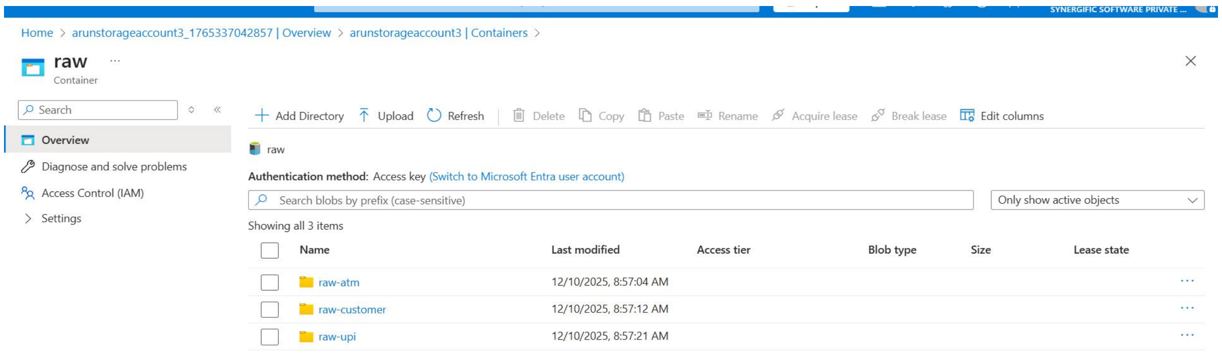
Day-1 Synopsis :

Services used : Azure ADLS Gen2 Storage, Azure Function, Azure EventGridTrigger, Azure ServiceBus.

Step 1: I have Created ADLS Gen2 storage layer forms the foundation of the ingestion pipeline by acting as the primary landing zone for raw data. Separate containers such as **raw/customers**, **raw/atm**, and **raw/upi** ensure logical isolation and clean organization of different data domains..

Step 2: Using Azure Event Grid enabled event-driven ingestion by triggering workflows automatically when new files arrive in the ADLS containers. Subscriptions are created on blob events (e.g., *BlobCreated*) so that the system instantly reacts when data appeared.



Home > arunstorageaccount3_1765337042857 | Overview > arunstorageaccount3 | Containers >

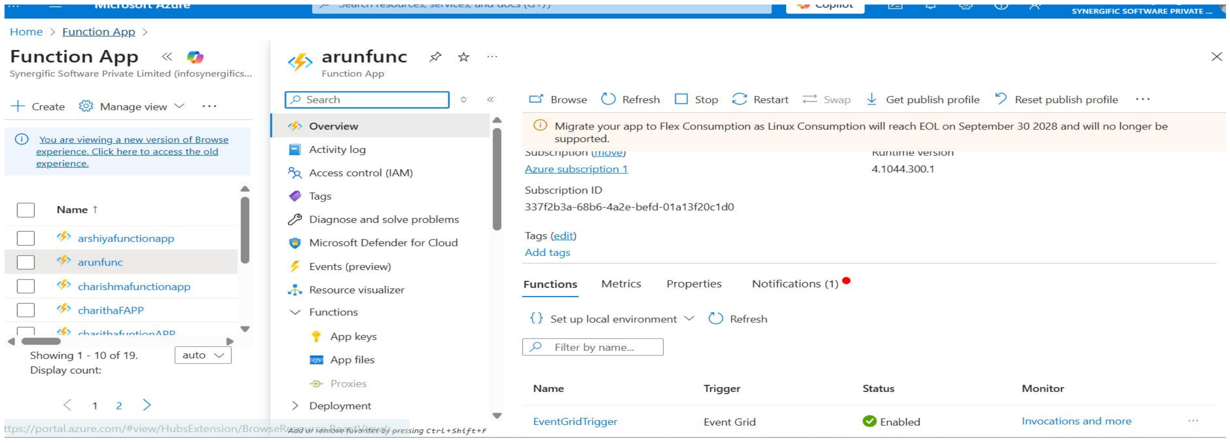
raw
Container

raw

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

Showing all 3 items

Name	Last modified	Access tier	Blob type	Size	Lease state
raw-atm	12/10/2025, 8:57:04 AM				
raw-customer	12/10/2025, 8:57:12 AM				
raw-upi	12/10/2025, 8:57:21 AM				



Home > Function App >

Function App
Synergific Software Private Limited (infosynergifics...

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

Name ↑
arshiyafunctionapp
arunfunc
charishmafunctionapp
charithaFAPP
charithafuntionAPP

Showing 1 - 10 of 19. Display count: auto

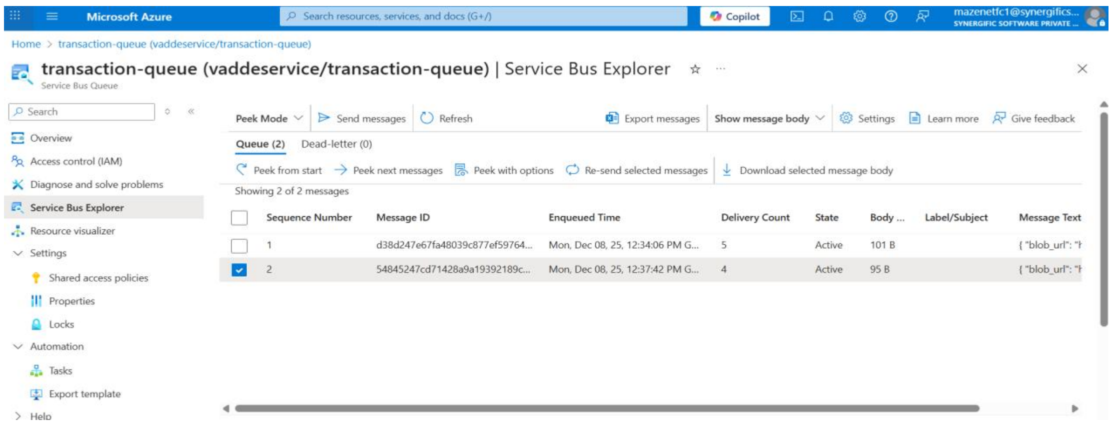
arunfunc
Function App

Migrate your app to Flex Consumption as Linux Consumption will reach EOL on September 30 2028 and will no longer be supported.

Subscription (move): Azure subscription 1
Runtime version: 4.1044.300.1
Subscription ID: 337f2b3a-68b6-4a2e-befd-01a13f20c1d0
Tags (edit): Add tags

Name	Trigger	Status	Monitor
EventGridTrigger	Event Grid	Enabled	Invocations and more

Step 3: Using Azure Service Bus provides a durable, high-throughput messaging layer used for coordinating real-time ingestion tasks. It acts as an intermediary message queue or topic where Event Grid or Functions can push validated ingestion instructions. With features like dead-lettering and message sessions, it ensures guaranteed delivery and fault-tolerant workflows.


Home > transaction-queue (vaddeservice/transaction-queue)

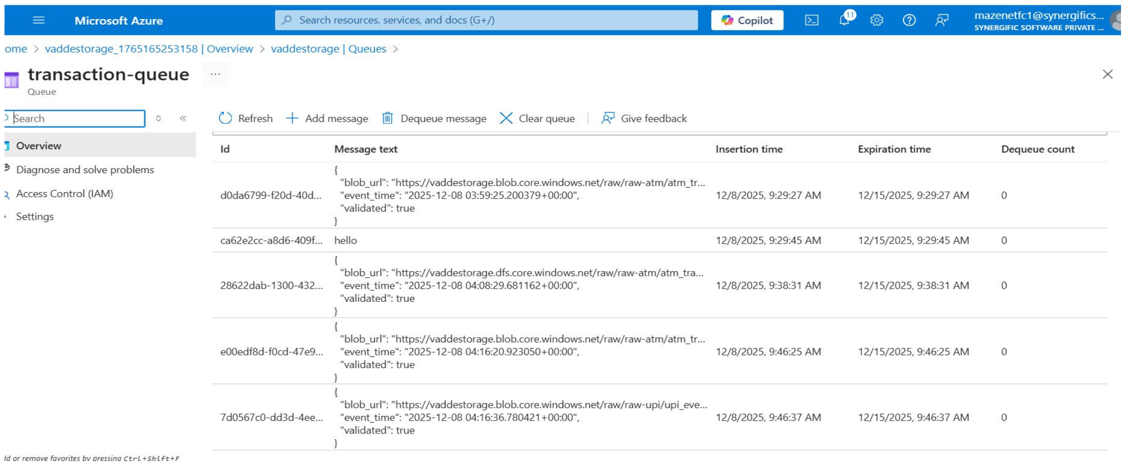
transaction-queue (vaddeservice/transaction-queue) | Service Bus Explorer

Service Bus Queue

Queue (2) Dead-letter (0)

Showing 2 of 2 messages

Sequence Number	Message ID	Enqueued Time	Delivery Count	State	Body ...	Label/Subject	Message Text
1	d38d247e67fa48039c877ef59764...	Mon, Dec 08, 25, 12:34:06 PM G...	5	Active	101 B		{ "blob_url": "h
2	54845247cd71428a9a19392189c...	Mon, Dec 08, 25, 12:37:42 PM G...	4	Active	95 B		{ "blob_url": "h


ome > vaddestorage_1765165253158 | Overview > vaddestorage | Queues >

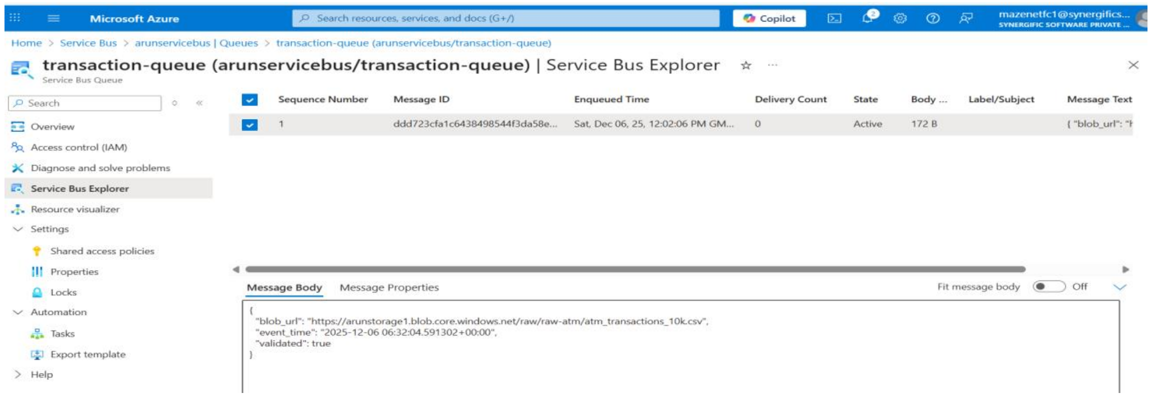
transaction-queue

Queue

Id	Message text	Insertion time	Expiration time	Dequeue count
d0da6799-f20d-40d...	{ "blob_url": "https://vaddestorage.blob.core.windows.net/raw/raw-atm/atm_tr... "event_time": "2025-12-08 03:59:25.200379+00:00", "validated": true }	12/8/2025, 9:29:27 AM	12/15/2025, 9:29:27 AM	0
ca62e2cc-a8d6-409f...	hello	12/8/2025, 9:29:45 AM	12/15/2025, 9:29:45 AM	0
28622dab-1300-432...	{ "blob_url": "https://vaddestorage.dfs.core.windows.net/raw/raw-atm/atm_tra... "event_time": "2025-12-08 04:08:29.681162+00:00", "validated": true }	12/8/2025, 9:38:31 AM	12/15/2025, 9:38:31 AM	0
e00edf8d-f0cd-47e9...	{ "blob_url": "https://vaddestorage.blob.core.windows.net/raw/raw-atm/atm_tr... "event_time": "2025-12-08 04:16:20.923050+00:00", "validated": true }	12/8/2025, 9:46:25 AM	12/15/2025, 9:46:25 AM	0
7d0567c0-dd3d-4ee...	{ "blob_url": "https://vaddestorage.blob.core.windows.net/raw/raw-upi/upi_eve... "event_time": "2025-12-08 04:16:36.780421+00:00", "validated": true }	12/8/2025, 9:46:37 AM	12/15/2025, 9:46:37 AM	0

Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
mazenetfc1@synergifics...
SYNERGIFIC SOFTWARE PRIVATE ...
Search
Overview
Diagnose and solve problems
Access Control (IAM)
Settings
Refresh
Add message
Dequeue message
Clear queue
Give feedback
ld or remove favorites by pressing Ctrl+Shift+F

Step 4: Using Python Azure Function acts as the ingestion controller that processes events received from Event Grid. It validates the metadata of arriving files, checks naming conventions, ensures schema compliance, and routes clean ingestion requests to Service Bus. Running as a serverless compute service, it automatically scales based on incoming events.


Home > Service Bus > arunservicebus | Queues > transaction-queue (arunservicebus/transaction-queue)

transaction-queue (arunservicebus/transaction-queue) | Service Bus Explorer

Service Bus Queue

Sequence Number	Message ID	Enqueued Time	Delivery Count	State	Body ...	Label/Subject	Message Text
1	ddd723cfa1c6438498544f3da58e...	Sat, Dec 06, 25, 12:02:06 PM GM...	0	Active	172 B		{ "blob_url": "h

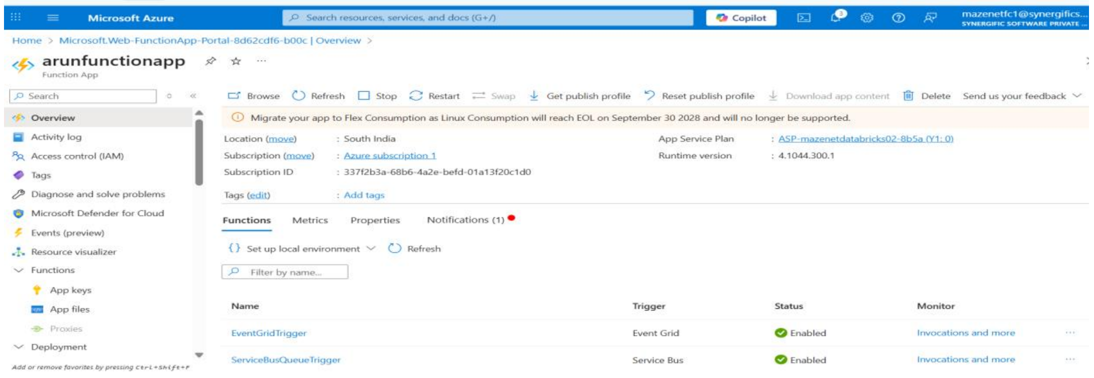
Message Body

```
{
  "blob_url": "https://arunstorage1.blob.core.windows.net/raw/raw-atm/atm_transactions_10k.csv",
  "event_time": "2025-12-06 06:32:04.591302+00:00",
  "validated": true
}
```

Day 2 Synopsis :

Services used : Azure ADLS Gen2 Storage, Azure Function App, Azure Cosmos DB, Azure Queue, Azure Data Bricks.

Step 1: Using Queue-triggered Azure Functions act as automated processors that pick up events from the Storage Queue and perform early-stage validation. These functions load the raw files from Blob Storage like ATM, UPI.They also remove duplicate transactions using unique IDs to maintain consistency. Each transaction is classified into ATM, UPI, IMPS, or NEFT based on metadata.

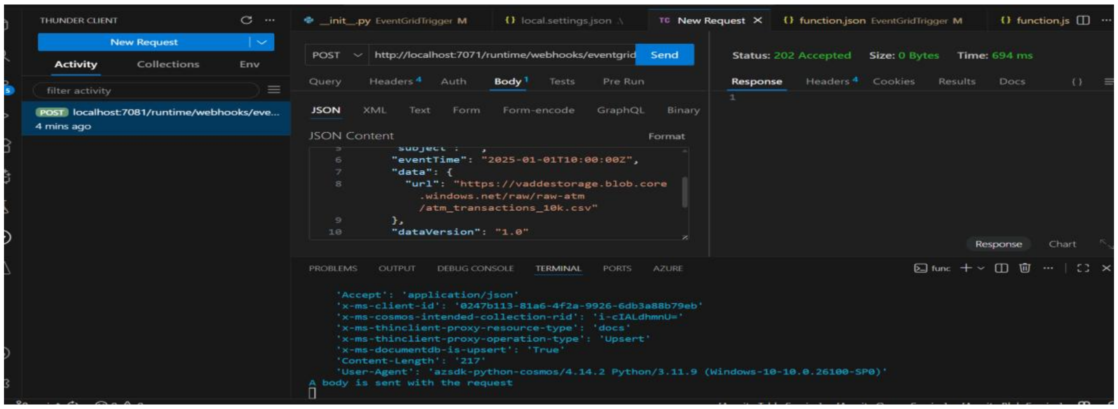


Migrate your app to Flex Consumption as Linux Consumption will reach EOL on September 30 2028 and will no longer be supported.

Location (move)	: South India	App Service Plan	: ASP-mazenetdatabricks02-8b5a (Y1: 0)
Subscription (move)	: Azure subscription 1	Runtime version	: 4.1044.300.1
Subscription ID	: 337f2b3a-68b6-4a2e-befd-01a13f20c1d0		
Tags (edit)	: Add tags		

Name	Trigger	Status	Monitor
EventGridTrigger	Event Grid	✓ Enabled	Invocations and more
ServiceBusQueueTrigger	Service Bus	✓ Enabled	Invocations and more

Step 2: Azure Cosmos DB serves as the system's operational data store Created BankDB database, providing fast and scalable performance for real-time lookups. Different collections store specialized Containers such as ATMTransactions, UPIEvents, and FraudAlerts. Each collection uses a specific partition key to ensure efficient read/write distribution across the database.



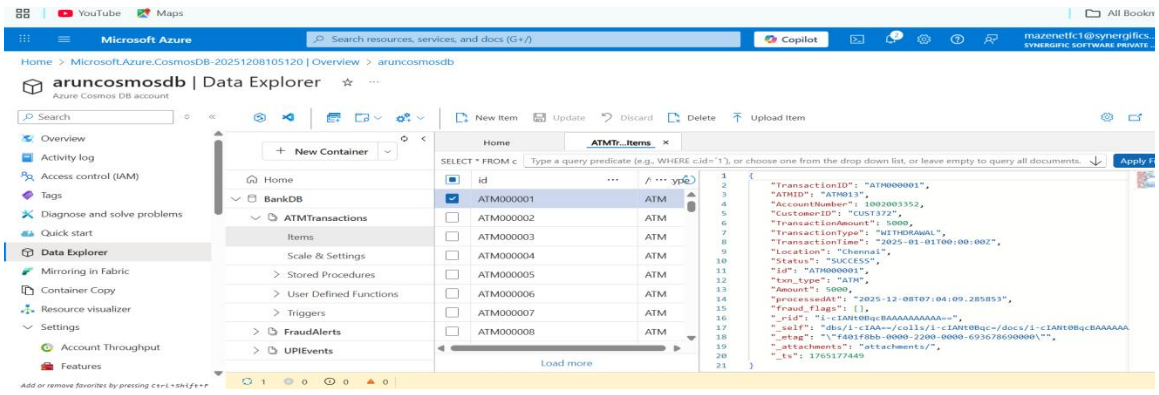
POST localhost:7081/runtime/webhooks/eve...
4 mins ago

POST http://localhost:7071/runtime/webhooks/eventgrid

Status: 202 Accepted Size: 0 Bytes Time: 694 ms

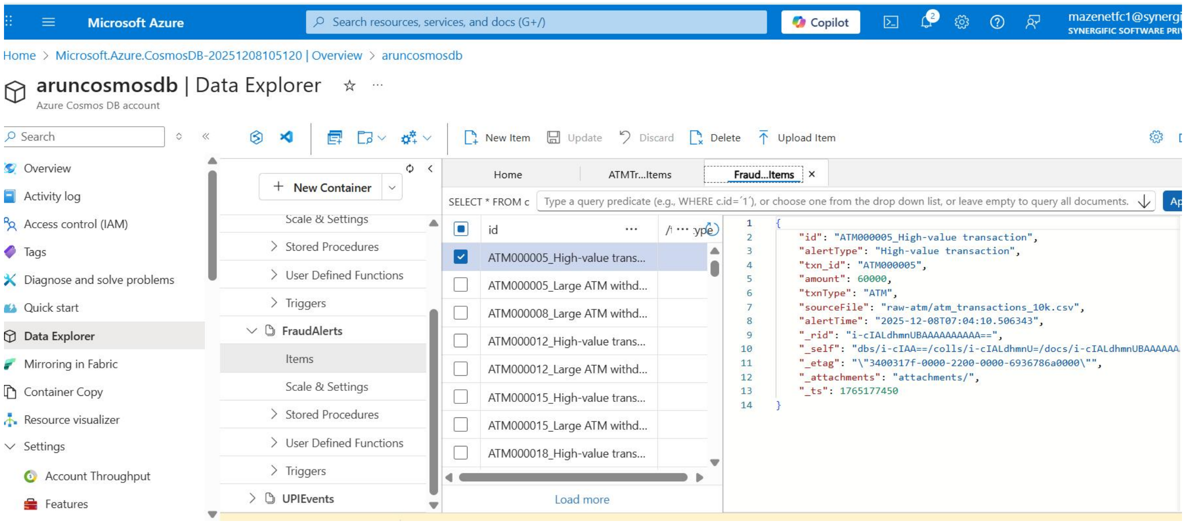
```
    "subject": "",
    "eventTime": "2025-01-01T10:00:00Z",
    "data": {
      "url": "https://vaddestorage.blob.core
        .windows.net/raw/raw-atm
        /atm_transactions_10k.csv"
    },
    "dataVersion": "1.0"
```

```
    'Accept': 'application/json'
    'x-ms-client-id': '0247b113-81a6-4f2a-9926-6db3a88b79eb'
    'x-ms-cosmos-intended-collection-rid': 'i-cIALdhmnU='
    'x-ms-thinclient-proxy-resource-type': 'docs'
    'x-ms-thinclient-proxy-operation-type': 'Upsert'
    'x-ms-documentdb-is-upsert': 'True'
    'Content-Length': '217'
    'User-Agent': 'azsdk-python-cosmos/4.14.2 Python/3.11.9 (Windows-10-10.0.26100-SP0)'
A body is sent with the request
```



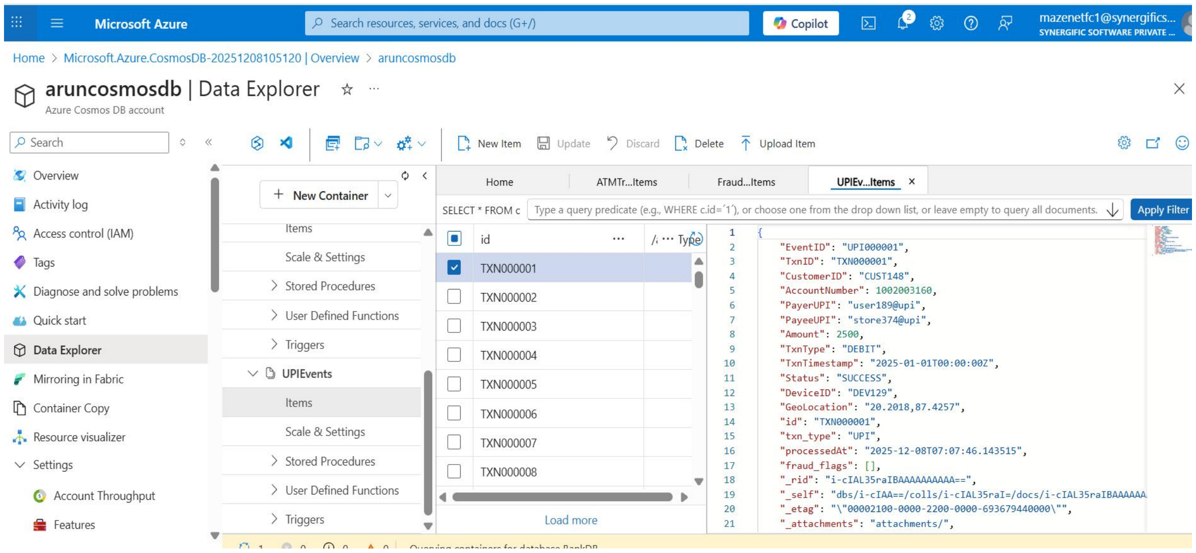
	id	ype
✓	ATM000001	ATM
	ATM000002	ATM
	ATM000003	ATM
	ATM000004	ATM
	ATM000005	ATM
	ATM000006	ATM
	ATM000007	ATM
	ATM000008	ATM

```
{
    "TransactionID": "ATM000001",
    "ATMID": "ATM013",
    "AccountNumber": 1002003352,
    "CustomerID": "CUST372",
    "TransactionAmount": 5000,
    "TransactionType": "WITHDRAWAL",
    "TransactionTime": "2025-01-01T00:00:00Z",
    "Location": "Chennai",
    "Status": "SUCCESS",
    "id": "ATM000001",
    "txn_type": "ATM",
    "Amount": 5000,
    "processedAt": "2025-12-08T07:04:09.285853",
    "fraud_flags": [],
    "_rid": "i-cIANt0BqcBAAAAAAAAAA==",
    "_self": "dbs/i-cIAA==/colls/i-cIANt0Bqc=/docs/i-cIANt0BqcBAAAAAA
    "_etag": "\"f401f8bb-0000-2200-0000-693678690000\"",
    "_attachments": "attachments/",
    "_ts": 1765177449
}
```



	id	ype
✓	ATM000005_High-value trans...	
	ATM000005_Large ATM withd...	
	ATM000008_Large ATM withd...	
	ATM000012_High-value trans...	
	ATM000012_Large ATM withd...	
	ATM000015_High-value trans...	
	ATM000015_Large ATM withd...	
	ATM000018_High-value trans...	

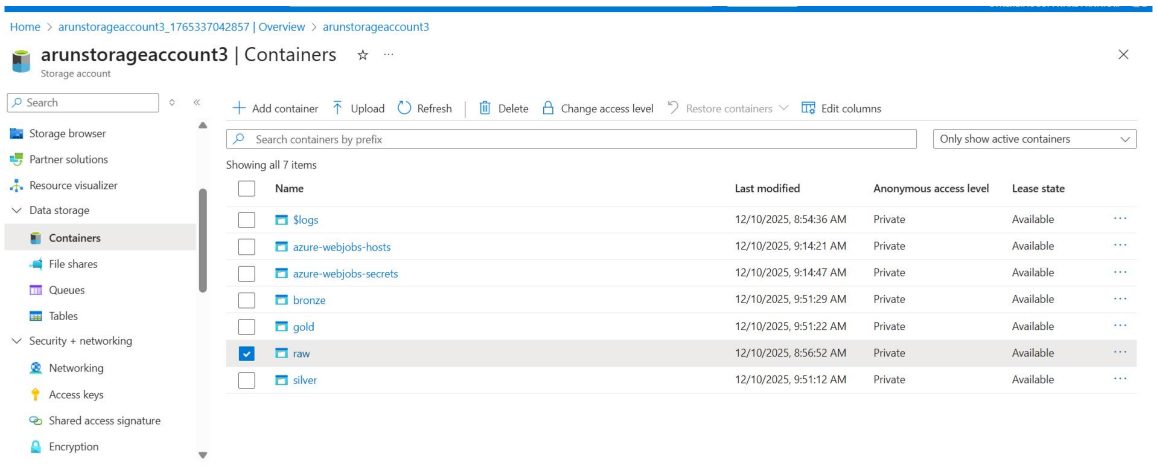
```
{
    "id": "ATM000005_High-value transaction",
    "alertType": "High-value transaction",
    "txn_id": "ATM000005",
    "amount": 60000,
    "txnType": "ATM",
    "sourceFile": "raw-atm/atm_transactions_10k.csv",
    "alertTime": "2025-12-08T07:04:10.506343",
    "_rid": "i-cIALdhmnUBAAAAAAAAAA==",
    "_self": "dbs/i-cIAA==/colls/i-cIALdhmnU=/docs/i-cIALdhmnUBAAAAAA
    "_etag": "\"3400317f-0000-2200-0000-6936786a0000\"",
    "_attachments": "attachments/",
    "_ts": 1765177450
}
```



	id	Type
✓	TXN000001	
	TXN000002	
	TXN000003	
	TXN000004	
	TXN000005	
	TXN000006	
	TXN000007	
	TXN000008	

```
{
    "EventID": "UPI000001",
    "TxnID": "TXN000001",
    "CustomerID": "CUST148",
    "AccountNumber": 1002003160,
    "PayerUPI": "user189@upi",
    "PayeeUPI": "store374@upi",
    "Amount": 2500,
    "TxnType": "DEBIT",
    "TxnTimestamp": "2025-01-01T00:00:00Z",
    "Status": "SUCCESS",
    "DeviceID": "DEV129",
    "GeoLocation": "20.2018,87.4257",
    "id": "TXN000001",
    "txn_type": "UPI",
    "processedAt": "2025-12-08T07:07:46.143515",
    "fraud_flags": [],
    "_rid": "i-cIAL35raIBAAAAAAAAAA==",
    "_self": "dbs/i-cIAA==/colls/i-cIAL35raI=/docs/i-cIAL35raIBAAAAAA
    "_etag": "\"00002100-0000-2200-0000-693679440000\"",
    "_attachments": "attachments/",
```

Step 3: Using PySpark cleaned-up pipeline runs on Databricks to process large datasets from Cosmos DB and ADLS. The pipeline normalizes all columns, ensuring consistent formats for timestamps, currency, and identifiers. ATM and UPI data are merged into a unified Fact_Transactions dataset for downstream analytics. Finally, the processed data is written into Bronze, Silver, and Gold Delta Lake layers to ADLS.



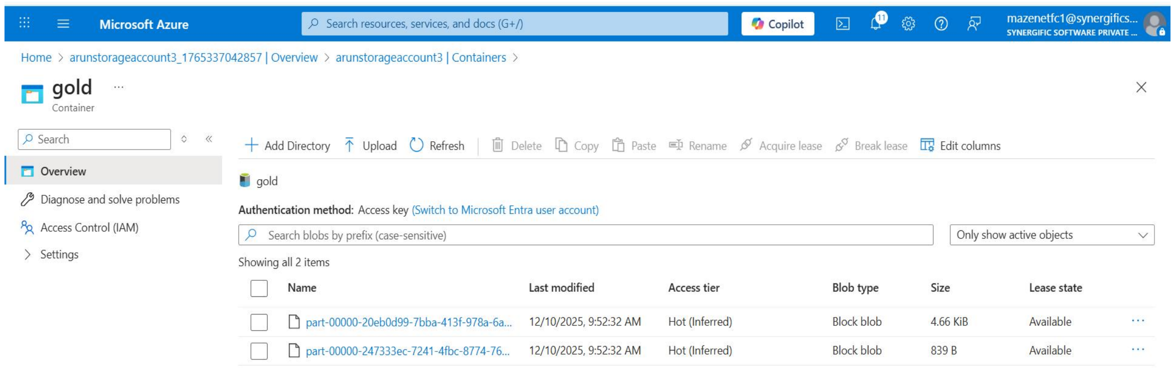
Home > arunstorageaccount3_1765337042857 | Overview > arunstorageaccount3

arunstorageaccount3 | Containers

Storage account

Showing all 7 items

Name	Last modified	Anonymous access level	Lease state
$logs	12/10/2025, 8:54:36 AM	Private	Available
azure-webjobs-hosts	12/10/2025, 9:14:21 AM	Private	Available
azure-webjobs-secrets	12/10/2025, 9:14:47 AM	Private	Available
bronze	12/10/2025, 9:51:29 AM	Private	Available
gold	12/10/2025, 9:51:22 AM	Private	Available
raw	12/10/2025, 8:56:52 AM	Private	Available
silver	12/10/2025, 9:51:12 AM	Private	Available



Home > arunstorageaccount3_1765337042857 | Overview > arunstorageaccount3 | Containers >

gold

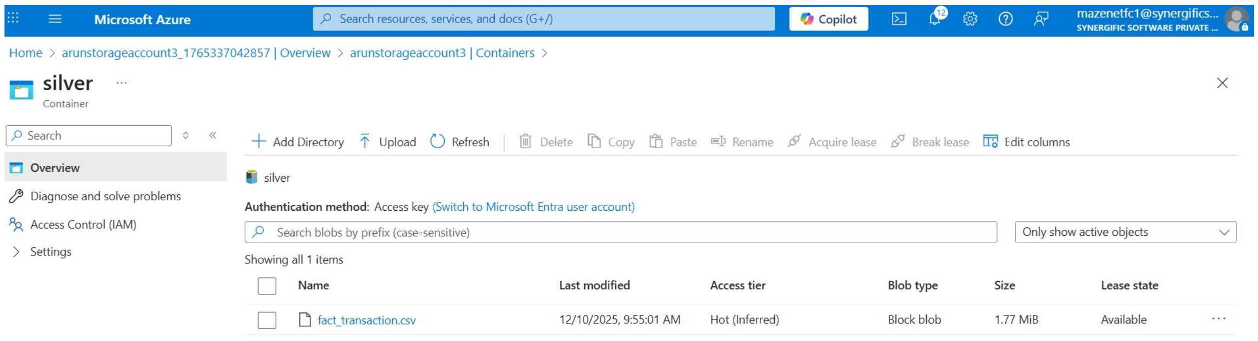
Container

gold

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

Showing all 2 items

Name	Last modified	Access tier	Blob type	Size	Lease state
part-00000-20eb0d99-7bba-413f-978a-6a...	12/10/2025, 9:52:32 AM	Hot (Inferred)	Block blob	4.66 KiB	Available
part-00000-247333ec-7241-4fbc-8774-76...	12/10/2025, 9:52:32 AM	Hot (Inferred)	Block blob	839 B	Available



Home > arunstorageaccount3_1765337042857 | Overview > arunstorageaccount3 | Containers >

silver

Container

silver

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

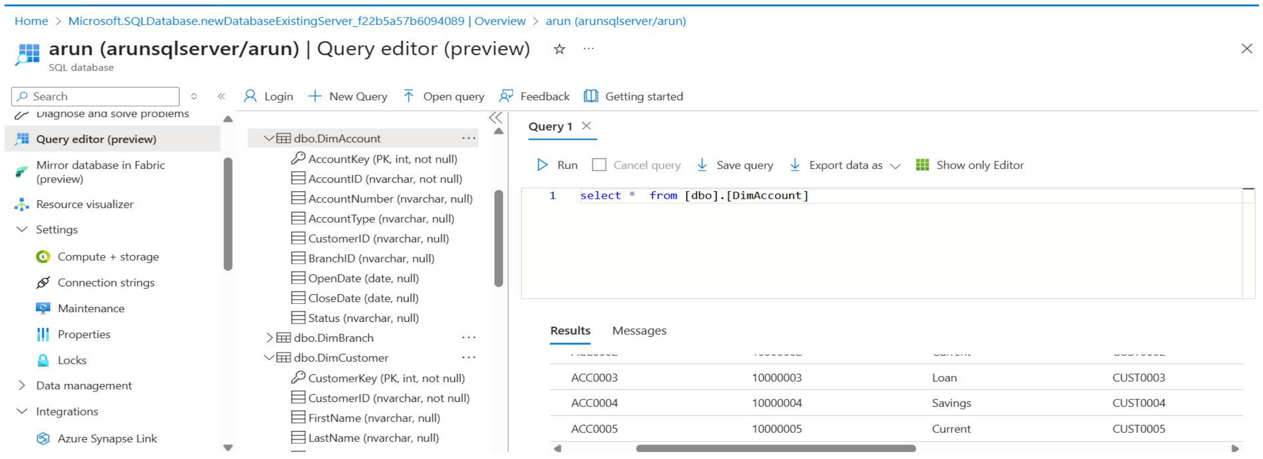
Showing all 1 items

Name	Last modified	Access tier	Blob type	Size	Lease state
fact_transaction.csv	12/10/2025, 9:55:01 AM	Hot (Inferred)	Block blob	1.77 MiB	Available

Day 3 Synopsis:

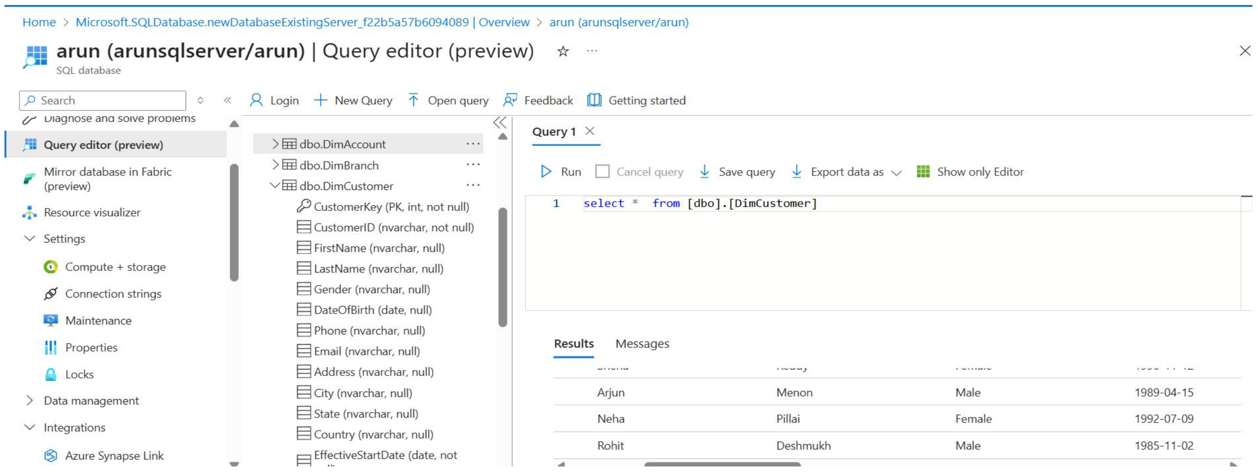
Services Used : Azure SQL Database, Sql Server, Azure DataBricks, Azure ADLS Gen2 Storage, Azure Function App, Azure Timer Triggers.

Step 1: Created data warehouse using Azure SQL follows a star schema to support fast reporting and dashboards. All master data such as dimcustomers, dimaccounts, dimbranches, and dimproducts, dimdate are kept in dimension tables. DimCustomer is maintained as an SCD Type-2 table, so historical changes can be tracked reliably. Under Fact tables, FactTransactions, FactFraudDetection, FactCustomerActivity are created.


Home > Microsoft.SQLDatabase.newDatabaseExistingServer_f22b5a57b6094089 | Overview > arun (arunsqlserver/arun)
arun (arunsqlserver/arun) | Query editor (preview)
SQL database
Search
Login
New Query
Open query
Feedback
Getting started
Diagnose and solve problems
Query editor (preview)
Mirror database in Fabric (preview)
Resource visualizer
Settings
Compute + storage
Connection strings
Maintenance
Properties
Locks
Data management
Integrations
Azure Synapse Link
dbo.DimAccount
AccountKey (PK, int, not null)
AccountID (nvarchar, not null)
AccountNumber (nvarchar, null)
AccountType (nvarchar, null)
CustomerID (nvarchar, null)
BranchID (nvarchar, null)
OpenDate (date, null)
CloseDate (date, null)
Status (nvarchar, null)
dbo.DimBranch
dbo.DimCustomer
CustomerKey (PK, int, not null)
CustomerID (nvarchar, not null)
FirstName (nvarchar, null)
LastName (nvarchar, null)
Query 1
Run
Cancel query
Save query
Export data as
Show only Editor
Results
Messages

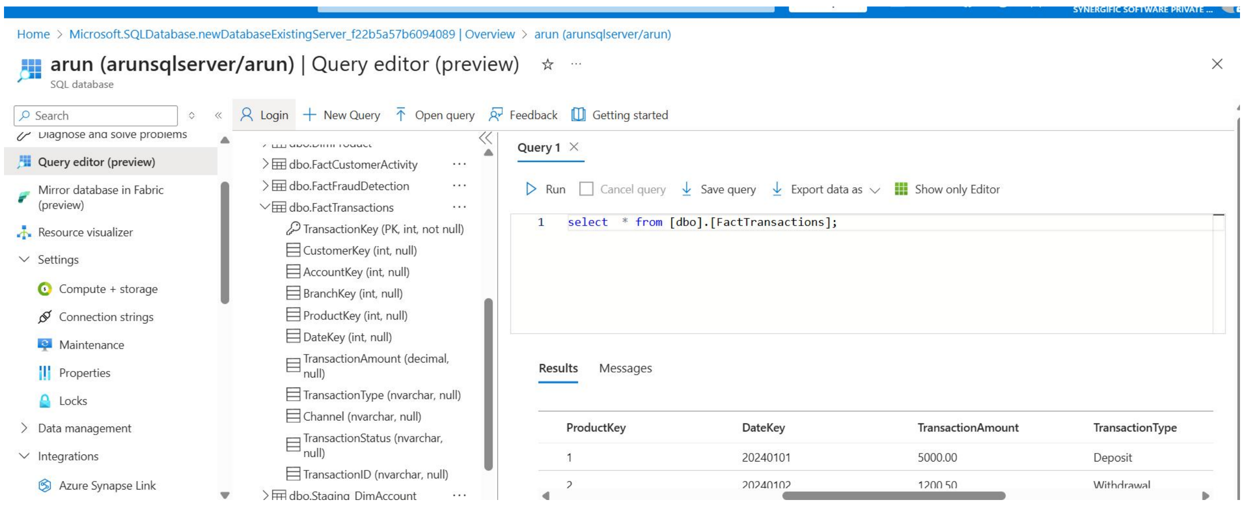
```
select * from [dbo].[DimAccount]
```

ACC0003	10000003	Loan	CUST0003
ACC0004	10000004	Savings	CUST0004
ACC0005	10000005	Current	CUST0005



```
select * from [dbo].[DimCustomer]
```

Arjun	Menon	Male	1989-04-15
Neha	Pillai	Female	1992-07-09
Rohit	Deshmukh	Male	1985-11-02


Home > Microsoft.SQLDatabase.newDatabaseExistingServer_f22b5a57b6094089 | Overview > arun (arunsqlserver/arun)

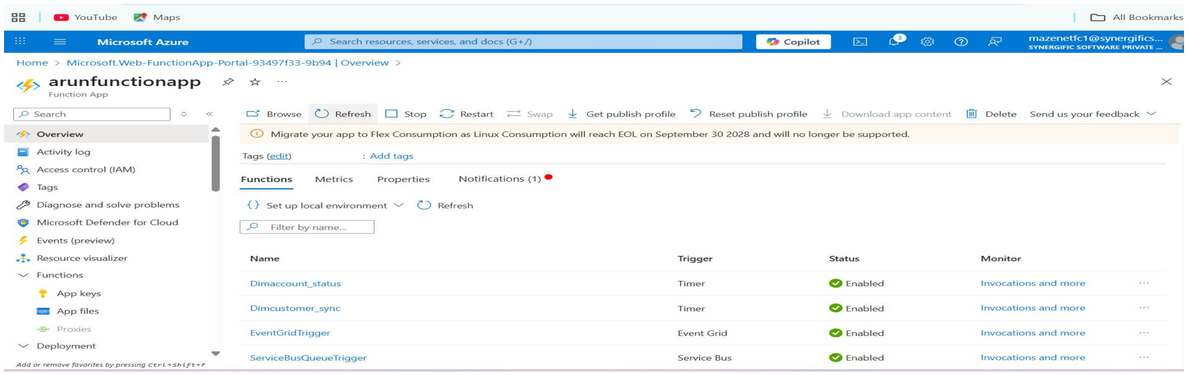
arun (arunsqlserver/arun) | Query editor (preview)
SQL database

```sql
select * from [dbo].[FactTransactions];
```

ProductKey	DateKey	TransactionAmount	TransactionType
1	20240101	5000.00	Deposit
2	20240102	1200.50	Withdrawal

Step 2: Using PySpark jobs read cleaned and structured data from the Silver Delta layer to populate the Azure SQL Data Warehouse. These jobs apply final business rules, derive new fields, and prepare dimension and fact-ready datasets. UPSERT (MERGE) logic is used to refresh dimension data and maintain SCD2 records. Fact tables are incrementally loaded to ensure fast and consistent updates.

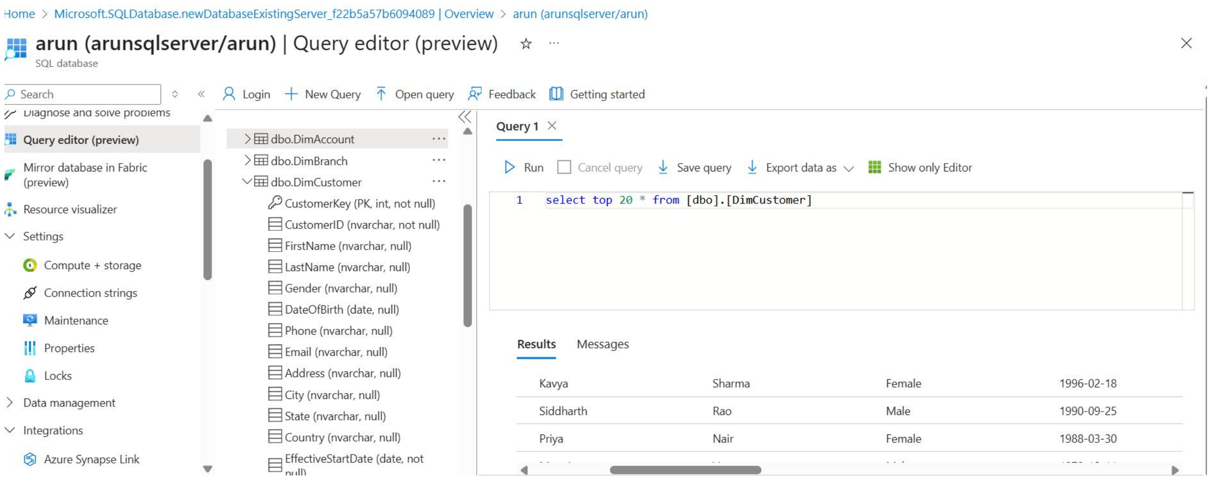
Step 3: Using timer-triggered Azure Functions Daily automate data refreshing into the warehouse. The first function reloads the full customer master and ensures DimCustomer stays updated. Another function updates account-related metadata, such as whether an account is active. Using this automated schedule reduces manual overhead and keeps reporting data accurate every day.


Home > Microsoft.Web-FunctionApp-Portal-93497f33-9b94 | Overview >

arunfunctionapp
Function App

Migrate your app to Flex Consumption as Linux Consumption will reach EOL on September 30 2028 and will no longer be supported.

Name	Trigger	Status	Monitor
Dimaccount_status	Timer	✓ Enabled	Invocations and more
Dimcustomer_sync	Timer	✓ Enabled	Invocations and more
EventGridTrigger	Event Grid	✓ Enabled	Invocations and more
ServiceBusQueueTrigger	Service Bus	✓ Enabled	Invocations and more


Home > Microsoft.SQLDatabase.newDatabaseExistingServer_f22b5a57b6094089 | Overview > arun (arunsqlserver/arun)

arun (arunsqlserver/arun) | Query editor (preview)

SQL database

dbo.DimAccount
dbo.DimBranch
dbo.DimCustomer
CustomerKey (PK, int, not null)
CustomerID (nvarchar, not null)
FirstName (nvarchar, null)
LastName (nvarchar, null)
Gender (nvarchar, null)
DateOfBirth (date, null)
Phone (nvarchar, null)
Email (nvarchar, null)
Address (nvarchar, null)
City (nvarchar, null)
State (nvarchar, null)
Country (nvarchar, null)
EffectiveStartDate (date, not null)

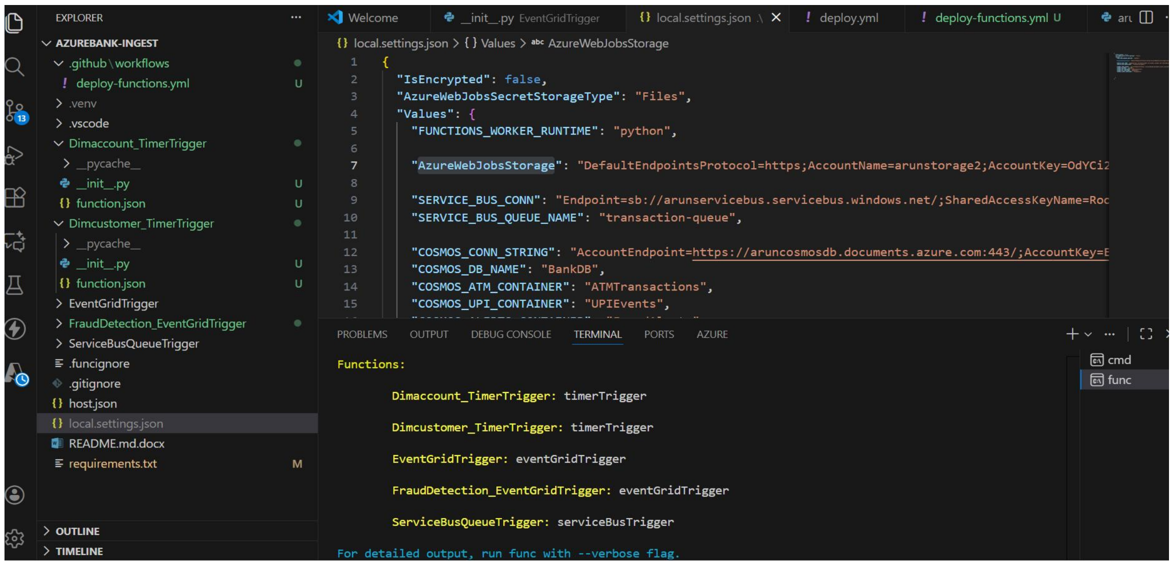
```sql
select top 20 * from [dbo].[DimCustomer]
```

Kavya	Sharma	Female	1996-02-18
Siddharth	Rao	Male	1990-09-25
Priya	Nair	Female	1988-03-30

Day 4 Synopsis :

Services used : Azure Function App, Azure Event Grid, Azure Cosmos DB, CI/CD.

Step 1: In this task, I built a real-time alerting system that reacts instantly to suspicious banking transactions. Any high-value activity triggers an event through Event Grid, which is processed by an Azure Function. When a transaction looks risky, a fraud alert is written to the FraudAlerts collection in Cosmos DB. At the same time, the system sends an alert message through Service Bus so customers can be notified immediately via SMS or email.

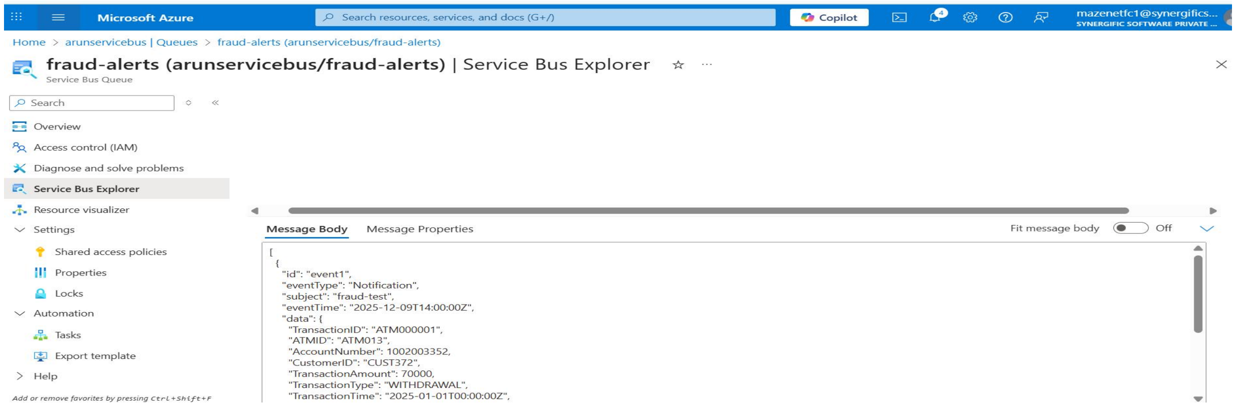


AZUREBANK-INGEST
.github\workflows
deploy-functions.yml U
.venv
.vscode
Dimaccount_TimerTrigger
__pycache__
__init__.py U
function.json U
Dimcustomer_TimerTrigger
__pycache__
__init__.py U
function.json U
EventGridTrigger
FraudDetection_EventGridTrigger
ServiceBusQueueTrigger
.funcignore
.gitignore
host.json
local.settings.json
README.md.docx
requirements.txt M

local.settings.json > { } Values > AzureWebJobsStorage

```json
{
  "IsEncrypted": false,
  "AzureWebJobsSecretStorageType": "Files",
  "Values": {
    "FUNCTIONS_WORKER_RUNTIME": "python",

    "AzureWebJobsStorage": "DefaultEndpointsProtocol=https;AccountName=arunstorage2;AccountKey=OdYCi2

    "SERVICE_BUS_CONN": "Endpoint=sb://arunservicebus.servicebus.windows.net/;SharedAccessKeyName=Roc
    "SERVICE_BUS_QUEUE_NAME": "transaction-queue",

    "COSMOS_CONN_STRING": "AccountEndpoint=https://aruncosmosdb.documents.azure.com:443/;AccountKey=E
    "COSMOS_DB_NAME": "BankDB",
    "COSMOS_ATM_CONTAINER": "ATMTransactions",
    "COSMOS_UPI_CONTAINER": "UPIEvents",
```

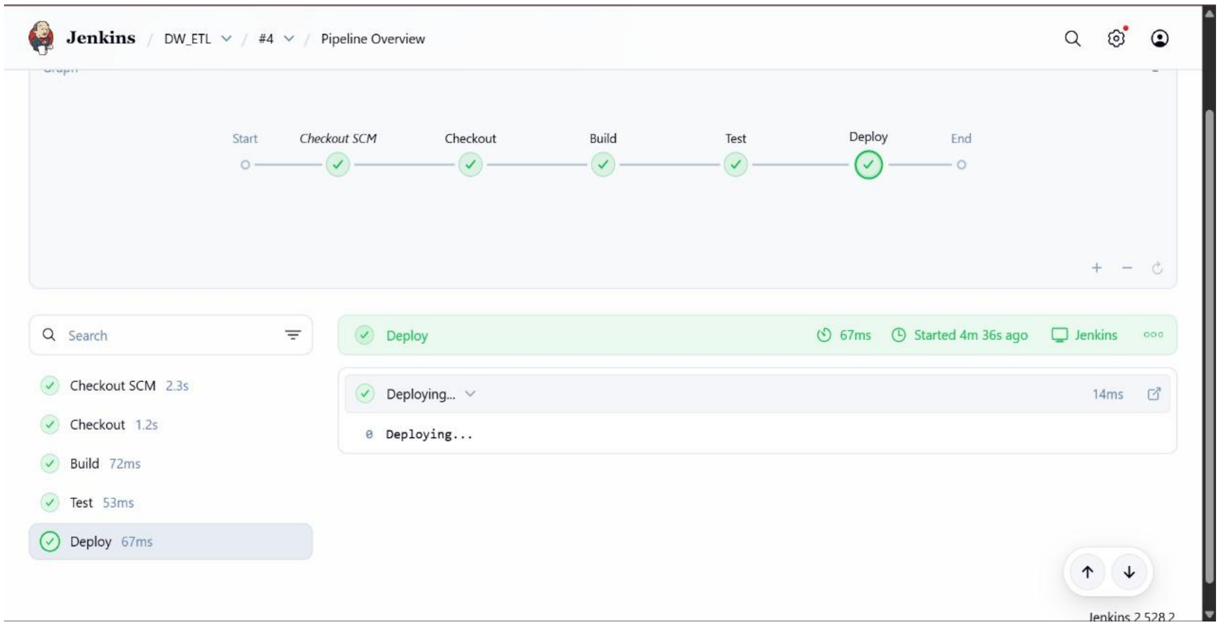
```
Functions:

        Dimaccount_TimerTrigger: timerTrigger

        Dimcustomer_TimerTrigger: timerTrigger

        EventGridTrigger: eventGridTrigger

        FraudDetection_EventGridTrigger: eventGridTrigger

        ServiceBusQueueTrigger: serviceBusTrigger

For detailed output, run func with --verbose flag.
```


Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
mazenetfc1@synergifics...
SYNERGIFIC SOFTWARE PRIVATE ...
Home > arunservicebus | Queues > fraud-alerts (arunservicebus/fraud-alerts)
Service Bus Queue
Search
Overview
Access control (IAM)
Diagnose and solve problems
Service Bus Explorer
Resource visualizer
Settings
Shared access policies
Properties
Locks
Automation
Tasks
Export template
Help
Add or remove favorites by pressing Ctrl+Shift+F
Message Body
Message Properties
Fit message body
Off

fraud-alerts (arunservicebus/fraud-alerts) | Service Bus Explorer

```
[
 {
  "id": "event1",
  "eventType": "Notification",
  "subject": "fraud-test",
  "eventTime": "2025-12-09T14:00:00Z",
  "data": {
   "TransactionID": "ATM000001",
   "ATMID": "ATM013",
   "AccountNumber": 1002003352,
   "CustomerID": "CUST372",
   "TransactionAmount": 70000,
   "TransactionType": "WITHDRAWAL",
   "TransactionTime": "2025-01-01T00:00:00Z",
```

Step 2: In this task focuses on implementing complete automation for deploying analytics and integration components. PySpark notebooks are automatically synchronized and deployed to Databricks or Synapse using Workspace APIs. Database schema changes for the warehouse are automated through tools like Flyway.

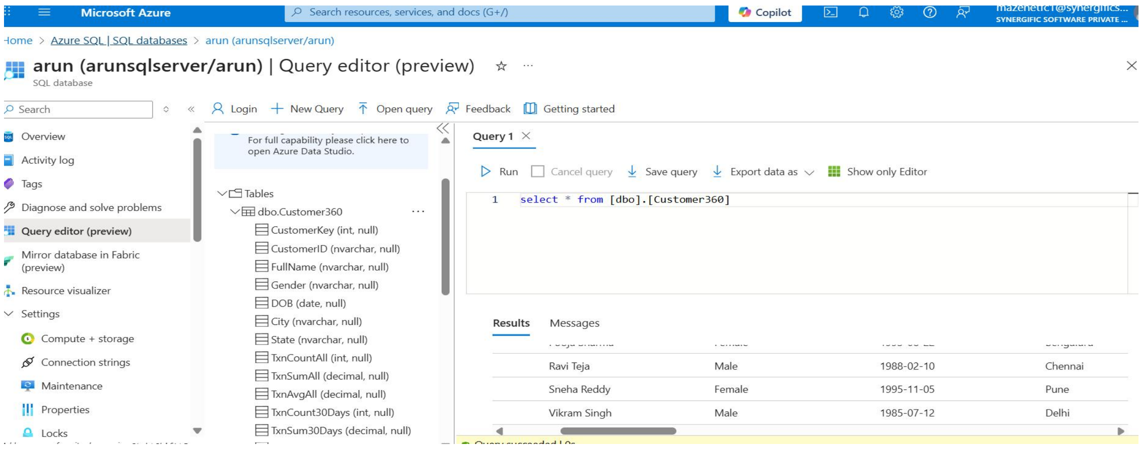


```
0 Deploying...
```

Day 5 Synopsis :

Tool used : Power BI Desktop.

Step 1: In this task, I created a unified Customer 360 model that brings together all key customer information in one place. It merges transaction history from the data warehouse, demographic details from DimCustomer, and login/device activity from Cosmos DB. This combined dataset gives a full view of customer behaviour across channels. Using this unified view, I can calculate balance trends, spending habits, credit usage, and overall customer risk scores.



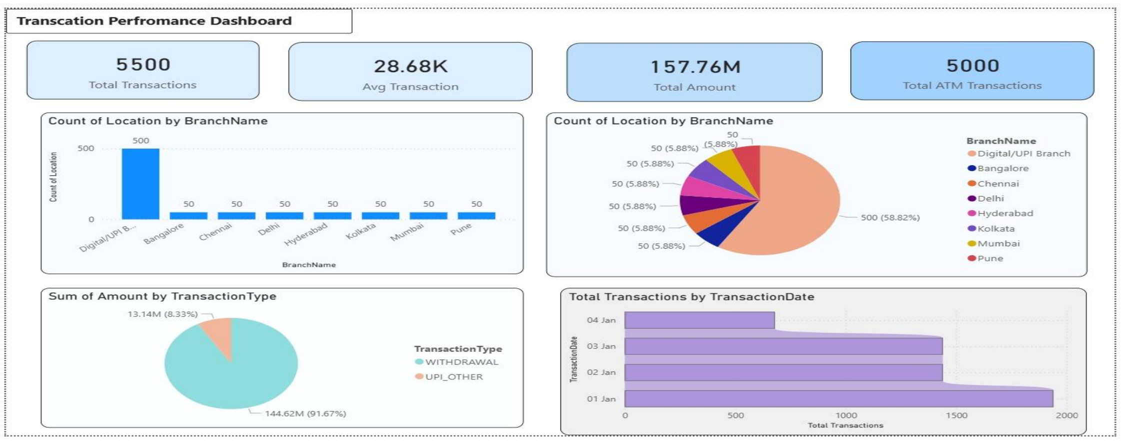
For full capability please click here to open Azure Data Studio.

Tables
- dbo.Customer360
  - CustomerKey (int, null)
  - CustomerID (nvarchar, null)
  - FullName (nvarchar, null)
  - Gender (nvarchar, null)
  - DOB (date, null)
  - City (nvarchar, null)
  - State (nvarchar, null)
  - TxnCountAll (int, null)
  - TxnSumAll (decimal, null)
  - TxnAvgAll (decimal, null)
  - TxnCount30Days (int, null)
  - TxnSum30Days (decimal, null)

```
select * from [dbo].[Customer360]
```

Ravi Teja	Male	1988-02-10	Chennai
Sneha Reddy	Female	1995-11-05	Pune
Vikram Singh	Male	1985-07-12	Delhi

Step 2: Using Power BI dashboards were designed to transform warehouse data into meaningful insights for business users. These dashboards cover areas like branch performance, daily transaction volumes, and ATM/UPI operational trends. Each report includes interactive visuals for filtering, drilling down, and analyzing patterns over time. Fraud analytics dashboards help teams monitor alert types, detection patterns, and prevent financial losses.


Transcation Perfromance Dashboard
Total Transactions
28.68K
Avg Transaction
157.76M
Total Amount
Total ATM Transactions
Count of Location by BranchName
Count of Location
Digital/UPI B...
Bangalore
Chennai
Delhi
Hyderabad
Kolkata
Mumbai
Pune
BranchName
Count of Location by BranchName
50 (5.88%)
500 (58.82%)
BranchName
Digital/UPI Branch
Bangalore
Chennai
Delhi
Hyderabad
Kolkata
Mumbai
Pune
Sum of Amount by TransactionType
13.14M (8.33%)
144.62M (91.67%)
TransactionType
WITHDRAWAL
UPI_OTHER
Total Transactions by TransactionDate
TransactionDate
04 Jan
03 Jan
02 Jan
01 Jan
Total Transactions