Heartbeat AZURE DATA FACTORY

Pipeline Overview:

In all, we have Four pipelines for this whole process of Data Movement Form Azure File share as .gz file to Azure Blob Storage as. JSON.

So, as we started to create a pipeline to simply move data from the source to the destination, we faced some exceptions while moving files when we were testing these pipelines.

Exceptions are as follows,

1) In AZURE DATA FACTORY Copy activity is not able to extract the corrupted file.

2) The Fault tolerance option is not available with the JSON dataset in the Copy activity.

3) The copy activity will fail even if there is one corrupt file that occurs in the ongoing data movement and because of which the whole data will remain in the same place.

Pipeline Description:

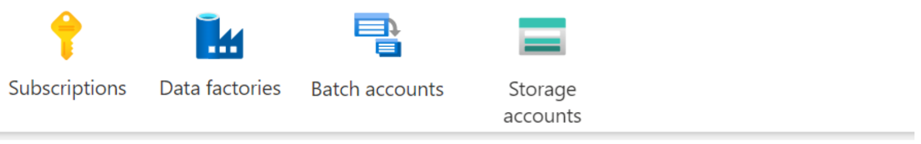
Pipeline Name	Overview
01_DataMove_BatchTime	This pipeline will run in every 1hr and move data from Azure file share to Azure Blob Storage in a 15 min batch each. So, for 1hr Execution 4 batches will be run for 15 min each
02_ErrorFiles_ExceptionHandling	This pipeline will Trigger only if any corrupt file occurs in a specific 15 min batch and that data will be validated by Python code to detect the corrupt file and will move those files to "Error" directory of the heartbeatlogs File Share and the rest file will move from "adfbatchdumps" folder in heartbeatlogs File Share to "heartbeat" Blob Storage
03_FallBack_DataMove_BatchTime	This pipeline would run every 3hr to cross-check if any files are left for the past 3hr in the azure file share rest the overall structure is almost same as the "01_DataMove_BatchTime" pipeline
04_DataMove_Custom_DateTime_Batch	This pipeline works with custom date time for example if for some reason yesterday 01_Pipeline fails from 1 pm to 6 pm so to get the data from files shared to blob we can use this pipeline and pass the start time as 1 pm and end time as 6 pm.

Note: The pipeline mentioned in the above table will be described in detail in the below structure.

- Overview of Pipeline
- Problem Statement
- Solution
- Pipeline Structure
    - Link service
    - Datasets
    - Parameters, Variable, Activity: Details
    - Activity, Dependency
    - Trigger/Debug

Prerequisite:

- Azure Subscription
- Azure Data factory
- Azure Storage Account
- Azure Batch Account

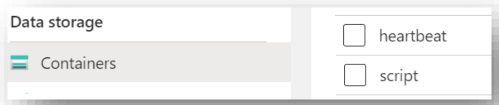


No	Index
1	Azure Storage Account
2	Azure Batch Account
3	Linked Services
4	Datasets
5	Pipeline • Overview of Pipeline • Problem Statement • Solution • Pipeline Structure ○ Link service ○ Datasets ○ Parameters, Variable, Activity: Details ○ Activity, Dependency ○ Trigger/Debug
6	Python Code
7	Alerts & Metrics
8	Point's to be Noted

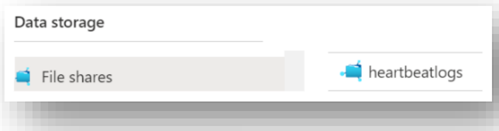
Azure Storage Account

Storage Account Name: “enprodeventhubtest”: In the azure storage account, we need to have some Folders before starting with the Data factory some folders in files Share, and some containers in the Blob Storage account as follows -

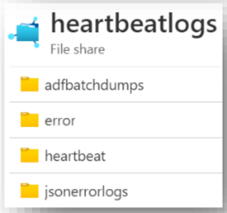
Azure Blob Storage: **Containers**: [heartbeat, script]



Azure File Share: File Share: [heartbeatlogs**]**



Folders: [heartbeat, jsonerrorlogs, adfbatchdumps, error]



Storage Account: Containers /Folder Structure.

Storage Account	**“enprodeventhubtest”**	**Folder**	**Sub-Folder**
Containers	“heartbeat”	“/YYYY/MM/DD/HH/*.json”	
	“script”	main.py	
File Share	**“heartbeatlogs”**	“heartbeat”	“/YYYY/MM/DD/HH/*.gz”
		“jsonerrorlogs”	ErrorLogs.json
		“adfbatchdumps”	“/YYYY/MM/DD/HH/*.gz”
		“error”	.gz

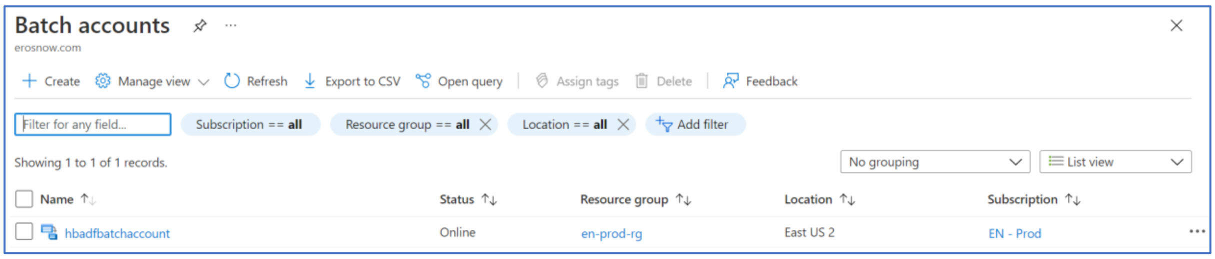
Folder Objective:

Containers	**Description**
“heartbeat”	This Container will save all data in JSON format which will be passed through AZURE DATA FACTORY
“script”	This Container will have one python script file which will check for error files.
File Share	
“heartbeat”	This Folder contains data in .gz format files which will be generated by the heartbeat application over the Network.
“jsonerrorlogs”	This Folder contains an ErrorLogs.json file which contains a list of files name that is invalid or corrupted.
“adfbatchdumps”	This Folder will work as staging in this whole process of data movement when any error file in there in the pipeline batch the whole batch data will move to this Folder and AZURE DATA FACTORY will handle this separately. **Note: Do not put any data in this folder manually and do not make any changes in this folder**
“error”	The Error file detected by AZURE DATA FACTORY will be Moved to this “error” folder **Note: you must manually investigate these Files**

Azure Batch Account

We need an Azure Batch account and one pool in this account which will give us a VM that will help python code to Compile and get the results containing the error file name along with its path which would be logged in Errorlogs.json as output.

1st Need a batch Account.

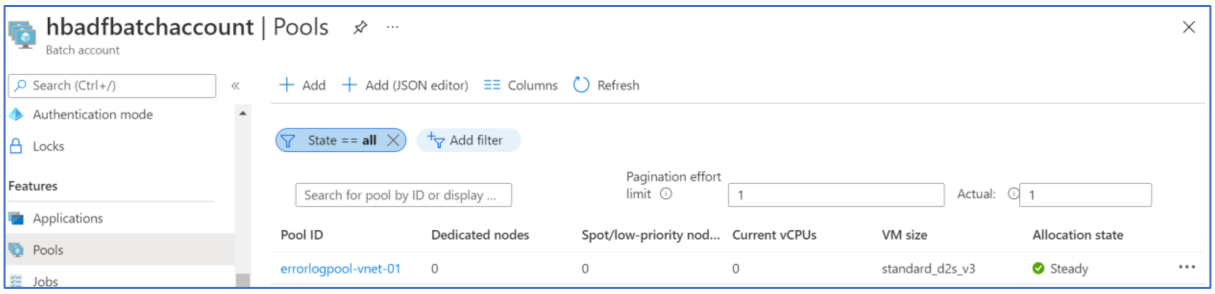


Subscription == all | Resource group == all | Location == all

Showing 1 to 1 of 1 records.

Name	Status	Resource group	Location	Subscription
hbadfbatchaccount	Online	en-prod-rg	East US 2	EN - Prod

2nd Need to have a Pool in that.



State == all

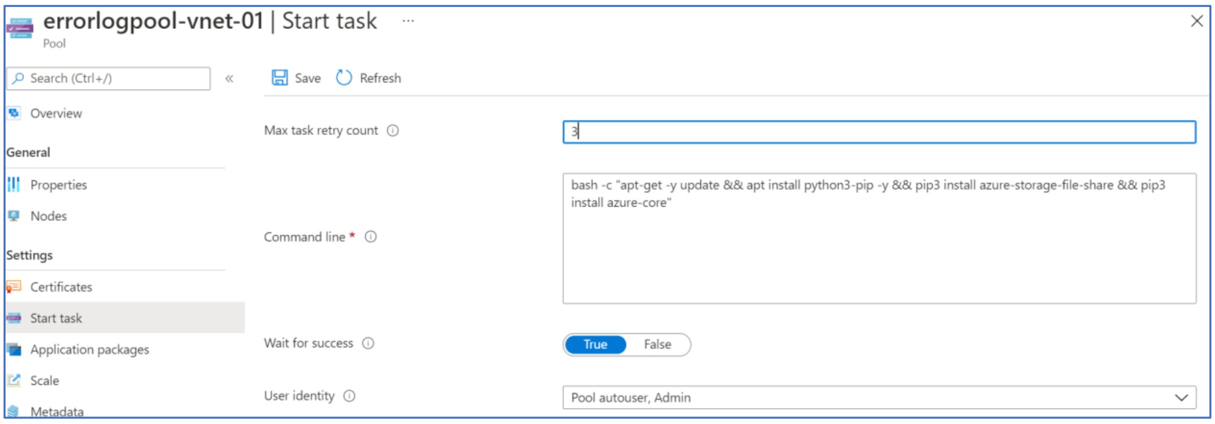
Pagination effort limit: 1 Actual: 1

Pool ID	Dedicated nodes	Spot/low-priority nod...	Current vCPUs	VM size	Allocation state
errorlogpool-vnet-01	0	0	0	standard_d2s_v3	Steady

3rd Configure Start Task of the same pool

Command Line.

```
bash -c "apt-get -y update && apt install python3-pip -y && pip3 install azure-storage-file-share && pip3 install azure-core"
```



Max task retry count: 3

Command line *: bash -c "apt-get -y update && apt install python3-pip -y && pip3 install azure-storage-file-share && pip3 install azure-core"

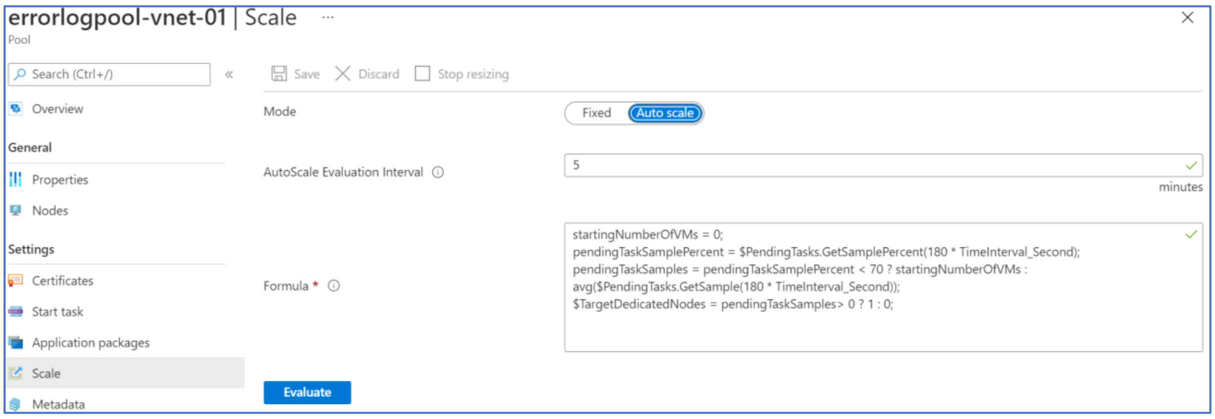
Wait for success: [✓ True] False

User identity: Pool autouser, Admin

4th Scale The pool for cost optimization.

#When there are no jobs 0 VM on and when the job arrived Max VM on is 1 **Formula** below.

```
startingNumberOfVMs = 0;
pendingTaskSamplePercent = $PendingTasks.GetSamplePercent(180 * TimeInterval_Second);
pendingTaskSamples = pendingTaskSamplePercent < 70? startingNumberOfVMs :
avg($PendingTasks.GetSample(180 * TimeInterval_Second));
$TargetDedicatedNodes = pendingTaskSamples> 0 ? 1 : 0;
```



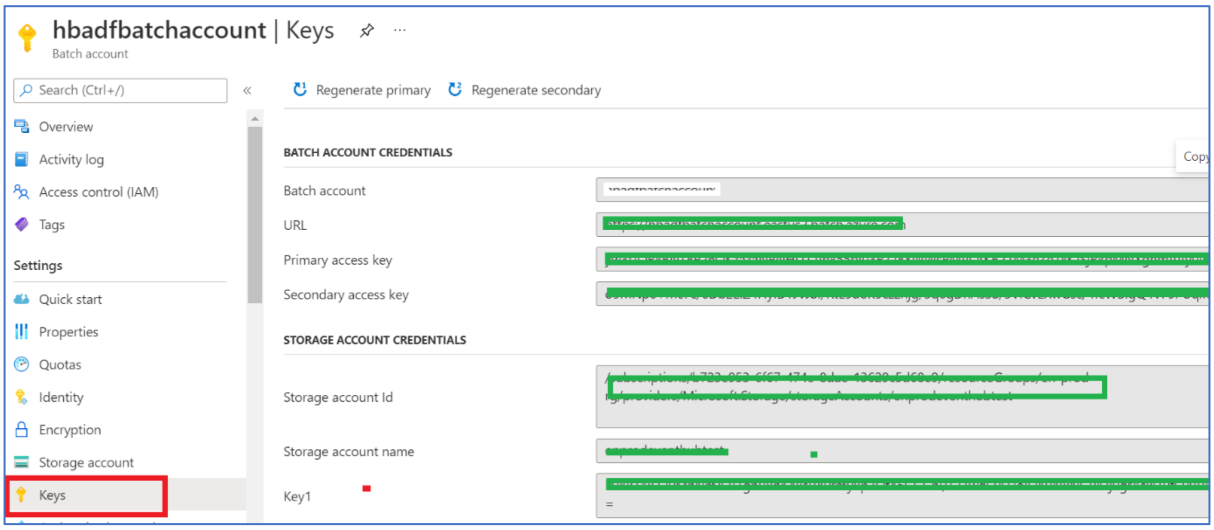
Mode: Fixed [✓ Auto scale]

AutoScale Evaluation Interval: 5 minutes

Formula *:

```
startingNumberOfVMs = 0;
pendingTaskSamplePercent = $PendingTasks.GetSamplePercent(180 * TimeInterval_Second);
pendingTaskSamples = pendingTaskSamplePercent < 70 ? startingNumberOfVMs :
avg($PendingTasks.GetSample(180 * TimeInterval_Second));
$TargetDedicatedNodes = pendingTaskSamples> 0 ? 1 : 0;
```

5th Keys & Credential Key's will be required in AZURE DATA FACTORY To Configure Batch Account Link-Services



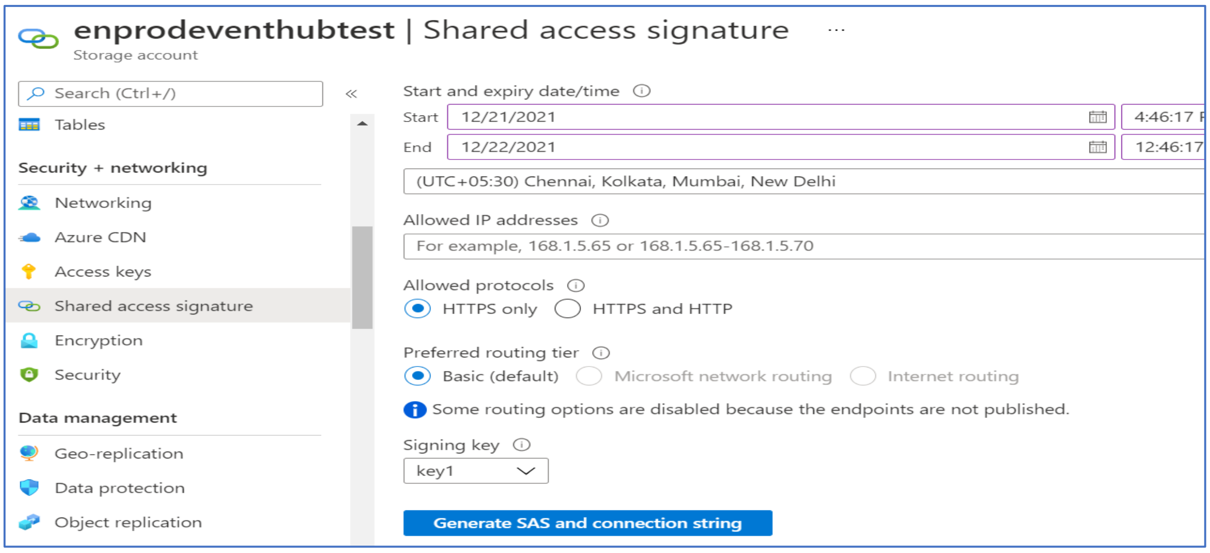
BATCH ACCOUNT CREDENTIALS

Batch account	[redacted]
URL	[redacted]
Primary access key	[redacted]
Secondary access key	[redacted]

STORAGE ACCOUNT CREDENTIALS

Storage account Id	[redacted]
Storage account name	[redacted]
Key1	[redacted]

5.1 SAS Key we will require in Python Code. To get the key click on generate SAS keys.


enprodeventhubtest | Shared access signature ...

Storage account

Start and expiry date/time

Start 12/21/2021 4:46:17 P

End 12/22/2021 12:46:17

(UTC+05:30) Chennai, Kolkata, Mumbai, New Delhi

Allowed IP addresses

For example, 168.1.5.65 or 168.1.5.65-168.1.5.70

Allowed protocols

[✓ HTTPS only] HTTPS and HTTP

Preferred routing tier

[✓ Basic (default)] Microsoft network routing Internet routing

Some routing options are disabled because the endpoints are not published.

Signing key

key1

Generate SAS and connection string

Linked Services

Linked services are much like connection strings, which define the connection information needed for the service to connect to external resources.

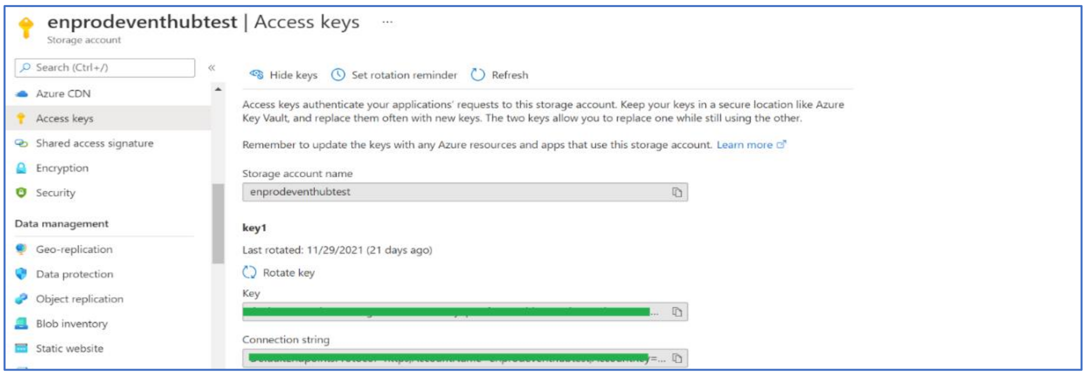
We must create 3 Linked Services for our AZURE DATA FACTORY and all pipeline to work.

No	**Linked Services Name**	**Storage account name**	**File share**
1	filesharetest	enprodeventhubtest	heartbeatlogs
2	blobstoragetest	enprodeventhubtest	Null
		Account Name	**Pool Name**
3	AzureBatchfreshpool_link	hbadfbatchaccount	errorlogpool-vnet-01

While Configuring the **Linked Services it will ask you keys and some other details configuration**

Keys you will get those from the Azure portal for all the services follow below screenshot

1st For “**filesharetest**”, “**blobstoragetest**” as we have the same storage account for file share and blob the key will be the same for both the **Linked Services.**


enprodeventhubtest | Access keys

Storage account

Access keys authenticate your applications' requests to this storage account. Keep your keys in a secure location like Azure Key Vault, and replace them often with new keys. The two keys allow you to replace one while still using the other.

Remember to update the keys with any Azure resources and apps that use this storage account. Learn more

Storage account name: enprodeventhubtest

key1

Last rotated: 11/29/2021 (21 days ago)

Rotate key

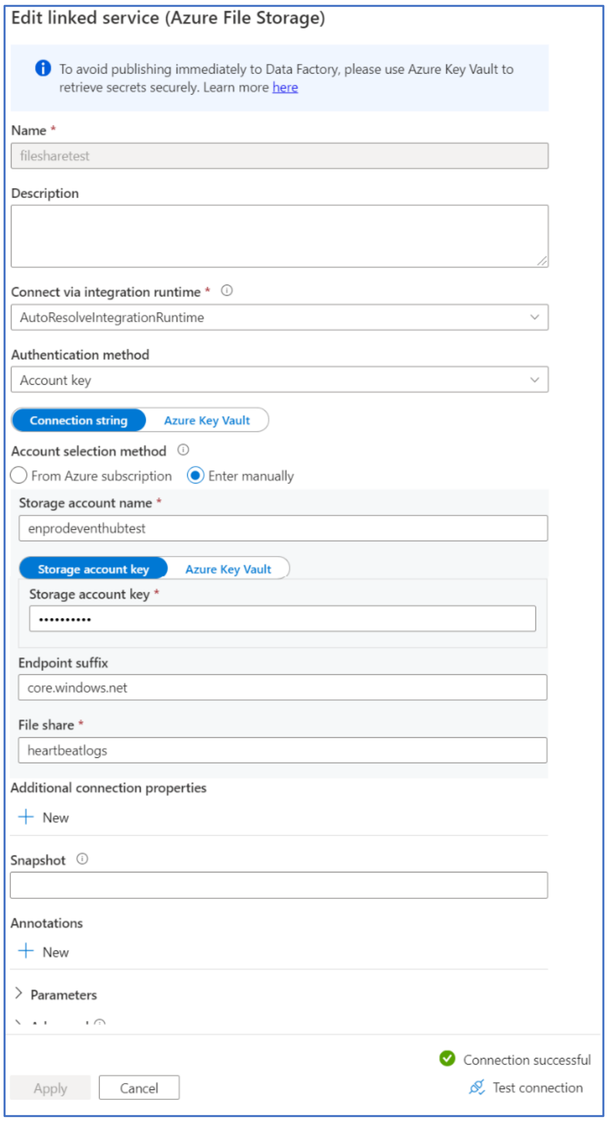
Key: [redacted]...

Connection string: [redacted]=...

2nd For AzureBatchfreshpool_link keys are mentioned in the above steps of **Azure Batch Account** 5th step's

Note: Below is the screenshot of the Link service in AZURE DATA FACTORY

Azure file Share: filesharetest


Edit linked service (Azure File Storage)

To avoid publishing immediately to Data Factory, please use Azure Key Vault to retrieve secrets securely. Learn more here

Name *: filesharetest

Description:

Connect via integration runtime *: AutoResolveIntegrationRuntime

Authentication method: Account key

[✓ Connection string] Azure Key Vault

Account selection method: From Azure subscription [✓ Enter manually]

Storage account name *: enprodeventhubtest

[✓ Storage account key] Azure Key Vault

Storage account key *: ••••••••••

Endpoint suffix: core.windows.net

File share *: heartbeatlogs

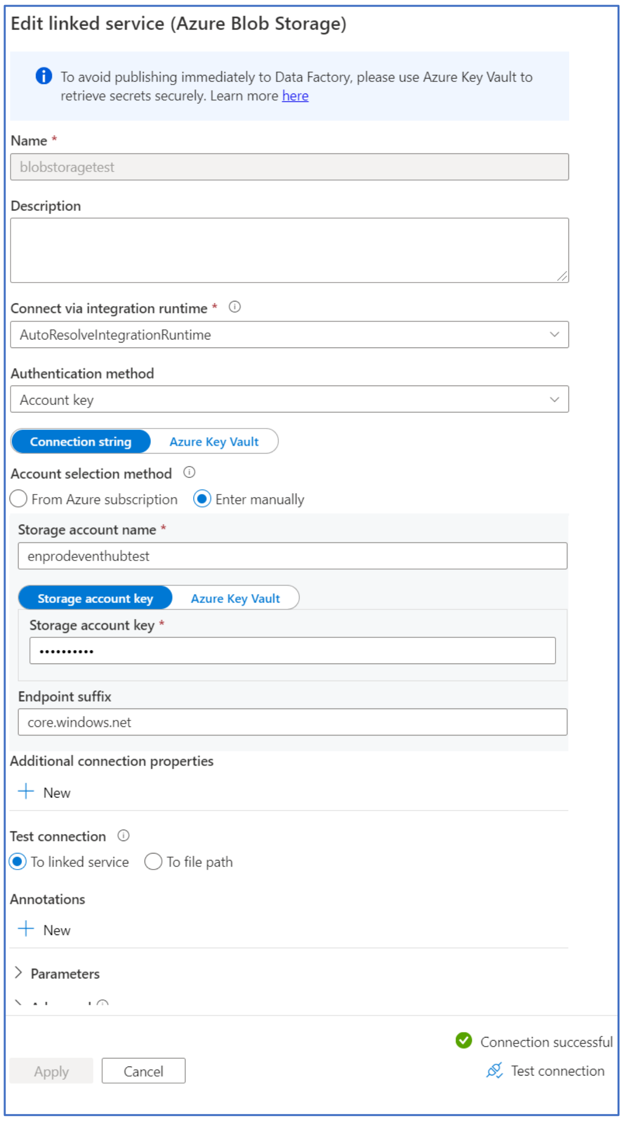
Additional connection properties

Snapshot:

Annotations

Connection successful

Azure Blob Storage: blobstoragetest


Edit linked service (Azure Blob Storage)

To avoid publishing immediately to Data Factory, please use Azure Key Vault to retrieve secrets securely. Learn more here

Name *: blobstoragetest

Description:

Connect via integration runtime *: AutoResolveIntegrationRuntime

Authentication method: Account key

[✓ Connection string] Azure Key Vault

Account selection method: From Azure subscription [✓ Enter manually]

Storage account name *: enprodeventhubtest

[✓ Storage account key] Azure Key Vault

Storage account key *: ••••••••••

Endpoint suffix: core.windows.net

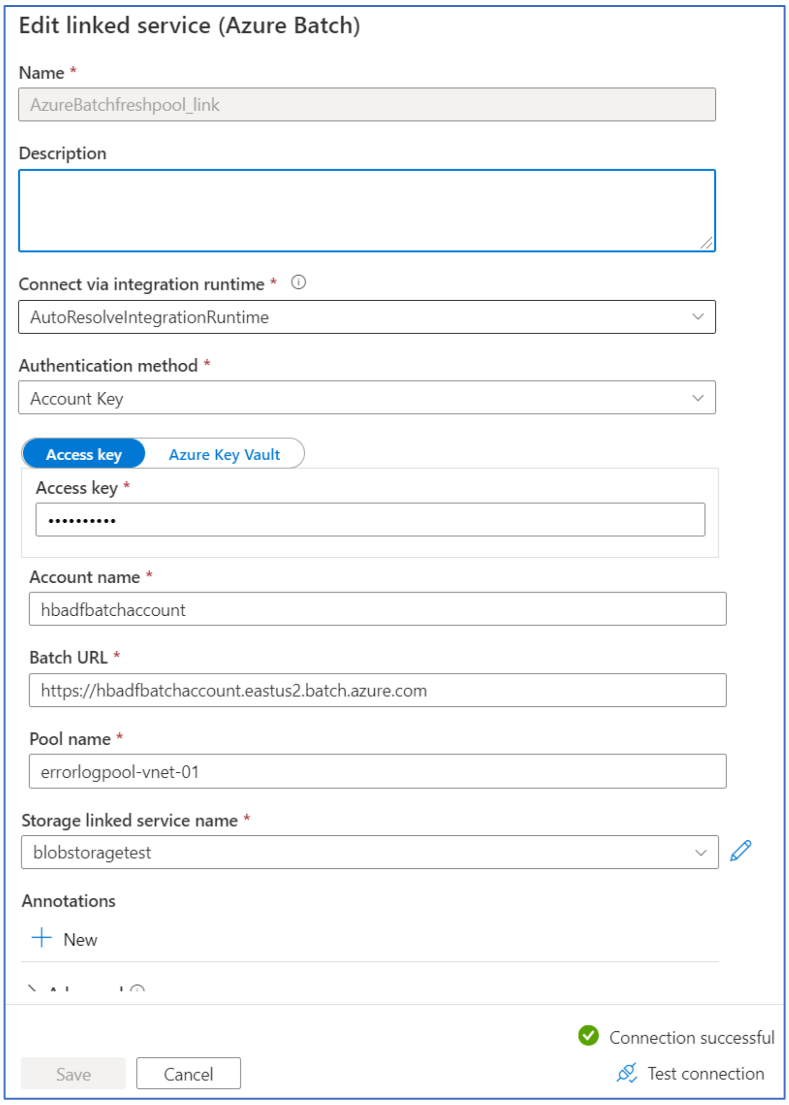
Additional connection properties

Test connection: [✓ To linked service] To file path

Annotations

Connection successful

Azure Batch: AzureBatchfreshpool_link



Edit linked service (Azure Batch)

Name *
AzureBatchfreshpool_link

Description

Connect via integration runtime *
AutoResolveIntegrationRuntime

Authentication method *
Account Key

Access key *
••••••••••

Account name *
hbadfbatchaccount

Batch URL *
https://hbadfbatchaccount.eastus2.batch.azure.com

Pool name *
errorlogpool-vnet-01

Storage linked service name *
blobstoragetest

Annotations

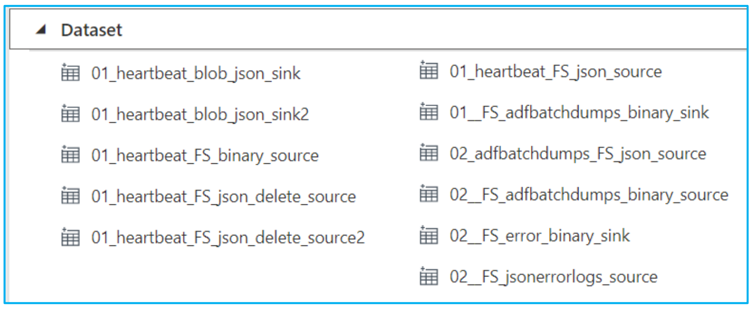
Connection successful

Datasets

The activities in a pipeline define actions to perform on your data. Now, a dataset is a named view of data that simply points to or references the data you want to use in your activities as inputs and outputs. Datasets identify data within different data stores, such as tables, files, folders, and documents.

We have 10 datasets that will be used to set as a source or sink in our activities in different pipelines.

Below is a screenshot of all the datasets witch we will require in the pipeline.



Dataset

- 01_heartbeat_blob_json_sink
- 01_heartbeat_blob_json_sink2
- 01_heartbeat_FS_binary_source
- 01_heartbeat_FS_json_delete_source
- 01_heartbeat_FS_json_delete_source2
- 01_heartbeat_FS_json_source
- 01__FS_adfbatchdumps_binary_sink
- 02_adfbatchdumps_FS_json_source
- 02__FS_adfbatchdumps_binary_source
- 02__FS_error_binary_sink
- 02__FS_jsonerrorlogs_source

Pipeline link-service

No	Pipeline Name	Type	Link-Service
01	01_DataMove_BatchTime	Blob	blobstoragetest
02	02_ErrorFiles_ExceptionHandling	File Share	filesharetest
03	03_FallBack_DataMove_BatchTime		
04	04_DataMove_Custom_DateTime_Batch		

Dataset Details Table

N0	Name	Format	L-S	Pipeline Numb: Activity: use as in
1	01_heartbeat_blob_json_sink	JSON	Blob	01: Copy_To_Destination_Blob: Sink
2	01_heartbeat_FS_binary_source	Binary	File Share	02: Copy_Batch_To_Dumps: Source
3	01_heartbeat_FS_json_delete_source	JSON	File Share	01: Delete_From_Source_FileShare: Source
4	01_heartbeat_FS_json_delete_source2	JSON	File Share	03: Delete_From_Source_FileShare: Source 04: Delete_From_Source_FileShare: Source
5	01_heartbeat_FS_json_source	JSON	File Share	01: Copy_To_Destination_Blob: Source 03: Copy_To_Destination_Blob: Source 04: Copy_To_Destination_Blob: Source
6	01__FS_adfbatchdumps_binary_sink	Binary	File Share	02: Copy_Batch_To_Dumps: Sink
7	02_adfbatchdumps_FS_json_source	JSON	File Share	02: Copy_To_Destination_Blob_2: Source 02: Delete_From_Source_AZURE DATA FACTORYDumps_FileShare: Source
8	02__FS_adfbatchdumps_binary_source	Binary	File Share	02: Move_Error_Files: Source
9	02__FS_error_binary_sink	Binary	File Share	02: Move_Error_Files: Sink
10	02__FS_jsonerrorlogs_source	JSON	filesharetest	02: Lookup_Json_ErrorFiles: Source
11	01_heartbeat_blob_json_sink2	JSON	Blob	02: Copy_To_Destination_Blob_2: Sink 03: Copy_To_Destination_Blob: Sink 04: Copy_To_Destination_Blob: Sink

Pipeline: 01_DataMove_BatchTime

Problem Statement:

We must move data from Azure File share to Azure blob storage using AZURE DATA FACTORY Pipeline. we were facing the problem of inconsistent data movement concerning the time So for 11GB of data movement, it's taking more than an hour and for 15 min of the pipeline, it's taking more than 15min to complete pipeline.

To move the data with good speed and data size and in the expected duration, we need a better approach.

Solution:

we can divide the pipeline into small batches of time for example data move for the last 1hr which is 60 min can be divided into 4 groups of 15 min each then these 4 batches will run at the same time to achieve the speed once we get this the pipeline will end in expected duration

and, when we will divide this into time batches the load on the pipeline will also divide data among 4 batches through which we will achieve consistency in data size.

Overview of Pipeline:

Note: This is our main pipeline.

01_DataMove_BatchTime: In this pipeline, we have 2 parameters 4 variables 10 activities, and 1 trigger. This pipeline will run every 1 hour every day with the help of Trigger.

Pipeline Structure

Link service:

NO	Name	Describe
1	filesharetest	This link service has been configured with Azure File Share which is "heartbeatlogs" in "enprodeventhubtest" storage account
2	blobstoragetest	This link service has been config with Blob Storage Container which is "heartbeat" in "enprodeventhubtest" storage account

Datasets:

N0	Name	Format	L-S	Activity Name: use as in
1	01_heartbeat_blob_json_sink	JSON	Blob	Copy_To_Destination_Blob: Sink
2	01_heartbeat_FS_json_delete_source	JSON	File Share	Delete_From_Source_FileShare: Source
3	01_heartbeat_FS_json_source	JSON	File Share	Copy_To_Destination_Blob: Source

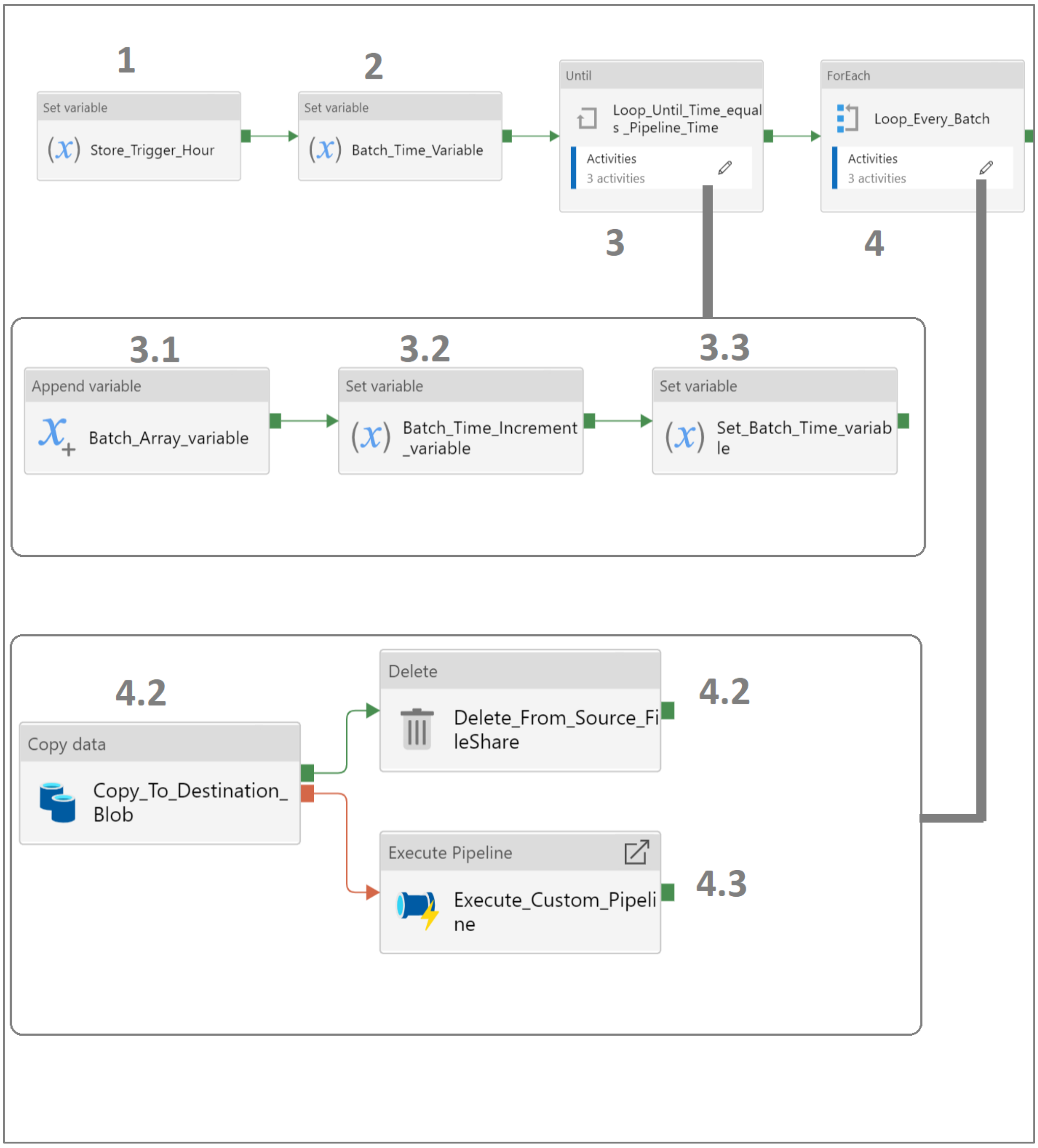
Parameters, Variable, Activity:

Parameters			
Name	Type	Default Value	Explanation
Pipeline_Time	String	60	This value will decide the duration of the data to be moved if we pass 60 that means for last 60 min of data will be moved from source to destination
Batch_Size	String	15	This value will divide the above 60 min into batches of 15min each that is 4 batches of 15min [0-15],[15-30],[30-45],[45-60]

Variables			
Name	Type	Default value	Explanation
Batch_Array	Array	Null	This will load an array of time like for time range 1:00 pm to 2:00 pm this would be [1,1:15] [1:15,1:30][1:30:1:45][1:45,2:00] pass this array in sequence in a loop
Batch_Time	String	Null	This variable converts the Batch_Size value to negative and store that value.
Batch_Time_Increment	String	Null	This variable Increate the time by 15 min in each loop.
Time_Path	String	Null	This variable will take the time Hour only and the minutes will become 00 for example if it’s 1:10 pm it will convert to 1:00 pm and pass this hour as the Folder path.

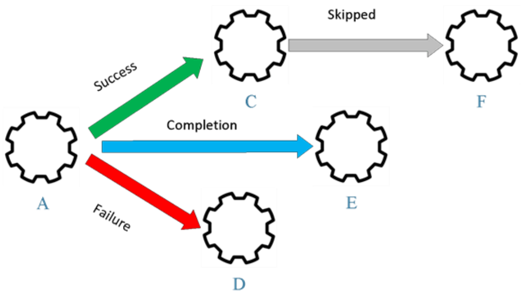
No	Activities	Name	Working
1	Set variable	Store_Trigger_Hour	This will Convert the Current UTC Hour to String and Pass The same in File Path of 4.1 Copy data
2	Set variable	Batch_Time_Variable	This will load the batch time from the parameter
3	until	Loop_Until_Time_equals _Pipeline_Time	This until activity will loop till the given time meets the condition
3.1	Append variable	Batch_Array_variable	This will load Batch Time 15 min
3.2	Set variable	Batch_Time_Increment_variable	This will increase Batch Time 15*1,1*2, and so on in this 1,2 is the count of the loop
3.3	Set variable	Set_Batch_Time_variable	This will load the previous Bath time 1 loop 15, 2nd loop 30,3 loops 45, 4th loop 60
4	ForEach	Loop_Every_Batch	The until loop output Of Array time will be passed in this loop array will be [0:15][15:30][30:45][45:60]
4.1	Copy Data	Copy_To_Destination_Blob	This activity will Copy data from File Share to blob
4.2	Delete	Delete_From_Source_FileShare	This activity will delete the data from the file share
4.3	Execute Pipeline	Execute_Custom_Pipeline	This activity will call 02 pipelines in case of any error occurs.

Activity, Dependency



Dependency

- **Success -** If activity C has a Success dependency condition
  on activity A, it only runs, if activity A succeeds.
- **Failure -** If activity D has Failure dependency condition on activity A, it only runs, if activity A fails.



No	Activities	Dependency	Name	Configs	Sub-Configs	Syntax
1	Set variable	Success	Store_Trigger_Hour			
				Variables:	**Name:**	**Time_Path**
					value:	"@concat(formatDateTime(pipeline().TriggerTime,'HH'),':00')"
2	Set variable	Success	Batch_Time_Variable	**Variables:**	**Name:**	**Batch_Time**
					value:	"@string(mul(int(pipeline().parameters.Batch_Size),-1))"
3	until	Success	Loop_Until_Time_equals_Pipeline_Time	**Settings:**	**Expression:**	"@less(int(pipeline().parameters.Pipeline_Time),mul(int(variables('Batch_Time')),-1))"
3.1	Append variable	Success	Batch_Array_variable	**Variables:**	**Name:**	**Batch_Array**
					value:	"@variables('Batch_Time')"
3.2	Set variable	Success	Batch_Time_Increment_variable	**Variables:**	**Name:**	**Batch_Time_Increment**
					value:	"@string(sub(int(variables('Batch_Time')),int(pipeline().parameters.Batch_Size)))"
3.3	Set variable	Success	Set_Batch_Time_variable	**Variables:**	**Name:**	**Batch_Time**
					value:	"@variables('Batch_Time_Increment')"
4	ForEach	Success	Loop_Every_Batch	**Settings:**	**Items:**	"@variables('Batch_Array')"
4.1	Copy Data	Success	Copy_To_Destination_Blob	**General:**	**Retry:**	3
				Source:	**Source Dataset:**	**01_heartbeat_FS_json_source**
					File Path Type:	Wildcard file path
					Wildcard Path:	heartbeatlogs/@concat('heartbeat/',formatDateTime(subtractFromTime(pipeline().TriggerTime,1,'Hour'), 'yyyy/M/dd/HH'))/*.gz
					Filter by last Modified:	

						Start Time UTC :"@addminutes(variables('Time_Path'),int(item()))"
						End Time UTC :"@addminutes(variables('Time_Path'),add(int(pipeline().parameters.Batch_Size),int(item())))"
					Recursively	YES
				Sink:	**Sink Dataset:**	**01_heartbeat_blob_json_sink**
					Copy behaviour	Preserve hierarchy
				Settings:	**Data integration unit**	32
					degree of copy parallelism	48
4.2	Delete	Success	Delete_From_Source_FileShare	**Source:**	**Source Dataset :**	**01_heartbeat_FS_json_delete_source** :Open:Connection:File path:"heartbeatlogs/@concat('heartbeat/',formatDateTime(subtractFromTime(pipeline().TriggerTime,1,'Hour'), 'yyyy/M/dd/HH'))/Null"
					File Path Type:	Wildcard file path
					Wildcard File name:	*.gz
					Filter by last Modified:	
						Start Time UTC :"@addminutes(variables('Time_Path'),int(item()))"
						End Time UTC :"@addminutes(variables('Time_Path'),add(int(pipeline().parameters.Batch_Size),int(item())))"
					Recursively	YES
4.3	Execute Pipeline	**Failure**	Execute_Custom_Pipeline	**Settings:**	**invoked Pipeline**	**02_ErrorFiles_ExceptionHandling**

					wait on completion	NO
					Parameters	
						Start_Time :"@addminutes(variables('Time_Path'),int(item()))"
						End_Time:"@addminutes(variables('Time_Path'),add(int(pipeline().parameters.Batch_Size),int(item())))"

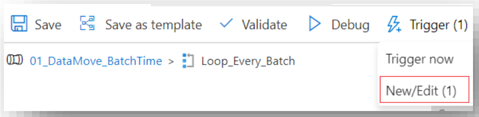
Trigger

What are triggers in AZURE DATA FACTORY?

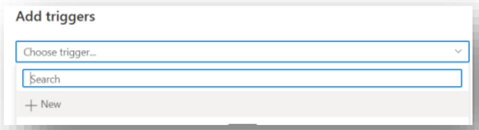
Currently, the service supports three types of triggers: **Schedule trigger: A trigger that invokes a pipeline** on a wall-clock schedule. Tumbling window trigger: A trigger that operates on a periodic interval, while also retaining state. Event-based trigger: A trigger that responds to an event

So as per our use case, we will set a Schedule trigger that will run the pipeline every one hour.

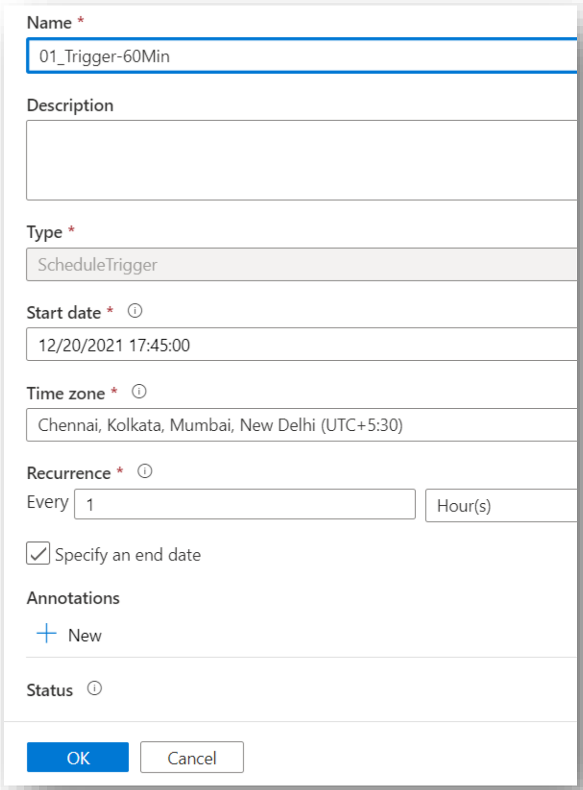
1. Open pipeline Add trigger.



2. Click on + NEW.



3. Configs same as below image. And click on OK



Name *
01_Trigger-60Min

Description

Type *
ScheduleTrigger

Start date *
12/20/2021 17:45:00

Time zone *
Chennai, Kolkata, Mumbai, New Delhi (UTC+5:30)

Recurrence *
Every 1 Hour(s)

[✓] Specify an end date

Annotations
New

Status

OK Cancel

Pipeline: 02_ErrorFiles_ExceptionHandling

Problem Statement:

When moving data from the source to the destination we also want to decompress the ".gz" Zip files to JSON.

But somehow if we get the ".gz" corrupted files in our source Then the AZURE DATA FACTORY is not able to handle those error files separately and because of that, the whole pipeline gets failed.

Solution:

So, we have used python code to check all files and get the list of names and paths for files that are invalid or corrupted.

This python code will give an output in "ErrorLogs.json" File and we will use this JSON list to move that error .gz files to the error folder without uncompressing and the rest file will move to destination Blob as JSON

Overview of Pipeline:

Note: This Pipeline will automatically run only when some error file is there.

02_ErrorFiles_ExceptionHandling: In this pipeline, we have 2 parameters 2 variables 9 activities, and No trigger. This pipeline will run only when some error file occurs in pipeline 01_DataMove_BatchTime.

Pipeline Structure

Link service:

NO	Name	Describe
1	filesharetest	This link service has been config with Azure File Share which is "heartbeatlogs" in "enprodeventhubtest" storage account
2	blobstoragetest	This link service has been config with Blob Storage Container which is "heartbeat" in "enprodeventhubtest" storage account
3	AzureBatchfreshpool_link	This link service will call Batch Account which has VM in it that will run our python code to detect the error files

Datasets:

N0	Name	Format	L-S	Activity Name: use as in
1	01_heartbeat_FS_binary_source	Binary	File Share	Copy_Batch_To_Dumps: Source
2	01__FS_adfbatchdumps_binary_sink	Binary	File Share	Copy_Batch_To_Dumps: Sink
3	02__FS_jsonerrorlogs_source	JSON	File Share	Lookup_Json_ErrorFiles: Source
4	02__FS_adfbatchdumps_binary_source	Binary	File Share	Move_Error_Files: Source
5	02__FS_error_binary_sink	Binary	File Share	Move_Error_Files: Sink
6	02_adfbatchdumps_FS_json_source	JSON	File Share	Copy_To_Destination_Blob_2: Source Delete_From_Source_AZURE DATA FACTORYDumps_FileShare:Source
7	01_heartbeat_blob_json_sink2	JSON	Blob	Copy_To_Destination_Blob_2: Sink

Parameters, Variable, Activity :Details

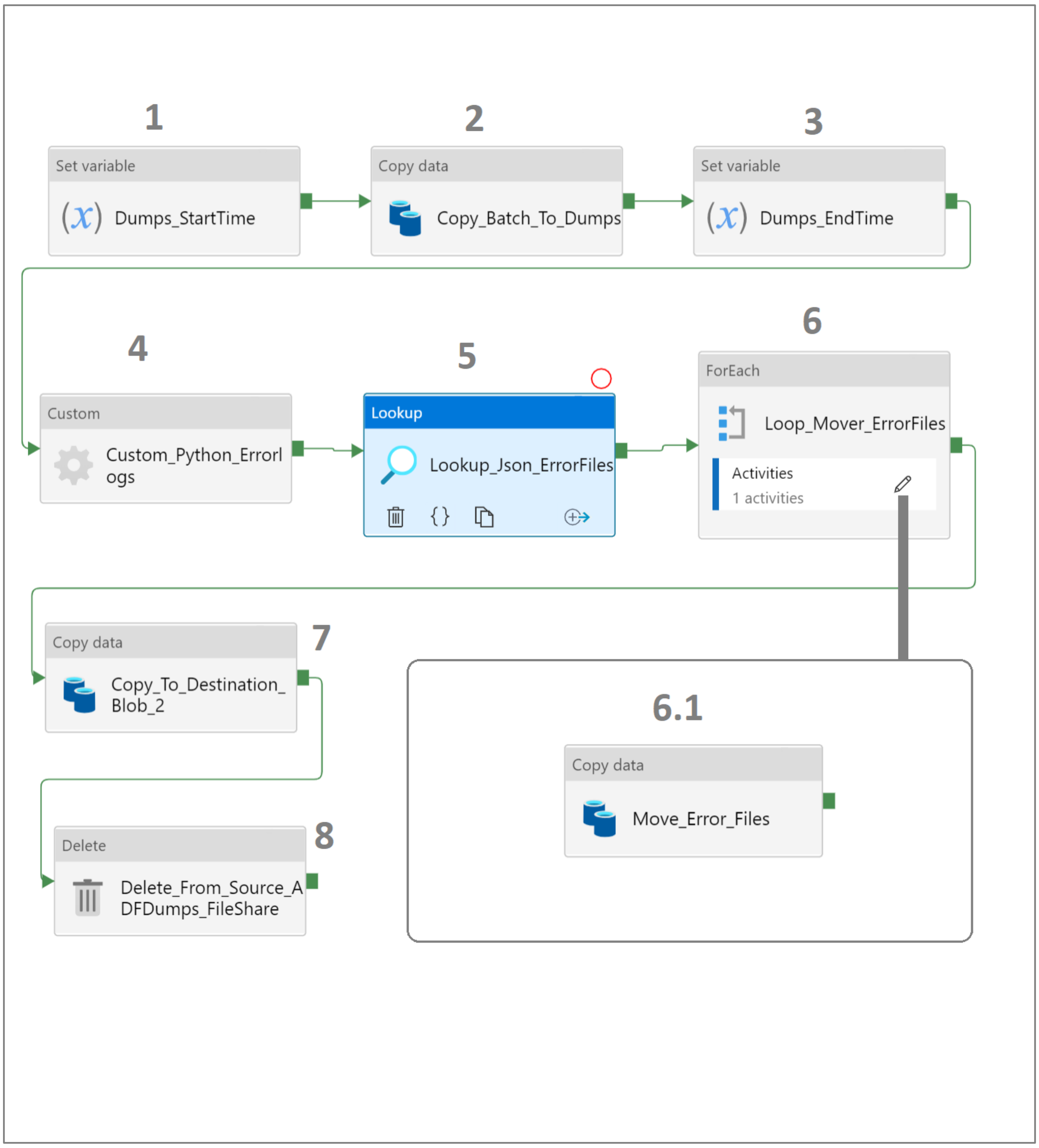
Parameters			
Name	Type	Default Value	Explanation
Start_Time	String	Null	This value will be passed by Pipeline 01 if any error occurs in the specific batch that batch **start time** value will be passed in this parameter automatically.
End_Time	String	Null	This value will be passed by Pipeline 01 if any error occurs in the specific batch that batch **End time** value will be passed in this parameter automatically.

Variables			
Name	Type	Default Value	Explanation
D_StartTime	String	Null	This variable will load the current UTC at the time of this activity run and we will pass this to the below 2 activities in start time. [Copy_To_Destination_Blob_2, Delete_From_Source_AZURE DATA FACTORYDumps_FileShare]
D_EndTime	String	Null	This variable will load the current UTC at the time of this activity run and we will pass this to the below 2 activities in End time. [Copy_To_Destination_Blob_2, Delete_From_Source_AZURE DATA FACTORYDumps_FileShare]

No	**Activities**	**Name**	**Working**
1	Set variable	Dumps_StartTime	This will load Current UTC Time
2	Copy Data	Copy_Batch_To_Dumps	This activity will move data from **heartbeat** folder to **adfbatchdumps** folder in .gz without decompressing
3	Set variable	Dumps_EndTime	This will load Current UTC Time
4	Custom	Custom_Python_Errorlogs	This activity will call **Batch Account** with will run our **main.py** python code from blob storage **Script** container
5	Lookup	Lookup_Json_ErrorFiles	This will load ErrorLogs.json Files from the **jsonerrorlogs** folder in Files share this **Json** File will be generated by the above activity
6	ForEach	Loop_Mover_ErrorFiles	This will loop and move one by one file witch files name is present in ErrorLogs.json
6.1	Copy data	Move_Error_Files :	This will move error file in the **error** folder in **.gz** format without decompressing
7	Copy Data	Copy_To_Destination_Blob_2	This will move data from file share to Blob in JSON format

8	Delete	Delete_From_Source_ADFDumps_FileShare	This activity will Delete the data from the File share

Activity, Dependency:



No	Activities	Dependency	Name	Configs	Sub-Configs	Syntax
1	Set variable		Dumps_StartTime	**Variables:**	**Name:**	**D_StartTime**
					value:	"@utcnow()"
2	Copy Data	Success	**Copy_Batch_To_Dumps**	**General:**	**Retry:**	3
				Source:	**Source Dataset:**	01_heartbeat_FS_binary_source
					File Path Type :	Wildcard file path
					Wildcard Path:	heartbeatlogs/heartbeat/*
					Filter By Last Modified:	
						Start Time UTC:"@pipeline().parameters.Start_Time"
						End Time UTC:"@pipeline().parameters.End_Time"
					Recursively:	YES
					Delete files after completion:	YES
				Sink:	**Sink Dataset:**	01__FS_adfbatchdumps_binary_sink
					Copy Behaviour	Preserve hierarchy
				Settings:	**Data Integration Unit**	32
					Degree of Copy Parallelism	48
3	Set variable	Success	**Dumps_EndTime**	**Variables:**	**Name:**	**D_EndTime**
					value:	"@utcnow()"
4	Custom	Success	**Custom_Python_Errorlogs**	**General:**	**Retry:**	3
				Azure Batch:	**Azure batch linked services:**	AzureBatchfreshpool_link
				Settings:	**Command:**	python3 main.py
					Resource linked Service	blobstoragetest
					Folder Path:	script
5	Lookup	Success	**Lookup_Json_ErrorFiles**	**Settings:**	**Source Dataset:**	**02_FS_jsonerrorlogs_source :Open:Connection:File Path:**"heartbeatlogs/jsonerrorlogs/ErrorLogs.json"

					File Path Type:	File-path in dataset
					Recursively:	YES
6	ForEach	Success	Loop_Mover_ErrorFiles	**Settings:**	**items:**	"@activity('Lookup_Json_ErrorFiles').output.value"
	Note:6.1 is a sub-activity of the 6th activity					
6.1	Copy data		**Move_Error_Files:**	**General:**	**Retry:**	3
				Source:	**Source Dataset:**	02__FS_adfbatchdumps_binary_source
					File Path Type:	Wildcard file path
					Wildcard Path:	heartbeatlogs/@item().file_path/@item().file_name
					Recursively:	YES
					Delete files after completion	YES
				Sink:	**Sink Dataset:**	02__FS_error_binary_sink
					Copy Behavior	Preserve hierarchy
				Settings:	**Data integration unit**	32
					degree of copy parallelism	48
7	Copy Data	Success	Copy_To_Destination_Blob_2	**General:**	**Retry:**	3
				Source:	**Source Dataset:**	02__FS_adfbatchdumps_json_source
					File Path Type:	Wildcard file path
					Wildcard Path:	heartbeatlogs/adfbatchdumps/*
					Filter By Last Modified:	
						Start Time UTC :"@variables('D_StartTime')"
						End Time UTC:"@variables('D_EndTime')"
					Recursively:	YES
				Sink:	**Sink Dataset:**	01_heartbeat_blob_json_sink2
					Copy behavior	Preserve hierarchy

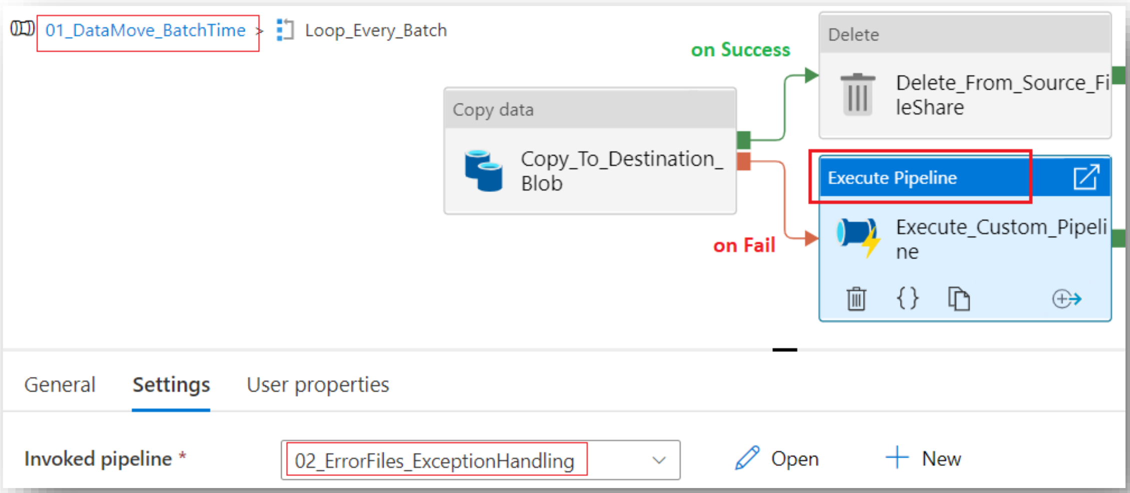
				Settings:	**Data Integration Unit**	32
					Degree of Copy Parallelism	48
8	Delete	Success	Delete_From_Source_AZURE DATA FACTORYDumps_FileShare	**General:**	**Retry:**	3
				Source:	**Source Dataset:**	02__FS_adfbatchdumps_json_source :Open:Connection:FilePath:"heartbeatlogs/adfbatchdumps/Null
					File Path Type:	Wildcard file path
					Wildcard Filename:	*.gz
					Filter by last Modified:	
						Start Time UTC :"@variables('D_StartTime')"
						End Time UTC:"@variables('D_EndTime')"
					Recursively:	YES

Trigger

What are triggers in AZURE DATA FACTORY?

Currently, the service supports three types of triggers: **Schedule trigger: A trigger that invokes a pipeline** on a wall-clock schedule. Tumbling window trigger: A trigger that operates on a periodic interval, while also retaining state. Event-based trigger: A trigger that responds to an event

So, for this Pipeline, we will not set a trigger manually because we have configured this pipeline in the "Execute pipeline" activity in case of failure in pipeline "01_DataMove_BatchTime" it will work as an Event-Based trigger.



Invoked pipeline * 02_ErrorFiles_ExceptionHandling

Pipeline: 03_FallBack_DataMove_BatchTime

Problem Statement:

In case of ADF Pipeline failure, the past data will not Move from source to destination because our main pipeline is built for 1hr Data move based on past 1hr UTC only for that we need to build something which will make sure nothing is left behind.

We need something which will double-check that no data is not left for the last 3hr.

Solution:

So, for the above problem, we have built a pipeline almost the same as "01_DataMove_BatchTime" pipeline with few changes in it we will shift the last 1hr for example if we run the pipeline at 1:00 pm the data will move for [1:00 pm Skip this 1hr 12:00 pm 2hr 11:00 am 3hr 10:00 am 4hr 9:00 am]

Skip the 1st 1hr because the main pipeline will be running at that time, so we don't want to clash 2 pipelines for the same Time therefore we are skipping 1hr for this pipeline.

And we will set the trigger to run this pipeline for the last 4 hr and 1st hr skip as explained above with an example.

Overview of Pipeline:

Note: This is our Fall-back pipeline over the main pipeline.

03_FallBack_DataMove_BatchTime: in this pipeline, we have 3 parameters 3 variables 9 activities, and 1 trigger. This pipeline will run every 1 hour every day with the help of Trigger.

Pipeline Structure

Link service:

NO	Name	Describe
1	filesharetest	This link service has been config with Azure File Share which is "heartbeatlogs" in "enprodeventhubtest" storage account
2	blobstoragetest	This link service has been config with Blob Storage Container which is "heartbeat" in "enprodeventhubtest" storage account

Datasets:

N0	Name	Format	L-S	Activity Name: use as in
1	01_heartbeat_blob_json_sink2	JSON	Blob	Copy_To_Destination_Blob: Sink
2	01_heartbeat_FS_json_delete_source2	JSON	File Share	Delete_From_Source_FileShare: Source
3	01_heartbeat_FS_json_source	JSON	File Share	Copy_To_Destination_Blob: Source

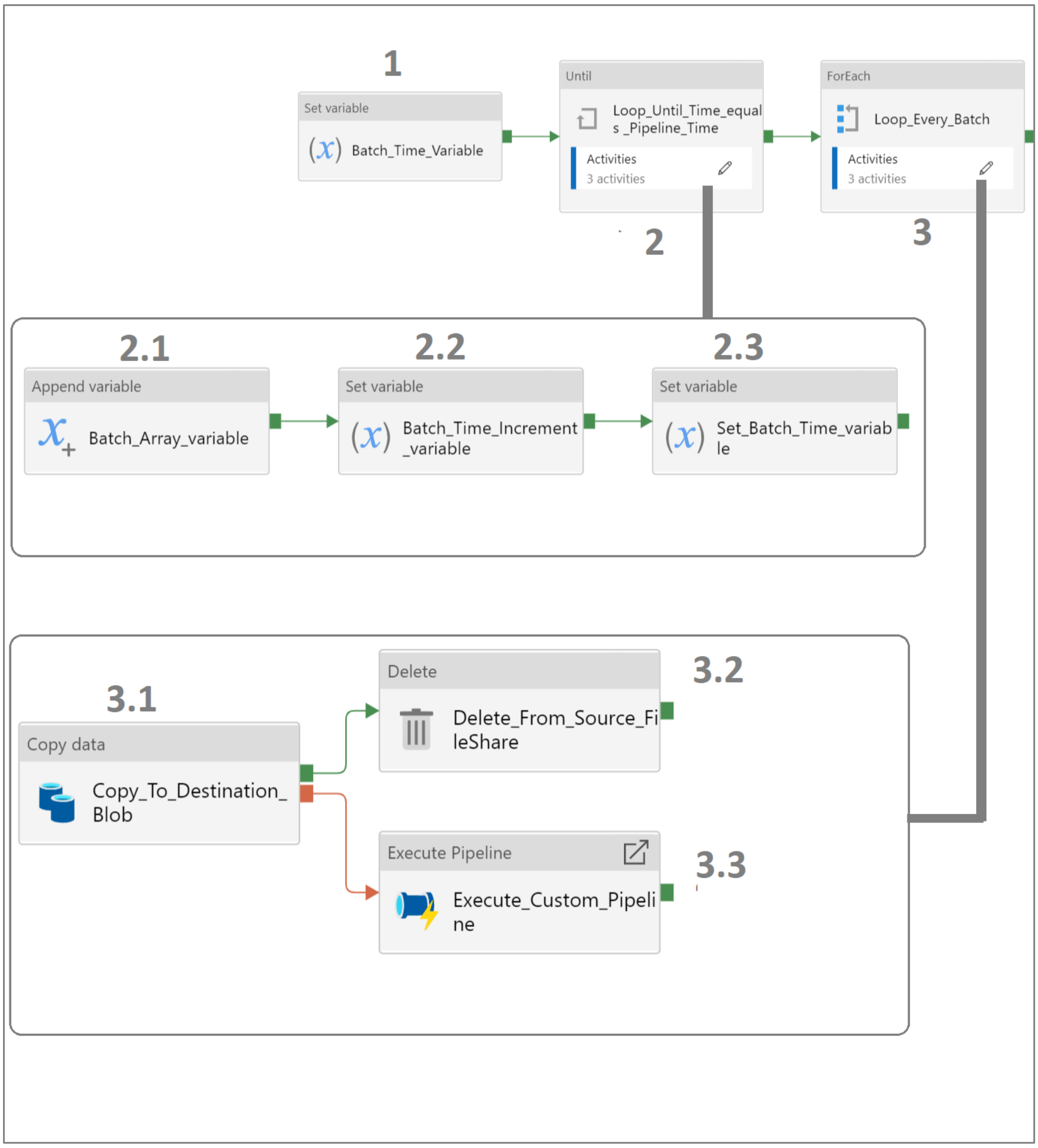
Parameters, Variable:

Parameters			
Name	Type	Default Value	Explanation
Pipeline_Time	String	360	This value will decide the duration of the data move if we pass 360 that means for last 360 min of data will be moved from source to destination
Batch_Size	String	15	This value will divide the above 60 min value to 15min each that is 4-time 15min [0-15],[15-30],[30-45],[45-60]
Duration_Shift_Time	String	60	This parameter will shift End Time 60 min back

Variables			
Name	Type	Default value	Explanation
Batch_Array	Array	Null	This will load an array of times like for time range 1:00 pm to 2:00 pm this would be [1,1:15] [1:15,1:30][1:30:1:45][1:45,2:00] pass this array in sequence in a loop
Batch_Time	String	Null	This variable converts the Batch_Size value to negative and store that value.
Batch_Time_Increment	String	Null	This variable Increate the time by 15 min in each loop.

No	Activities	Name	Working
1	Set variable	Batch_Time_Variable	This will load the batch time from parameter
2	until	Loop_Until_Time_equals _Pipeline_Time	This Will loop until the give time meet condition
2.1	Append variable	Batch_Array_variable	This will load Batch Time 15 min
2.2	Set variable	Batch_Time_Increment_variable	This will increase Batch Time 15*1 ,15*2, and so on in this 1,2 is count of loop
2.3	Set variable	Set_Batch_Time_variable	This will load the previous Bath time 1 loop 15, 2nd loop 30 ,3 loop 45, 4th loop 60 .so on
3	ForEach	Loop_Every_Batch	The until loop output Of Array time will be passed in this loop array will be [0:15][15:30][30:45][45:60].so on for the give time
3.1	Copy Data	Copy_To_Destination_Blob	This activity will Copy data from File share to blob
3.2	Delete	Delete_From_Source_FileShare	This activity will delete the data from file share
3.3	Execute Pipeline	Execute_Custom_Pipeline	This activity will call 02 pipeline in case of any error occurs.

Activity, Dependency



No	Activities	Dependency	Name	Configs	Sub-Configs	Syntax
1	Set variable	Success	Batch_Time_Variable	**Variables:**	**Name:**	**Batch_Time**
					value:	"@string(mul(int(pipeline().parameters.Batch_Size),-1))"
2	until	Success	Loop_Until_Time_equals_Pipeline_Time	**Settings:**	**Expression:**	"@less(int(pipeline().parameters.Pipeline_Time),mul(int(variables('Batch_Time')),-1))"
2.1	Append variable	Success	Batch_Array_variable	**Variables:**	**Name:**	**Batch_Array**
					value:	"@variables('Batch_Time')"
2.2	Set variable	Success	Batch_Time_Increment_variable	**Variables:**	**Name:**	**Batch_Time_Increment**
					value:	"@string(sub(int(variables('Batch_Time')),int(pipeline().parameters.Batch_Size)))"
2.3	Set variable	Success	Set_Batch_Time_variable	**Variables:**	**Name:**	**Batch_Time**
					value:	"@variables('Batch_Time_Increment')"
3	ForEach	Success	Loop_Every_Batch	**Settings:**	**Items:**	"@variables('Batch_Array')"
3.1	Copy Data	Success	Copy_To_Destination_Blob	**General:**	**Retry:**	3
				Source:	**Source Dataset:**	**01_heartbeat_FS_json_source**
					File Path Type:	Wildcard file path
					Wildcard Path:	heartbeatlogs/heartbeat/*
					Filter by last Modified:	
						Start Time UTC :"@addminutes(addminutes(pipeline().TriggerTime,mul(int(pipeline().parameters.Duration_Shift_Time),-1)),int(item()))"

						End Time UTC :"@addminutes(addminutes(pipeline().TriggerTime,mul(int(pipeline().parameters.Duration_Shift_Time),-1)),add(int(pipeline().parameters.Batch_Size),int(item())))"
					Recursively :	YES
				Sink:	**Sink Dataset:**	**01_heartbeat_blob_json_sink2:Open:Connection: File Path:** “heartbeat/Null/Null"
					Copy behaviour	Preserve hierarchy
				Settings:	**Data integration unit**	32
					degree of copy parallelism	48
3.2	Delete	Success	Delete_From_Source_FileShare	**General:**	**Retry:**	3
				Source:	**Source Dataset:**	**01_heartbeat_FS_json_delete_source2: Open:Connection:FilePath:**"heartbeatlogs/heartbeat/Null"
					File Path Type:	Wildcard file path
					Wildcard File name:	*.gz
					Filter by last Modified:	
						Start Time UTC :"@addminutes(addminutes(pipeline().TriggerTime,mul(int(pipeline().parameters.Duration_Shift_Time),-1)),int(item()))"
						End Time UTC :"@addminutes(addminutes(pipeline().TriggerTime,mul(int(pipeline().parameters.Duration_Shift_Time),-1)),add(int(pipeline().parameters.Batch_Size),int(item())))"
					Recursively :	YES

3.3	Execute Pipeline	**Failure**	Execute_Custom_Pipeline	**Settings:**	**invoked Pipeline**	**02_ErrorFiles_ExceptionHandling**
					wait on completion	NO
					Parameters	
						Start_Time :"@addminutes(addminutes(pipeline().TriggerTime,mul(int(pipeline().parameters.Duration_Shift_Time),-1)),int(item()))"
						End_Time :"@addminutes(addminutes(pipeline().TriggerTime,mul(int(pipeline().parameters.Duration_Shift_Time),-1)),add(int(pipeline().parameters.Batch_Size),int(item())))"

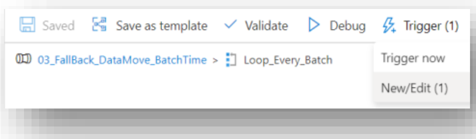
Trigger

What are triggers in AZURE DATA FACTORY?

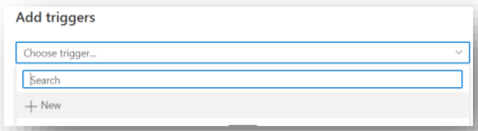
Currently, the service supports three types of triggers: **Schedule trigger: A trigger that invokes a pipeline** on a wall-clock schedule. Tumbling window trigger: A trigger that operates on a periodic interval, while also retaining state. Event-based trigger: A trigger that responds to an event

So as per our use case, we will set a Schedule trigger that will run the pipeline every 3 hours.

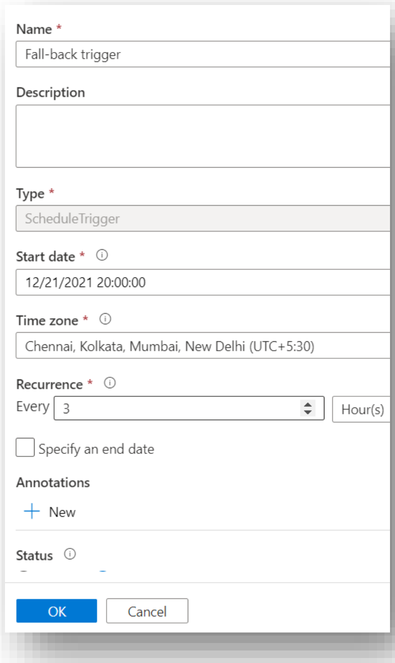
- Open pipeline Add trigger.



- Click on + NEW.



3. Configs same as below image. And click on OK


Name *
Fall-back trigger

Description

Type *
ScheduleTrigger

Start date *
12/21/2021 20:00:00

Time zone *
Chennai, Kolkata, Mumbai, New Delhi (UTC+5:30)

Recurrence *
Every 3 Hour(s)

Specify an end date

Annotations
New

Status

OK Cancel

Pipeline: 04_DataMove_Custom_DateTime_Batch

Problem Statement:

In case of AZURE DATA FACTORY Pipeline failure, for a specific time for example like yesterday, the pipeline fails and because of that some data of 2,3 Hours left behind then the data folder for 11:00, 21:00,9:00 UTC will be there in File Share then for this folder we need to create some custom time pipeline so that we are not losing any data.

Solution:

So, for the above problem, we have built a pipeline This pipeline is almost the same as Pipeline 01, but the only difference is instead of static trigger time we must change the time to dynamic so the pipeline can work with different past times.

Overview of Pipeline:

Note: This is our Custom time base pipeline in case of any failure occurs and the pipeline is not able to move data for the specific UTC folder.

04_DataMove_Custom_DateTime_Batch: in this pipeline, we have 3 parameters 4 variables 10 activities, no trigger for this pipeline needs to run on manually debug.

Pipeline Structure

Link service:

NO	Name	Describe
1	filesharetest	This link service has been config with Azure File Share which is "heartbeatlogs" in "enprodeventhubtest" storage account
2	blobstoragetest	This link service has been config with Blob Storage Container which is "heartbeat" in "enprodeventhubtest" storage account

Datasets:

N0	Name	Format	L-S	Activity Name: use as in
1	01_heartbeat_blob_json_sink2	JSON	Blob	Copy_To_Destination_Blob: Sink
2	01_heartbeat_FS_json_delete_source2	JSON	File Share	Delete_From_Source_FileShare: Source
3	01_heartbeat_FS_json_source	JSON	File Share	Copy_To_Destination_Blob: Source

Parameters, Variable, Activity :Details

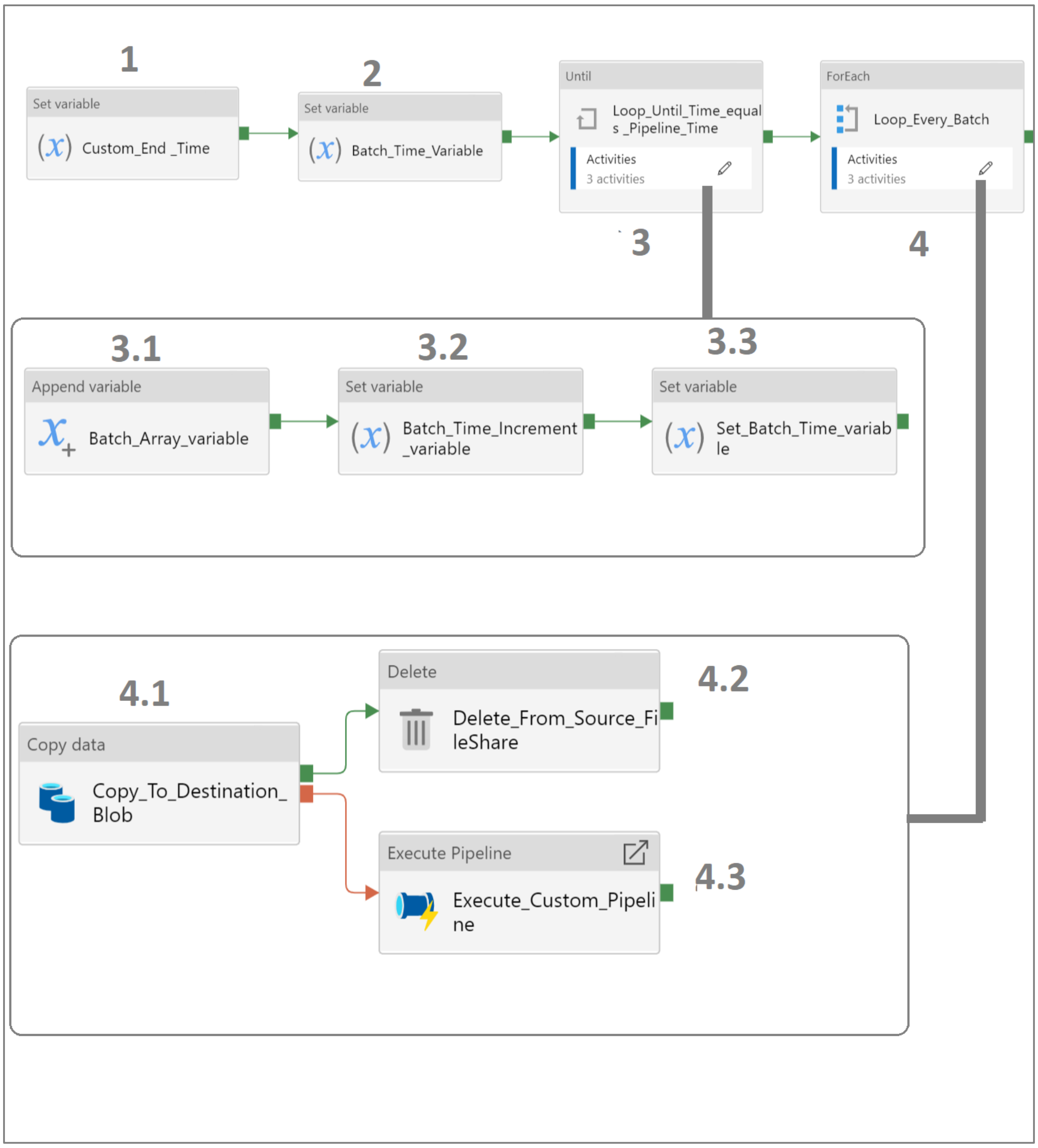
Parameters			
Name	Type	Default Value	Explanation
Pipeline_Time	String	360	This value will decide the duration of the data move if we pass 360 that means for last 360 min of data will be moved from source to destination
Batch_Size	String	15	This value will divide the above 60 min value to 15min each that is 4-time 15min [0-15],[15-30],[30-45],[45-60]
End_Time	String	Null	The pipeline will work based on the given date-time from this parameter "2021-12-22T07:10:00" The Date Time Structure should be the same as given above "yyyy-mm-ddTHH:MM:SS"

Variables			
Name	Type	Default value	Explanation
Batch_Array	Array	Null	This will load an array of times like for time range 1:00 pm to 2:00 pm this would be [1,1:15] [1:15,1:30] [1:30:1:45][1:45,2:00] pass this array in sequence in a loop
Batch_Time	String	Null	This variable converts the Batch_Size value to negative and store that value.
Batch_Time_Increment	String	Null	This variable Increate the time by 15 min in each loop.
Pipeline_End_Time	String	Null	This variable will hold the value which will come from END_Time Parameter and covert that string value to Date Time value

activities	**Activities**	**Name**	**Working**
1	Set variable	Custom_End_Time	This will load the End_Time from the parameter
2	Set variable	Batch_Time_Variable	This will load the batch time from the parameter
3	until	Loop_Until_Time_equals _Pipeline_Time	This until activity will loop till the given time meet the condition
3.1	Append variable	Batch_Array_variable	This will load Batch Time 15 min
3.2	Set variable	Batch_Time_Increment_variable	This will increase Batch Time 15*1,15*2, and so on in this 1,2 is the count of the loop
3.3	Set variable	Set_Batch_Time_variable	This will load the previous Bath time 1 loop 15, 2nd loop 30,3 loops 45, 4th loop 60 ...so on

4	ForEach	Loop_Every_Batch	The until loop output Of Array time will be passed in this loop array will be [0:15][15:30][30:45][45:60]...so on for the given time
4.1	Copy Data	Copy_To_Destination_Blob	This activity will Copy data from File share to blob
4.2	Delete	Delete_From_Source_FileShare	This activity will delete the data from a file share
4.3	Execute Pipeline	Execute_Custom_Pipeline	This activity will call 02 pipelines in case of any error occurs.

Activity, Dependency



No	Activities	Dependency	Name	Configs	Sub-Configs	Syntax
1	Set variable	Success	Custom_End _Time	**Variables:**	**Name:**	**Pipeline_End_Time**
					value:	"@formatDateTime(pipeline().parameters.End_Time)"
2	Set variable	Success	Batch_Time_Variable	**Variables:**	**Name:**	**Batch_Time**
					value:	"@string(mul(int(pipeline().parameters.Batch_Size),-1))"
3	until	Success	Loop_Until_Time_equals _Pipeline_Time	**Settings:**	**Expression:**	"@less(int(pipeline().parameters.Pipeline_Time),mul(int(variables('Batch_Time')),-1))"
3.1	Append variable	Success	Batch_Array_variable	**Variables:**	**Name:**	**Batch_Array**
					value:	"@variables('Batch_Time')"
3.2	Set variable	Success	Batch_Time_Increment_variable	**Variables:**	**Name:**	**Batch_Time_Increment**
					value:	"@string(sub(int(variables('Batch_Time')),int(pipeline().parameters.Batch_Size)))"
3.3	Set variable	Success	Set_Batch_Time_variable	**Variables:**	**Name:**	**Batch_Time**
					value:	"@variables('Batch_Time_Increment')"
4	ForEach	Success	Loop_Every_Batch	**Settings:**	**Items:**	"@variables('Batch_Array')"
4.1	Copy Data	Success	Copy_To_Destination_Blob	**General:**	**Retry:**	3
				Source:	**Source Dataset:**	**01_heartbeat_FS_json_source**
					File Path Type:	Wildcard file path
					Wildcard Path:	heartbeatlogs/heartbeat/*

					Filter by last Modified:	
						Start Time UTC :"@addminutes(variables('Pipeline_End_Time'),int(item()))"
						End Time UTC :"@addminutes(variables('Pipeline_End_Time'),add(int(pipeline().parameters.Batch_Size),int(item())))"
					Recursively:	YES
				Sink:	**Sink Dataset :**	**01_heartbeat_blob_json_sink2:Open:Connection:FilePath:**"heartbeat/Null/Null"
					Copy behaviour	Preserve hierarchy
				Settings:	**Data integration unit**	32
					degree of copy parallelism	48
4.2	Delete	🟩 Success	Delete_From_Source_FileShare	**General:**	**Retry:**	3
				Source:	**Source Dataset:**	**01_heartbeat_FS_json_delete_source2:Open:Connection:FilePath:**"heartbeatlogs/heartbeat/Null"
					File Path Type:	Wildcard file path
					Wildcard File name:	*.gz
					Filter by last Modified:	
						Start Time UTC :"@addminutes(variables('Pipeline_End_Time'),int(item()))"
						End Time UTC :"@addminutes(variables('Pipeline_End_Time'),add(int(pipeline().parameters.Batch_Size),int(item())))"
					Recursively:	YES

4.3	Execute Pipeline	**Failure**	Execute_Custom_Pipeline	**Settings:**	**invoked Pipeline**	**02_ErrorFiles_ExceptionHandling**
					wait on completion	NO
					Parameters	
						Start_Time :"@addminutes(variables('Pipeline_End_Time'),int(item()))"
						End_ Time:"@addminutes(variables('Pipeline_End_Time'),add(int(pipeline().parameters.Batch_Size),int(item())))"

Trigger/Debug

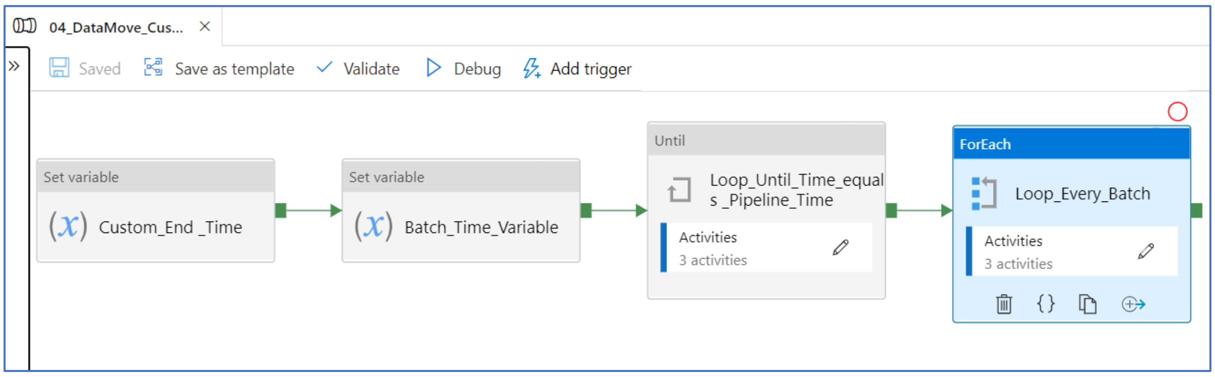
What is debug mode in AZURE DATA FACTORY?

Azure Data Factory and Synapse Analytics mapping data flow's debug mode **allows you to interactively watch the data shape transform while you build and debug your data flows**. ... When Debug mode is on, you'll interactively build your data flow with an active Spark cluster. The session will close once you turn debug off.

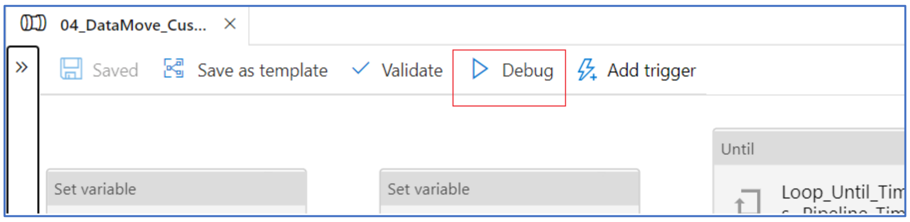
So as per our use case, we will not set any type of trigger as we only want to run this pipeline when we have so we must manually debug this

Debug Steps:

- open pipeline **04_DataMove_Custom_DateTime_Batch from** AZURE DATA FACTORY.

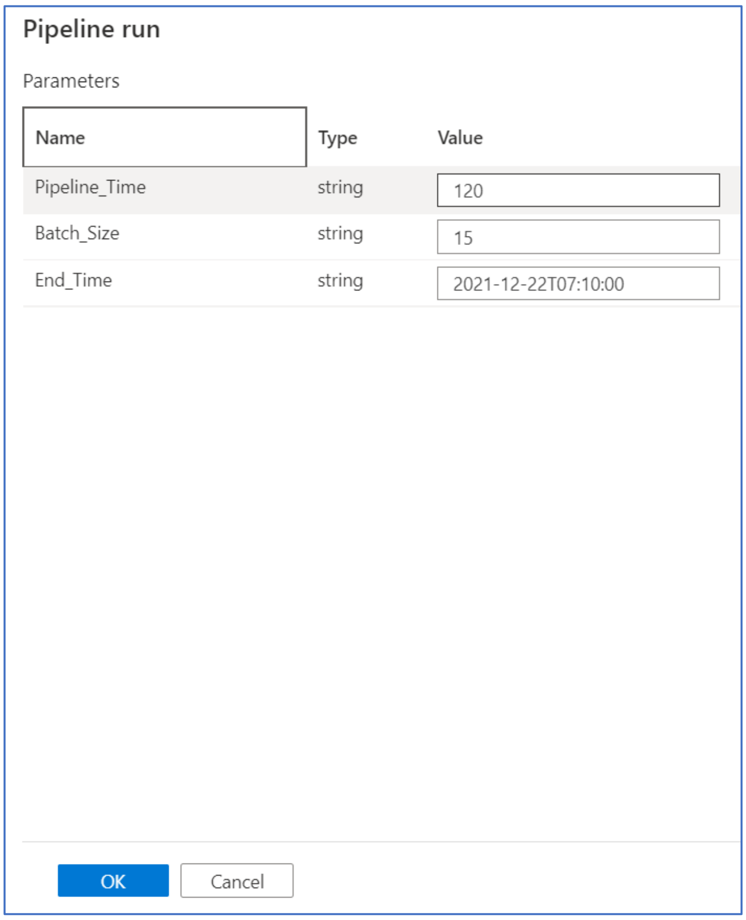


- Click on Debug button.



- This wizard will appear it will ask you 3 parameter values.

Pipeline_Time	In the below screenshot, it's 120 that means 120 min which is 2hr
Batch_Size	These 15 means 15 min and it will divide the **Pipeline_Time** that is 120 min into 15min small batch so 120/15 = 8 so the pipeline will have 8 batches to run.
End_Time	This is our main parameter for this pipeline you need to pass the end time if you are passing "2021-12-22T07:00:00" then the pipeline will copy data from Explanation of logic: **End_Time - Pipeline_Time = will give us Start_Time** 2021-12-22T07:00:00 – 120 min = 2021-12-22T05:00:00 So now the pipeline will run for 5 to 7 UTC now whatever data are left there for this time range will get moved to Blob.

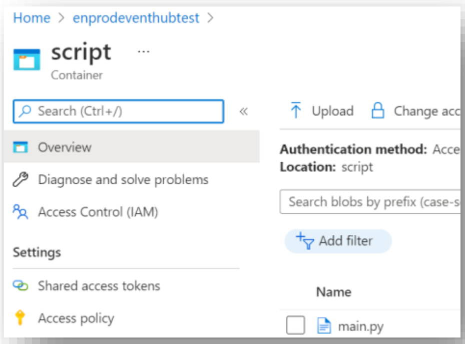

Pipeline run

Parameters

Name	Type	Value
Pipeline_Time	string	120
Batch_Size	string	15
End_Time	string	2021-12-22T07:10:00

Python Code

The python code file is main.py which is in the **Script** container of the **Azure Storage Account** under Blob


Home > enprodeventhubtest >

script
Container

Search (Ctrl+/)
Overview
Diagnose and solve problems
Access Control (IAM)
Settings
Shared access tokens
Access policy

Upload
Change acc

Authentication method: Acce
Location: script

Search blobs by prefix (case-s
Add filter

Name
main.py

Main.py

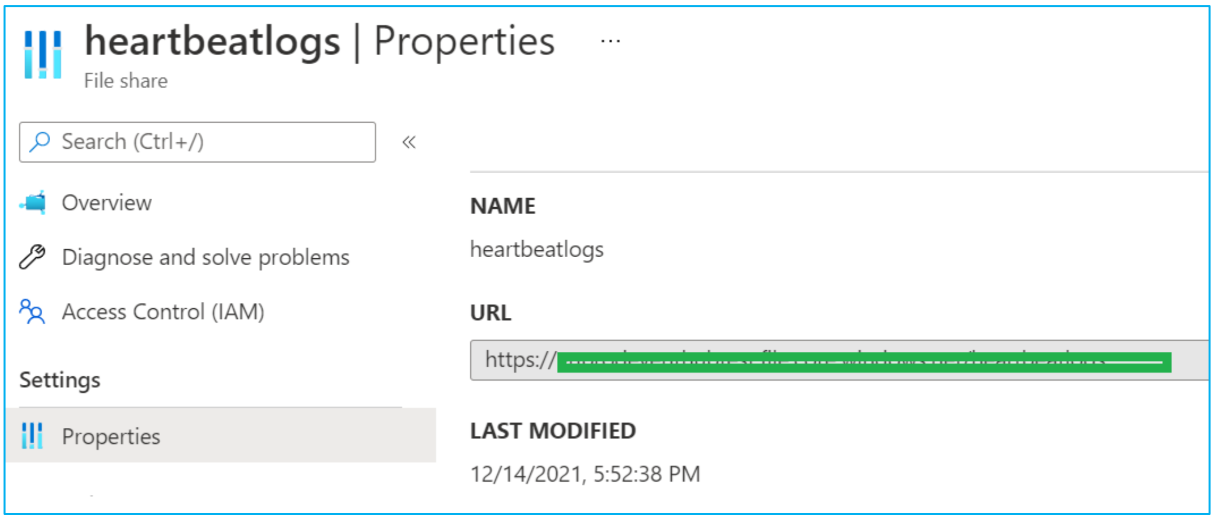
```
from azure.core.exceptions import (
    ResourceExistsError,
    ResourceNotFoundError
)

from azure.storage.fileshare import (
    ShareServiceClient,
    ShareClient,
    ShareDirectoryClient,
    ShareFileClient
)
from datetime import date
import urllib.request
import gzip
import json

# Write to azure file share
CONNECTION_STRING = ""

# URL & KEY for reading from Azure file share
URL = ""
KEY = ""

# Json file name where logs will be stored in azure file share
ERROR_LOG_FILE_NAME = "ErrorLogs.json"

rows = []

# Parent directory name, by default taken as current year
SHARE = "heartbeatlogs"
DIRECTORY_READ = "adfbatchdumps"
```

```
DIRECTORY_WRITE = "jsonerrorlogs"

def list_recursive(current_directory,directory_client,directory_name):

    sub_client = directory_client.get_subdirectory_client(directory_name)
    myfiles = sub_client.list_directories_and_files()

    for file in myfiles:
        if file.get('is_directory'):
            list_recursive(current_directory+"/"+file.get('name'),sub_client,file.get('name'))
        else:
            url = URL+current_directory+"/"+file.get('name')+KEY
            with urllib.request.urlopen(url) as response:
                with gzip.GzipFile(fileobj=response) as uncompressed:
                    try:
                        file_header = uncompressed.read()
                        print(file.get('name')+"- working")
                    except:
                        print(file.get('name')+"- corrupt")
                        # rows.append([file.get('name'),"Corrupt"])
                        rows.append({"file_path":current_directory, "file_name":file.get('name')})

if __name__ == '__main__':
    conn_str = CONNECTION_STRING
    file_service = ShareServiceClient.from_connection_string(conn_str)
    share_client = file_service.get_share_client(SHARE)
    d_client = share_client.get_directory_client(DIRECTORY_READ)
    myfiles = d_client.list_directories_and_files()

    for file in myfiles:
        if file.get('is_directory'):
            list_recursive(DIRECTORY_READ+"/"+file.get('name'),d_client,file.get('name'))

    json_object = json.dumps(rows, indent = 4)

    share_client_error = file_service.get_share_client(SHARE)
    d_client_error = share_client_error.get_directory_client(DIRECTORY_WRITE)
    d_client_error.upload_file(file_name=ERROR_LOG_FILE_NAME, data=json_object)
```

In the above code, I have removed the keys for safety reasons. You can get these keys from Azure Storage Account

```
# Write to Azure file share
CONNECTION_STRING = ""

# URL & KEY for reading from Azure file share
URL = ""
KEY = ""
```

- **URL: Azure File Share: Properties**



heartbeatlogs | Properties ···

File share

Search (Ctrl+/)

Overview

Diagnose and solve problems

Access Control (IAM)

Settings

Properties

NAME

heartbeatlogs

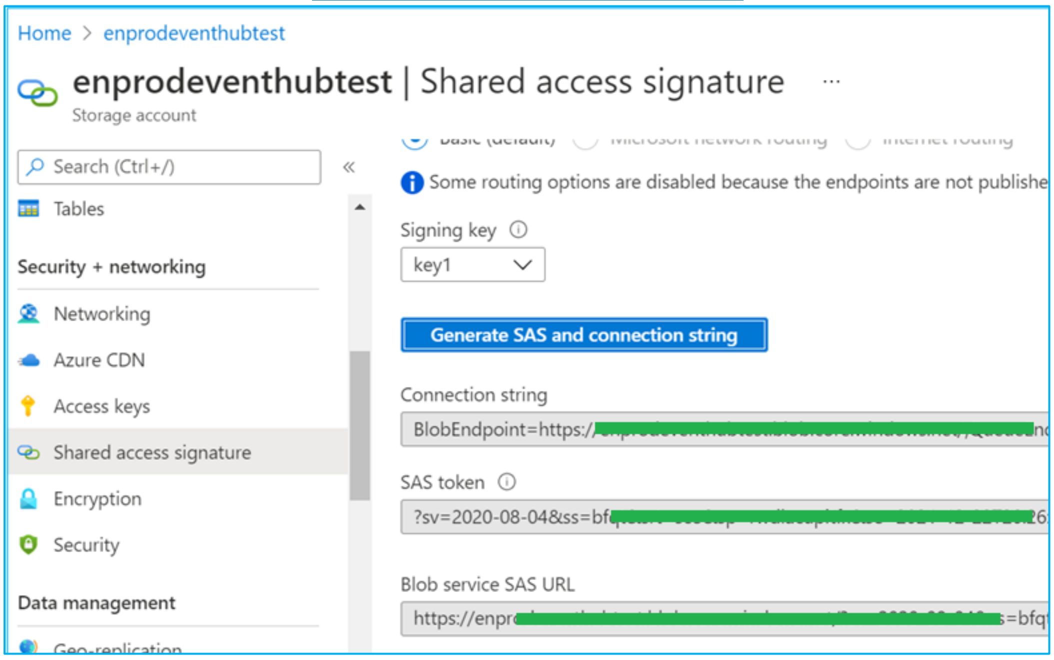
URL

https://[redacted]

LAST MODIFIED

12/14/2021, 5:52:38 PM

- KEY & CONNECTION_STRING you will get this from the azure Storage account In Shared access Signature: Click on Generate SAS and Connection string



Home > enprodeventhubtest

enprodeventhubtest | Shared access signature ···

Storage account

Search (Ctrl+/)

Tables

Security + networking

Networking

Azure CDN

Access keys

Shared access signature

Encryption

Security

Data management

Geo-replication

Some routing options are disabled because the endpoints are not publishe

Signing key

key1

Generate SAS and connection string

Connection string

BlobEndpoint=https:/[redacted]

SAS token

?sv=2020-08-04&ss=bf[redacted]

Blob service SAS URL

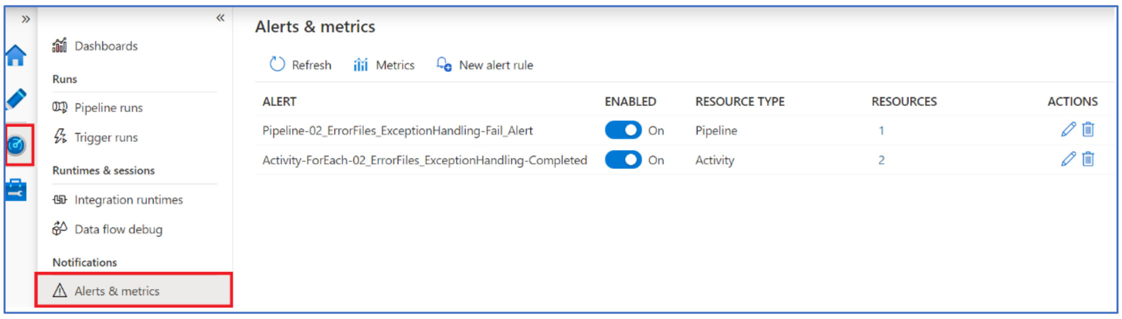
https://enpr[redacted]=bfq

⚠ Alerts & Metrics

With Azure Monitor, you can gain visibility into the performance and health of your Azure workloads. The most important type of Monitor data is the metric, which is also called the performance counter. Metrics are emitted by most Azure resources. Monitor provides several ways to configure and consume these metrics for monitoring and troubleshooting.

ⓘ Note
Except for *PipelineElapsedTimeRuns*, only events from completed, triggered activity and pipeline runs are emitted. In-progress and debug runs are *not* emitted.

We have set 2 alerts using Alerts & Metrics



ALERT	ENABLED	RESOURCE TYPE	RESOURCES	ACTIONS
Pipeline-02_ErrorFiles_ExceptionHandling-Fail_Alert	On	Pipeline	1	
Activity-ForEach-02_ErrorFiles_ExceptionHandling-Completed	On	Activity	2	

Alert	Action
Pipeline-02_ErrorFiles_ExceptionHandling-Fail_Alert	This Alert will send an Email to the action group if the Pipeline Fails
Activity-ForEach-02_ErrorFiles_ExceptionHandling-Completed	This Alert will send an Email to the action group once the "ForEach: Loop_Mover_ErrorFiles" Activity is completed successfully

Point's to be Noted

- Don't use any syntax with inverted commas in ADF just to show the start and end of the "Syntax" I have used " " in this document.
- You also need to take care of Key expiry for the Azure Storage Account as well as the Azure Batch Account.
- Do not upload any files in **adfbatchdumps** folder in **heartbeatlogs** Azure Files Share which is in **enprodeventhubtest** Azure Storage account
- When an Error Occurs, you must act on those files manually this file will be moved-in **error** folder in **heartbeatlogs** Azure Files Share which is in **enprodeventhubtest** Azure Storage Account.
- Do not pass any parameter value to 02_ErrorFiles_ExceptionHandling. Pipeline

- Be cautious while setting up the trigger for the pipelines 01_DataMove_BatchTime.& 03_FallBack_DataMove_BatchTime as they need to be triggered at the same time for the proper functioning of the same