



Self-hosted Integrated Runtime in ADF

In Azure Data Factory (ADF), the **Self-Hosted Integration Runtime (SHIR)** is a runtime infrastructure that allows ADF to interact with data sources that are located on-premises or in a private network. This runtime is especially useful when you need to securely transfer data between on-premises systems, private networks, and cloud services, or when certain data sources are not accessible directly via Azure's native connectors.

Key Features of Self-Hosted Integration Runtime:

1. Data Movement Across Boundaries:

- Enables data movement between on-premises data stores and cloud-based data stores.
- Facilitates integration with systems that are behind firewalls or have network restrictions.

2. Custom Activity Execution:

- Supports the execution of custom activities, such as running your own Python, PowerShell, or .NET scripts.

3. Secure Connectivity:

- Uses a secure outbound connection to communicate with Azure, meaning no inbound ports need to be opened in your firewall.

4. Scalability:

- You can scale out the runtime by installing it on multiple machines, which then function as a single logical unit to increase performance.

5. Data Transformation and Integration:

- Allows you to perform lightweight transformations on data before it is transferred or processed further in the cloud.

How It Works:

1. Installation:

- The SHIR is installed on a local machine (e.g., a server in your on-premises network or a virtual machine in a private network).

2. Registration:

- After installation, the runtime is registered with your Azure Data Factory instance using a generated authentication key.

3. Configuration:

- Data Factory pipelines are configured to use the SHIR for activities like copying data, running lookups, or executing custom scripts.

4. Communication:

- The SHIR establishes a secure outbound connection to Azure Data Factory and performs data movement or activity execution as configured in the pipeline.

Use Cases:

1. On-Premises to Cloud Data Transfer:

- Migrating data from on-premises SQL Server to Azure Blob Storage or Azure SQL Database.

2. Private Network to Cloud Integration:

- Integrating private network systems, like SAP or Oracle databases, with Azure services.

3. Custom Processing:

- Running custom scripts or commands that need to be executed on local infrastructure.

Comparison to Azure Integration Runtime:

- **Azure Integration Runtime** is cloud-based and works for most scenarios where data is fully hosted in the cloud.
- **Self-Hosted Integration Runtime** is designed for hybrid or on-premises scenarios where data cannot be accessed directly from Azure.

To begin with the Lab

1. In this lab we are going to create a Self-hosted Integrated runtime in Azure Data Factory. So, first, we will create a Virtual Machine on which we are going to install a web server. After that, we will install the self-hosted integrated runtime over the virtual machine then we will establish a connection between the self-hosted runtime and azure data factory. Once the connection is successful then we will create a mapping data flow in which we will copy the data created by the web server into our data lake and then we will create a table in our dedicated SQL pool and load that data into the table.
2. So, now we will Create a **Virtual Machine** based on **Windows Server 2022** in a new resource group and install IIS on it then a log file will be generated.

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ

MSDN Platforms Subscription

Resource group * ⓘ

virtualMachine

[Create new](#)

Instance details

Virtual machine name * ⓘ

appvm1



Region * ⓘ

(Europe) North Europe



Availability options ⓘ

No infrastructure redundancy required



Security type ⓘ

Trusted launch virtual machines

[Configure security features](#)

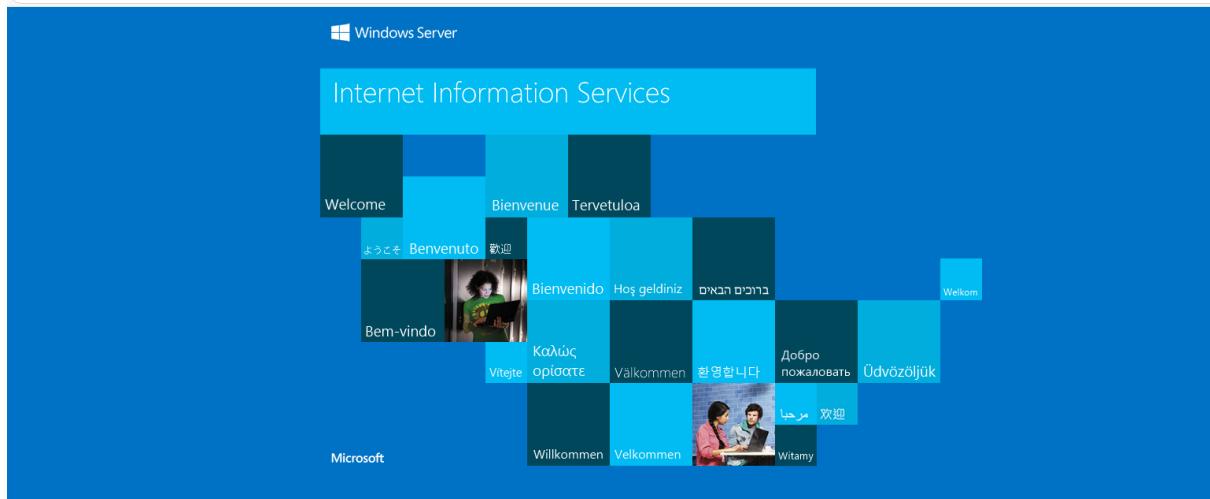
Image * ⓘ

Windows Server 2022 Datacenter - x64 Gen2



[See all images](#) | [Configure VM generation](#)

⚠ Not secure | 52.169.178.77



3. After installing the Web server on your VM you have to open the Microsoft Edge browser in your VM and then search for self-hosted integration runtime. Open the first link from the official Microsoft website.

Microsoft Bing

self hosted integration runtime

SEARCH COPILOT VIDEOS IMAGES MAPS NEWS SHOPPING MORE

About 296,000 results

A self-hosted integration runtime is an instance of the Azure Integration Runtime that is installed and managed on-premises [1](#) [2](#) [3](#). It provides a secure connection between on-premises resources and the cloud, allowing for the transfer of data between the two [1](#). The self-hosted integration runtime is a service running in Azure Data Factory, but you can add local compute nodes on local servers in your on-premises network [3](#).

Learn more:

- [1 Configuration of Self-Hosted Integration Runt...](#)
- [2 Create and manage Integration Runtimes | ...](#)
- [3 Use Azure Data Factory to Load /](#)

[Feedback](#)

Microsoft
https://www.microsoft.com › en-us › download › details.aspx ›

Download Microsoft Integration Runtime from Official Microsoft ...

Learn how to download and install the Microsoft Integration Runtime, a customer managed data integration infrastructure for Azure Data Factory and Azure Synapse Analytics. The integration runtime can move data in and out of private network and dispatch activities against compute ...

Other content from microsoft.com

4. Open the link and click on Download then choose the latest version of the application. Then you have to install the application in your VM.

Microsoft | Download Center Windows Office Web browsers Developer tools Xbox All Microsoft Search Cart Sign in

Copilot is your AI companion
Always by your side, ready to support you whenever and wherever you need it.
[Download the Copilot app](#)

Microsoft Integration Runtime
The Microsoft Integration Runtime is a customer managed data integration infrastructure used by Azure Data Factory and Azure Synapse Analytics to provide data integration capabilities across different network environments.

Important! Selecting a language below will dynamically change the complete page content to that language.

Select language [Download](#)

Choose the download you want

<input type="checkbox"/> File Name	Size
<input type="checkbox"/> IntegrationRuntime_5.45.8999.1.msi	1.2 GB
<input type="checkbox"/> IntegrationRuntime_5.46.9020.1.msi	1.2 GB
<input type="checkbox"/> IntegrationRuntime_5.47.9060.1.msi	1.1 GB
<input checked="" type="checkbox"/> IntegrationRuntime_5.48.9076.1.msi	1.1 GB
<input type="checkbox"/> Release Notes.doc	266.0 KB

[Download](#)

Total size: 1.1 GB

5. The installation is simple just double-click on the application and click on OK until it starts to install.
6. When it starts to install go to Azure Data Factory and navigate to the manage section. Go to integration runtimes and choose Azure self-hosted runtime choose it and click on continue.

Integration runtimes

The integration runtime (IR) is the compute infrastructure to provide the following data integration capabilities

[+ New](#) [Refresh](#)

Showing 1 - 1 of 1 items

Name ↑↓	Type ↑↓	Sub-type ↑↓	Status ↑↓
AutoResolveIntegrationR...	Azure	Public	✓ Running

Integration Runtime is the native compute used to execute or dispatch activities. Choose what integration runtime to create based on required capabilities. [Learn more](#)

 **Azure, Self-Hosted**
Perform data flows, data movement and dispatch activities to external compute.

 **Azure-SSIS**
Lift-and-shift existing SSIS packages to execute in Azure.

 **Airflow (Preview)**
Use this for running your existing DAGs

[Continue](#) [Cancel](#)

7. Then for the network environment choose the self-hosted again and click on continue.

Integration runtime setup

Network environment:

Choose the network environment of the data source / destination or external compute to which the integration runtime will connect to for data flows, data movement or dispatch activities:



Azure

Use this for running data flows, data movement, external and pipeline activities in a fully managed, serverless compute in Azure.



Self-Hosted

Use this for running activities in an on-premises / private network

[View more](#) ▾

External Resources:

You can use an existing self-hosted integration runtime that exists in another resource. This way you can reuse your existing infrastructure where self-hosted integration runtime is setup.



Linked Self-Hosted

[Learn more](#)

[Continue](#)

[Back](#)

[Cancel](#)

8. Give it a name and click on Create.

Name * ⓘ

Description

Type

9. This will give us 2 keys; we can choose any one of them.

Settings Nodes Auto update Sharing Links

Install integration runtime on Windows machine or add further nodes using the Authentication Key.

Name ⓘ

Option 1: Express setup

[Click here to launch the express setup for this computer](#)

Option 2: Manual setup

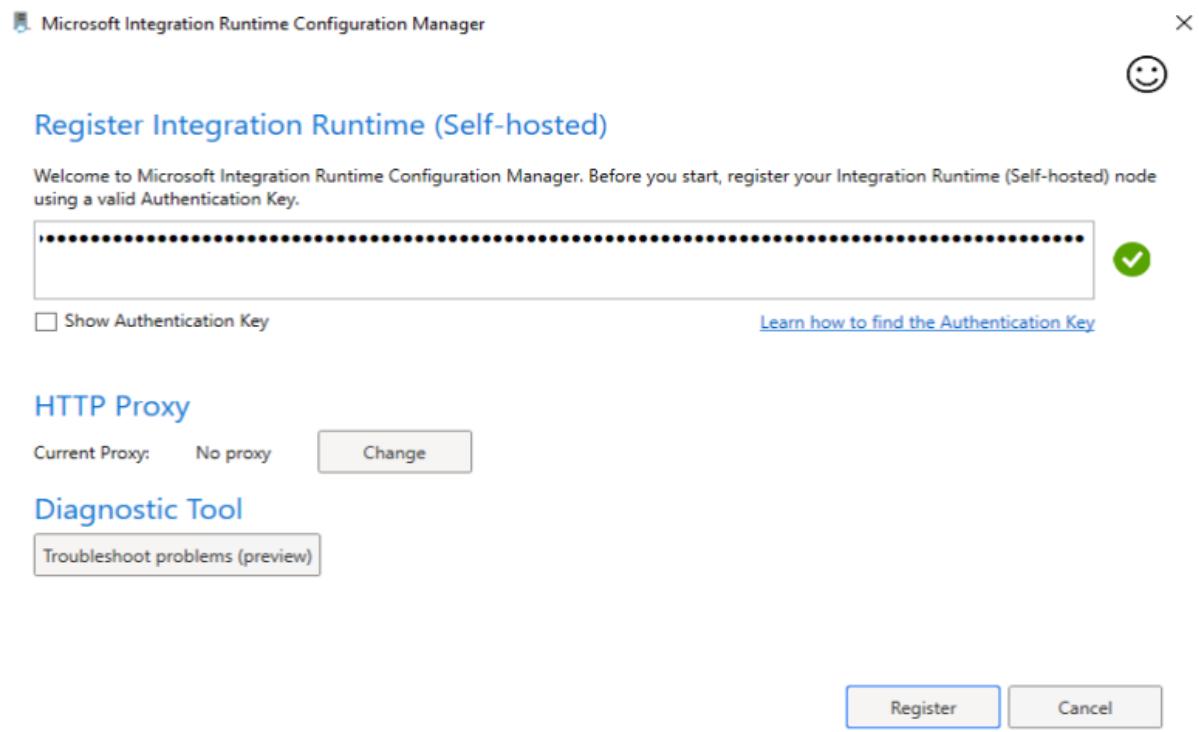
Step 1: [Download and install integration runtime](#)

Step 2: Use this key to register your integration runtime

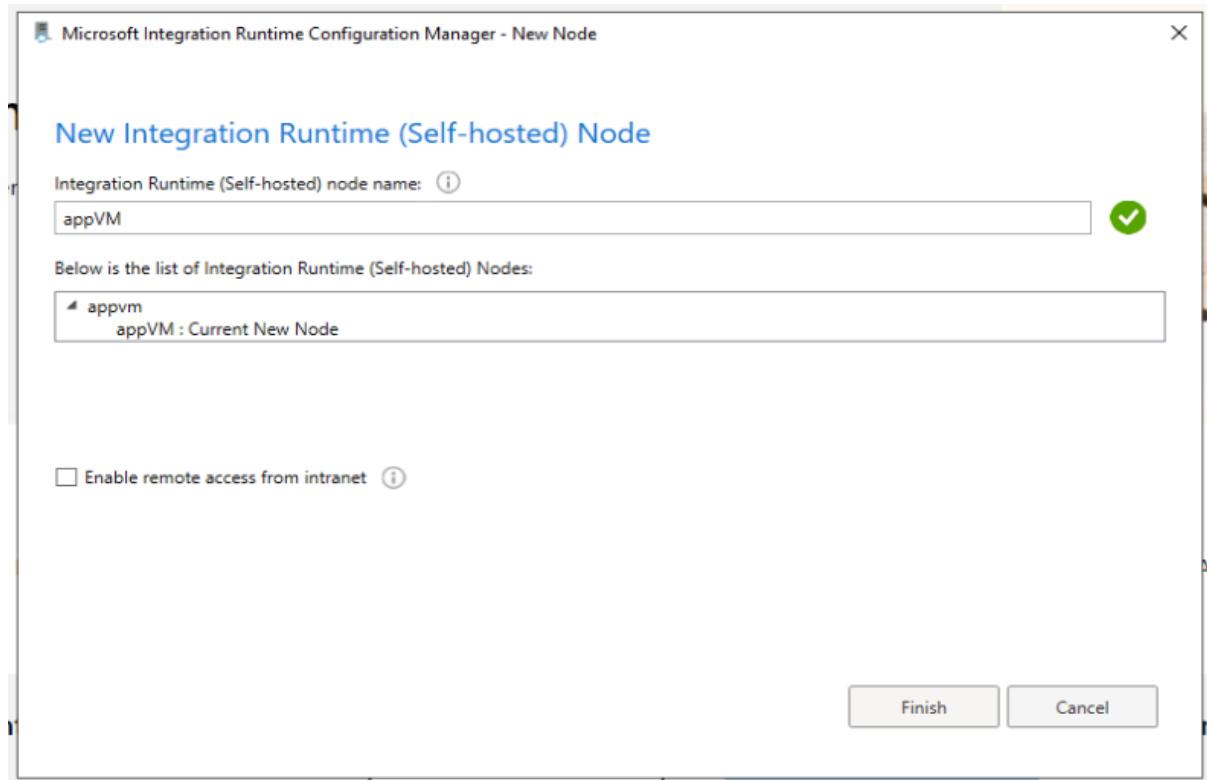
Name	Authentication key	Actions
Key1	IR@5fa7e530-0bfe-45ab-a912-292805ae399e@datafactory2361@Serv	
Key2	IR@5fa7e530-0bfe-45ab-a912-292805ae399e@datafactory2361@Serv	

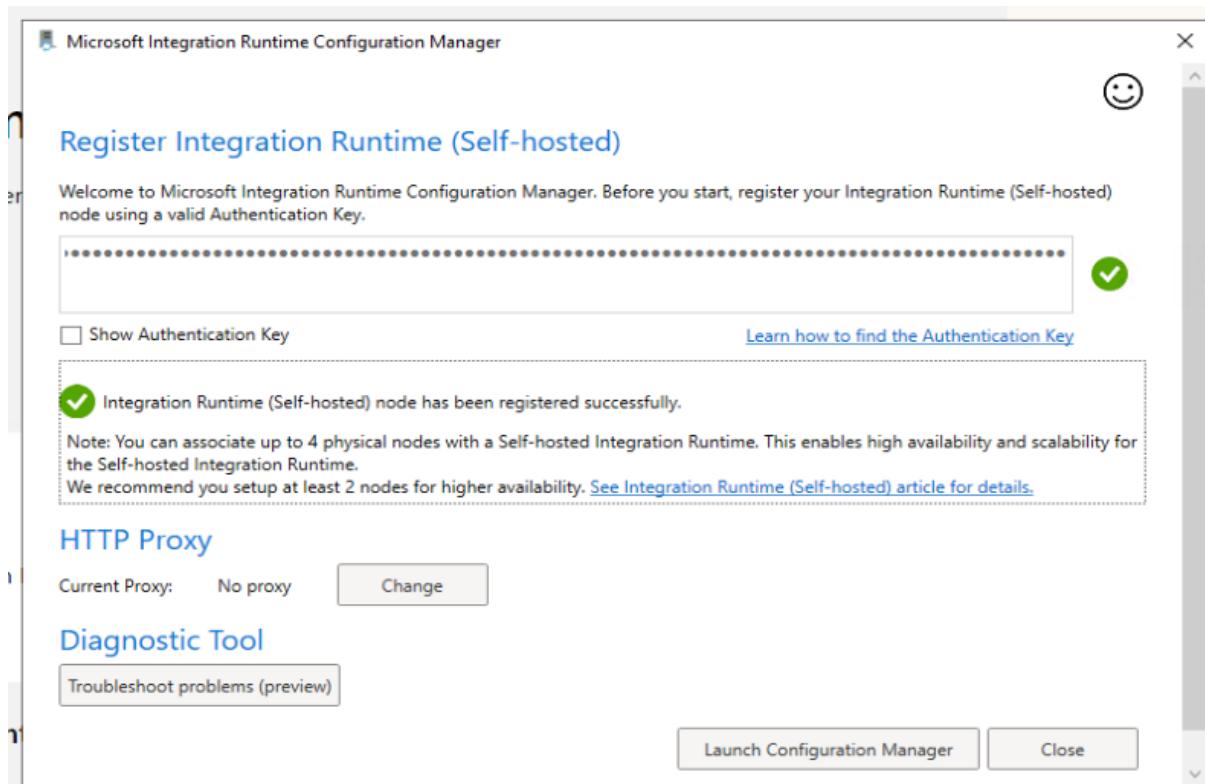
10. Once the installation is complete click on the finish button and it will give us the register page where we can register our self-hosted integrated runtime.

11. So, copy one of the keys given in ADF and paste it here in the authentication area then click on register.



12. You will see the name of your integrated runtime. Now click on Finish.





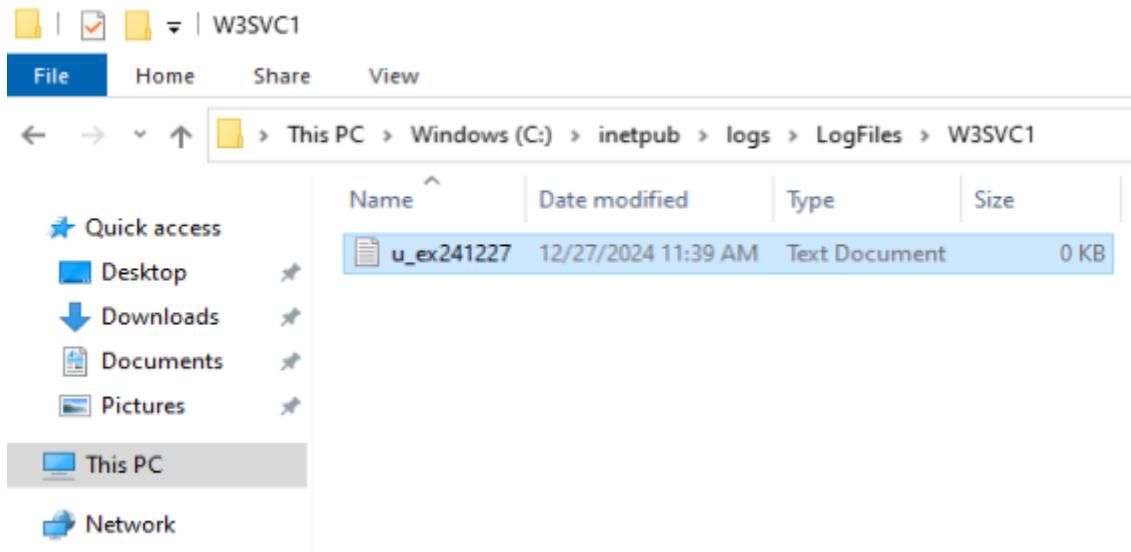
13. Now go back to your Azure Data Factory and here if you refresh the page for integration runtimes you will see your runtime.

Integration runtimes

The integration runtime (IR) is the compute infrastructure to provide the following data integration capabilities across different network environment. [Learn more](#)

Showing 1 - 2 of 2 items							
Name ↑	Type ↑	Sub-type ↑	Status ↑	Related ↑	Region ↑	Version ↑	
AutoResolveIn... (1)	Azure	Public	Running	0	Auto Resolve	---	
appvm	Self-Hosted	---	Running	0	---	5.48.9076.1	

14. Also, in your VM if you go to this location you will see the logs file which we will copy inside our data lake.



Create copy data activity to copy the data from appvm to the data lake.

15. Create a new pipeline in this we will use the copy data activity. Then we will add a source.

A screenshot of the Azure Data Factory pipeline editor. At the top, there are validation status icons: 'Validate' (green checkmark), 'Validate copy runtime' (green checkmark), 'Debug' (blue play button), and 'Add trigger' (lightning bolt icon). On the right, there's a 'Properties' panel with tabs for 'General' (selected) and 'Related'. The 'Name' field is set to '07-PipelineProcessLogData'. Below the properties, there are sections for 'Description' (empty) and 'Annotations' (with a '+ New' button). The main workspace contains a 'Copy data' activity card. The card has a red circle with a question mark icon at the top right. It shows the source dataset as 'copy log data' and the sink dataset as 'appvm'. There are four small circular icons below the names: a trash can, a copy/paste, a refresh, and a plus sign. Below the card, there are tabs for 'General', 'Source' (selected), 'Sink' (with a count of 1), 'Mapping', 'Settings', and 'User properties'. Under 'Source dataset*', there is a dropdown menu labeled 'Select...' and a '+ New' button. A vertical line with a circle at the top connects the 'Copy data' card to a downstream component.

16. To create a source data set, go to file and choose file system.

Select a data store

 Search



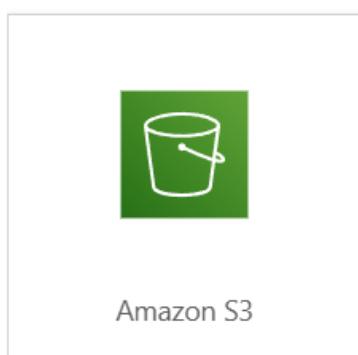
All

Azure

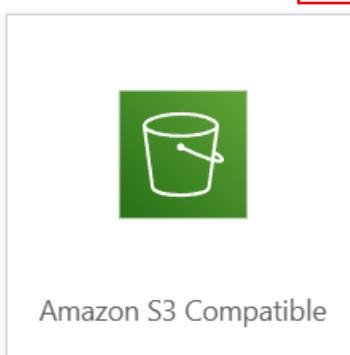
Database

File

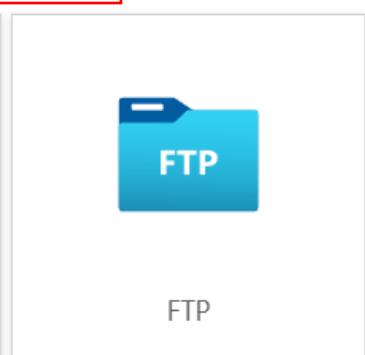
Generic protocol



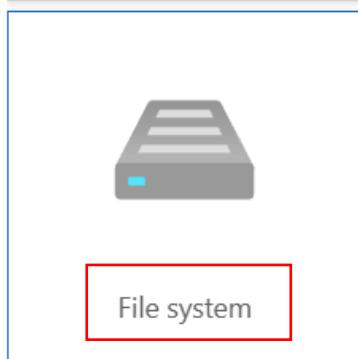
Amazon S3



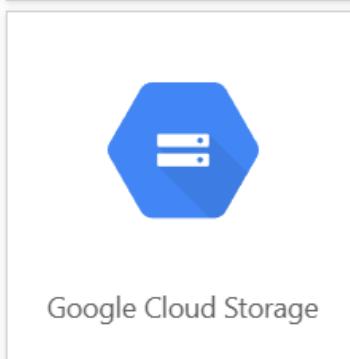
Amazon S3 Compatible



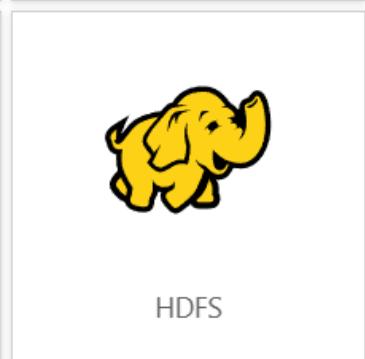
FTP



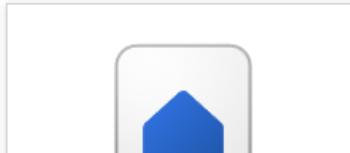
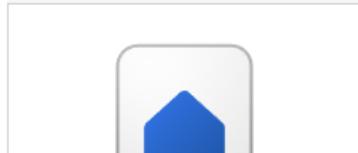
File system



Google Cloud Storage



HDFS

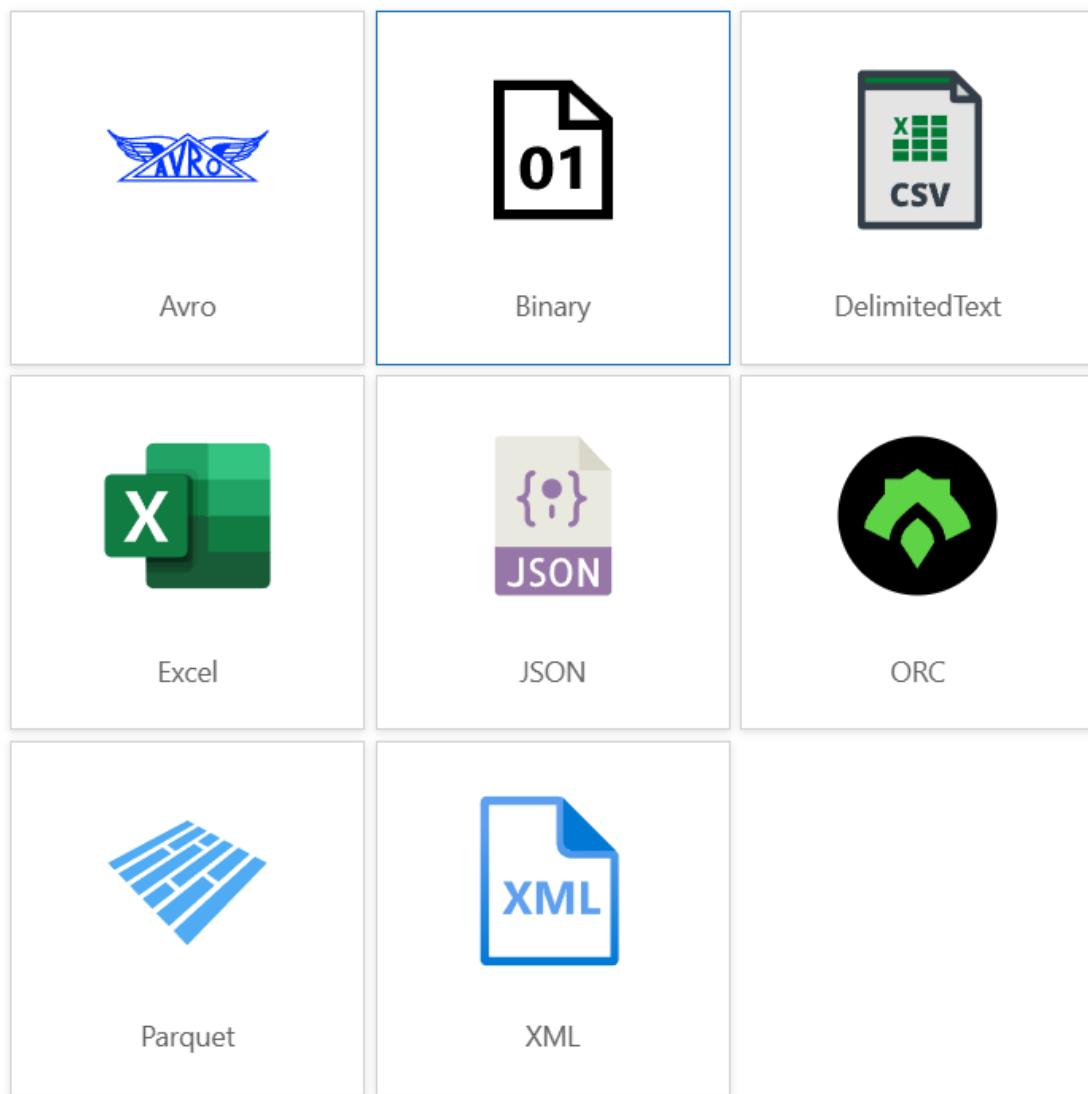


Continue

Cancel

17. Choose the binary format as my file does not belong to any of these formats.

Choose the format type of your data



Continue

Back

Cancel

18. Here give a name and choose the integration runtime as appvm and in the host give the file location then give the admin username and password of your virtual machine. When you try to test the connection it will give you an error because of security reasons so, go to the command prompt in your VM.

New linked service

 File system [Learn more](#) 

Name *

Description

Connect via integration runtime * 

appvm 



 The credentials are stored in the machines of self-hosted integration runtime if you don't choose to store them in Azure Key Vault.

Host * 

User name *

 Password

 Azure Key Vault

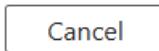
Password *

Annotations

 New

 Parameters

 Create

 Cancel

 Test connection

19. Here in the CMD you have to do the same things as shown in the snapshot. Just the same commands.

```

Administrator: Command Prompt
Microsoft Windows [Version 10.0.20348.2849]
(c) Microsoft Corporation. All rights reserved.

C:\Users\demouser>cd ..
C:\Users>cd ..
C:\>cd "Program Files"
C:\Program Files>cd "Microsoft Integration Runtime"
C:\Program Files\Microsoft Integration Runtime>cd 5.0
C:\Program Files\Microsoft Integration Runtime\5.0>cd shared
C:\Program Files\Microsoft Integration Runtime\5.0\Shared>.\dmgcmd.exe -DisableLocalFolderPathValidation
C:\Program Files\Microsoft Integration Runtime\5.0\shared>_

```

20. After that test the connection this time it will be successful.

Connect via integration runtime * ⓘ

appvm ▼ 

! The credentials are stored in the machines of self-hosted integration runtime if you don't choose to store them in Azure Key Vault.

Host * ⓘ

C:\inetpub\logs\LogFiles\W3SVC1

User name *

demouser

Password **Azure Key Vault**

Password *

Annotations

+ New

✓ Connection successful  Test connection

Create Cancel

21. Once the linked service is created click on OK.

Set properties

Name
appvm_logfile

Linked service *
appvm

Connect via integration runtime * ⓘ
✓ appvm

File path
C:\inetpub\logs\LogFiles\W3SVC1 / Directory / File name

> Advanced

22. Then give the wildcard file path for everything here.

General Source ¹ Sink Mapping Settings User properties

Source dataset * appvm_logfile Open New Learn more

File path type File path in dataset File filter Wildcard file path List of files

Wildcard paths Wildcard folder path / **.***

Start time (UTC)
End time (UTC)

Filter by last modified

Recursively

Delete files after completion

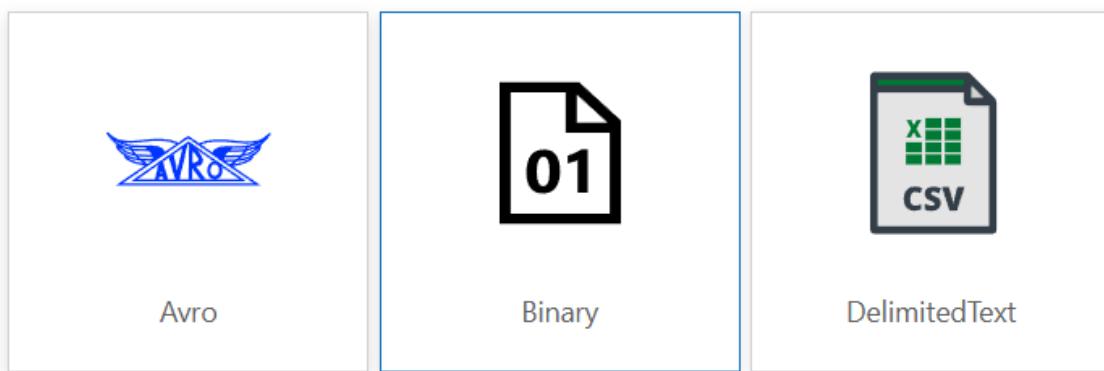
Max concurrent connections

NOTE: After creating the source go to the Logs file inside your VM and delete it then open the web server and refresh it 4-5 times it will create the Logs file once more then come back to ADF and create your sink.

23. Now create a sink. Choose to create a new dataset choose binary, choose your linked service, and create the sink. Run the pipeline we'll have the data inside our data lake

Select format

Choose the format type of your data



General Source **Sink** Mapping Settings User properties

Sink dataset * [gen2datafactory2163_binarydatalog.csv](#) [Open](#) [New](#) [Learn more](#)

Copy behavior [Preserve hierarchy](#)

Max concurrent connections

Block size (MB)

Metadata [New](#)

Upload Add Directory Refresh Rename Delete Change tier Acquire lease Break lease Give feedback

Authentication method: Access key (Switch to Microsoft Entra user account)
Location: data

Search blobs by prefix (case-sensitive)

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
<input type="checkbox"/> u_ex241227.log	12/27/2024, 5:45:18 ...	Hot (Inferred)		Block blob	2.56 KiB	Available

24. Now we will extract data from this file and put it in our dedicated SQL pool.

25. But first we have to create a table in the dedicated SQL pool.

Run Undo Publish Query plan Connect to datapool

```
1 CREATE TABLE WebLogs
2 (
3     LogDate date,
4     IPAddress varchar(20),
5     RequestMethod varchar(10),
6     RequestResource varchar(1000)
7 );
8
9 SELECT * FROM WebLogs
```

26. After creating the table come to the storage account and view the data of your file. Here first 4 lines are of no use to us.

u_ex241227.log ...

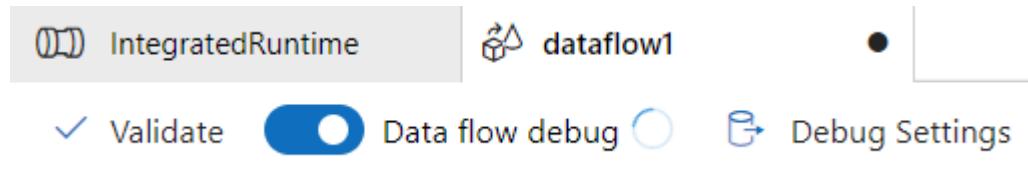
Blob

Save Discard Download Refresh Delete

Overview Versions Edit Generate SAS

```
1 #Software: Microsoft Internet Information Services 10.0
2 #Version: 1.0
3 #Date: 2024-12-27 11:39:16
4 #Fields: date time s-ip cs-method cs-uri-stem cs-uri-query s-port cs-username c-ip cs(User-Agent) cs(Referer)
5 2024-12-27 11:39:16 10.0.0.4 GET / - 80 - 103.226.202.21 Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+AppleWebKit/602.1.27.90.2 Safari/537.45
6 2024-12-27 11:39:16 10.0.0.4 GET /iisstart.png - 80 - 103.226.202.21 Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64;+Safari/537.45) AppleWebKit/602.1.27.90.2
7 2024-12-27 11:39:16 10.0.0.4 GET /favicon.ico - 80 - 103.226.202.21 Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64;+Safari/537.45) AppleWebKit/602.1.27.90.2
8 2024-12-27 11:44:40 10.0.0.4 GET / - 80 - 167.94.146.56 - - 200 0 0 25
9 2024-12-27 11:44:43 10.0.0.4 GET / - 80 - 167.94.146.56 Mozilla/5.0+(compatible;+CensysInspect/1.1;+https://censys.io) AppleWebKit/602.1.27.90.2
10 2024-12-27 11:44:45 10.0.0.4 GET /favicon.ico - 80 - 167.94.146.56 Mozilla/5.0+(compatible;+CensysInspect/1.1;+https://censys.io) AppleWebKit/602.1.27.90.2
11 2024-12-27 11:45:12 10.0.0.4 GET / - 80 - 95.214.55.79 Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+AppleWebKit/602.1.27.90.2 Safari/537.45
12 2024-12-27 11:49:50 10.0.0.4 GET / - 80 - 206.168.34.217 - - 200 0 0 101
13 2024-12-27 11:49:55 10.0.0.4 GET / - 80 - 206.168.34.217 Mozilla/5.0+(compatible;+CensysInspect/1.1;+https://censys.io) AppleWebKit/602.1.27.90.2
14 2024-12-27 11:49:57 10.0.0.4 GET /favicon.ico - 80 - 206.168.34.217 Mozilla/5.0+(compatible;+CensysInspect/1.1;+https://censys.io) AppleWebKit/602.1.27.90.2
15 2024-12-27 12:01:37 10.0.0.4 GET / - 80 - 69.164.217.245 Mozilla/5.0+zgrab/0.x - 200 0 64 130
16 2024-12-27 12:01:43 10.0.0.4 GET / - 80 - 172.104.11.34 Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+13_1)+AppleWebKit/602.1.27.90.2 Safari/537.45
17 2024-12-27 12:10:05 10.0.0.4 GET /shell/killall+-9+arm7;killall+-9+arm4;killall+-9+arm;killall+-9+/bin/sh;killall+-9+/bin/bash;killall+-9+/bin/dash;killall+-9+/bin/zsh;killall+-9+/bin/sh;- - 200 0 0 36
18 2024-12-27 12:13:16 10.0.0.4 GET / - 80 - 185.16.39.9 - - 200 0 0 36
19
```

27. Go to ADF and create a new Data Flow. Since this is not going to be easy, we will start the data flow debugging at the start.



28. Our first source will be the data file from our data lake. We will create a new data set for the data lake.
29. Even though our file is not a CSV. We will use this option because binary is not working for us and this is the only way that I know to extract the data.

Select format

Choose the format type of your data



Avro



DelimitedText



Excel



JSON



ORC



Parquet



XML



Binary

Continue

Back

Cancel

30. Then browse for the log file and click on continue to create your dataset.

Set properties

Name

Linked service *
 X ▼ ✎

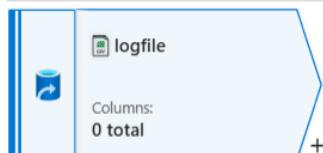
File path
 / / 📁 ▼

First row as header

Import schema
 From connection/store From sample file None

[Advanced](#)

31. Currently I cannot see any columns so I will browse for the file itself. So, click on open to open the dataset.



Source settings Source options Projection Optimize Inspect Data preview ●

Output stream name * Learn more 🔗

Description Reset ⟳

Source type * Dataset Inline

Dataset * Test connection 🔗 Open + New

Options
 Allow schema drift ⓘ
 Infer drifted column types ⓘ
 Validate schema ⓘ

32. Then if I click on import schema, I can see that we get this column which we don't want initially.

Connection Schema Parameters

Import schema **Clear**

Column name	Type
#Software: Microsoft Internet Information Services 10.0	String

33. I came back to the source settings and in the skip line count I skipped the first 4 lines of our data.

Skip line count

Sampling * Enable Disable

34. Then I imported the projections once more and I see that it is taking the whole value as a column.

Source settings Source options **Projection** Optimize Inspect Data preview ●

Define default format Detect data type Import projection Reset schema

Column name	Type	Format
2024-11-21 16:01:05 10.0.0.4 GET / - 80 - 103.226.202.3 Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+AppleWebKit/537.36+(KHTML+	abc string	Specify format
+like+Gecko)+Chrome/131.0.0.0+Safari/537.36 - 200 0 397	abc string	Specify format

35. I want to add a Select transformation to select my column and rename it. First, remove the second column and then rename the first column.

Select settings Optimize Inspect Data preview ●

Output stream name * Learn more

Description Reset

Incoming stream *

Options Skip duplicate input columns Skip duplicate output columns

Input columns * Auto mapping Reset Add mapping Delete 2 mappings: All inputs mapped

Name as
2024-11-21 16:01:05 10...
+like+Gecko)+Chrome/...

Select settings Optimize Inspect Data preview

Output stream name * renamecolumn [Learn more](#)

Description Renaming logfile to renamecolumn with columns 'logdata'

Incoming stream * logfile

Options

- Skip duplicate input columns
- Skip duplicate output columns

Input columns *

Auto mapping	Reset	Add mapping	Delete	1 mappings: 1 column(s) from the inputs left unmapped
<input type="checkbox"/> logfile's column	Y	Name as	<input type="text" value="logdata"/>	Y
<input type="checkbox"/> abc 2024-11-21 16:01:05 10.0.0.4 GET / - 80 - 10..	Y			+ Delete

36. Add another transformation for the derived column. I want to get everything as an array so for that I will split my data into an array of elements. Let me select my column logdata and go to expression builder.

Derived column's settings Optimize Inspect Data preview

Output stream name * derivedColumn1 [Learn more](#)

Description Creating/updating the columns 'logdata'

Incoming stream * renamecolumn

[Add](#) [Clone](#) [Delete](#) [Open expression builder](#)

Columns *

Column	Expression
<input type="checkbox"/> logdata	<input type="text" value="logdata"/> Enter expression... ANY + Delete

37. I am using the split expression that can split the entire string and our logdata is nothing but the string and we can split this based on a space.

38. Split expression is used to split an entire string onto an array. Then use the data preview option and look at the values they are separated. This is the progress, click on save and finish.

Dataflow expression builder [Expression reference documentation](#)

derivedColumn1

Derived Columns [Create new](#)

logdata

Column name * logdata

Expression

Expression elements

- All
- Functions
- Input schema
- Parameters
- Cached lookup

Expression values

Filter by keyword

abc dropLeft(abc string to drop characters , ANY number of items to drop)

abc dropRight(abc string to drop characters , ANY number of items to drop)

Data preview [Refresh](#)

Original Log Entry	Resulting Array
["2024-11-21","16:01:36","10.0.0.4","GET","/isstart.png","-","80","-","103.226.202.3","Mozilla/5.0+(Windows+NT...]	2024-11-21,16:01:36,10.0.0.4,GET,/isstart.png,-,80,-,103.226.202.3,Mozilla/5.0+(Windows+NT+10.0+Win64;x...]
["2024-11-21","16:03:59","10.0.0.4","GET","/","-","80","-","114.33.126.55","Mozilla/5.0+(Linux;+U;+Android+4.0.3;+ko-kr;+LG-L160L+Build/IML74K)+AppleWebKit/534...	2024-11-21,16:03:59,10.0.0.4,GET,/,-,80,-,114.33.126.55,Mozilla/5.0+(Linux;+U;+Android+4.0.3;+ko-kr;+LG-L160L+Build/IML74K)+AppleWebKit/534...
["2024-11-21","16:05:23","10.0.0.4","GET","/ReportServer","-","80","-","52.228.153.245","Mozilla/5.0+zgrab/0.x","..."]	2024-11-21,16:05:23,10.0.0.4,GET,/ReportServer,-,80,-,52.228.153.245,Mozilla/5.0+zgrab/0.x,-,404,0,2,92

39. We got the array in place. The last thing we need to do is map the columns.

40. For that we will use the derived column once again. Map them like this only.

Derived column's settings

Optimize Inspect Data preview ●

Output stream name *	<input type="text" value="Mapcolumns"/>	Learn more
Description	Creating/updating the columns 'logdata, LogDate, IPAddress, RequestMethod, RequestResource'	
Incoming stream *	<input type="text" value="derivedColumn1"/> ▼	
Columns *	+ Add Clone Delete Open expression builder	
<input type="checkbox"/> Column	Expression	
<input type="checkbox"/> LogDate	<input type="text" value="logdata[1]"/>	abc + trash
<input type="checkbox"/> IPAddress	<input type="text" value="logdata[9]"/>	abc + trash
<input type="checkbox"/> RequestMethod	<input type="text" value="logdata[4]"/>	abc + trash
<input type="checkbox"/> RequestResource	<input type="text" value="logdata[5]"/>	abc + trash

41. Now if I use the data preview option to look at the data beforehand I can see the separated data.

Derived column's settings Optimize Inspect **Data preview** ●

Number of rows + **INSERT** 12 * **UPDATE** 0 ✘ **DELETE** 0 * **UPSERT** 0 🔎 **LOOKUP** 0 ✗ **ERROR** 0 **TOTAL** 12

Refresh | abc Typecast | Modify | Map drifted | Statistics | Remove | Export to CSV | ↴

↑↓	logdata	[] ↑↓	LogDate	abc ↑↓	IPAddress	abc ↑↓	RequestMethod	abc ↑↓	RequestResource	abc ↑↓
+	[...]		2024-11-21		103.226.202.3		GET		/iisstart.png	
+	[...]		2024-11-21		103.226.202.3		GET		/favicon.ico	
+	[...]		2024-11-21		103.226.202.3		GET		/	
+	[...]		2024-11-21		103.226.202.3		GET		/iisstart.png	
+	[...]		2024-11-21		103.226.202.3		GET		/	
+	[...]		2024-11-21		103.226.202.3		GET		/iisstart.png	
+	[...]		2024-11-21		103.226.202.3		GET		/	
+	[...]		2024-11-21		103.226.202.3		GET		/iisstart.png	
+	[...]		2024-11-21		103.226.202.3		GET		/	
+	[...]		2024-11-21		103.226.202.3		GET		/iisstart.png	

42. Now add the sink for the weblogs table from the dedicated SQL pool. Create a new dataset, choose Azure Synapse Analytics and your linked service.

Set properties

Name

Linked service *

✎

Select from existing table New table

Table name

↻
 Enter manually

Import schema

From connection/store None

› Advanced

43. Then open the dataset choose the weblogs table and go to schema, import the schema.

Connection Schema Parameters

Import schema

Clear

Column name	Type
LogDate	date
IPAddress	varchar
RequestMethod	varchar
RequestResource	varchar

44. After that Map the schema and publish the changes to save everything.

Sink Settings Errors **Mapping** Optimize Inspect Data preview ●

⚠ At least one incoming column is mapped to a column in the sink dataset schema with a conflicting type, which can cause NULL values or runtime errors.

Options Skip duplicate input columns ⓘ Skip duplicate output columns ⓘ Auto mapping ⓘ 4 mappings: All

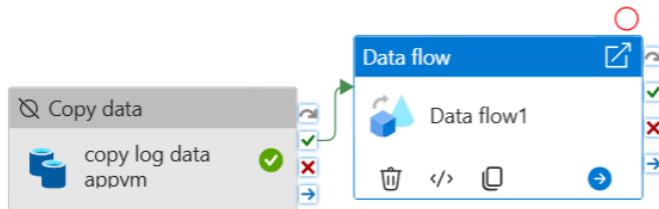
Input columns	Output columns
abc LogDate	LogDate
abc IPAddress	IPAddress
abc RequestMethod	RequestMethod
abc RequestResource	RequestResource

45. Also, check your data in the data preview option you will see that everything is sorted here.

Sink Settings Errors Mapping Optimize Inspect **Data preview** ●

↑↓	LogDate	↑↓	IPAddress	↑↓	RequestMethod	↑↓	RequestResource	↑↓
+	2024-11-21		103.226.202.3		GET		/iisstart.png	
+	2024-11-21		103.226.202.3		GET		/favicon.ico	
+	2024-11-21		103.226.202.3		GET		/	
+	2024-11-21		103.226.202.3		GET		/iisstart.png	
+	2024-11-21		103.226.202.3		GET		/	
+	2024-11-21		103.226.202.3		GET		/iisstart.png	
+	2024-11-21		103.226.202.3		GET		/	
+	2024-11-21		103.226.202.3		GET		/iisstart.png	
+	2024-11-21		114.33.126.55		GET		/	
+	2024-11-21		52.228.153.245		GET		/ReportServer	

46. Then come to the pipeline we created to import the logs data file. In the same pipeline, we will add another activity for data flow and choose our data flow here. Also, we will disable the copy data tool activity beforehand.



General **Settings** Parameters User properties

Data flow * [Open](#) [New](#)

Run on (Azure IR) * ⓘ AutoResolveIntegrationRuntime

Compute size * ⓘ

> Advanced

Logging level * ⓘ Verbose Basic None

> Sink properties

▽ Staging ⓘ

Staging linked service ⓘ [Test connection](#) [Edit](#) [New](#)

Staging storage folder / [Browse](#) | ▾

47. After that publish all the changes again and trigger your pipeline. Once your pipeline succeeds then check the data inside your table.