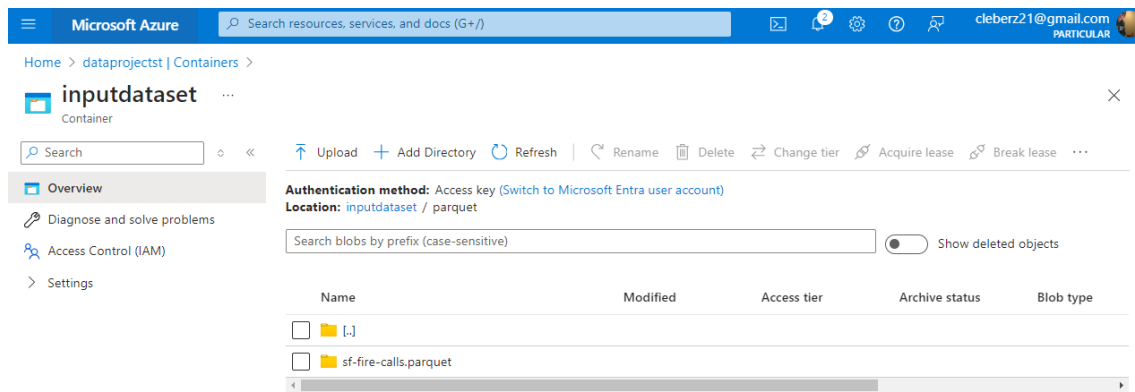
Search blobs by prefix (case-sensitive) Show deleted objects

Name	Modified	Access tier	Archive status	Blob type
<input type="checkbox"/> _\$azuretmpfolder\$				
<input type="checkbox"/> parquet				
<input type="checkbox"/> sf-fire-calls.csv	7/24/2024, 5:02:54 PM	Hot (Inferred)		Block blob

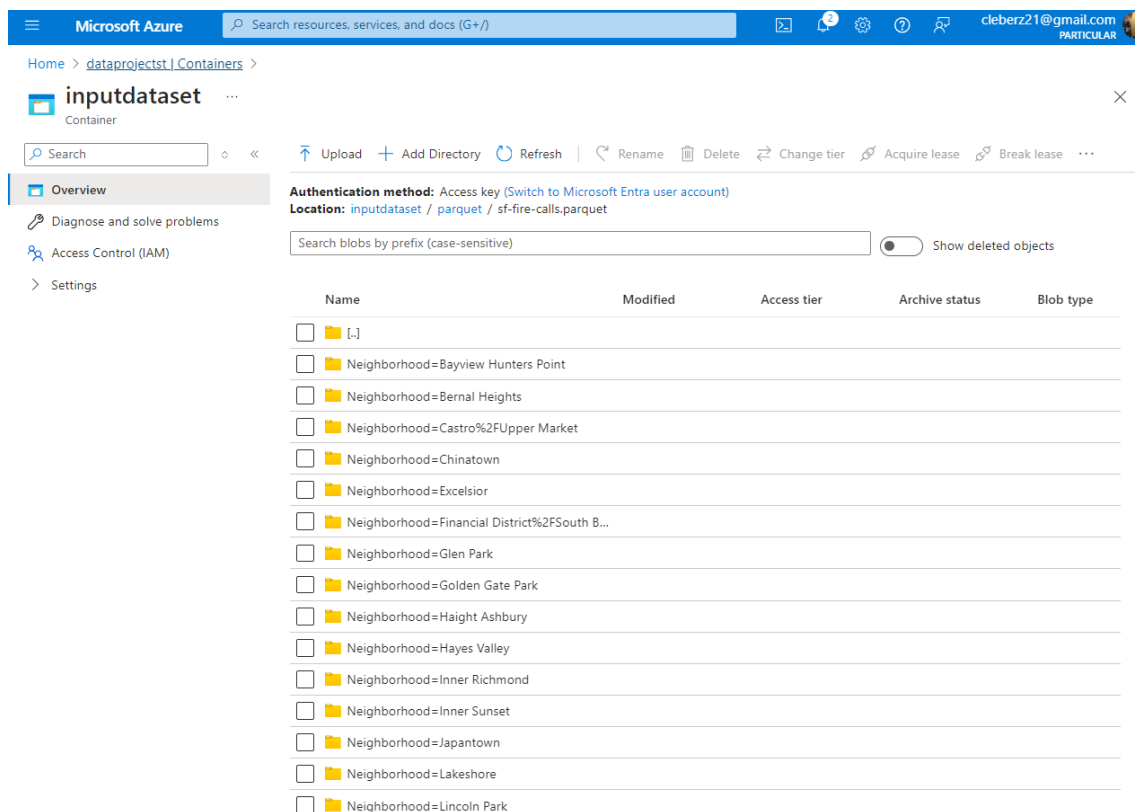
Data saved in parquet format in the Azure Data Lake Storage (ADLS) container.



# Data Engineering with Microsoft Azure Databricks



## Checking the data saved in the container



## Partitioned data in the container

# Data Engineering with Microsoft Azure Databricks

Microsoft Azure

Home > dataprojectst | Containers >

inputdataset

Container

Search

Upload Add Directory Refresh Rename Delete Change tier Acquire lease Break lease Give feedback

Overview

Diagnose and solve problems

Access Control (IAM)

Settings

Authentication method: Access key (Switch to Microsoft Entra user account)

Location: inputdataset / parquet / sf-fire-calls.parquet / Neighborhood=Chinatown

Search blobs by prefix (case-sensitive)

Name	Modified
[.]	
_committed_343785197172446366	7/24/2024, 6:01:16 PM
_started_343785197172446366	7/24/2024, 6:00:25 PM
_SUCCESS	7/24/2024, 6:01:16 PM
part-00000-tid-343785197172446366-41ab3922-a4d8-4492-b852-fdff7dc4d552-203-18.c000.snappy.parquet	7/24/2024, 6:00:26 PM
part-00001-tid-343785197172446366-41ab3922-a4d8-4492-b852-fdff7dc4d552-204-20.c000.snappy.parquet	7/24/2024, 6:00:27 PM
part-00002-tid-343785197172446366-41ab3922-a4d8-4492-b852-fdff7dc4d552-205-20.c000.snappy.parquet	7/24/2024, 6:00:27 PM
part-00003-tid-343785197172446366-41ab3922-a4d8-4492-b852-fdff7dc4d552-206-20.c000.snappy.parquet	7/24/2024, 6:00:26 PM
part-00004-tid-343785197172446366-41ab3922-a4d8-4492-b852-fdff7dc4d552-207-22.c000.snappy.parquet	7/24/2024, 6:00:53 PM
part-00005-tid-343785197172446366-41ab3922-a4d8-4492-b852-fdff7dc4d552-208-19.c000.snappy.parquet	7/24/2024, 6:00:52 PM
part-00006-tid-343785197172446366-41ab3922-a4d8-4492-b852-fdff7dc4d552-209-19.c000.snappy.parquet	7/24/2024, 6:00:52 PM
part-00007-tid-343785197172446366-41ab3922-a4d8-4492-b852-fdff7dc4d552-210-21.c000.snappy.parquet	7/24/2024, 6:00:53 PM
part-00008-tid-343785197172446366-41ab3922-a4d8-4492-b852-fdff7dc4d552-211-22.c000.snappy.parquet	7/24/2024, 6:01:09 PM

## Partitioned data in the container

Microsoft Azure databricks

Search data, notebooks, recents, and more... CTRL + P

dataproject-azure-databricks

New

Workspace

Recents

Catalog

Workflows

Compute

SQL

SQL Editor

Queries

Dashboards

Alerts

Query History

SQL Warehouses

Data Engineering

Job Runs

Data Ingestion

Delta Live Tables

Machine Learning

Playground

Experiments

Features

Models

Serving

Marketplace

Partner Connect

San Francisco Fire Calls Python

File Edit View Run Help Last edit was now Provide feedback

Run all dataprojectCluster Schedule Share

12) How can we save the data in Delta format?

Auto-fix error in Chat: ON

```
fire_ts_df.write.format("delta").mode("overwrite").partitionBy("Neighborhood").save("/mnt/inputdataset/delta/sf-fire-calls.delta")
```

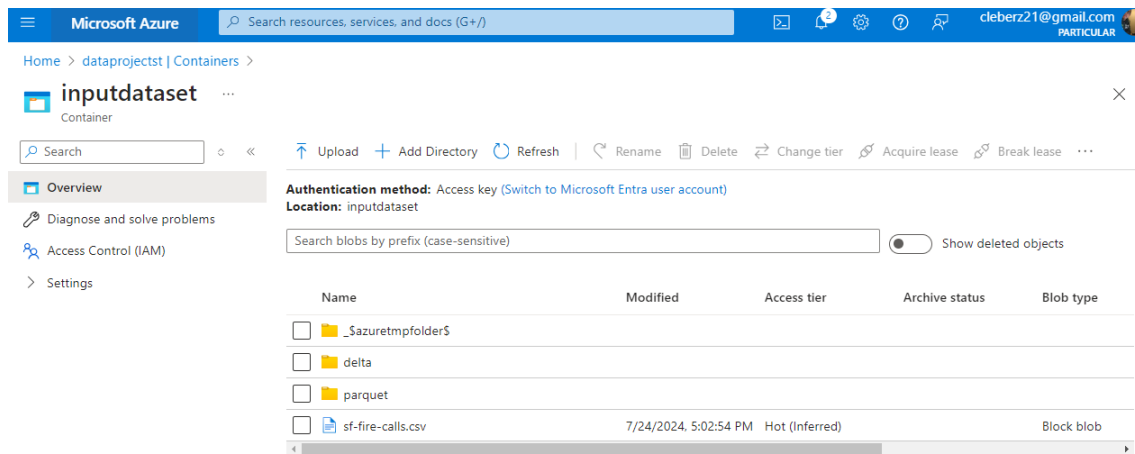
(2) Spark Jobs

```
%fs ls /mnt/inputdataset/delta/sf-fire-calls.delta
```

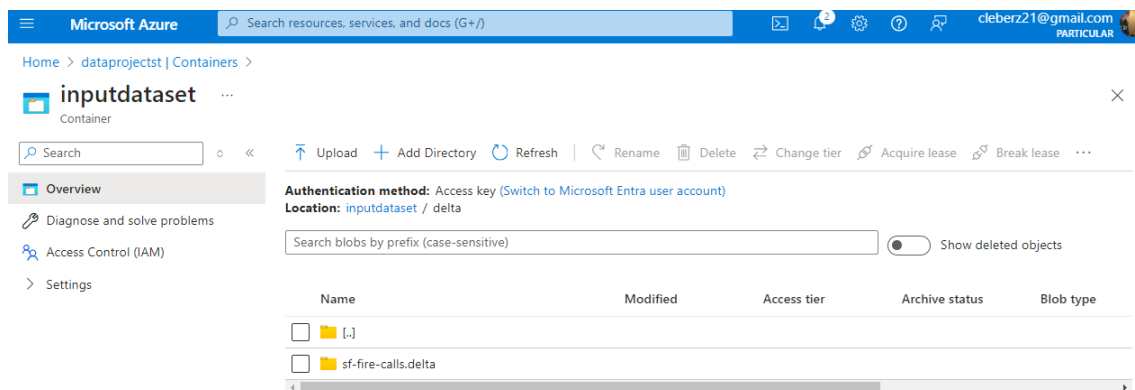
Table	path	name
1	dbfs/mnt/inputdataset/delta/sf-fire-calls.delta/Neighborhood=Bayview Hunters Point/	Neighborhood=Bayview Hunters Point/
2	dbfs/mnt/inputdataset/delta/sf-fire-calls.delta/Neighborhood=Bernal Heights/	Neighborhood=Bernal Heights/
3	dbfs/mnt/inputdataset/delta/sf-fire-calls.delta/Neighborhood=Castro%2FUpper Market/	Neighborhood=Castro%2FUpper Market/
4	dbfs/mnt/inputdataset/delta/sf-fire-calls.delta/Neighborhood=Chinatown/	Neighborhood=Chinatown/
5	dbfs/mnt/inputdataset/delta/sf-fire-calls.delta/Neighborhood=Excelsior/	Neighborhood=Excelsior/
6	dbfs/mnt/inputdataset/delta/sf-fire-calls.delta/Neighborhood=Financial District%2FSouth Beach/	Neighborhood=Financial District%2FSouth Beach/
7	dbfs/mnt/inputdataset/delta/sf-fire-calls.delta/Neighborhood=Glen Park/	Neighborhood=Glen Park/
8	dbfs/mnt/inputdataset/delta/sf-fire-calls.delta/Neighborhood=Golden Gate Park/	Neighborhood=Golden Gate Park/
9	dbfs/mnt/inputdataset/delta/sf-fire-calls.delta/Neighborhood=Haight Ashbury/	Neighborhood=Haight Ashbury/
10	dbfs/mnt/inputdataset/delta/sf-fire-calls.delta/Neighborhood=Hayes Valley/	Neighborhood=Hayes Valley/
11	dbfs/mnt/inputdataset/delta/sf-fire-calls.delta/Neighborhood=Inner Richmond/	Neighborhood=Inner Richmond/
12	dbfs/mnt/inputdataset/delta/sf-fire-calls.delta/Neighborhood=Inner Sunset/	Neighborhood=Inner Sunset/
13	dbfs/mnt/inputdataset/delta/sf-fire-calls.delta/Neighborhood=Japanatown/	Neighborhood=Japanatown/

Saving data in delta format to the Databricks file system.

# Data Engineering with Microsoft Azure Databricks



Data saved in delta format in Azure Data Lake Storage (ADLS) container.



Checking the data saved in the container

Microsoft Azure

Search resources, services, and docs (G+ /)

Home > dataprojectst | Containers >

inputdataset

Container

Search

<<

Upload Add Directory Refresh Rename Delete Change tier Acquire lease Break lease ...

Overview

Diagnose and solve problems

Access Control (IAM)

Settings

Authentication method:

 Access key ([Switch to Microsoft Entra user account](#))  

Location:

 inputdataset / delta / sf-fire-calls.delta

Search blobs by prefix (case-sensitive)

Show deleted objects

Name	Modified	Access tier	Archive status	Blob type
<input type="checkbox"/> [-]				
<input type="checkbox"/> Neighborhood=Bayview Hunters Point				
<input type="checkbox"/> Neighborhood=Bernal Heights				
<input type="checkbox"/> Neighborhood=Castro%2FUpper Market				
<input type="checkbox"/> Neighborhood=Chinatown				
<input type="checkbox"/> Neighborhood=Excelsior				
<input type="checkbox"/> Neighborhood=Financial District%2FSouth B...				
<input type="checkbox"/> Neighborhood=Glen Park				
<input type="checkbox"/> Neighborhood=Golden Gate Park				
<input type="checkbox"/> Neighborhood=Haight Ashbury				
<input type="checkbox"/> Neighborhood=Hayes Valley				
<input type="checkbox"/> Neighborhood=Inner Richmond				
<input type="checkbox"/> Neighborhood=Inner Sunset				
<input type="checkbox"/> Neighborhood=Japantown				
<input type="checkbox"/> Neighborhood=Lakeshore				
<input type="checkbox"/> Neighborhood=Lincoln Park				

**Microsoft Azure** Search resources, services, and docs (G+I)

Home > dataprojectst1.Containers >

## inputdataset

Container

Search

Upload Add Directory Refresh Rename Delete Change tier Acquire lease Break lease ...

**Authentication method:** Access key ([Switch to Microsoft Entra user account](#))  
**Location:** inputdataset / delta / sf-fire-calls.delta / \_delta\_log

Search blobs by prefix (case-sensitive) ☐ Show deleted objects

Name	Modified	Access tier	Archive status	Blob type
<input type="checkbox"/> _tmp_path_dir				
<input type="checkbox"/> 00000000000000000000.crc	7/24/2024, 6:08:43 PM	Hot (Inferred)		Block blob
<input type="checkbox"/> 00000000000000000000.json	7/24/2024, 6:08:42 PM	Hot (Inferred)		Block blob

Cleber Zumba  
Data Engineer

# Data Engineering with Microsoft Azure Databricks

The screenshot displays the Microsoft Azure portal interface. On the left, the 'inputdataset' container is selected, showing its overview and settings. The main pane displays the 'delta/sf-fire-calls.delta/\_delta\_log/00000000000000000000.json' file. The file content is a JSON array of log entries, each containing a timestamp, user ID, user name, and a list of paths. The paths are organized into a hierarchical structure, likely representing a file system or a database schema. The file is shown in a 'Blob' view, and the 'Overview' tab is selected.

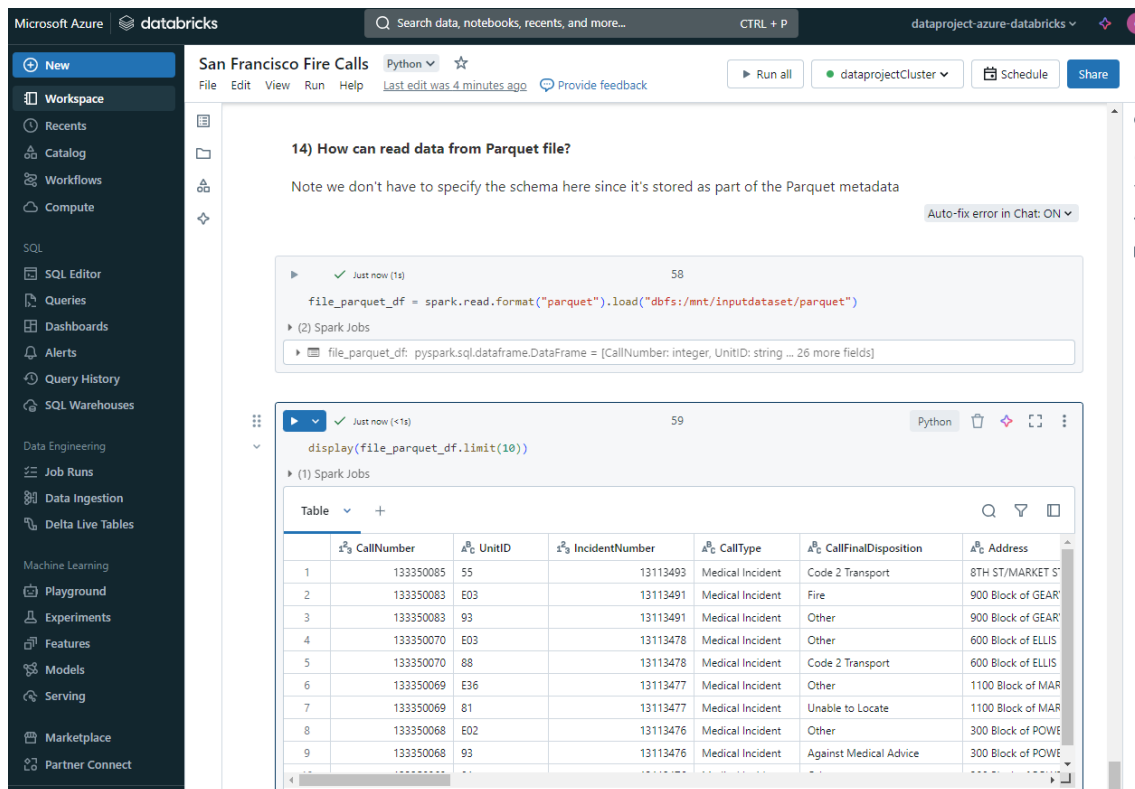
Viewing data in the container.

## 11 Data Query Layer

The screenshot displays the Microsoft Azure Databricks workspace. The left sidebar contains navigation options like New, Workspace, Recents, Catalog, Workflows, Compute, SQL, SQL Editor, Queries, Dashboards, Alerts, Query History, SQL Warehouses, Data Engineering, Job Runs, Data Ingestion, Delta Live Tables, Machine Learning, Playground, Experiments, Features, Models, Serving, Marketplace, and Partner Connect. The main area shows a notebook titled 'San Francisco Fire Calls' with a Python language selected. The notebook content includes a cell with a Spark job that writes data to a Delta table named 'FireServiceCalls'. Below this, another cell shows a SQL query: `%sql select * from FireServiceCalls limit 10`. The result of the query is displayed as a table with 10 rows and 7 columns: CallNumber, UnitID, IncidentNumber, CallType, CallFinalDisposition, and Address. The table shows various fire incidents, including traffic collisions and medical incidents. The bottom of the interface indicates that the result is stored as a `_sqldf` and can be used in other Python cells.

	CallNumber	UnitID	IncidentNumber	CallType	CallFinalDisposition	Address
1	40260133	E10	4007271	Traffic Collision	Other	DIVISADERO ST/PI
2	40260133	M10	4007271	Traffic Collision	Other	DIVISADERO ST/PI
3	40260133	M29	4007271	Traffic Collision	Other	DIVISADERO ST/PI
4	40260133	T03	4007271	Traffic Collision	Other	DIVISADERO ST/PI
5	40260138	E09	4007274	Medical Incident	Other	100 Block of BAY S
6	40260138	M32	4007274	Medical Incident	Other	100 Block of BAY S
7	40260138	RC4	4007274	Medical Incident	Other	100 Block of BAY S
8	40260139	B02	4007275	Structure Fire	Other	FRANKLIN ST/PINE
9	40260139	B04	4007275	Structure Fire	Other	FRANKLIN ST/PINE

Querying the result of data transformation and analysis in delta format.



**San Francisco Fire Calls** Python

Note we don't have to specify the schema here since it's stored as part of the Parquet metadata

```
file_parquet_df = spark.read.format("parquet").load("dbfs:/mnt/inputdataset/parquet")
```

```
display(file_parquet_df.limit(10))
```

	CallNumber	UnitID	IncidentNumber	CallType	CallFinalDisposition	Address
1	133350085	55	13113493	Medical Incident	Code 2 Transport	8TH ST/MARKET S
2	133350083	E03	13113491	Medical Incident	Fire	900 Block of GEAR
3	133350083	93	13113491	Medical Incident	Other	900 Block of GEAR
4	133350070	E03	13113478	Medical Incident	Other	600 Block of ELLIS
5	133350070	88	13113478	Medical Incident	Code 2 Transport	600 Block of ELLIS
6	133350069	E36	13113477	Medical Incident	Other	1100 Block of MAR
7	133350069	81	13113477	Medical Incident	Unable to Locate	1100 Block of MAR
8	133350068	E02	13113476	Medical Incident	Other	300 Block of POWE
9	133350068	93	13113476	Medical Incident	Against Medical Advice	300 Block of POWE

Querying the result of data transformation and analysis in parquet format.

# 12 Conclusion

In this project, I leveraged the powerful capabilities of AWS, Azure, and Databricks to build a robust and scalable data pipeline focused on analyzing and transforming emergency call data from the San Francisco Fire Department.

Performed data extraction, transformation, and loading (ETL) processes, highlighting the seamless integration between AWS S3, Azure Data Lake Storage (ADLS), and Databricks for scalable data processing. We used Azure Data Factory to orchestrate data movement and ensure secure transfer between cloud environments.

In Databricks, I used PySpark and SparkSQL to execute queries and transformations, demonstrating their ability to handle large-scale data analysis with ease. I implemented performance optimization techniques, such as caching, which are essential for achieving high-performance data processing.

I followed best practices using the advanced features of Azure Key Vault for secure secrets management and Databricks for distributed processing.

Cleber Zumba de Souza



## 13 Reference

PARSIAN, Mahmoud. **Data Algorithms with Spark**. Sebastopol, California, United States: O'Reilly Media, 2022.