Project: Fintech SQL Data migration into Azure Portal

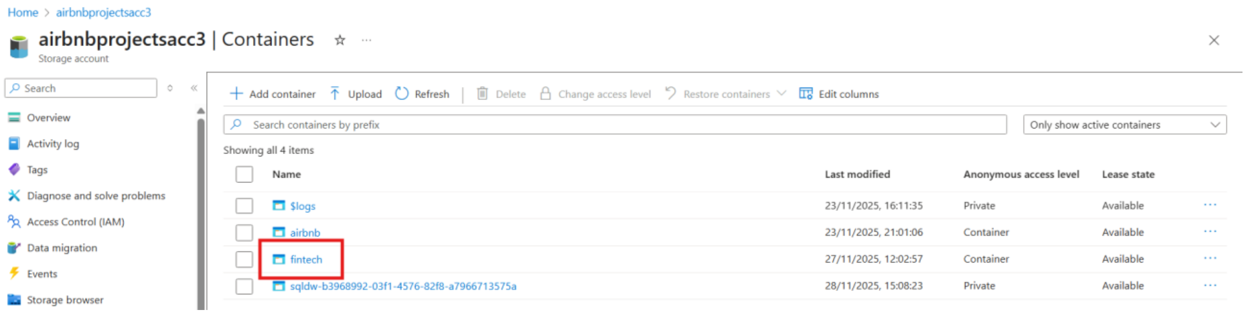
Objective:

Migrate an entire Fintech relational database (Accounts, Customers, Transactions, Loans, Payments, etc.) from Azure SQL DB → ADLS Gen2 using industry-standard Medallion Architecture (Bronze → Silver → Gold) with full automation and monitoring.

Firstly, I created

Storage Setup

- Created ADLS Gen2 container: **fintech**
- Organized into three layers: **bronze / silver / gold** (Medallion Architecture)



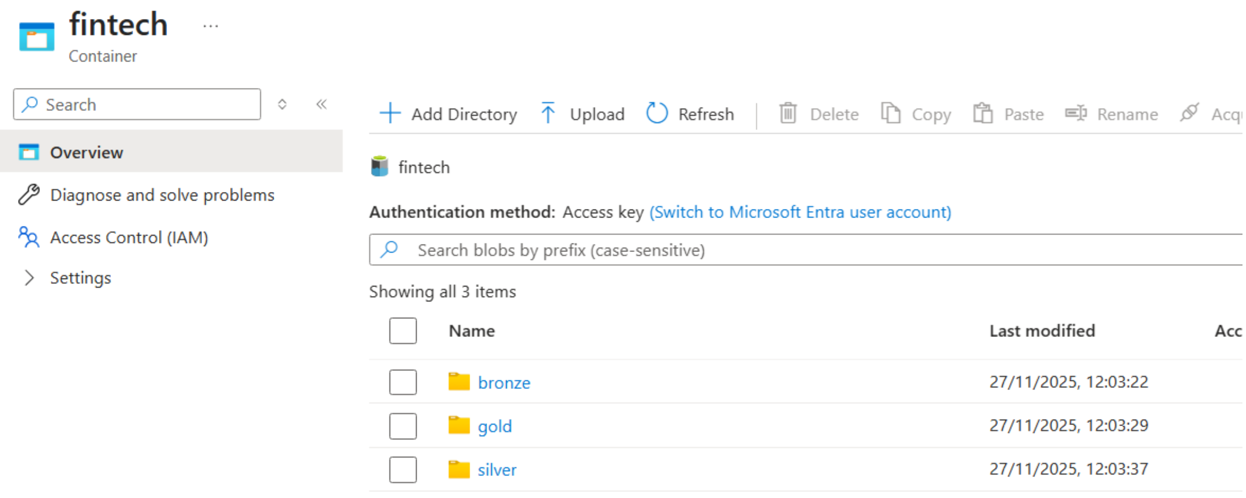
Home > airbnbprojectsacc3

airbnbprojectsacc3 | Containers

Storage account

Showing all 4 items

Name	Last modified	Anonymous access level	Lease state
$logs	23/11/2025, 16:11:35	Private	Available
airbnb	23/11/2025, 21:01:06	Container	Available
fintech	27/11/2025, 12:02:57	Container	Available
sqldw-b3968992-03f1-4576-82f8-a7966713575a	28/11/2025, 15:08:23	Private	Available



fintech

Container

fintech

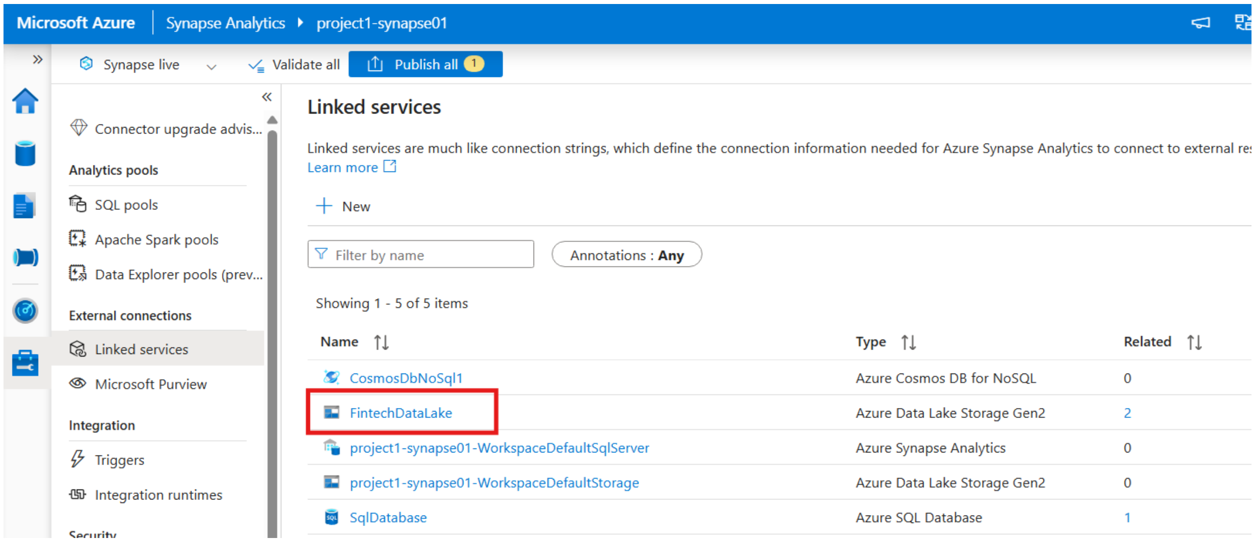
Authentication method: Access key (Switch to Microsoft Entra user account)

Search blobs by prefix (case-sensitive)

Showing all 3 items

Name	Last modified	Acc
bronze	27/11/2025, 12:03:22	
gold	27/11/2025, 12:03:29	
silver	27/11/2025, 12:03:37	

Then i created a linked service for azure data lake gen2 in synapse analytics



Linked services

Linked services are much like connection strings, which define the connection information needed for Azure Synapse Analytics to connect to external re
Learn more

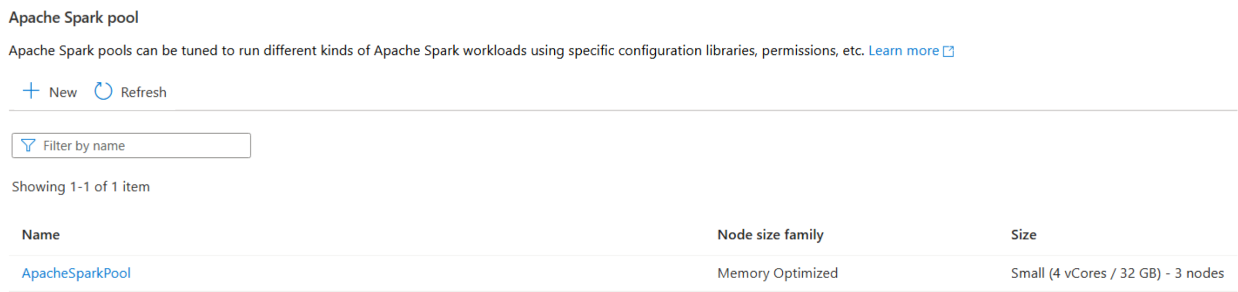
+ New

Filter by name | Annotations : **Any**

Showing 1 - 5 of 5 items

Name	Type	Related
CosmosDbNoSql1	Azure Cosmos DB for NoSQL	0
FintechDataLake	Azure Data Lake Storage Gen2	2
project1-synapse01-WorkspaceDefaultSqlServer	Azure Synapse Analytics	0
project1-synapse01-WorkspaceDefaultStorage	Azure Data Lake Storage Gen2	0
SqlDatabase	Azure SQL Database	1

Then I created Apache spark pool to run notebooks in synapse

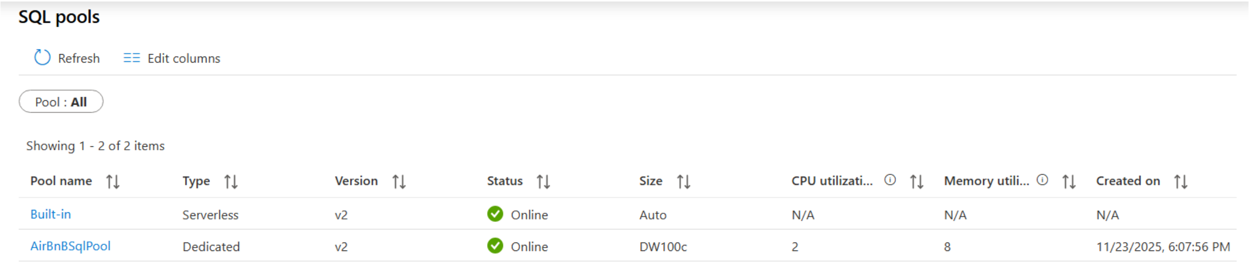

Apache Spark pool

Apache Spark pools can be tuned to run different kinds of Apache Spark workloads using specific configuration libraries, permissions, etc. Learn more

Showing 1-1 of 1 item

Name	Node size family	Size
ApacheSparkPool	Memory Optimized	Small (4 vCores / 32 GB) - 3 nodes

Then I resumed dedicated SQL pool which I already had created


SQL pools

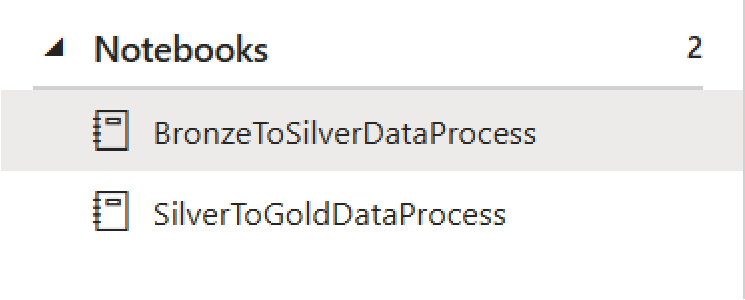
Showing 1 - 2 of 2 items

Pool name	Type	Version	Status	Size	CPU utilizati...	Memory utili...	Created on
Built-in	Serverless	v2	✓ Online	Auto	N/A	N/A	N/A
AirBnBSqlPool	Dedicated	v2	✓ Online	DW100c	2	8	11/23/2025, 6:07:56 PM

Then I created two notebooks in Synapse containing Spark code.

BronzeToSilverDataProcess notebook reads data from the Bronze layer, applies initial transformations, and writes the output to the Silver layer.

The second notebook performs additional transformations on the Silver layer and then loads the final curated output into the Gold layer.



BronzeTOSilverDataProcess:

```
from pyspark.sql.functions import *

# Define paths
base_path =
"abfss://fintech@airbnbprojectsacc3.dfs.core.windows.net/bronze/fintech/"
output_base_path =
"abfss://fintech@airbnbprojectsacc3.dfs.core.windows.net/silver/fintech/"

spark.conf.set("spark.databricks.delta.schema.autoMerge.enabled", "true")

# Transformation for Accounts dataset
def transform_accounts():
    df = spark.read.parquet(f"{base_path}Accounts/Accounts.parquet")
    # Example transformation: Calculate account age in years
    df_transformed = df.withColumn("AccountAgeYears",
                                   round(datediff(current_date(),
col("OpenDate")) / 365.25, 2))

df_transformed.write.format("delta").mode("overwrite").save(f"{output_base_path}A
ccounts/")

# Transformation for Customers dataset
def transform_customers():
    df = spark.read.parquet(f"{base_path}Customers/Customers.parquet")
    # Example transformation: Create a full name column and mask the email
address
    df_transformed = df.withColumn("FullName", concat_ws(" ", col("FirstName"),
col("LastName"))) \
                       .withColumn("MaskedEmail",
                                   concat(lit("***@"),
substring_index(col("Email"), "@", -1)))
```

```
df_transformed.write.format("delta").mode("overwrite").save(f"{output_base_path}C
ustomers/")

# Transformation for Loans dataset with explicit casting
def transform_loans():
    df = spark.read.parquet(f"{base_path}Loans/Loans.parquet")
    # Example transformation: Calculate total interest with explicit casting to
match the Delta table
    df_transformed = df.withColumn("TotalInterest",
                                   (col("LoanAmount") * col("InterestRate") /
100).cast("decimal(28,8)")) \
                       .withColumn("LoanDurationYears",
                                   round(datediff(col("LoanEndDate"),
col("LoanStartDate")) / 365.25, 2))

df_transformed.write.format("delta").mode("overwrite").save(f"{output_base_path}L
oans/")

# Transformation for Payments dataset
def transform_payments():
    df = spark.read.parquet(f"{base_path}Payments/Payments.parquet")
    # Example transformation: Calculate days since last payment
    df_transformed = df.withColumn("DaysSinceLastPayment",
                                   datediff(current_date(), col("PaymentDate")))

df_transformed.write.format("delta").mode("overwrite").save(f"{output_base_path}P
ayments/")

# Transformation for Transactions dataset
def transform_transactions():
    df = spark.read.parquet(f"{base_path}Transactions/Transactions.parquet")
    # Example transformation: Categorize transaction types
    df_transformed = df.withColumn("TransactionCategory",
                                   when(col("TransactionType") == "Deposit",
"Income")
                                   .when(col("TransactionType") == "Withdrawal",
"Expense")
                                   .otherwise("Other"))

df_transformed.write.format("delta").mode("overwrite").save(f"{output_base_path}T
ransactions/")

# Process each table
transform_accounts()
transform_customers()
```

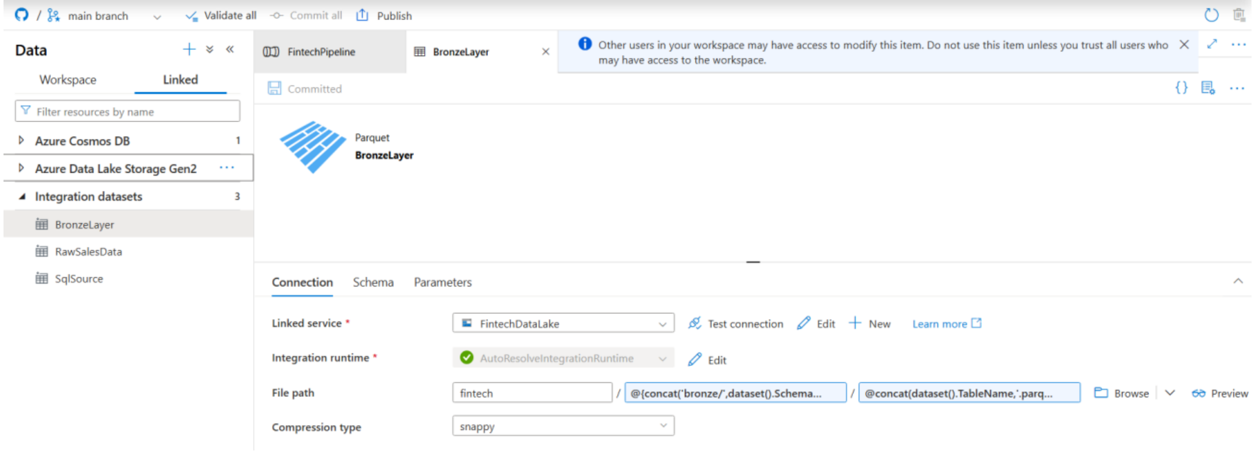
```
transform_loans()
transform_payments()
transform_transactions()

print("Bronze To Silver Completed !!")
```

SilverToGoldDataProcess:

```
from pyspark.sql.functions import *

# Define paths
silver_base_path =
"abfss://fintech@airbnbprojectsacc3.dfs.core.windows.net/silver/fintech/"
output_base_path =
"abfss://fintech@airbnbprojectsacc3.dfs.core.windows.net/gold/fintech/"

# Load data from the silver layer
accounts_df = spark.read.format("delta").load(f"{silver_base_path}Accounts/")
customers_df = spark.read.format("delta").load(f"{silver_base_path}Customers/")
loans_df = spark.read.format("delta").load(f"{silver_base_path}Loans/")
payments_df = spark.read.format("delta").load(f"{silver_base_path}Payments/")
transactions_df =
spark.read.format("delta").load(f"{silver_base_path}Transactions/")


dim_customers_df = customers_df.select(
    col("CustomerID").alias("customer_id"),
    col("FirstName").alias("first_name"),
    col("LastName").alias("last_name"),
    col("Email").alias("email"),
    col("PhoneNumber").alias("phone_number"),
    col("Address").alias("address"),
    col("City").alias("city"),
    col("State").alias("state"),
    col("Country").alias("country"),
    col("ZipCode").alias("zip_code"),
    col("SignupDate").alias("signup_date")
)

dim_customers_df.write.format("delta").mode("overwrite").save(f"{output_base_path
}dim_customers/")
```

```
dim_accounts_df = accounts_df.select(
    col("AccountID").alias("account_id"),
    col("AccountType").alias("account_type"),
    col("Balance").alias("balance"),
    col("OpenDate").alias("open_date"),
    col("AccountAgeYears").alias("account_age_years")
)

dim_accounts_df.write.format("delta").mode("overwrite").save(f"{output_base_path}dim_accounts/")



dim_loans_df = loans_df.select(
    col("LoanID").alias("loan_id"),
    col("LoanType").alias("loan_type"),
    col("LoanAmount").alias("loan_amount"),
    col("InterestRate").alias("interest_rate"),
    col("LoanStartDate").alias("loan_start_date"),
    col("LoanEndDate").alias("loan_end_date"),
    col("TotalInterest").alias("total_interest"),
    col("LoanDurationYears").alias("loan_duration_years")
)

dim_loans_df.write.format("delta").mode("overwrite").save(f"{output_base_path}dim_loans/")



fact_payments_df = payments_df \
    .join(loans_df.select("LoanID", "CustomerID"), "LoanID") \
    .select(
        col("PaymentID").alias("payment_id"),
        col("LoanID").alias("loan_id"),
        col("CustomerID").alias("customer_id"),
        col("PaymentDate").alias("payment_date"),
        col("PaymentAmount").alias("payment_amount"),
        col("PaymentMethod").alias("payment_method")
    )

fact_payments_df.write.format("delta").mode("overwrite").save(f"{output_base_path}fact_payments/")



fact_transactions_df = transactions_df \
    .join(accounts_df.select("AccountID", "CustomerID"), "AccountID") \
    .select(
        col("TransactionID").alias("transaction_id"),
        col("AccountID").alias("account_id"),
        col("CustomerID").alias("customer_id"),
        col("TransactionDate").alias("transaction_date"),
        col("Amount").alias("amount"),
        col("TransactionType").alias("transaction_type"),
        col("Description").alias("description")
    )

fact_transactions_df.write.format("delta").mode("overwrite").save(f"{output_base_path}fact_transactions/")
```

Now I created Integrated Datasets



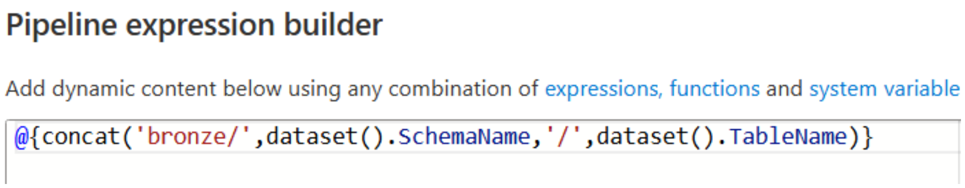
Other users in your workspace may have access to modify this item. Do not use this item unless you trust all users who may have access to the workspace.

Linked service *: FintechDataLake

Integration runtime *: AutoResolveIntegrationRuntime

File path: fintech / @{concat('bronze/',dataset().Schema... / @concat(dataset().TableName,'.parq...

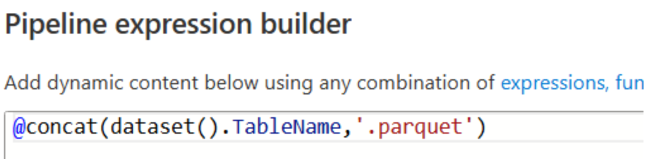
Compression type: snappy



Pipeline expression builder

Add dynamic content below using any combination of expressions, functions and system variable

```
@{concat('bronze/',dataset().SchemaName,'/',dataset().TableName)}
```

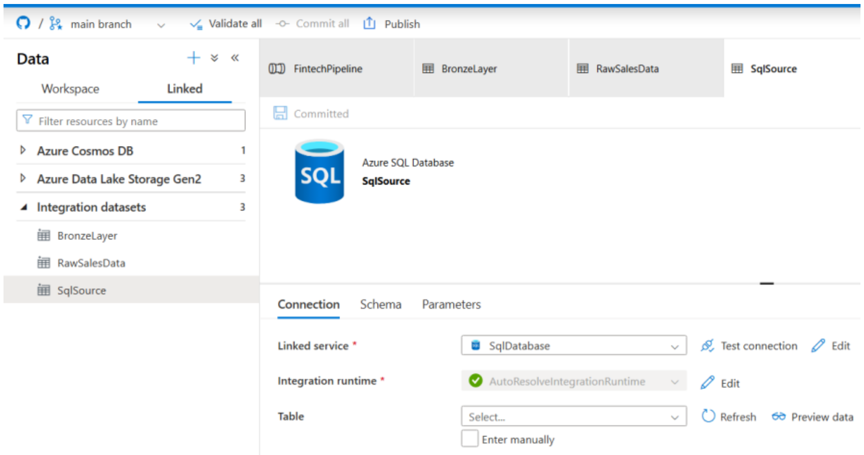


Pipeline expression builder

Add dynamic content below using any combination of expressions, fun

```
@concat(dataset().TableName,'.parquet')
```

Table field is empty to fetch all tables not a particular one



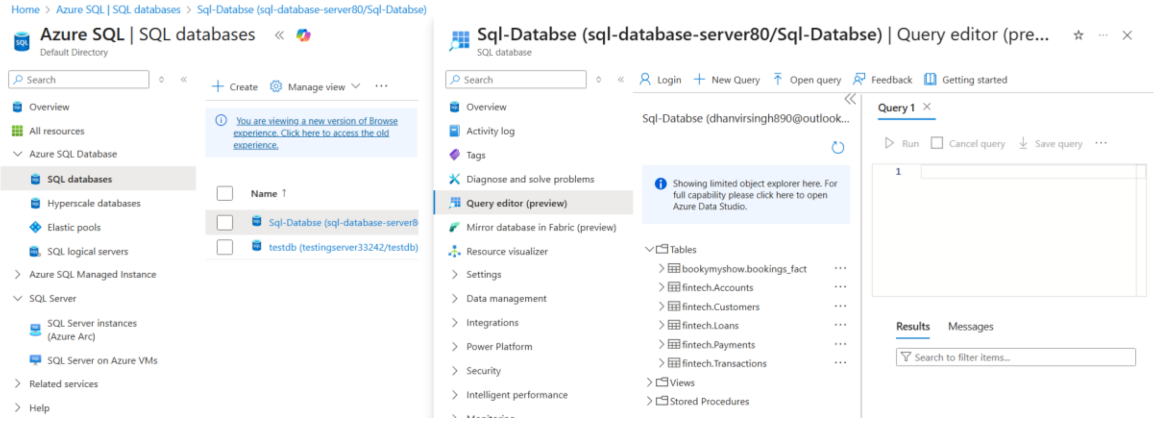
Linked service * SqlDatabase

Integration runtime * AutoResolveIntegrationRuntime

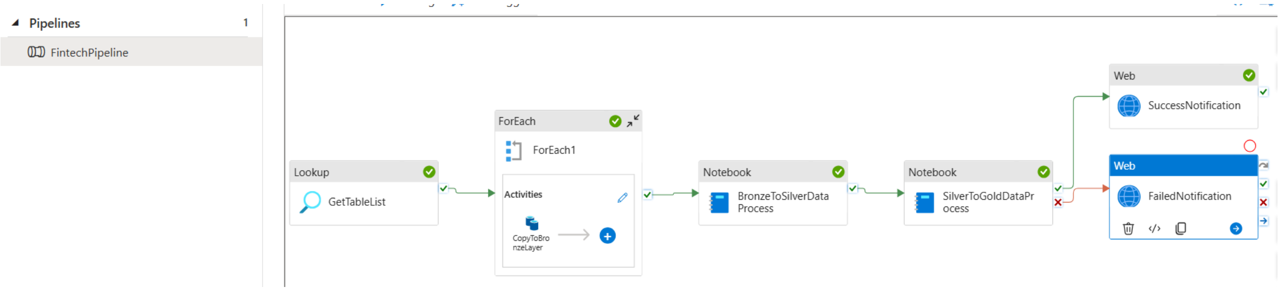
Table Select...

☐ Enter manually

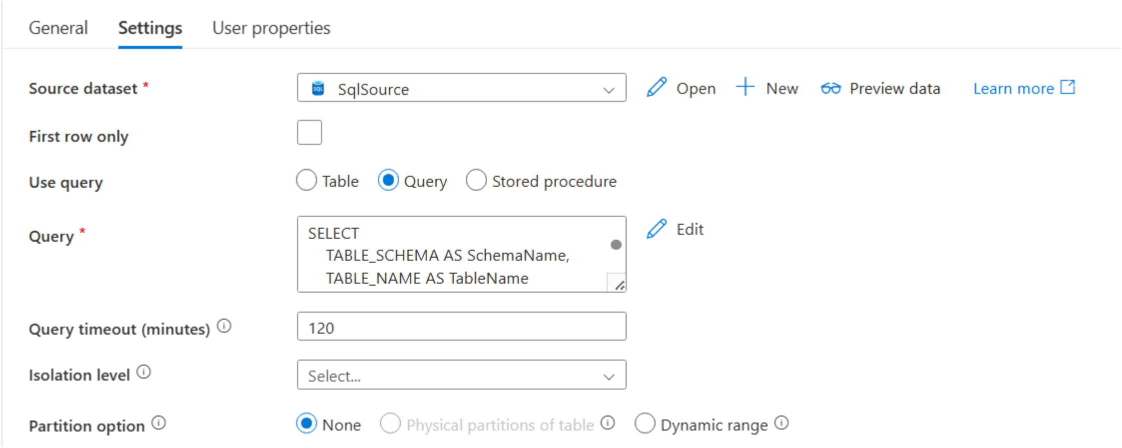
Then i created Azure Sql Database and also created azure sql server


Home > Azure SQL | SQL databases > Sql-Databse (sql-database-server80/Sql-Databse)
Azure SQL | SQL databases
Default Directory
Search
Create
Manage view
Overview
All resources
Azure SQL Database
SQL databases
Hyperscale databases
Elastic pools
SQL logical servers
Azure SQL Managed Instance
SQL Server
SQL Server instances (Azure Arc)
SQL Server on Azure VMs
Related services
Help
You are viewing a new version of Browse experience. Click here to access the old experience.
Name ↑
Sql-Databse (sql-database-server8
testdb (testingserver33242/testdb)
Sql-Databse (sql-database-server80/Sql-Databse) | Query editor (pre...
SQL database
Search
Overview
Activity log
Tags
Diagnose and solve problems
Query editor (preview)
Mirror database in Fabric (preview)
Resource visualizer
Settings
Data management
Integrations
Power Platform
Security
Intelligent performance
Login
New Query
Open query
Feedback
Getting started
Sql-Databse (dhanvirsingh890@outlook...
Showing limited object explorer here. For full capability please click here to open Azure Data Studio.
Tables
bookymyshow.bookings_fact
fintech.Accounts
fintech.Customers
fintech.Loans
fintech.Payments
fintech.Transactions
Views
Stored Procedures
Query 1
Run
Cancel query
Save query
Results
Messages
Search to filter items...

Now i created a pipeline in azure synapse analytics



Firstly, i choose lookup activity under it i choose source dataset sqlsource



General | **Settings** | User properties

Source dataset * : SqlSource — Open | New | Preview data | Learn more

First row only: □

Use query: ○ Table ● Query ○ Stored procedure

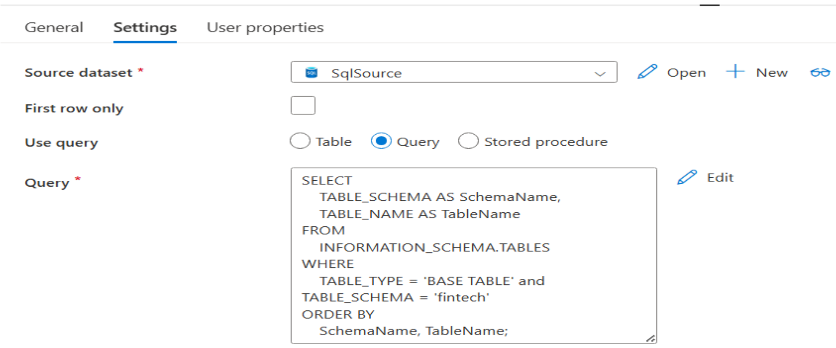
Query * : Edit

```
SELECT
   TABLE_SCHEMA AS SchemaName,
   TABLE_NAME AS TableName
```

Query timeout (minutes): 120

Isolation level: Select...

Partition option: ● None ○ Physical partitions of table ○ Dynamic range



General | **Settings** | User properties

Source dataset * : SqlSource — Open | New

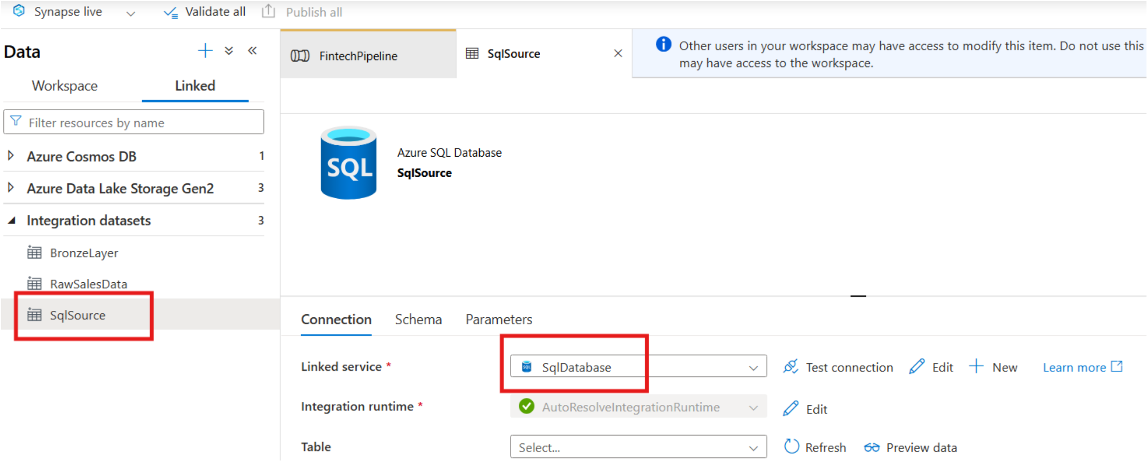
First row only: □

Use query: ○ Table ● Query ○ Stored procedure

Query * : Edit

```
SELECT
   TABLE_SCHEMA AS SchemaName,
   TABLE_NAME AS TableName
FROM
   INFORMATION_SCHEMA.TABLES
WHERE
   TABLE_TYPE = 'BASE TABLE' and
TABLE_SCHEMA = 'fintech'
ORDER BY
   SchemaName, TableName;
```

I created SqlSource under integrated dataset during creating it I choose SQL Database as data source and connected with SqlDatabase which is linked service I created in synapse to link connection with synapse to azure SQL database



Other users in your workspace may have access to modify this item. Do not use this may have access to the workspace.

Azure SQL Database
SqlSource

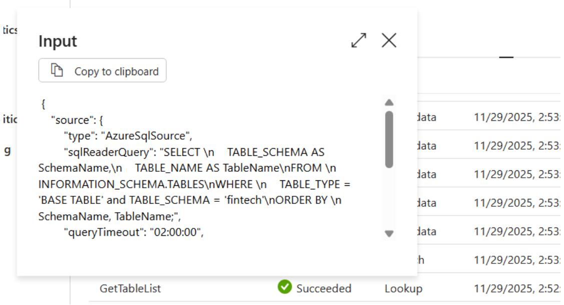
Connection | Schema | Parameters

Linked service *: SqlDatabase

Integration runtime *: AutoResolveIntegrationRuntime

Table: Select...

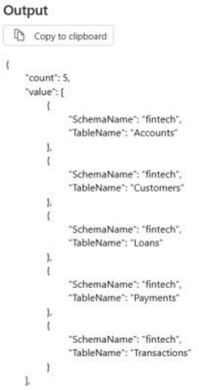
This is input and output to gettable list



Input

```
{
    "source": {
        "type": "AzureSqlSource",
        "sqlReaderQuery": "SELECT \n    TABLE_SCHEMA AS
SchemaName,\n    TABLE_NAME AS TableName\nFROM \n
INFORMATION_SCHEMA.TABLES\nWHERE \n    TABLE_TYPE =
'BASE TABLE' and TABLE_SCHEMA = 'fintech'\nORDER BY \n
SchemaName, TableName;",
        "queryTimeout": "02:00:00",
```

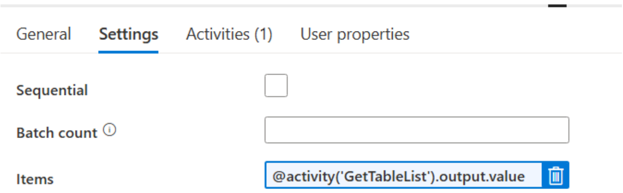
		data	11/29/2025, 2:53
		data	11/29/2025, 2:53
		data	11/29/2025, 2:53
		data	11/29/2025, 2:53
		data	11/29/2025, 2:53
		h	11/29/2025, 2:53
GetTableList	✓ Succeeded	Lookup	11/29/2025, 2:52



Output

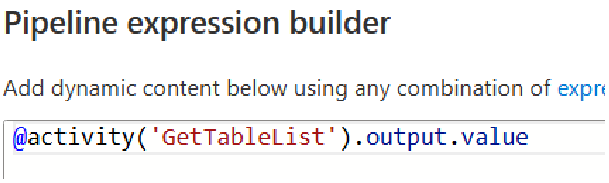
```
{
    "count": 5,
    "value": [
        {
            "SchemaName": "fintech",
            "TableName": "Accounts"
        },
        {
            "SchemaName": "fintech",
            "TableName": "Customers"
        },
        {
            "SchemaName": "fintech",
            "TableName": "Loans"
        },
        {
            "SchemaName": "fintech",
            "TableName": "Payments"
        },
        {
            "SchemaName": "fintech",
            "TableName": "Transactions"
        }
    ],
```

Under foreach activity i mentioned



General | **Settings** | Activities (1) | User properties

Sequential	□
Batch count ⓘ	
Items	@activity('GetTableList').output.value

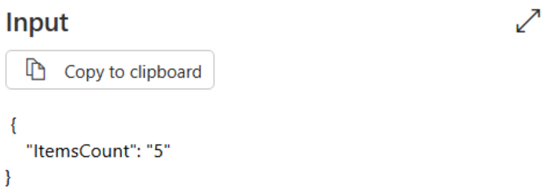


Pipeline expression builder

Add dynamic content below using any combination of expre

```
@activity('GetTableList').output.value
```

This will fetch output of get table list where all the names of files are present and will fetch one by one each file name



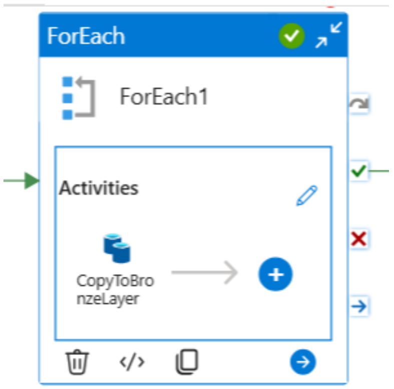
Input

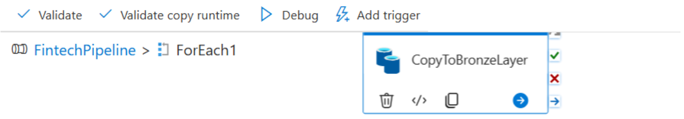
Copy to clipboard

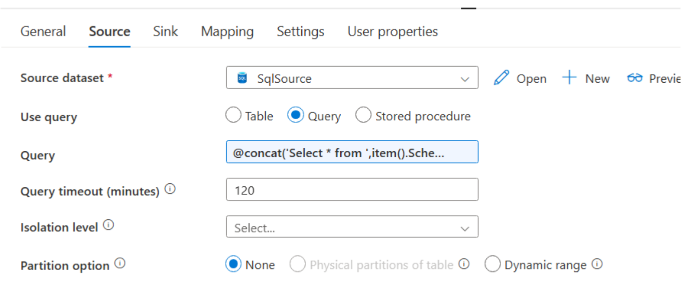
```
{
  "ItemsCount": "5"
}
```

ForEach1 ✓ Succeeded

under it i have copy activity







Source dataset * SqlSource

Use query ○ Table ● Query ○ Stored procedure

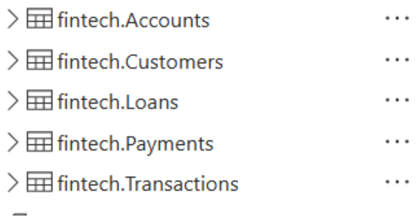
Query `@concat('Select * from ',item().Sche...`

Query timeout (minutes) 120

Isolation level Select...

Partition option ● None ○ Physical partitions of table ○ Dynamic range

For example: Select * from fintech.Accounts



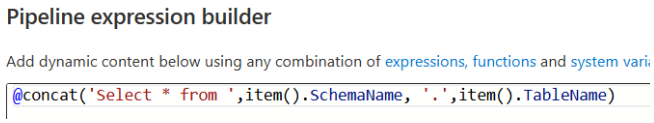
fintech.Accounts

fintech.Customers

fintech.Loans

fintech.Payments

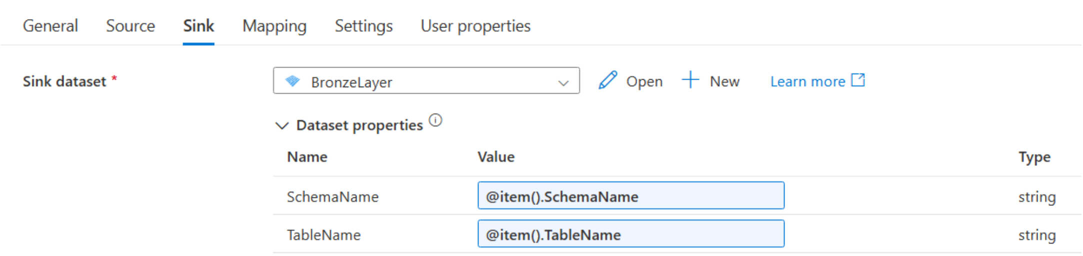
fintech.Transactions



Pipeline expression builder

Add dynamic content below using any combination of expressions, functions and system vari

```
@concat('Select * from ',item().SchemaName, '.',item().TableName)
```

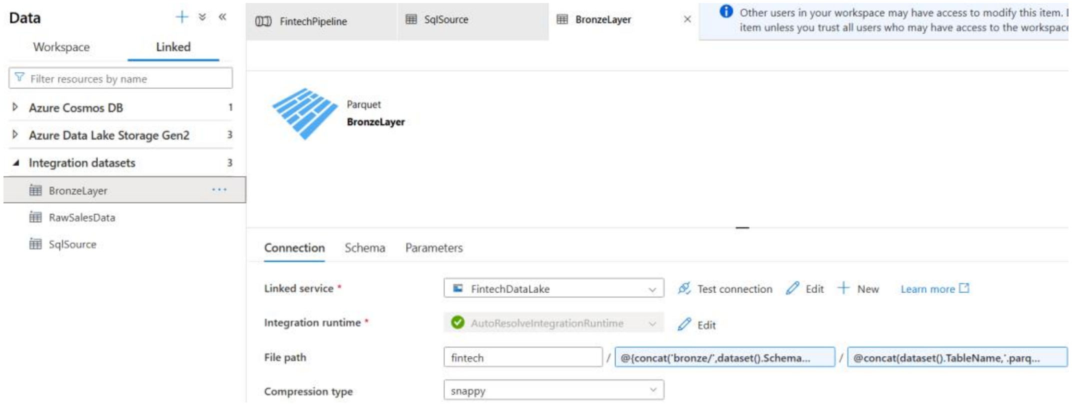


Sink dataset * BronzeLayer

Dataset properties

Name	Value	Type
SchemaName	@item().SchemaName	string
TableName	@item().TableName	string

Above will copy all files and will put it into bronze container in Parquet file format



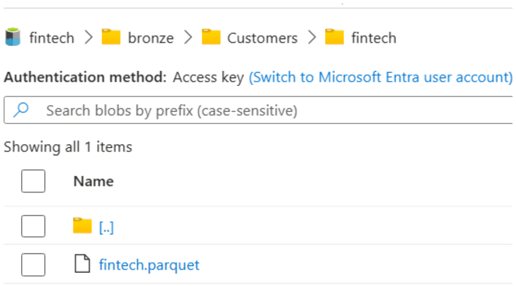
Other users in your workspace may have access to modify this item. I
item unless you trust all users who may have access to the workspac

Linked service * FintechDataLake

Integration runtime * AutoResolveIntegrationRuntime

File path fintech / @{concat('bronze/',dataset().Schema... / @concat(dataset().TableName,'.parq...

Compression type snappy



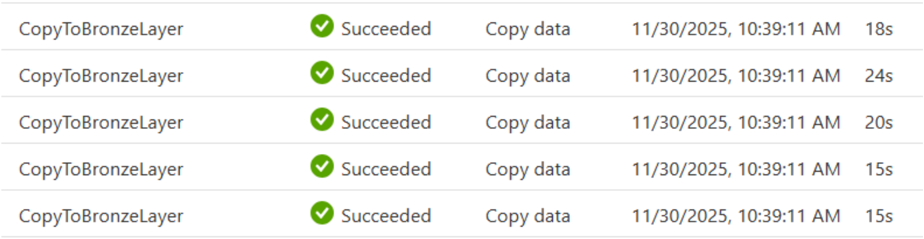
fintech > bronze > Customers > fintech

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

Search blobs by prefix (case-sensitive)

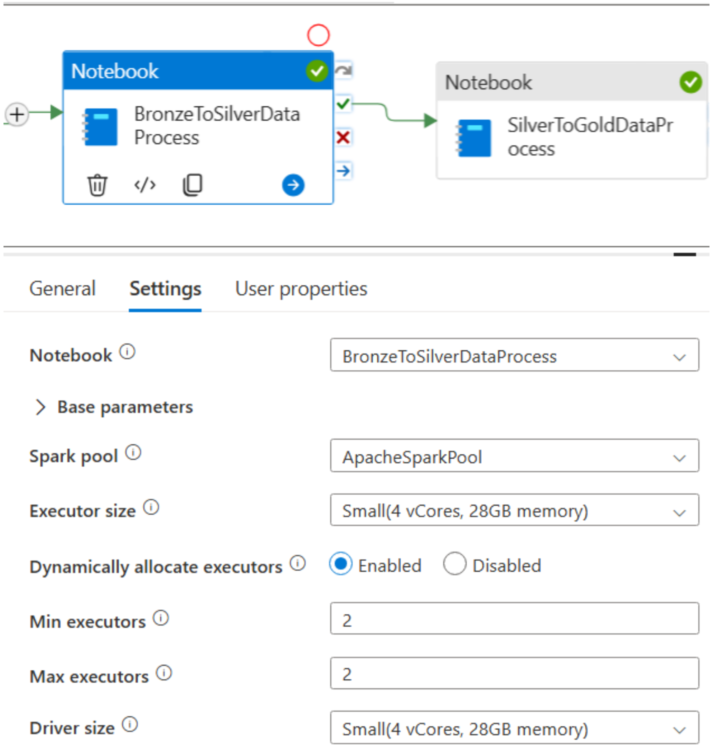
Showing all 1 items

□	**Name**
□	[..]
□	fintech.parquet



CopyToBronzeLayer	✓ Succeeded	Copy data	11/30/2025, 10:39:11 AM	18s
CopyToBronzeLayer	✓ Succeeded	Copy data	11/30/2025, 10:39:11 AM	24s
CopyToBronzeLayer	✓ Succeeded	Copy data	11/30/2025, 10:39:11 AM	20s
CopyToBronzeLayer	✓ Succeeded	Copy data	11/30/2025, 10:39:11 AM	15s
CopyToBronzeLayer	✓ Succeeded	Copy data	11/30/2025, 10:39:11 AM	15s

Then I attached notebook bronze to silver



Notebook: BronzeToSilverDataProcess

> Base parameters

Spark pool: ApacheSparkPool

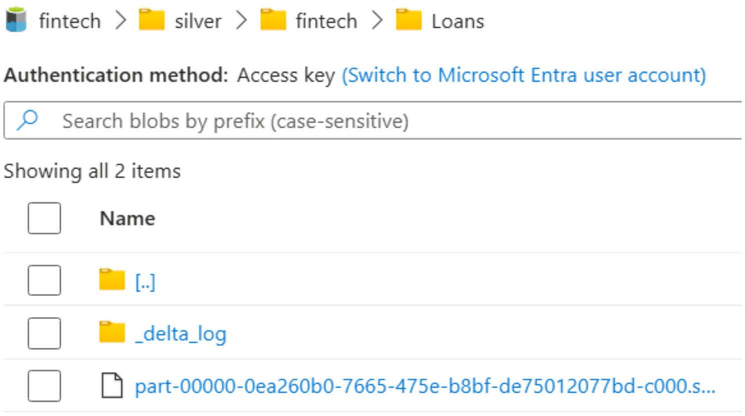
Executor size: Small(4 vCores, 28GB memory)

Dynamically allocate executors: ● Enabled ○ Disabled

Min executors: 2

Max executors: 2

Driver size: Small(4 vCores, 28GB memory)



fintech > silver > fintech > Loans

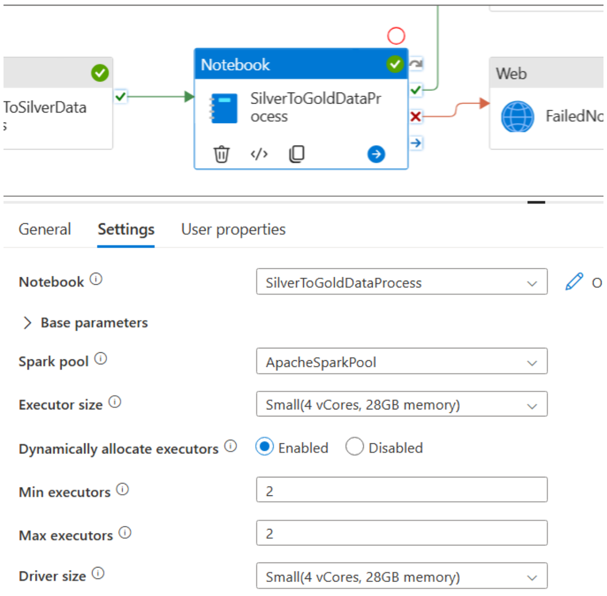
Authentication method: Access key (Switch to Microsoft Entra user account)

Search blobs by prefix (case-sensitive)

Showing all 2 items

□	Name
□	[..]
□	_delta_log
□	part-00000-0ea260b0-7665-475e-b8bf-de75012077bd-c000.s...

Then i attached notebook silver to gold



General | **Settings** | User properties

Notebook: SilverToGoldDataProcess

Base parameters

Spark pool: ApacheSparkPool

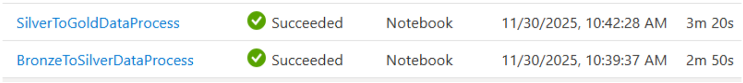
Executor size: Small(4 vCores, 28GB memory)

Dynamically allocate executors: ● Enabled ○ Disabled

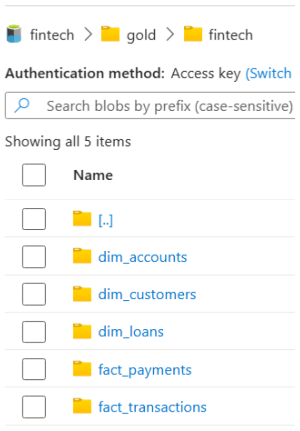
Min executors: 2

Max executors: 2

Driver size: Small(4 vCores, 28GB memory)



SilverToGoldDataProcess	✓ Succeeded	Notebook	11/30/2025, 10:42:28 AM	3m 20s
BronzeToSilverDataProcess	✓ Succeeded	Notebook	11/30/2025, 10:39:37 AM	2m 50s



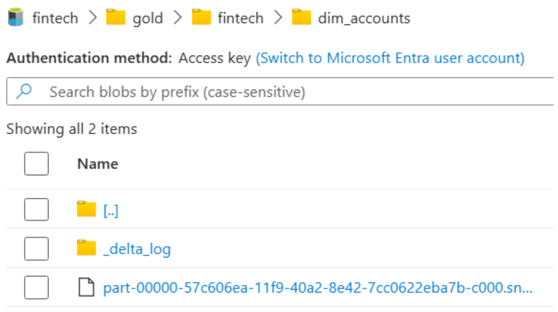
fintech > gold > fintech

Authentication method: Access key (Switch

Search blobs by prefix (case-sensitive)

Showing all 5 items

□	Name
□	[..]
□	dim_accounts
□	dim_customers
□	dim_loans
□	fact_payments
□	fact_transactions



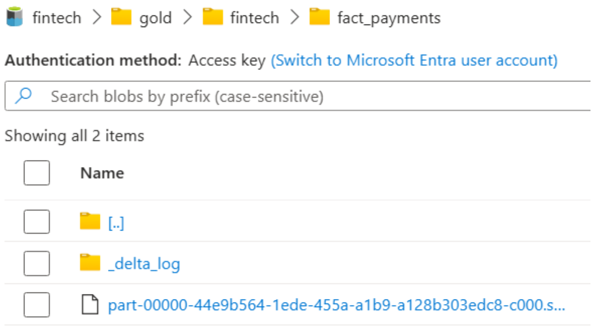
fintech > gold > fintech > dim_accounts

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

Search blobs by prefix (case-sensitive)

Showing all 2 items

□	Name
□	[..]
□	_delta_log
□	part-00000-57c606ea-11f9-40a2-8e42-7cc0622eba7b-c000.sn...



fintech > gold > fintech > fact_payments

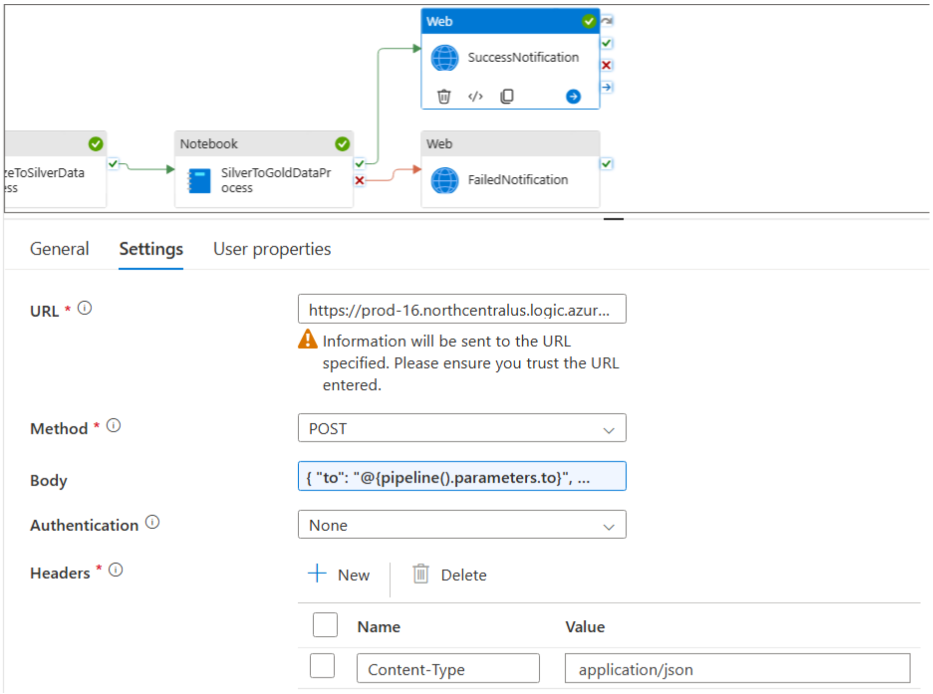
Authentication method: Access key (Switch to Microsoft Entra user account)

Search blobs by prefix (case-sensitive)

Showing all 2 items

□	Name
□	[..]
□	_delta_log
□	part-00000-44e9b564-1ede-455a-a1b9-a128b303edc8-c000.s...

Then i need to send notification via mail so i choose web activity and made 2 web activity if gold to silver notebook successful succeeded then will navigate to success notification web otherwise failed notification web



General | **Settings** | User properties

URL * https://prod-16.northcentralus.logic.azur...

⚠ Information will be sent to the URL specified. Please ensure you trust the URL entered.

Method * POST

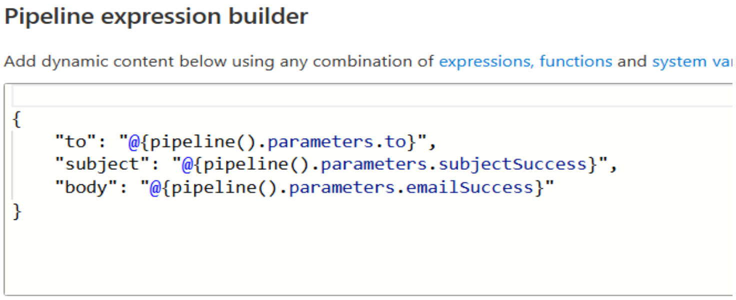
Body { "to": "@{pipeline().parameters.to}", ...

Authentication None

Headers * + New | Delete

□	Name	Value
□	Content-Type	application/json

Under body section

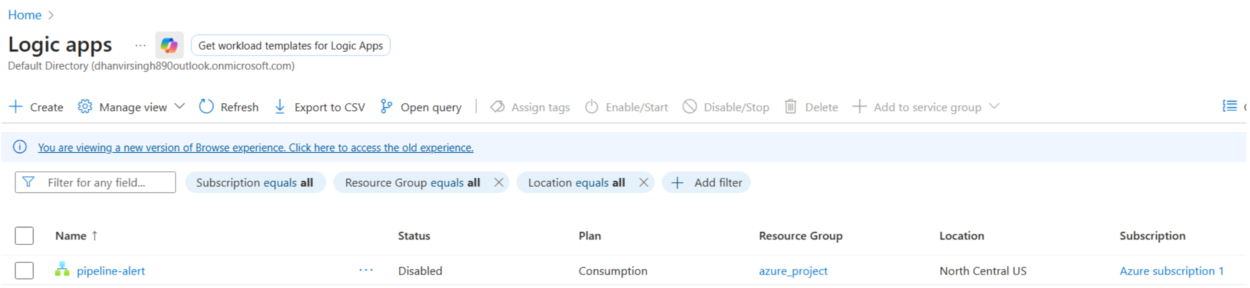


Pipeline expression builder

Add dynamic content below using any combination of expressions, functions and system va

```
{
    "to": "@{pipeline().parameters.to}",
    "subject": "@{pipeline().parameters.subjectSuccess}",
    "body": "@{pipeline().parameters.emailSuccess}"
}
```

Then I created logic app for sending email



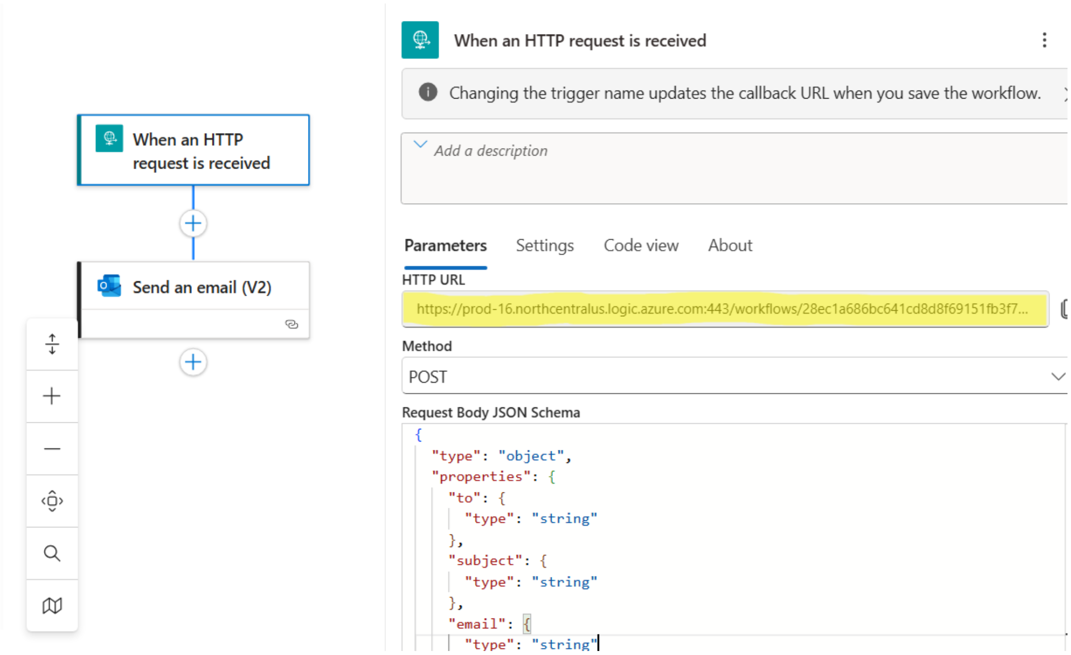
Logic apps ... Get workload templates for Logic Apps

Default Directory (dhanvirsingh890outlook.onmicrosoft.com)

You are viewing a new version of Browse experience. Click here to access the old experience.

Subscription equals **all** | Resource Group equals **all** | Location equals **all**

Name ↑	Status	Plan	Resource Group	Location	Subscription
pipeline-alert	Disabled	Consumption	azure_project	North Central US	Azure subscription 1



When an HTTP request is received

Changing the trigger name updates the callback URL when you save the workflow.

Add a description

Parameters

HTTP URL

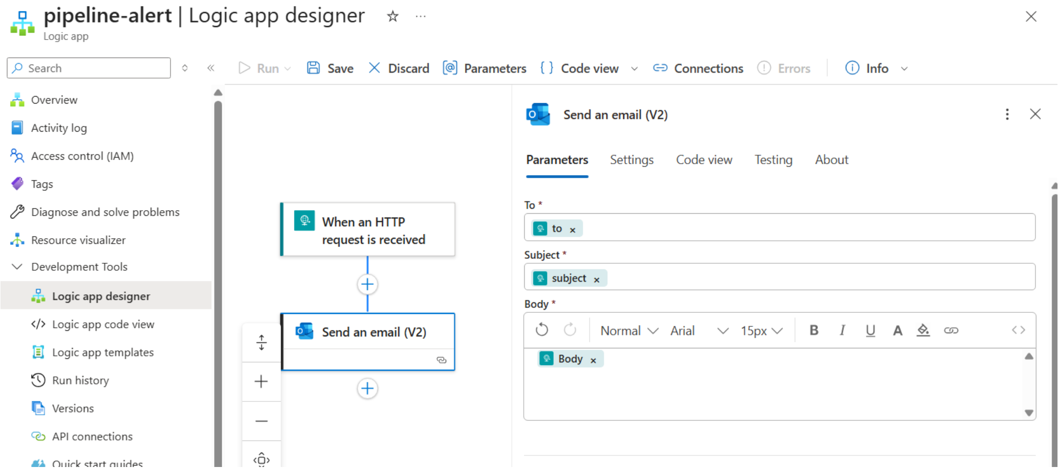
https://prod-16.northcentralus.logic.azure.com:443/workflows/28ec1a686bc641cd8d8f69151fb3f7...

Method

POST

Request Body JSON Schema

```
{
  "type": "object",
  "properties": {
    "to": {
      "type": "string"
    },
    "subject": {
      "type": "string"
    },
    "email": {
      "type": "string"
```



pipeline-alert | Logic app designer

Logic app

Send an email (V2)

Parameters

To *

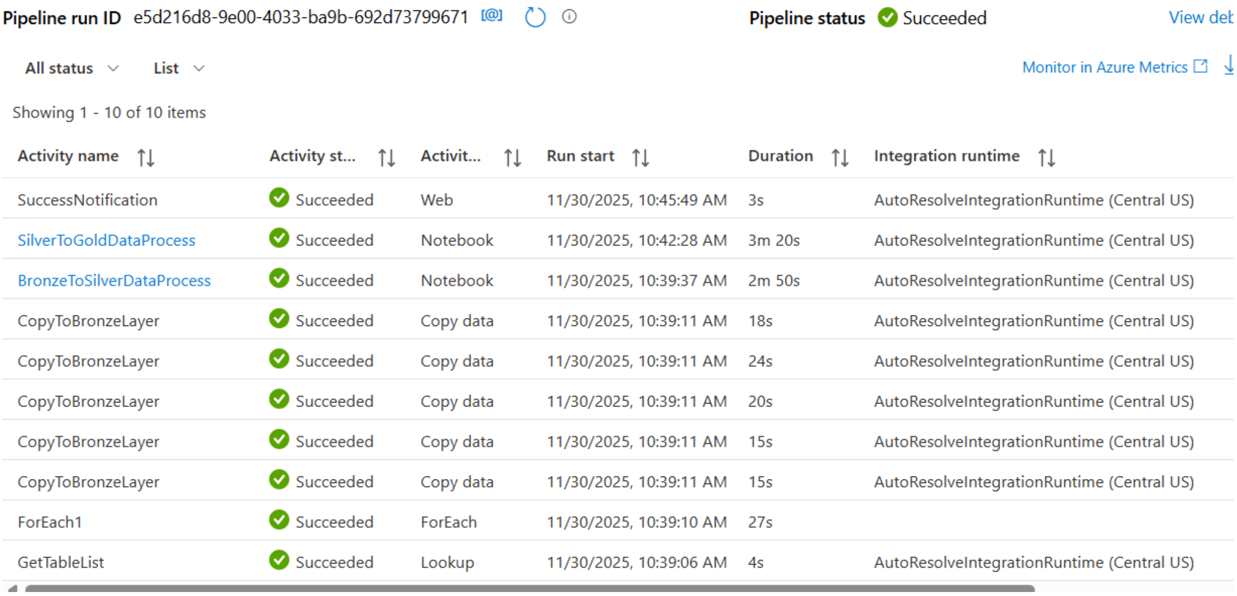
to ×

Subject *

subject ×

Body *

Body ×



Pipeline run ID e5d216d8-9e00-4033-ba9b-692d73799671 **Pipeline status** ✓ Succeeded View det

Showing 1 - 10 of 10 items

Activity name	Activity st...	Activit...	Run start	Duration	Integration runtime
SuccessNotification	✓ Succeeded	Web	11/30/2025, 10:45:49 AM	3s	AutoResolveIntegrationRuntime (Central US)
SilverToGoldDataProcess	✓ Succeeded	Notebook	11/30/2025, 10:42:28 AM	3m 20s	AutoResolveIntegrationRuntime (Central US)
BronzeToSilverDataProcess	✓ Succeeded	Notebook	11/30/2025, 10:39:37 AM	2m 50s	AutoResolveIntegrationRuntime (Central US)
CopyToBronzeLayer	✓ Succeeded	Copy data	11/30/2025, 10:39:11 AM	18s	AutoResolveIntegrationRuntime (Central US)
CopyToBronzeLayer	✓ Succeeded	Copy data	11/30/2025, 10:39:11 AM	24s	AutoResolveIntegrationRuntime (Central US)
CopyToBronzeLayer	✓ Succeeded	Copy data	11/30/2025, 10:39:11 AM	20s	AutoResolveIntegrationRuntime (Central US)
CopyToBronzeLayer	✓ Succeeded	Copy data	11/30/2025, 10:39:11 AM	15s	AutoResolveIntegrationRuntime (Central US)
CopyToBronzeLayer	✓ Succeeded	Copy data	11/30/2025, 10:39:11 AM	15s	AutoResolveIntegrationRuntime (Central US)
ForEach1	✓ Succeeded	ForEach	11/30/2025, 10:39:10 AM	27s	
GetTableList	✓ Succeeded	Lookup	11/30/2025, 10:39:06 AM	4s	AutoResolveIntegrationRuntime (Central US)