

Ingestão e Análise de Dados com Azure Databricks

Sumário

1	Versão.....	3
2	Introdução.....	4
3	Ingestão de Dados.....	6
3.1	Introdução.....	6
3.2	Armazenamento Inicial no Amazon S3	6
4	Camadas de Provisionamento e Configuração de Serviços no Azure	7
4.1	Resource Group.....	7
4.2	Storage Account	8
4.3	Container.....	9
4.4	Key Vault	10
5	Transferência de Dados para o Azure Data Lake Storage (ADLS).....	13
6	Execução do Pipeline de Dados.....	18
7	Criação do Azure Databricks Workspace e Cluster	19
8	Camada de Extração de Dados.....	22
9	Camada de Transformação e Análise de Dados.....	25
10	Camada de Armazenamento de Dados.....	32
11	Camada de Consulta de Dados.....	38
12	Conclusão	40
13	Referência	41

1 Versão

Este documento foi criado por Cleber Zumba de Souza e pode ser distribuído livremente, desde que a fonte seja mencionada.

Versão	Ação	Data
1.0	Criação do documento	26/07/2024

2 Introdução

Neste projeto, desenvolvi uma solução de engenharia de dados integrando as tecnologias da AWS, Azure e Databricks. O objetivo principal foi criar um pipeline de dados eficiente e escalável que abrange desde a ingestão de dados até a transformação e análise, garantindo armazenamento seguro e processamento robusto. Os dados tratados neste pipeline são relativos às chamadas de emergência do Corpo de Bombeiros de São Francisco.

Contexto e Objetivos

Em ambientes modernos de dados, é essencial integrar diferentes plataformas de nuvem e ferramentas para otimizar o fluxo de trabalho de dados. Este projeto foi projetado para:

- **Ingestão de Dados:** Ler dados de um bucket S3 na AWS.
- **Armazenamento Centralizado:** Armazenar os dados em um Data Lake no Azure Data Lake Storage (ADLS).
- **Processamento e Análise:** Utilizar o Databricks para transformar, analisar e armazenar os dados transformados.

Visão Geral dos Dados

Os dados utilizados neste pipeline são registros das chamadas de emergência atendidas pelo Corpo de Bombeiros de São Francisco. Esses dados contêm informações críticas, como a natureza da chamada, o bairro onde ocorreu, e os tempos de resposta. Analisar esses dados é vital para melhorar os serviços de emergência e a segurança pública.

Visão Geral das Tecnologias Utilizadas

1. **AWS S3:** Serviço de armazenamento na nuvem da Amazon onde os dados brutos são inicialmente armazenados.
2. **Azure Resource Group:** Agrupa e gerencia todos os recursos relacionados ao projeto na Azure.
3. **Azure Storage Account:** Fornece armazenamento seguro e escalável para dados no Azure.
4. **Azure Data Factory:** Serviço de integração de dados utilizado para orquestrar e automatizar o movimento de dados entre os serviços AWS S3 e ADLS.
5. **Azure Data Lake Storage (ADLS):** Solução de armazenamento centralizada no Azure, que permite armazenamento eficiente e seguro dos dados.
6. **Databricks:** Plataforma de análise e processamento de dados baseada em Apache Spark, usada para transformar, analisar e armazenar dados.

Fluxo de Trabalho do Projeto

1. Ingestão de Dados com Azure Data Factory:

- Um pipeline de dados no Azure Data Factory lê o arquivo de dados armazenado no bucket S3 da AWS.
- Os dados são então transferidos e armazenados no Data Lake no Azure Data Lake Storage (ADLS).

2. Processamento e Análise com Databricks:

- O Databricks integra-se ao ADLS para acessar os dados armazenados.
- Utilizando o poder de processamento do Apache Spark, os dados são transformados e analisados conforme as necessidades do projeto.
- Os dados transformados são então armazenados de volta no ADLS ou em outros destinos conforme necessário.

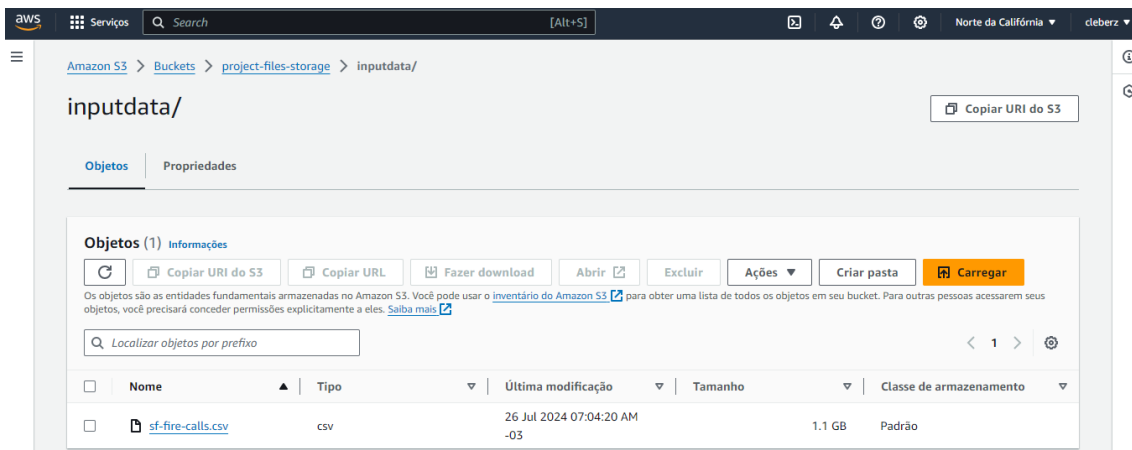
3 Ingestão de Dados

3.1 Introdução

Neste capítulo, detalhei o processo de ingestão de dados, desde a leitura do arquivo armazenado no bucket Amazon S3 até a transferência dos dados para o Azure Data Lake Storage (ADLS). Este é o primeiro passo no pipeline de dados e é crucial para garantir que os dados brutos estejam disponíveis para processamento e análise subsequentes.

3.2 Armazenamento Inicial no Amazon S3

Os dados de chamadas do Corpo de Bombeiros de São Francisco são inicialmente armazenados em um bucket Amazon S3. A imagem abaixo mostra a estrutura de armazenamento do arquivo `sf-fire-calls.csv` no bucket S3:

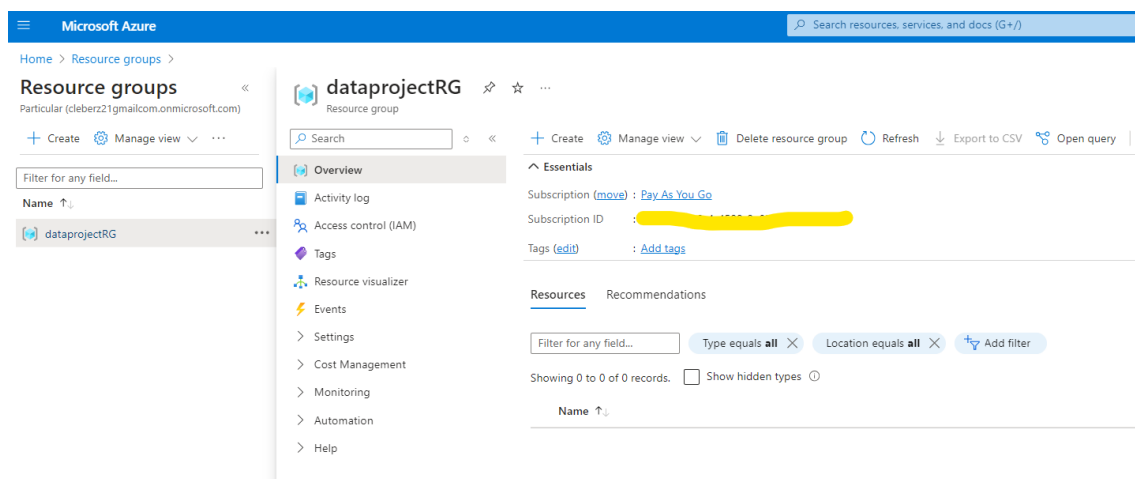


- **Bucket:** project-files-storage
- **Diretório:** inputdata
- **Arquivo:** sf-fire-calls.csv
- **Tamanho:** 1.1 GB
- **Última Modificação:** 26 de Julho de 2024, 07:04 AM

4 Camadas de Provisionamento e Configuração de Serviços no Azure

Antes de realizar a transferência de dados, foi necessário provisionar e configurar os serviços no Azure. As etapas incluem a criação de um Resource Group, uma Storage Account, um Container e um Key Vault

4.1 Resource Group



Um Resource Group foi criado para agrupar e gerenciar todos os recursos relacionados ao projeto na Azure. Isso facilita a organização e a administração dos recursos.

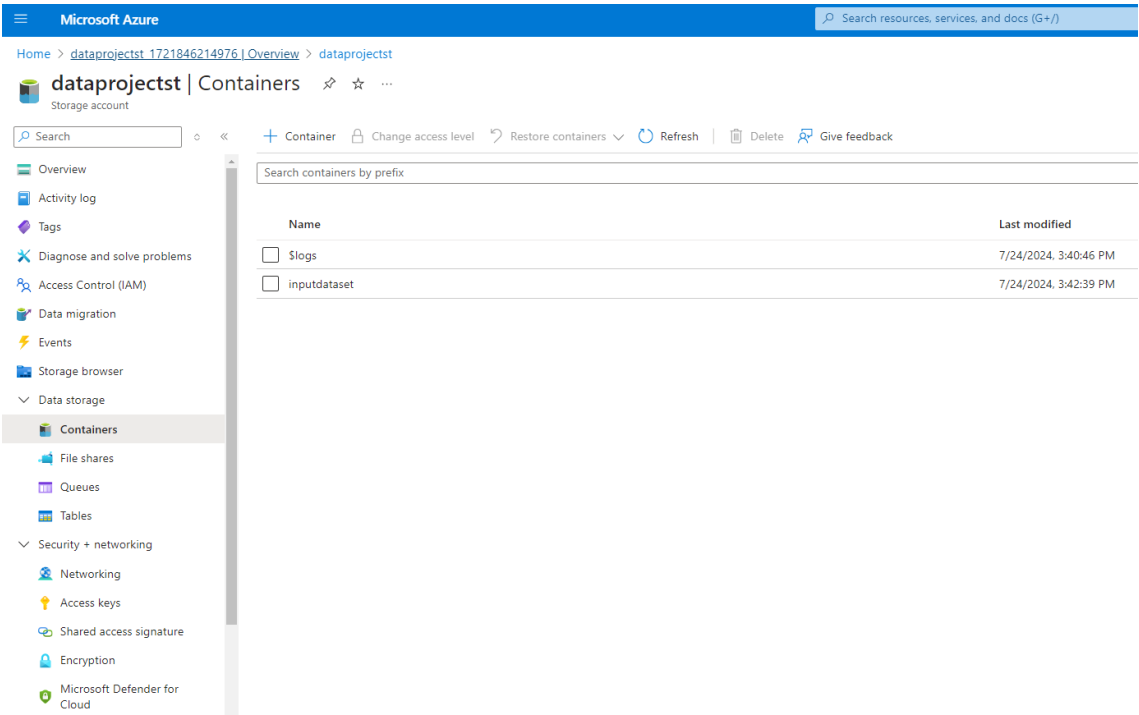
4.2 Storage Account

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 'Overview' tab is selected, showing a list of 'Essentials' for the storage account. These include the 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), and Tags (Add tags). Below the Essentials section, the 'Properties' tab is active, displaying a table of 'Data Lake Storage' settings.

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

Uma Storage Account foi criada para fornecer armazenamento seguro e escalável para os dados. Esta conta de armazenamento é essencial para armazenar os dados transferidos do S3.

4.3 Container

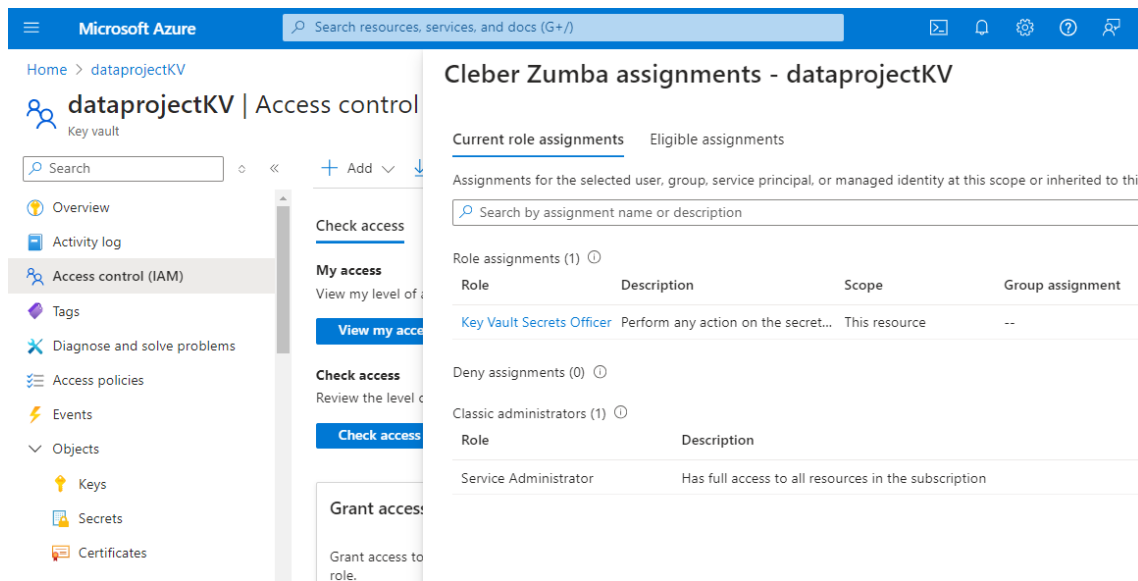


Dentro da Storage Account, um container foi criado para armazenar os dados no formato desejado.

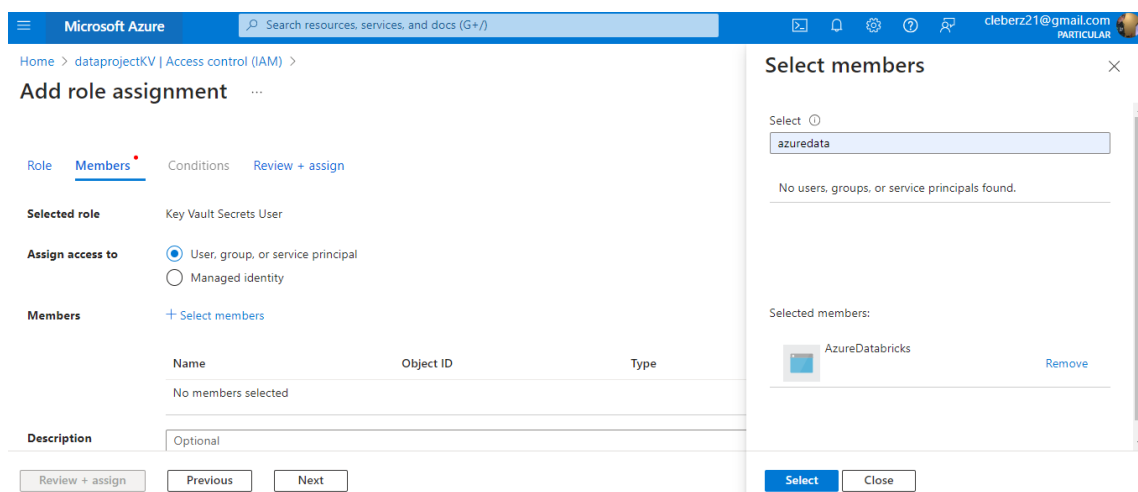
4.4 Key Vault

The screenshot displays the Microsoft Azure portal interface. At the top, a blue header bar contains the 'Microsoft Azure' logo. Below this, a breadcrumb trail shows 'Home >'. The main content area is titled 'dataprojectKV' with a 'Key vault' subtitle. A search bar is located below the title. On the left, a navigation pane lists various options: Overview (selected), Activity log, Access control (IAM), Tags, Diagnose and solve problems, Access policies, Events, Objects (expanded to show Keys, Secrets, and Certificates), Settings (expanded to show Access configuration, Networking, Microsoft Defender for Cloud, Properties, Locks), and Monitoring (expanded to show Alerts and Metrics). The main content area on the right shows the 'Overview' page for the Key Vault. It includes a search bar, action buttons (Delete, Move, Refresh, Open in mobile), and a list of properties: Location (Canada Central), Subscription (Pay As You Go), Subscription ID (redacted), and Tags (Add tags). Below the properties, there are tabs for 'Get started', 'Properties', 'Monitoring', 'Tools + SDKs', and 'Tutorials'.

Criação da Key Vault



Adicionando a role Key Vault Secrets Officer



Adicionando membro AzureDatabricks a role Key Vault Secrets Officer

Microsoft Azure

Home > dataprojectKV

dataprojectKV | Secrets

Search resources, services, and docs (G+)

Generate/Import Refresh Restore Backup View sample code Manage deleted secrets

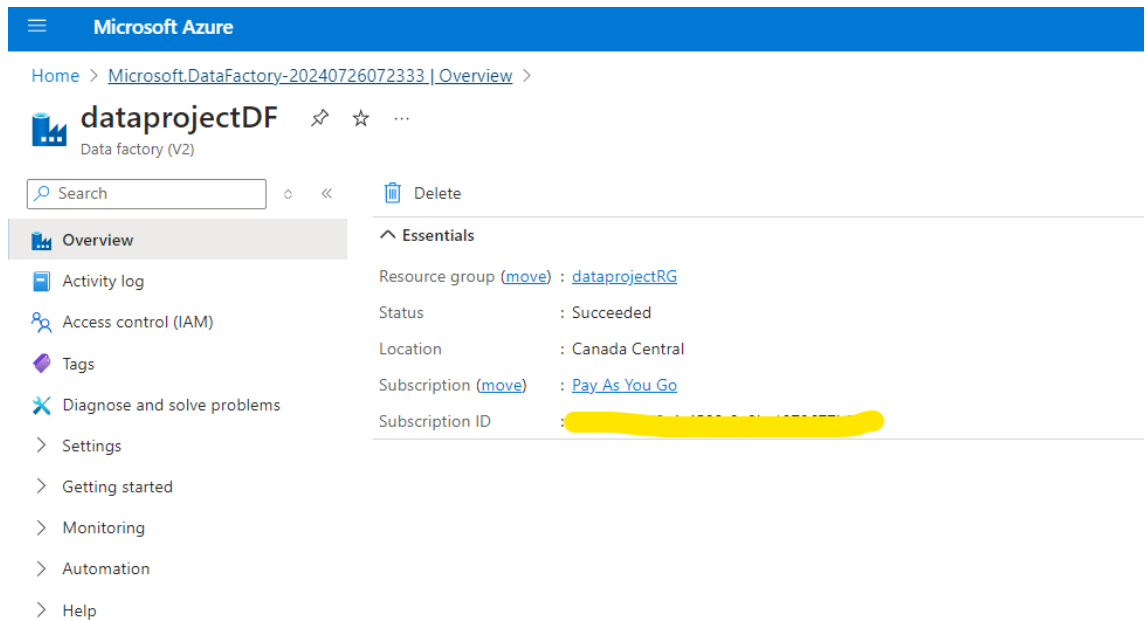
The secret 'dataproject-databrickskey' has been successfully created.

Name	Type	Status
dataproject-databrickskey		✓ Enabled

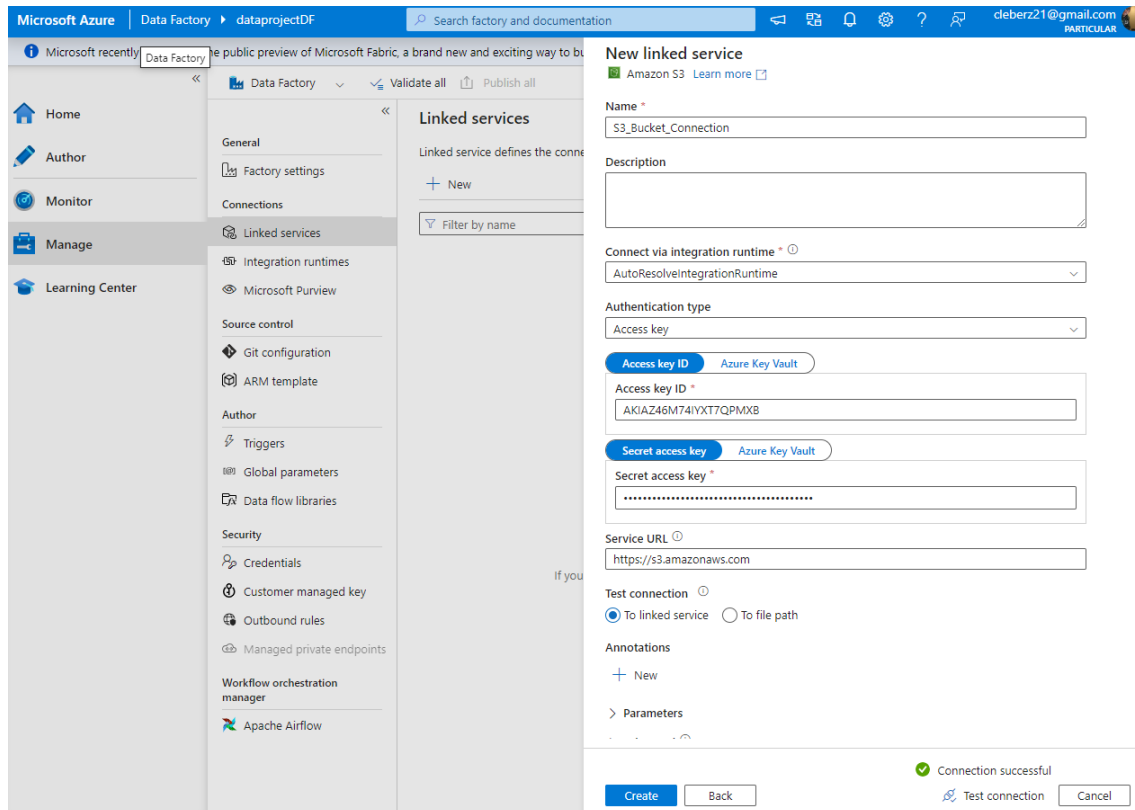
Criando uma secret na Kay Vault dataprojectKV

5 Transferência de Dados para o Azure Data Lake Storage (ADLS)

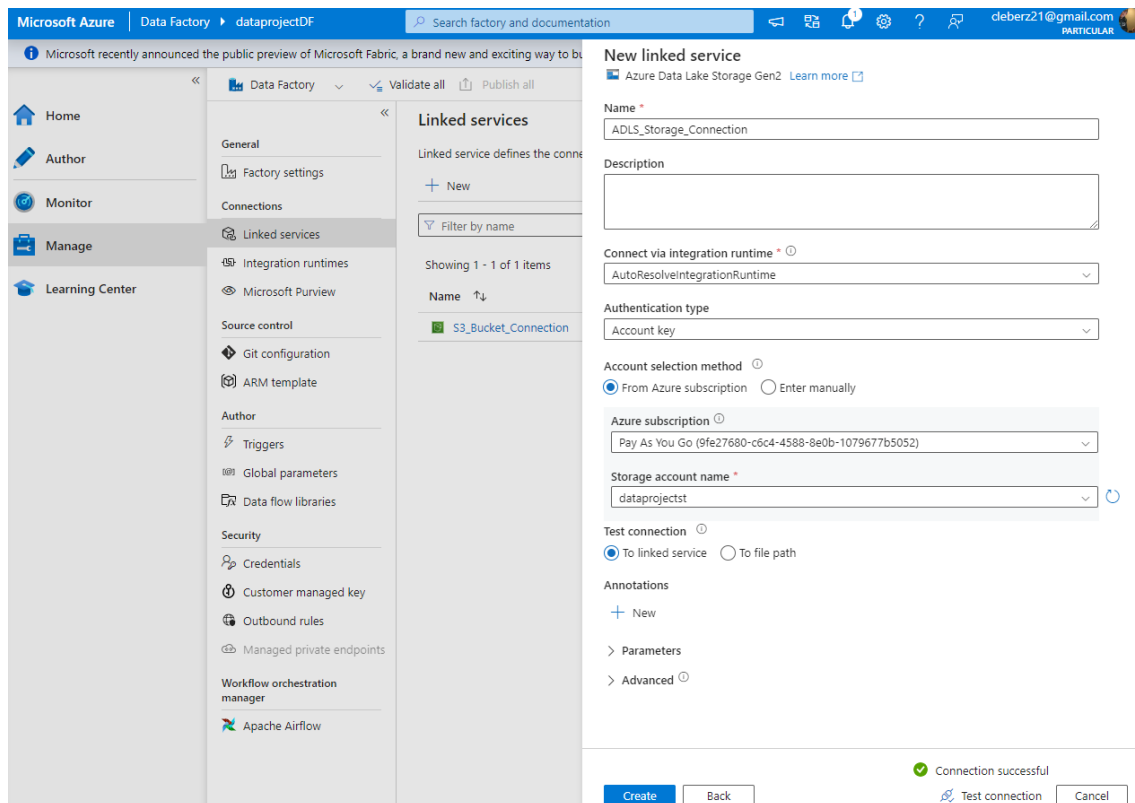
Utilizando o Azure Data Factory, configurei um pipeline de dados para ler o arquivo armazenado no bucket Amazon S3 e transferi-lo para o Azure Data Lake Storage (ADLS). Este processo envolve:



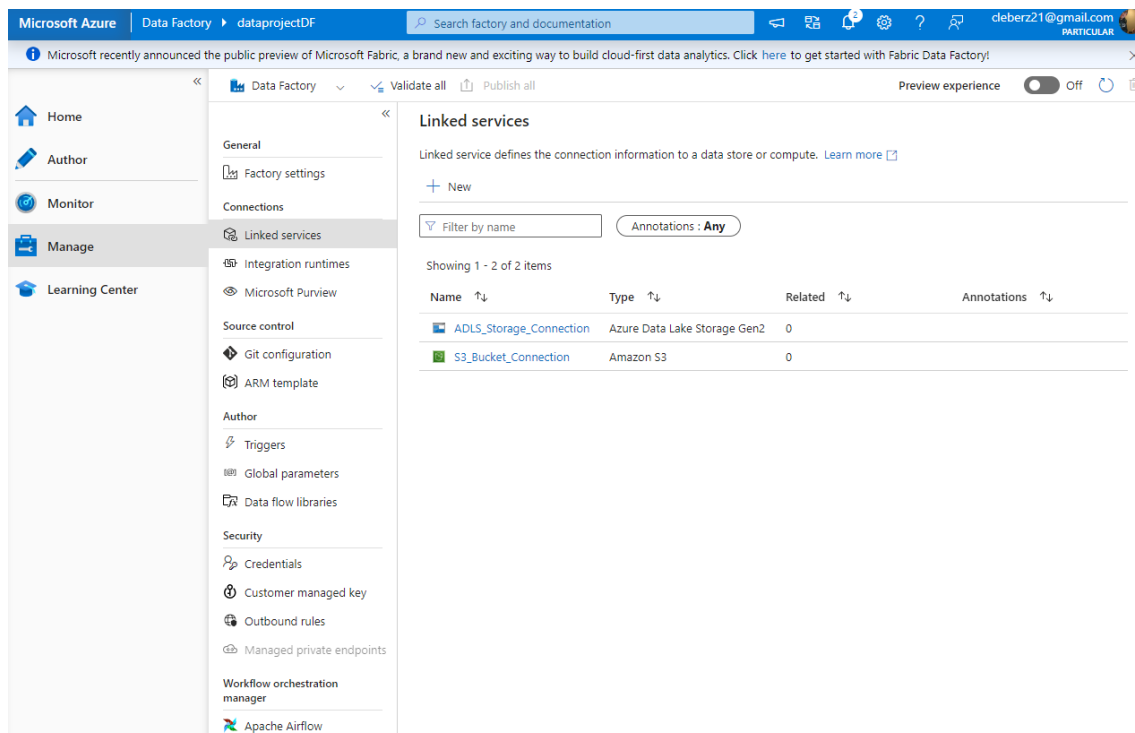
Criação de um Data Factory



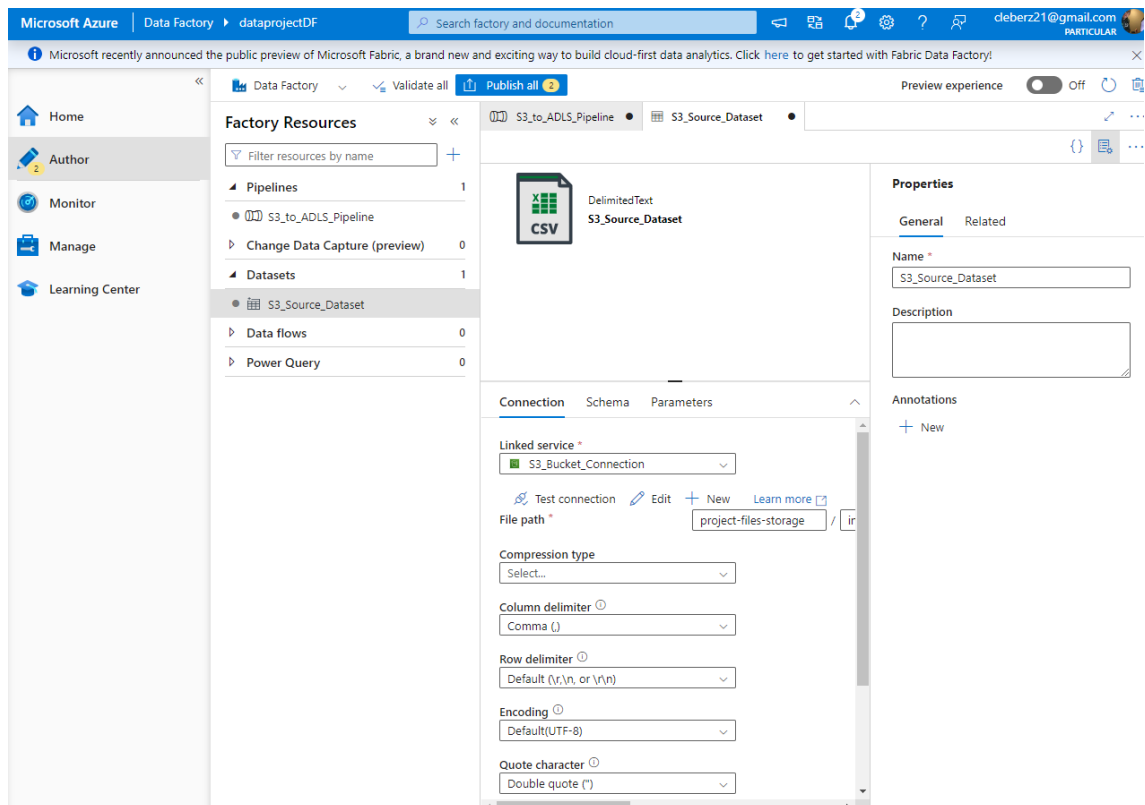
Configuração de um Linked Service no Azure Data Factory para conectar ao bucket S3.



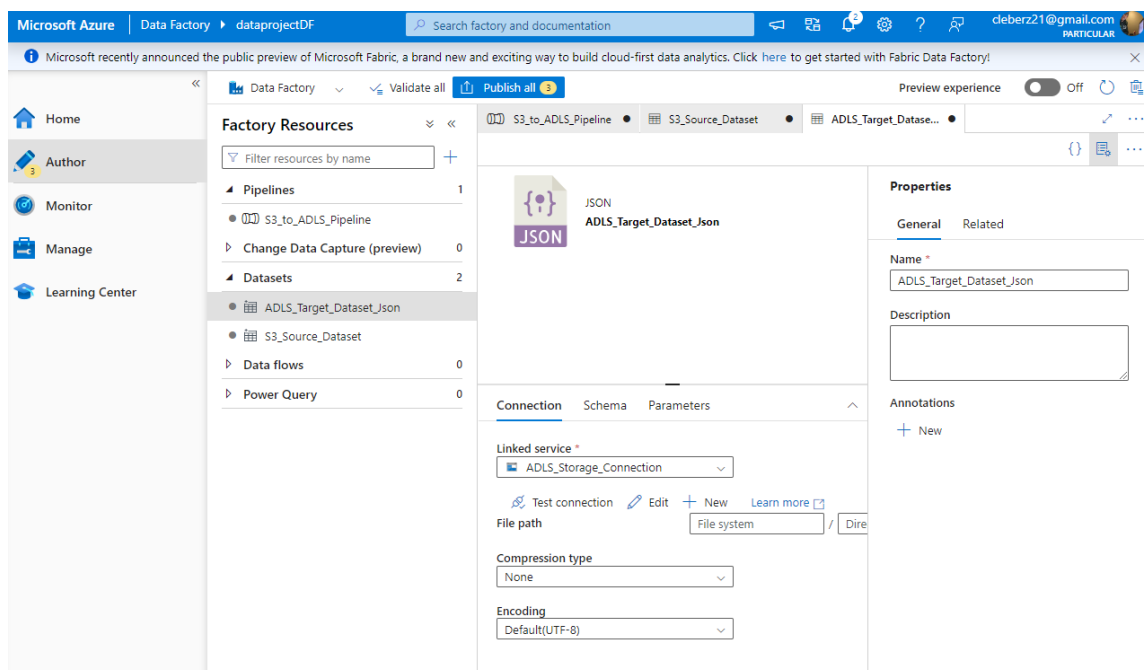
Configuração de um Linked Service no Azure Data Factory para conectar ao ADLS.



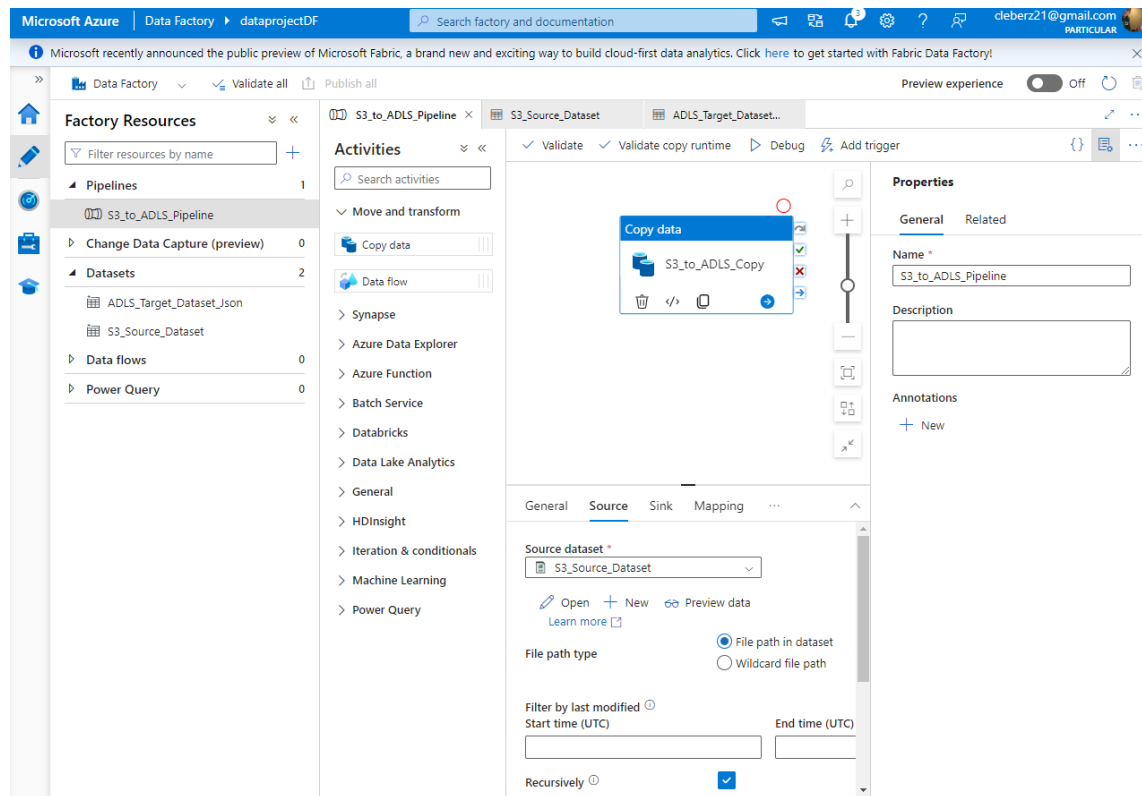
Linked Service criado



Criação de um Dataset que representa a origem dos dados.



Criação de um Dataset que representa o destino dos dados.



Desenvolvimento de um Pipeline que copia os dados do S3 para o ADLS.

6 Execução do Pipeline de Dados

O pipeline de dados foi configurado para executar a transferência de dados de forma eficiente e segura, garantindo que o arquivo `sf-fire-calls.csv` seja disponibilizado no Data Lake do ADLS, no formato Json, para processamento subsequente no 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 | Search resources, services, and docs (G+)

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

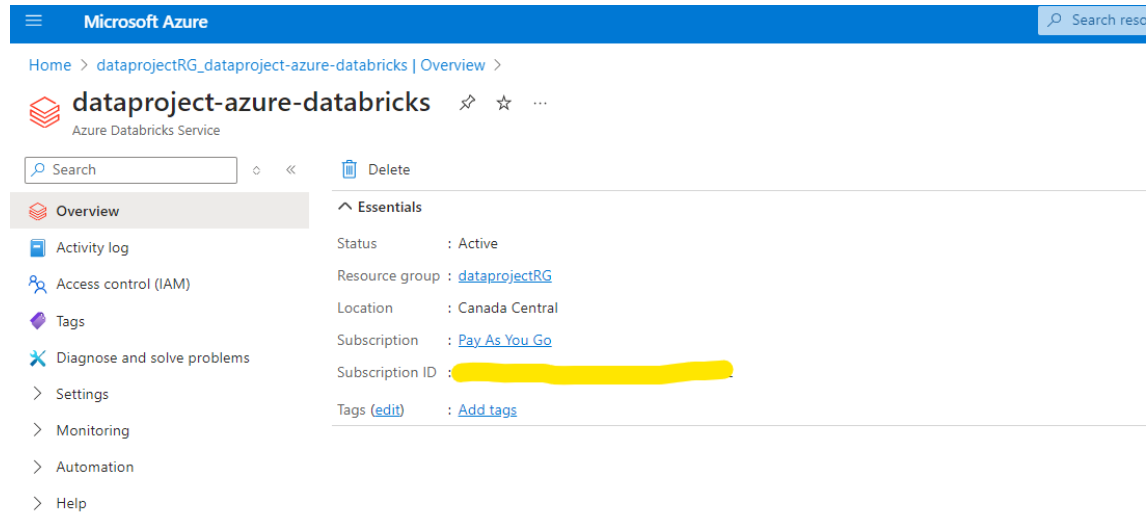
Search blobs by prefix (case-sensitive) Show deleted objects

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

Transferência dos dados concluída

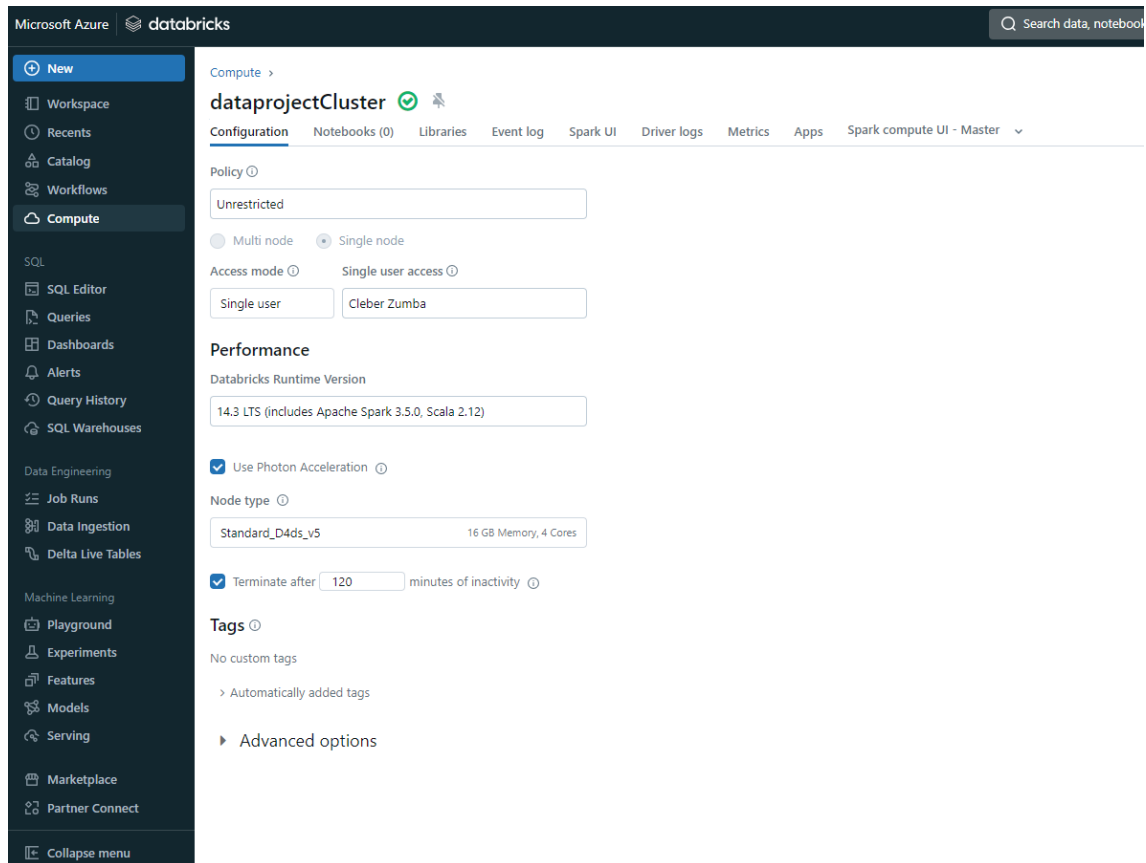
7 Criação do Azure Databricks Workspace e Cluster

O Azure Databricks foi provisionado para realizar a transformação e análise dos dados. Isso inclui a criação de um workspace, um cluster e um 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. The breadcrumb trail indicates the path: 'Home > dataprojectRG_dataproject-azure-databricks | Overview >'. The service name 'dataproject-azure-databricks' is prominently displayed, along with the 'Azure Databricks Service' label. A search bar and a 'Delete' button are visible above the main content area. On the left, a sidebar menu lists various management 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 provides key information about the service: Status is 'Active', Resource group is 'dataprojectRG', Location is 'Canada Central', Subscription is 'Pay As You Go', and Subscription ID is redacted with a yellow highlight. A link to 'Add tags' is also present.

Essentials	
Status	: Active
Resource group	: dataprojectRG
Location	: Canada Central
Subscription	: Pay As You Go
Subscription ID	: [Redacted]
Tags (edit)	: Add tags



Criação de um Cluster no Databricks

Microsoft 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

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 ⓘ

dataproject-datbricks-scope

Manage Principal ⓘ

All workspace users

Azure Key Vault ⓘ

DNS Name

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

Resource ID

/subscriptions/[redacted]/resourceGroups/dataproje

Criação de um Secret Scope no 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 Camada de Extração de Dados

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 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.

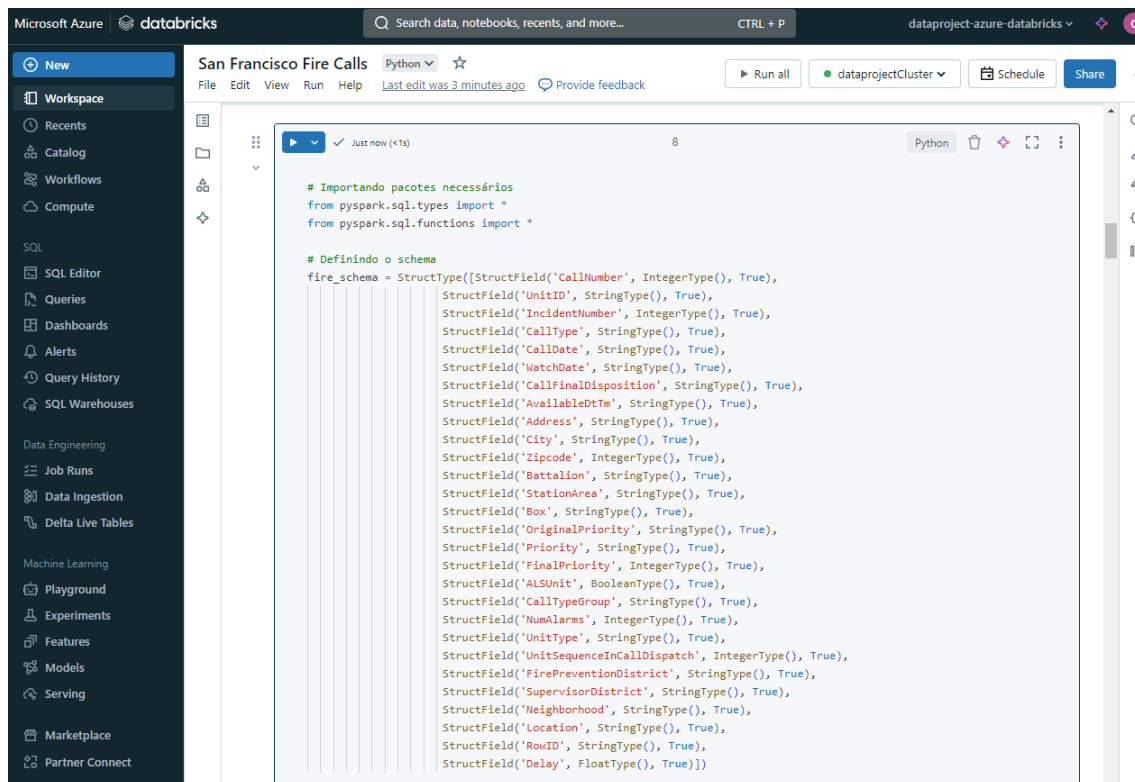
ETL Process

Source Systems → Extract → Transform → Load → Destination

- 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.

23

Engenharia de Dados com 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

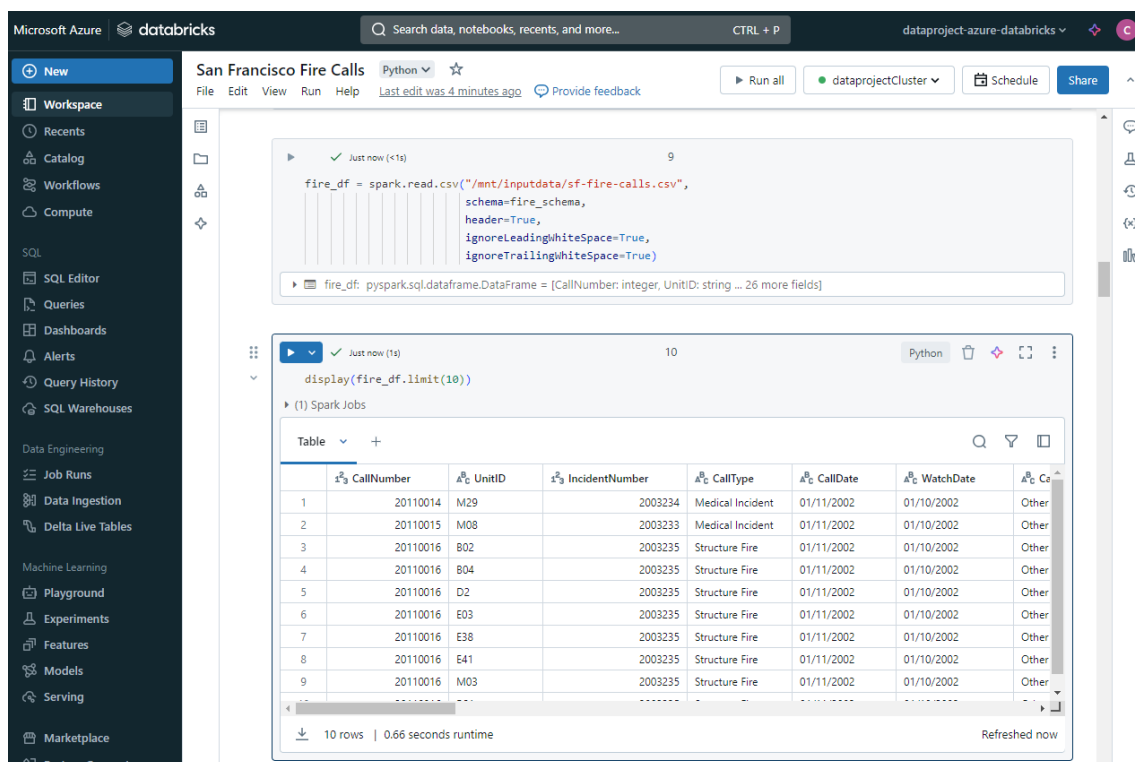
San Francisco Fire Calls Python ☆

File Edit View Run Help 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 ☆

File Edit View Run Help 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]

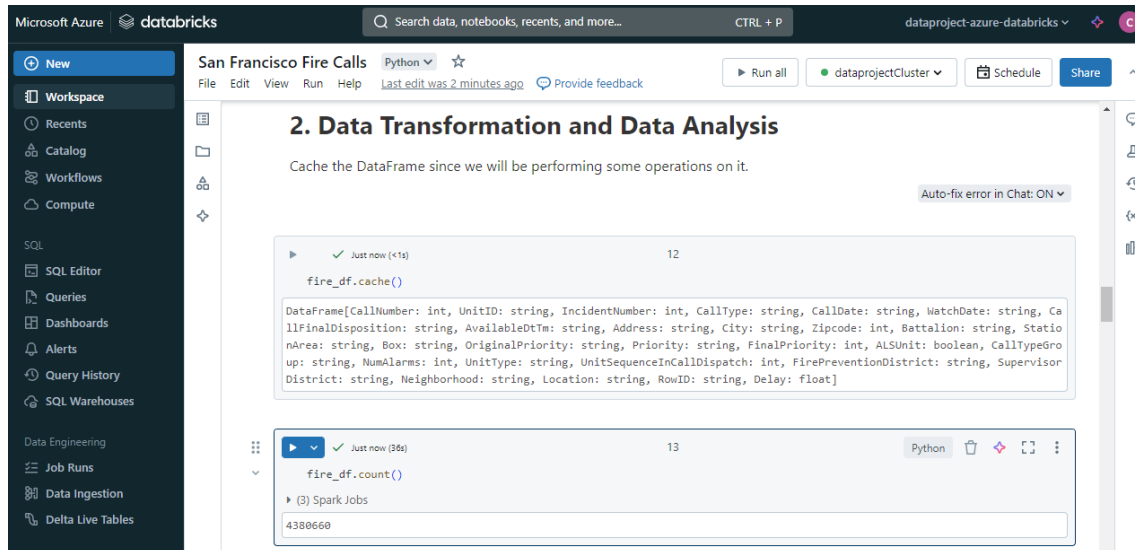
```
display(fire_df.limit(10))
```

(1) Spark Jobs

	CallNumber	UnitID	IncidentNumber	CallType	CallDate	WatchDate	CallFinalDisposition
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 Camada de Transformação e Análise de Dados



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.

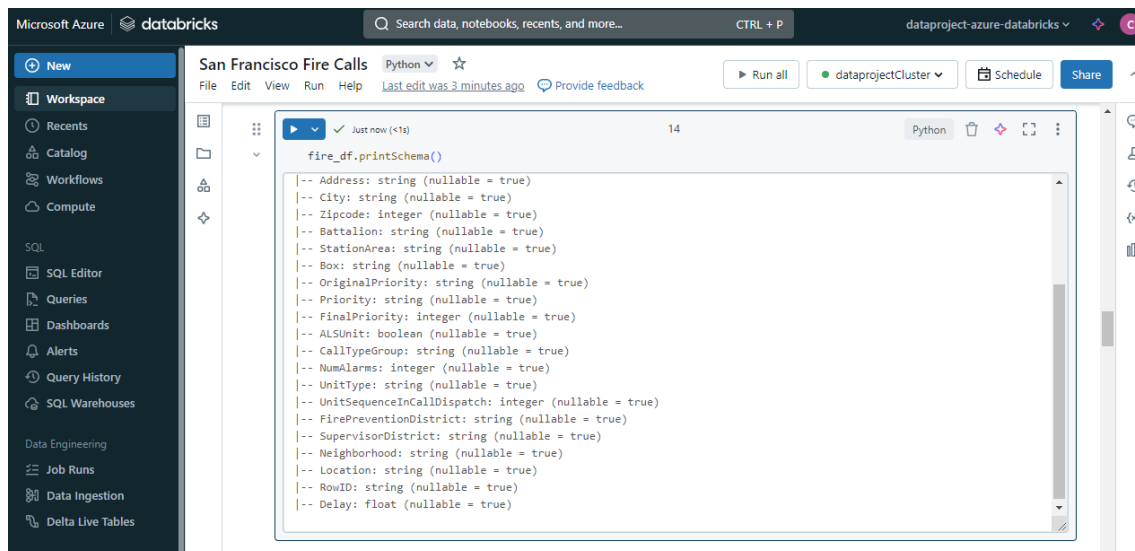
```
fire_df.cache()
```

DataFrame[CallNumber: int, UnitID: string, IncidentNumber: int, CallType: string, CallDate: string, WatchDate: string, CallFinalDisposition: string, AvailableDtTm: 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]

```
fire_df.count()
```

(3) Spark Jobs

4380660

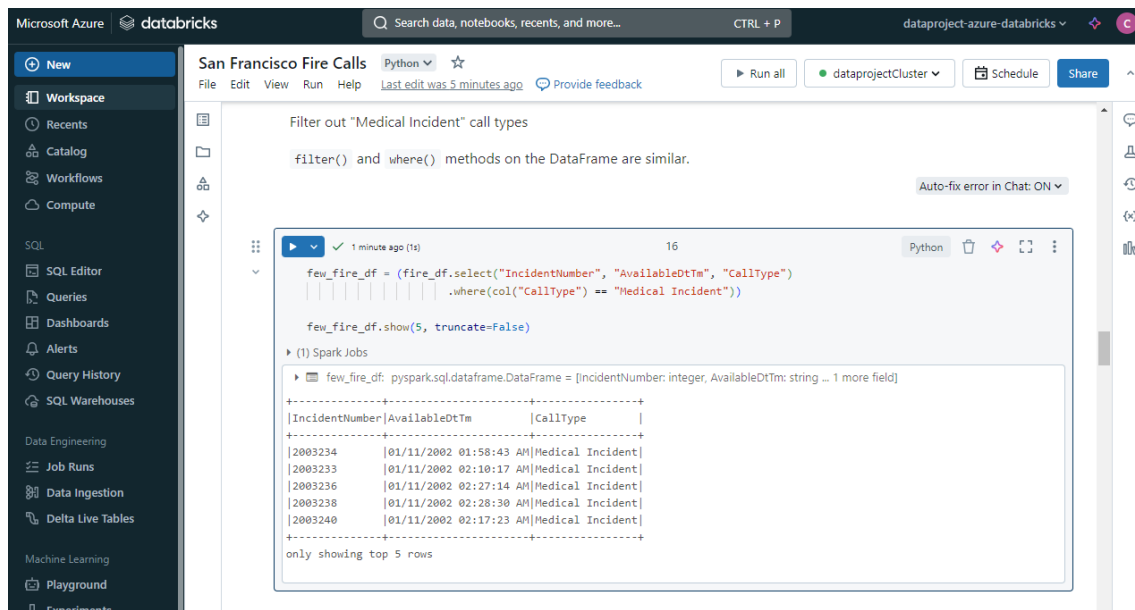


Microsoft Azure databricks

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)
```



Microsoft Azure | databricks

San Francisco Fire Calls Python

Filter out "Medical Incident" call types

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

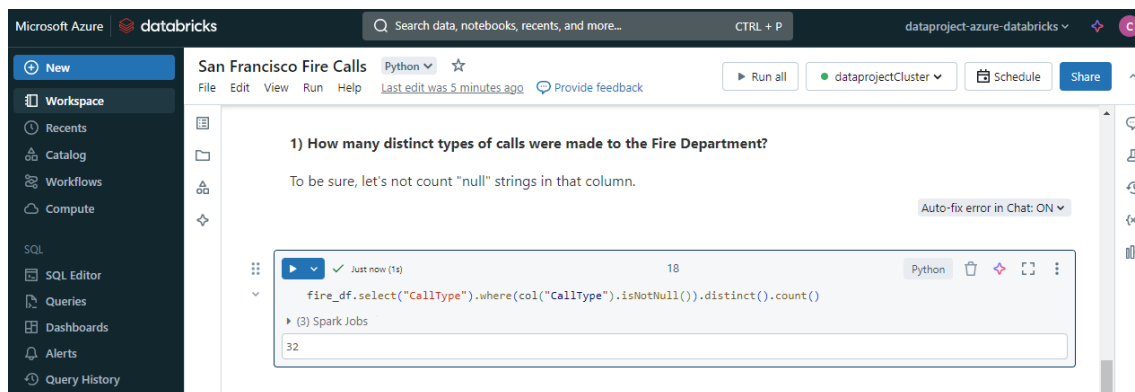
```
few_fire_df = (fire_df.select("IncidentNumber", "AvailableDtTm", "CallType")
                  .where(col("CallType") != "Medical Incident"))

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

(1) Spark Jobs

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

San Francisco Fire Calls Python

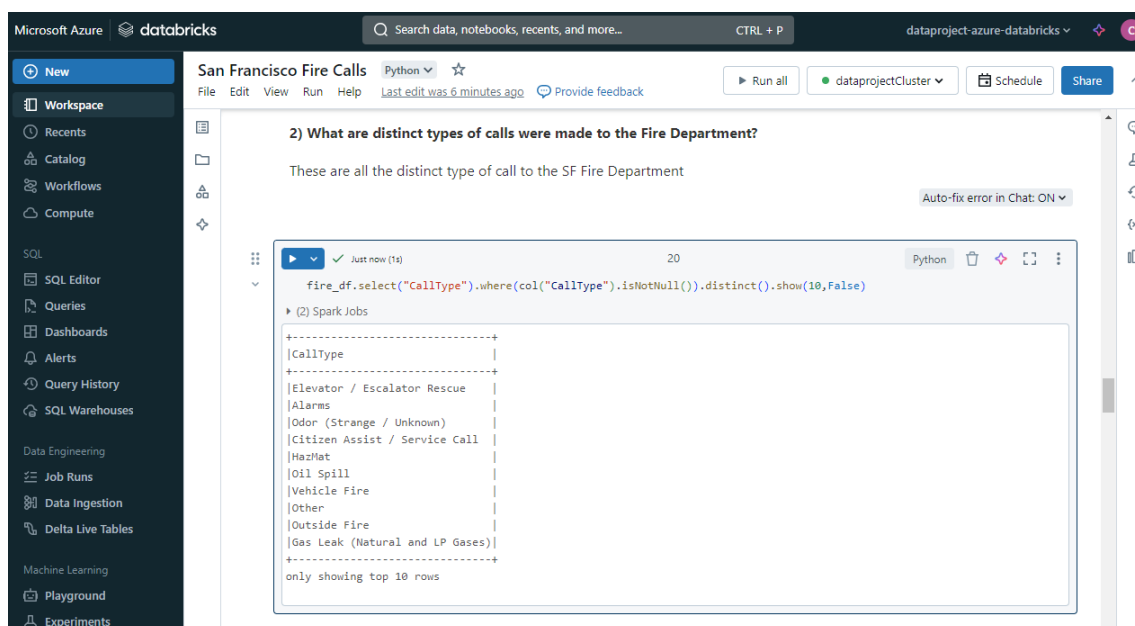
1) How many distinct types of calls were made to the Fire Department?

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

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

(3) Spark Jobs

32



Microsoft Azure | databricks

San Francisco Fire Calls Python

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

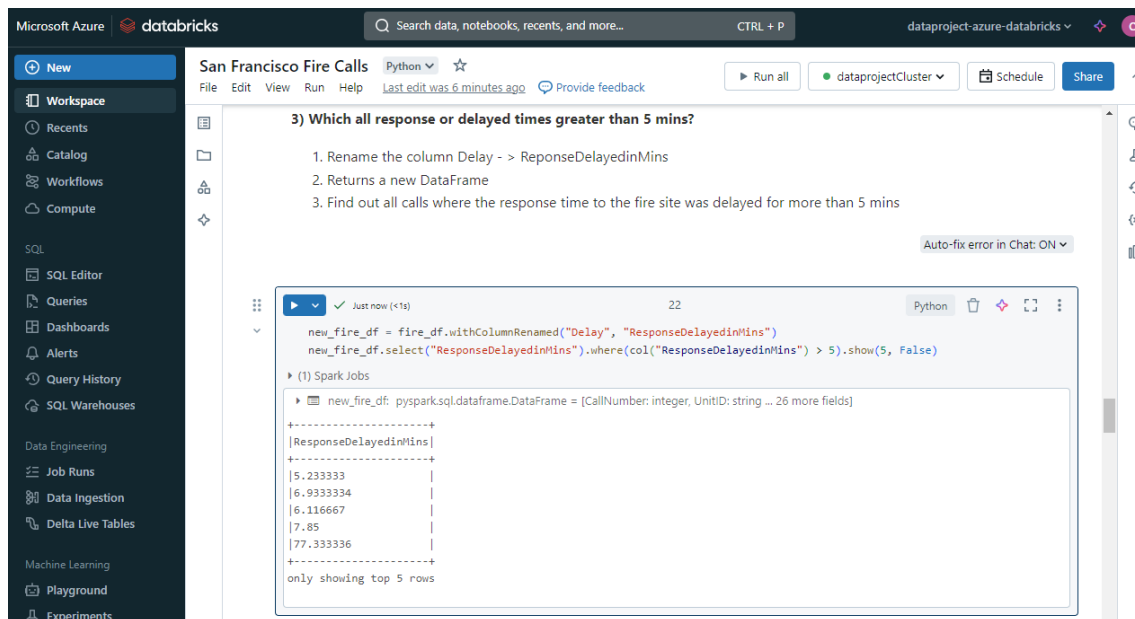
These are all the distinct type of call to the SF Fire Department

```
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



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

San Francisco Fire Calls Python Last edit was 6 minutes ago Provide feedback Run all dataprojectCluster Schedule Share

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

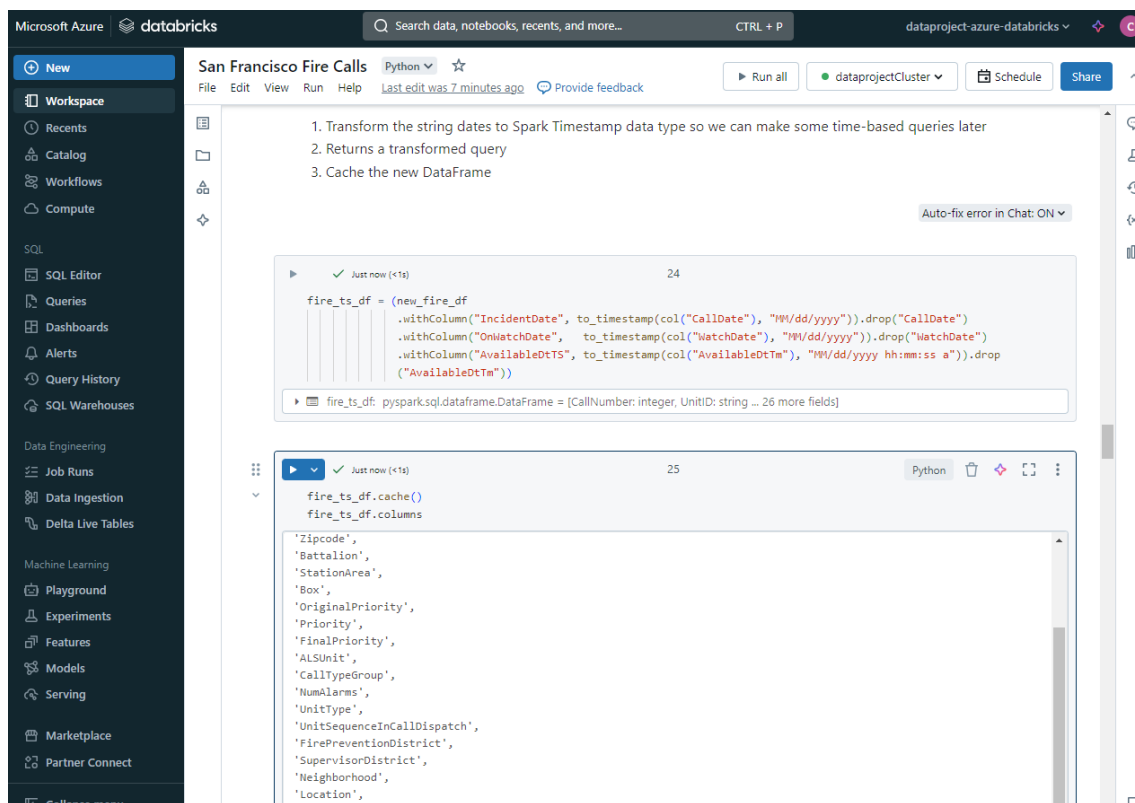
1. Rename the column Delay -> ReponseDelayedinMins
2. Returns a new DataFrame
3. Find out all calls where the response time to the fire site was delayed for more than 5 mins

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 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 Collapse menu

San Francisco Fire Calls Python Last edit was 7 minutes ago Provide feedback Run all dataprojectCluster 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

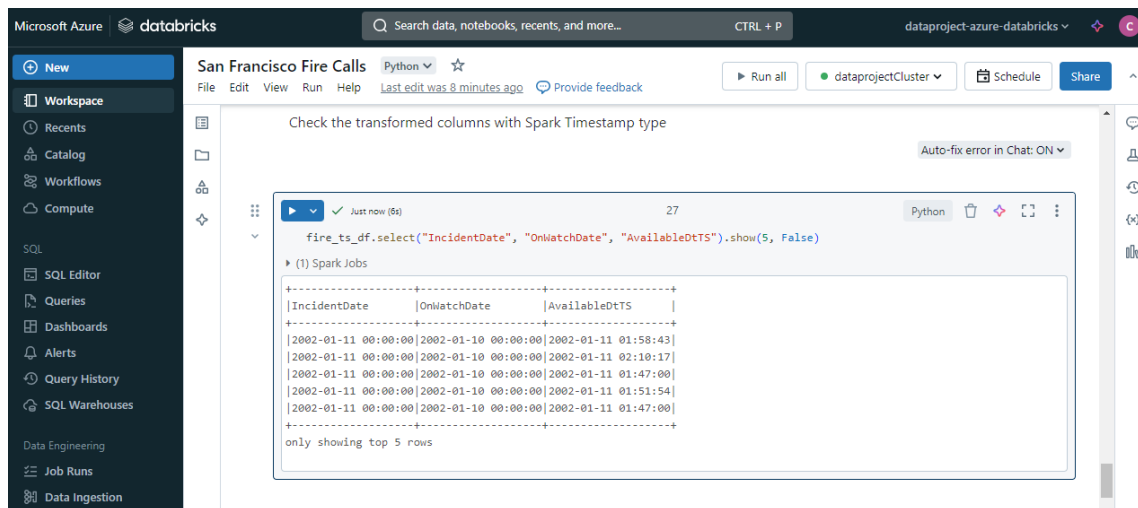
Auto-fix error in Chat: ON

```
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("AvailableDTIS", 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

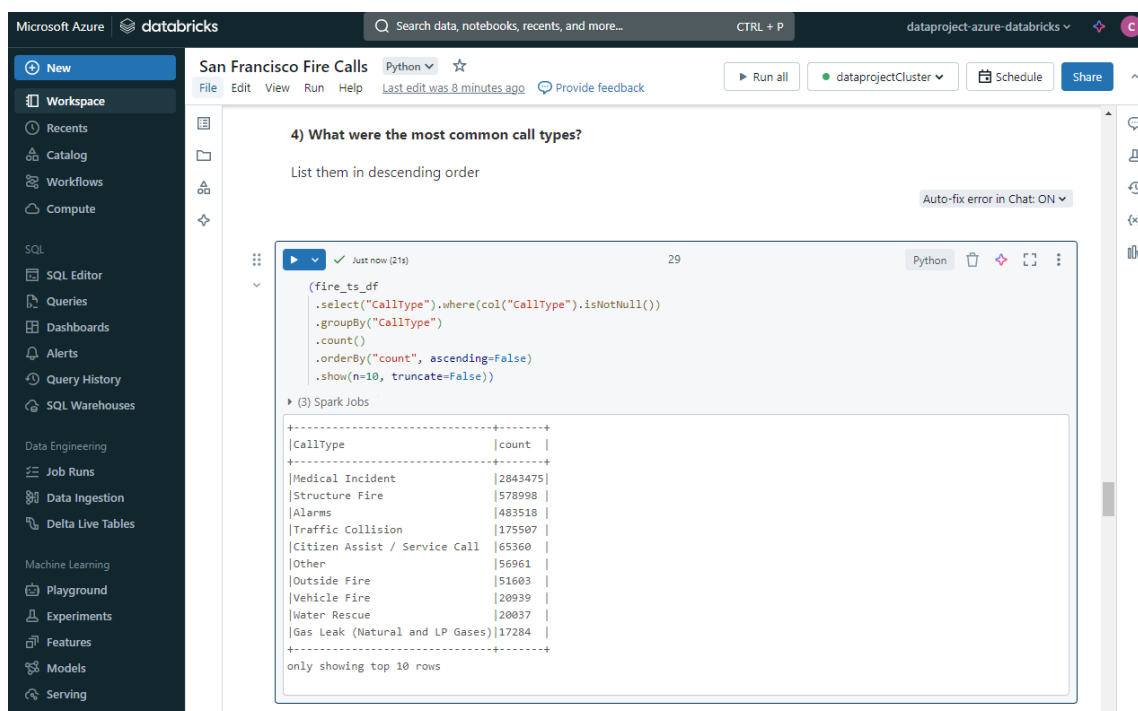
Check the transformed columns with Spark Timestamp type

```
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

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 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 | Models | Serving | Marketplace | Partner Connect

San Francisco Fire Calls

Python | 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.

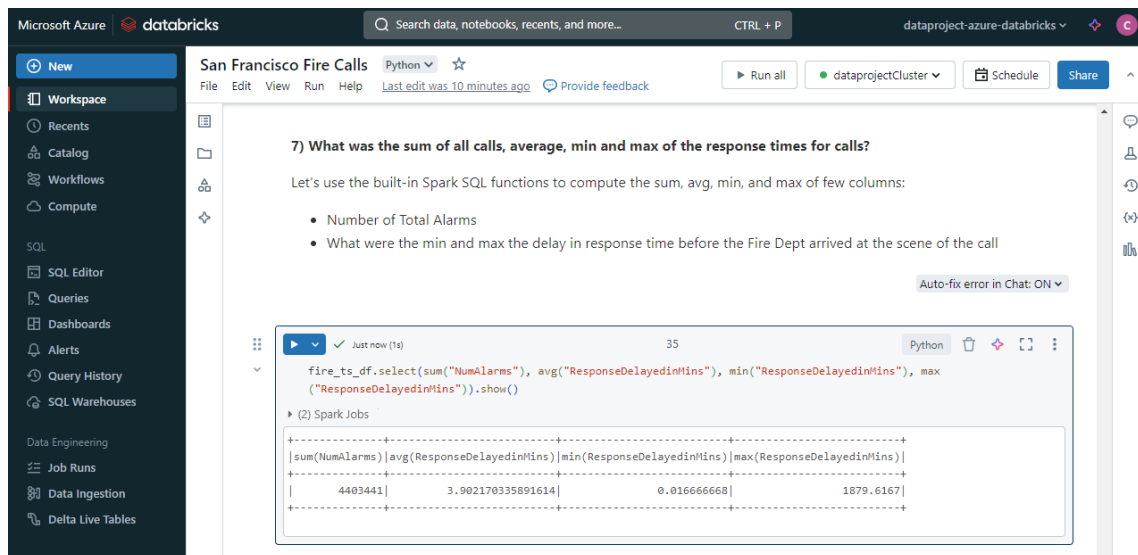
Auto-fix error in Chat: ON

```
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
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

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

San Francisco Fire Calls

Python

File Edit View Run Help

Last edit was 10 minutes ago

Provide feedback

Run all

dataportCluster

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

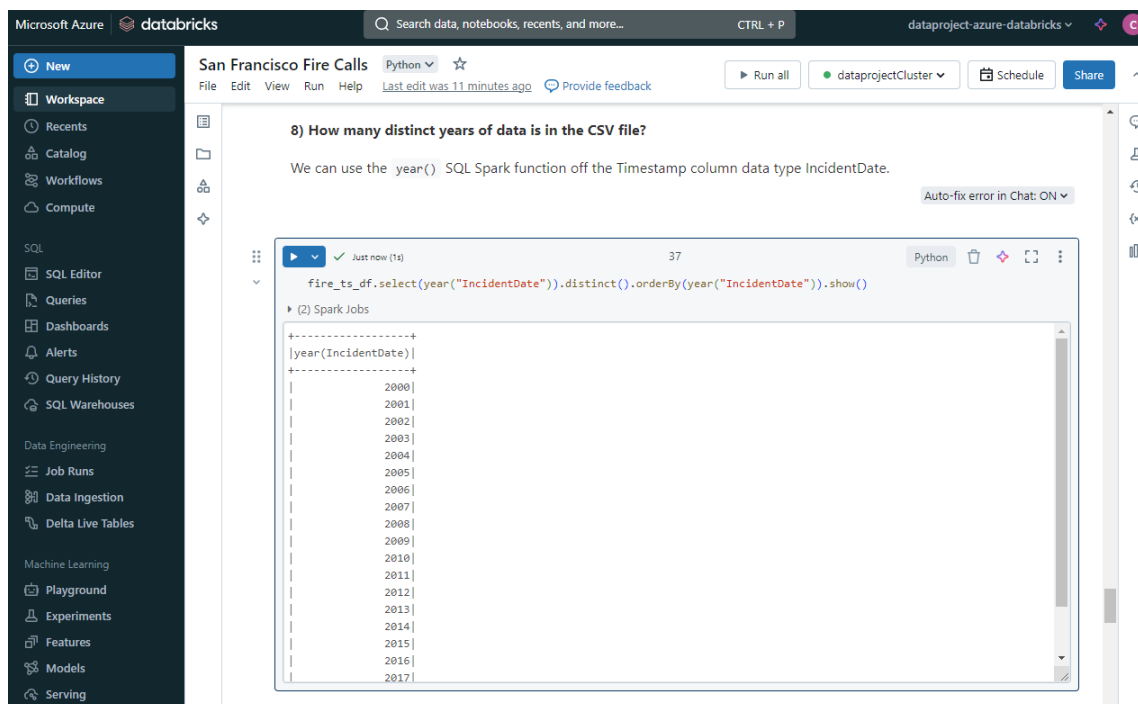
Just now (1s) 35

Python

```
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

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

San Francisco Fire Calls

Python

File Edit View Run Help

Last edit was 11 minutes ago

Provide feedback

Run all

dataportCluster

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

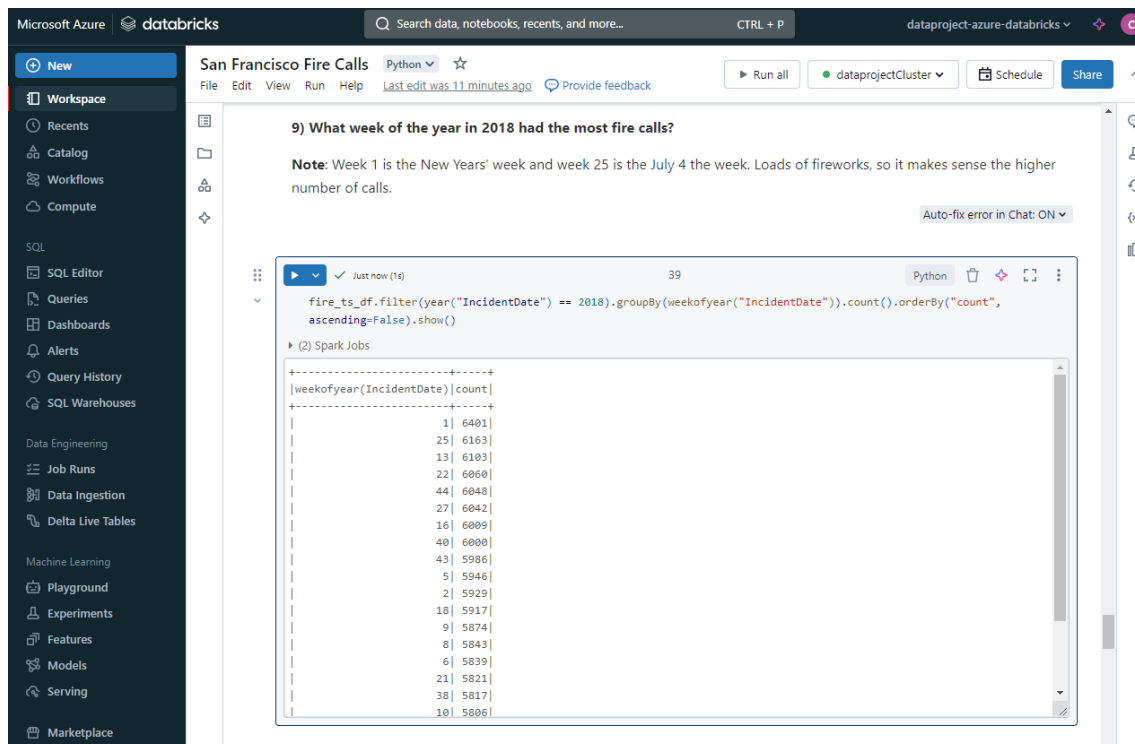
Just now (1s) 37

Python

```
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, recent, 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

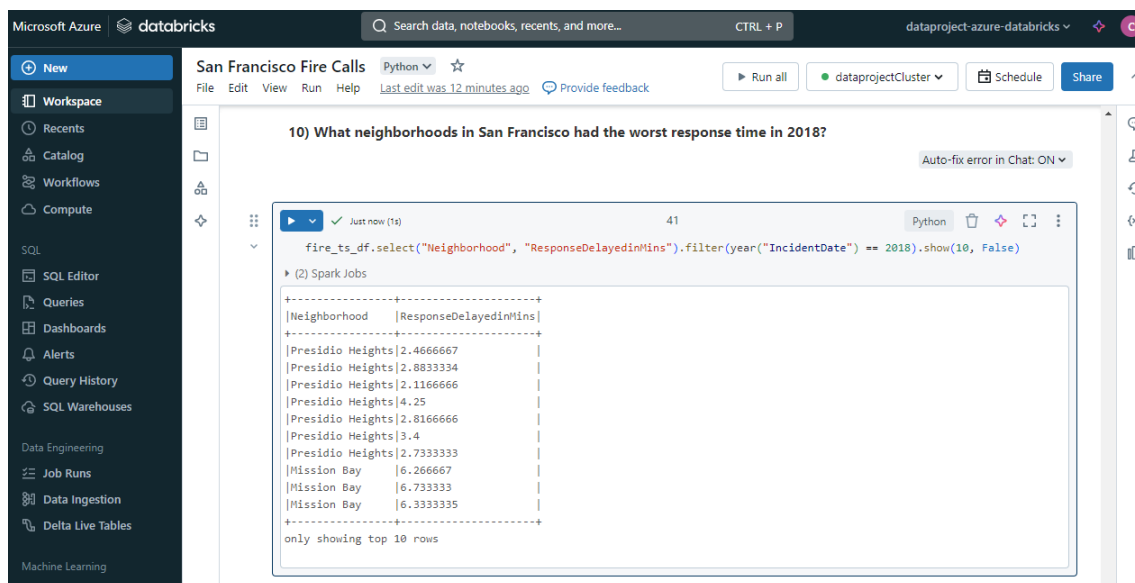
9) What week of the year in 2018 had the most fire calls?

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, recent, 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 Camada de Armazenamento de Dados

San Francisco Fire Calls Python

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	

Salvando os dados em formato parquet no sistema de arquivos do Databricks.

Microsoft Azure

Home > dataprojectst | Containers >

inputdataset Container

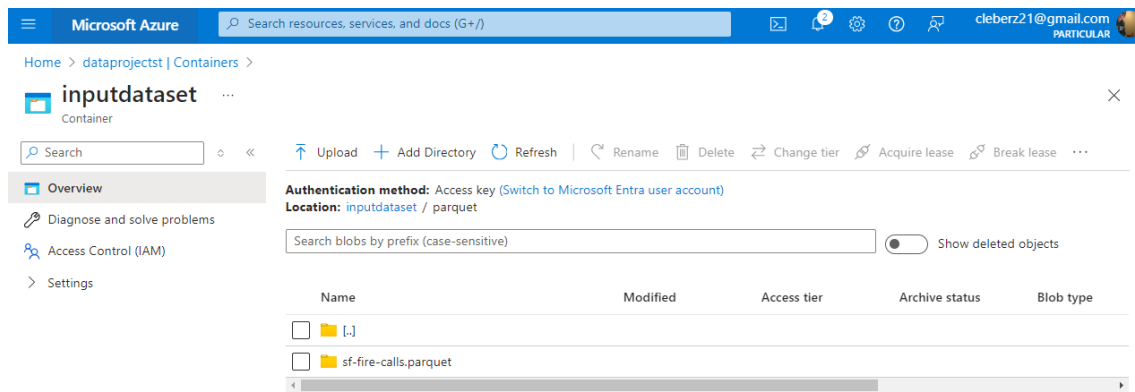
Authentication method: Access key (Switch to Microsoft Entra user account)

Location: inputdataset

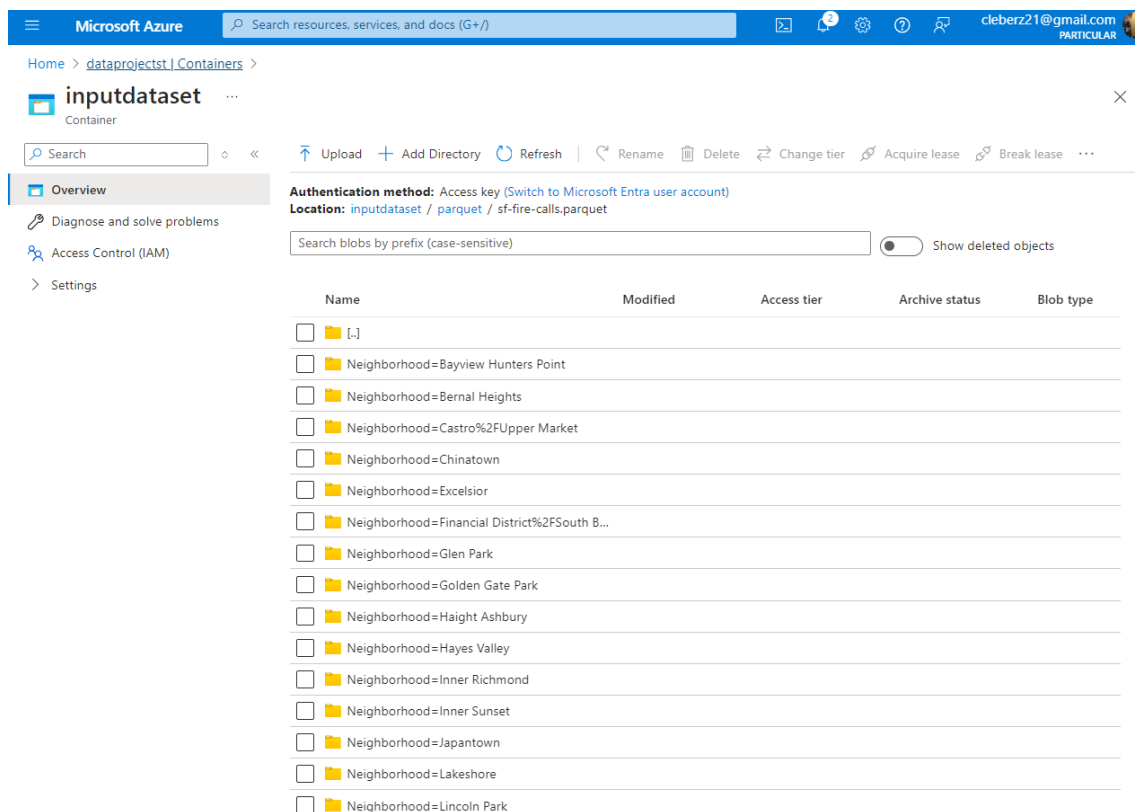
Search blobs by prefix (case-sensitive)

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

Dados salvos em formato parquet no container do Azure Data Lake Storage (ADLS).



Verificando os dados salvos no container



Dados particionados no container

Engenharia de Dados com 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

Dados particionados no 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

1 minute ago (1m) 46

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

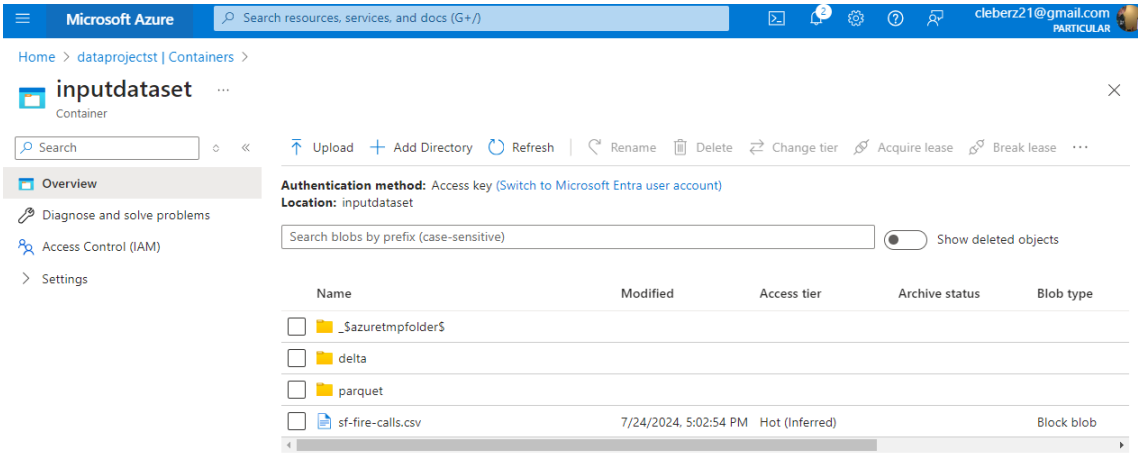
(2) Spark Jobs

Just now (1s) 47 Python

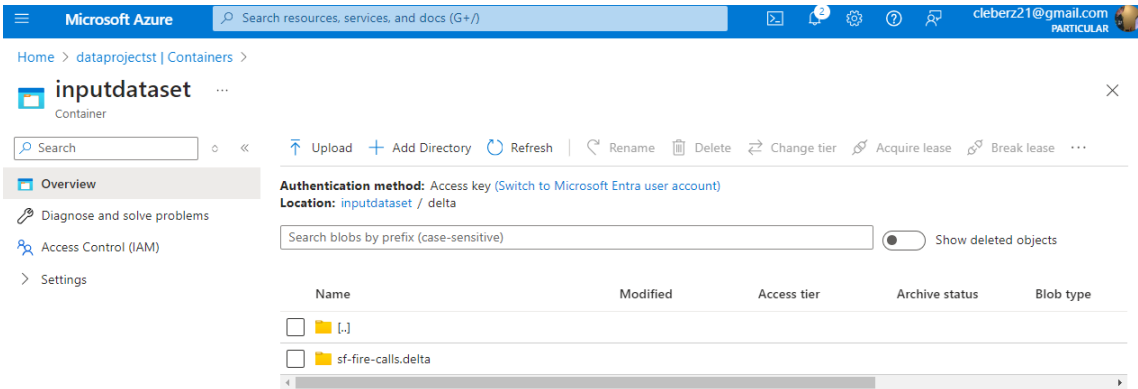
%fs ls /mnt/inputdataset/delta/sf-fire-calls.delta

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/

Salvando os dados em formato delta no sistema de arquivos do Databricks.



Dados salvos em formato delta no container do Azure Data Lake Storage (ADLS).



Verificando os dados salvos no 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+/)

Home > dataprojectstj Containers >

inputdataset

Container

Search

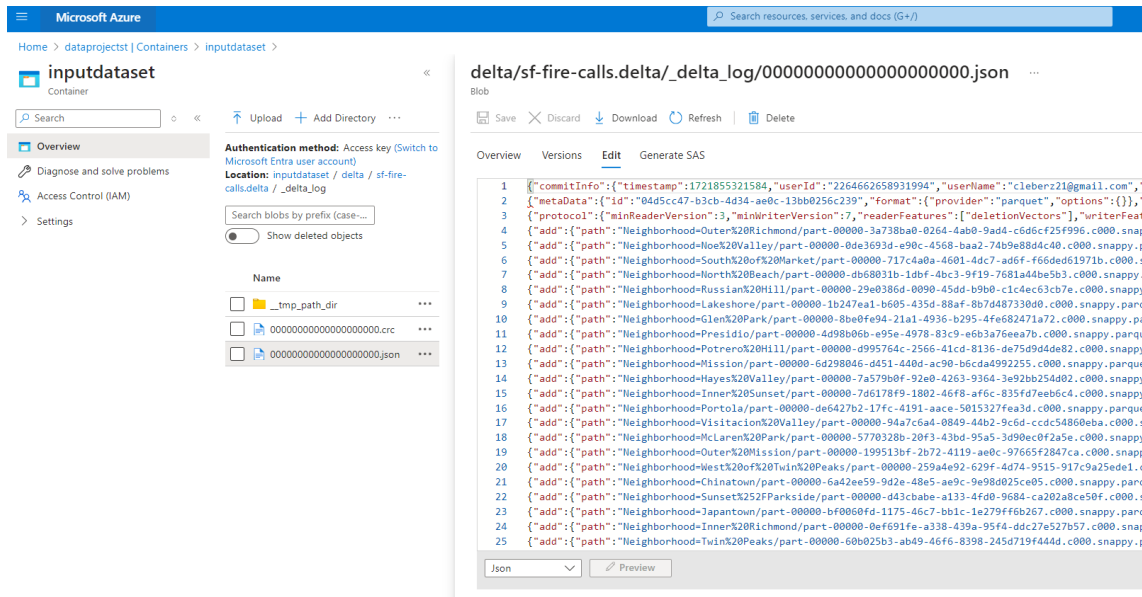
Upload Add Directory Refresh Rename Delete Change tier Acquire lease Break lease ...

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

36



Visualizando os dados no container.

11 Camada de Consulta de Dados

Microsoft Azure databricks

San Francisco Fire Calls Python

13) How can we use Delta SQL table to store data and read it back?

```
fire_ts_df.write.format("delta").mode("overwrite").saveAsTable("FireServiceCalls")
```

(4) Spark Jobs

```
%sql
select * from FireServiceCalls limit 10
```

(1) Spark Jobs

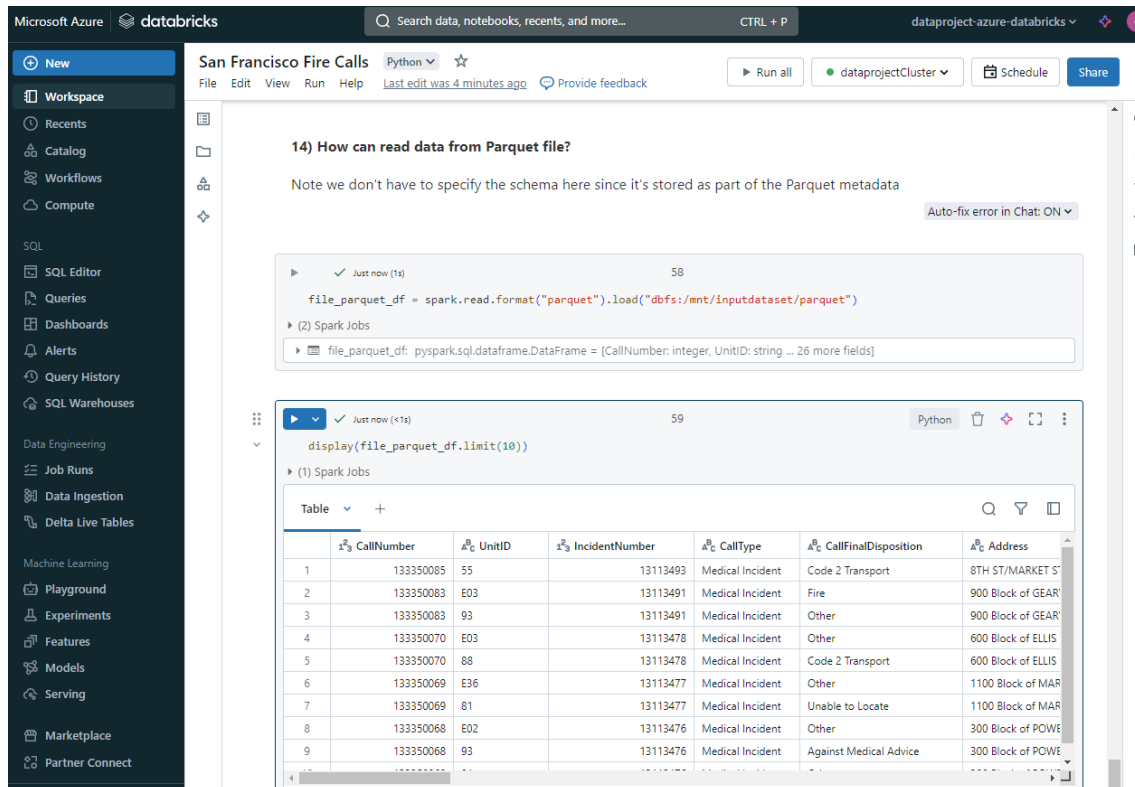
_sqldf: pyspark.sql.dataframe.DataFrame = [CallNumber: integer, UnitID: string ... 26 more fields]

	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

10 rows | 1.32 seconds runtime

This result is stored as _sqldf and can be used in other Python cells.

Consultando o resultado da transformação e da análise de dados no formato delta.



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

Consultando o resultado da transformação e da análise de dados no formato parquet.

12 Conclusão

Neste projeto, explorei os recursos poderosos das tecnologias AWS, Azure e Databricks para criar um pipeline de dados robusto e escalável, focado na análise e transformação de dados de chamadas de emergência do Corpo de Bombeiros de São Francisco.

Realizei processos de extração, transformação e carregamento de dados (ETL), destacando a integração perfeita entre AWS S3, Azure Data Lake Storage (ADLS) e Databricks para processamento de dados escalável. Utilizamos o Azure Data Factory para orquestrar a movimentação de dados e garantir a transferência segura entre os ambientes de nuvem.

No Databricks, utilizei PySpark e SparkSQL para executar consultas e transformações, demonstrando sua capacidade de lidar com análises de dados em larga escala com facilidade. Implementei técnicas de otimização de desempenho, como cache, que são essenciais para atingir o processamento de dados de alto desempenho.

Segui as práticas recomendadas utilizando os recursos avançados do Azure Key Vault para gestão segura de segredos e do Databricks para processamento distribuído.

Cleber Zumba de Souza

13 Referência

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