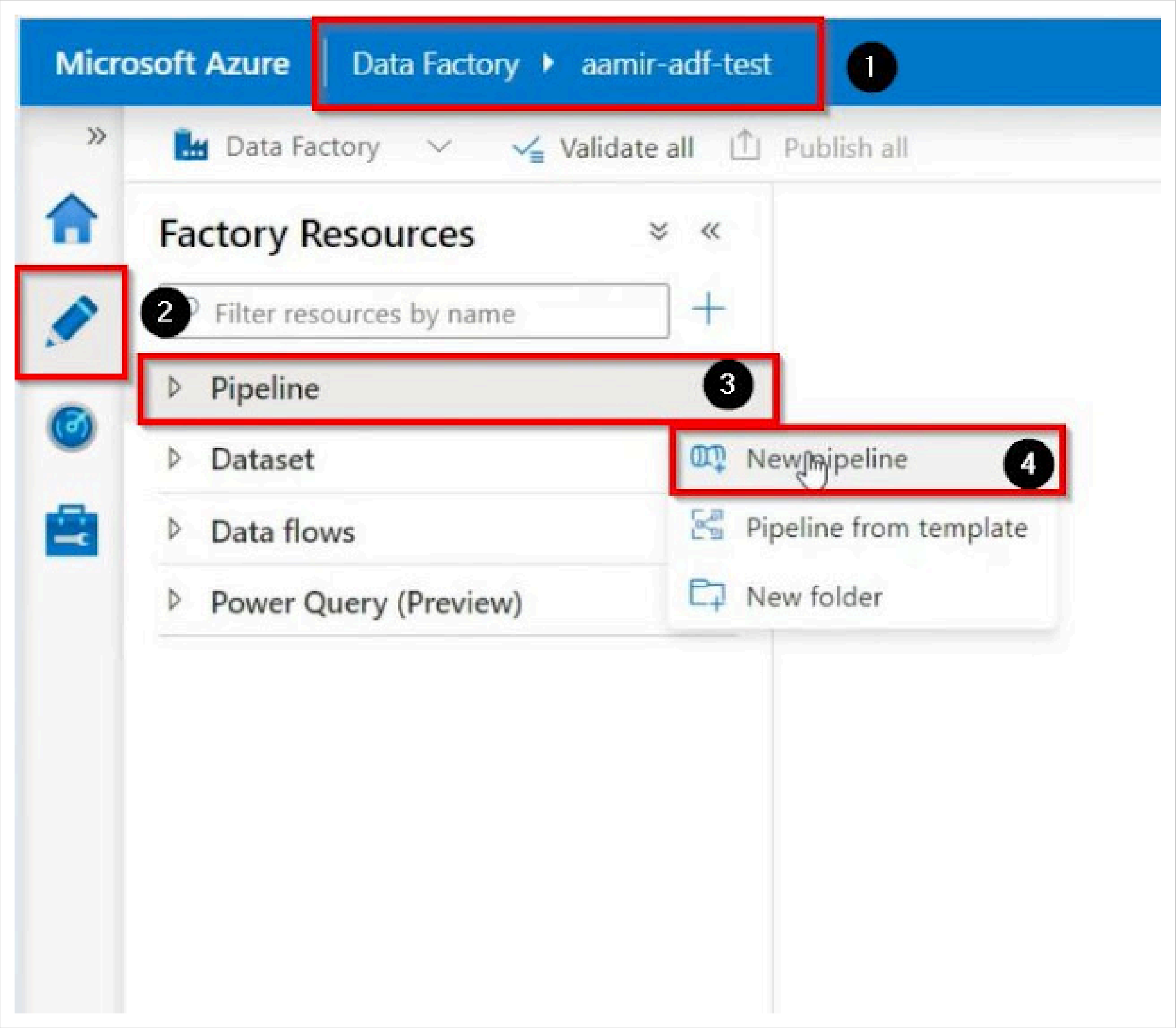
How to Create CSV Files Dynamically in Azure Blob Storage from On-Prem SQL Server Tables in Azure Data Factory

Issue: How to Create CSV Files Dynamically in Azure Blob Storage from On-Prem SQL Server Tables in Azure Data Factory

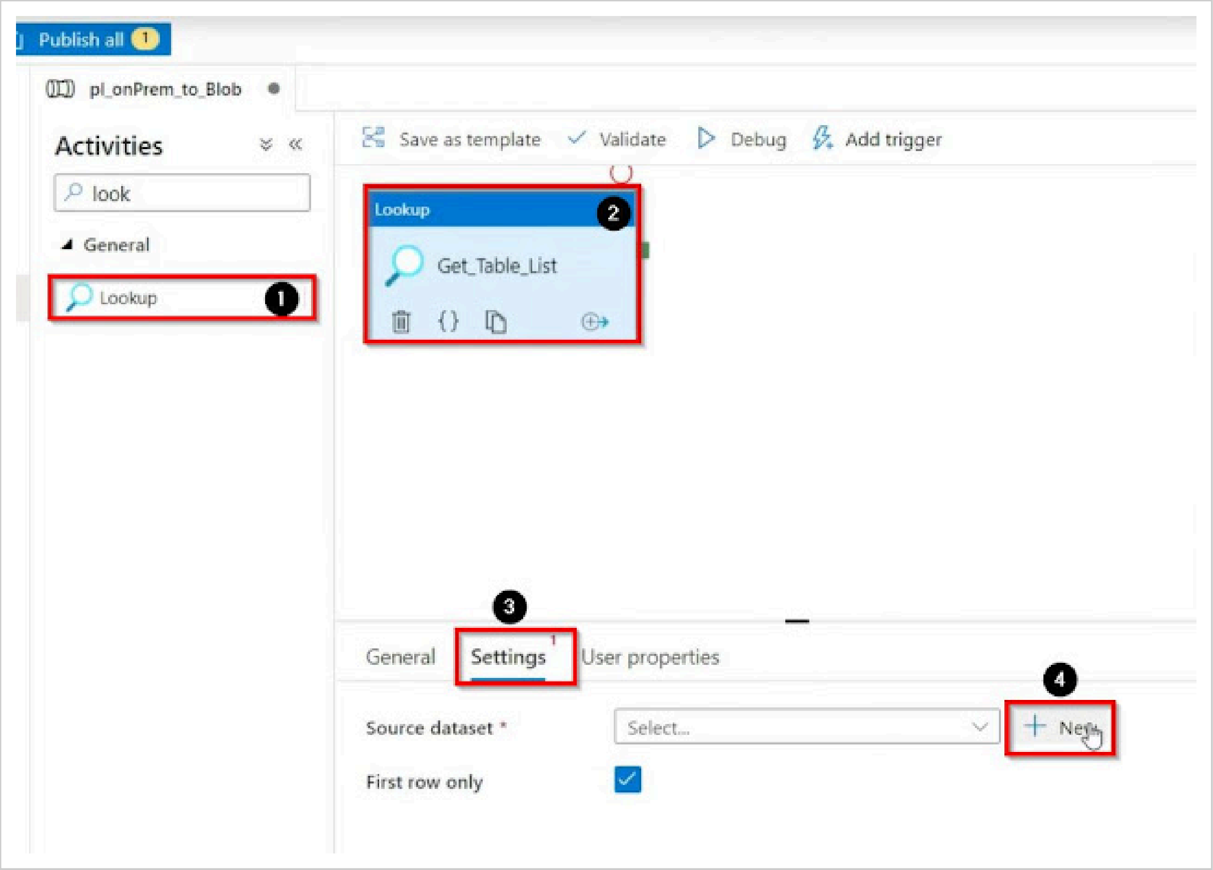
In this article, we are going to learn, how to create CSV files dynamically in Azure Blob storage from on-premises SQL server tables in Azure Data Factory, to create CSV files dynamically in Azure blob storage from on-prem, you need an active Self-Hosted integration runtime also configured in your Azure Data Factory, in this demo we will read the data from our On-Premises to the Azure blob storage by using Azure Data Factory.

Create a Pipeline:

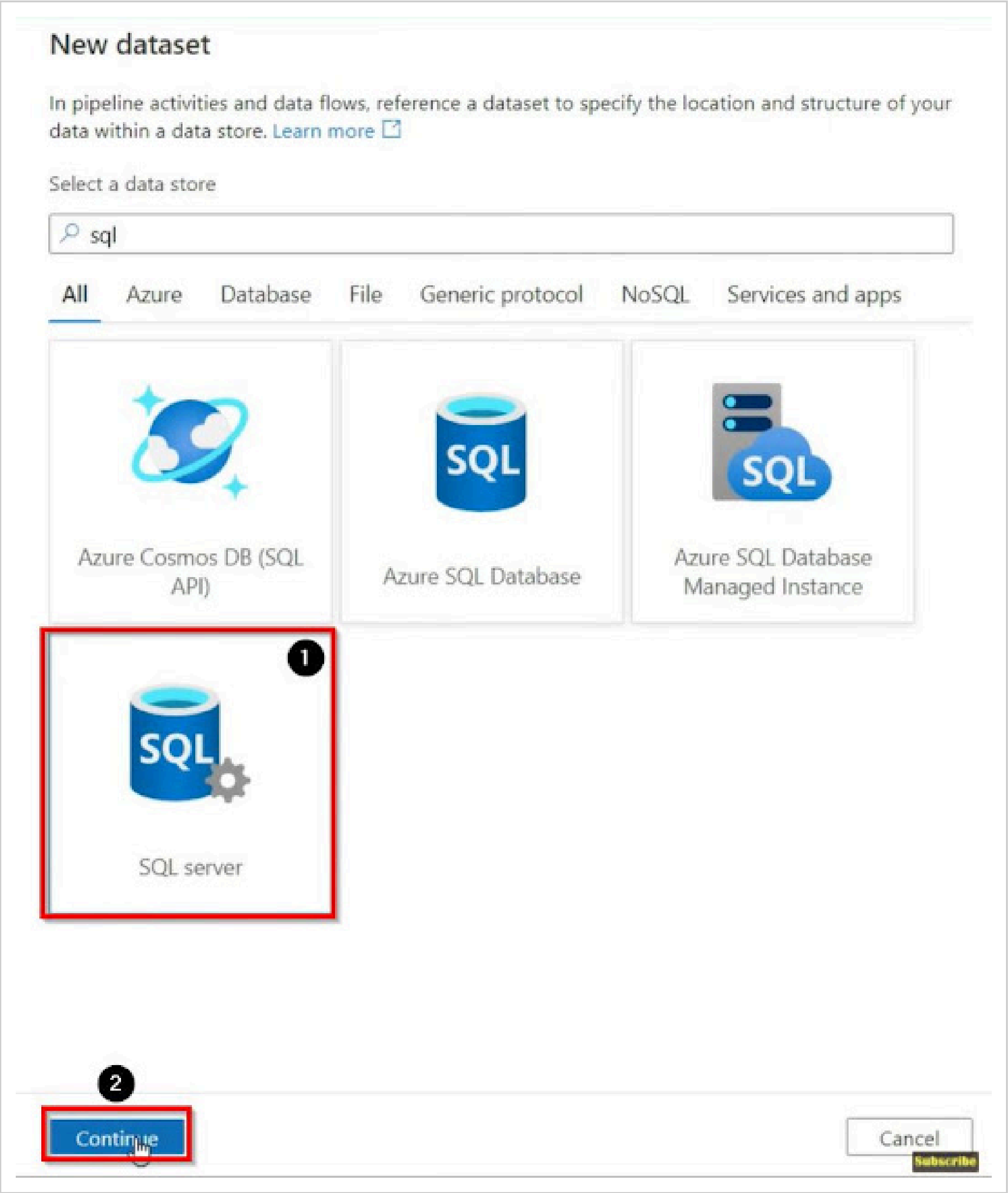
Open your Azure Data Factory studio, go to the author tab, click on the pipelines, then click on the new pipeline, to create a pipeline.



Find and drag the Lookup activity, go to the settings tab, and click on the + New button to create a new source dataset.



Select SQL Server and click on continue.

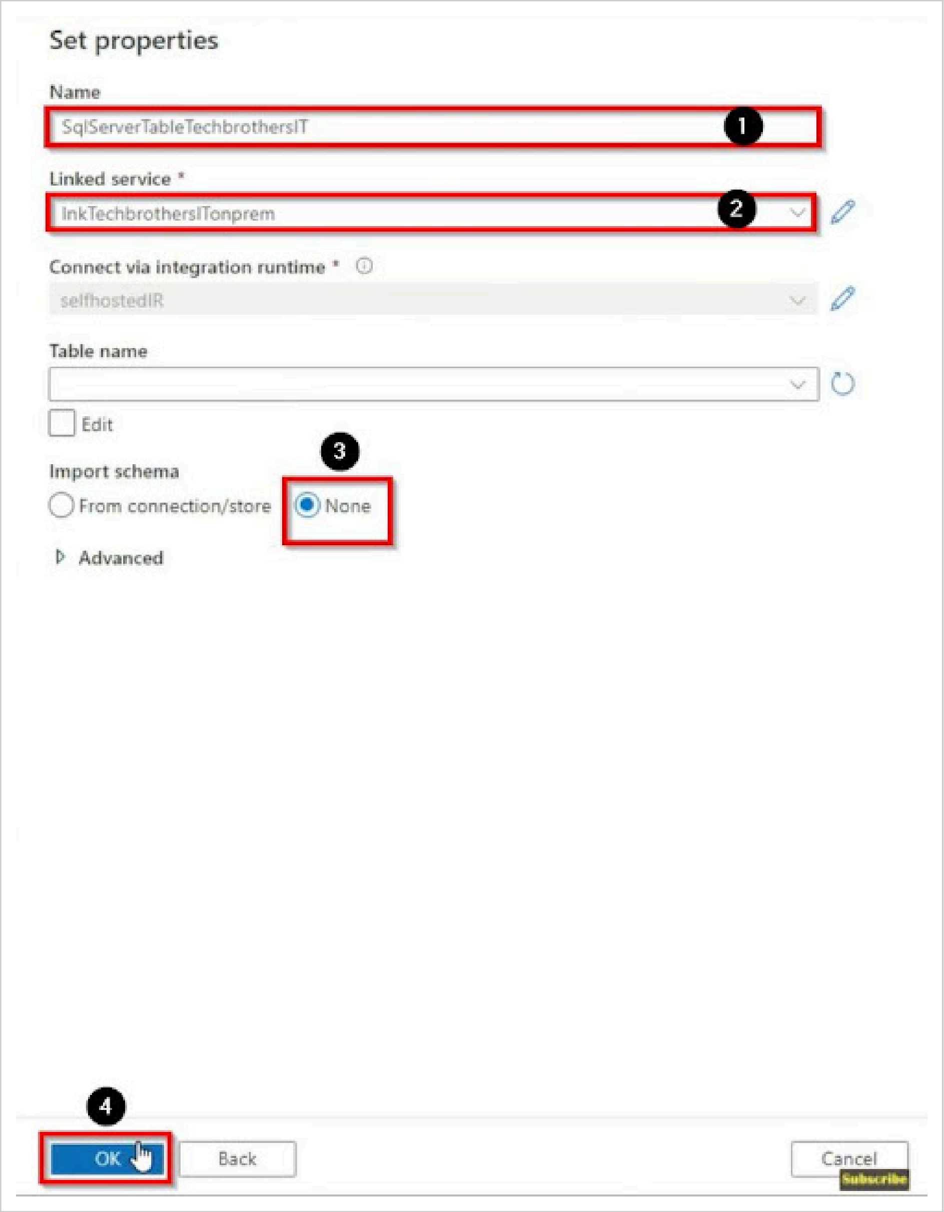


New dataset

In pipeline activities and data flows, reference a dataset to specify the location and structure of your data within a data store. Learn more

Select a data store: sql

Name your source dataset, Select the linked service, if you have already created otherwise create the linked service and then select none for import schema and click on ok.



Set properties

Name: SqlServerTableTechbrothersIT

Linked service *: lnkTechbrothersITonprem

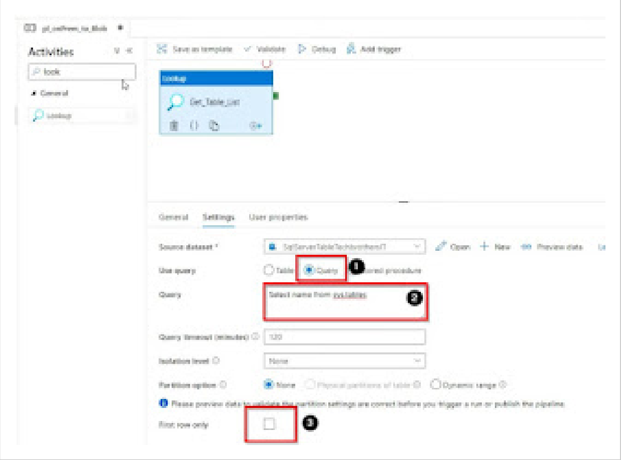
Connect via integration runtime *: selfhostedIR

Table name:

□ Edit

Import schema: ○ From connection/store ● None

Write the query to get the list of the files, then uncheck the box "First Row Only".



Source dataset *: SqlServerTableTechbrothersIT

Use query: ○ Table ● Query ○ Stored procedure

Query: Select name from sys.tables

Query timeout (minutes): 120

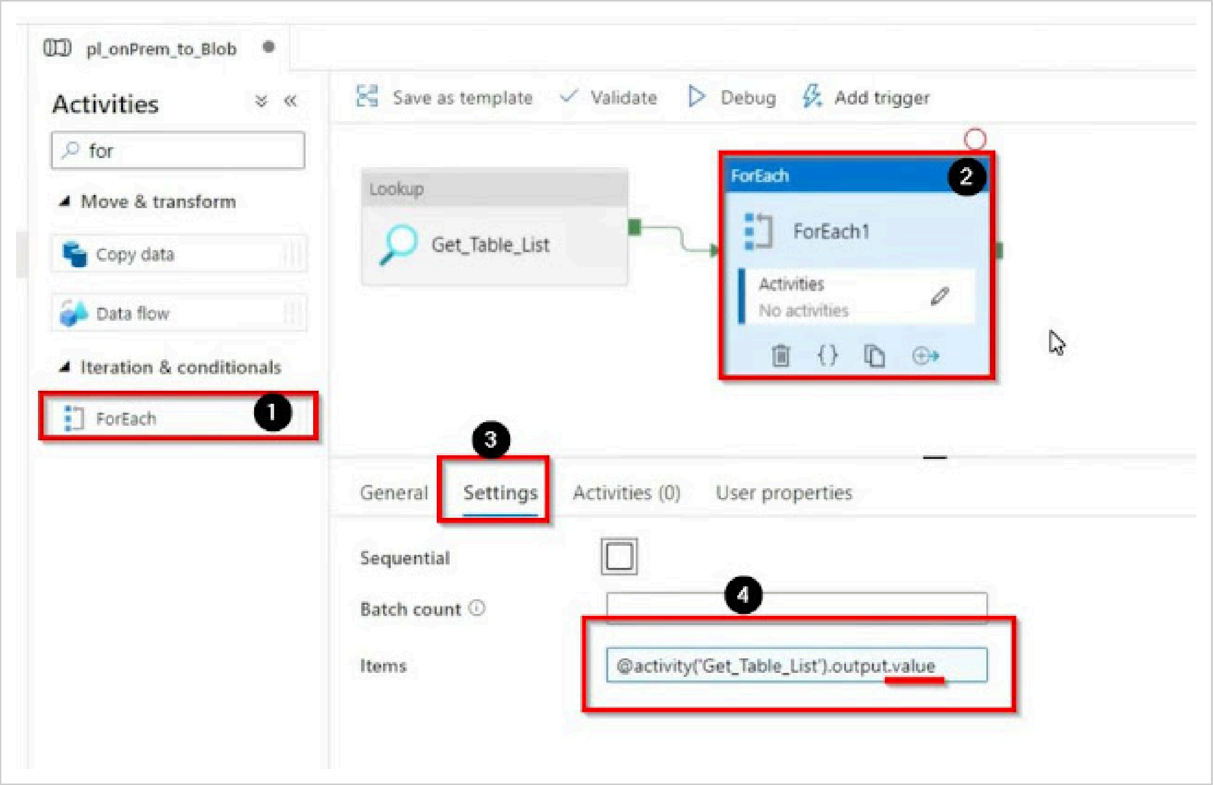
Isolation level: None

Partition option: ● None ○ Physical partitions of table ○ Dynamic range

Please preview data to validate the partition settings are correct before you trigger a run or publish the pipeline.

First row only: □

Find and drag the ForEach loop Activity, then go to the settings tab and in the items, click "Add dynamic content" then add the lookup, and then add .value at the end of the expressions.

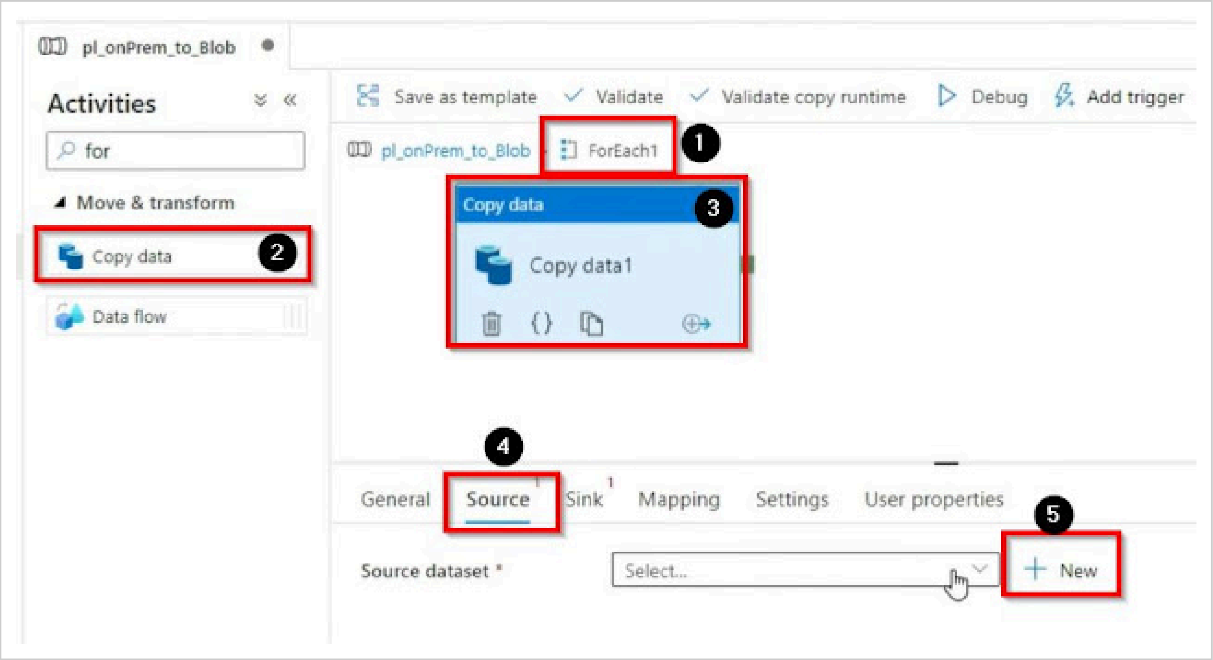


Sequential □

Batch count

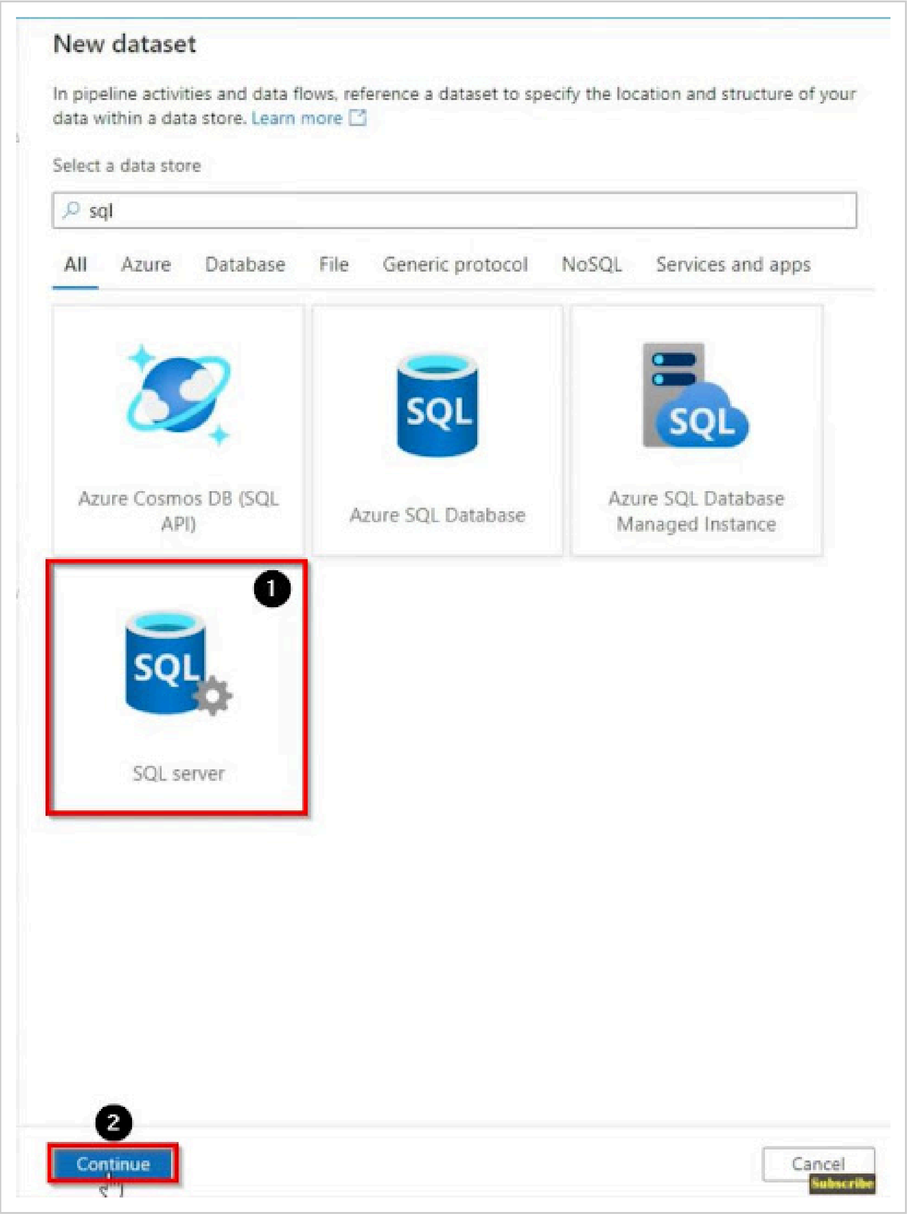
Items @activity('Get_Table_List').output.value

Go inside the ForEach loop by clicking the pencil sign on the activity, find and drag the copy data activity, then go to the source tab and click on the + New button to create a new source dataset.



Source dataset * Select...

Select SQL server, then click on continue.



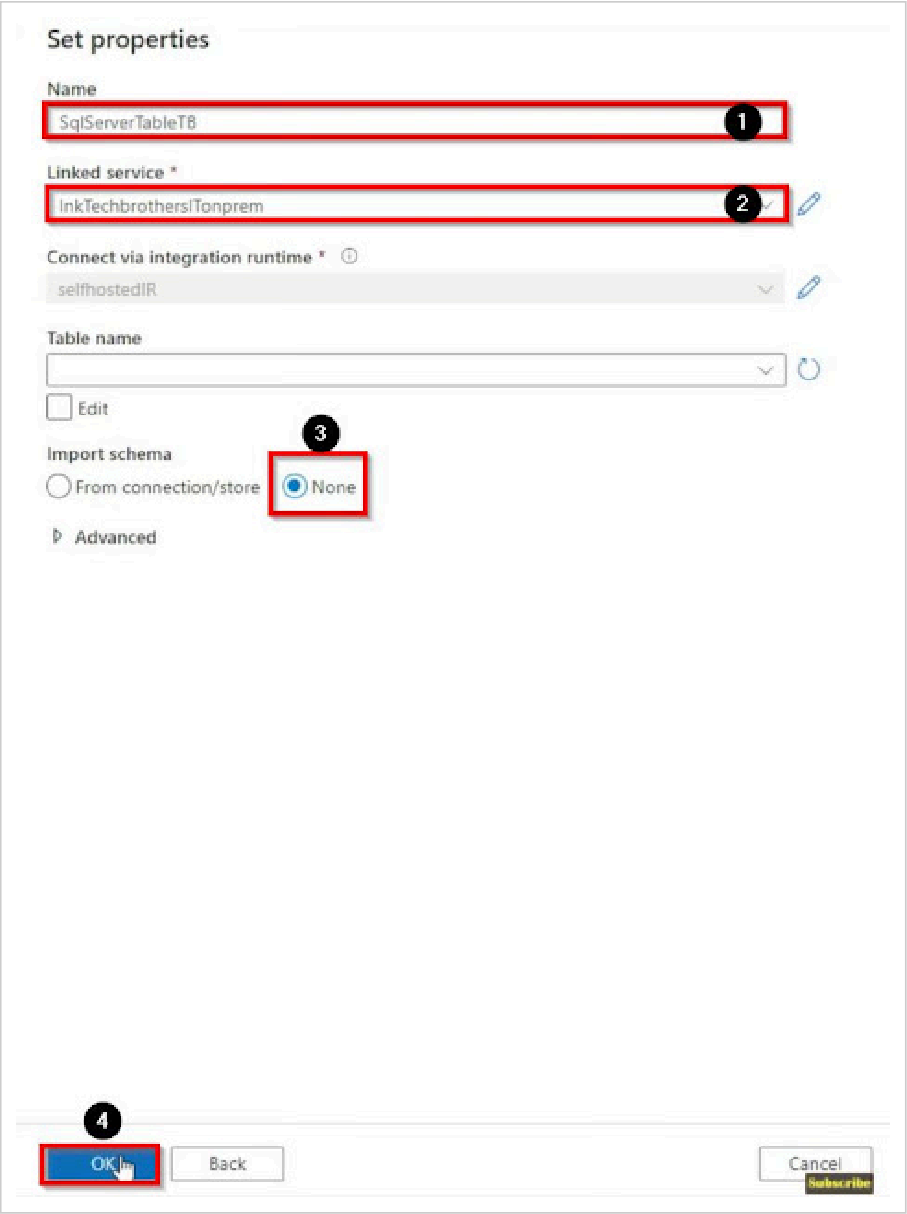
New dataset

In pipeline activities and data flows, reference a dataset to specify the location and structure of your data within a data store. Learn more

Select a data store

sql

Name your dataset, select the linked service, select none for import schema and click on ok.



Set properties

Name: SqlServerTableTB

Linked service *: lnkTechbrothersITonprem

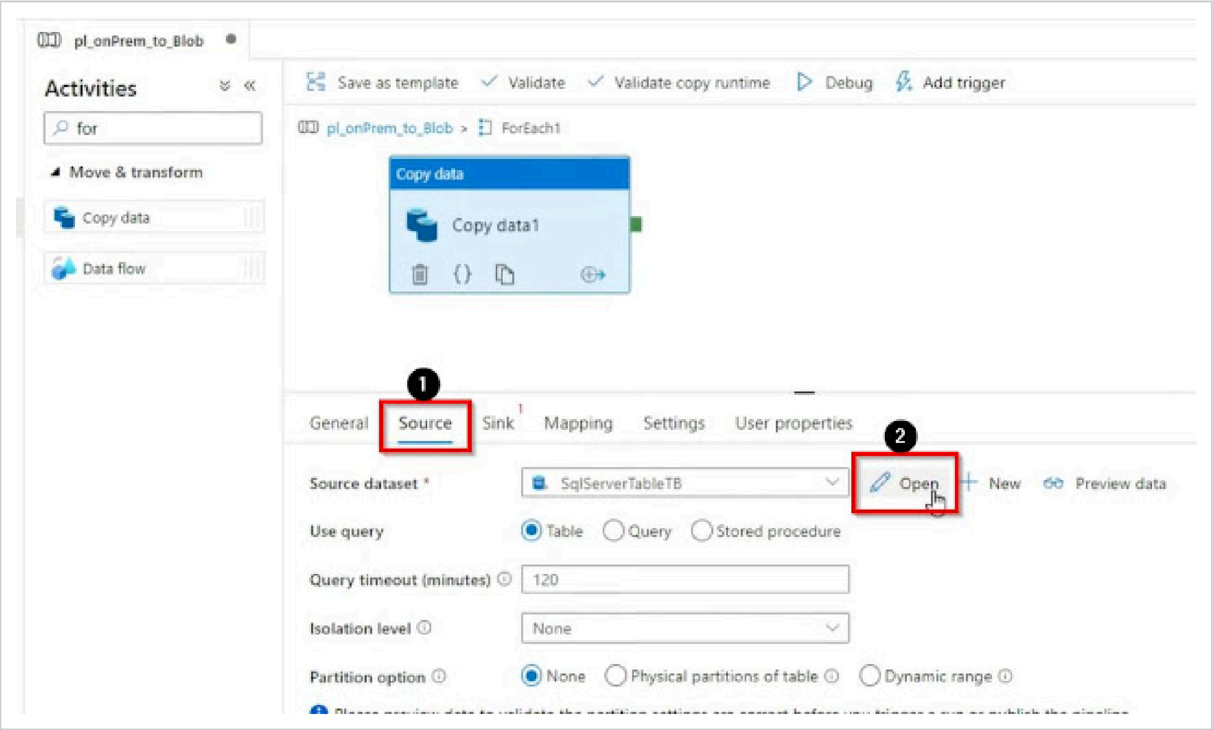
Connect via integration runtime *: selfhostedIR

Table name:

□ Edit

Import schema: ○ From connection/store ● None

Once our dataset is completed, in the source tab click on Open button and create some parameters.



Source dataset * SqlServerTableTB

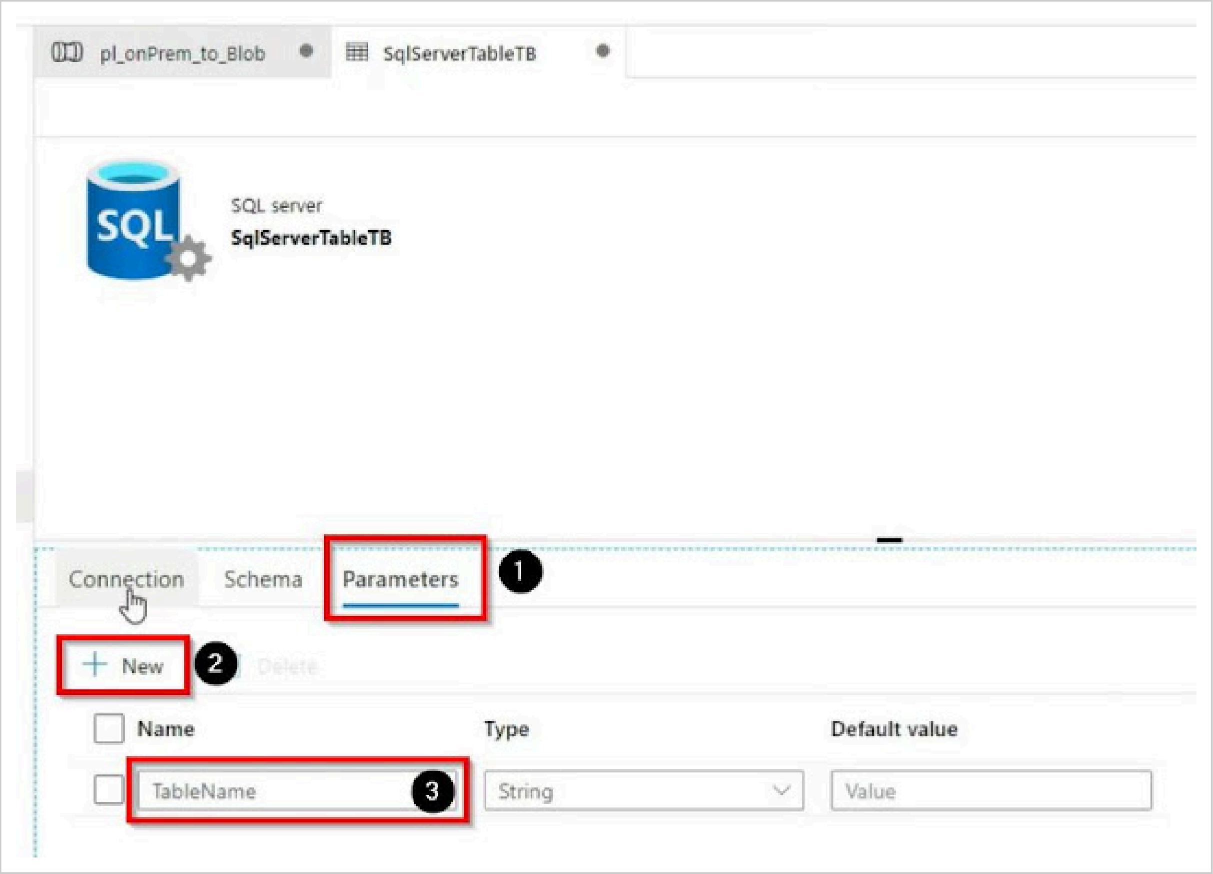
Use query ● Table ○ Query ○ Stored procedure

Query timeout (minutes) 120

Isolation level None

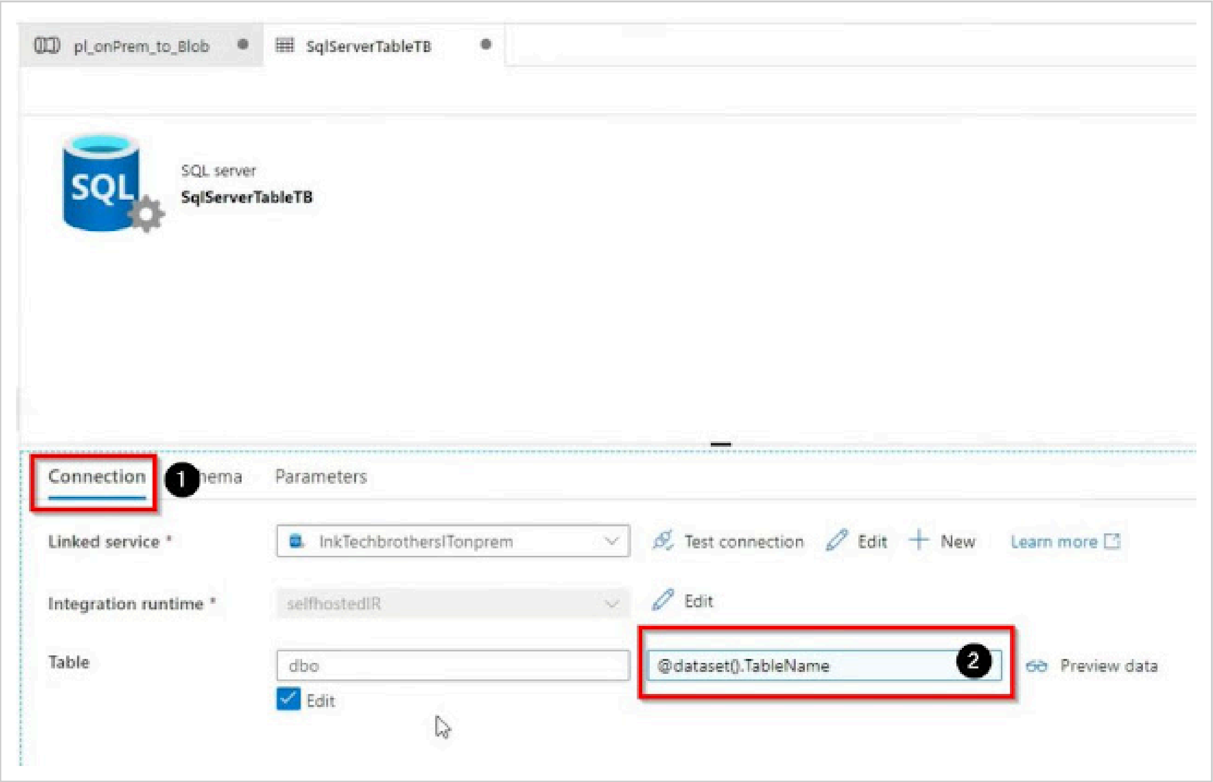
Partition option ● None ○ Physical partitions of table ○ Dynamic range

In the parameters, tab click on the + sign and name the parameter.



□	Name	Type	Default value
□	TableName	String	Value

In the connection tab, select provide the parameter that we have created.



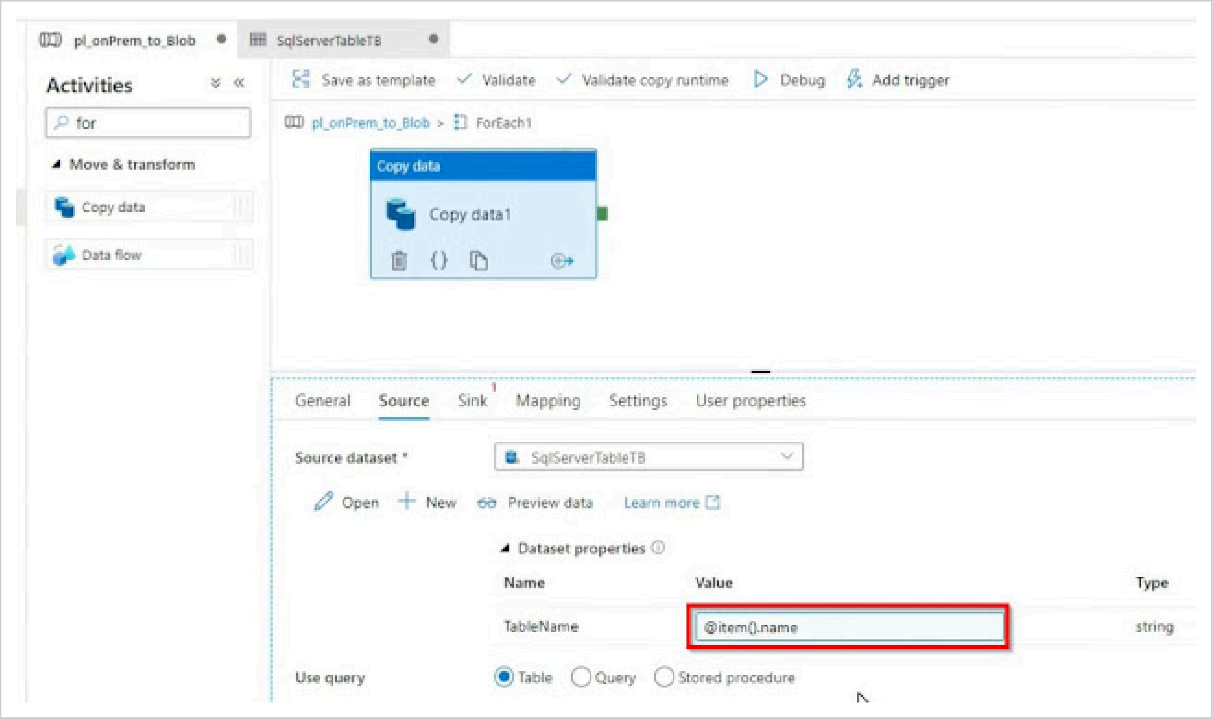
Linked service * InkTechbrothersITonprem

Integration runtime * selfhostedIR

Table dbo @dataset().TableName

☑ Edit

In the source tab provide the values.



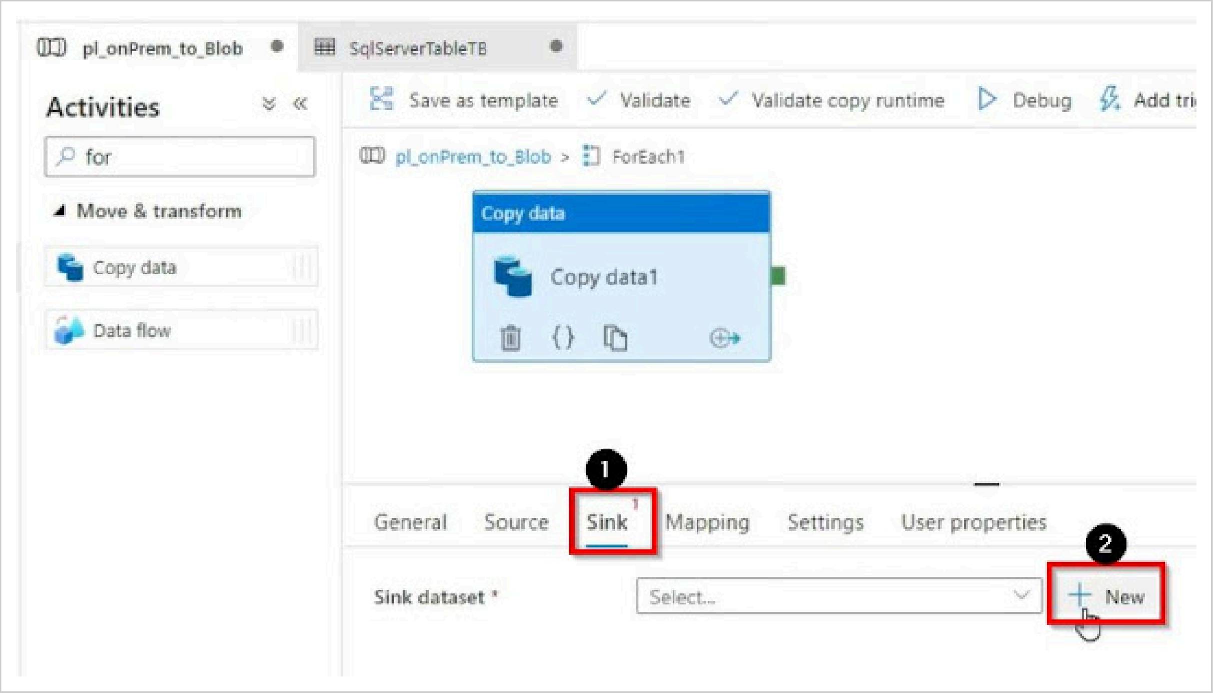
Source dataset * SqlServerTableTB

Dataset properties

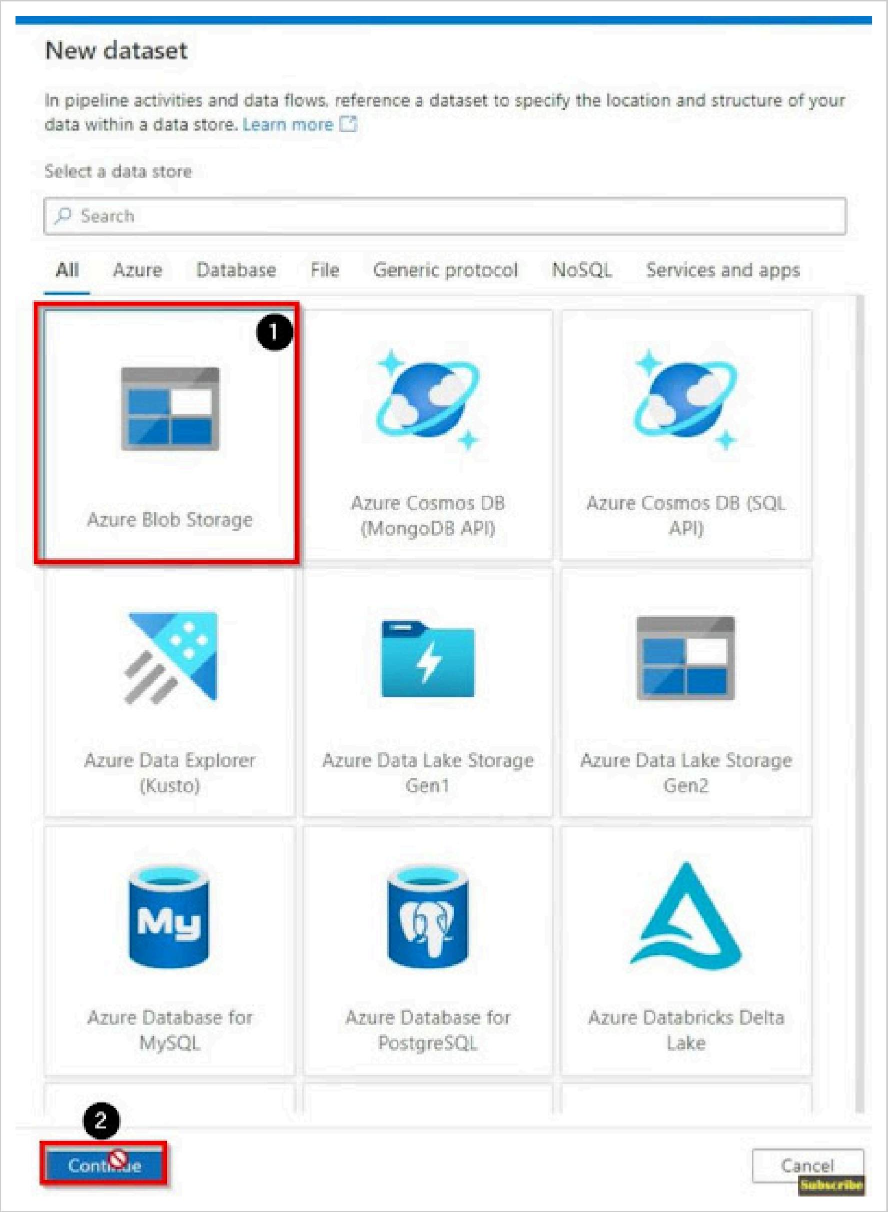
Name	Value	Type
TableName	@item().name	string

Use query ● Table ○ Query ○ Stored procedure

Next, go to the sink tab and click on + New button to create a new sink dataset.



Select Azure blob storage, then click on continue.



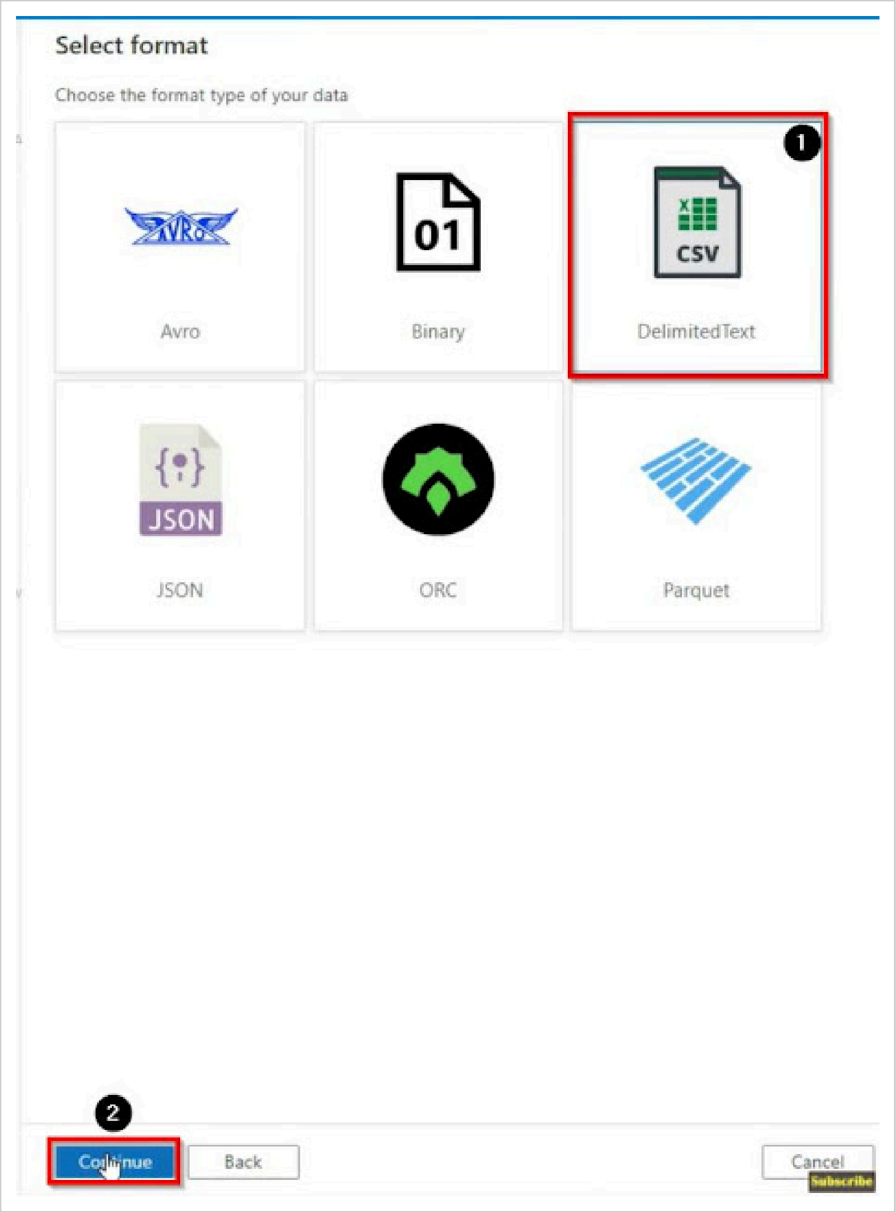
New dataset

In pipeline activities and data flows, reference a dataset to specify the location and structure of your data within a data store. Learn more

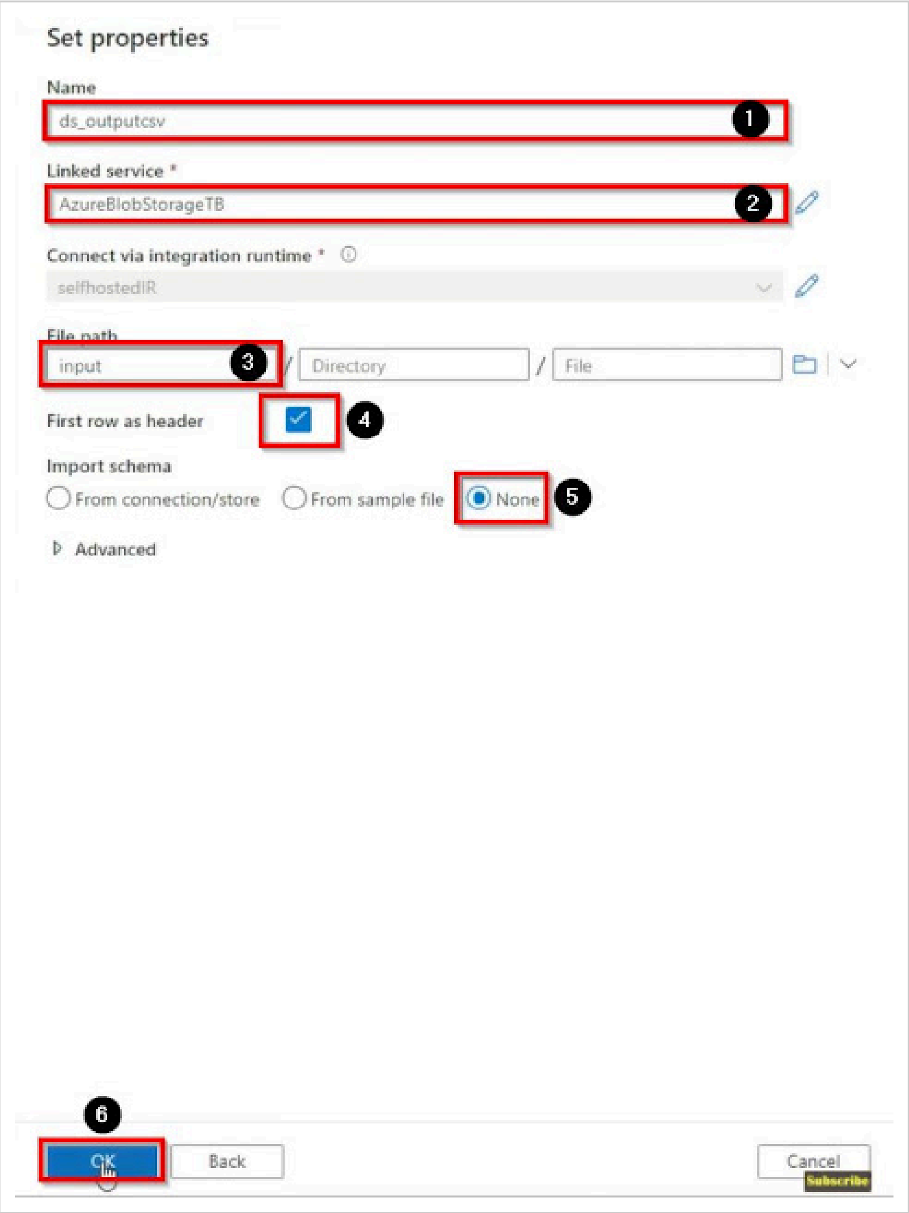
Select a data store

Search
All
Azure
Database
File
Generic protocol
NoSQL
Services and apps
1
Azure Blob Storage
Azure Cosmos DB (MongoDB API)
Azure Cosmos DB (SQL API)
Azure Data Explorer (Kusto)
Azure Data Lake Storage Gen1
Azure Data Lake Storage Gen2
Azure Database for MySQL
Azure Database for PostgreSQL
Azure Databricks Delta Lake
2
Continue
Cancel
Subscribe

Select CSV as format and click on continue.



Name your sink dataset, select the linked service, provide the folder path where our data will be copied, select the first row as header, select none for import schema, and then click on ok.



Set properties

Name

ds_outputcsv

Linked service *

AzureBlobStorageTB

Connect via integration runtime *

selfhostedIR

File path

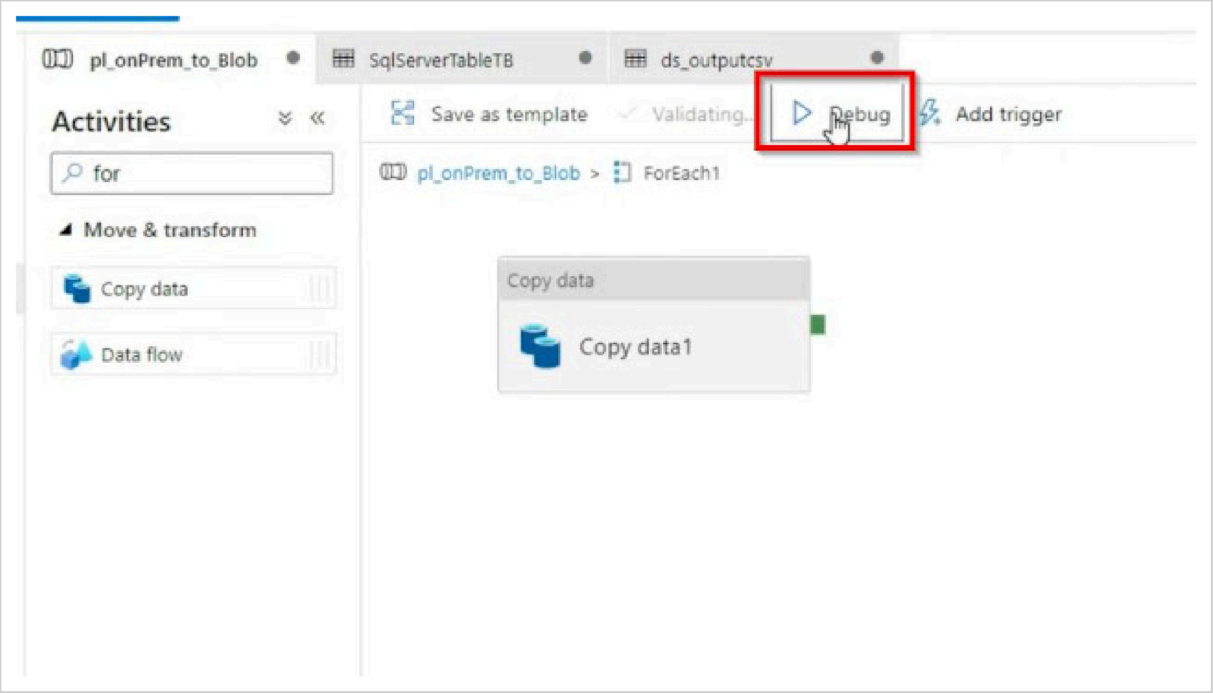
input / Directory / File

First row as header ☑

Import schema

○ From connection/store ○ From sample file ● None

Now, we are all set, and let's click on Debug.


pl_onPrem_to_Blob
SqlServerTableTB
ds_outputcsv
Activities
for
Move & transform
Copy data
Data flow
Save as template
Validating...
Debug
Add trigger
pl_onPrem_to_Blob > ForEach1
Copy data
Copy data1