1, Basics of Azure ADF → Explanation

What is Azure Data Factory

- Azure Data Factory is Azure's 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 transforming data at scale.

Why Azure Data Factory

- Big data requires service that can orchestrate and operationalize processes to refine these enormous stores of raw data into actionable business insights. 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.

Top level Concepts in Azure Data Factory

- Pipeline
- Activity
- Data Sets
- Linked Services
- Triggers

Piepline

- A data factory might have one or more pipelines. A pipeline is a logical grouping of activities that performs a unit of work.

  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.

Activity

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

Linked services & Data sets

- Linked services are much like connection strings, which define the connection information that's needed for Data Factory to connect to external resources.
- Datasets represent data structures within the data stores, which simply point to or reference the data you want to use in your activities
- 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.

Triggers

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

Example :-

ADF - its an Azure Service

ETL:
E - extarction
T - transmation
L: load

SSIS - ETL pipelines - desitination data - Pwer BI.

Example:
Game -
logs - Azure data lake stiorgae : Bigdata

Customer infor and market campings - On primes storage

extarct

Transform

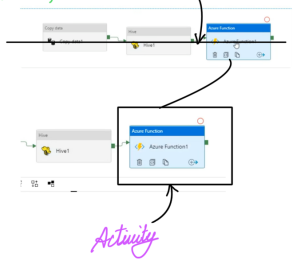
Load - Azure SQL DB

Azure data factory

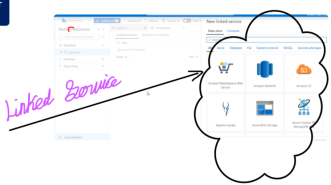
PowerBI -

Example :-

pipe Line



Activity



Linked Service

* Trigger is used to run a pipe line at Specific times like a Schedulad Job.

* Linked Service is Setting up a Connection b/w Azure and another data Source.
* Data Set - it point to Exact data you want to use.

Example :- Linked Service get's you into the building, and data Set gets you into the Specific room you need

Make Sense

2) Triggers :-

Azure Data Factory

- Azure Portal UI
- Azure PowerShell(Install Azure PowerShell)
- .NET
- Python
- REST
- Resource Manager Template(Azure PowerShell Az Module)

* Azure Data factory we can Create by using all this technology.

Types of Activities

- Data Movement Activities
- Data transformation Activities
- Control flow Activities

Ex. to transfer one place to another place.

Ex. U-Sql

Ex. If-Condition

Linked Services and Datasets

- Linked services are used to connect Other resources with Azure Data factory. Linked services are like connection strings for resources to connect
- Datasets are simply points or references the data, which you want to use in your activities as input or output

Explanation :-

* Activity is going to Consume the data first from any Storage and perform the activity and produce data.
* The group of activity called as pipeline.
* Linked Services holds the Connection of Data Sets
* Data Sets required linked Services

Triggers in Azure Data Factory

- Triggers – you can execute your pipeline.
- Triggers determines when a pipeline execution needs to be kicked off.
- Pipelines and triggers have a many-to-many relationship (except for the tumbling window trigger)
- Multiple triggers can kick off a single pipeline, or a single trigger can kick off multiple pipelines.

Event based Triggers in ADF

- An event-based trigger runs pipelines in response to an event, such as the arrival of a file, or the deletion of a file, in Azure Blob Storage.
- Data integration scenarios often require Data Factory customers to trigger pipelines based on events such as the arrival or deletion of a file in your Azure Storage account.
- Data Factory is now integrated with Azure Event Grid, which lets you trigger pipelines on an event.

Types of Triggers in Azure Data Factory

- Below are the Types of Triggers available in Azure Data Factory
  1. Schedule Trigger - A trigger that invokes a pipeline on a wall-clock schedule.
  2. Tumbling Window Trigger - A trigger that operates on a periodic interval, while also retaining state.
  3. Event-based Trigger - A trigger that responds to an event.

Example :

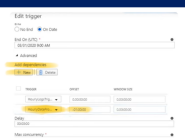
* Trigger can be spent Monthly (or) Daily (or) in Specific time.
* Tumbling window will work for historical (or) past date.
* Scheduled trigger will work Start of current time.

Tumbling Window Trigger in ADF

- Tumbling window triggers are a type of trigger that fires at a periodic time interval from a specified start time, while retaining state.
- A tumbling window trigger has a one-to-one relationship with a pipeline and can only reference a singular pipeline.

Self dependency Trigger

- Self dependency Trigger is going to dependent on its own. We have to provide offset for this in negative only



Tumbling Window Trigger Dependency in ADF

- In Order to build dependency chain and make sure that a trigger is executed only after successful execution of another trigger using Tumbling window Trigger Dependency Feature.

Note: A tumbling window trigger can depend on a maximum of two other triggers.

- You will be having access to Window Start time and Window End Time values using below System Properties
  - trigger().outputs.windowStartTime
  - trigger().outputs.windowEndTime

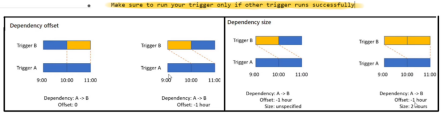
Example:

HourlyLogsPipeline - process logs for every hour and load data in to SQL

data from this

HourlyDataProcessPipeline - take logs hourly data + cust data and it will do some transfomation

Make sure to run your trigger only if other trigger runs successfully



Integration Runtime :-

Integration Runtime in ADF

- The Integration Runtime (IR) is the compute infrastructure used by Azure Data Factory to provide the following data integration capabilities across different network environments:
  - → **Data Flow:** Execute a Data Flow in managed Azure compute environment
  - → **Data Movement:** Copy data across data stores in public network and private network
  - → **Activity Dispatch:** Dispatch and monitor transformation activities running on a variety of compute services
  - → **SSIS Package Execution:** Execute SQL Server Integration Services (SSIS) packages in a managed Azure compute environment.

Azure Integration runtime

- Azure integration runtime is capable of performing below in public network.
  - →Running DataFlows
  - →Running Copy Activities between Cloud data stores
  - → Running Transform Activities
- Azure Integration Runtime supports connecting to data stores and compute services with public accessible endpoints.
- Azure integration runtime provides a fully managed, serverless compute in Azure
- Azure IR is elastically scaled up accordingly without you having to explicitly adjusting size of the Azure Integration Runtime.

Self-Hosted Integration runtime

- Self-Hosted Integration runtime is capable
  - Performing data movement activities between cloud data stores and in private network
  - Running transform activities against compute resources in on-premises or Azure virtual network
- Self-hosted IR needs to be installed on an on-premises machine or a virtual machine inside a private network. Currently, its supports running the self-hosted IR on a Windows operating system.

Example :-

Linked service - Target resource data store or compute service.

Activities - Task which u want to perform

Integration runtime as bridge - LS and ACT

Integration Runtime [IR] refered by azure Data factory to perform data integration capabilities.

Azure Integration runtime

- Azure integration runtime will come by default with location as auto-resolve.

Create Azure Integration runtime

- You only need to explicitly create an Azure IR when you would like to explicitly define the location of the IR, or if you would like to virtually group the activity executions on different IRs for management purpose.

Example :- Expression keyword

SQL server:

10 DBs

@LinkedService().

creating 10 linked services for all 10 DBs.

Prameterization....

parameterize Linked Service

* you can parameterize a linked service and pass dynamic values at run time.

System Variables in ADF

- System Variables are available at below three scopes. We can use these variables in expressions in Azure data factory
  - → Pipeline Scope
  - → Schedule Trigger Scope
  - → Tumbling Window Trigger Scope

Example :-

Pipeline Scope System Variables:

Copy data from one storage to another and then log details of pipeline execution in to SQL DB.

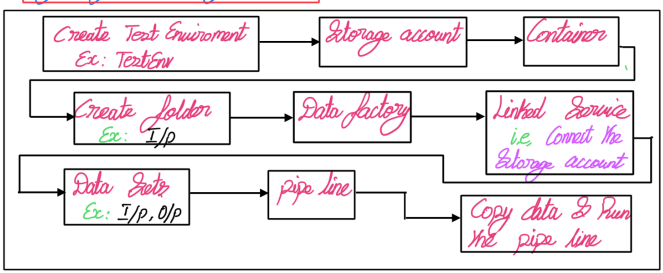
Schedule trigger Scope System Variables:

Copy data one storage to another storage daily for the given date.

Tumbling Windows Trigger Scope System Variables:

Copy data one storage to another storage for every hours

Layout for ADF (My understand)



Connectors, Copy data, Monitor :-

Connectors in ADF

- Azure Data Factory supports more than 80 data stores to work with
- Supported Formats

Azure data lake is build on Apache Hadoop.

Avro:

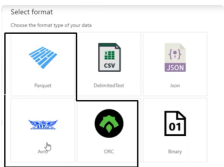
Its a row based storage format for Hadoop.
Avro stores data definition(schema) in JSON format making it easy to read and interpret by any program

Parquet: Its Column based storage format

File Formats in ADF

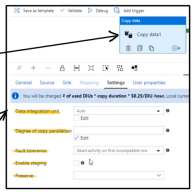
- Supported Formats

Binary Format: Text files
JSON Format: JSON files
Delimited text Format: csv files(csv – comma separated values)



Parquet, ORC & Avro

- ORC, Avro, Parquet are the formats which are part of Apache Hadoop eco system
- All 3 formats works on compression algorithms. Data will stored in compression hence query results will be much faster
- Using ADF, if you read data from SQL Table and load that data in to ORC, Parquet, Avro, Txt, JSON files then JSON and TXT files size will be very more compare to ORC, Parquet & Avro



Copy Data Activity in ADF

- Copy data activity is core activity in ADF. You can copy data from more that 90 connectors one to another.
- Connector Specific properties

Settings section in Copy Data Activity in ADF

- Data Integration Units – Think its like a combination of CPU, memory and networking power.
- Degree of Parallelism – number to connection or threads to perform read or write
- Copy data can copy data from csv file to SQL Table

Note:- If the Staging check box is Enabled it will move the data to staging but it is a temporary table when the Execution completes data will be lost.

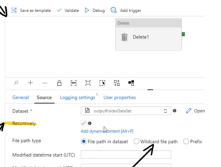
Monitor Copy Data Activity in ADF

- Once you've created and published a pipeline in Azure Data Factory, you can associate it with a trigger or manually kick off an ad hoc run. You can monitor all of your pipeline runs natively in the Azure Data Factory user experience

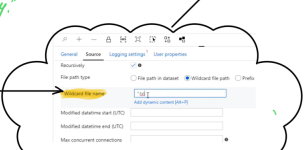
Delete Activity in ADF

- You can use the Delete Activity in Azure Data Factory to delete files or folders from on-premises storage stores or cloud storage stores
- Deleted files or folders cannot be restored.

https://docs.microsoft.com/en-us/azure/data-factory/delete-activity



Whatever files in the folder it will delete "Recursively"



It will delete by the format whatever we are Entering in the box i.e, *.txt

Variables, Execute pipe line, Filter Activity :-

Ex :-

Variables in ADF

- Variables are like internal to pipeline and they can be changed inside your pipeline
- Variables support 3 data types: string, bool, array
- We refer these user variables as below:

@variables('variableName')

* Variables - using in pipe line
* parameters - using in trigger

Set Variable Activity in ADF

Variables are internal to your pipelines

you can variables values u can change - set variable & append variable

- Use the Set Variable activity to set the value of an existing variable of type String, Bool, or Array defined in a Data Factory pipeline.

User Properties in ADF

- User Properties helps to view addition information while monitoring Activity runs
- You can only create only 5 properties under User Properties

pipe line tab

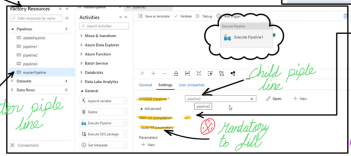
Note : Variables are in under General tab drag and drop.

(o/p)

Monitor tab

Execute Pipeline Activity in ADF

- The Execute Pipeline activity allows a Data Factory pipeline to invoke another pipeline.

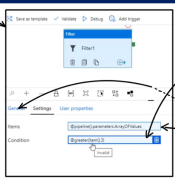


Master pipe line

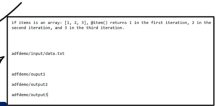
once the child pipe line get Executed then only the master pipe line go for next activity

Filter Activity in ADF

- You can use a Filter activity in a pipeline to apply a filter expression to an input array.



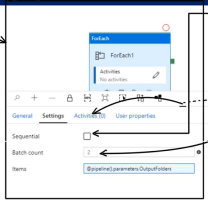
Note: Condition will check the item has greater than 3 Item variable are declared in General



ForEach Activity in ADF

- ForEach Activity defines a repeating control flow in your pipeline. This activity is used to iterate over a collection and executes specified activities in a loop
- The items property is collection and each item inside collection is referred by @item()

Ex: o/p → f-1 'f' - folder
f-2
f-3



It will Execute the f-1 after that it will go activities tab work then Execute f-2 then go to activities tab it run until f-3 like a loop

The Batch Count If its '2' It will run f-1, f-2 then go to activities tab Then it will come back to loop.

Meta data, If Condition, wait/unit/Web Activity :-

Get Metadata Activity in ADF

- You can use the Get Metadata activity to retrieve the metadata of any data in Azure Data Factory.

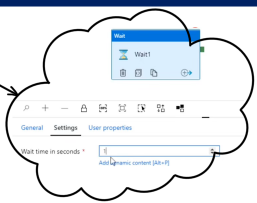
Metadata type	Description
Name	Name of the file or folder.
Type	Type of the file or folder. Returned value is `File` or `Folder`.
	Size of the file, in bytes. Applicable only to files.
ted	Created datetime of the file or folder.
Modified	Last modified datetime of the file or folder.
childItems	List of subfolders and files in the given folder. Applicable only to folders. Returned value is a list of the name and type of each child item.
contentMD5	MD5 of the file. Applicable only to files.
structure	Data structure of the file or relational database table. Returned value is a list of column names and column types.
columnCount	Number of columns in the file or relational table.
exists	Whether a file, folder, or table exists. Note that if `exists` is specified in the Get Metadata field list, the activity won't fail even if the file, folder, or table doesn't exist. Instead, `exists: false` is returned in the output.

If Condition Activity in ADF

- If Condition activity provides the same functionality that an if statement provides in programming languages.
- If the expressions resolves to true, then set of activities will run and if the expression resolves to false then another set of activities will run

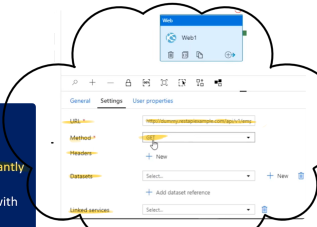
Wait Activity in ADF

- When you use a Wait activity in a pipeline, the pipeline waits for the specified period of time before continuing with execution of subsequent activities.



unit Activity in ADF

- This activity is like Do Until or Do While Activities in programming languages. That means, it is guarantee that at least one loop will defiantly run as condition evaluation will happen at the end of loop.
- It executes a set of activities in a loop until the condition associated with the activity evaluates to true



Web Activity in ADF

- Web Activity can be used to call a custom REST endpoint from a Data Factory pipeline.
- You can pass Datasets and Linked Services also to REST API
- Web Activity can call only publicly exposed URLs. It's not supported for URLs that are hosted in a private virtual network

WebHook Activity in ADF

- WebHook Activity we can call an endpoint and pass it a callback URL. The pipeline run waits for the callback invocation before it proceeds to the next activity.
- Because of this WebHook Activity in Synchoronocs in nature.

web activity & WebHook activity are Similar