AZURE DATA FACTORY

Azure Data Factory (ADF) is a cloud-based data integration service that allows you to create, schedule, and orchestrate ETL (Extract, Transform, Load) and ELT (Extract, Load, Transform) workflows. It is widely used for moving data between various sources and destinations in Azure and on-premises environments.

Key Features of Azure Data Factory

1. **Data Integration** – Connects to multiple data sources, including Azure services, on-prem databases, SaaS applications, and cloud storage.
2. **Data Transformation** – Uses **Mapping Data Flows** or **Azure Databricks** for transforming data before loading it into a destination.
3. **Data Orchestration** – Schedules and automates workflows across different services.
4. **Monitoring & Logging** – Provides real-time monitoring and error-handling capabilities.
5. **Scalability** – Supports big data workloads and high-performance parallel processing.

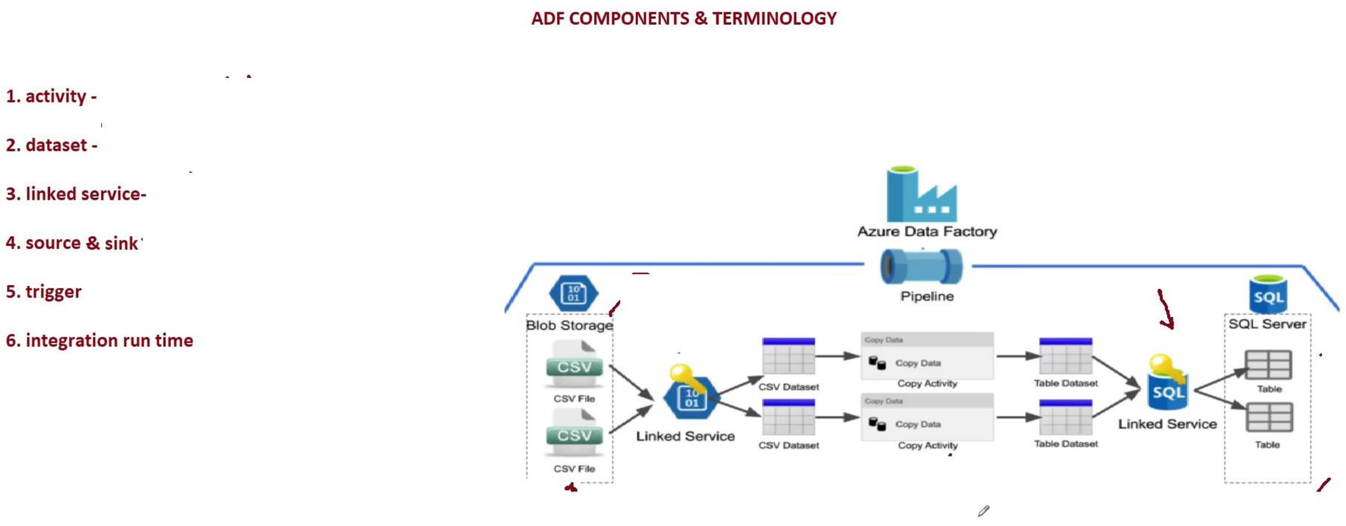
Core Components of ADF

1. **Pipelines** – A logical grouping of activities that perform a data workflow.
2. **Activities** – Tasks like data movement, transformation, or control flow (e.g., executing stored procedures).
3. **Datasets** – References to data stored in linked services (e.g., Azure Blob Storage, SQL Database).
4. **Linked Services** – Connectors to various data stores (e.g., Azure SQL, Amazon S3, SAP, Oracle).
5. **Integration Runtimes (IR)** – Compute infrastructure that executes data flows. There are three types:
   - **Azure IR** – For cloud-based transformations.
   - **Self-hosted IR** – For on-prem data movement.
   - **SSIS IR** – For running SSIS packages in Azure.

Common Use Cases

☑ Data migration from on-prem to cloud
☑ ETL & ELT pipeline creation

☑ Data warehousing (e.g., loading data into Azure Synapse Analytics)
☑ Incremental data load and CDC (Change Data Capture)
☑ IoT and streaming data processing



ADF COMPONENTS & TERMINOLOGY

1. activity -
2. dataset -
3. linked service-
4. source & sink
5. trigger
6. integration run time

ADLS GEN2 (**Azure Data Lake Storage Gen2)**

Azure Data Lake Storage Gen2 (ADLS Gen2) – Overview & Setup Guide

What is ADLS Gen2?

Azure Data Lake Storage Gen2 (ADLS Gen2) is an advanced storage solution built on **Azure Blob Storage**, designed specifically for **big data analytics** and **Hadoop-compatible workloads**. It supports hierarchical namespace features and is optimized for **high-performance data lake solutions**.

🔷 **Key Features of ADLS Gen2**

☑ **Hierarchical Namespace** – Organizes data into directories for better performance.
☑ **Hadoop-Compatible** – Works with **Azure Synapse, Databricks, HDInsight, and Apache Spark**.
☑ **Scalable & Secure** – Supports RBAC (Role-Based Access Control) and ACLs (Access Control Lists).
☑ **Cost-Effective** – Lower cost for storing massive amounts of structured/semi-structured data.
☑ **Optimized for Analytics** – Works with **Azure Data Factory, Azure Databricks, and Power BI**.

1️⃣ How to Create an ADLS Gen2 Storage Account

Step 1: Log in to Azure Portal

🔗 Go to Azure Portal and sign in.

Step 2: Create a Storage Account

1. Search for **"Storage Accounts"** in the Azure search bar.
2. Click **Create**.

Step 3: Configure Basic Settings

1. **Subscription** – Select your Azure subscription.
2. **Resource Group –** Choose an existing one or create a new one.
3. **Storage Account Name –** Provide a unique name (e.g., adlsgen2mystore).
4. **Region** – Select the nearest Azure data center.
5. **Performance** – Choose **Standard** (for most use cases) or **Premium** (for low-latency workloads).
6. **Redundancy** – Select one (LRS, ZRS, GRS, RA-GRS).
7. **Enable Hierarchical Namespace** ✅ (This is required for ADLS Gen2).

Step 4: Review and Create

1. Click **Review + Create**.
2. Once validation passes, click **Create**.
3. Wait for deployment to complete.

2️⃣ How to Create & Access a Data Lake Container

Step 1: Navigate to Storage Account

1. Open your **Storage Account** in the Azure portal.
2. Click on **Containers** (under Data storage).
3. Click **+ Container**, give it a name (e.g., datalakefiles), and set access level as **Private**.

Step 2: Upload & Manage Data

1. Click on the container you created.
2. Click **Upload** to add files/folders.
3. Use **Azure Storage Explorer** or **AzCopy** for bulk data transfers.

3️⃣ Secure & Manage Access for ADLS Gen2

Option 1: Using Access Control (ACLs)

1. Open the **container** in your storage account.
2. Click **Manage Access**.
3. Assign **Read, Write, Execute** permissions to users/groups.

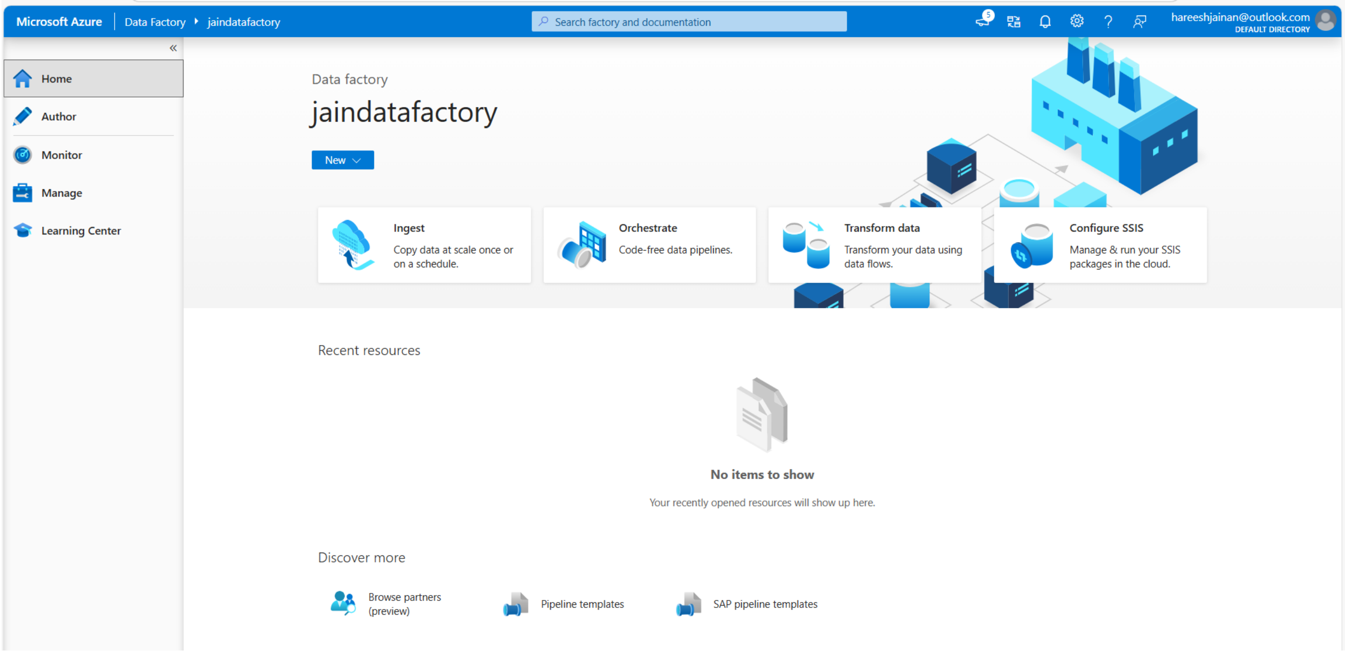
Option 2: Using Azure RBAC (Role-Based Access Control)

1. Go to the **Storage Account → Access Control (IAM)**.
2. Click **Add Role Assignment**.
3. Select roles like **Storage Blob Data Contributor** or **Storage Blob Data Owner**.

4️⃣ Connect ADLS Gen2 to Azure Services

✅ **Azure Data Factory (ADF)** – Use ADLS Gen2 as a source/destination for ETL pipelines.
✅ **Azure Databricks** – Read/write ADLS Gen2 data using Spark.
✅ **Power BI** – Connect for real-time data analysis.
✅ **Azure Synapse Analytics** – Perform advanced analytics on data lake files.

ADF



Data factory

jaindatafactory

Ingest
Copy data at scale once or on a schedule.

Orchestrate
Code-free data pipelines.

Transform data
Transform your data using data flows.

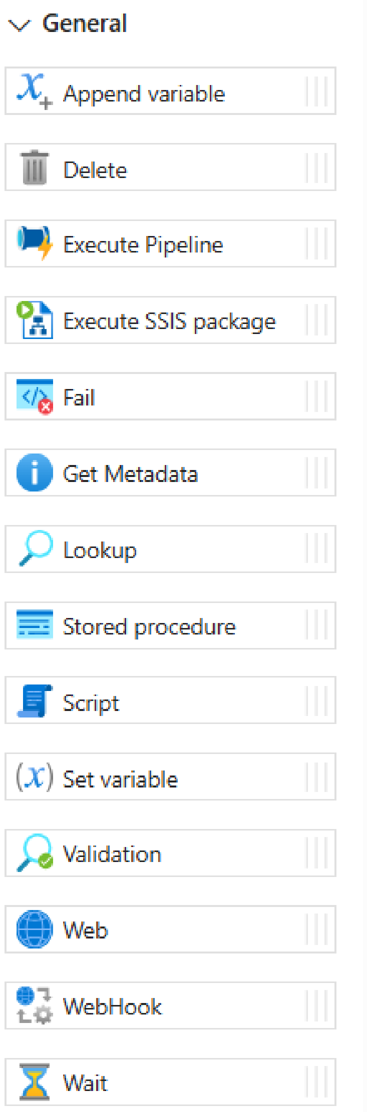
Configure SSIS
Manage & run your SSIS packages in the cloud.

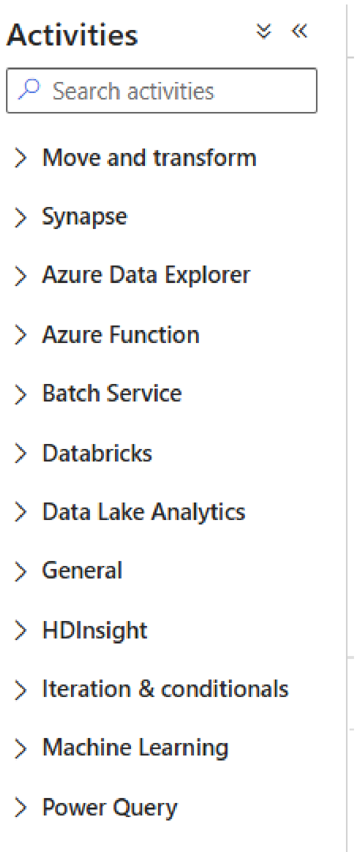
Recent resources

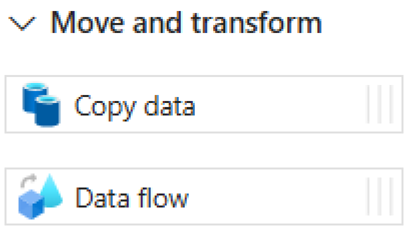
No items to show

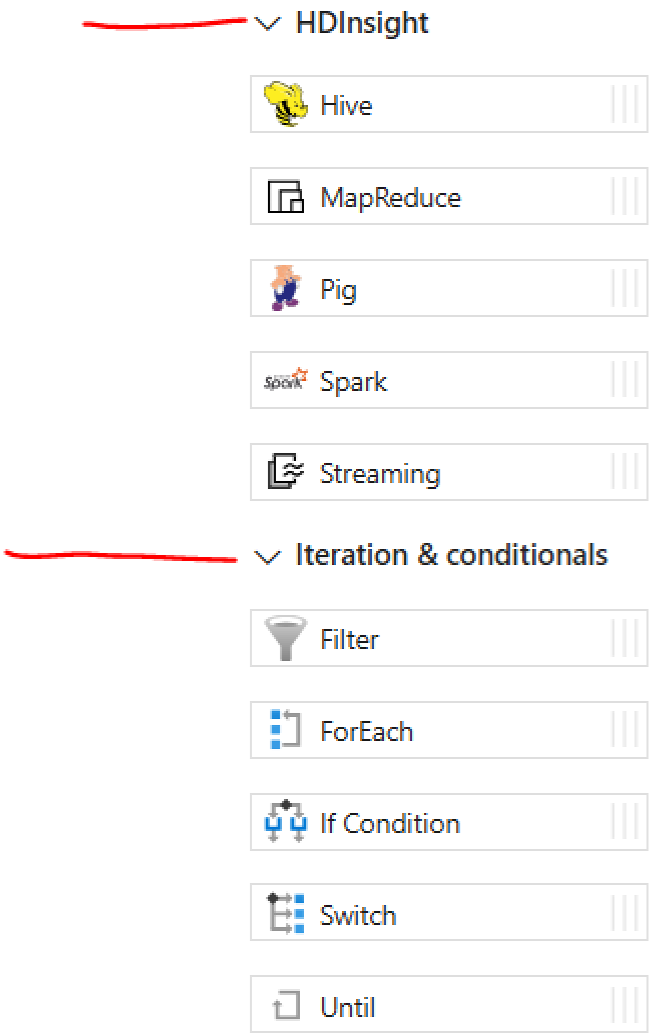
Your recently opened resources will show up here.

Discover more


General
Append variable
Delete
Execute Pipeline
Execute SSIS package
Fail
Get Metadata
Lookup
Stored procedure
Script
Set variable
Validation
Web
WebHook
Wait


Activities
Search activities
Move and transform
Synapse
Azure Data Explorer
Azure Function
Batch Service
Databricks
Data Lake Analytics
General
HDInsight
Iteration & conditionals
Machine Learning
Power Query


Move and transform
Copy data
Data flow



Azure Data Factory

Azure Data Factory is a cloud-based integration services that transforms and orchestrates data [sources to destination (cloud).

Components of ADF

1. Pipeline

2. Activities

3. Datasets

4. Dataflows

5. Linked Services

6. Integration Runtimes

Pipeline:

It is used to represent Logical Group of Activities that performs one unit of work. Azure Data Factory contains multiple pipelines

Activity:

It is used to represent single processing step in the pipeline

Datasets:

It is used represent data that is required for pipeline

Integration Runtimes:

Integration runtime is nothing but a compute structure used by Azure Data Factory to give integration capabilities across different network environments

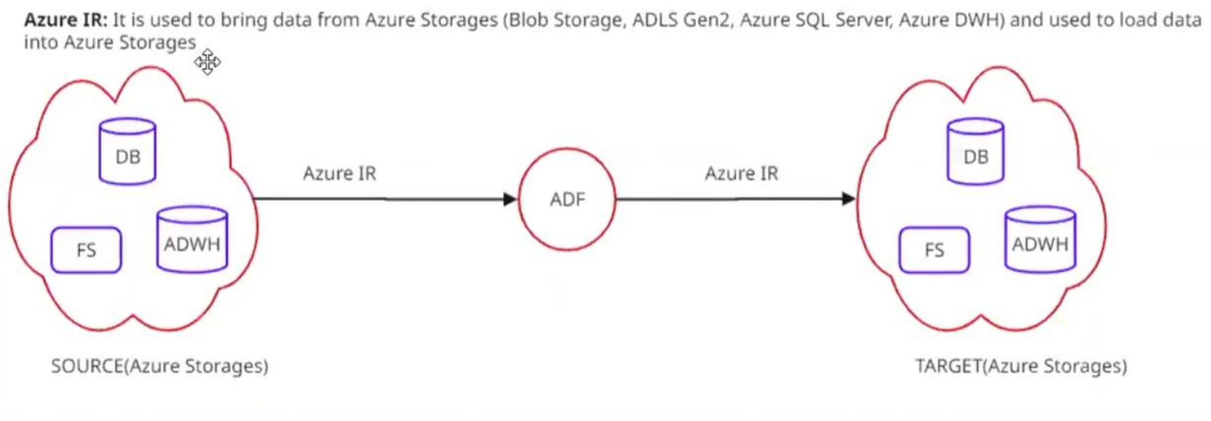
There are three types of Integration Runtimes

1. Azure IR from Azure to Azure

2. Self-Hosted IR -- Onpremise to Azure

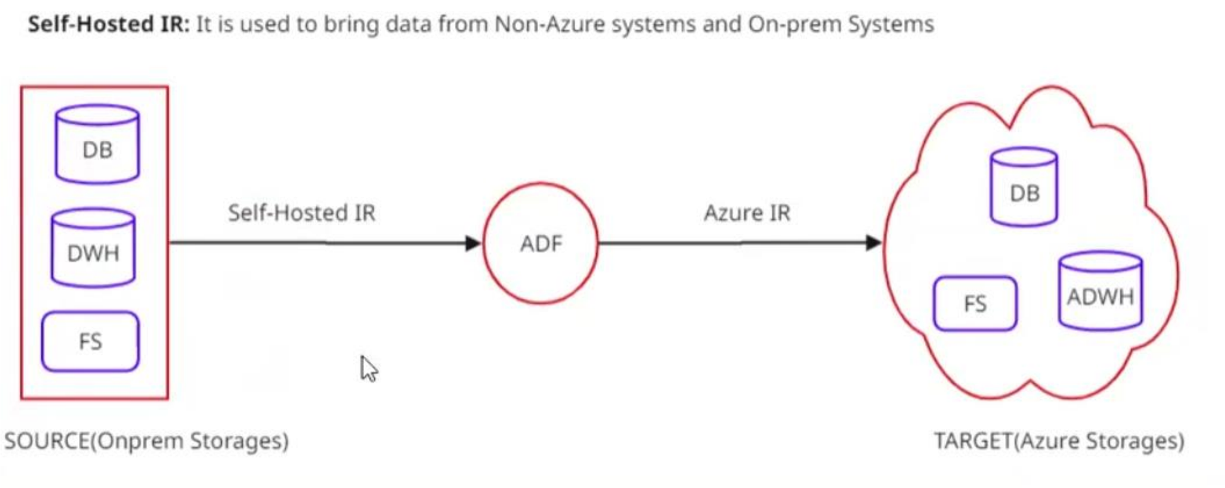
3. SSIS - IR --> To execute SSIS Packages from ADF

Azure IR from Azure to Azure



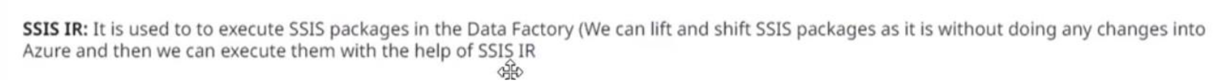
Azure IR: It is used to bring data from Azure Storages (Blob Storage, ADLS Gen2, Azure SQL Server, Azure DWH) and used to load data into Azure Storages

Self-Hosted IR -- Onpremise to Azure



Self-Hosted IR: It is used to bring data from Non-Azure systems and On-prem Systems

SSIS - IR --> To execute SSIS Packages from ADF



SSIS IR: It is used to to execute SSIS packages in the Data Factory (We can lift and shift SSIS packages as it is without doing any changes into Azure and then we can execute them with the help of SSIS IR

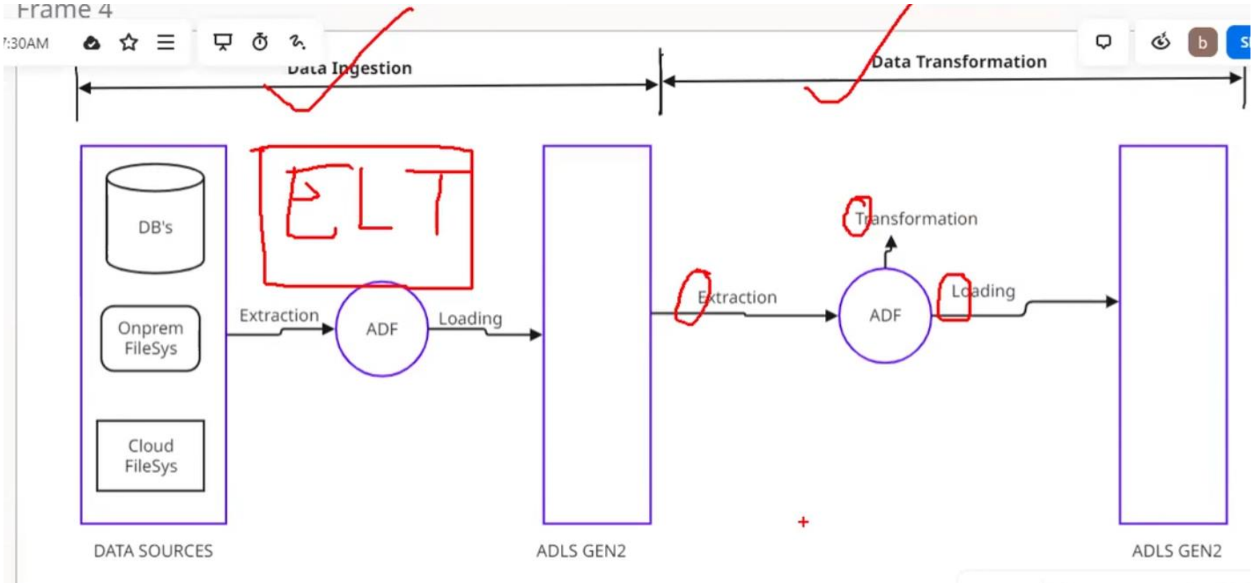
Linked Services:

It is used to have connection details of the Data source and target(sink) There are various connectors to read data and to load data.

Ex: Azure SQL Database

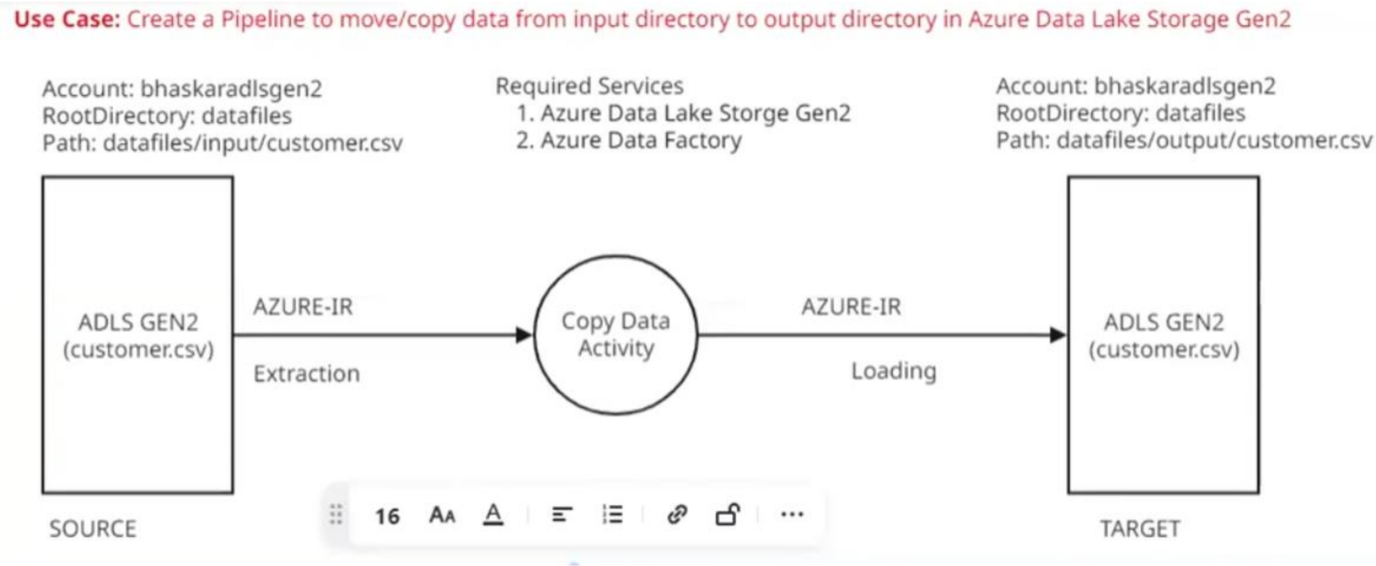
Azure Data Lake Storage Gen2

SQL Server Database



1. COPY

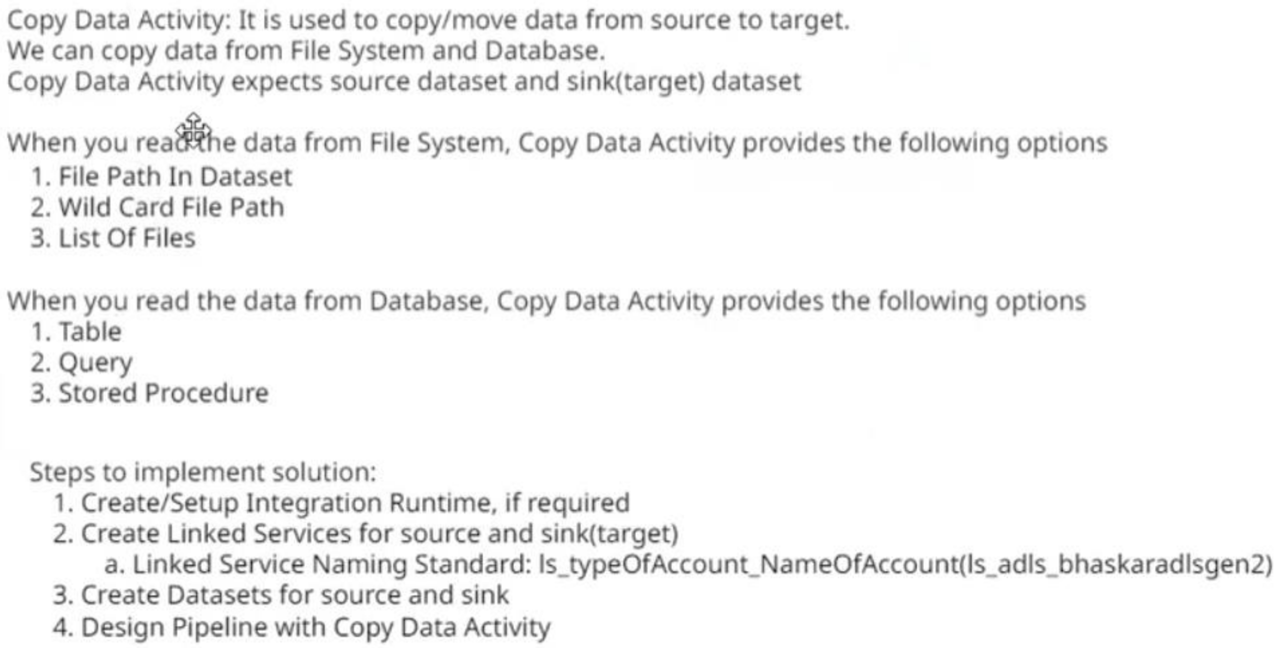
Copy data


Use Case: Create a Pipeline to move/copy data from input directory to output directory in Azure Data Lake Storage Gen2

Account: bhaskaradlsgen2
RootDirectory: datafiles
Path: datafiles/input/customer.csv

Required Services
1. Azure Data Lake Storge Gen2
2. Azure Data Factory

Account: bhaskaradlsgen2
RootDirectory: datafiles
Path: datafiles/output/customer.csv



Copy Data Activity: It is used to copy/move data from source to target.
We can copy data from File System and Database.
Copy Data Activity expects source dataset and sink(target) dataset

When you read the data from File System, Copy Data Activity provides the following options

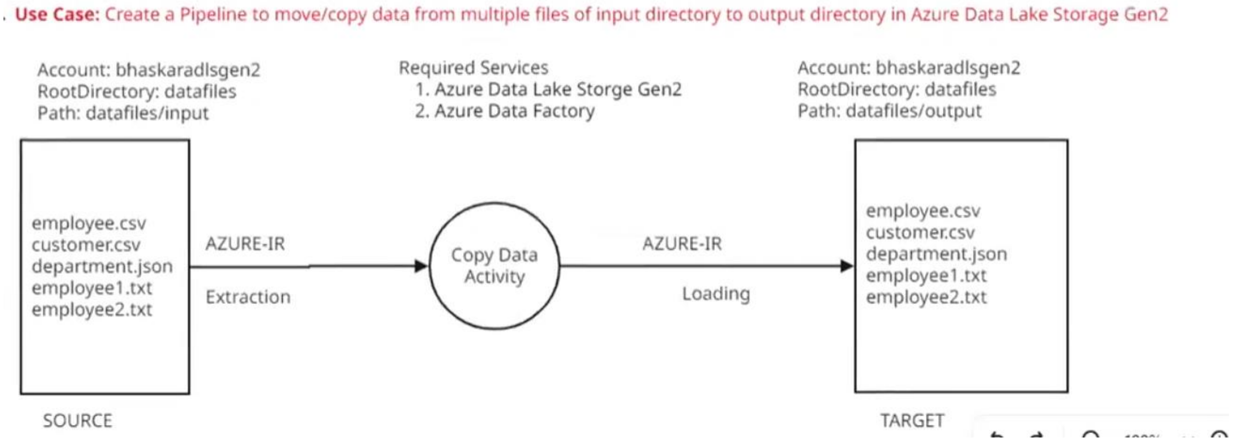
1. File Path In Dataset
2. Wild Card File Path
3. List Of Files

When you read the data from Database, Copy Data Activity provides the following options

1. Table
2. Query
3. Stored Procedure

Steps to implement solution:

1. Create/Setup Integration Runtime, if required
2. Create Linked Services for source and sink(target)
    a. Linked Service Naming Standard: ls_typeOfAccount_NameOfAccount(ls_adls_bhaskaradlsgen2)
3. Create Datasets for source and sink
4. Design Pipeline with Copy Data Activity

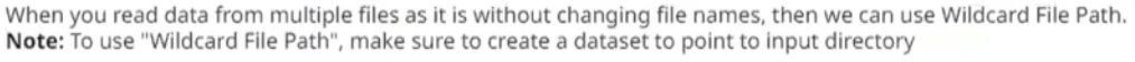


Use Case: Create a Pipeline to move/copy data from multiple files of input directory to output directory in Azure Data Lake Storage Gen2

Account: bhaskaradlsgen2
RootDirectory: datafiles
Path: datafiles/input

Required Services
1. Azure Data Lake Storge Gen2
2. Azure Data Factory

Account: bhaskaradlsgen2
RootDirectory: datafiles
Path: datafiles/output

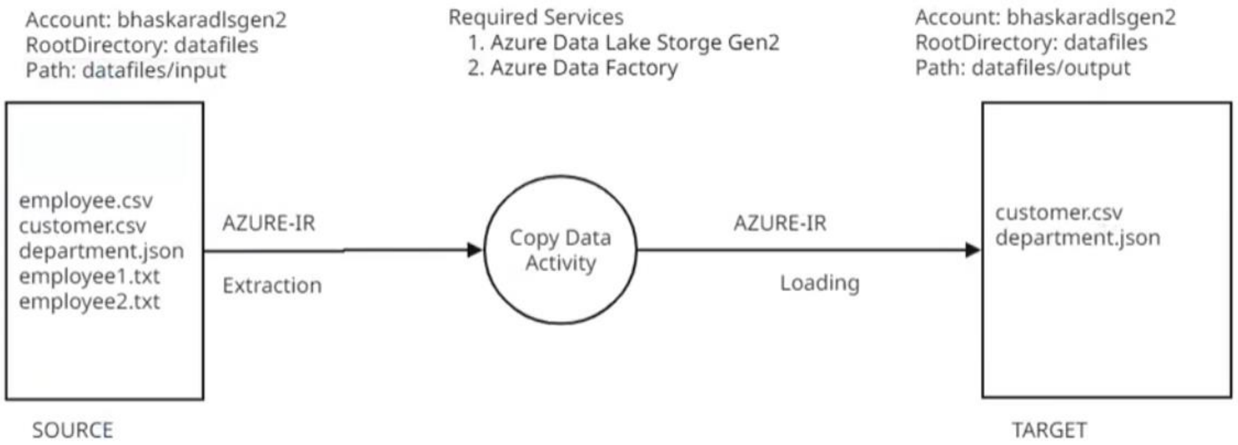


When you read data from multiple files as it is without changing file names, then we can use Wildcard File Path.
Note: To use "Wildcard File Path", make sure to create a dataset to point to input directory

Use Case: Create a Pipeline to move/copy data from input directory to output directory with specified list of files.

I have a folder which is having multiple files, I wanted to move specified files into target with single pipeline. How can you do it.

To provide the solution for above requirement, we have to specify list of files in a separate file and invoke this file in the Copy Data Activity



Account: bhaskaradlsgen2
RootDirectory: datafiles
Path: datafiles/input

Required Services
1. Azure Data Lake Storge Gen2
2. Azure Data Factory

Account: bhaskaradlsgen2
RootDirectory: datafiles
Path: datafiles/output

employee.csv
customer.csv
department.json
employee1.txt
employee2.txt

AZURE-IR
Extraction

Copy Data Activity

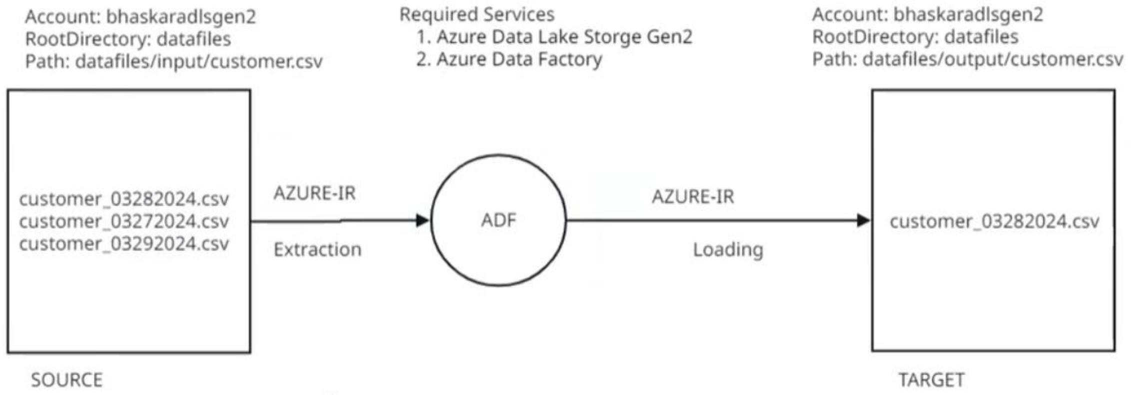
AZURE-IR
Loading

customer.csv
department.json

SOURCE

TARGET

Use Case: Create a pipeline to read data from data source if the file is current day file



Account: bhaskaradlsgen2
RootDirectory: datafiles
Path: datafiles/input/customer.csv

Required Services
1. Azure Data Lake Storge Gen2
2. Azure Data Factory

Account: bhaskaradlsgen2
RootDirectory: datafiles
Path: datafiles/output/customer.csv

customer_03282024.csv
customer_03272024.csv
customer_03292024.csv

AZURE-IR
Extraction

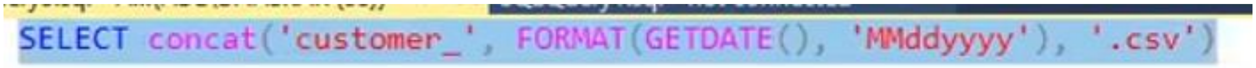
ADF

AZURE-IR
Loading

customer_03282024.csv

SOURCE

TARGET



```
SELECT concat('customer_', FORMAT(GETDATE(), 'MMddyyyy'), '.csv')
```

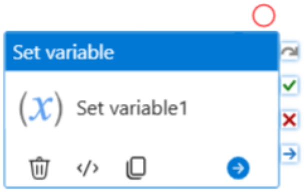
To provide solution for above use case,

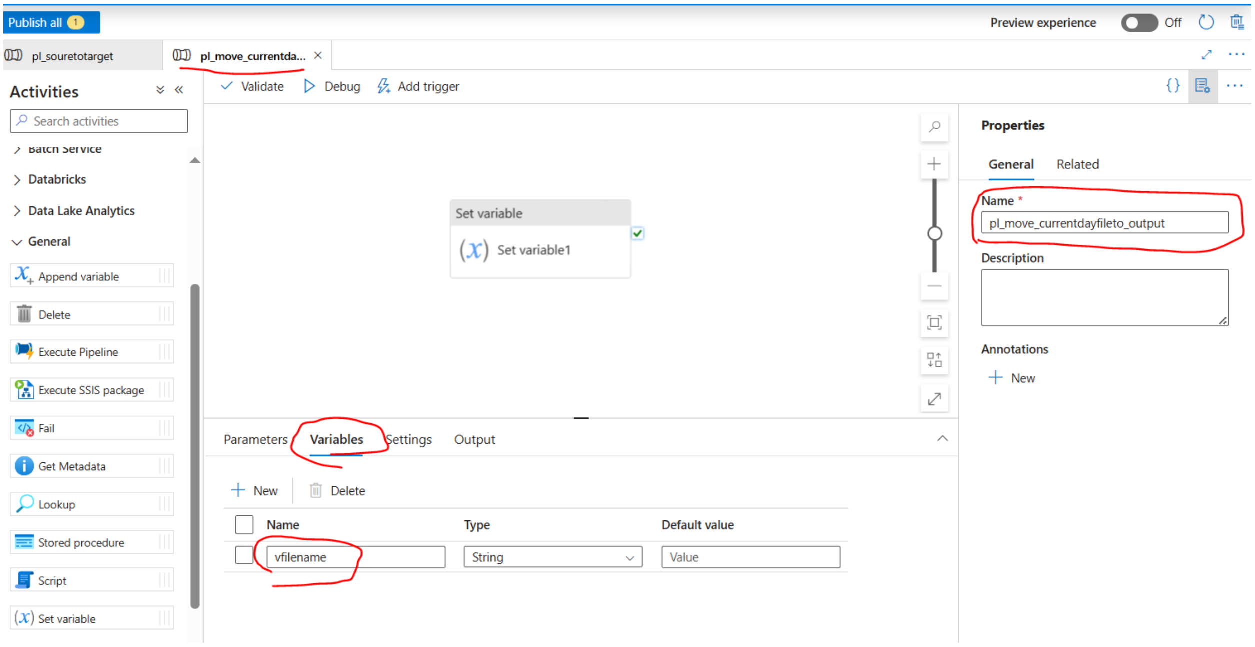
we need to create variable and make sure to get value dynamically by writing expression.

To set the value to the variable, we need to "Set Variable" Activity.

To write the expression, we can use "Add Dynamic Content" with the following items

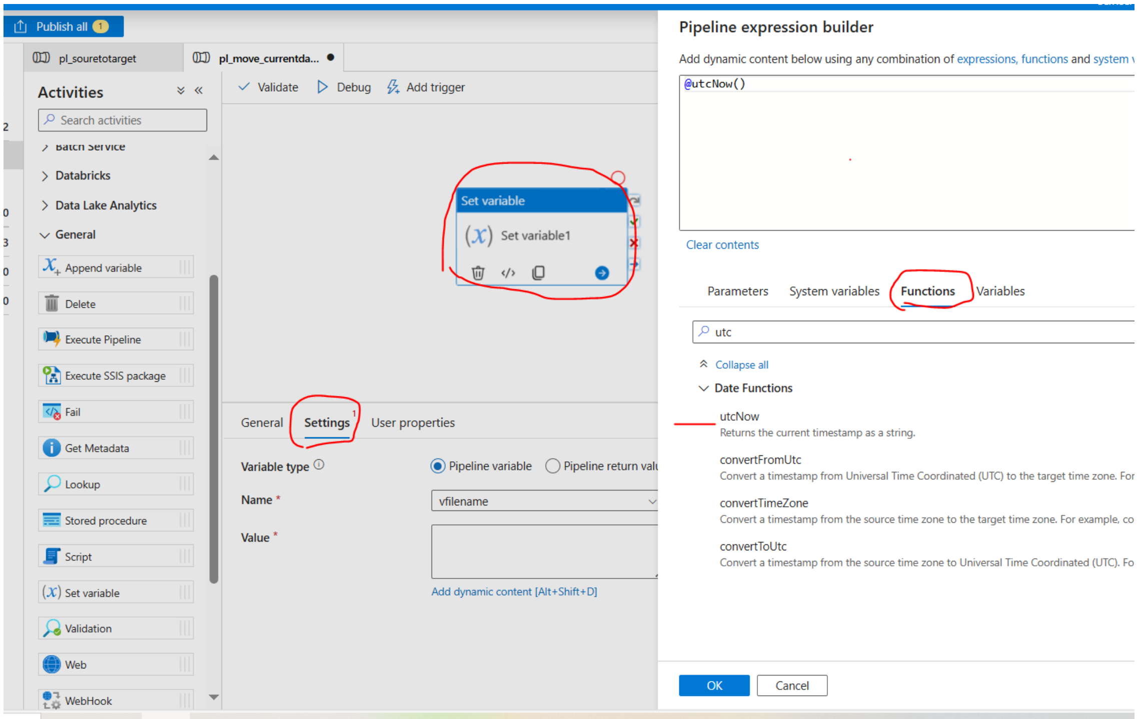
1. Activity Outputs
2. System Variables
3. Variables
4. Functions
5. Parameters





Name * pl_move_currentdayfileto_output

□	Name	Type	Default value
□	vfilename	String	Value



Variable type ● Pipeline variable ○ Pipeline return valu

Name * vfilename

Value *

Pipeline expression builder

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

```
@utcNow()
```

utc

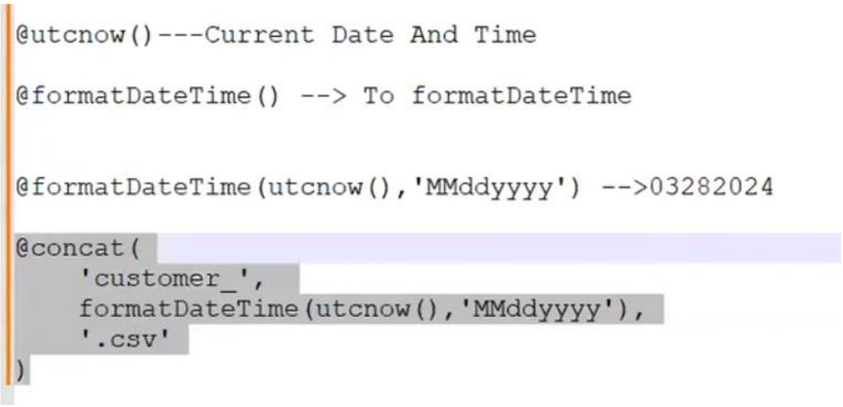
Date Functions

utcNow
Returns the current timestamp as a string.

convertFromUtc
Convert a timestamp from Universal Time Coordinated (UTC) to the target time zone. For

convertTimeZone
Convert a timestamp from the source time zone to the target time zone. For example, co

convertToUtc
Convert a timestamp from the source time zone to Universal Time Coordinated (UTC). Fo



```
@utcnow()---Current Date And Time

@formatDateTime() --> To formatDateTime


@formatDateTime(utcnow(),'MMddyyyy') -->03282024

@concat(
    'customer_',
    formatDateTime(utcnow(),'MMddyyyy'),
    '.csv'
)
```

@utcnow()-----------Current Date And Time

@formatDateTime() --------------> To formatDateTime

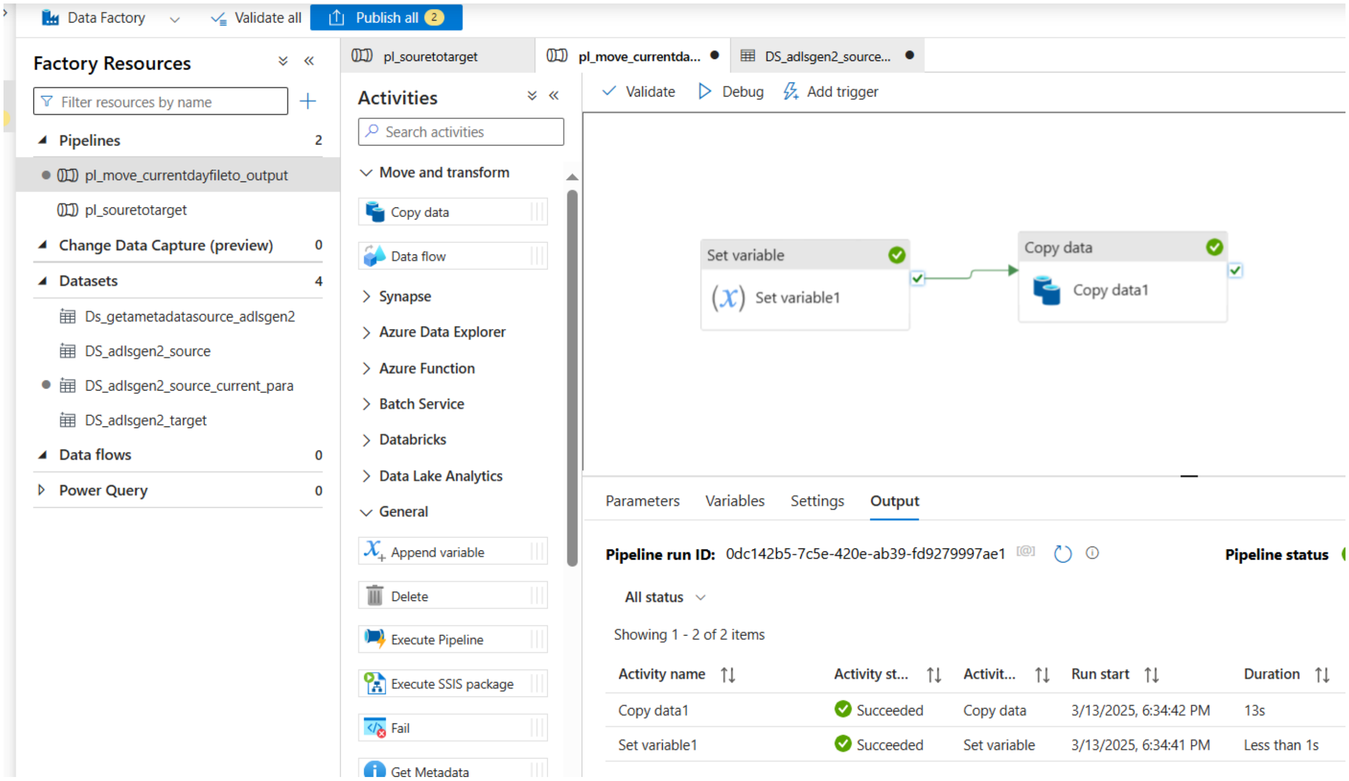
@formatDateTime (utcnow(), 'MMddyyyy') ------->03282024

@concat(

'customer_',

formatDateTime (utcnow (), 'MMddyyyy'),

'.csv'



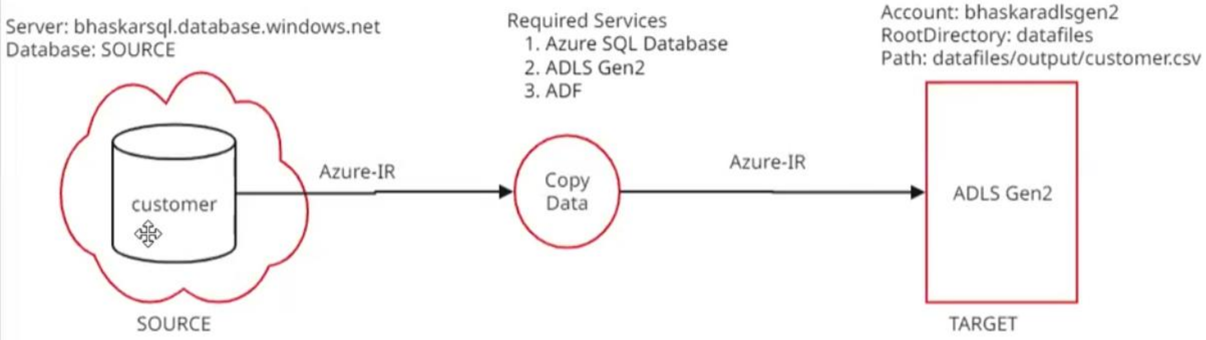
Pipeline run ID: 0dc142b5-7c5e-420e-ab39-fd9279997ae1 **Pipeline status**

All status

Showing 1 - 2 of 2 items

Activity name	Activity st...	Activit...	Run start	Duration
Copy data1	✓ Succeeded	Copy data	3/13/2025, 6:34:42 PM	13s
Set variable1	✓ Succeeded	Set variable	3/13/2025, 6:34:41 PM	Less than 1s

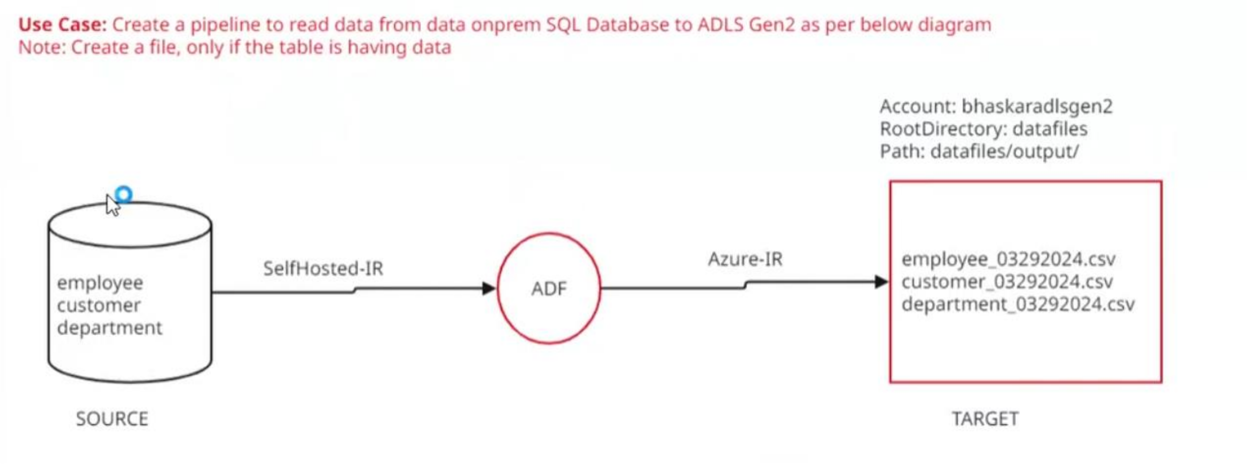
Use Case: Create a pipeline to read data from data azure SQL Database to ADLS Gen2 as per below diagram



Server: bhaskarsql.database.windows.net
Database: SOURCE

Required Services
1. Azure SQL Database
2. ADLS Gen2
3. ADF

Account: bhaskaradlsgen2
RootDirectory: datafiles
Path: datafiles/output/customer.csv



Use Case: Create a pipeline to read data from data onprem SQL Database to ADLS Gen2 as per below diagram
Note: Create a file, only if the table is having data