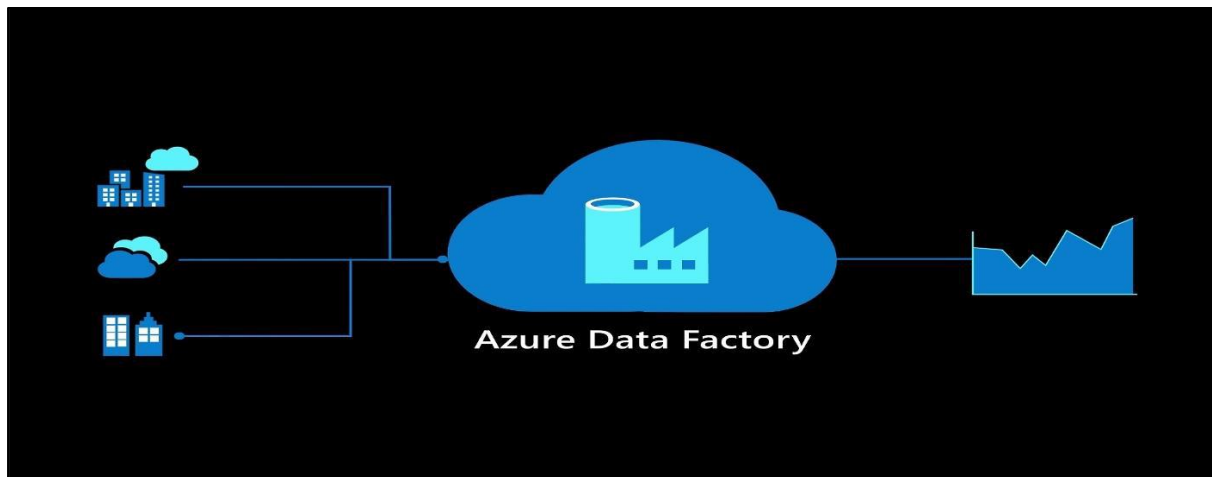


## What is Azure Data Factory?

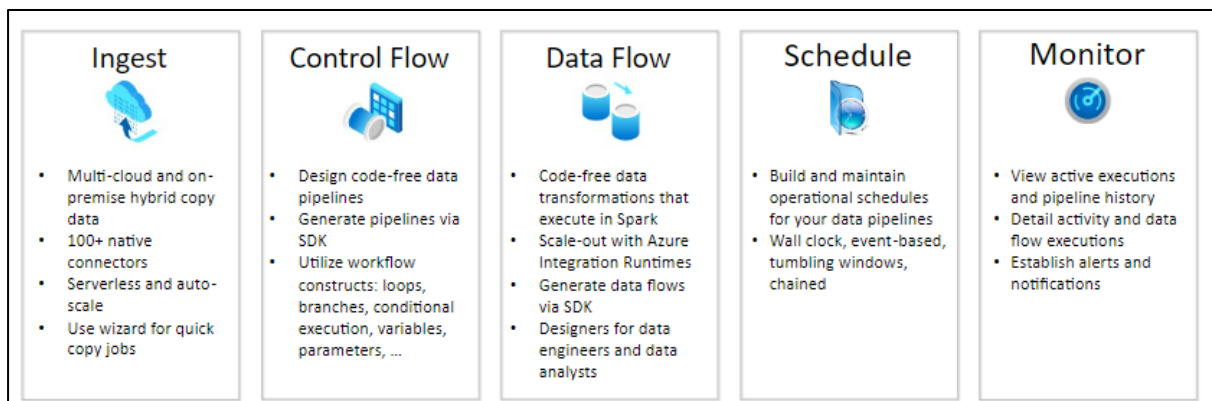


Azure Data Factory is Azure cloud ETL service for scale-out serverless data integration and data transformation. You can also lift and shift existing SSIS packages to azure and run them with full compatibility in ADF.

It is the cloud based ETL and data integration service that allows you to create data driven workflows for orchestrating data movement and transformation data at scale.

Azure Data Factory is a managed cloud service that's built for complex hybrid Extract-Transform-Load (ETL), Extract-Load-Transform (ELT) and data integration projects.

## How does it work?



Azure Data Factory contains a series of interconnected systems that provide a complete end-to-end platform for data engineers.

## Ingest

Ingestion refers to the process of collecting and importing data from various sources into Azure for storage, processing, and analysis.

Azure data Factory provides several built-in connectors for ingesting data from different sources such as Azure Blob Storage, Azure data Lake, Azure SQL Databases and many more.

## Control Flow

Control Flow is a collection of activities that helps you build, schedule, and orchestrate your data integration workflows.

It allows you to manage the execution order of the activities, set conditions for the workflow and define the workflow's dependencies.

## Data Flow

Data Flow is a cloud-based data transformation service that allows you to build data transformation workflows using a graphical interface.

Data Flow enables you to perform various data integration tasks such as data cleansing, data enrichment, data aggregation, data transformation, and data validation.

## Schedule

We can schedule the execution of your pipelines using the built-in scheduling feature. This allows you to run your data integration workflows automatically at specific intervals or times.

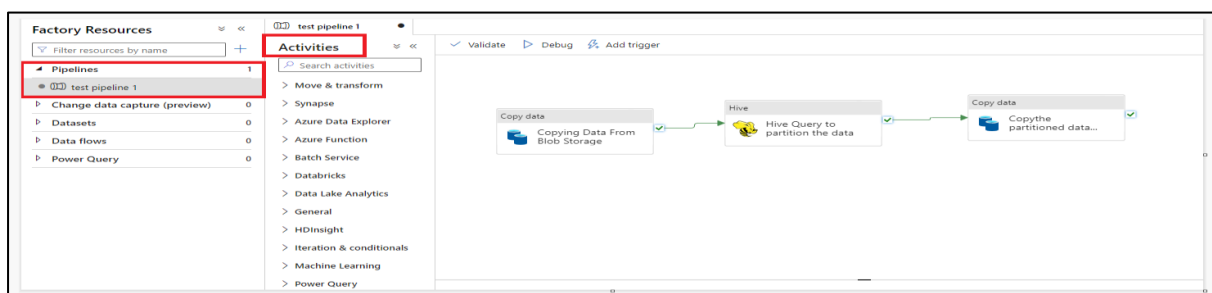
## Monitor

Monitoring Allows you to track the status of your data integration workflows and troubleshooting issues that may occur during pipeline execution.

It provides real-time visibility into the health and performance of your data factory, enabling you to identify and resolve quickly.

## Concepts in Azure Data Factory

### Pipeline



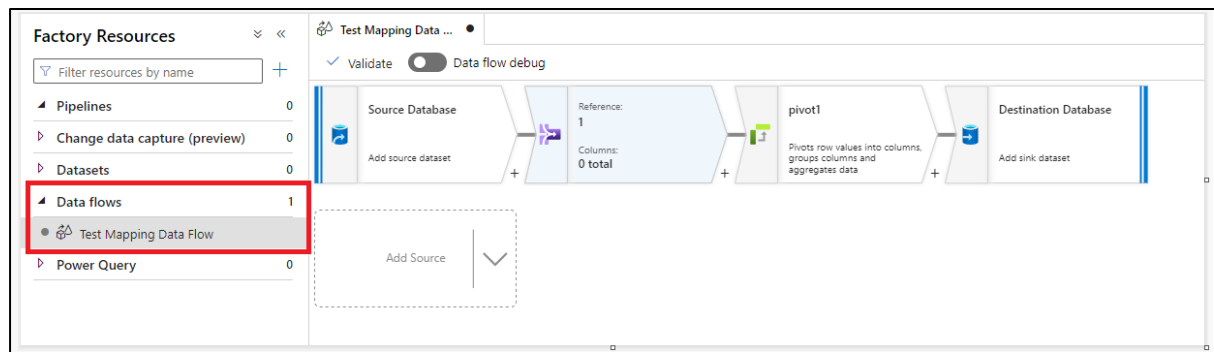
A pipeline is a logical grouping of activities that performs a unit of work. Together, the activities in a pipeline perform a task.

For achieving any task in Azure Data Factory, we create a pipeline which contains the various types of activities as required for full filling the business purpose.

For example, A pipeline can contain a group of activities that ingests data from an Azure blob and then runs a Hive query on an HDInsight cluster to partition the data.

The Activities in a pipeline can be chained together to operate sequentially, or they can operate independently in parallel.

## Mapping Data Flows



Mapping Data Flows is a drag-and-drop interface that allows you to define data transformations using wide range of data sources, transformations, and destinations.

You can create complex data transformation logic using a visual interface, without the need to write code.

Mapping Data Flows offers a brand range of transformations including aggregations, joins, pivots, filters and many more.

You can build-up a reusable library of data transformation routines and execute those processes in a scaled-out manner from your ADF pipelines.

Data factory will execute your logic on a Spark cluster that spins-up and spins-down when you need it. You wont ever have to manage or maintain clusters.

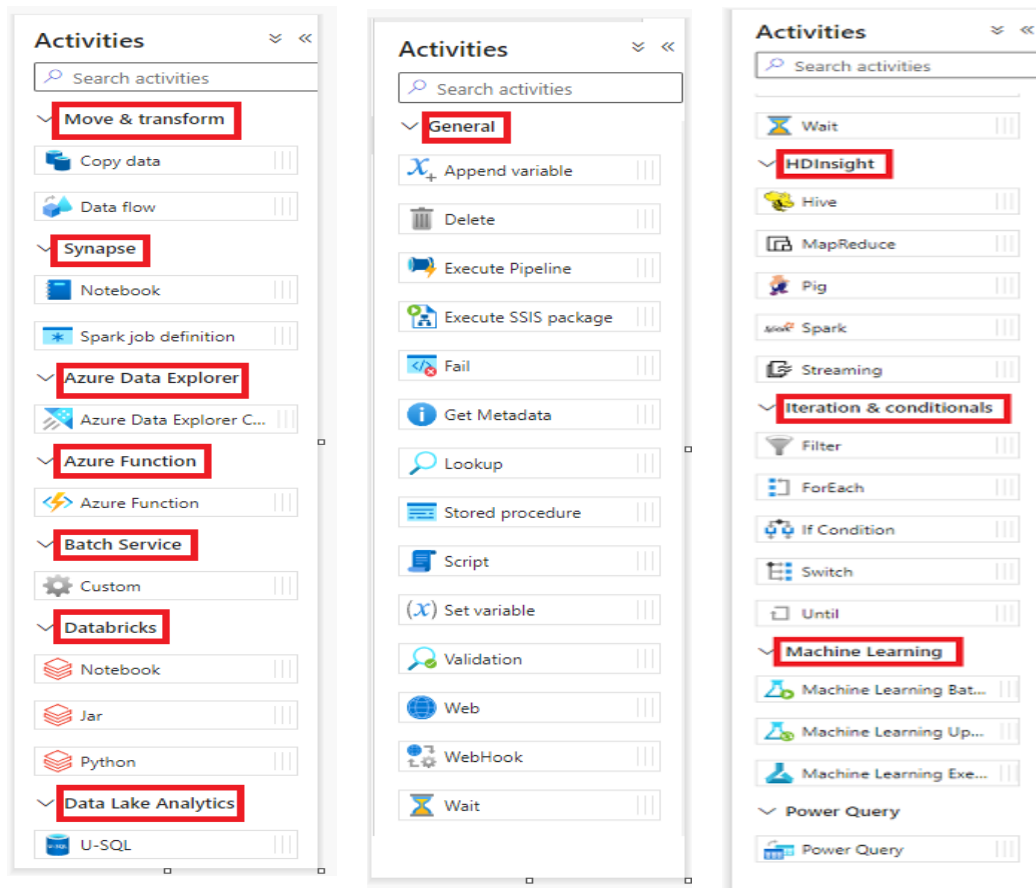
### Activities

Activities represent a processing step in a pipeline. For ex: you might use a copy activity to copy data from one data store to another data store.

Data factory supports three types of activities:

1. Data movement activities
2. Data transformation activities

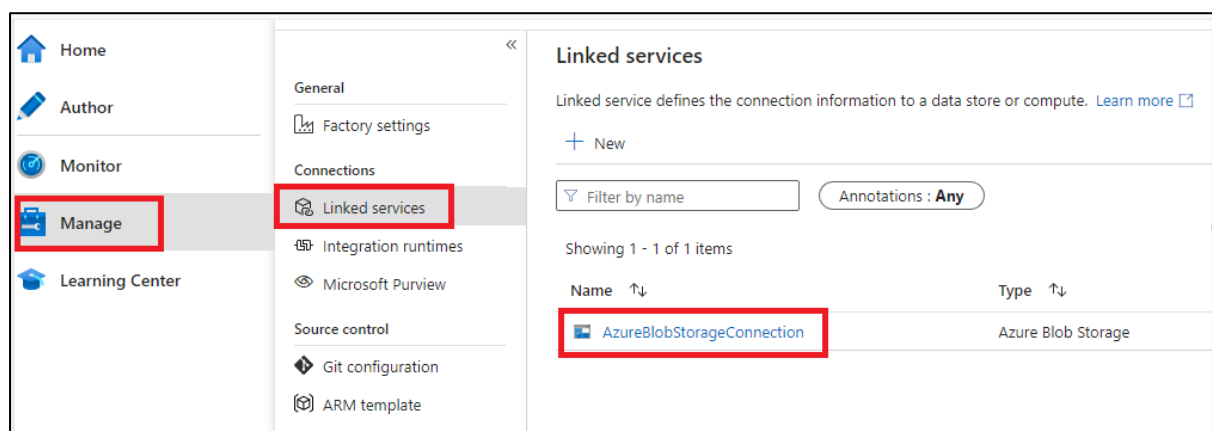
### 3. Control Activities



### Linked Services

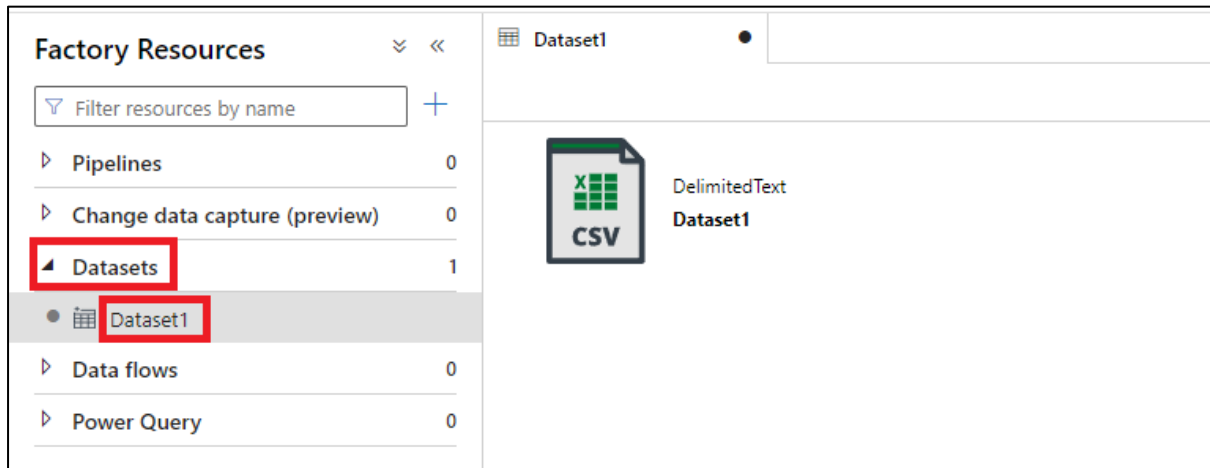
Linked services are much like connection strings, which define the connection information that's needed for Data factory to connect to external resources.

A linked service defines the connection to the data source, and a dataset represents the structure of the data. For example, An Azure storage-linked service specifies a connection string to connect to the Azure Storage account. Additionally, an Azure blob dataset specifies the blob container and the folder that contains the data.

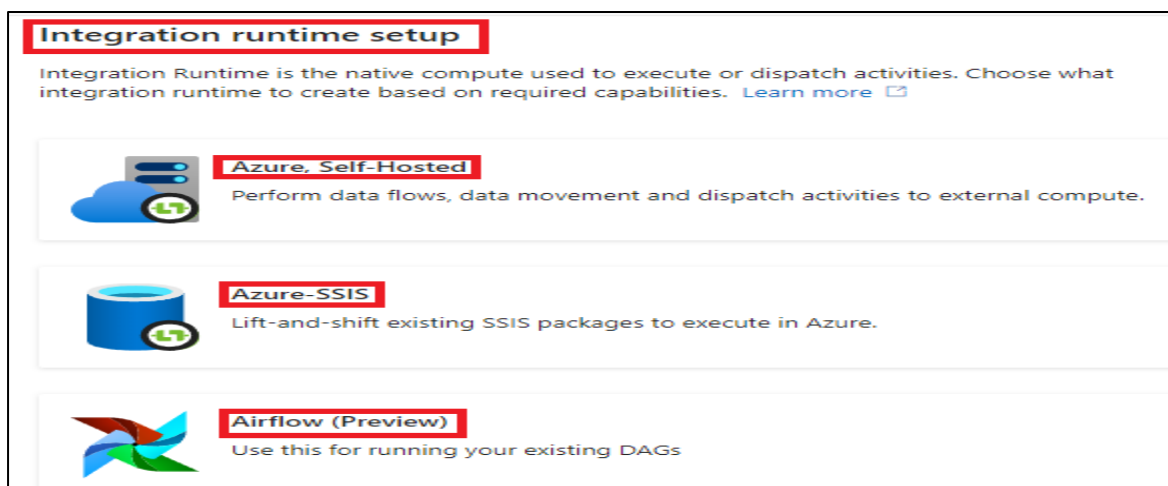


## Datasets

Datasets represent data structures within the data stores, which simply point to or reference the data you want to use in your activities.



## Integration Runtime



Integration Runtime (IR) is a data integration component in Azure Data Factory that provides a secure and scalable way to move and transform data from various sources to different destinations.

Integration runtime provides the bridge between the activity and linked services.

Integration has three deployment models:

### Self-hosted IR

The self-hosted IR is installed on your own infrastructure, such as an on-premises server or a virtual machine in a public cloud and provides a secure way to move data between on-premises and cloud data stores.

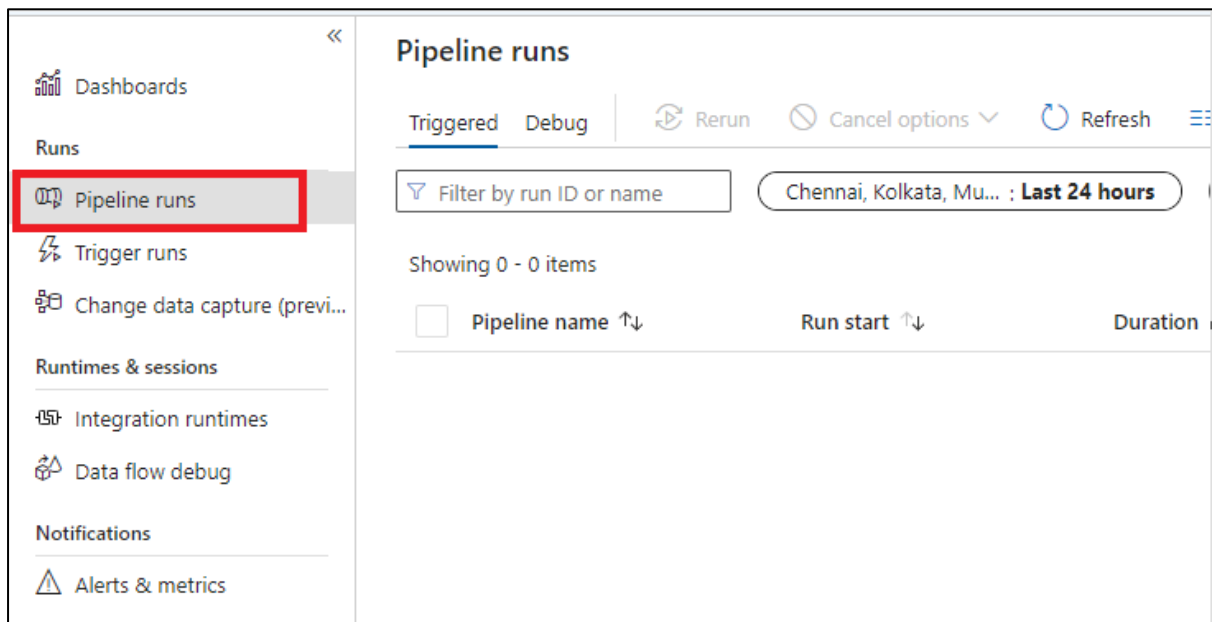
### Azure-hosted IR

It is a fully managed service in Azure that provides a secure and scalable way to move between different cloud data stores.

### Airflow (Preview)

Airflow is now available as a preview feature in Azure Data Factory. Airflow provides platform for creating, scheduling, and monitoring data pipelines, and its integration with Azure Data Factory allows you to create more complex and sophisticated data integration workflows.

## Pipeline Runs



Pipeline Run is an instance of the pipeline execution.

Pipeline runs can be triggered manually or automatically based on a defined schedule or trigger event.

## Triggers

Triggers determines when a pipeline execution needs to be kicked off. There are different types of triggers.

### Schedule Trigger

A trigger that invokes a pipeline on a wall-clock schedule.

Check below link for more details on Schedule triggers.

Create schedule triggers - Azure Data Factory & Azure Synapse | Microsoft Learn

### Tumbling window trigger

Tumbling window triggers are type of trigger that fires at a periodic time interval from a specified start time, while retaining state.

Tumbling windows are a series of fixed-sized, non-overlapping, and contiguous time intervals.

Tumbling window trigger has a one-to-one relationship with a pipeline and can only reference a singular pipeline.

Check below link for more details on Tumbling window triggers.

Create tumbling window triggers - Azure Data Factory & Azure Synapse | Microsoft Learn

## Event-based trigger

An event-based trigger runs pipelines in response to an event. There are two flavours of event-based triggers.

## Storage Event Trigger

Storage event trigger runs a pipeline against events happening in a storage account, such as the arrival of a file, or the deletion of a file in Azure Blob Storage account.

Check below link for more details on Storage Event Triggers

Create event-based triggers - Azure Data Factory & Azure Synapse | Microsoft Learn

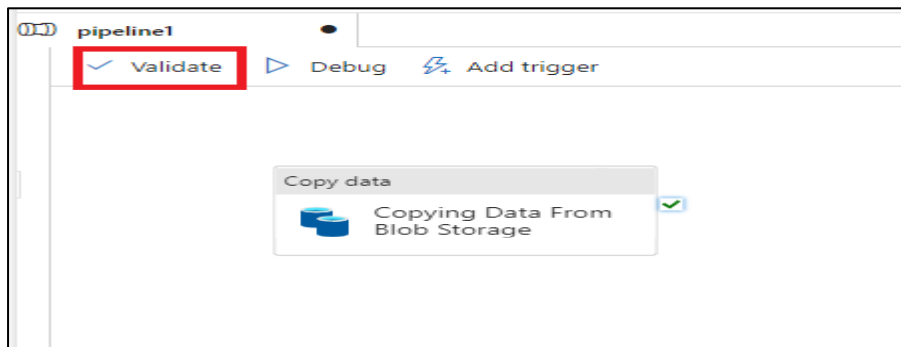
## Custom Event trigger

It processes and handles custom articles in Event Grid.

Check below link for more details on custom event triggers.

Create custom event triggers in Azure Data Factory - Azure Data Factory | Microsoft Learn

## Validate



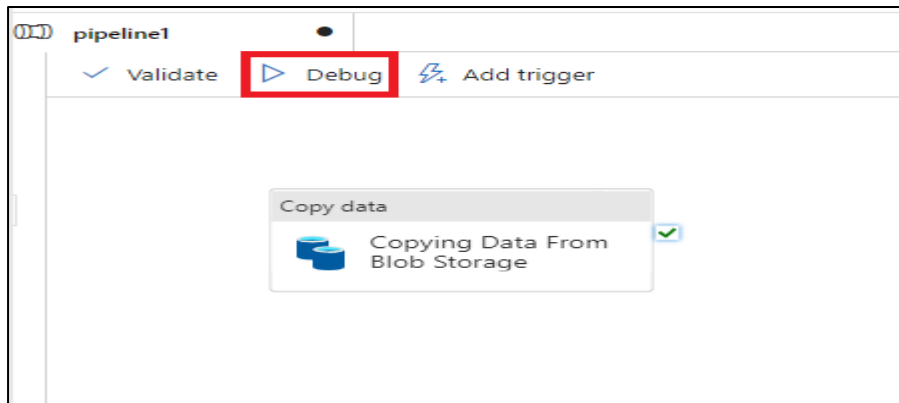
Azure Data Factory will perform a validation check on the pipeline to ensure that all of the inputs, outputs and activities are correct and that the pipeline can be executed successfully.

Any errors/warning found during the validation process, will be notified and we can take action to correct them.

Once the Validation process is completed, we can deploy the pipeline.

It's always a good practice to validate the pipeline before deploying them to production.

## Debug



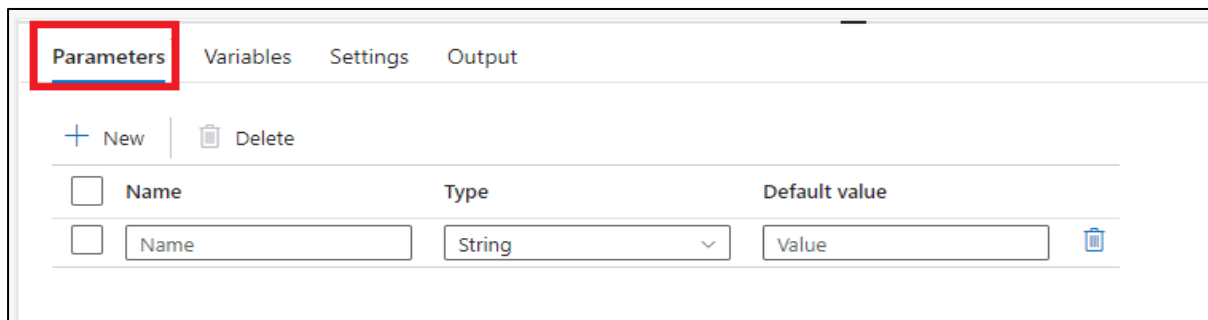
Debug allows you to test and troubleshoot your data integration pipelines before they are deployed to production.

With Debug, you can run your pipeline or a specific activity within the pipeline in a test environment to verify that it works as expected.

During debug session, you can monitor the progress of each activity and view detailed logs to help identify any errors.

We can also modify the inputs and parameters of your pipeline or activity to test different scenarios and ensure that your pipeline is robust and reliable.

## Parameters



Parameters are key-value pairs of read-only configuration, its defined at the pipeline level, and cannot be modified during a pipeline run.

Parameters are defined in the pipeline and passed as input values to activities within the pipeline.

When defining a parameter, we provide the name, datatype, and default value. We can then reference the parameter in the activities within the pipeline by using its name and passing it as an input value.

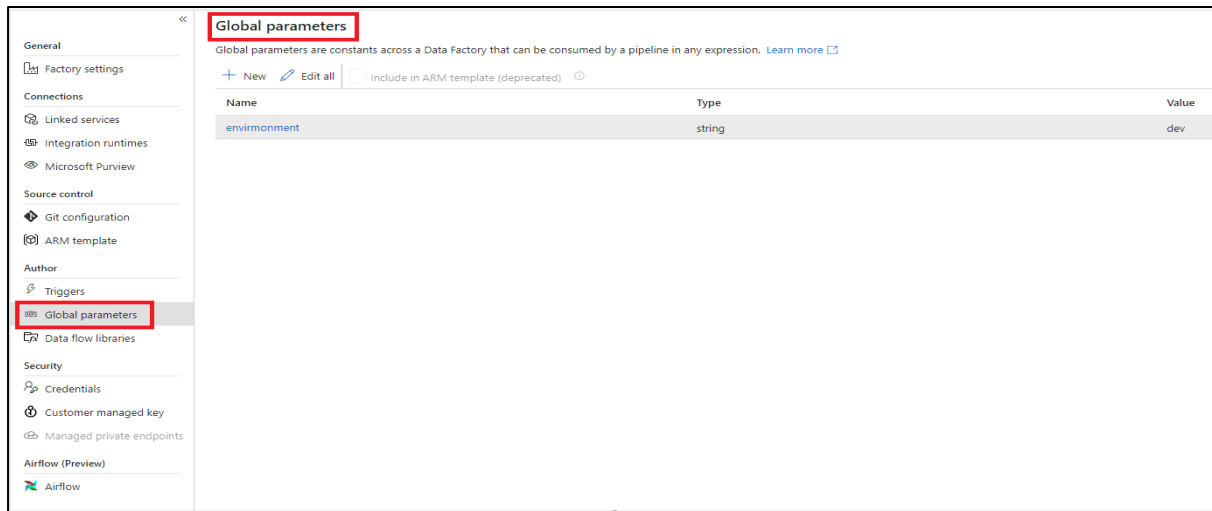
Dataset is a strongly typed parameter and a reusable or referenceable entity. An activity can reference datasets and can consume the properties that are defines in the dataset definition.

Linked service is also a strongly typed parameter and a reusable or referenceable entity, that contains the connection information to either a data store or a compute environment.

You can use parameters to enable dynamic behaviour within pipelines, such as conditionally executing activities based on input values or applying transformations to input data based on the value of a parameter.



## Global Parameters



The screenshot shows the 'Global parameters' configuration page. On the left is a navigation menu with categories: General, Connections, Source control, Author, Security, and Airflow (Preview). The 'Global parameters' option under the 'Author' category is highlighted with a red box. The main content area has a title 'Global parameters' with a red box around it, followed by a description: 'Global parameters are constants across a Data Factory that can be consumed by a pipeline in any expression. [Learn more](#)'. Below this are '+ New' and 'Edit all' buttons, and a checkbox 'Include in ARM template (deprecated)'. A table lists the parameters:

Name	Type	Value
environment	string	dev

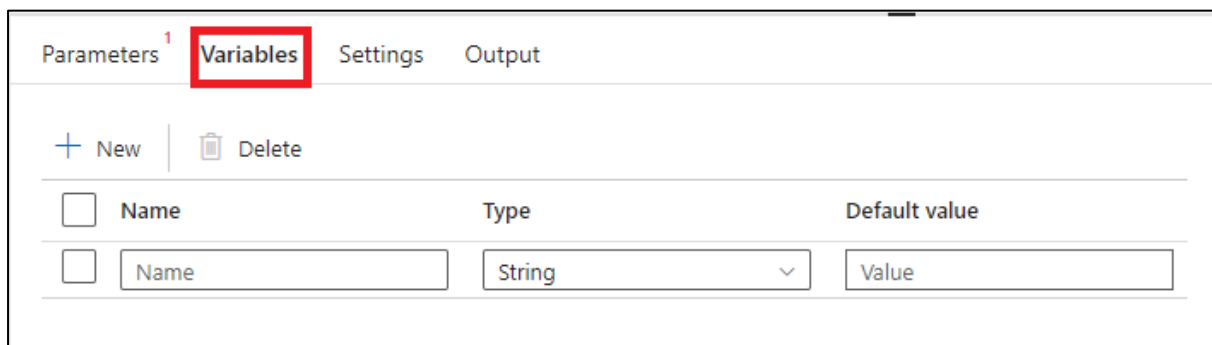
Global parameters are constants across a data factory that can be consumed by a pipeline in any expression. They are useful when you have multiple pipelines with identical parameters names and values.

When promoting a data factory using the continuous integration and deployment process (CI/CD), you can override these parameters in each environment.

Global parameters can be used in any pipeline expression. If a pipeline is referencing another resource such as a dataset or data flow, you can pass down the global parameters value via those resource's parameters.

Global parameters are referenced as `pipeline().globalParameters.<parameterName>`.

## Variables



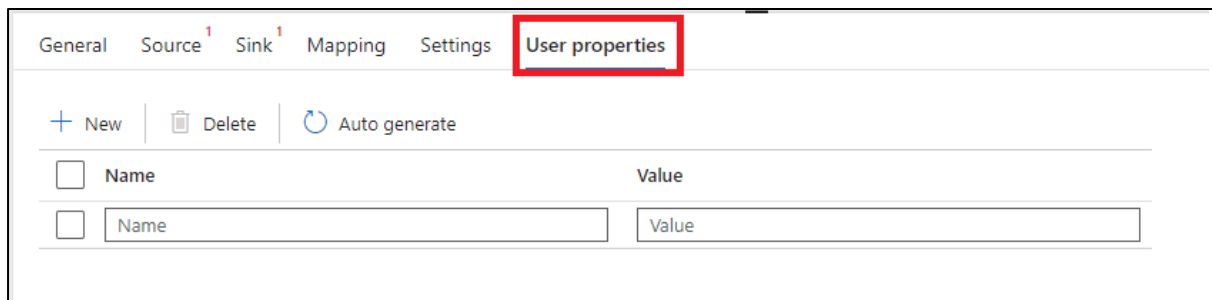
The screenshot shows the 'Variables' configuration page in a pipeline editor. The 'Variables' tab is highlighted with a red box. The page has tabs for 'Parameters', 'Variables', 'Settings', and 'Output'. Below the tabs are '+ New' and 'Delete' buttons. A table lists the variables:

Name	Type	Default value
<input type="text" value="Name"/>	<input type="text" value="String"/>	<input type="text" value="Value"/>

Variables can be used inside of pipelines to store temporary values and can also be used in conjunction with parameters to enable passing values between pipelines, data flows, and other activities.

Pipeline Variables are values that can be set and modified during a pipeline run. Unlike pipeline parameters, which are defined at the pipeline level and cannot be changed during a pipeline run, pipeline variables can be set and modified within a pipeline using Set Variable activity or set it using dynamic expression in other activities.

## User Properties



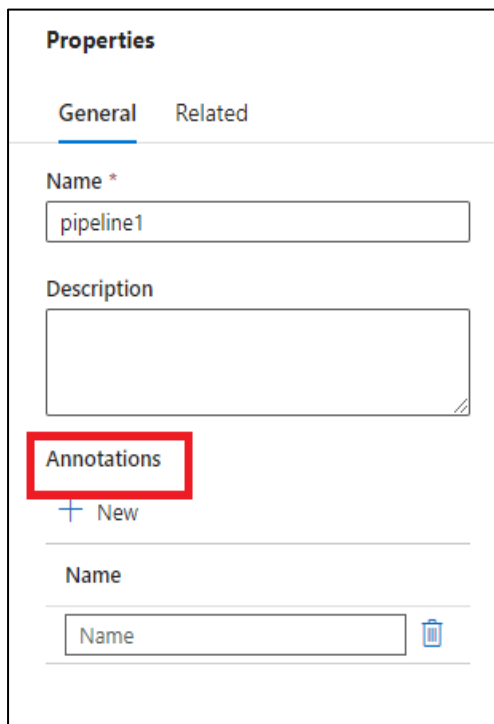
The screenshot shows the 'User properties' tab selected in the top navigation bar. Below the tabs, there are three buttons: '+ New', 'Delete', and 'Auto generate'. A table with two columns, 'Name' and 'Value', is visible. The 'Name' column has a checkbox and a text input field with the placeholder 'Name'. The 'Value' column has a text input field with the placeholder 'Value'.

User Properties are key-value pairs defined at the activity level.

By adding user properties, you can view additional information about activities under activity runs window that may help you to monitor your activity executions.

If you want to monitor for dynamic values at the activity level, you can do so by leveraging the user properties.

## Annotations



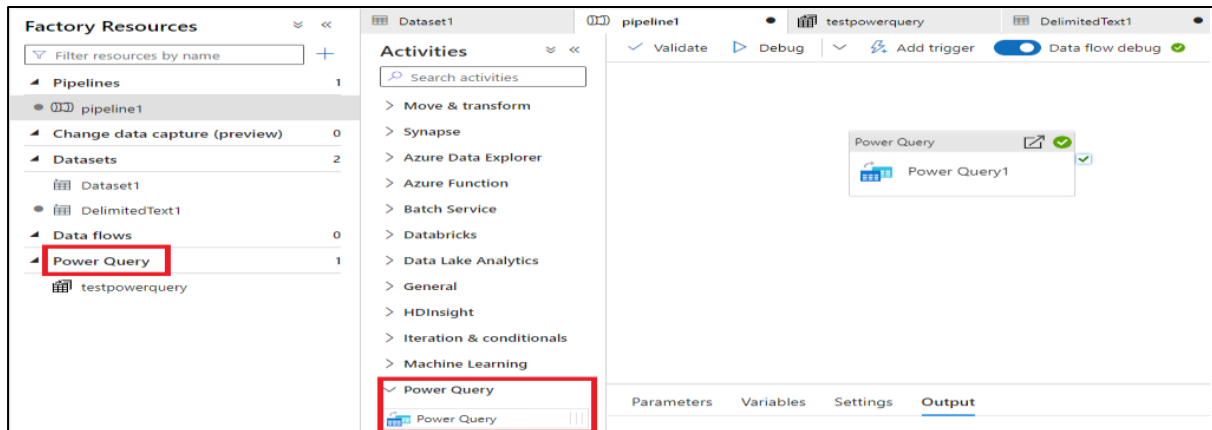
The screenshot shows the 'Annotations' section in the 'Properties' panel. The 'General' tab is selected. There is a 'Name \*' field with the value 'pipeline1' and a 'Description' text area. Below these, the 'Annotations' section is highlighted with a red box. It contains a '+ New' button and a table with one column, 'Name', which has a text input field with the placeholder 'Name' and a delete icon.

Annotations are tags that you can add to your Azure Data Factory entities to easily identify them.

An Annotation allows you to classify or group different entities in order to easily monitor or filter them after an execution.

Annotations only allow you to define static values and can be added to pipelines, datasets, linked services and triggers.

## Power Query

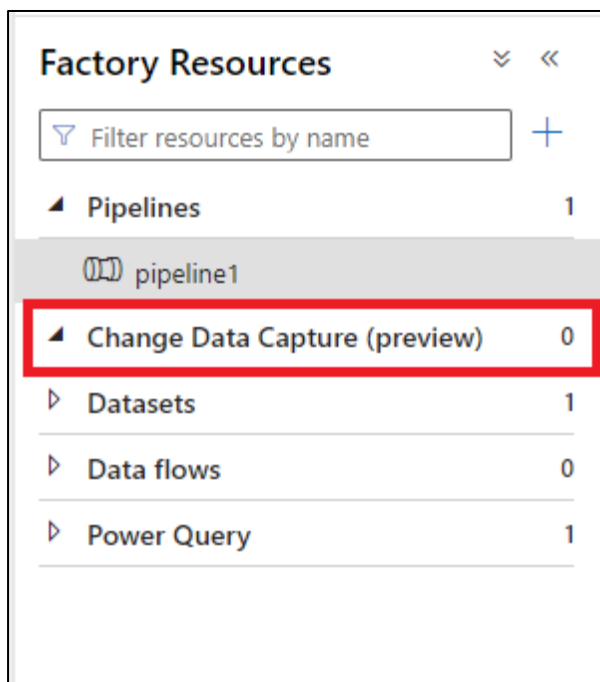


In Azure Data Factory power query is available as a data transformation activity that can perform complex data transformations on your data.

Data wrangling in data factory allows you to build interactive power query mashups natively in ADF and then execute those at scale inside of an ADF pipeline.

Power query activity allows users to create data transformation pipelines using power query expressions. These pipelines can be used to extract data from various sources and transform it using power query expressions and load it into a destination store.

## Change Data Capture



Change resource in ADF allows for full fidelity change data capture that continuously runs in near real-time through a guided configuration experience.

Change Data Change will now exist as a new native top-level resource in the Azure Data Factory studio where we can quickly configure continuous running jobs to process big data at scale with extreme efficiency.

Change Data Capture (CDC) is a technique used to track and capture changes made to data in a data sources.