

Connected Vehicle Data Integration - - - - - Geetika Gupta

- Industry: Earth**
- Project Title:** Connected Vehicle Data Integration
- Problem Statement/Opportunity:** This project falls into the "Earth" category, as it focuses on implementing solutions to address climate change by optimizing fleet management practices, improving vehicle performance, and reducing emissions through sustainable data integration and analysis , aligning with the commitment to environmental responsibility .

This project aims to leverage technology to not only enhance the performance of G-Motors' heavy vehicle fleet but also minimize its environmental footprint.

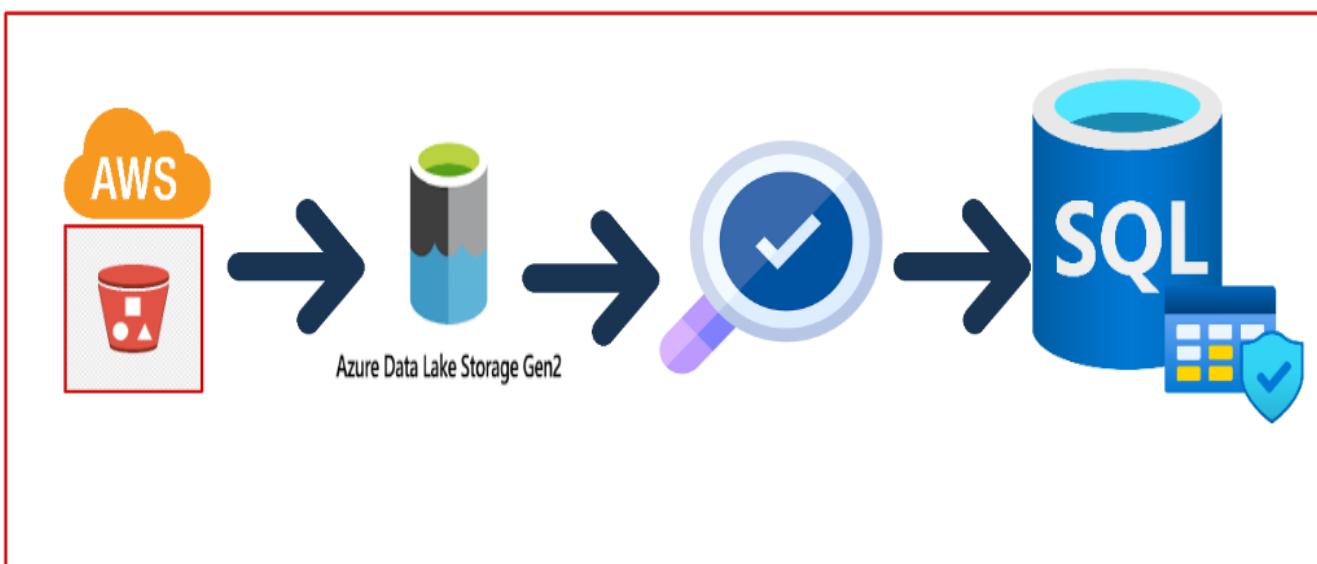
The project addresses the challenges we face in seamlessly integrating telemetry data from third-party IoT devices hosted on AWS into our Azure cloud environment. This initiative is poised to revolutionize the fleet management practices, making them more sustainable, data-driven, and future-ready.

- Core Azure Services:** Azure Data Factory (ADF), Azure SQL Database,Azure Function App
- Additional Azure Services:** Azure Key Vault , Azure Data Lake Storage Gen 2

□ **Project Description:** G-Motors (GM) relies on third-party IoT devices hosted on AWS for valuable telemetry data. However, the absence of a streamlined data integration and validation process obstructs our ability to derive actionable insights from this data. The core challenge lies in the efficient transfer, validation, and storage of this data within GM's Azure cloud environment.

We must ensure the highest standards of data quality, security, and scalability to leverage this data for informed decision-making, improved vehicle performance, and optimized fleet management.

Our project aims to overcome these challenges, paving the way for a more sustainable future and solidifying our commitment to environmental stewardship.



Project Objectives:

Data Ingestion and Transfer:

- Ingest JSON telemetry data from third-party IoT devices hosted on AWS cloud.
- Transfer the telemetry data from AWS to General Motors' Azure cloud environment.

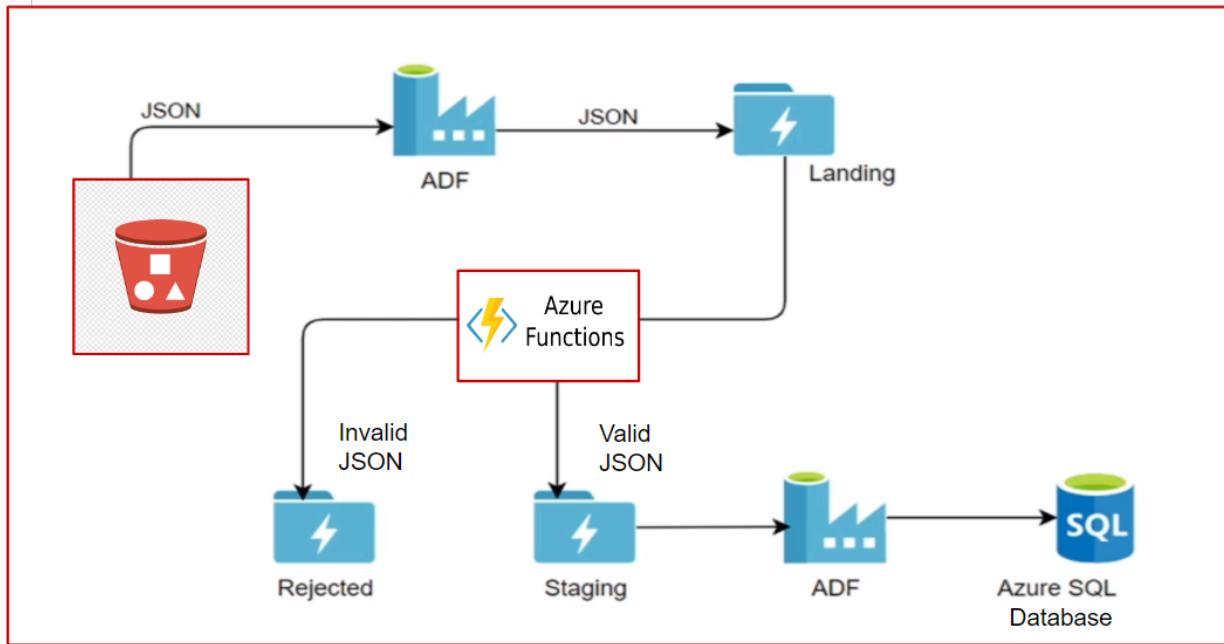
Data Validation:

- Validate the incoming JSON data to ensure it adheres to the expected format and standards.
- Implement data quality checks and error handling mechanisms.

Data Storage and Analysis:

- Store the validated telemetry data in an Azure SQL Database.
- Enable data indexing and optimization for efficient querying and analysis.
- Implement mechanisms for data retention and archiving.

ARCHITECTURE DIAGRAM



Project Phases:

Data Ingestion and Transfer:

- Identify AWS IoT Data Sources:
- Set Up Data Ingestion: Create data ingestion pipelines using Azure Data Factory to transfer data from AWS to Azure securely. I have stored the access key and secret access key from AWS S3, in Azure Key Vault, as best practice.
- Set up an Azure Data Factory pipeline using copy activity, to ingest data from the third-party IoT devices hosted on AWS. copy activity is used to move data from AWS S3 buckets to Azure Storage ADLS Gen2 input/landing folder.

s3://iotdata-pro1-s3/2023/09/08/

The screenshot shows the AWS S3 console at the URL <https://s3.console.aws.amazon.com/s3/upload/iotdata-pro1-s3?region=us-east-1&prefix=2023/09/08/>. A green header bar indicates "Upload succeeded". Below it, a summary table shows the destination as "s3://iotdata-pro1-s3/2023/09/08/" with 1 file successfully uploaded (85.2 KB) and 0 files failed. The "Files and folders" tab is selected, displaying a table with one item: "Customer_Valid1_pro..." which is an application/json file of size 85.2 KB and status "Succeeded".

The screenshot shows the Azure Data Factory pipeline editor for a pipeline named "pipelineadls2sqldb". A specific activity named "Ingestion_S3_to_A..." is selected. The "Source" tab is active, showing a "Copy data" configuration. The "Source dataset" is set to "Json1_DS1". Under "Dataset properties", "FolderPath" is set to "@concat(formatDateTime(utcNow(),'...', '...'))". The "File path type" is set to "Wildcard file path" with the path "iotdata-pro1-s3 / Wildcard folder path / *.json". The "Start time (UTC)" and "End time (UTC)" fields are empty.

Data Landing Zone:

- Create a landing folder in Azure Storage to temporarily store the incoming JSON data files. You can organize this storage account with a container specifically for landing data.
- input/landing

The screenshot shows the Microsoft Azure Storage Container Overview page for a container named 'input'. The top navigation bar includes the Microsoft Azure logo, a search bar, and various navigation links. The main content area displays the container's settings, including its name ('input'), location ('input / landing'), authentication method ('Access key'), and access control ('Switch to Azure AD User Account'). A 'Search blobs by prefix (case-sensitive)' field is also present. On the left, there is a sidebar with 'Overview' selected, along with other options like 'Diagnose and solve problems' and 'Access Control (IAM)'. Below the sidebar is a 'Settings' section with links to 'Shared access tokens', 'Manage ACL', 'Access policy', 'Properties', and 'Metadata'. The main right-hand pane lists blobs in the container, showing columns for Name, Modified, Access tier, Archive status, and Blob type. One blob entry is visible: a file named '[..]' with a yellow icon.

Azure Function for Validation:

- Develop an Azure Function that uses a storage-based trigger (Blob Trigger-BlobTrigger1) to monitor the landing folder for incoming JSON files.
- When a new file arrives, the Azure Function is triggered to validate the JSON format and content. If validation fails, move the file to the "rejectedFolder" folder; otherwise, move it to the "stagingFolder" folder.

FunctionApp1GG Function App

Storage is not configured properly. Function scaling will be limited. Click to learn more.

Resource group (move)	: iotDataProject1	URL	: https://functionapp1gg.azurewebsites.net
Status	: Running	Operating System	: Windows
Location (move)	: East US	App Service Plan	: ASP-iotDataProject1-9557 (Y1:0)
Subscription (move)	: Free Trial	Properties	: See More
Subscription ID	: 6cfa0b74-1fe2-40a3-8ff7-98c4d8b50d76	Runtime version	: 4.25.2.21138

Functions

Name	Trigger	Status	Monitor
BlobTrigger1	Blob	Enabled	Invocations and more

Deployment

Settings

Type here to search

Windows Start button

28°C ENG IN 11:49 AM 9/8/2023

Data Validation and Movement:

- Configure the Azure Function to perform JSON validation using built-in JSON parsing libraries and validation logic to check the correctness of JSON files.
- Use Azure Function bindings to move files between folders based on the validation result. If validation passes, move the file to the "stagingFolder" folder; if it fails, move it to the "rejectedFolder" folder.

```
module.exports = async function (context, myBlob) {  
  
    context.log("JavaScript blob trigger function processed  
blob \n Blob:");  
  
    context.log("*****Azure Function Started*****");  
  
    var result =true;  
  
    try{  
  
        context.log(myBlob.toString());  
  
        JSON.parse(myBlob.toString().trim().replace('\n', ' '));  
  
    }catch(exception){  
  
        context.log(exception);  
  
        result =false;  
  
    }  
  
    if(result){
```

```

    context.bindings.stagingFolder = myBlob.toString();

    context.log("*****File Copied to Staging Folder
Successfully*****");

} else{

    context.bindings.rejectedFolder = myBlob.toString();

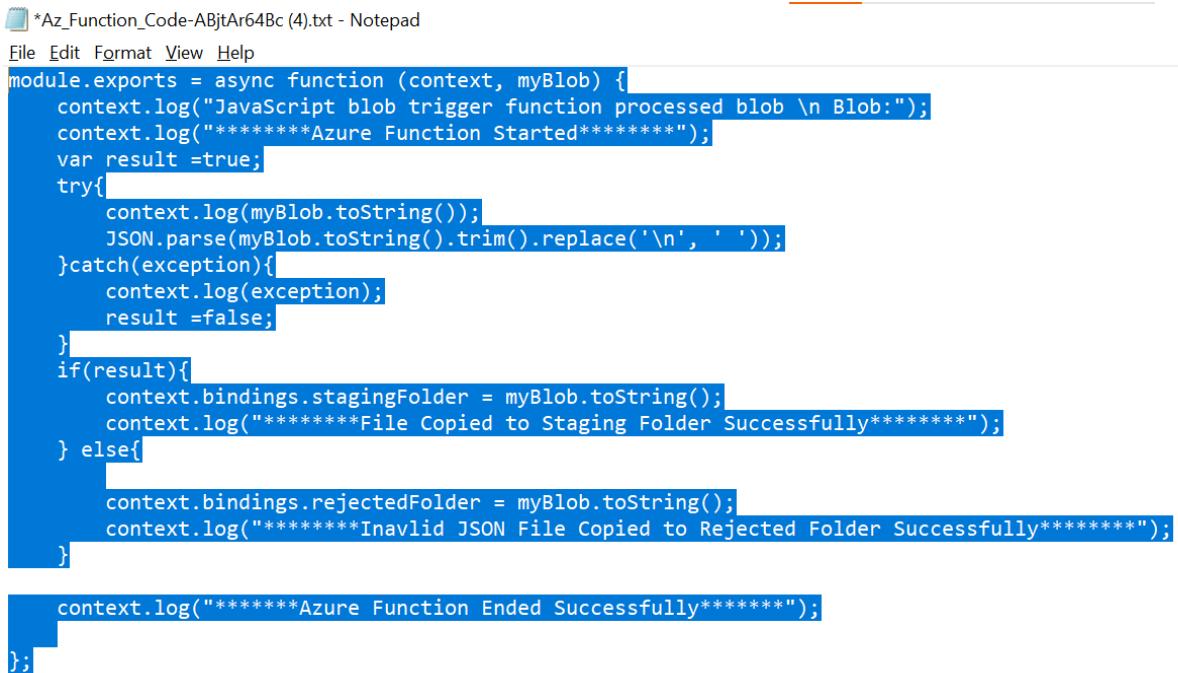
    context.log("*****Inavlid JSON File Copied to
Rejected Folder Successfully*****");

}

context.log("*****Azure Function Ended
Successfully*****");

};


```



```

*Az_Function_Code-ABjtAr64Bc (4).txt - Notepad
File Edit Format View Help
module.exports = async function (context, myBlob) {
    context.log("JavaScript blob trigger function processed blob \n Blob:");
    context.log("*****Azure Function Started*****");
    var result =true;
    try{
        context.log(myBlob.toString());
        JSON.parse(myBlob.toString().trim().replace('\n', ' '));
    }catch(exception){
        context.log(exception);
        result =false;
    }
    if(result){
        context.bindings.stagingFolder = myBlob.toString();
        context.log("*****File Copied to Staging Folder Successfully*****");
    } else{
        context.bindings.rejectedFolder = myBlob.toString();
        context.log("*****Inavlid JSON File Copied to Rejected Folder Successfully*****");
    }
}
context.log("*****Azure Function Ended Successfully*****");
};


```

Staging Data in Azure SQL DB:

- Create an Azure SQL Database to serve as the staging area for your data. Design the database schema to accommodate the JSON data structure.
- Develop another ADF Pipeline that triggers when files are moved to the "stagingFolder" folder. This function reads the JSON data, transforms it if needed, and inserts it into the Azure SQL Database.

The screenshot displays two windows side-by-side. On the left is the 'Query editor (preview)' window for the database 'sqlldb-proj1'. It shows the table 'dbo.VehicleData1' with its schema. On the right is the 'Azure Data Factory' pipeline editor, showing a 'Copy data' activity within a pipeline named 'Ingestion_S3_to_AzureSQLDB'.

Monitoring and Logging:

- Implement monitoring and logging in Azure Functions to track the validation process, successful data movements, and any errors encountered.

Security and Access Control:

- Ensure proper security and access control mechanisms are in place, including appropriate permissions for Azure Storage, Azure SQL Database, and Azure Functions.

PIPELINE RUNS:-

The screenshot shows the Microsoft Azure Data Factory interface for the 'Pipeline runs' section. The left sidebar includes options like Dashboards, Runs, Pipeline runs (selected), Trigger runs, Change Data Capture, Runtimes & sessions, Integration runtimes, Data flow debug, Notifications, and Alerts & metrics. The main area displays a table of pipeline runs with columns for Pipeline name, Run start, Run end, Duration, Triggered by, and Status. All runs listed have a status of 'Succeeded'. The table shows 10 items, with the most recent run starting at 9/8/2023, 12:27:19 PM and ending at 9/8/2023, 12:27:30 PM.

Pipeline name	Run start	Run end	Duration	Triggered by	Status
pipelineadls2sqldb	9/8/2023, 12:27:19 PM	9/8/2023, 12:27:30 PM	12s	trigger1_json_stagin...	Succeeded
pipelineadls2sqldb	9/8/2023, 12:27:09 PM	9/8/2023, 12:29:01 PM	1m 52s	trigger1_json_stagin...	Succeeded
pipelineadls2sqldb	9/8/2023, 11:14:37 AM	9/8/2023, 11:14:49 AM	12s	trigger1_json_stagin...	Succeeded
pipelineadls2sqldb	9/8/2023, 11:14:28 AM	9/8/2023, 11:14:38 AM	11s	trigger1_json_stagin...	Succeeded
pipelineadls2sqldb	9/8/2023, 10:37:27 AM	9/8/2023, 10:37:45 AM	19s	trigger1_json_stagin...	Succeeded
pipelineadls2sqldb	9/8/2023, 10:37:17 AM	9/8/2023, 10:37:31 AM	14s	trigger1_json_stagin...	Succeeded
pipelineadls2sqldb	9/8/2023, 10:37:06 AM	9/8/2023, 10:37:19 AM	13s	trigger1_json_stagin...	Succeeded
pipelineadls2sqldb	9/8/2023, 10:36:56 AM	9/8/2023, 10:37:08 AM	12s	trigger1_json_stagin...	Succeeded
pipelineadls2sqldb	9/8/2023, 10:22:03 AM	9/8/2023, 10:22:13 AM	11s	trigger1_json_stagin...	Succeeded

TRIGGER RUNS

Trigger runs

Trigger name ↑↓	Trigger type	Trigger time ↑↓	Status ↑↓	Pipelines	Run	Message
trigger1_json_staging_sqldb	Storage events tri...	9/8/2023, 12:27:19 PT	✓ Succeeded	1	Original	
trigger1_json_staging_sqldb	Storage events tri...	9/8/2023, 12:27:08 PT	✓ Succeeded	1	Original	
trigger1_json_staging_sqldb	Storage events tri...	9/8/2023, 11:14:37 AI	✓ Succeeded	1	Original	
trigger1_json_staging_sqldb	Storage events tri...	9/8/2023, 11:14:27 AI	✓ Succeeded	1	Original	
trigger1_json_staging_sqldb	Storage events tri...	9/8/2023, 10:37:27 AI	✓ Succeeded	1	Original	
trigger1_json_staging_sqldb	Storage events tri...	9/8/2023, 10:37:16 AI	✓ Succeeded	1	Original	
trigger1_json_staging_sqldb	Storage events tri...	9/8/2023, 10:37:06 AI	✓ Succeeded	1	Original	
trigger1_json_staging_sqldb	Storage events tri...	9/8/2023, 10:36:56 AI	✓ Succeeded	1	Original	
trigger1_json_staging_sqldb	Storage events tri...	9/8/2023, 10:22:02 AI	✓ Succeeded	1	Original	
trigger1_json_staging_sqldb	Storage events tri...	9/8/2023, 10:21:52 AI	✓ Succeeded	1	Original	

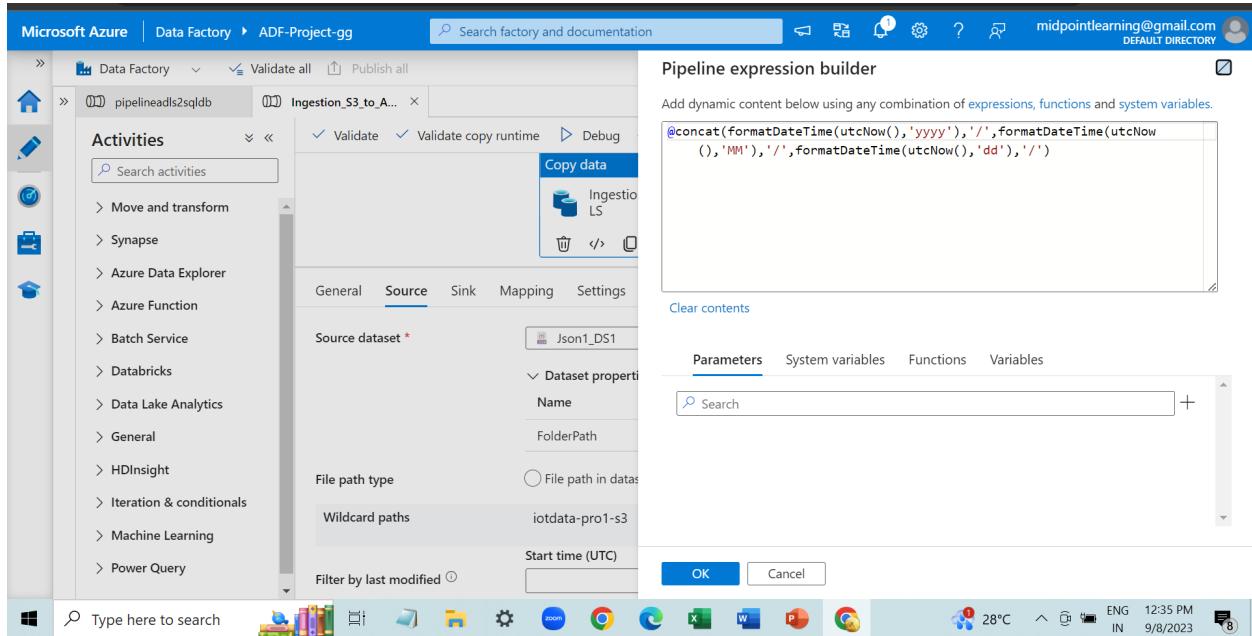
Ingestion_S3_to_ADLS PIPELINE RUN:-

Ingestion_S3_to_A... x

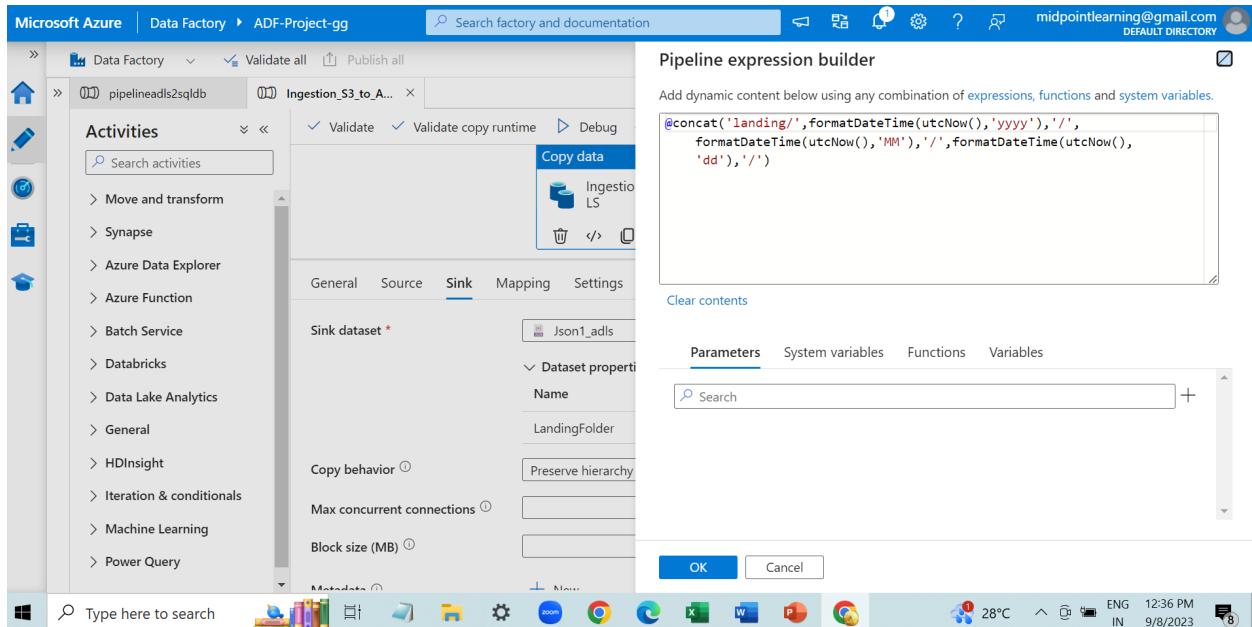
Output

Activity name ↑↓	Activity status ↑↓	Run start ↑↓	Duration ↑↓	Integration runtime	User properties ↑↓	Act
Ingestion_S3_to_ADLS	✓ Succeeded	9/8/2023, 12:26:45 PM	12s	AutoResolveIntegratio		3ac

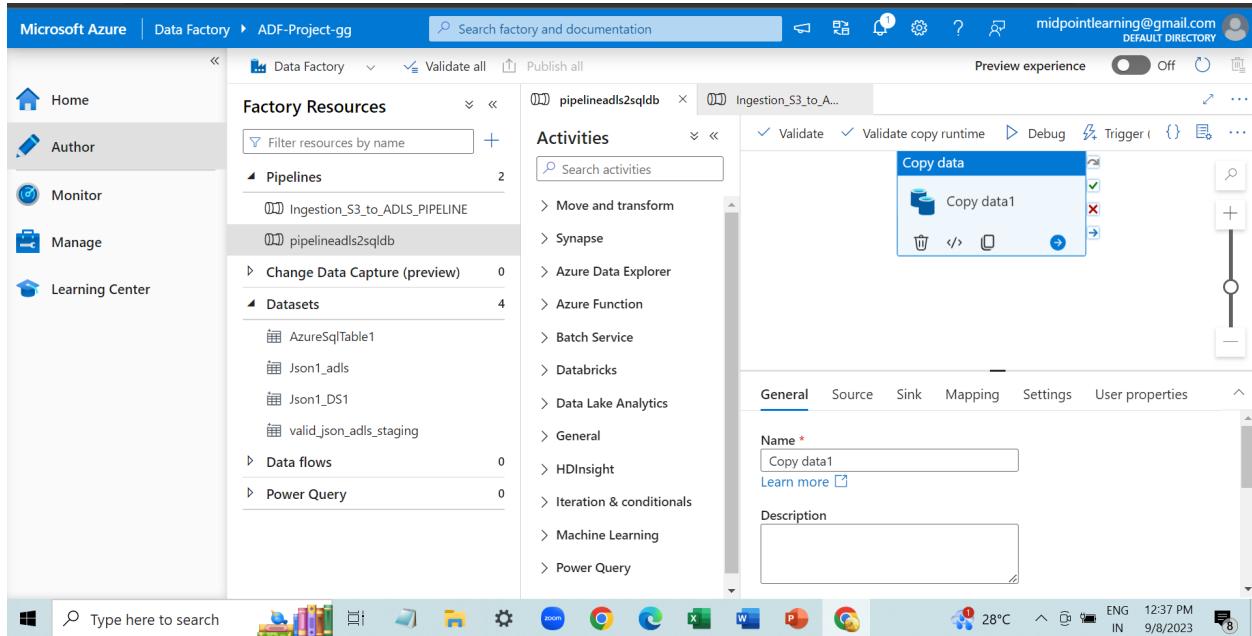
SOURCE:-



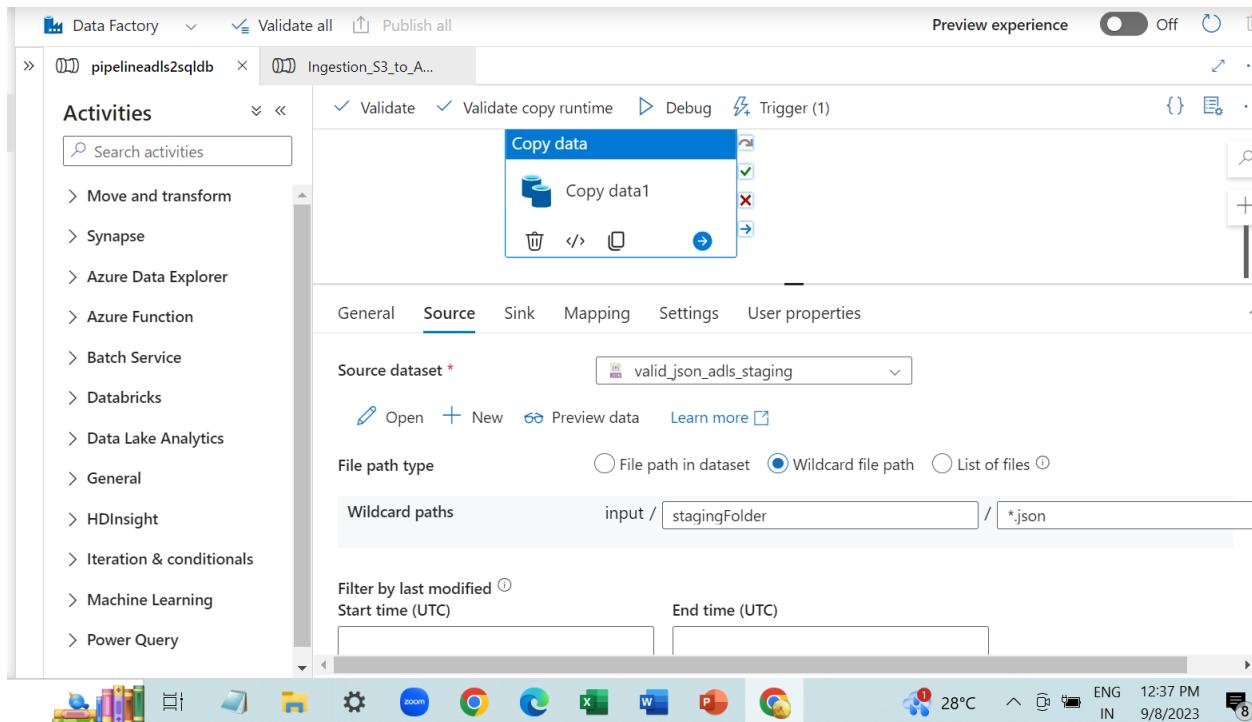
SINK:-



PIPELINE 2:- ADLS TO SQLDB



SOURCE



SINK

The screenshot shows the Azure Data Factory pipeline editor. A pipeline named "pipelineadls2sqldb" is open, and an activity named "Ingestion_S3_to_A..." is selected. The "Activities" pane on the left lists various data movement and transformation options. The main workspace displays the configuration for the "Copy data" activity. The "Sink" tab is active, showing the destination dataset as "AzureSqlTable1". The "Write behavior" is set to "Insert". Other settings include "Bulk insert table lock" set to "No" and "Table option" set to "None". The pipeline name "pipelineadls2sqldb" and activity name "Ingestion_S3_to_A..." are visible at the top of the editor.

BLOB EVENT TRIGGER:-

The screenshot shows the Microsoft Azure portal interface. On the left, the navigation menu includes Home, Author, Monitor, Manage, and Learning Center. The main content area shows a pipeline named "pipelineadls2sqldb" with an activity named "Ingestion_S3_to_A...". On the right, a modal dialog titled "Edit trigger" is open, allowing the configuration of a new trigger. The trigger is named "trigger1_json_staging_sqldb" and is of type "BlobEventsTrigger". It is configured to trigger from an Azure subscription (Free Trial) using a storage account named "proj1adls" and a container named "inout". The "Continue" button is visible at the bottom of the dialog.

ADLS CONTAINER AFTER PIPELINE RUN:-

Microsoft Azure Search resources, services, and docs (G+)

Home > proj1adls | Containers >

input Container

Search < Upload + Add Directory Refresh | Rename Delete Change tier Acquire lease Break lease Give feedback

Overview Authentication method: Access key (Switch to Azure AD User Account)
Location: input

Diagnose and solve problems
Access Control (IAM)

Settings Shared access tokens Manage ACL Access policy Properties Metadata

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
landing					-	...
rejectedFolder					-	...
stagingFolder					-	...

Search blobs by prefix (case-sensitive) Show deleted objects

Type here to search 28°C ENG IN 12:39 PM 9/8/2023

PIPELINE 1 SUCCESSFUL:-

Microsoft Azure Data Factory > ADF-Project-gg Search factory and documentation

Details Refresh

Learn more on copy performance details from here.

Activity run id: 3ad521d6-03b2-462c-9060-e75c2961c1d9

Amazon S3 → Azure Data Lake Storage Gen2 Region: East US

Succeeded

Data read: 87.295 KB Data written: 87.295 KB
Files read: 1 Files written: 1
Peak connections: 1 Peak connections: 1

Copy duration: 00:00:09 Throughput: 29.098 KB/s

Amazon S3 → Azure Data Lake Storage Gen2
Start time: 9/8/2023, 12:26:46 PM Used DIUS: 4

How satisfied or dissatisfied are you with the performance of this copy activity?

Type here to search 28°C ENG IN 12:40 PM 9/8/2023

INPUT/LANDING FOLDER FILE uploaded:-

The screenshot shows the Microsoft Azure Storage Explorer interface. The left sidebar shows the 'input' container under 'Containers'. The main area displays a table of blobs:

	Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
<input type="checkbox"/>	[..]						...
<input type="checkbox"/>	Customer_Valid1_pro1.json	9/8/2023, 12:26:56 PM	Hot (Inferred)		Block blob	85.25 KIB	Available

The taskbar at the bottom shows various application icons and system status.

INPUT/STAGING FOLDER RECEIVES FILE

The screenshot shows the Microsoft Azure Storage Explorer interface. The left sidebar shows the 'input/stagingFolder' container under 'Containers'. The main area displays a table of blobs:

	Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
<input type="checkbox"/>	[..]						...
<input type="checkbox"/>	fc57d36-1648-49e5-a9c9-aaf...	9/8/2023, 12:27:05 PM	Hot (Inferred)		Block blob	85.25 KIB	Available

The taskbar at the bottom shows various application icons and system status.

AZURE SQL DB:-

The screenshot shows the Microsoft Azure portal interface for a SQL database named 'sqlldb-proj1'. The left sidebar displays the database structure with a 'Tables' section expanded, showing the 'VehicleData' table containing columns: VehicleID, latitude, longitude, City, temeprature, and speed. The main area is a 'Query editor (preview)' window titled 'Query 1'. It contains the following SQL code:

```

1 select count(*) from [dbo].[VehicleData]
2
3 delete from VehicleData
4
5 select * from VehicleData

```

The results pane below shows the data from the 'VehicleData' table:

VehicleID	latitude	longitude	City	temeprature	speed
L795767537662365606	2	171	Höchst	3	86
ES3718234375365266902488	-22	59	Binjal	63	52
F152643097278235	-89	-77	Nicoya	38	127
FR9759633173944815188083...	-72	5	Hanau	98	165
SMAG0771421011365586d9111...	71	103	Melton	22	128

A status message at the bottom of the results pane says 'Query succeeded | 1s'.

Microsoft Azure Upgrade Search resources, services, and docs (G+)

Home > IoTDataProject1

iotDataProject1 | Deployments

Resource group

Refresh Cancel Redeploy Delete View template

Filter by deployment name or resources in the deployment...

Deployment name	Status	Last modified	Duration	Related events
Microsoft.SQLDatabase.newDatabaseNewServer_3...	Succeeded	9/7/2023, 2:10:52 PM	2 minutes, 14 seconds, 681 milliseconds	Related events
Microsoft.Web-Function-28d5d02d-b19f	Succeeded	9/6/2023, 11:08:25 PM	16 seconds, 233 milliseconds	Related events
Failure-Anomalies-Alert-Rule-Deployment-7dfcba83	Succeeded	9/6/2023, 10:42:28 PM	1 second, 282 milliseconds	Related events
Microsoft.Web-FunctionApp-Portal-c21fdcba-85c6	Succeeded	9/6/2023, 10:32:45 PM	36 seconds, 144 milliseconds	Related events
Failure-Anomalies-Alert-Rule-Deployment-5cb67f7a	Succeeded	9/6/2023, 5:57:26 PM	7 seconds, 169 milliseconds	Related events
Microsoft.Web-Function-d1300431-8156	Succeeded	9/6/2023, 5:50:46 PM	17 seconds, 454 milliseconds	Related events
Microsoft.Web-FunctionApp-Portal-168f4b64-851c	Succeeded	9/6/2023, 5:47:39 PM	1 minute, 10 seconds, 648 milliseconds	Related events
newKeyVaultS3	Succeeded	9/6/2023, 12:02:00 PM	25 seconds, 954 milliseconds	Related events
iotproj1KeyVault	Succeeded	9/6/2023, 10:47:59 AM	27 seconds, 749 milliseconds	Related events
Microsoft.DataFactory-20230906102339	Succeeded	9/6/2023, 10:26:19 AM	13 seconds, 87 milliseconds	Related events
proj1adls_1693975708678	Succeeded	9/6/2023, 10:19:19 AM	32 seconds, 198 milliseconds	Related events



iotDataProject1 - Resource Group | Deployments

Microsoft Azure Upgrade Search resources, services, and docs (G+) midpointlearning@gmail.com DEFAULT DIRECTORY

Home > **iotDataProject1** Overview ...

Essentials

Resources Recommendations

Filter for any field... Type equals all Location equals all Add filter

Showing 1 to 12 of 12 records. Show hidden types

Name	Type	Location
Failure Anomalies - FunctionApp1GG	Smart detector alert rule	Global
FunctionApp1GG	Function App	East US
FunctionApp1GG	Application Insights	East US
iotprojKeyVault	Key vault	East US
newKeyVault\$	Key vault	East US
proj1-sqldb1	SQL server	East US
proj1ads	Storage account	East US
sqldb-proj1 (proj1-sqldb1/sqldb-proj1)	SQL database	East US

< Previous Page 1 of 1 Next >

Give feedback

https://portal.azure.com/#blade/Microsoft_Azure_Billing/SubscriptionUpgra...

Type here to search

Microsoft Azure Upgrade Search resources, services, and docs (G+) midpointlearning@gmail.com DEFAULT DIRECTORY

Home > Recent > **iotDataProject1** > **FunctionApp1GG** Application Insights

Overview

Resource group (move) : [iotDataProject1](#)
 Location : East US
 Subscription (move) : [Free Trial](#)
 Subscription ID : 6cfa0b74-1fe2-40a3-8ff7-98c4d8b50d76
 Tags (edit) : Add tags

Instrumentation Key : 718f63a5-50a6-482c-97c7-1ed3d9f16689
 Connection String : InstrumentationKey=718f63a5-50a6-482c-97c7-1ed3d9f16689;IngestionEndpoint...
 Workspace : DefaultWorkspace-6cfa0b74-1fe2-40a3-8ff7-98c4d8b50d76-EUS

Show data for last: 30 minutes 1 hour 6 hours 12 hours 1 day 3 days 7 days 30 days

Failed requests

Server response time

Server requests

Type here to search

FUNCTIONS APPLICATION INSIGHTS

This architecture allows us to effectively manage incoming IoT telemetry data by landing it in Azure Storage, validating it using Azure Functions, and then moving validated data to a staging area in Azure SQL Database for further analysis and processing. Data that doesn't meet validation criteria is segregated into a "rejected" folder for review and troubleshooting.

Thankyou-----
