Scaling The Prototype

Capstone Step 6

Process Updating ETL to Spark

This process was quite more involved than it had to be. The main steps of the approach included:

1. Attempting and ultimately failing to debug **Azure Toolkit aztk**.

    This repository, prescribed by the project instructions, has become obsolete.

    The technical details of the issues are documented in **development_notes.txt**

2. Opted to use **Azure Databricks** instead of aztk.

3. Created **Azure Blob Storage Container**

    Uploaded data to blob container

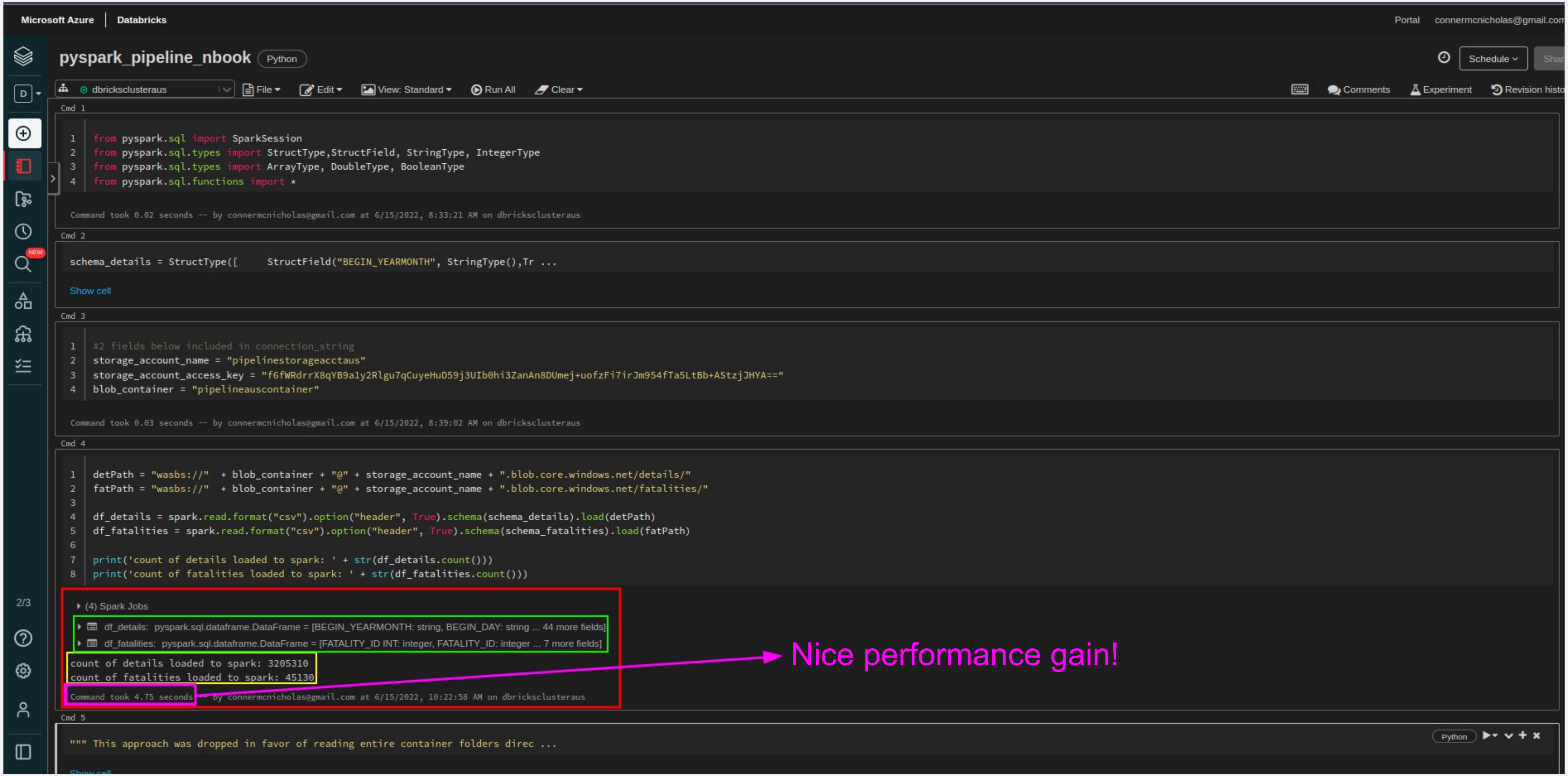
    Configured entitlements of new **Databricks Cluster** to read from blob container.

4. Created notebook **pyspark_pipeline_nbook.ipynb** in Databricks with **Pyspark kernel** to load data from Azure blob storage.

    Translated code from Step 5 from Python to Pyspark.

    Observed noticeable speed improvement from locally deployed python script.

Databricks Spark Notebook



Cmd 1

```
from pyspark.sql import SparkSession
from pyspark.sql.types import StructType,StructField, StringType, IntegerType
from pyspark.sql.types import ArrayType, DoubleType, BooleanType
from pyspark.sql.functions import *
```

```
Command took 0.02 seconds -- by connermcnicholas@gmail.com at 6/15/2022, 8:33:21 AM on dbricksclusteraus
```

Cmd 2

```
schema_details = StructType([     StructField("BEGIN_YEARMONTH", StringType(),Tr ...
```

Show cell

Cmd 3

```
#2 fields below included in connection_string
storage_account_name = "pipelinestorageacctaus"
storage_account_access_key = "f6fWRdrrX8qYB9aly2Rlgu7qCuyeHuD59j3UIb0hi3ZanAn8DUmej+uofzFi7irJm954fTa5LtBb+AStzjJHYA=="
blob_container = "pipelineauscontainer"
```

```
Command took 0.03 seconds -- by connermcnicholas@gmail.com at 6/15/2022, 8:39:02 AM on dbricksclusteraus
```

Cmd 4

```
detPath = "wasbs://"  + blob_container + "@" + storage_account_name + ".blob.core.windows.net/details/"
fatPath = "wasbs://"  + blob_container + "@" + storage_account_name + ".blob.core.windows.net/fatalities/"

df_details = spark.read.format("csv").option("header", True).schema(schema_details).load(detPath)
df_fatalities = spark.read.format("csv").option("header", True).schema(schema_fatalities).load(fatPath)

print('count of details loaded to spark: ' + str(df_details.count()))
print('count of fatalities loaded to spark: ' + str(df_fatalities.count()))
```

```
(4) Spark Jobs
df_details:  pyspark.sql.dataframe.DataFrame = [BEGIN_YEARMONTH: string, BEGIN_DAY: string ... 44 more fields]
df_fatalities:  pyspark.sql.dataframe.DataFrame = [FATALITY_ID INT: integer, FATALITY_ID: integer ... 7 more fields]
count of details loaded to spark: 3205310
count of fatalities loaded to spark: 45130
Command took 4.75 seconds -- by connermcnicholas@gmail.com at 6/15/2022, 10:22:58 AM on dbricksclusteraus
```

Nice performance gain!

Cmd 5

```
""" This approach was dropped in favor of reading entire container folders direc ...
```

Show cell