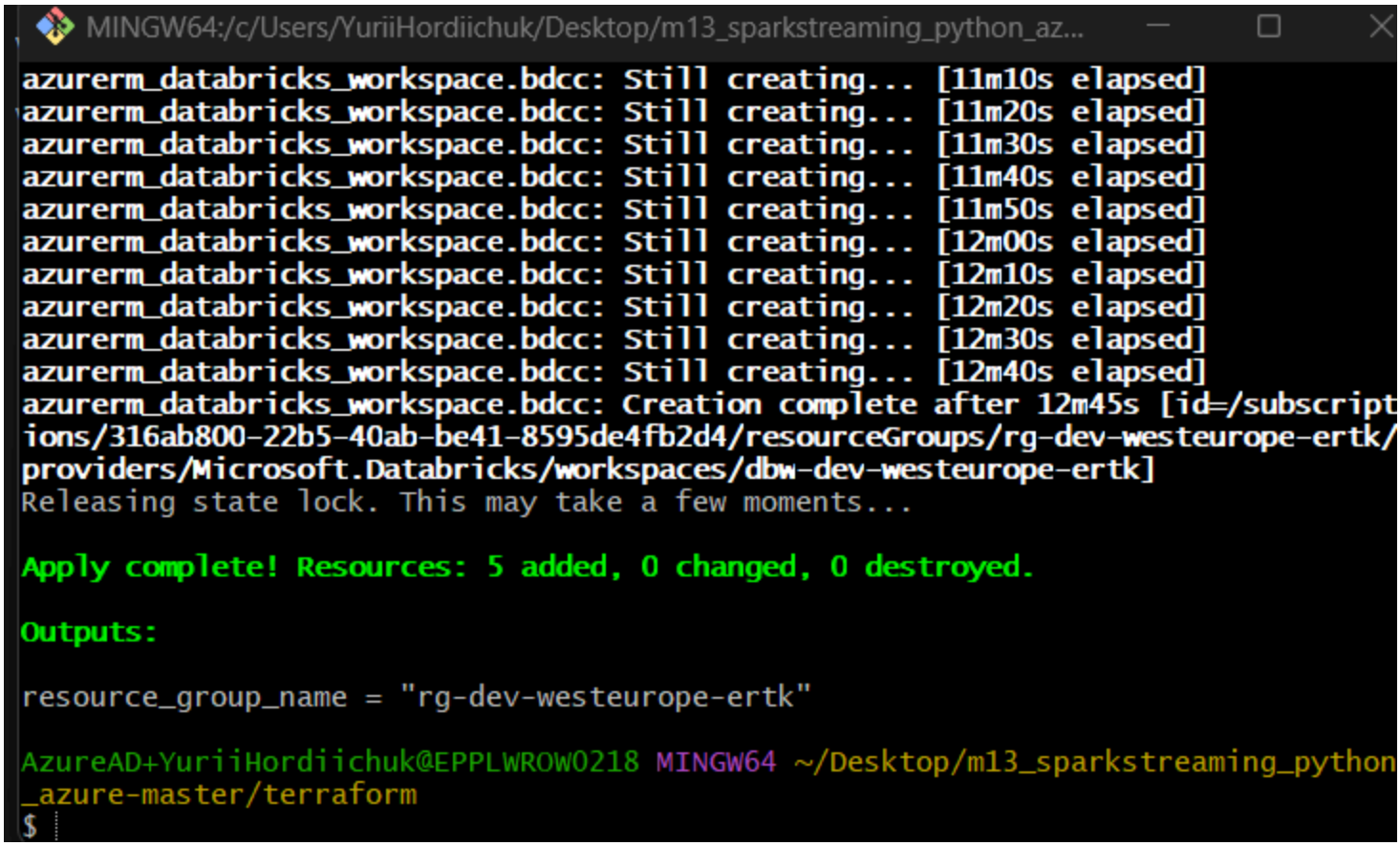
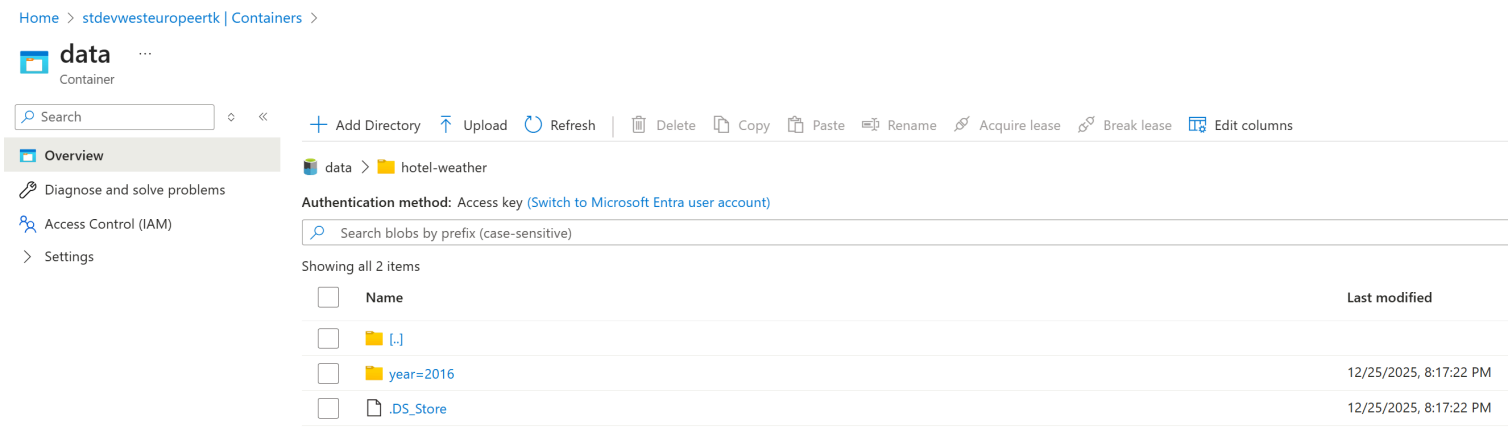
SPARK STREAMING

terraform apply:

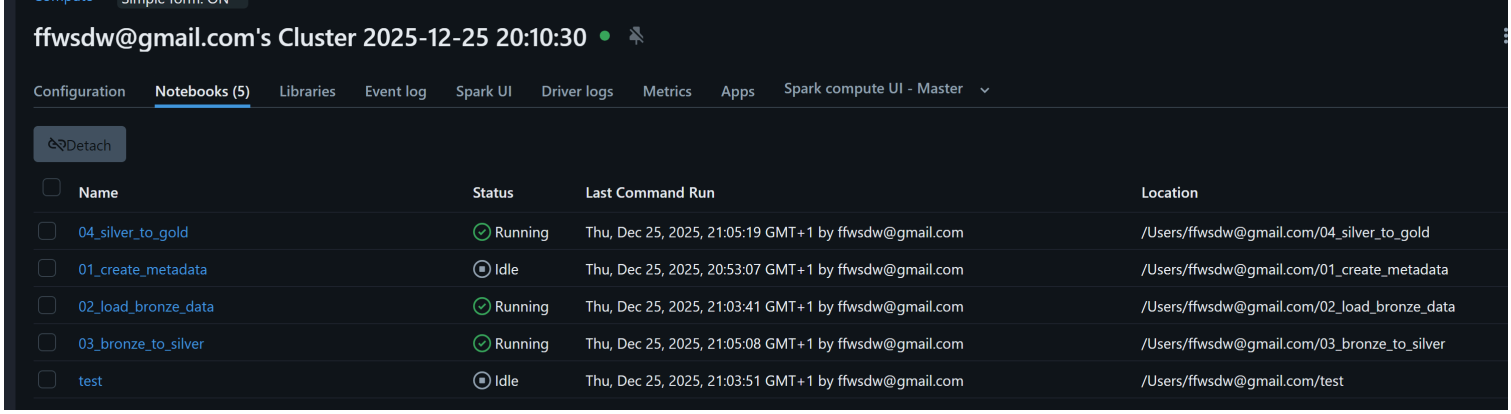


1.

I loaded only 2016 year:



2.I ran all the notebooks:



3.Results for 2016 year:

Preview gold table:

Just now (4s) 2

```
%sql
select * from gold.hotel_weather_metrics
```

(3) Spark Jobs

_sqldf: pyspark.sql.dataframe.DataFrame = [country: string, city: string ... 6 more fields]

Table

	country	city	wthr_date	num_distinct_hotels	avg_temp_c	max_temp_c	min_temp_c	1.
1	FR	Paris	2016-10-31	237	10.699999999999994	10.7	10.7	
2	US	Maumee	2016-10-20	1	12.8	12.8	12.8	
3	US	Milford	2016-10-10	1	13.8	13.8	13.8	
4	US	Ashland	2016-10-25	1	6.8	6.8	6.8	
5	US	Burdett	2016-10-26	1	2.3	2.3	2.3	
6	US	Aberdeen	2016-10-21	1	8.8	8.8	8.8	
7	US	Junction	2016-10-27	1	20.9	20.9	20.9	
8	US	Lumberton	2016-10-20	1	22.4	22.4	22.4	
9	US	Millbrook	2016-10-25	1	19.2	19.2	19.2	
10	US	Millington	2016-10-10	1	18.3	18.3	18.3	
11	US	Sweetwater	2016-10-20	1	22.7	22.7	22.7	
12	US	Washington	2016-10-22	1	7.4	7.4	7.4	
13	US	Palm Harbor	2016-10-04	2	25.9	25.9	25.9	
14	US	Grand Prairie	2016-10-13	1	18.1	18.1	18.1	
15								

4. Loading 2017 year:

Interrupt 5 Python

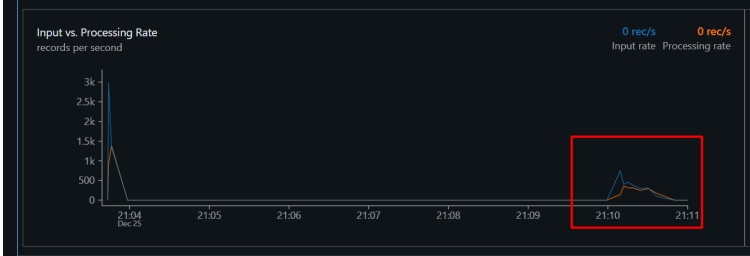
```
load_encrypt_stream_write(
    source_path=hotel_weather_source_path,
    schema=hotel_weather_schema,
    pii_columns=hotel_weather_pii_columns,
    encryptor=encryptor,
    fmt="parquet",
    checkpoint_path="/checkpoints/bronze/hotel_weather_raw",
    target_table="bronze.hotel_weather_raw"
)

print("Streaming Bronze ingestion started")
```

(1) Spark Jobs

Job 873 View (2 stages)

29426e59-7a8b-4407-82a9-f389dcf6fe82 *Last updated: 5 seconds ago*

Dashboard Raw Data




```
# write streaming silver
(
    hotel_clean.writeStream
        .format("delta")
        .outputMode("append")
        .option("checkpointLocation", "/mnt/checkpoints/silver_hotel_weather_n")
        .table("silver.hotel_weather_processed")
)
```

(1) Spark Jobs

57f250b4-7834-4b17-8a9e-88feb808d2e5 *Last updated: 1 minute ago*

Dashboard Raw Data

hotel_bronze: pyspark.sql.dataframe.DataFrame = [address: string, avg_tmpr_c double ... 13 more fields]

hotel_clean: pyspark.sql.dataframe.DataFrame = [address: string, avg_tmpr_c: string ... 13 more fields]

```
# Write Gold Delta table
(
    hotel_metrics.writeStream
        .format("delta")
        .outputMode("complete")
        .option("checkpointLocation", "/mnt/checkpoints/gold_hotel_weather_metrics_n")
        .table("gold.hotel_weather_metrics")
)
```

▸ (1) Spark Jobs

0f069b65-3f2b-4a34-97f4-561d3bce5c1e *Last updated: 30 seconds ago*

Dashboard Raw Data

hotel_metrics: pyspark.sql.dataframe.DataFrame = [country: string, city: string ... 6 more fields]

Result after loading 2017 year:

```
%sql
select 'count bronze.hotel_weather_raw' tbl_name, count(*) cnt from bronze.hotel_weather_raw
union
select 'count silver.hotel_weather_processed', count(*) from silver.hotel_weather_processed
union
select 'count gold.hotel_weather_metrics', count(*) from gold.hotel_weather_metrics
```

▸ (15) Spark Jobs

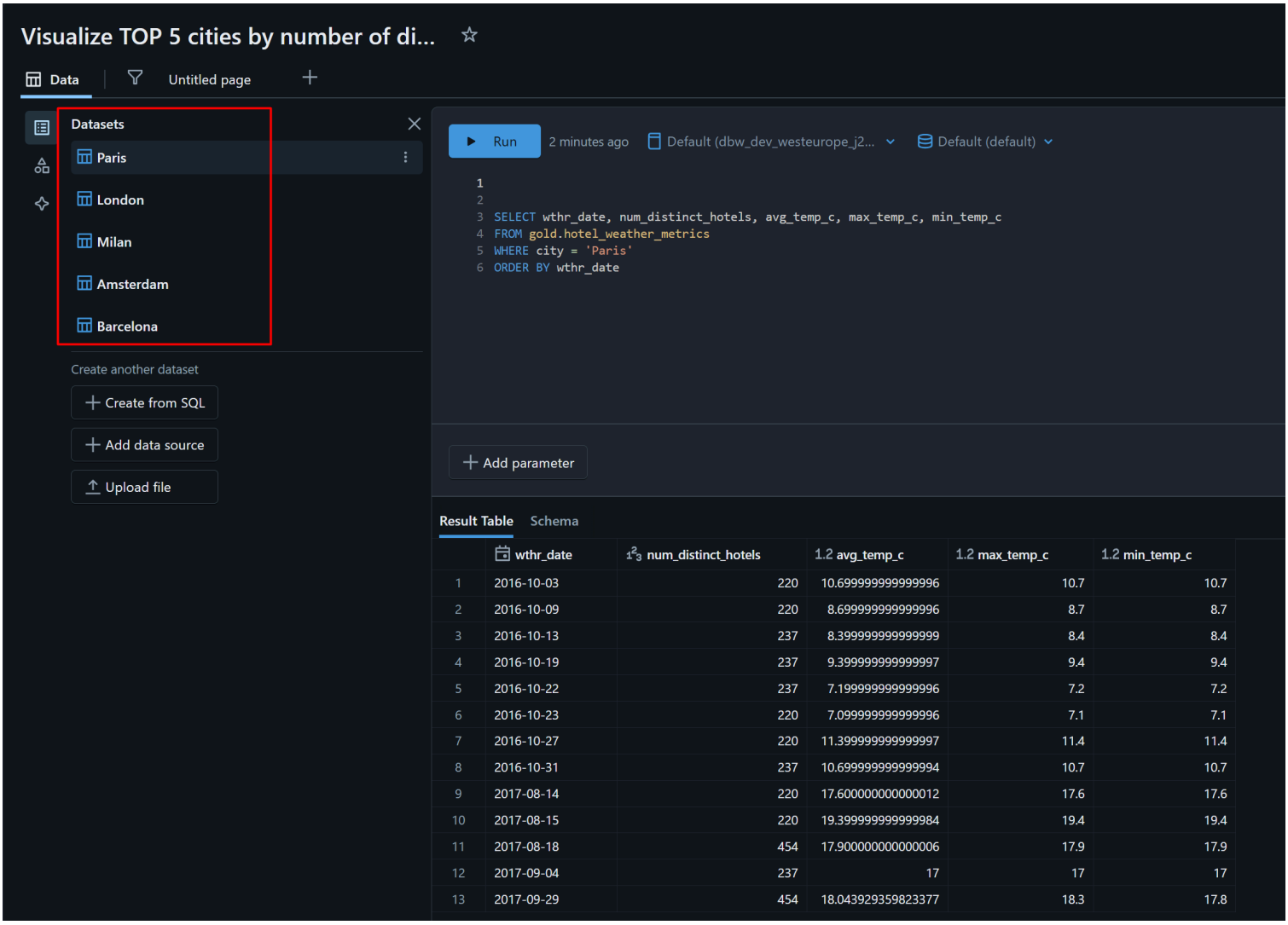
_sqldf: pyspark.sql.dataframe.DataFrame = [tbl_name: string, cnt: long]

Table +

	tbl_name	cnt
1	count bronze.hotel_weather_raw	13330
2	count silver.hotel_weather_processed	13330
3	count gold.hotel_weather_metrics	4324

5.Data Visualization:

Datasets:



	wthr_date	num_distinct_hotels	avg_temp_c	max_temp_c	min_temp_c
1	2016-10-03	220	10.6999999999999996	10.7	10.7
2	2016-10-09	220	8.6999999999999996	8.7	8.7
3	2016-10-13	237	8.3999999999999999	8.4	8.4
4	2016-10-19	237	9.3999999999999997	9.4	9.4
5	2016-10-22	237	7.1999999999999996	7.2	7.2
6	2016-10-23	220	7.0999999999999996	7.1	7.1
7	2016-10-27	220	11.3999999999999997	11.4	11.4
8	2016-10-31	237	10.6999999999999994	10.7	10.7
9	2017-08-14	220	17.6000000000000012	17.6	17.6
10	2017-08-15	220	19.3999999999999984	19.4	19.4
11	2017-08-18	454	17.9000000000000006	17.9	17.9
12	2017-09-04	237	17	17	17
13	2017-09-29	454	18.043929359823377	18.3	17.8