

Azure Project Steps/Demo

Obj: Use Data Flow to move data to Azure Data Lake Storage or Azure Blob Storage. Use Databricks to read and write data from and to these data stores, and also perform advanced analytics and machine learning tasks.

The screenshot shows the 'New linked service' dialog for 'Azure Data Lake Storage Gen2' in the Azure Data Factory portal. The dialog is divided into several sections:

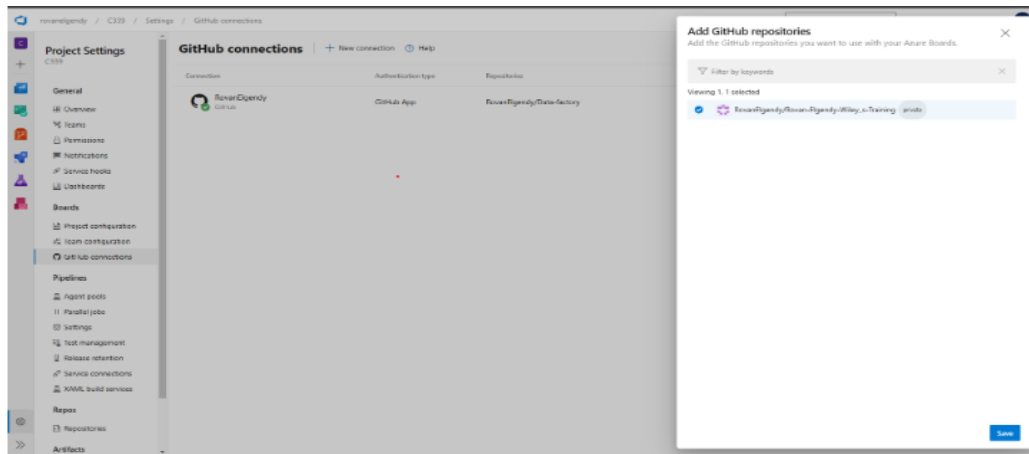
- Name:** 'AzureDataLakeStorage1'
- Description:** (Empty text box)
- Connect via integration runtime:** 'AutoResolveIntegrationRuntime'
- Authentication type:** 'Account key'
- Account selection method:** 'From Azure subscription' (selected), with 'Azure subscription' dropdown showing 'Azure subscription 1 (0b52c109-b0d1-4b02-9a0e-11d0c4b95081)'.
- Storage account name:** 'mslondy' (with a refresh icon)
- Test connection:** 'To linked service' (selected), with a 'Test connection' button.
- Annotations:** '+ New' button
- Parameters:** '+ New' button
- Advanced:** (Expanded section with '+ New' button)

On the right, the 'Set properties' pane is visible, showing:

- Name:** 'monthly_revenue_original'
- Linked service:** 'AzureDataLakeStorage1' (dropdown)
- File path:** 'historicaldata / Project/azure project / Month_Value_Lake' (with a refresh icon)
- First row as header:** ☒
- Import schema:** 'From connection/store' (selected), with 'From sample file' and 'None' as other options.

At the bottom, there are 'OK', 'Back', and 'Cancel' buttons.

Setting up the Linked Services



Git configuration/integration to Git Repo
from organisation settings
after adding new Repo in Git

Source settings Source options **Projection** Optimize Inspect Data preview

Define default format Detect data type Import projection Reset schema

Column name	Type	Format
Period	date	dd.MM.yyyy
Revenue	float	Specify format
Sales_quantity	integer	Specify format
Average_cost	float	Specify format
The_average_annual_payroll_of_the_region	integer	Specify format

1- setting the data type for each column

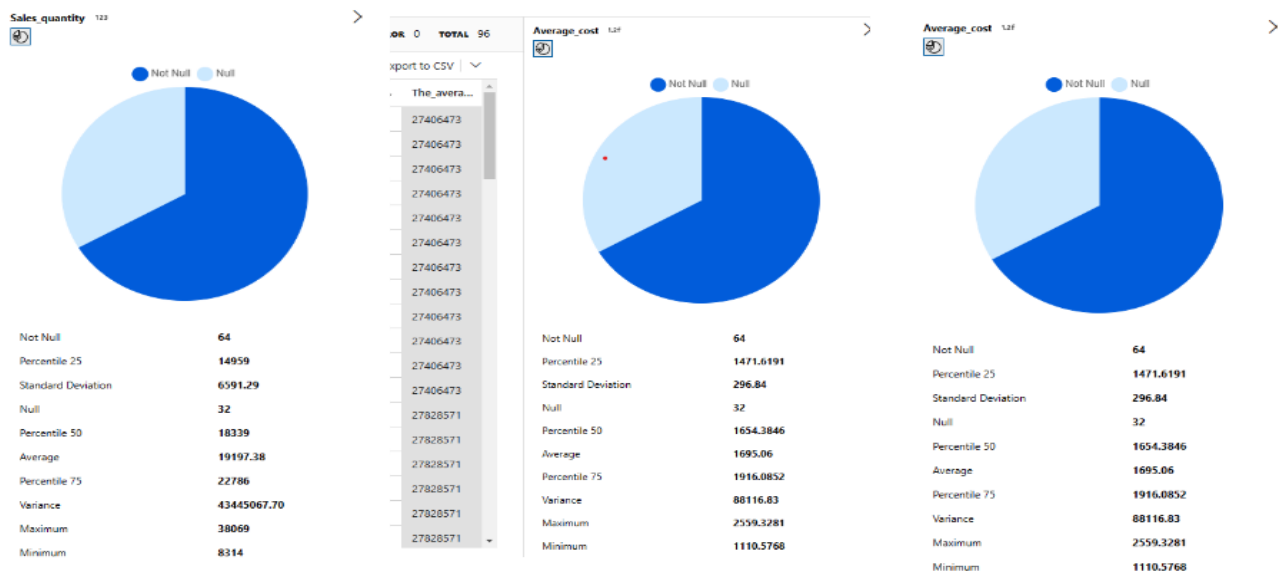
Source settings Source options Projection **Optimize** Inspect Data preview

Number of rows: INSERT 96 UPDATE DELETE UPSERT LOOKUP ERROR TOTAL 95

Refresh Typecast Modify Map drifted Statistics Remove Export to CSV

Period	Revenue	Sales_quantity	Average_cost	The_average_annua...
2015-01-01	1.6010072E7	12729	1257.7635	3002.4676
2015-02-01	1.5807587E7	11636	1358.507	3002.4676
2015-03-01	2.2047146E7	15922	1384.697	3002.4676
2015-04-01	1.8814584E7	15227	1285.6067	3002.4676
2015-05-01	1.402148E7	8620	1626.6217	3002.4676
2015-06-01	1.6783928E7	13160	1275.3745	3002.4676

2- Data preview



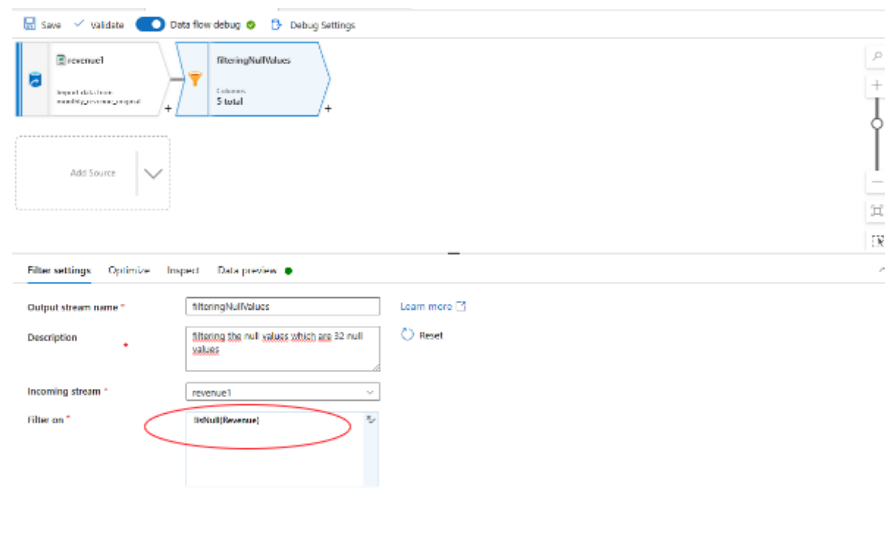
Revenue 1/27



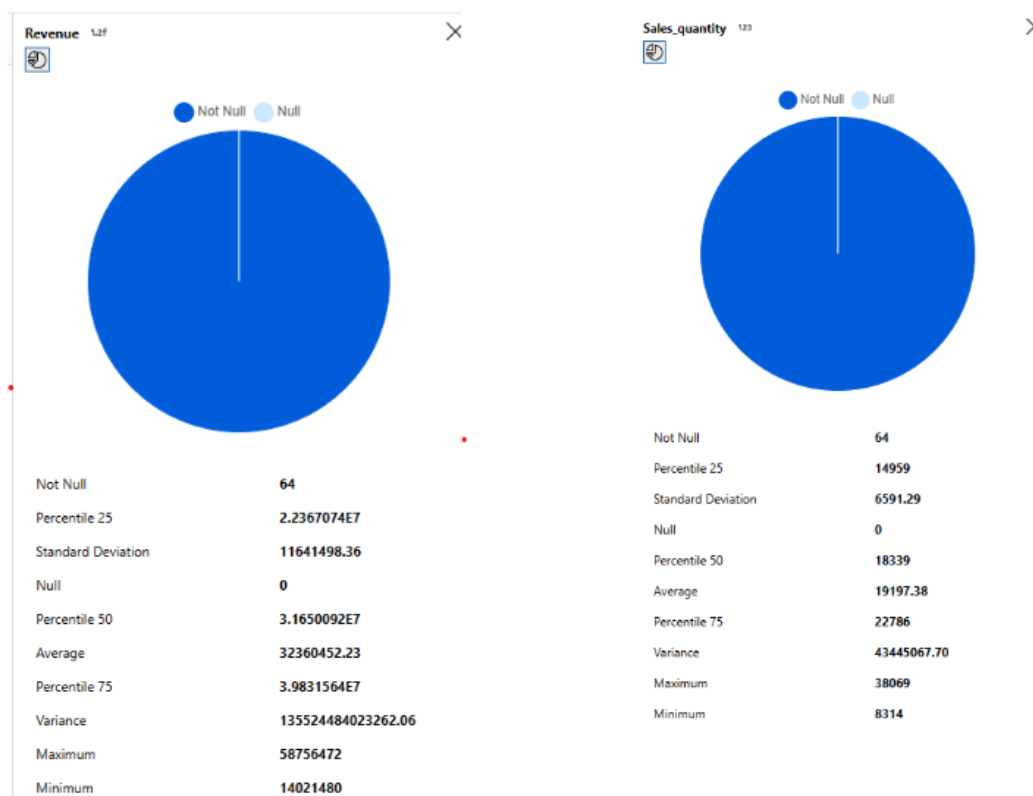
Not Null	64
Percentile 25	2.2367074E7
Standard Deviation	11641498.36
Null	32
Percentile 50	3.1650092E7
Average	32360452.23
Percentile 75	3.9831564E7
Variance	135524484023262.06
Maximum	58756472
Minimum	14021480

Data cleaning

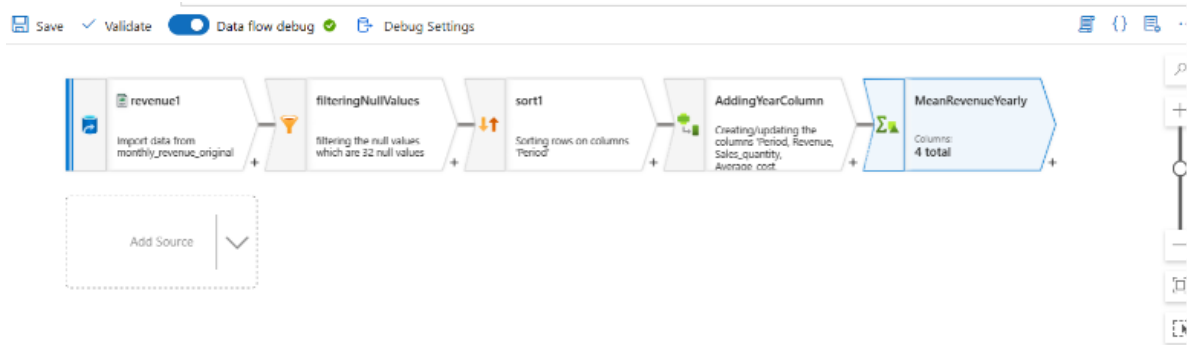
- Check the nan values from each column using the statistics above
- Using Filter data flow using the expression: `!isNull(Revenue)`



1- using Filter dataflow to remove the nan values using builder expression



2- Columns after filtering has zero Null values



Aggregate settings Optimize Inspect Data preview

Incoming stream * AddingYearColumn

Group by Aggregates

Columns	Name as
123 Year	Year

Aggregate settings Optimize Inspect Data preview

Output stream name * MeanRevenueYearly [Learn more](#)

Description [getting the mean revenue yearly](#) [Reset](#)

Incoming stream * AddingYearColumn

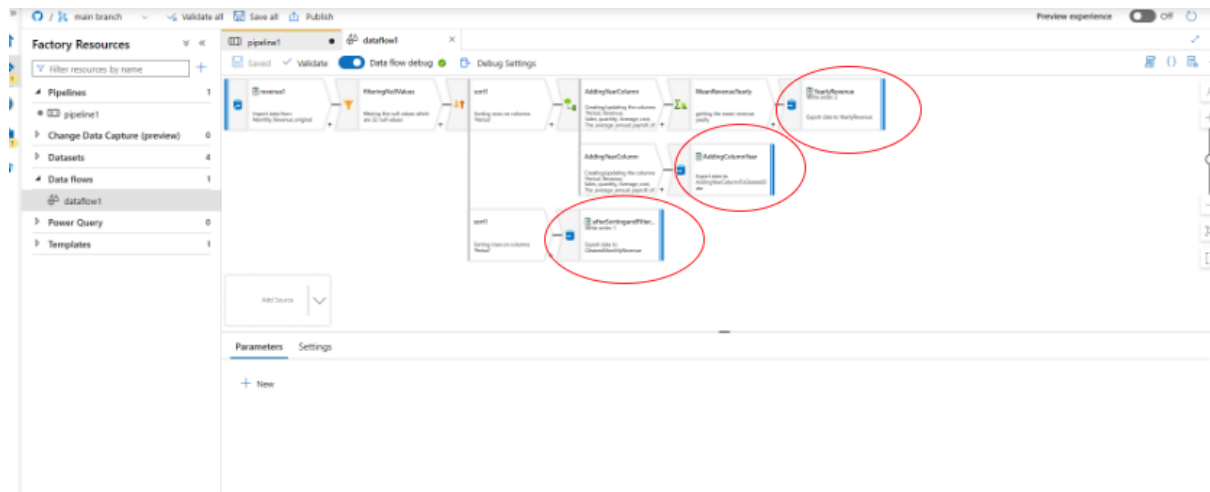
Group by Aggregates

Grouped by: Year

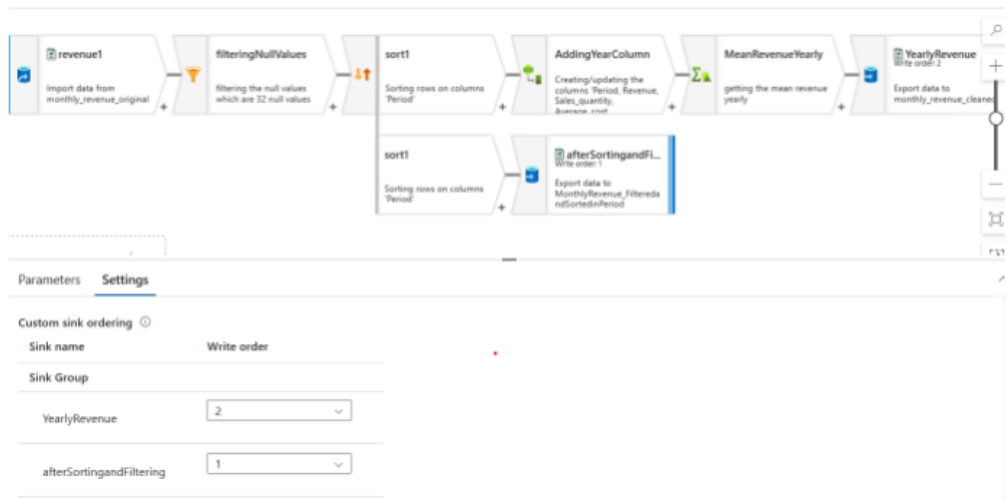
+ Add Clone Delete Open expression builder

Column	Expression
Mean_Revenue	toDecimal(avg(Revenue))
Mean_Sales_quantity	toDecimal(avg(Sales_quantity))
Mean_cost_of_Production	toDecimal(avg(Average_cost))

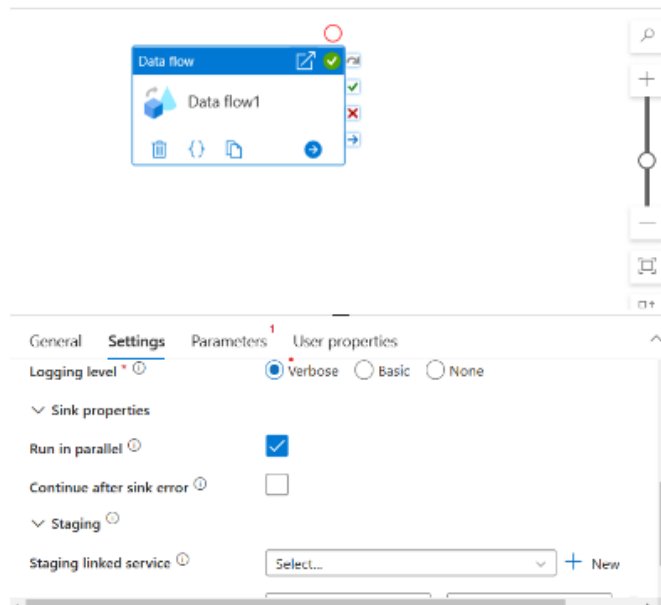
Getting the Avg of all columns per year using the aggregate transformation



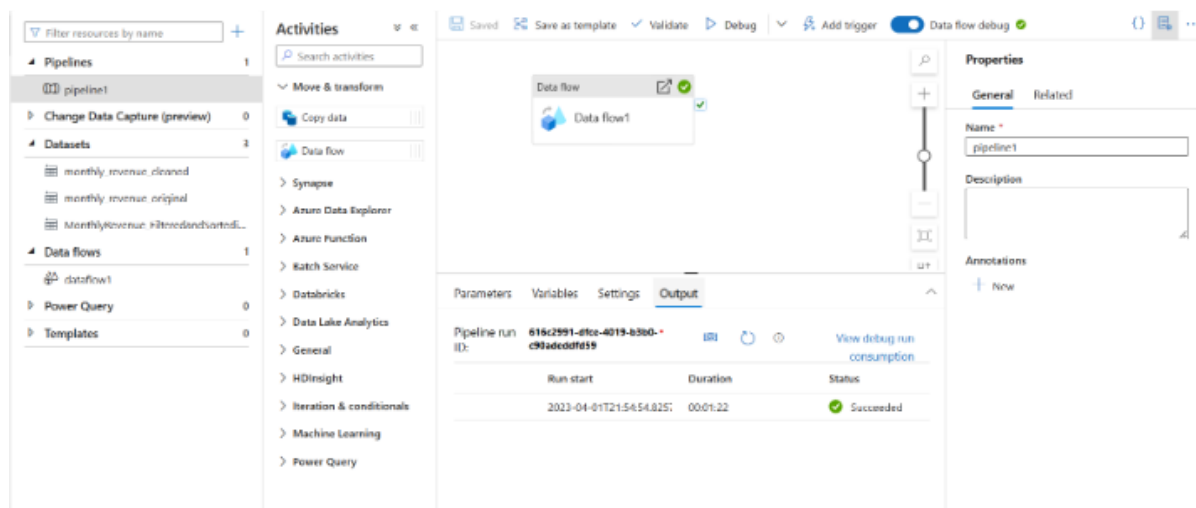
three sinks are added:
afterSortingandFiltering
AddingColumnYear
yearlyAvgRevenue



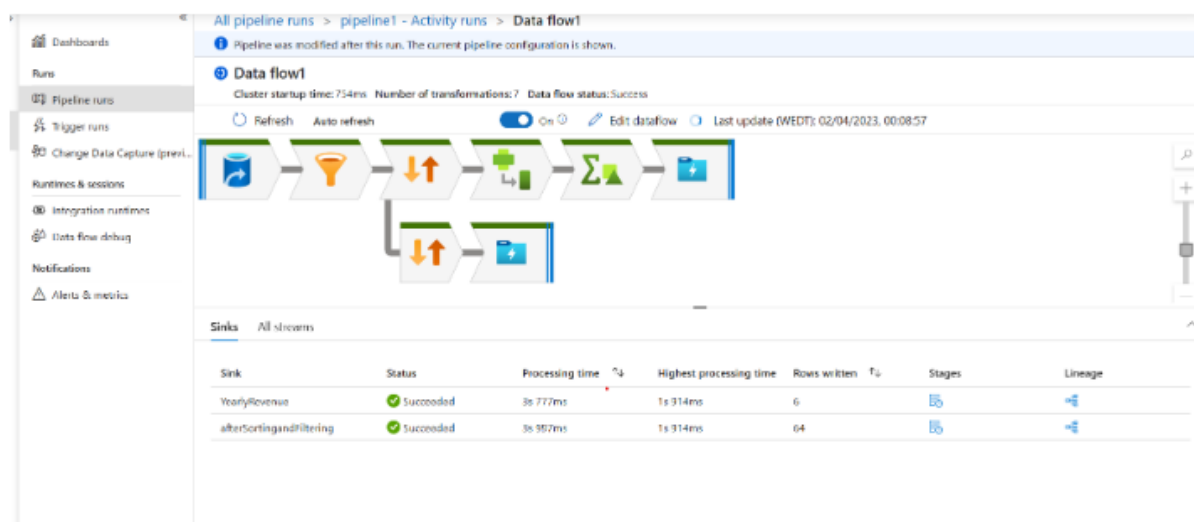
Writing the order of the three sinks



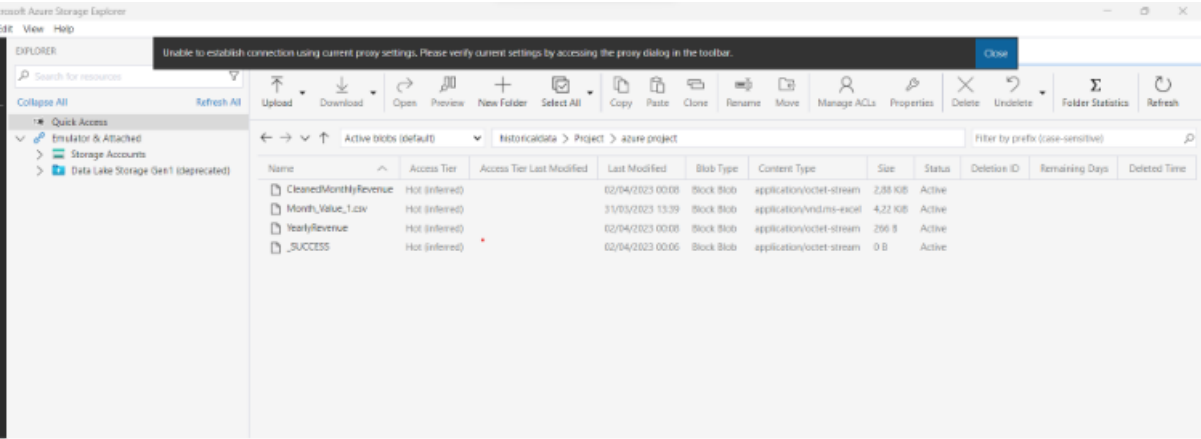
Creating new pipeline with our dataflow setting the sinks to run in parallel



Debugging the pipeline



Dataflow Debugging Dashboard after enabling the debug



the 3 Sinks have been added to the azure storage after debugging the pipeline

New trigger

Name *
UpdatingMonthRevenue

Description

Type *
Schedule

Start date * ⓘ
4/2/2023, 12:13:51 PM

Time zone * ⓘ
Coordinated Universal Time (UTC)

Recurrence * ⓘ
Every 1 Month(s)

Advanced recurrence options
☒ Month days ☐ Week days

Select day(s) of the month to execute

1	2	3	4	5	6	7
..

adding trigger to be triggered monthly

New trigger

☒ Month days ☐ Week days

Select day(s) of the month to execute

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	Last			

Execute at these times ⓘ

Hours
Minutes

Schedule execution times
12:13

☐ Specify an end date

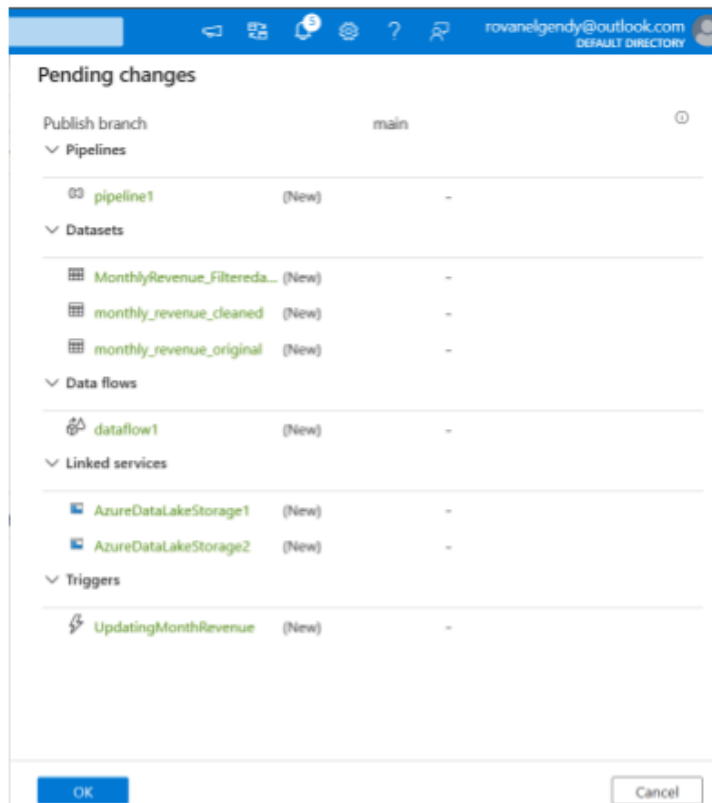
Annotations
+ New

Start trigger ⓘ
☒ Start trigger on creation

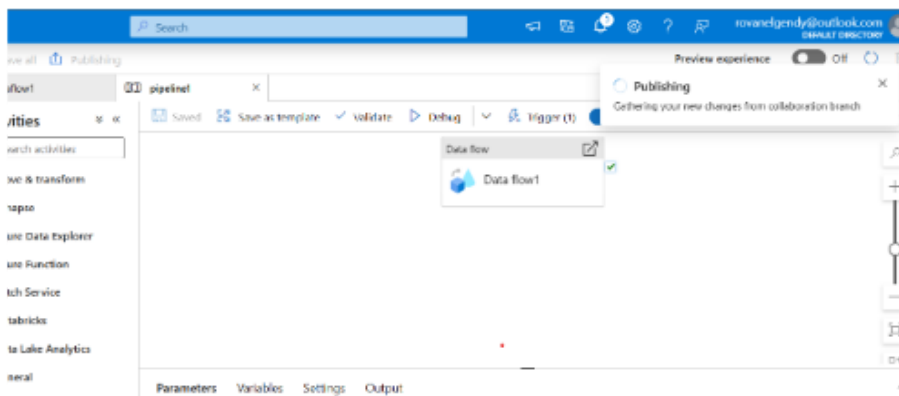
OK Cancel

Setting the trigger

Setting the trigger

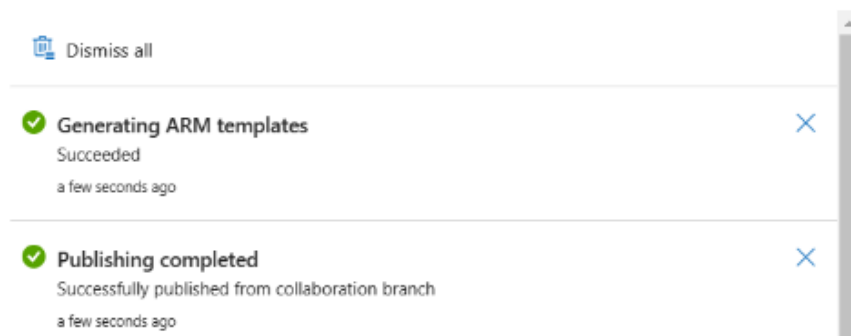


Saved and to be published



published

Notifications



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
55	01.01.2019	67.09601	6448.9998	1413.5955	246.4267	198.0605																	
56	01.02.2019	67.97141	5299.9777	2709.4094	81.553	198.0605																	
57	01.03.2019	69.0164	1077.1443	75.052	818.8994	78.0605																	
58	01.04.2019	68.56768	1544.7675	1086.8816	9846.17	198.0605																	
59	01.05.2019	64.0476	1104.2601	167.17	1088.07	78.0605																	
60	01.06.2019	64.8756	8784.8784	1786.164	438.0844	198.0605																	
61	01.07.2019	67.14275	1695.9809	1746.4744	444.895	198.0605																	
62	01.08.2019	67.8838	1827.1784	210.6408	41.0745	198.0605																	
63	01.09.2019	68.77586	164.75495	171.05	1389.0447	198.0605																	
64	01.10.2019	69.7716	2727.32505	157.807	749.9571	198.0605																	
65	01.11.2019	72.8869	1903.7615	1916.848	1912.115	198.0605																	
66	01.12.2019	72.9777	1627.7675	1916.848	1912.115	198.0605																	
67	01.01.2020																						
68	01.02.2020																						
69	01.03.2020																						
70	01.04.2020																						
71	01.05.2020																						
72	01.06.2020																						
73	01.07.2020																						
74	01.08.2020																						
75	01.09.2020																						
76	01.10.2020																						
77	01.11.2020																						
78	01.12.2020																						
79	01.01.2021																						
80	01.02.2021																						

Updating the original dataset by one row

Activity runs

pipeline run id: 93b5e6d0c7d8-1c4b-v515-7716a67022a

All status: Export to CSV

Activity name	Status	Activity type	Run start	Duration	Log	Integration runtime
Data flow1	In progress	Data flow	4/2/2023, 2:58:04 PM	00:00:21		

pushing the trigger we made earlier manually

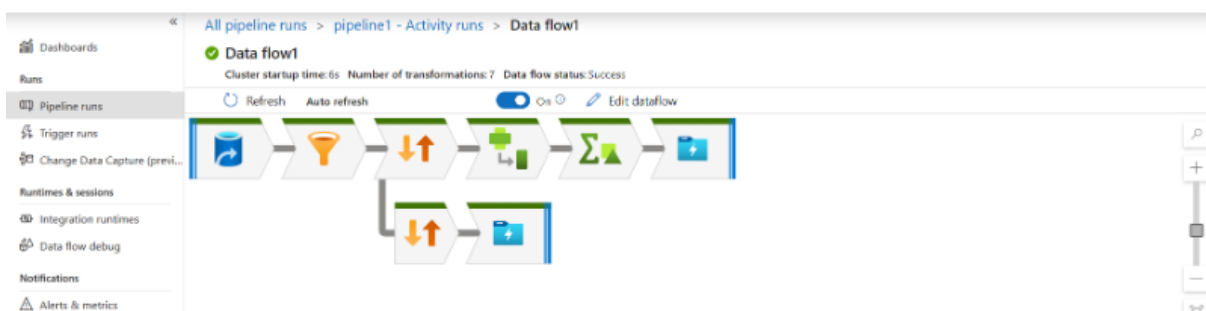


Notifications

Dismiss all

UpdatingMonthRevenue
UpdatingMonthRevenue has been saved.
a few seconds ago

Trigger success notification



Sinks All streams						
Sink	Status	Processing time	Highest processing time	Rows written	Stages	Lineage
YearlyRevenue	Succeeded	4s 189ms	2s 212ms	6		
afterSortingandFiltering	Succeeded	4s 439ms	2s 212ms	65		

here we can see after trigger the rows read is updated to 65 rows instead of 64 rows

pipeline1 dataflow1

Saved Validate Data flow debug Debug Settings

revenue1 Import data from monthly_revenue_original

filteringNullValues filtering the null values which are 32 null values

sort1 Sorting rows on columns Period

AddingYearColumn Creating/updating the columns Period, Revenue, Sales_quantity, Average_cost

MeanRevenueYearly getting the mean revenue yearly

YearlyRevenue Write order:2 Export data to monthly_revenue_cleaned

sort1 Sorting rows on columns Period

afterSortingandFiltering Write order:1 Columns: 5 total

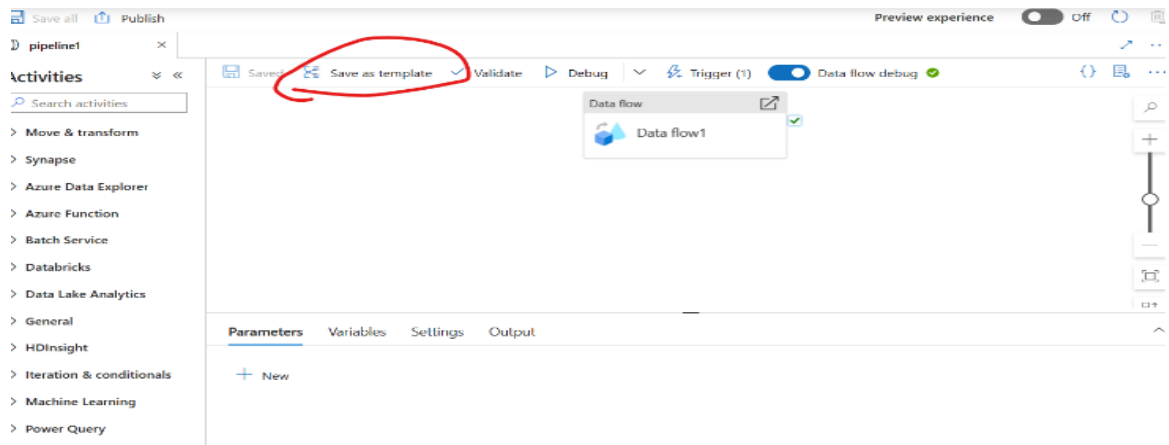
Sink Settings Errors Mapping Optimize Inspect Data preview

Number of rows + INSERT N/A + UPDATE N/A x DELETE N/A + UPSERT N/A LOOKUP N/A ERROR N/A TOTAL 65

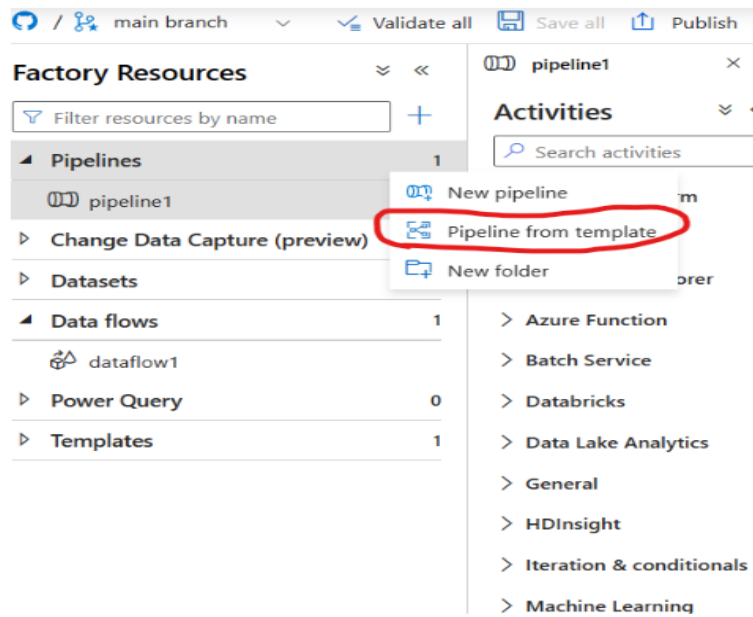
Refresh Statistics Export to CSV

Period	Revenue	Sales_quantity	Average_cost	The_average_annua...
2020-03-01	50022165	22369	1525.8827	29044998
2020-04-01	52320693	26615	1965.8348	29044998
2020-05-01	52320693	26615	1965.8348	29044998

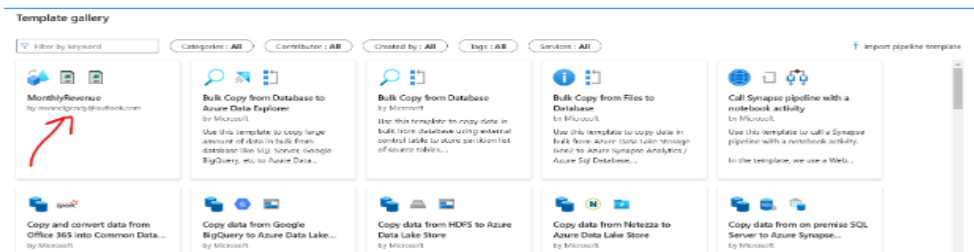
the updated row in the data preview



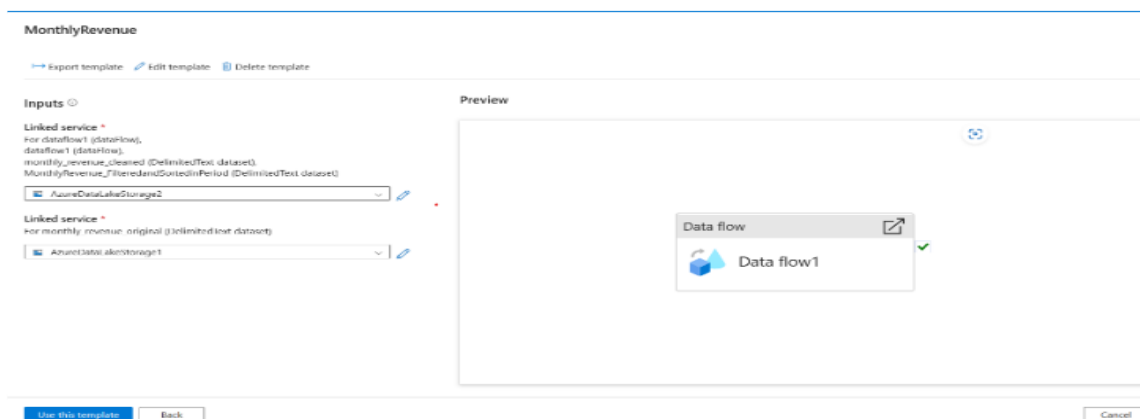
You can save the pipeline as template to import later



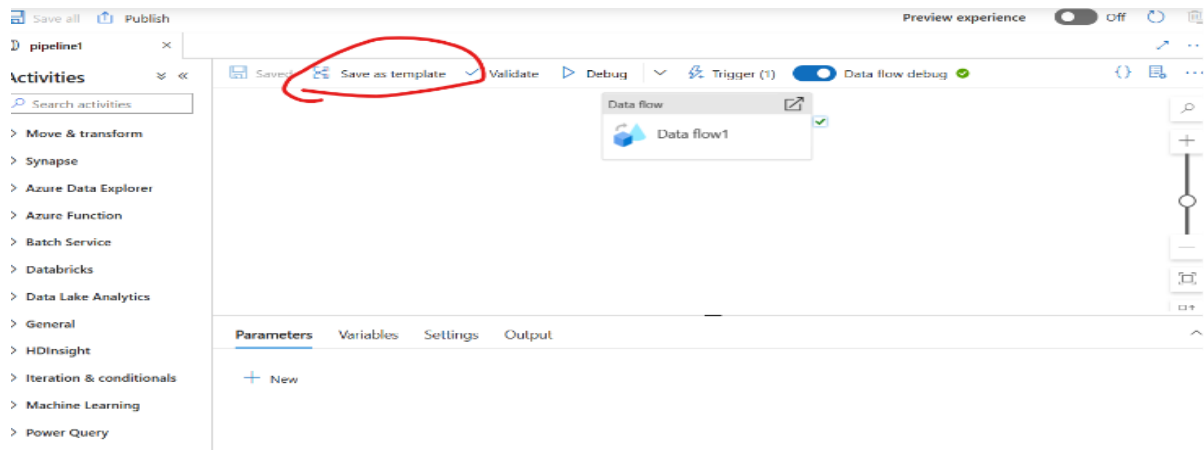
importing the template



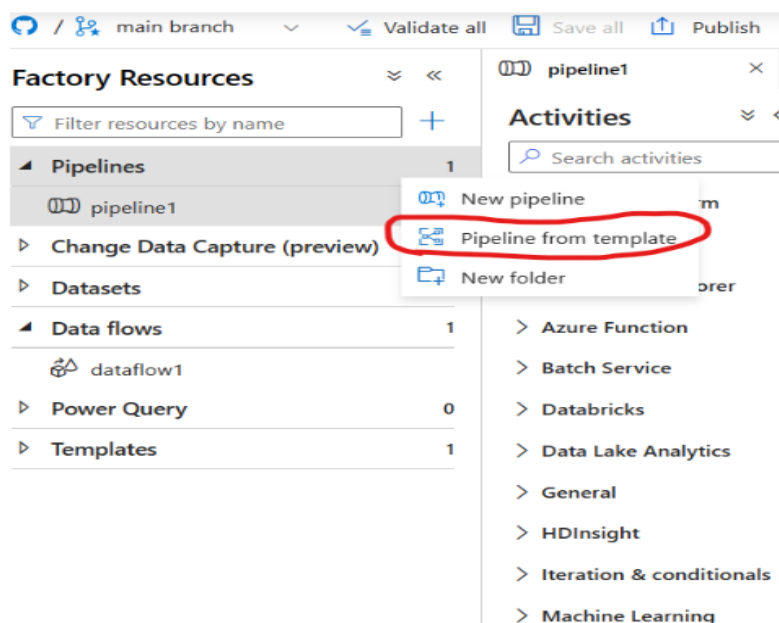
the template gallery



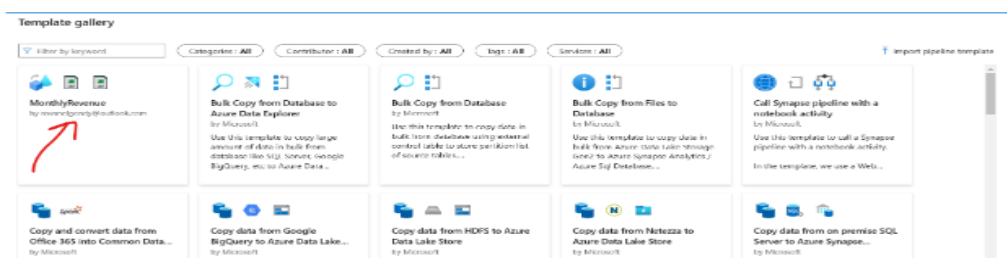
last step to click on use this template



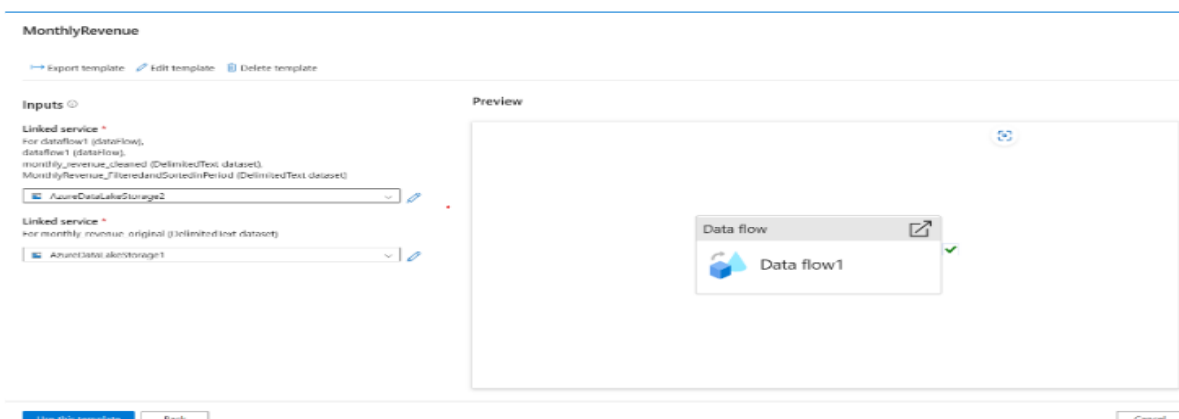
You can save the pipeline as template to import later



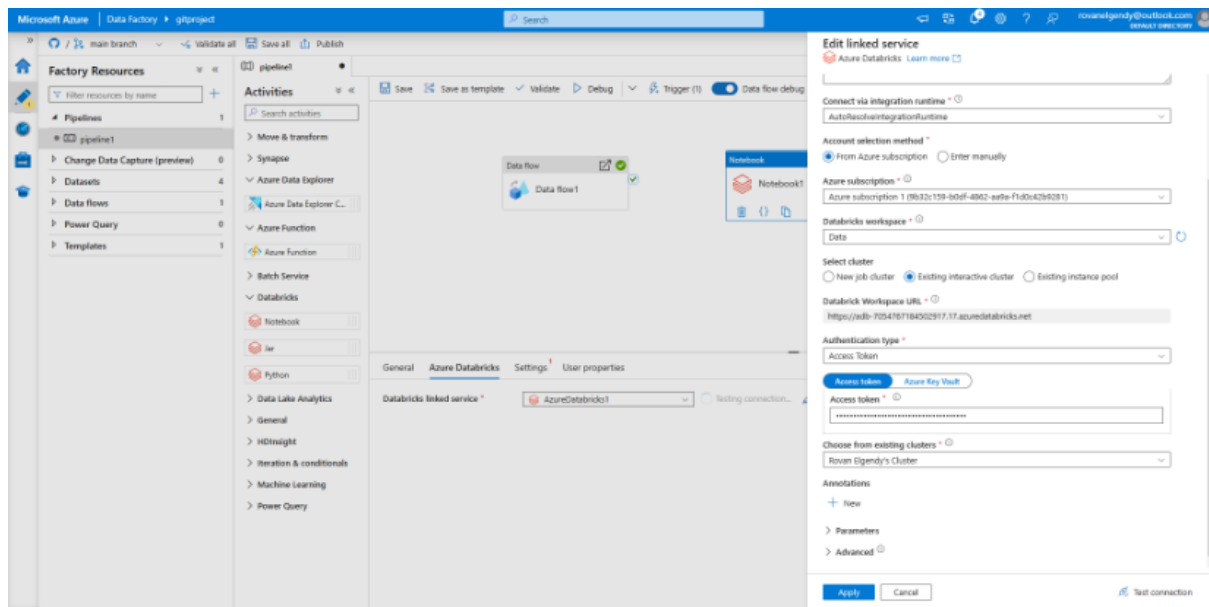
importing the template



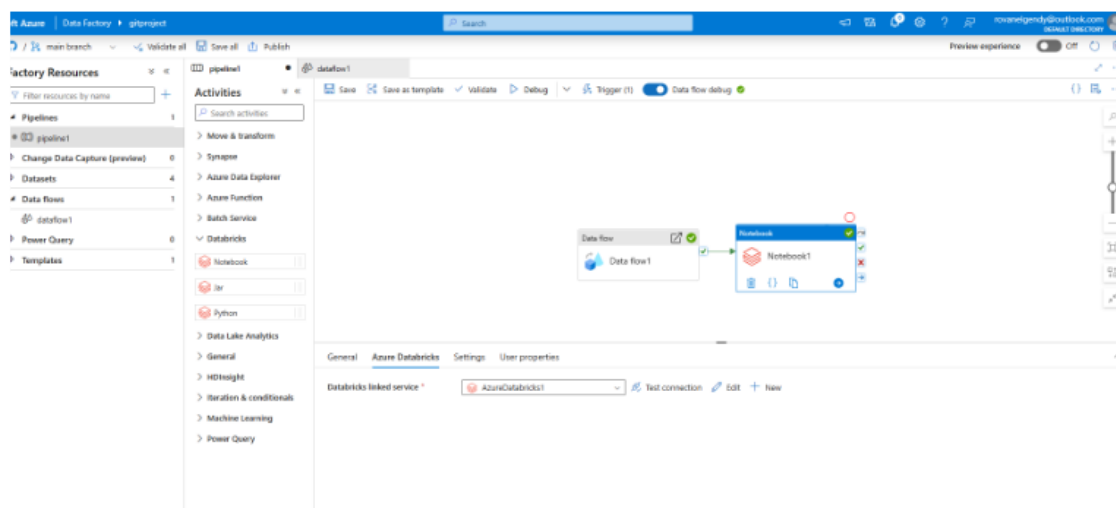
the template gallery



last step to click on use this template



Adding Databricks Service using existing cluster and the access token



Adding databricks notebook -which is already made in different file- to our dataflow

All pipeline runs > ✓ pipeline1 - Activity runs

Rerun

Refresh

Update pipeline

List

Gantt

Data flow

Data flow1

Notebook

Notebook1

+

−

[...]

[...]

Activity runs

Pipeline run ID f65edb3d-8150-4af6-b9ee-21350774a12b

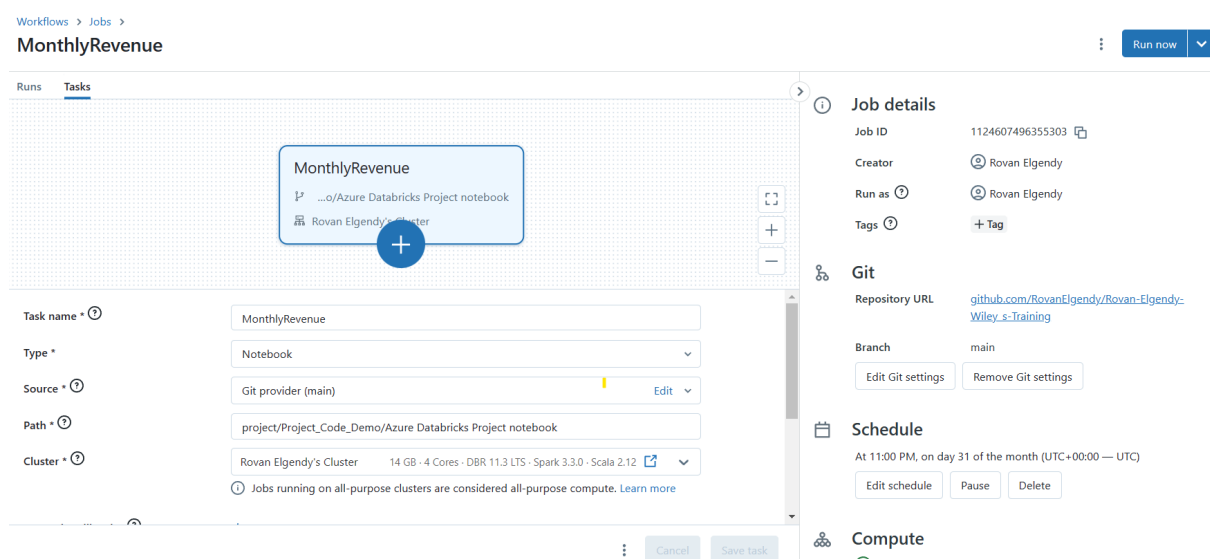
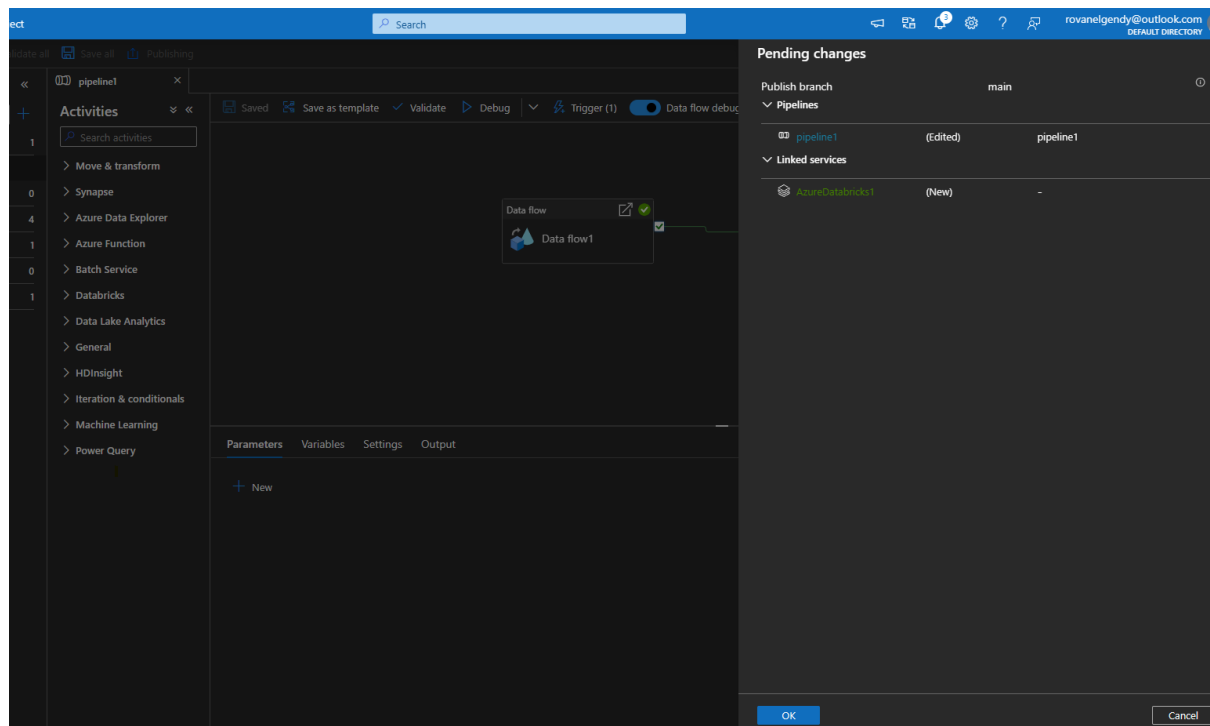
All status

Export to CSV

Showing 1 - 2 items

Activity name	Status	Activity type	Run start	Duration	Log	Integration runtime
Notebook1	✓ Succeeded	Notebook	4/5/2023, 4:16:23 PM	00:04:26		AutoResolveIntegrationRu
Data flow1	✓ Succeeded	Data flow	4/5/2023, 4:12:14 PM	00:04:08		AutoResolveIntegrationRu

Triggering the pipeline



creating workflow job in databricks

Workflows

JobsJob runsDelta Live Tables

Create job

Owned by meAccessible by meQFilter jobsColumns

Name	Created by	Trigger	Last run	Actions
MonthlyRevenue	rovanelgendy@outlook.com	At 11:00 PM, on day 31 of the month (UTC)	Succeeded	

1-1 of 1 items1

Workflows > Jobs > MonthlyRevenue

Run now

RunsTasks

Runs

Start datePreviousNext

Run total duration

MonthlyRevenue

Apr 05

Go to the latest successful run

Columns

Start time	Run ID	Launched	Duration	Spark	Status	Run parameters	Actions
Apr 5 2023, 00:34 ...	7136	Manually	23s	Spark UI / Logs / Metrics	Succeed...		

Job details

Job ID1124607496355303

CreatorRovan Elgendy

Run asRovan Elgendy

Tags+ Tag

Git

Repository URLgithub.com/RovanElgendy/Rovan-Elgendy-Wiley-s-Training

Branchmain

Edit Git settingsRemove Git settings

Schedule

At 11:00 PM, on day 31 of the month (UTC+00:00 — UTC)

Edit schedulePauseDelete

Workflows > Jobs > MonthlyRevenue >

MonthlyRevenue run

Delete job run

Output Dashboard: MonthlyRevenueDashboard Export as HTML

MonthlyRevenueDashboard

Monthly Revenue Dataset

	Period	Revenue	Sales_quantity	Average_cost	The_average_annual_payroll_of_the_region	Year
60	2019-12-01	58756472	38069	1543	29878525	2019
61	2020-01-01	56288300	27184	2070	29044998	2020
62	2020-02-01	40225244	23509	1711	29044998	2020
63	2020-03-01	50022164	32569	1535	29044998	2020
64	2020-04-01	52320692	26615	1965	29044998	2020
65	2020-05-01	52320692	26615	1965	29044998	2020
66	2020-06-01	58756472	38069	1543	29878525	2020
67	2020-07-01	44160416	26603	1659	29878525	2020

From Statistics, it is found that:

- Revenue increases through the years with increasing the

Yearly Revenue

Duration: 24s

Status: ✓ Succeeded

Git

Git URL: https://github.com/RovanElgendy/Rovan-Elgendy-Wiley_s-Training

Branch: main

Commit: 806a7756

Notebook

project/Project_Code_Demo/Azure Databricks Project notebook (806a7756)

Compute

✓ Rovan Elgendy's Cluster

Driver: Standard_DS3_v2 · Workers: Standard_DS3_v2 · 0 workers · 11.3 LTS (includes Apache Spark 3.3.0, Scala 2.12)

[View details](#) [Spark UI](#) [Logs](#) [Metrics](#)

relgendy | Containers

Storage account

Search + Container Change access level Restore containers Refresh Delete Give feedback

Search containers by prefix Show deleted containers

Name	Last modified	Public access level	Lease state	
<input type="checkbox"/> \$logs	3/20/2023, 4:02:19 PM	Private	Available	**
<input type="checkbox"/> \$web	4/5/2023, 1:29:35 AM	Blob	Available	**
<input type="checkbox"/> historicaldata	3/20/2023, 4:17:14 PM	Container	Available	**

Overview

Activity log

Tags

Diagnose and solve problems

Access Control (IAM)

Data migration

Events

Storage browser

Data storage

Containers

File shares

Queues

Tables

creating \$web container in blob storage to store web files .html to present the workflow of the project.