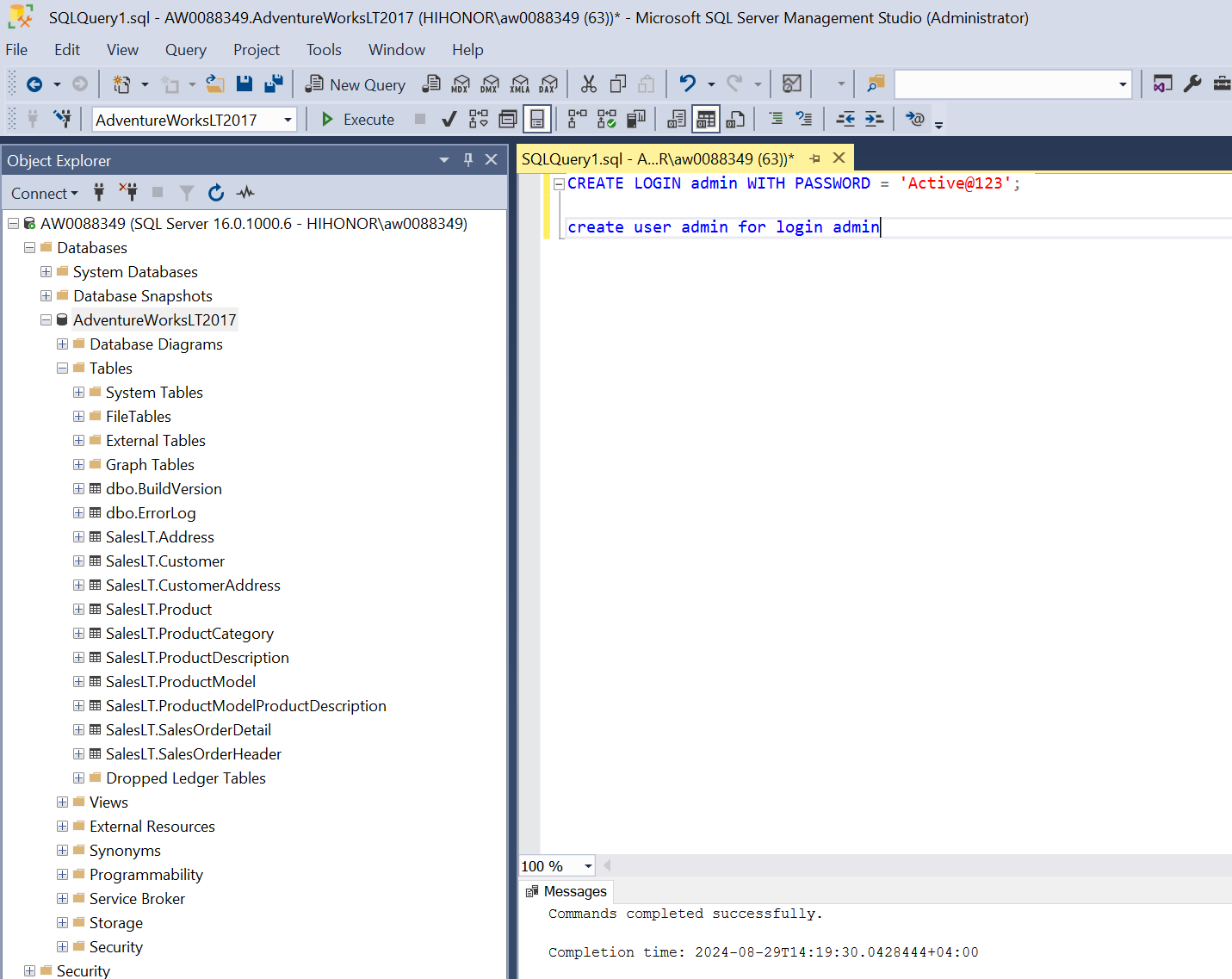
1. Install Microsoft SQl Server and also install SSMS
2. Restore AdventureWorksLT2017 databse in MSSQL server.
3. Create Login for a user:

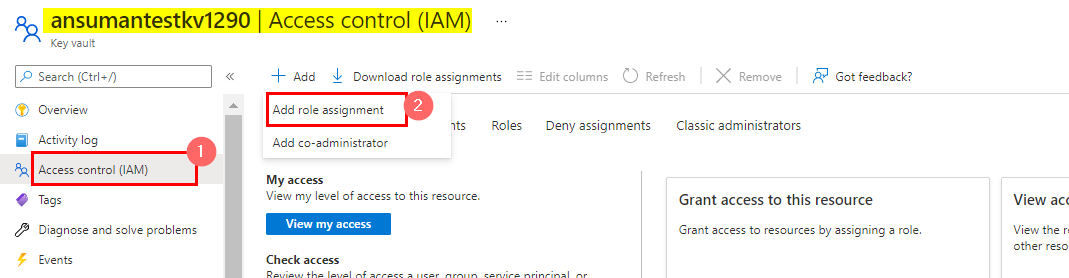


1. Go to security 🡪under user section 🡪click on user and select properties 🡪 click on membership🡪Assign **db\_datareader role**. (has reader access now)
2. Now setup Azure Resource🡪 create a resource group🡪Create a Key vault to store login credential to access on-premise SQL server

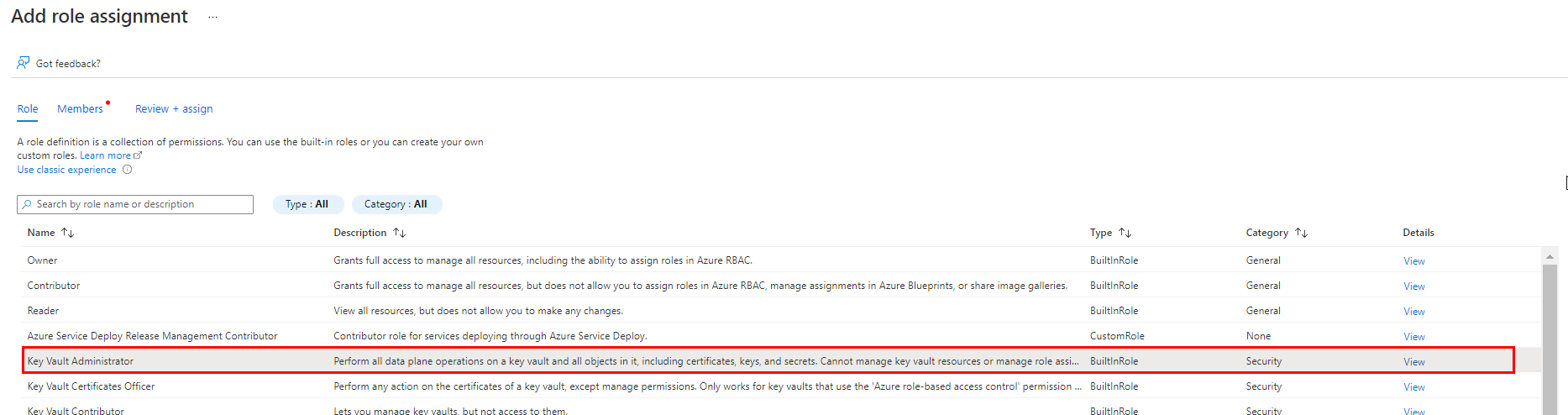
Go to Secrets🡪click on add a secret and add username and then again for password

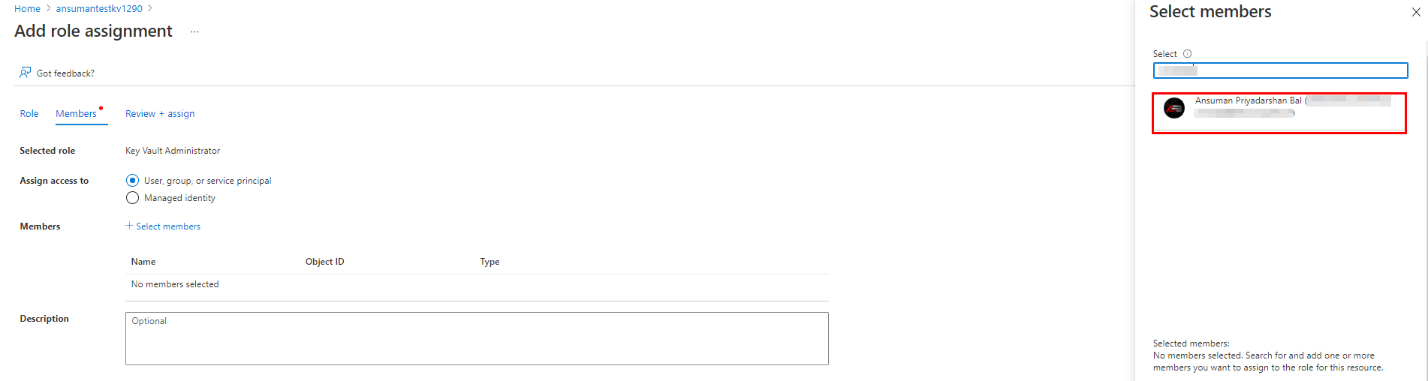
**Note if it gives error : “The operation is not allowed by RBAC. If role assignments were recently changed, please wait several minutes for role assignments to become effective.”**

1. Go to your Key vault after its created and then click on Access Control (IAM):

[](https://i.sstatic.net/xYa2e.png)

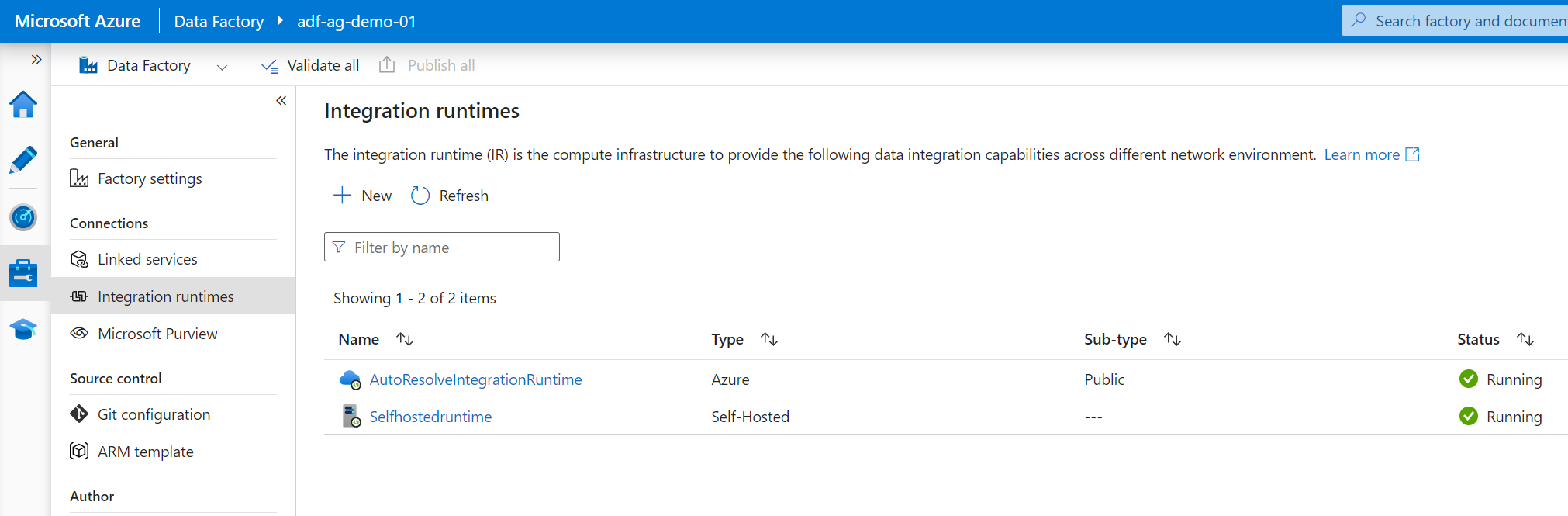
1. Then click on Add Role assignment and then add Key vault Administrator Role to your name:

[](https://i.sstatic.net/ut8yZ.png)

[](https://i.sstatic.net/954UB.png)

1. After you review and assign the role , you will be successfully able to create/manage the objects present inside the Key vault.
2. Now create a Azure data factory and then click on launch studio, we will be now copying data from on premise SQL server to Azure Data lake so for that we would need integration runtimes

Click on mange on left and then Click on Integration Runtimes🡪 Click on Add button and select Azure Self Hosted 🡪continue🡪Give name and create🡪It will download a file 🡪open the file it will download and install Microsoft Integration Runtime on your laptop.



1. Now we have to copy data from SQL server to Azure using pipeline🡪Click on Author on left 🡪Click on pipeline🡪create pipeline and the search for copy data 🡪Now we need to setup Source and Sink for this.

Select SQL server as dataset🡪create Linked service (Legacy)🡪Click on New🡪Giver Server name, Database, username and select azure key vault for password🡪Create AKV linked service🡪

Select secret name.

Test the connection.

**Note: Sometimes it’s shows Loading failed error message in that case you need to go to Key vault and then got to Access policies🡪 then create an access policy and grant all the access.**

🡪Select the respective table you want to copy.

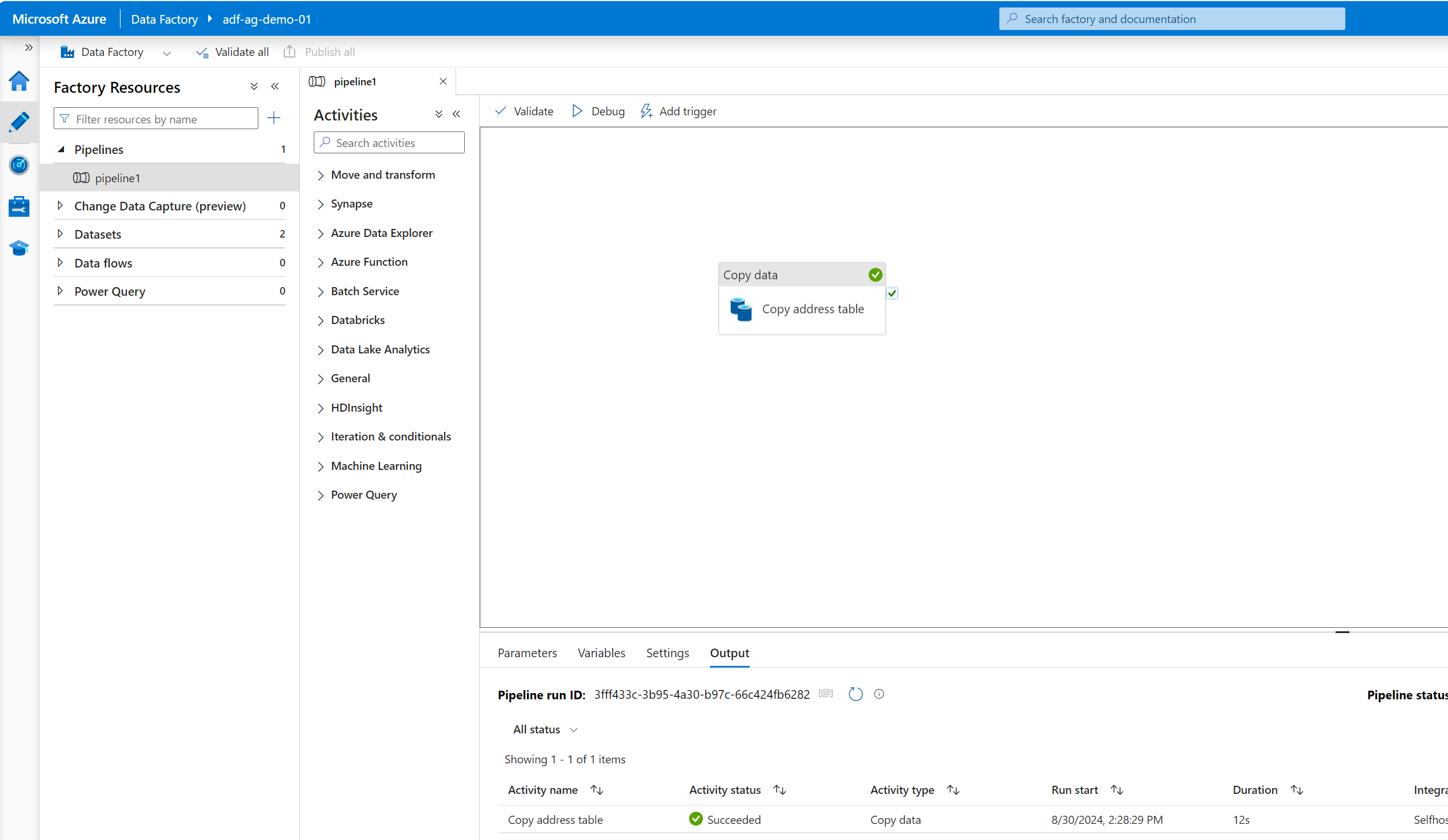
1. Now we need to configure Sink. 🡪Click on New🡪 Search for Azure gen lake2

And select format as parquet🡪Set properties🡪Configure Link Service

Test Connection

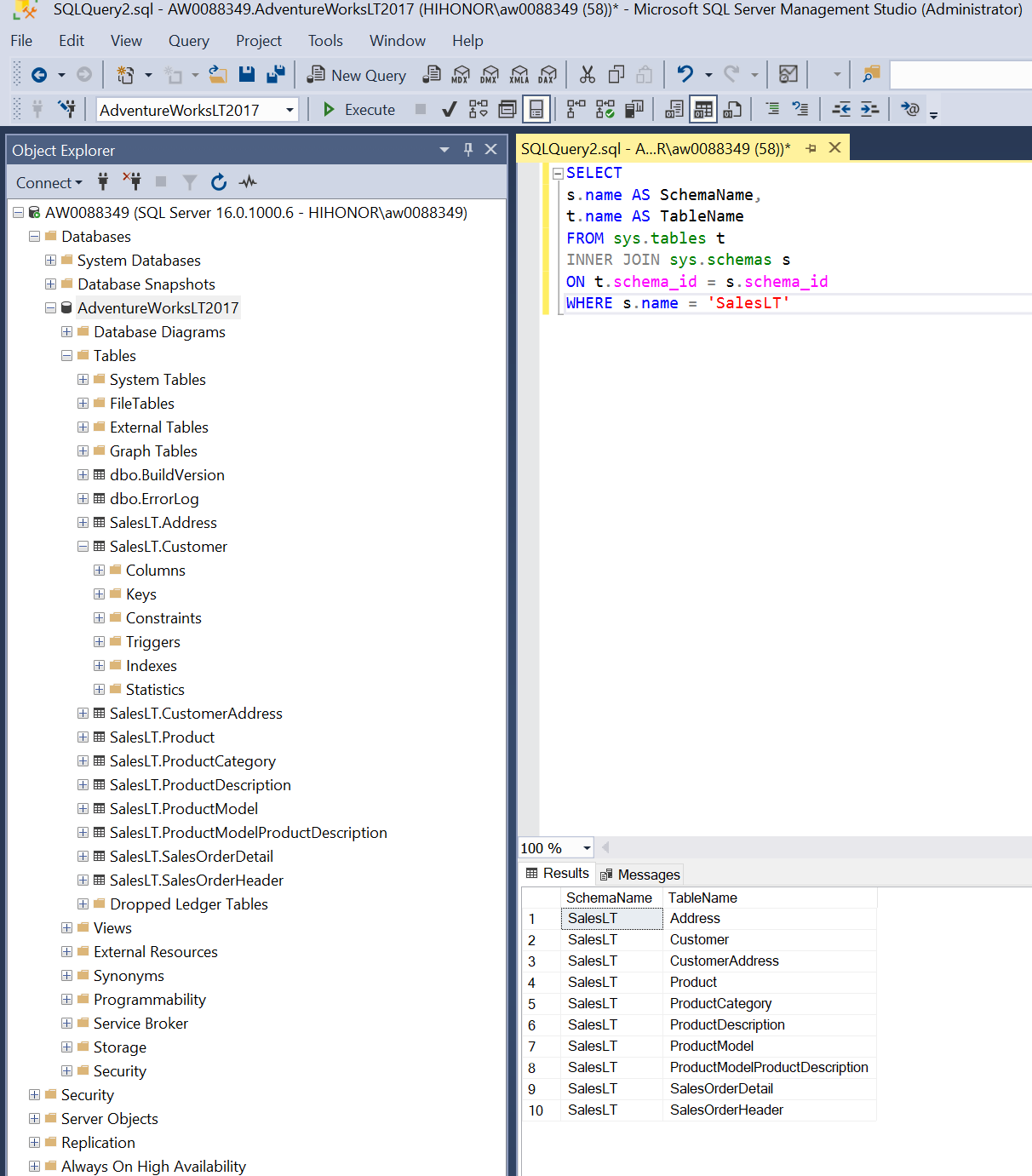
Specify the location where you need to copy files.

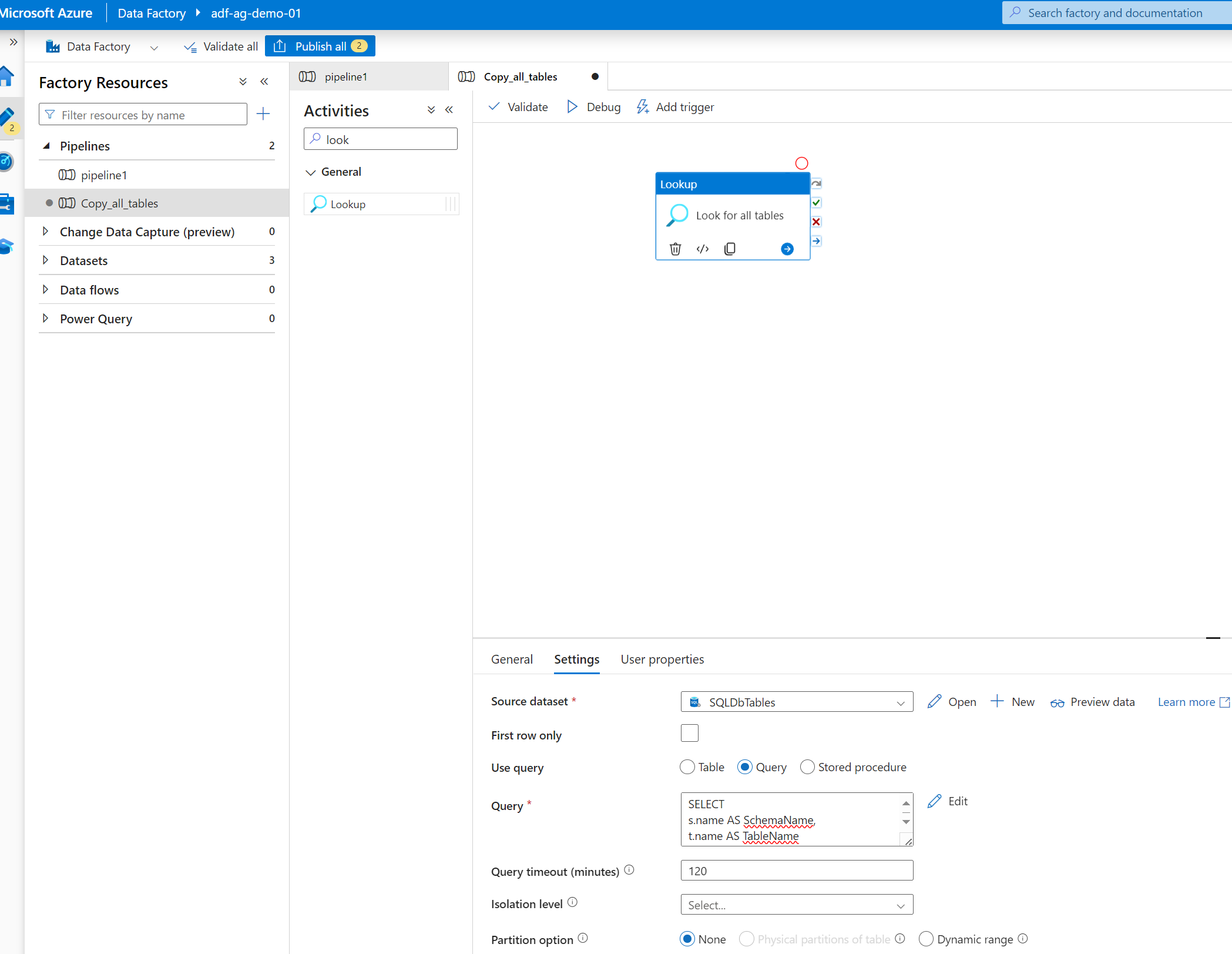
🡪Debug the copy data pipeline.



**Creating one single pipeline to copy all the tables from SQL server.**

1. First we will write query to list down all the tables to be copied.
2. Next got to ADF and select pipeline🡪give name
3. Search for lookup activity and drag it
4. Go to Settings 🡪create source dataset
5. Select use Query option🡪copy the sql query to list all the tables

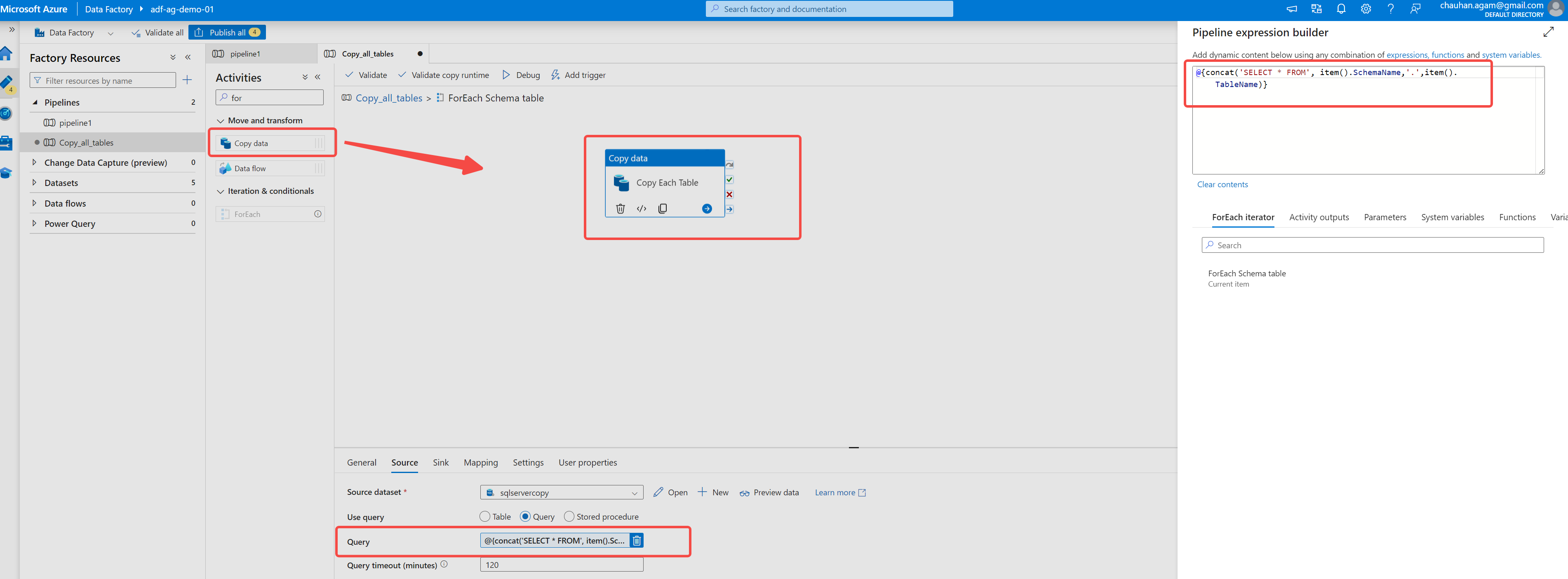




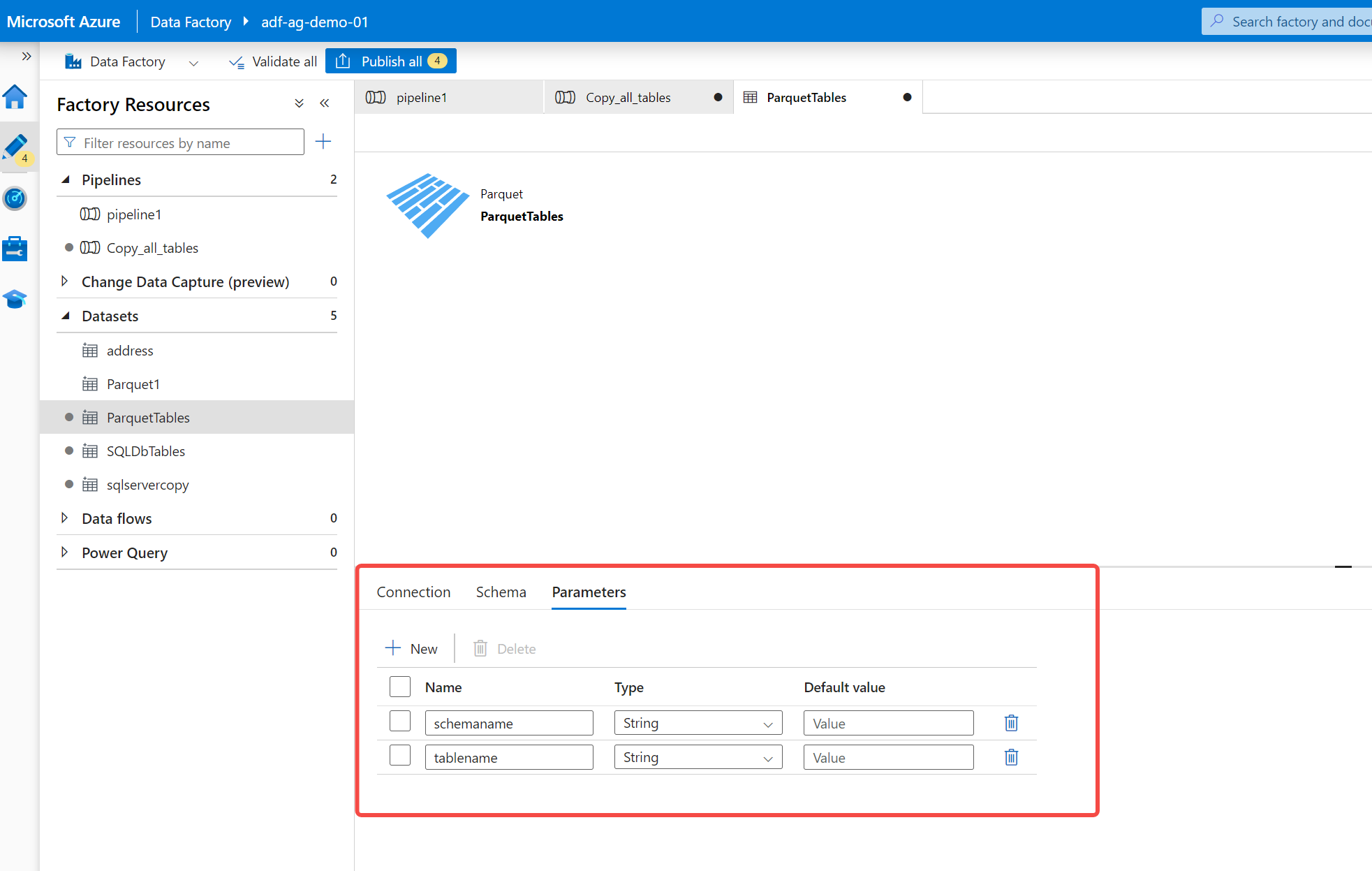
Now search for each activity and drag it 🡪click on setting 🡪select Items—then select dynamic option below.



Then click on Activity section and drag the copy data flow from left 🡪Click on Use Query option 🡪Under Query option click on dynamic field and write code in pipeline expression builder



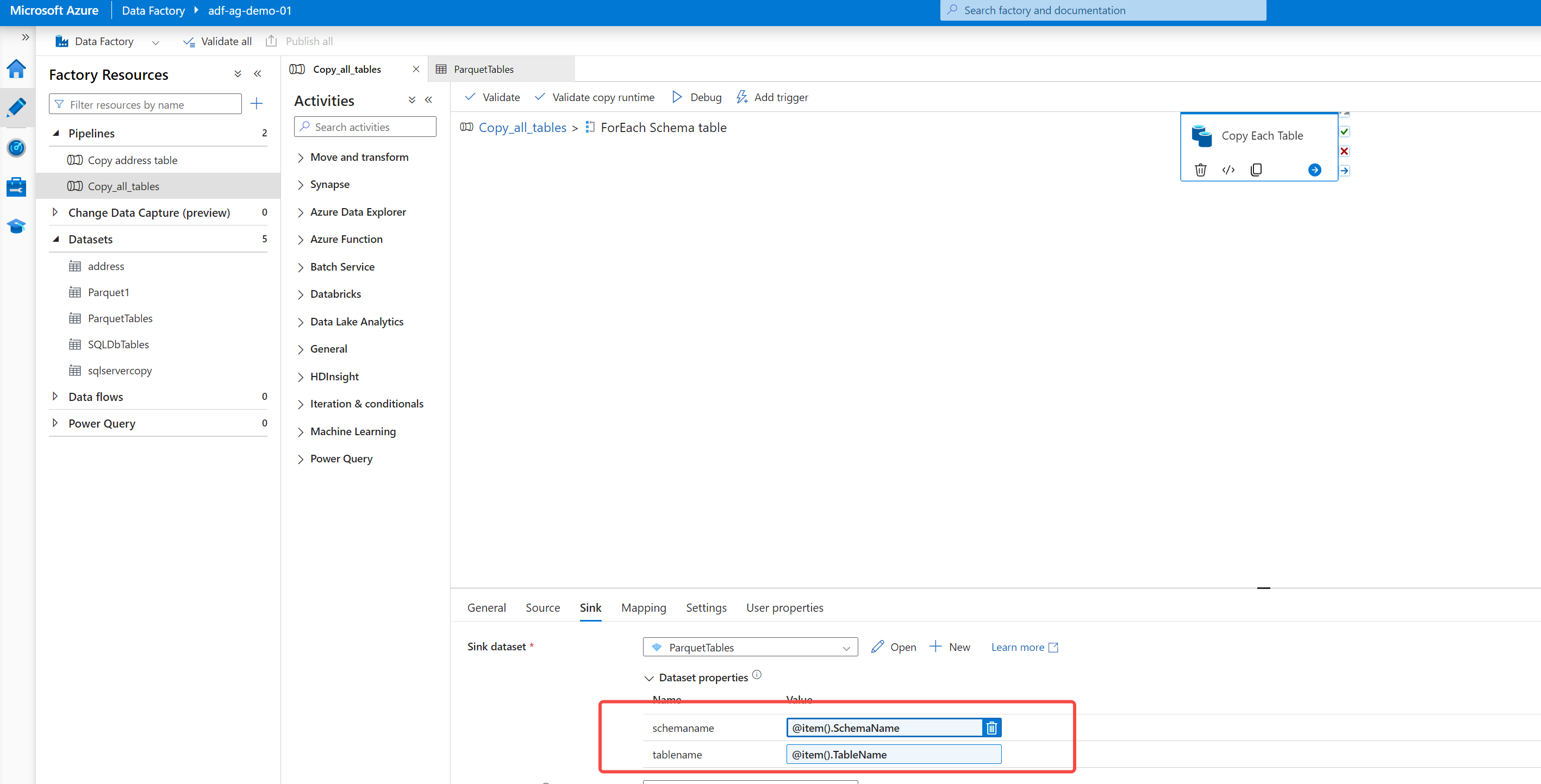
Now configure the Sink properties 🡪select azure data lake 🡪select parquet🡪go to parameters and add two parameters as below:



Folder structure need to be in **bronze/Schema/Tablename/tablename.parquet** format

We are getting schemaname and tablename from lookup activity so in source we already use the 2 item values

Now go to Sink dataset🡪you will see the above 2 parameter created 🡪click on values and click on dynamic section 🡪do same for other parameter as well



Now edit sink dataset and add these two values under the file path :



@{concat(dataset().schemaname, '/',dataset().tablename)}

@{concat(dataset().tablename,'.parquet')}

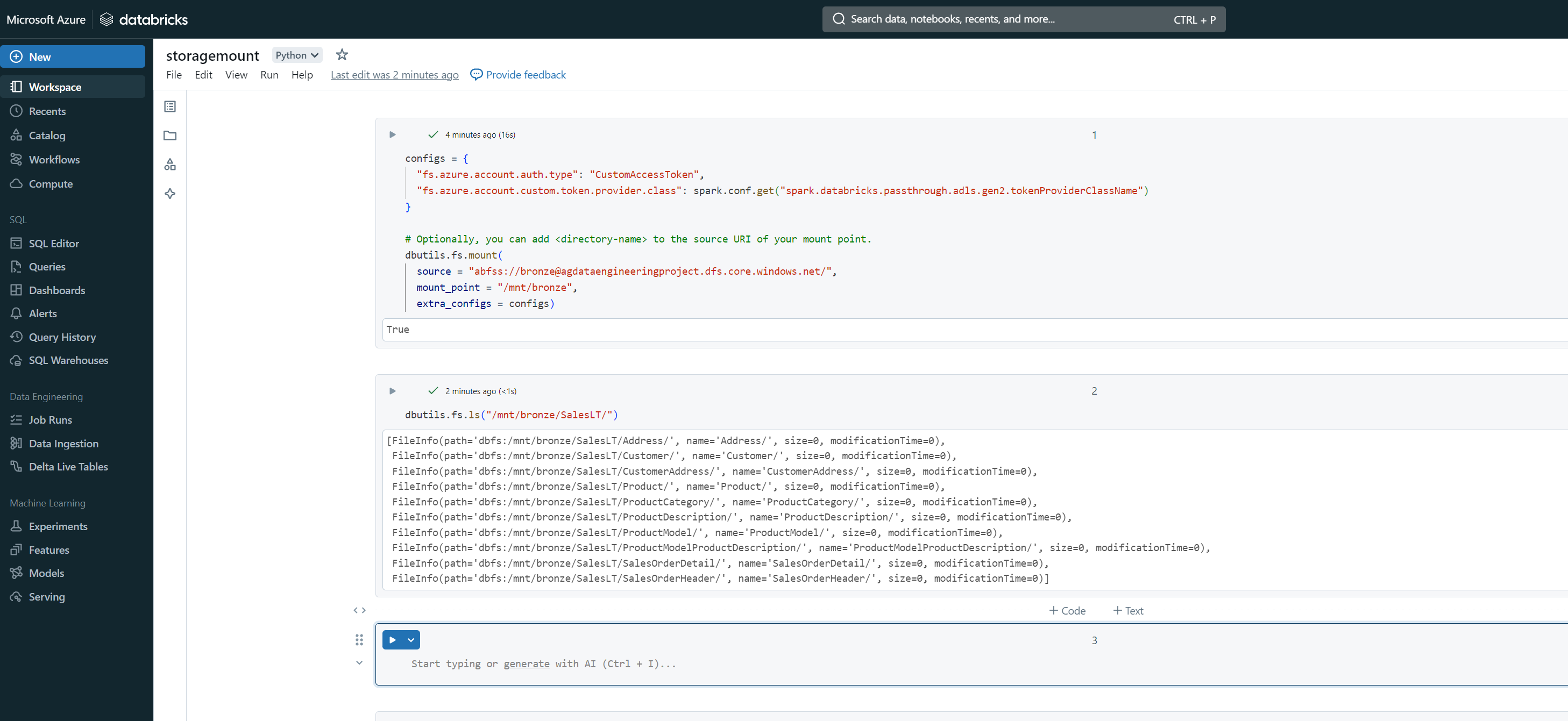
Publish all the changes🡪debug the pipeline and see if it runs successfully 🡪you can monitor the pipeline under monitor section.

After successful executing the pipeline, you will see all the tables in the container under bronze folder as per the directory.

**Creating Azure Databricks Workspace for Data transformation**

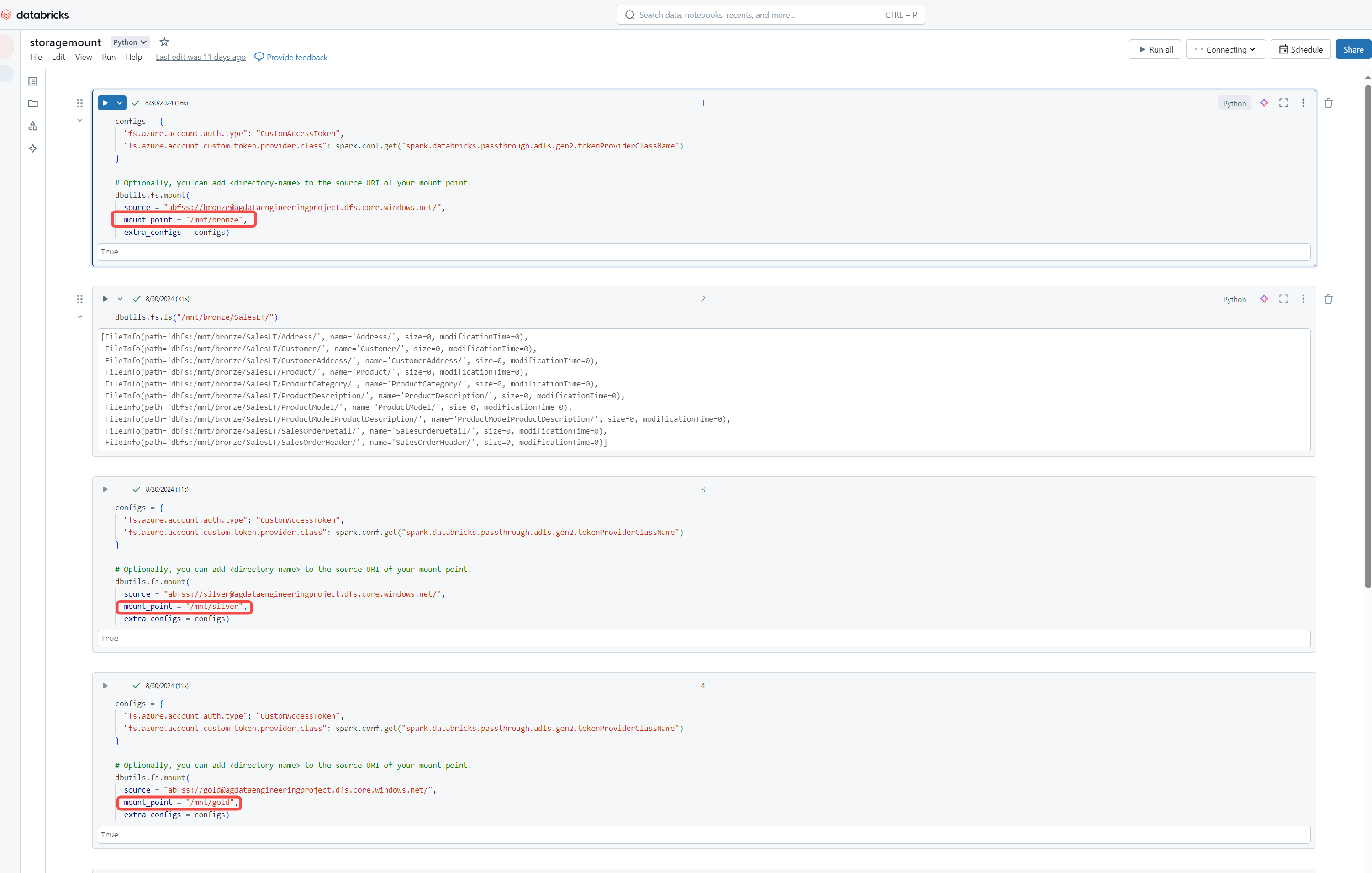
**First part: Mounting containers:**

* 1. Create databricks workspace🡪click on compute tab🡪create a cluster for compute (for now select single node, select timeout after 15 mins under advanced option)
  2. Now we will create a workbook to mount the container first

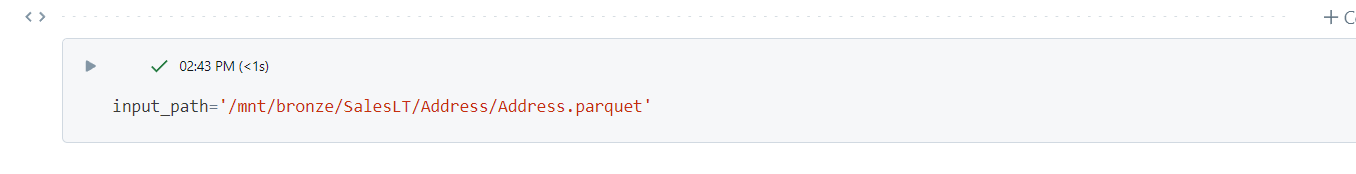


**Second part: Read data and do data transformation:**

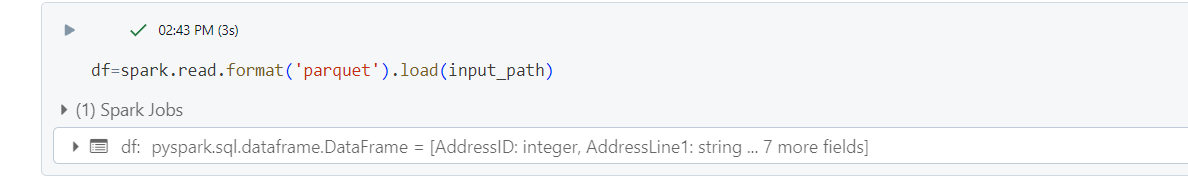
1. Create a new python workbook name it as “bronze to silver” for data transformation.
2. Mount bronze , silver and gold folder files



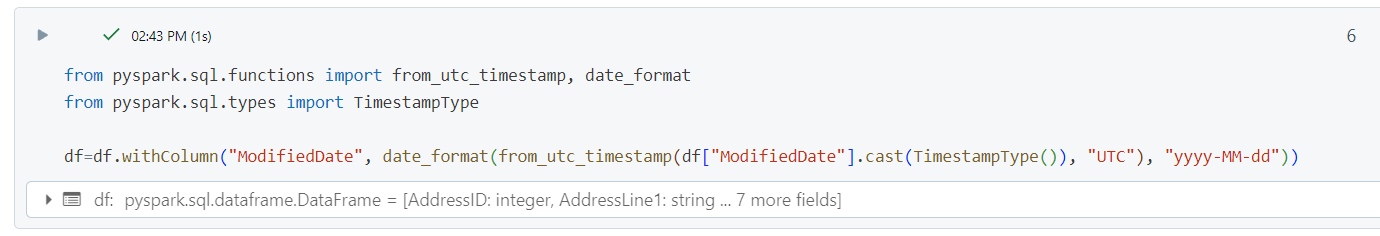
1. Setup input path for Address table from bronze folder



1. Create a dataframe to load data from address table 🡪display(df)



1. Write Pyspark code to transform timestamp data to date “yyyy-MM-dd” format for [ModifiedDate] column from address table



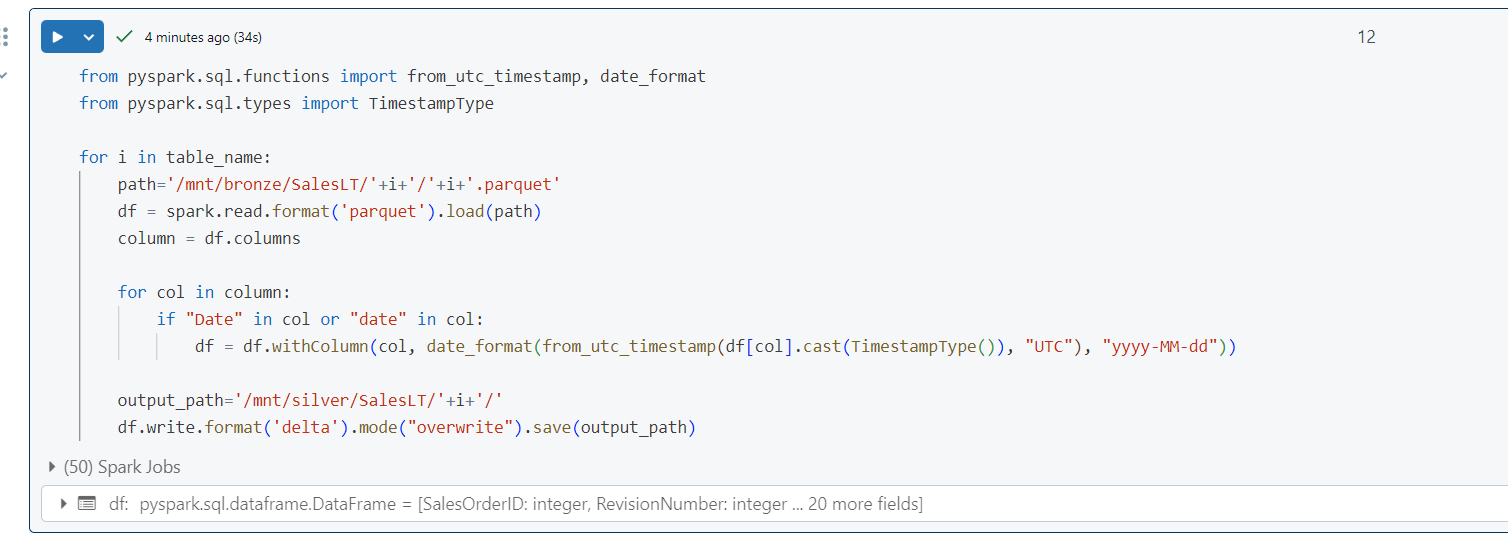
**Display(df)**

**Doing Transformation for all Tables (Bronze to silver)**

1. First, we need to loop through all the tables from the bronze folder



1. Now write down the transformation logic to transform timestamp data to date “yyyy-MM-dd” format for any column having timestamp format in the tables



**Doing Transformation for all Tables (Silver to Gold)-Level 2**

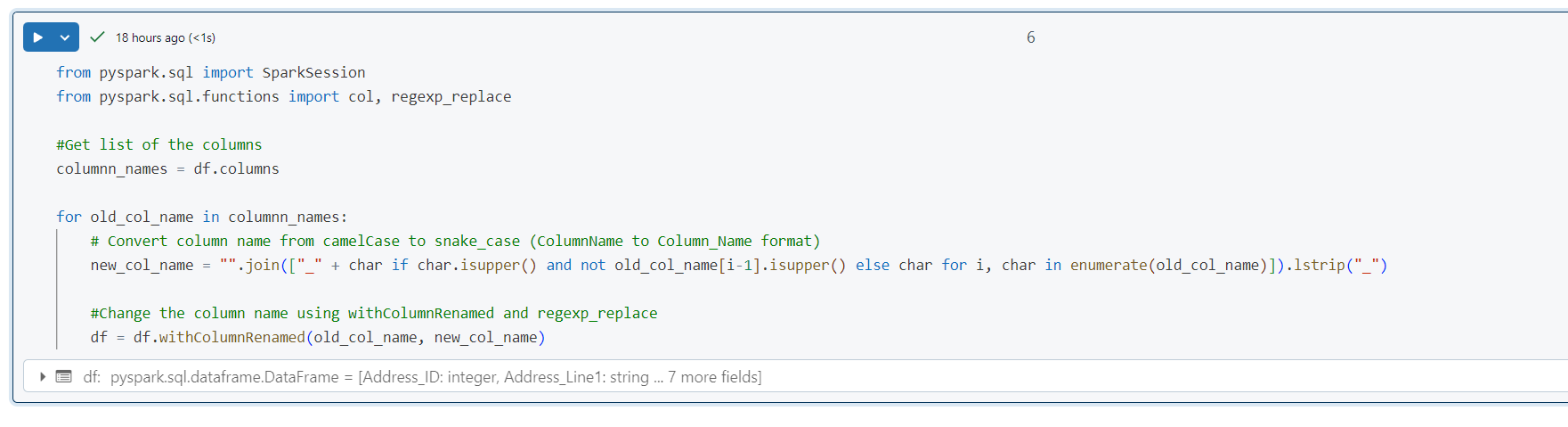
1. First, we have to mount files from the silver folder, also mount the Gold folder so that transformed data can be stored in it .



1. Define Input Path for silver folder and read the file.

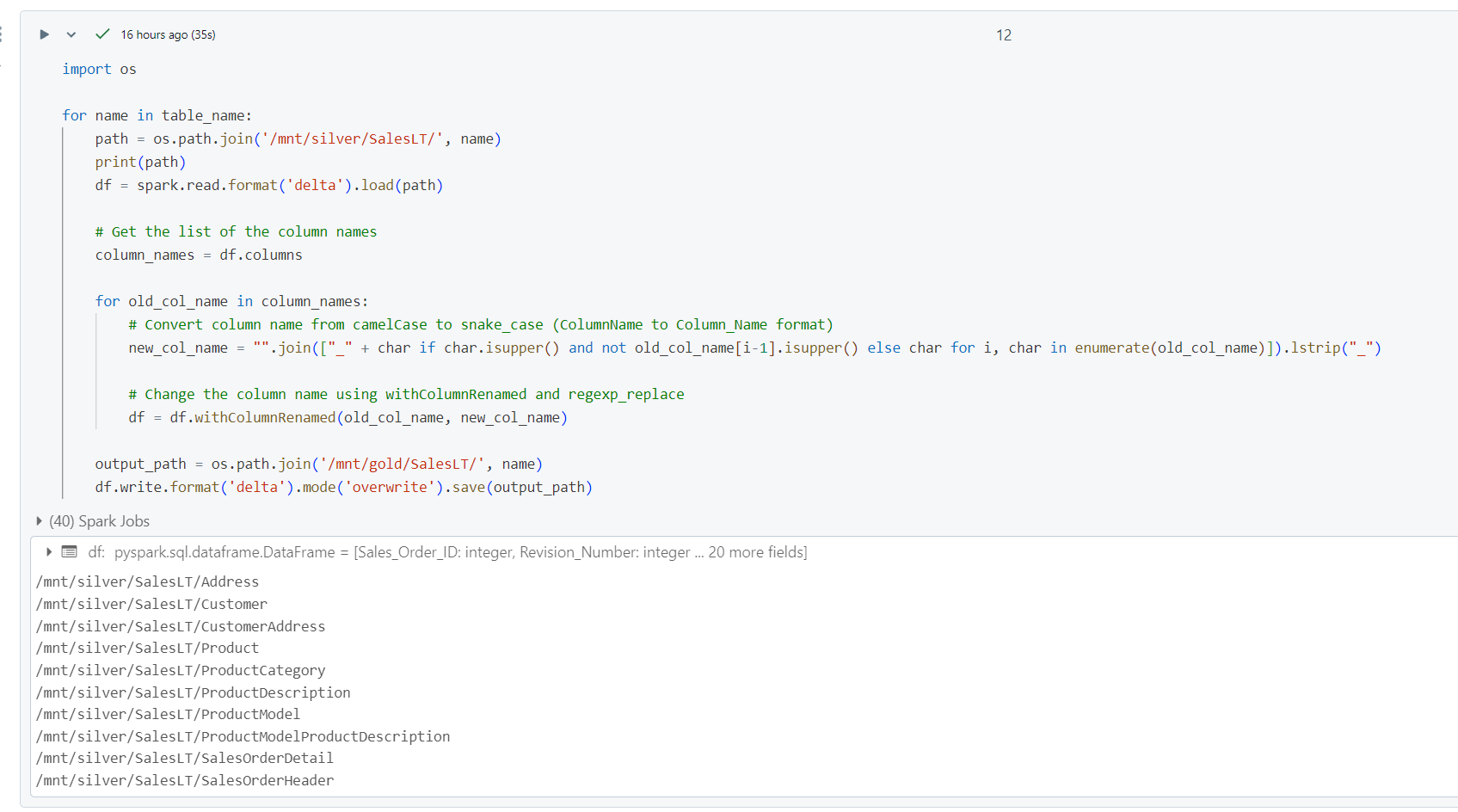


1. Do transformation 🡪 Get list of the columns🡪 Convert column name from camelCase to snake\_case (ColumnName to Column\_Name format)🡪 Change the column name using withColumnRenamed and regexp\_replace



1. Now we have to do the transformation for all the files/tables in silver folder.



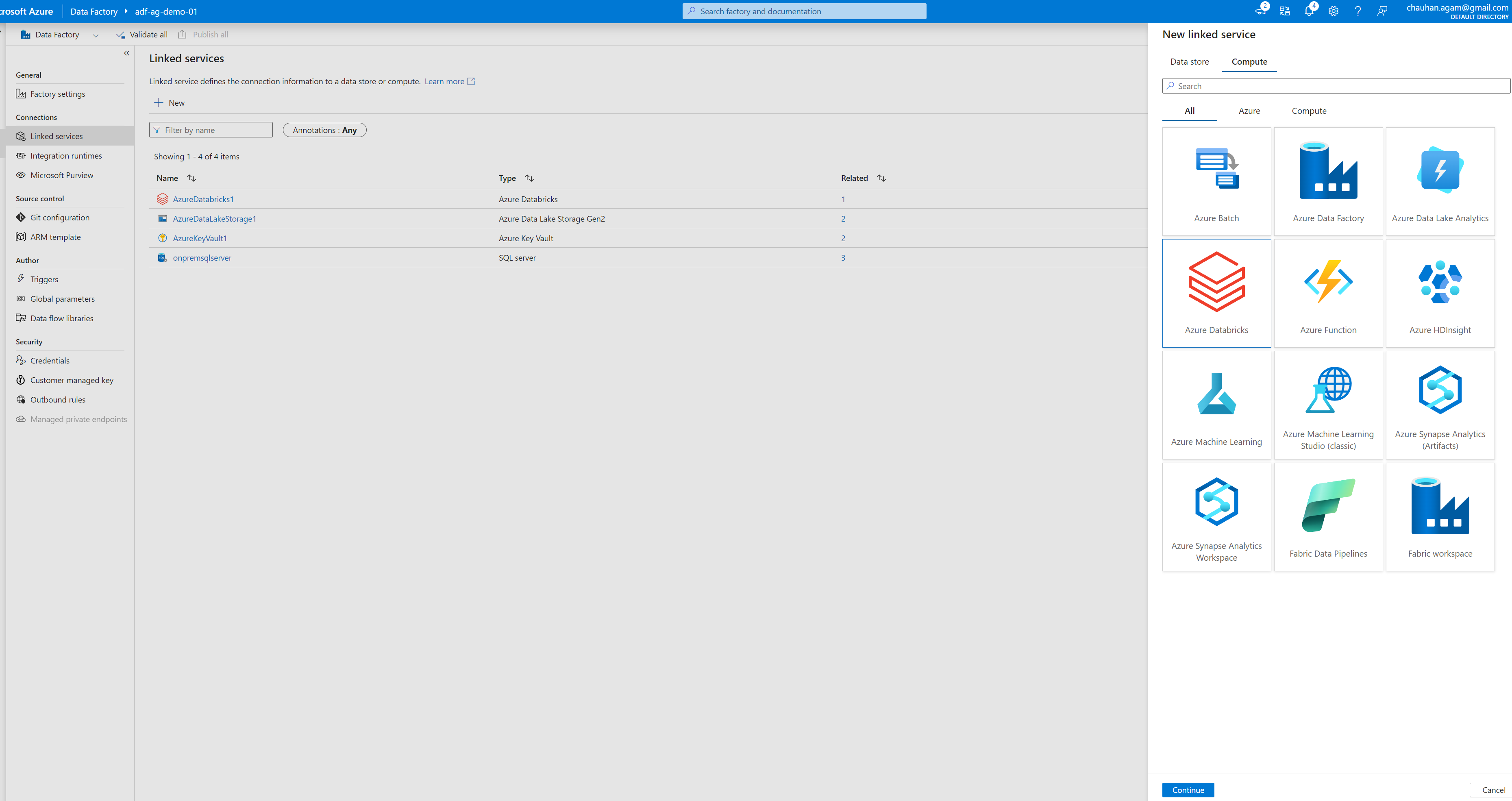


**Now we have completed the data transformation and we have 2 workbooks 🡪bronze to silver 🡪silver to gold and we have to run them as job , we also have another notebook storagemount which we only have to run once to mount the files**

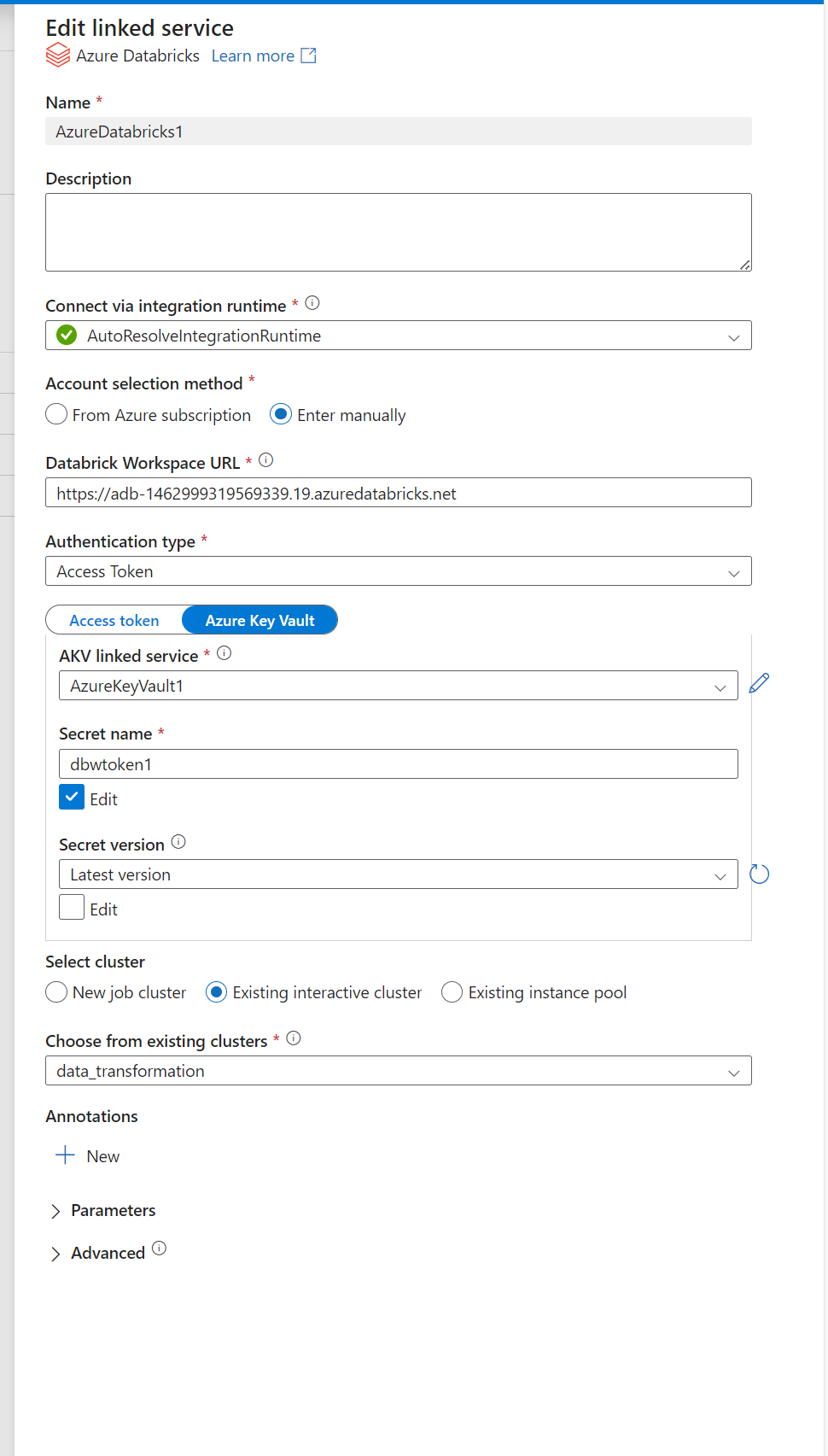
**We will use azure data pipelines.**

**Creating End to End data pipeline in ADF**

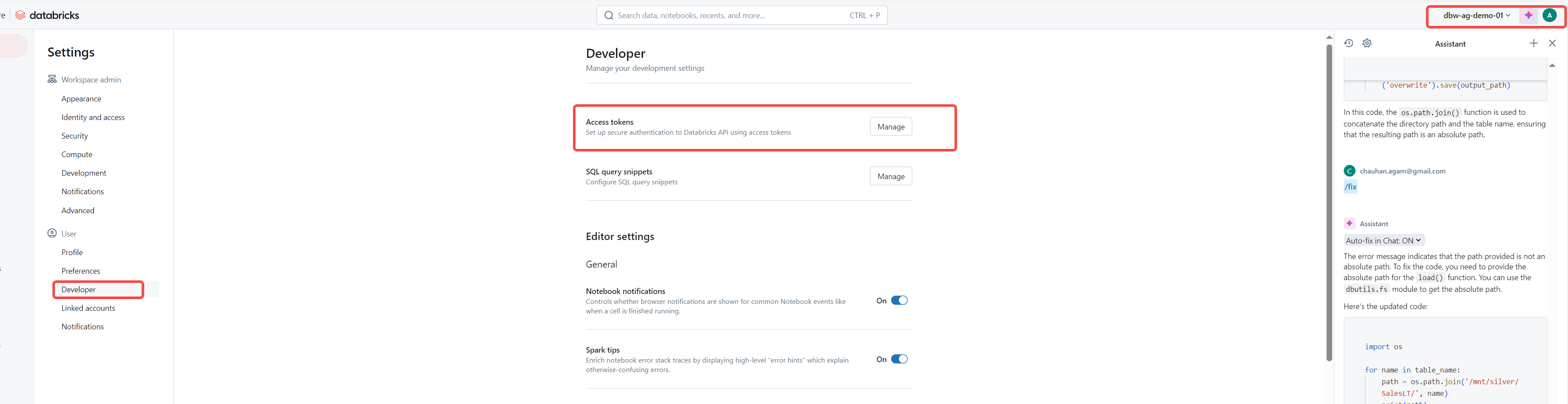
1. Go to azure data factory 🡪click on manage🡪 Click on create new 🡪select compute and search for azure databricks.



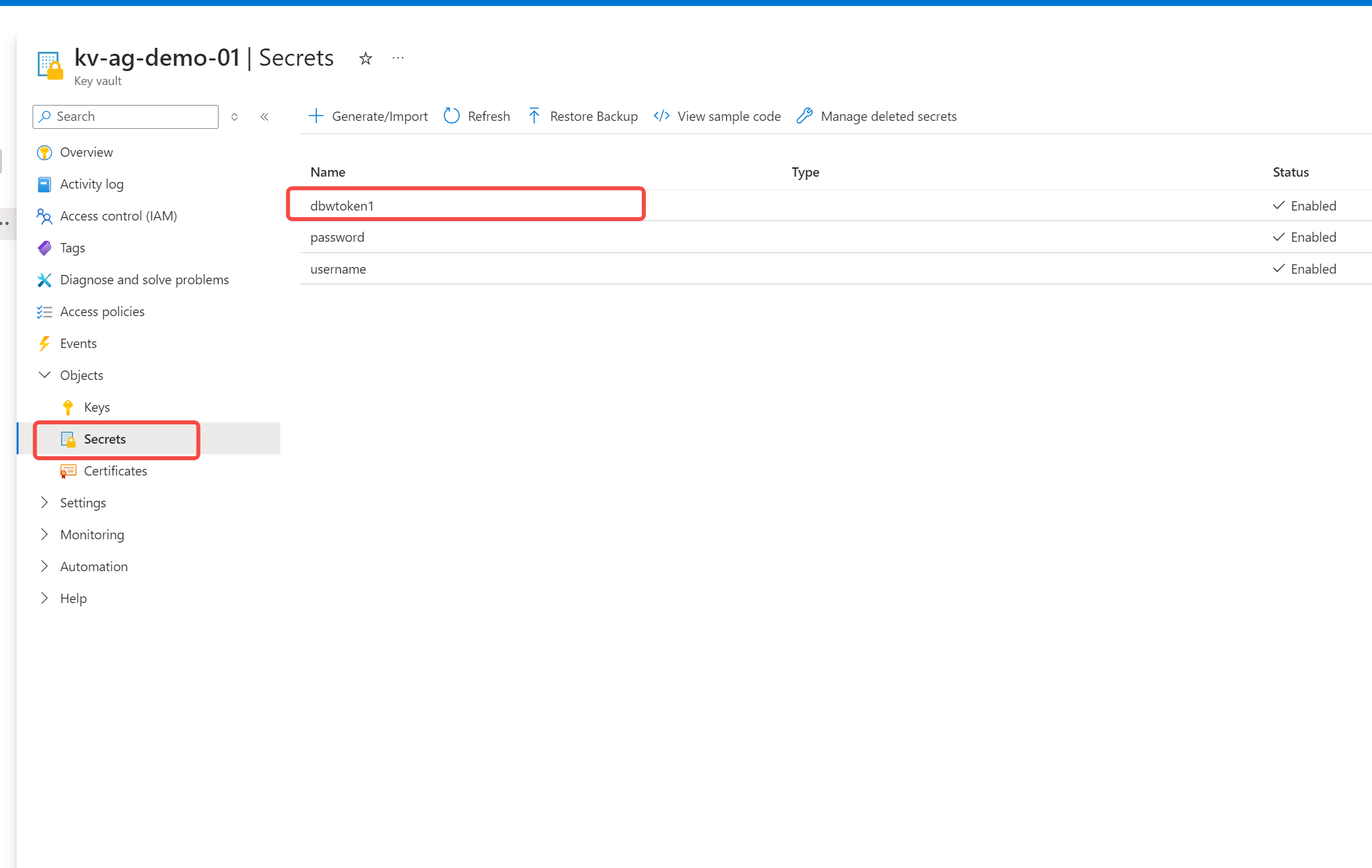
1. Now fill in the details🡪select authentication type as access token



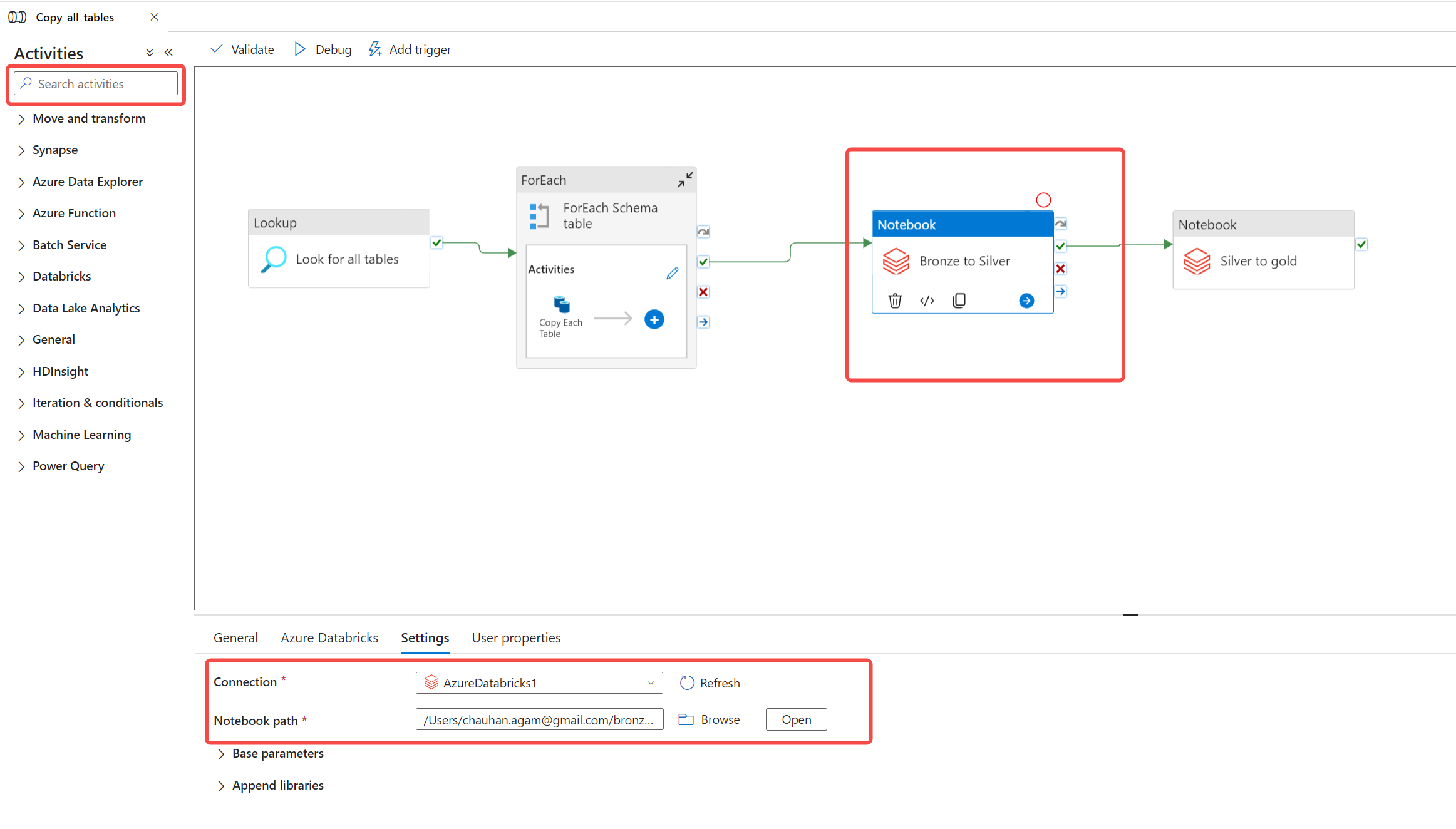
1. Now go to azure databricks and click on your name on right end top corner and select settings and then create access token🡪copy value



1. Now go to key valut 🡪create a secret key and copy the access token value



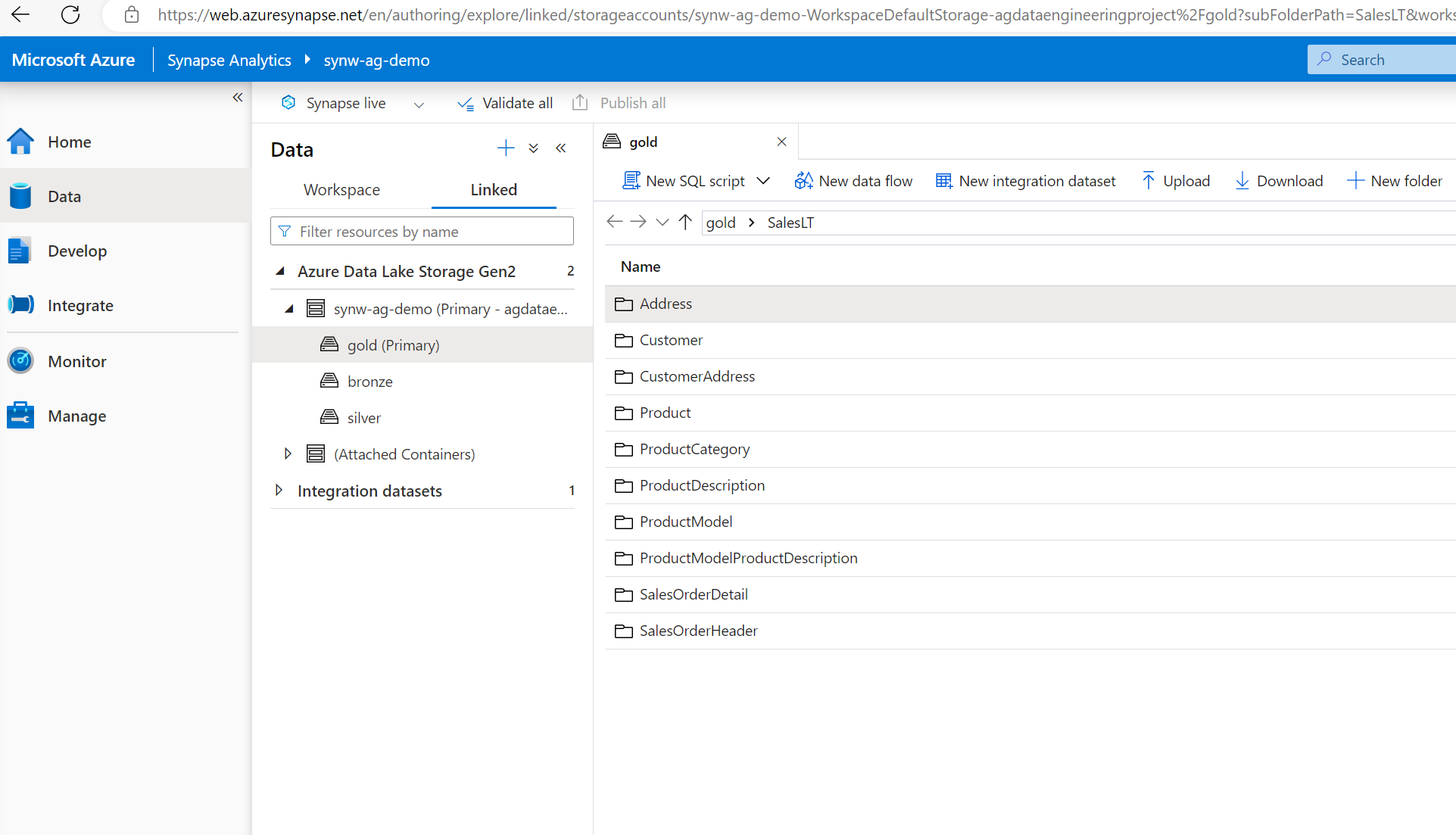
1. Now in step 2 select Azure key vault and select the recent created secret🡪choose existing cluster🡪test connection and create.
2. Now click on Author section in ADF and select the existing copy data pipeline (which was created earlier)
3. Search for notebook 🡪drag databricks notebook 🡪Define connection and notebook path



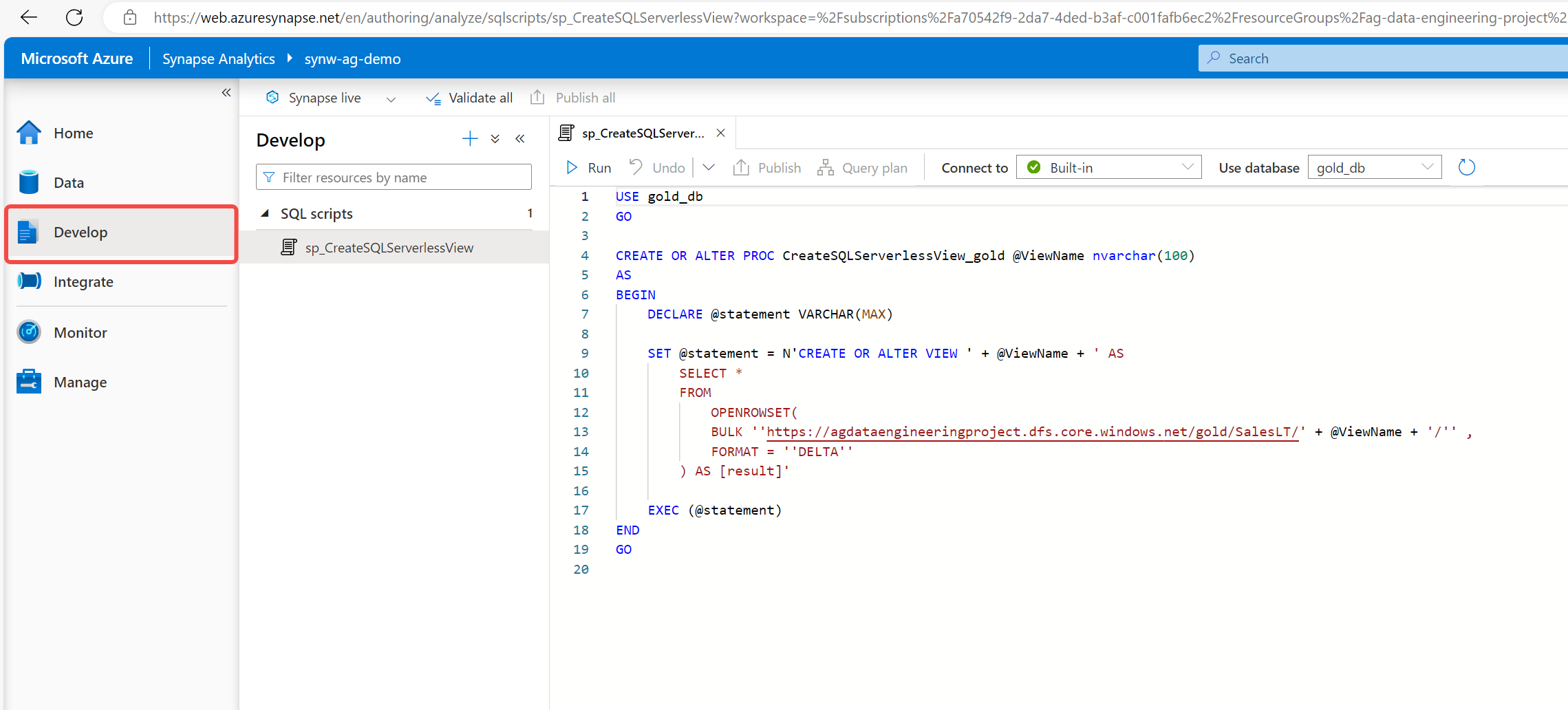
1. Configure another notebook connection for silver to gold notebook🡪publish the pipeline and run.

**Data loading using Azure Synapse Analytics**

1. Go to resource group and create synapse workspace 🡪open studio.
2. Go to Data section 🡪click on workspace🡪Now click on + icon and select SQL Database🡪Select Serverless and click create.

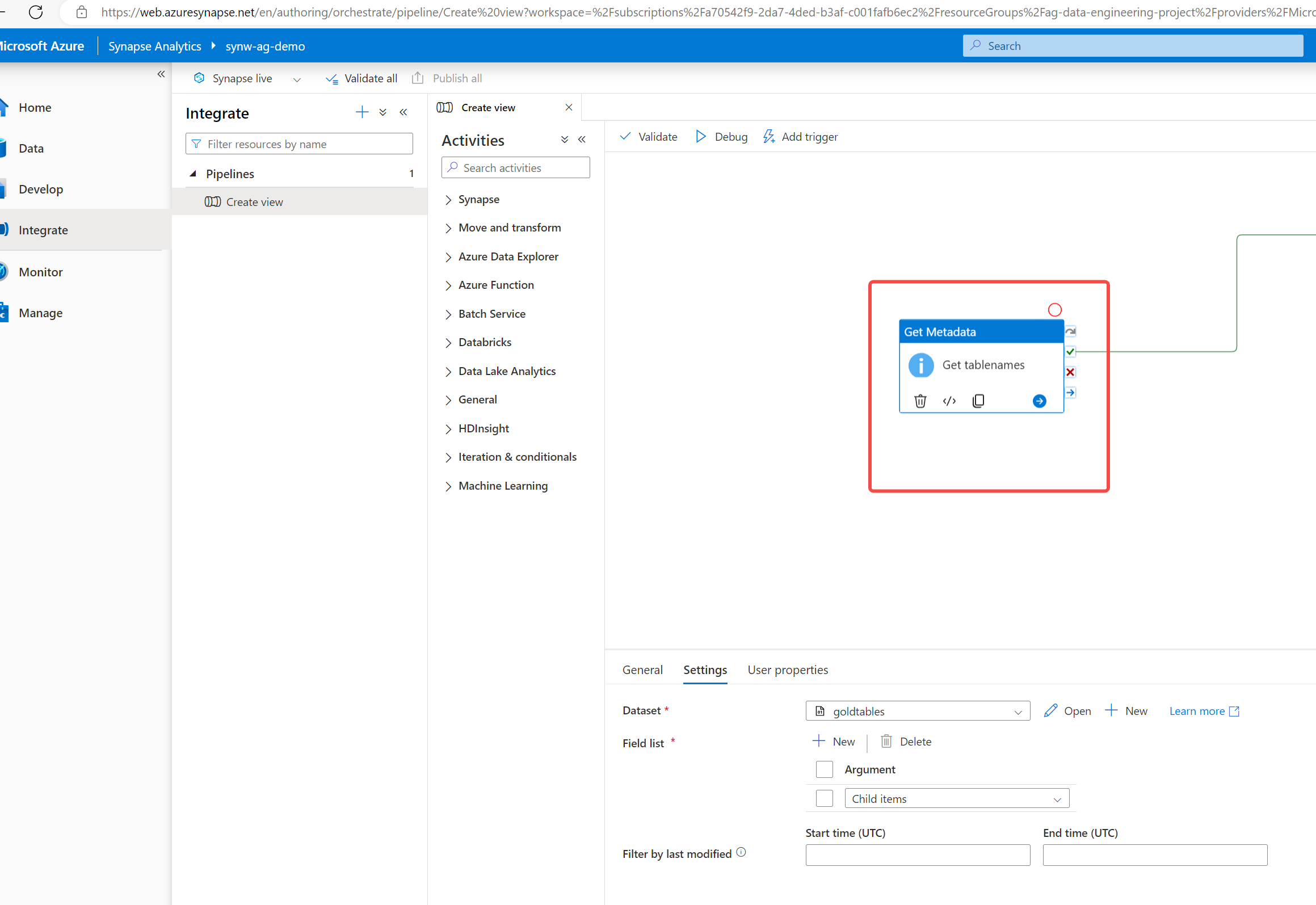


1. Now click on Linked option and you can see azure data lake is already linked with synapse workspace.
2. You can try querying data from data lake linked to synapse workspace.
3. Now we will create a view in serverless database using stored procedure for all the tables🡪Create below Sp and publish .

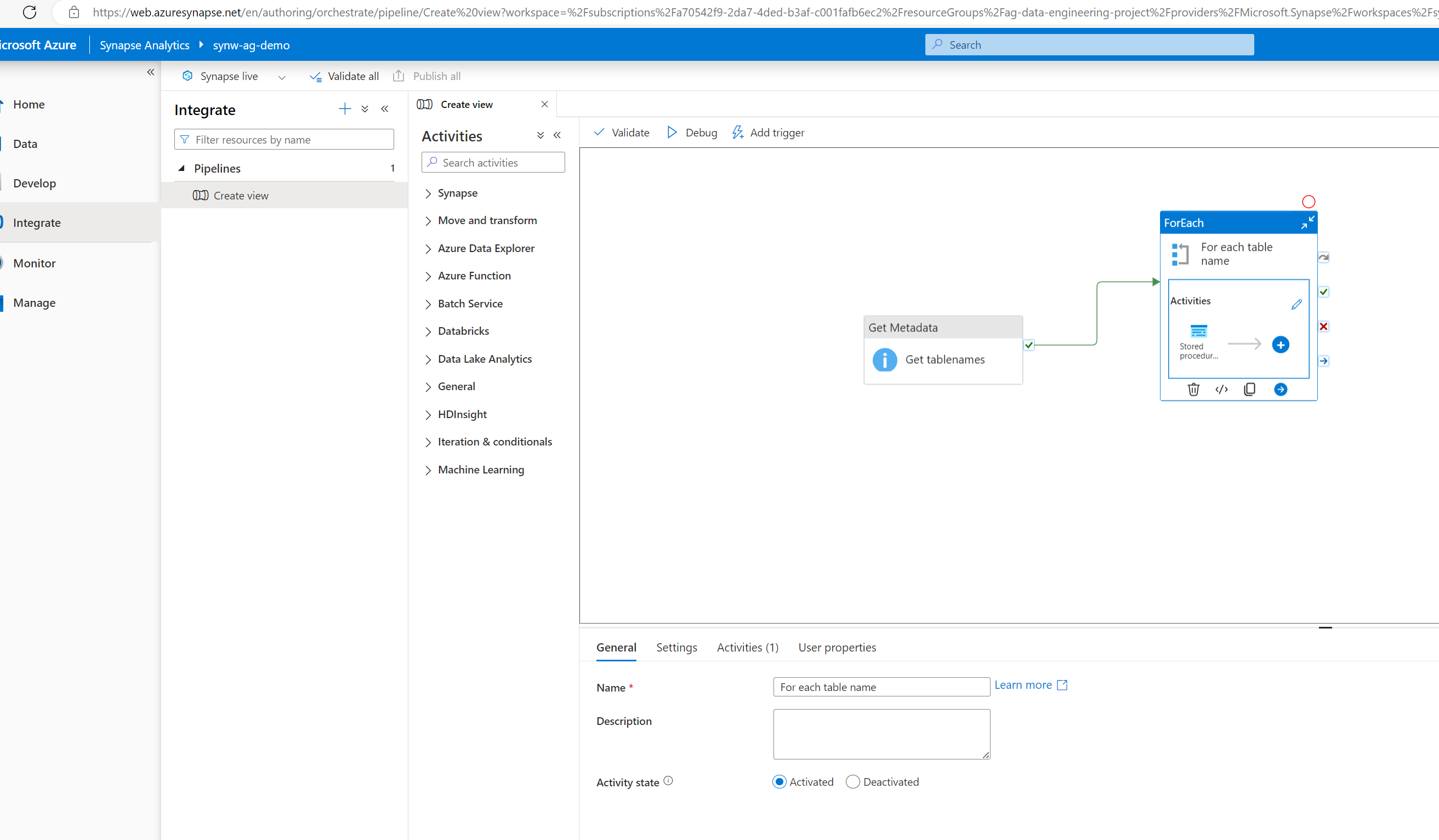




1. Now we will create a pipeline to create view for all the tables in the serverless database using above stored procedure🡪Create linked service connection🡪Go to Mange tab and select Linked Services🡪click on new select Azure SQL database🡪Select account selection method as Enter manually.
2. For fully qualified domain name 🡪go to azure synapse workspace 🡪click on Properties🡪Copy Serverless SQL endpoint and paste it in fully qualified domain name🡪Authentication type select System Assigned Managed Identify🡪 Test connection.
3. Go to Integrate Tab🡪Click on + and select pipeline🡪drag metadata activity.
4. Give name to metadata activity 🡪click on setting🡪click on new and select azure data lake Gen 2 and file type will be binary🡪 Select Linked service type –select gold folder path.
5. Configure field list 🡪click on new and select child items from dropdown.



1. Now after getting all the table names from the gold container, we need to iterate all the tables 🡪Drag for each activity🡪go to settings🡪Click on add dynamic content under items section🡪select get tablenames activity in then end add .childitems
2. Now click on edit button in foreach loop activity section🡪drag stored procedure in this activity 🡪go to setting🡪select stored procedure which we created earlier🡪click on Sp parameters🡪click on new🡪copy parameter from Sp (@viewName) and paste it in the Sp Parameter🡪datatype as string🡪click on add dynamic content under value 🡪 Select @item().name.



1. Publish all the changes and run the pipeline.

**Data Reporting using Power BI**

1. Open Power BI and click on get Data 🡪Select Azure Synapse Analytics🡪connect.
2. Copy Server details from Synapse workspace🡪click on Properties in synapse workspace🡪copy serverless SQL endpoint and paste it in powerBI🡪enter database name.
3. Use Microsoft account to connect the database source🡪 signIn🡪Load data.
4. Now you can create a interactive dashboard in PowerBi using the data which we loaded from synapse workspace.



**End to End Pipeline Testing**

1. Got to azure data factory and select the cpy\_all\_tables pipeline🡪click on Add trigger option 🡪Click on New/edit option.
2. Now we need to create a new trigger.
3. Schedule the trigger and then add 2 new customers to customer table.(just to check whether the end-to-end pipeline is working fine )