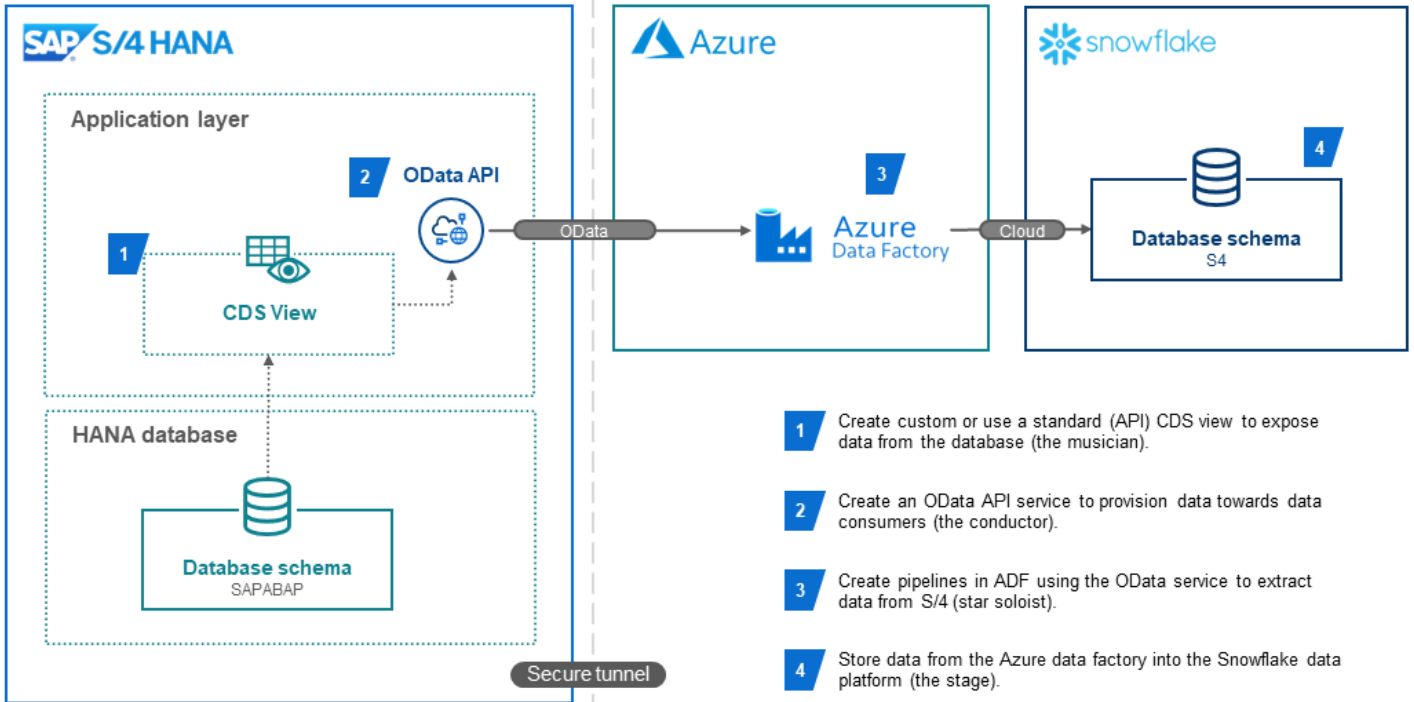


Snowflake



How-to easily enable your SAP S/4 data into Snowflake

WHAT ARE WE GOING TO DO?

Welcome to this practical step-by-step walkthrough, an integral part of our journey to seamlessly integrate SAP S/4 data into Snowflake using Azure Data Factory.

In an earlier blog post the high-level architecture was outlined (see page 1), illustrating how each component contributes to the orchestration of the data flow. Now, we take you by the hand, guiding you through all the technical details, ensuring that you can implement this architecture successfully.

WHAT ARE WE GOING TO DISCUSS?

This document consists of the following parts:

- Step 1: Expose data using CDS views
- Step 2: Create an OData API
- Step 3: Configure Snowflake
- Step 4: Configure Azure Data Factory
- Step 5: Build a data pipeline
- Step 6: Transfer the data from S/4 to Snowflake
- Step 7: Validate the data in Snowflake

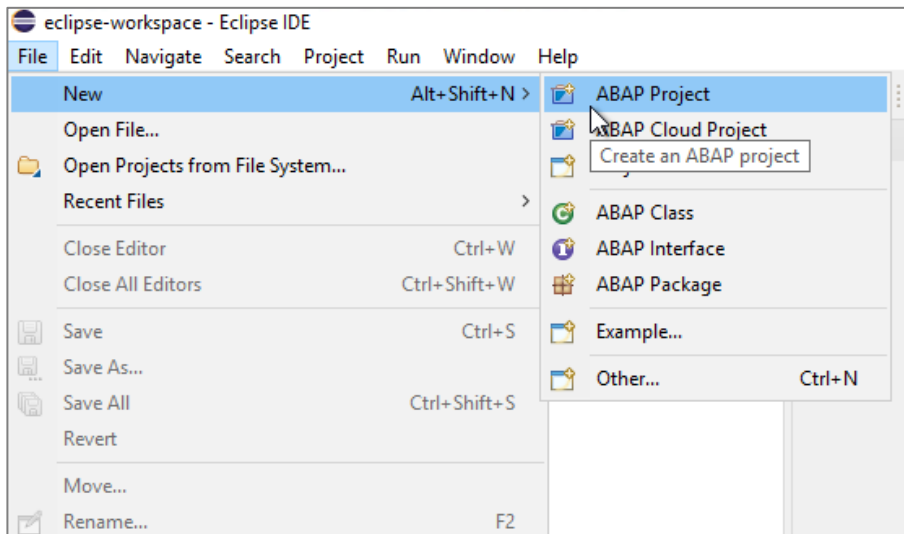
Note: This specific scenario assumes that the source data resides in a S/4 HANA on-premise server.

Step 1: Expose Data using CDS views

Step 1a: Launch Eclipse and create a new ABAP project

Launch the Eclipse and create a new ABAP project by:

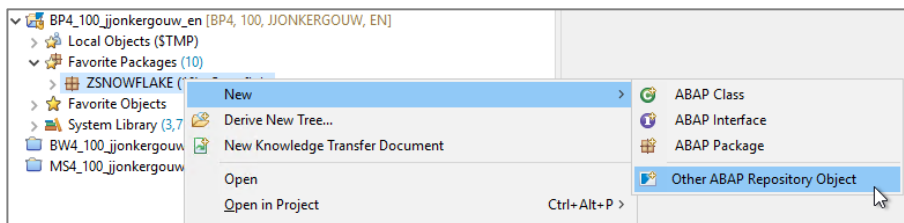
- Select "File" → "New" → "ABAP Project."
- Choose the right S/4 system and click "Next."
- Fill in your credentials and click "Finish."



Step 1c: Exposing Data Using CDS Views

In the ABAP Project go to the package in which you want to create the CDS view.

- Right click the package and choose "New" → "Other ABAP Repository Object."
- Choose "Data Definition" and click "Next."
- Define the package and name for your CDS view.
- Click "Finish."



Step 1d: Define CDS View Metadata

In the newly created CDS view, define the metadata for your view, including entity names and field definitions. This type of view type is required to enable it for extraction. Once your CDS view is defined, activate it.

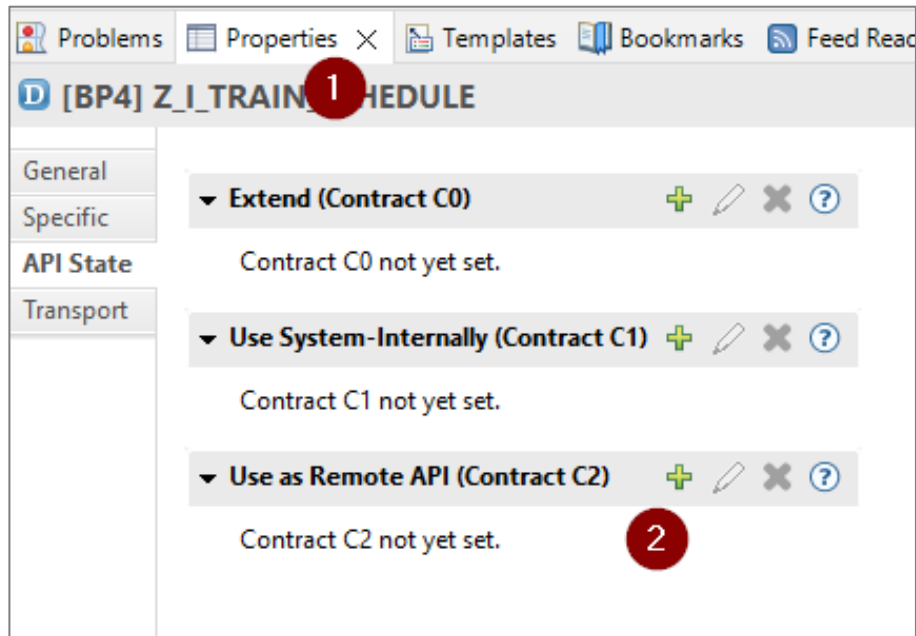
Below an example of a CDS with an additional "OData" annotation based on the table SFLIGHT.

```
1 @AbapCatalog.sqlViewName: 'ZITRAINSCHEDULE'  
2 @AbapCatalog.compiler.compareFilter: true  
3 @AccessControl.authorizationCheck: #NOT_REQUIRED  
4 @OData.publish: true  
5 define view Z_I_TRAIN_SCHEDULE as select from sflight  
6 {  
7   key sflight.carrid as TrainOperator,  
8   key sflight.connid as TrainConnection,  
9   key sflight.fldate as TrainDate,  
10  sflight.price as Price,  
11  sflight.currency as Currency,  
12  sflight.planetype as TrainType,  
13  sflight.seatsmax as SeatsMax,  
14  sflight.seatsocc as SeatSocc,  
15  sflight.paymentsum as PaymentSum,  
16  sflight.seatsmax_b as SeatsMaxB,  
17  sflight.seatsocc_b as SeatSoccB,  
18  sflight.seatsmax_f as SeatsMaxF,  
19  sflight.seatsocc_f as SeatSoccF  
20 }
```

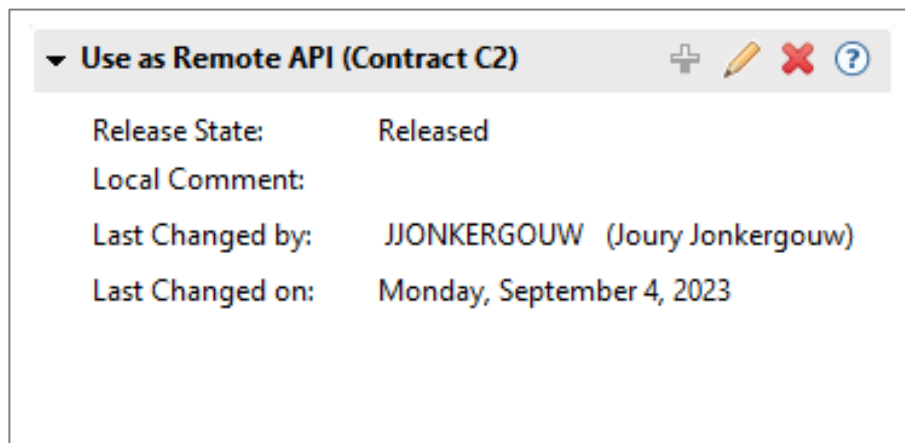
Notice that SFLIGHT is renamed to TRAIN_SCHEDULE as transportation by train is environmental friendly compared to travelling by plane 😊

Step 1e: Release CDS view

By release the CDS view using a C3 contract you enable it for external consumption via an API. If you skip this activity you are unable to publish this CDS view as an API. You can enable this C3 contract via the properties window.



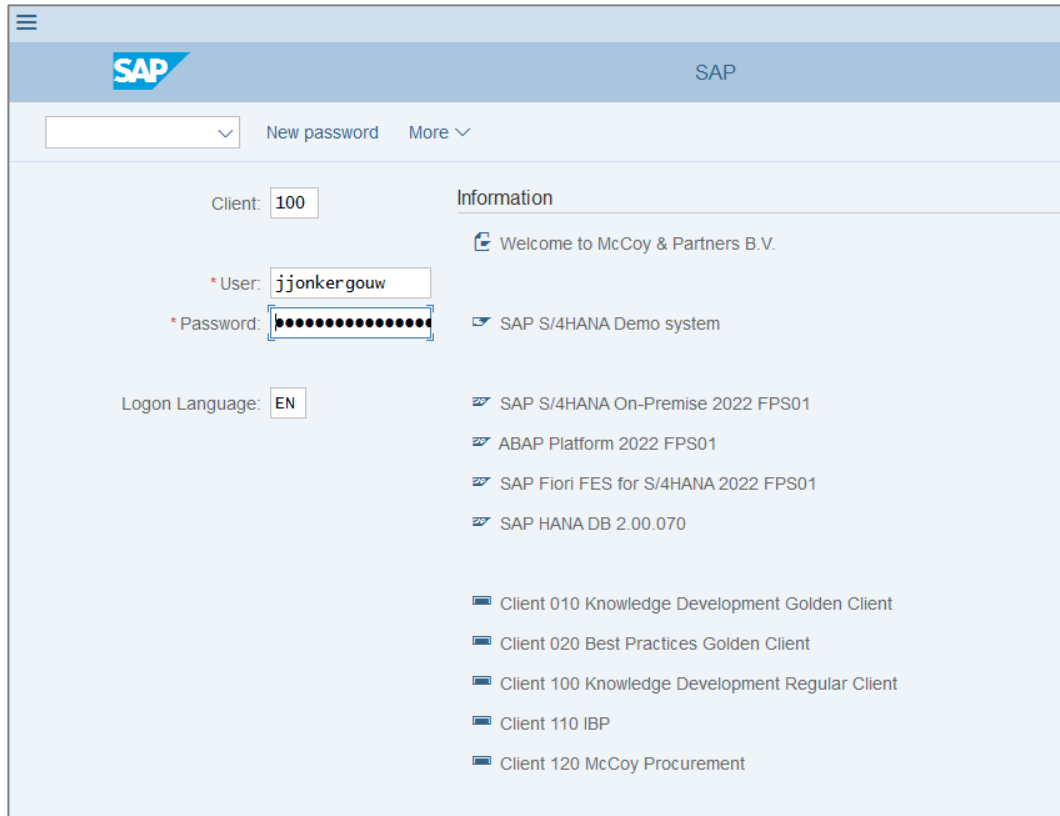
Once the contract has been added it will be shown in the properties tab.



Step 2: Create an OData API

Step 2a: Access SAP S/4 HANA via SAP GUI

Log in to your SAP S/4 HANA system using the appropriate credentials.



The screenshot displays the SAP login interface. At the top, there is a blue header with the SAP logo on the left and the word 'SAP' on the right. Below the header, there is a navigation bar with a dropdown menu, 'New password', and 'More'. The main content area is divided into two columns. The left column contains the login form with the following fields: 'Client: 100', '* User: jjonkergouw', '* Password: [masked]', and 'Logon Language: EN'. The right column is titled 'Information' and contains a list of system details, including 'Welcome to McCoy & Partners B.V.', 'SAP S/4HANA Demo system', and several SAP S/4HANA On-Premise 2022 FPS01 components (ABAP Platform, SAP Fiori FES, and SAP HANA DB). Below these are several client-specific entries, such as 'Client 010 Knowledge Development Golden Client', 'Client 020 Best Practices Golden Client', 'Client 100 Knowledge Development Regular Client', 'Client 110 IBP', and 'Client 120 McCoy Procurement'.

Step 2b: Enable OData Services

Generate runtime artifacts for the OData service using transaction code "/IWFND/MAINT_SERVICE" in SAP Gateway. Search for the CDS view and click on "Add Selected Services"

Filter

System Alias: LOCAL

Technical Service Name:

External Service Name: Z_I_TRAIN_SCHEDULE_CDS

Co-Dep

Select Backend Services

⊕ Add Selected Services

Type	Technical Service Name	Versi...	Service Description
<input type="checkbox"/>	Z_I_TRAIN_SCHEDULE_CDS	1	

Assign "\$TMP" as Package Assignment and click on "Continue"

Add Service

Service

* Technical Service Name: Z_I_TRAIN_SCHEDULE_CDS

Service Version: 1

Description:

External Service Name: Z_I_TRAIN_SCHEDULE_CDS

Namespace:

External Mapping ID:

External Data Source Type: C

Model

Technical Model Name: Z_I_TRAIN_SCHEDULE_CDS

Model Version: 1

Creation Information

Package Assignment: \$TMP

Local Object

ICF Node

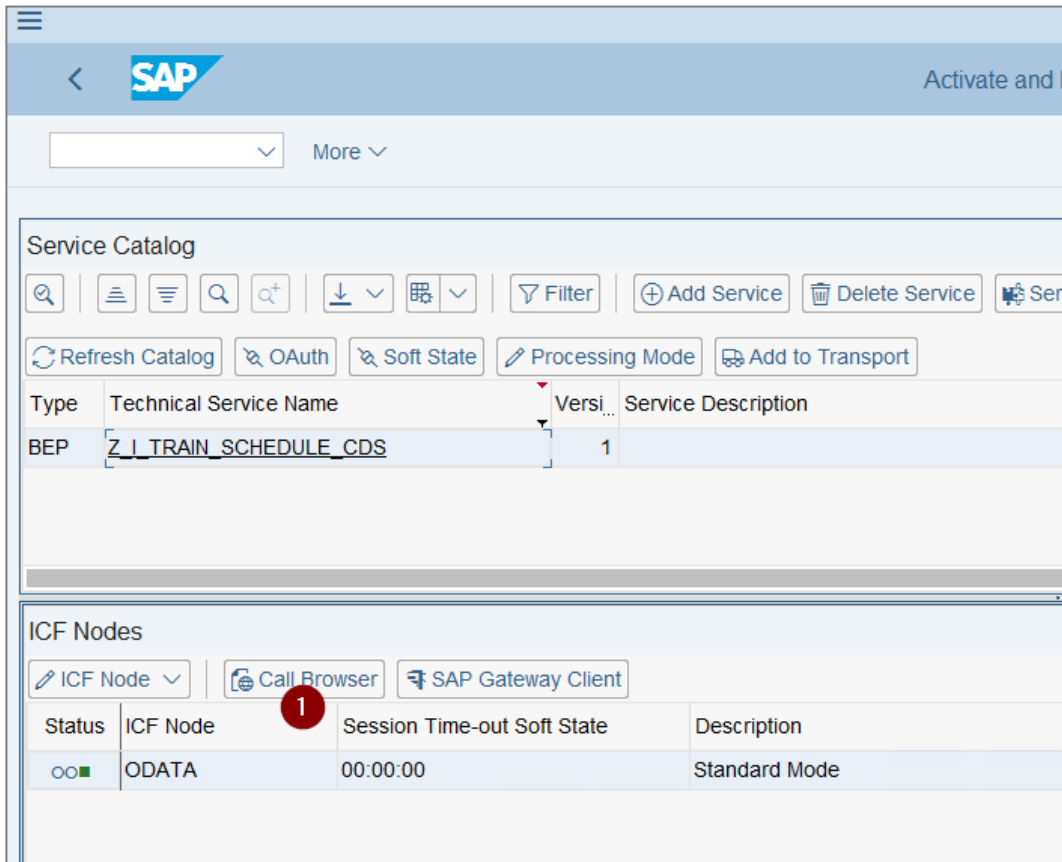
Standard Mode None

Set Current Client as Default Client in ICF Node

Continue Cancel

Step 2c: Test OData Service

Once created click on "Call Browser" to verify if API is working as expected. Note down the URL in the address bar.

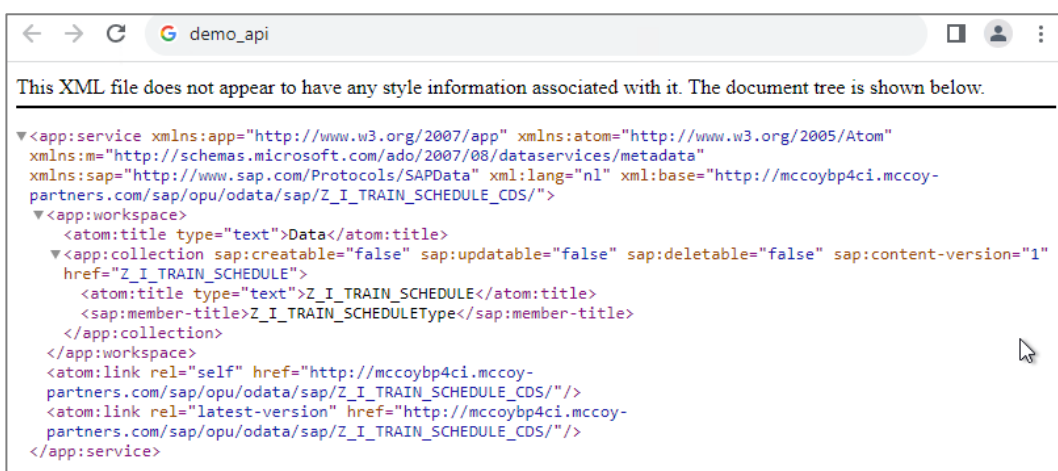


The screenshot shows the SAP Service Catalog and ICF Nodes configuration. The Service Catalog table lists the service Z_I_TRAIN_SCHEDULE_CDS with version 1. The ICF Nodes table shows the ODATA node in Standard Mode.

Type	Technical Service Name	Versi...	Service Description
BEP	Z_I_TRAIN_SCHEDULE_CDS	1	

Status	ICF Node	Session Time-out Soft State	Description
oo	ODATA	00:00:00	Standard Mode

Congratulations your OData API cannot be used for consumption.



The screenshot shows a browser window displaying an XML document tree. The XML content is as follows:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<app:service xmlns:app="http://www.w3.org/2007/app" xmlns:atom="http://www.w3.org/2005/Atom"
xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata"
xmlns:sap="http://www.sap.com/Protocols/SAPData" xml:lang="nl" xml:base="http://mccoybp4ci.mccoy-
partners.com/sap/opu/odata/sap/Z_I_TRAIN_SCHEDULE_CDS/">
  <app:workspace>
    <atom:title type="text">Data</atom:title>
    <app:collection sap:creatable="false" sap:updatable="false" sap:deletable="false" sap:content-version="1"
href="Z_I_TRAIN_SCHEDULE">
      <atom:title type="text">Z_I_TRAIN_SCHEDULE</atom:title>
      <sap:member-title>Z_I_TRAIN_SCHEDULEType</sap:member-title>
    </app:collection>
  </app:workspace>
  <atom:link rel="self" href="http://mccoybp4ci.mccoy-
partners.com/sap/opu/odata/sap/Z_I_TRAIN_SCHEDULE_CDS/">
  <atom:link rel="latest-version" href="http://mccoybp4ci.mccoy-
partners.com/sap/opu/odata/sap/Z_I_TRAIN_SCHEDULE_CDS/">
</app:service>
```


Step 3: Configure Snowflake staging table

It is necessary to create tables before creating the data pipeline in Azure Data Factory. Therefore the following CREATE TABLE statement is executed in Snowflake.

```
MCCOY.PUBLIC ▾ Settings ▾  
1 CREATE TABLE MCCOY.PUBLIC.T_TRAIN_SCHEDULE  
2 (  
3     trainoperator      varchar(1000),  
4     trainconnection    varchar(1000),  
5     traindate          datetime,  
6     price              decimal(16,2),  
7     currency           varchar(1000),  
8     traintype          varchar(1000),  
9     seatsmax           int,  
10    Seatsocc           int,  
11    paymentsum         decimal(16,2),  
12    seatsmaxb          int,  
13    seatsoccb         int,  
14    seatsmaxf          int,  
15    seatsoccf         int  
16 )  
17
```

Stap 4: Configure Azure Data Factory

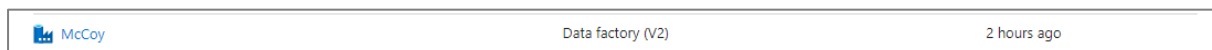
Step 4a: Access Azure Portal

Log in to the Azure Portal with your credentials.



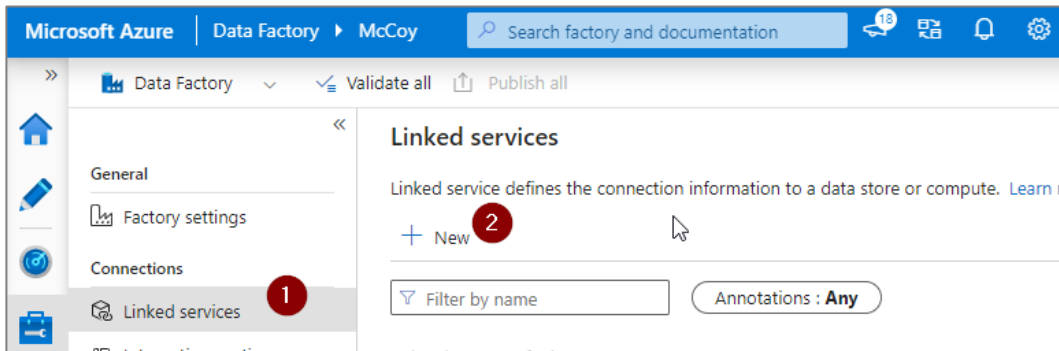
Step 4b: Create an Azure Data Factory (optional)

If you haven't already, create a new Azure Data Factory instance by following the Azure Portal's guided process.

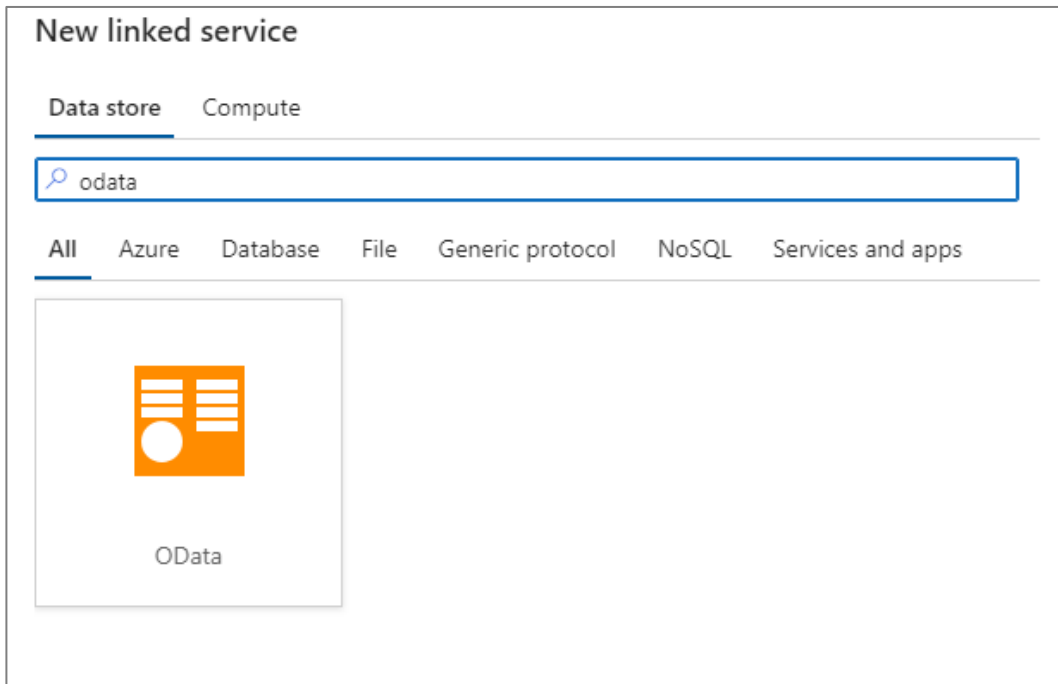


Step 4c: Add a Linked Service for SAP S/4

Within your Azure Data Factory, add a linked service for SAP S/4, providing the necessary connection details towards the OData API.





In the "New Linked Service" window, search for or select the "OData" linked service type.



Fill in the required fields and test the connection.

Edit linked service


 OData [Learn more](#) 

Name *

Description

Connect via integration runtime * ⓘ

Service URL *

 Information will be sent to the URL specified. Please ensure you trust the URL entered.

Authentication type *

User name *

Password * Azure Key Vault

Auth headers ⓘ
[+ New](#)

Annotations
[+ New](#)

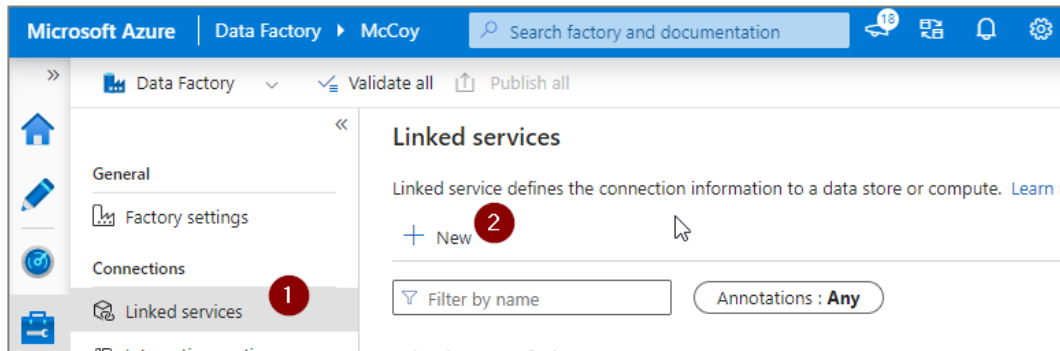
[> Parameters](#)

[> Advanced](#) ⓘ

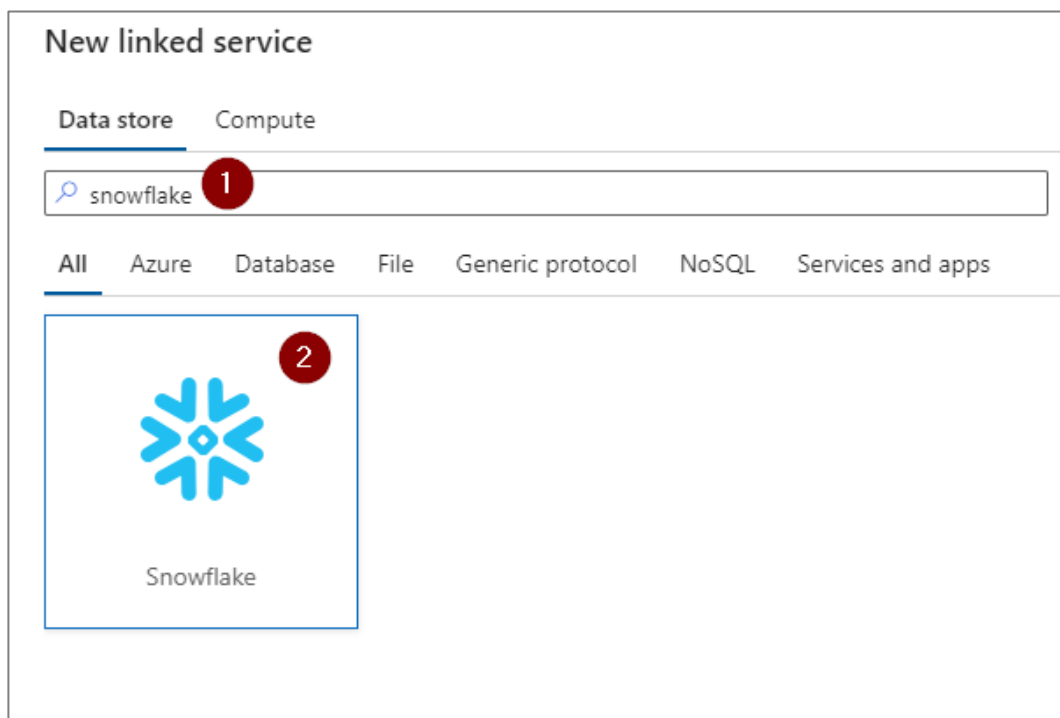
It's advisable to test the connection to ensure that the provided settings are correct and that ADF can connect to the OData API. Once the configuration is complete and the connection test is successful, save the linked service.

Step 4e: Add a Linked Service for Snowflake

In your Azure Data Factory, add a linked service for Snowflake. Provide the Snowflake connection details, including credentials.



In the "New Linked Service" window, search for or select the "Snowflake" linked service type.



Fill in the required fields and test the connection.

Edit linked service

Snowflake [Learn more](#)

Name *
Snowflake1

Description

Connect via integration runtime *
 integrationRuntime3

Connection string [Azure Key Vault](#)

Account name *
lvzifpu-fq92174 **1**

User name *
MCCOY

Password [Azure Key Vault](#)

Password *

Database *
MCCOY

Warehouse *
COMPUTE_WH

Role

Additional connection properties
[+ New](#)

Annotations
[+ New](#)
> Parameters
> Advanced

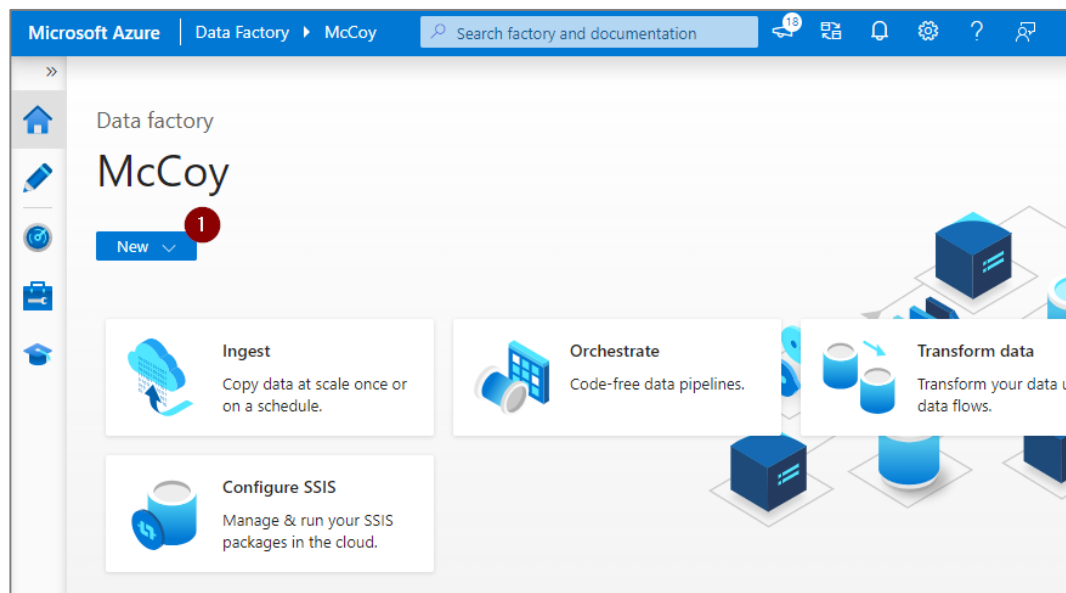
2 Connection successful [Test connection](#)

It's advisable to test the connection to ensure that the provided settings are correct and that ADF can connect to Snowflake. Once the configuration is complete and the connection test is successful, save the linked service.

Step 5: Build a data pipeline

Step 5a: Create a new “Pipeline”

In your Azure Data Factory, create a new pipeline specifically to copy data from SAP S/4 to Snowflake.



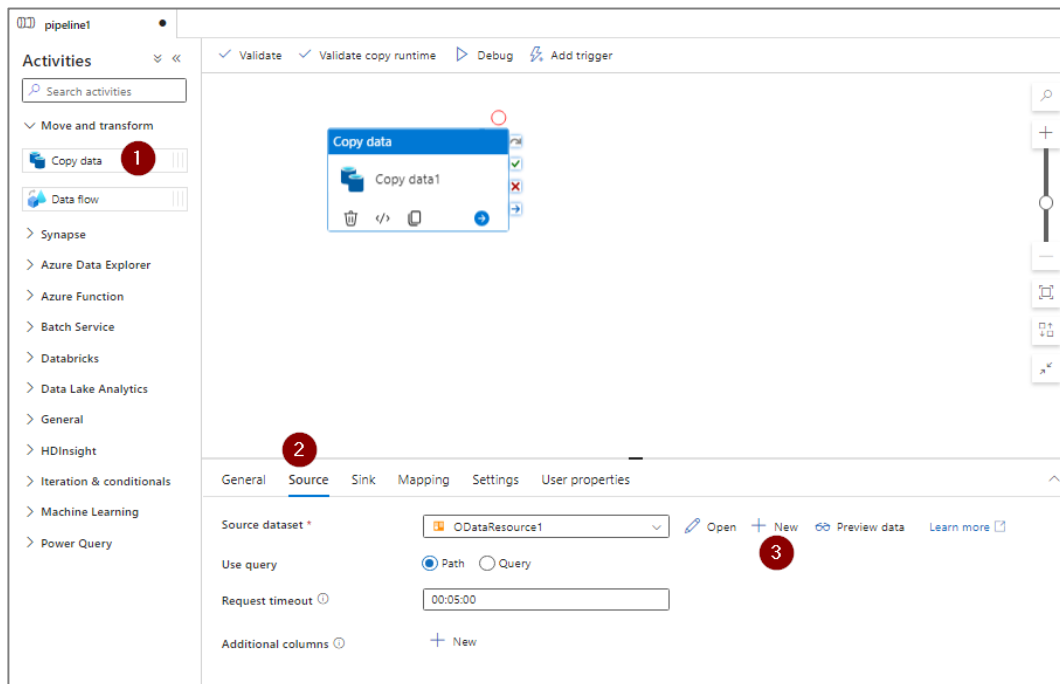
Step 5b: Add “Copy Data” activity

Within the pipeline, add activities that define the data extraction and transformation process. This may include data source, data transformation, and data sink activities.

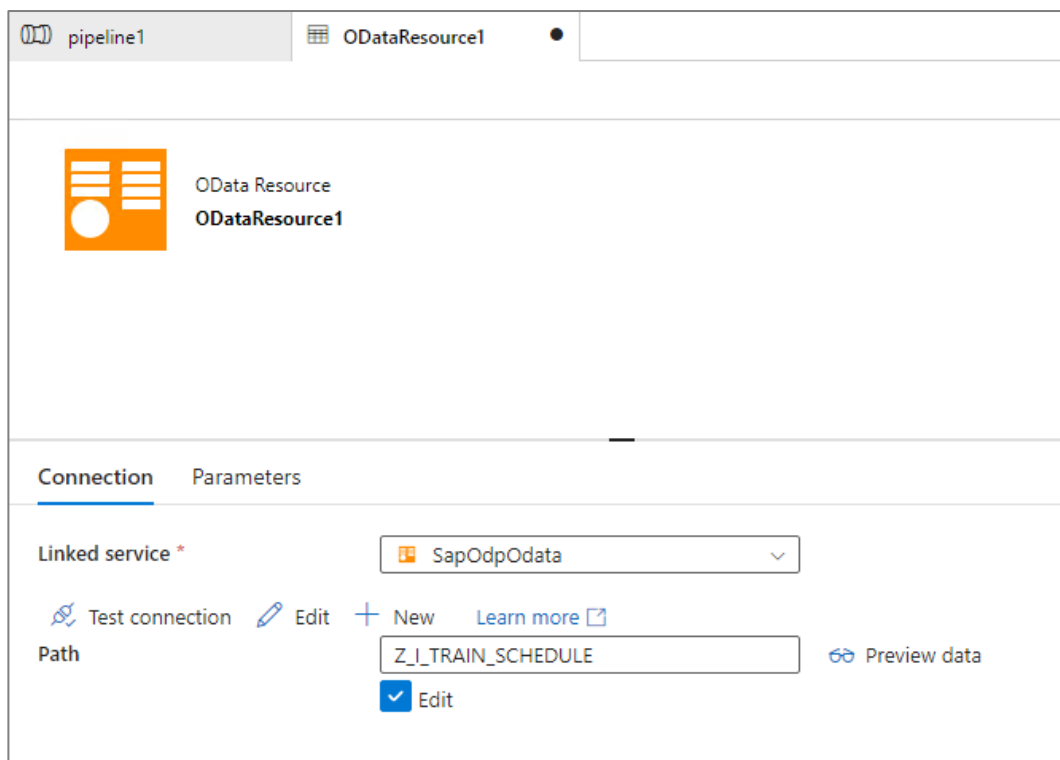
For this specific purpose a simple 1:1 is sufficient and therefore only add a “Copy Data” activity to the pipeline.

Step 5c: Define source data

Once added click on the "Copy data" activity and choose your "Odata" source by clicking on "New".

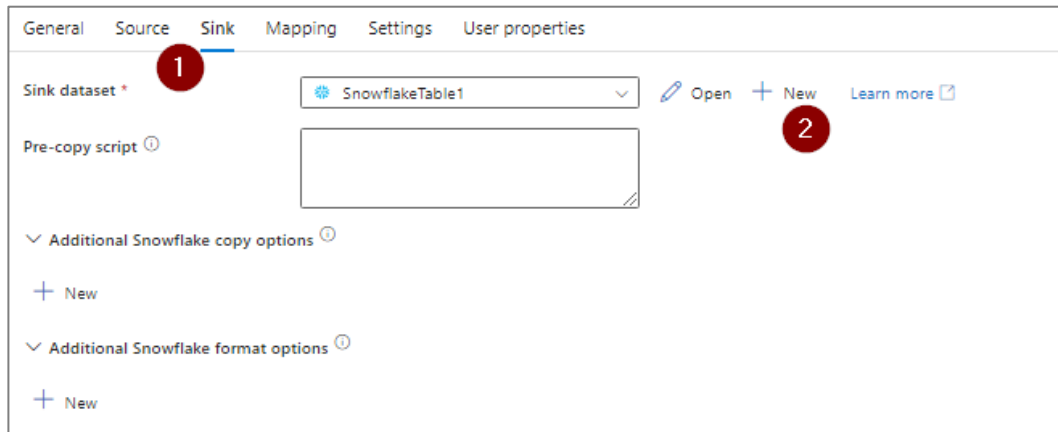


Now select your "Linked Services" and fill in your API technical name. Test the connection by previewing the data.

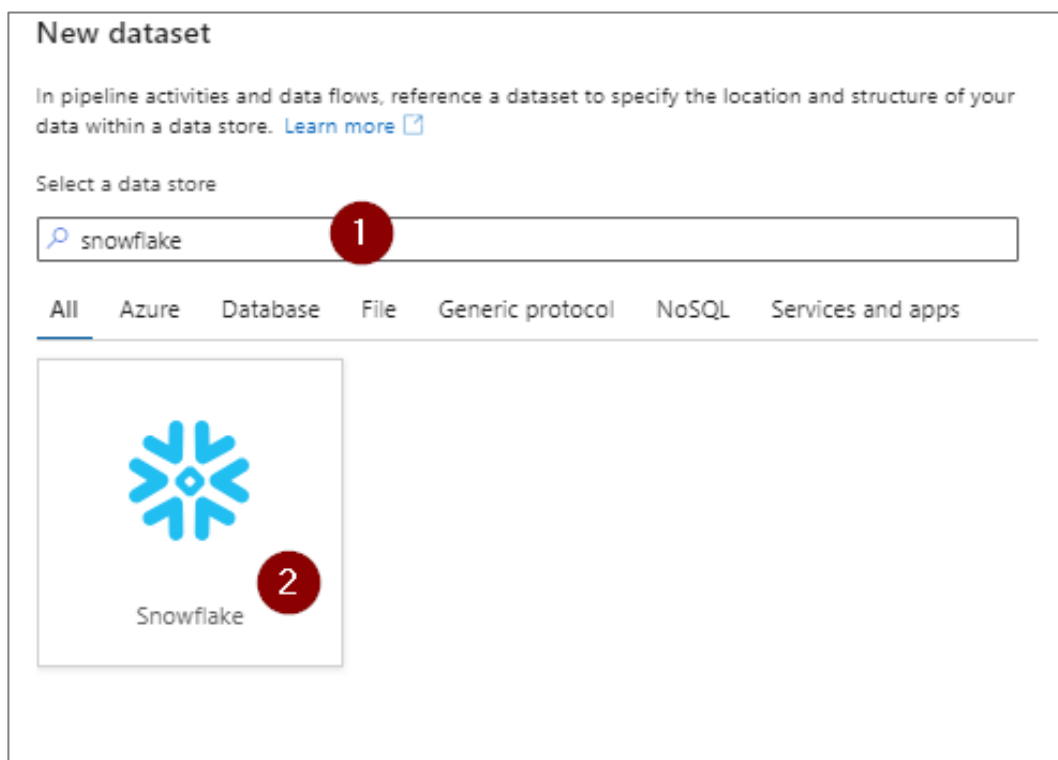


Step 5d: Define target data

In the "Sink" tab of the "Copy Data" activity, configure the sink dataset. This is where the data will be copied to. Similar to the source, specify the sink data store and provide connection details and credentials. For this purpose we use the Snowflake "Linked Service"



Search for "Snowflake" and select the option.



Define the target table of Snowflake as defined in Step 3

Set properties

Name
SnowflakeTable2

Linked service * **1**
Snowflake1

Connect via integration runtime * ⓘ
integrationRuntime3

Table name
MCCOY , T_TRAIN_SCHEDULE **2**

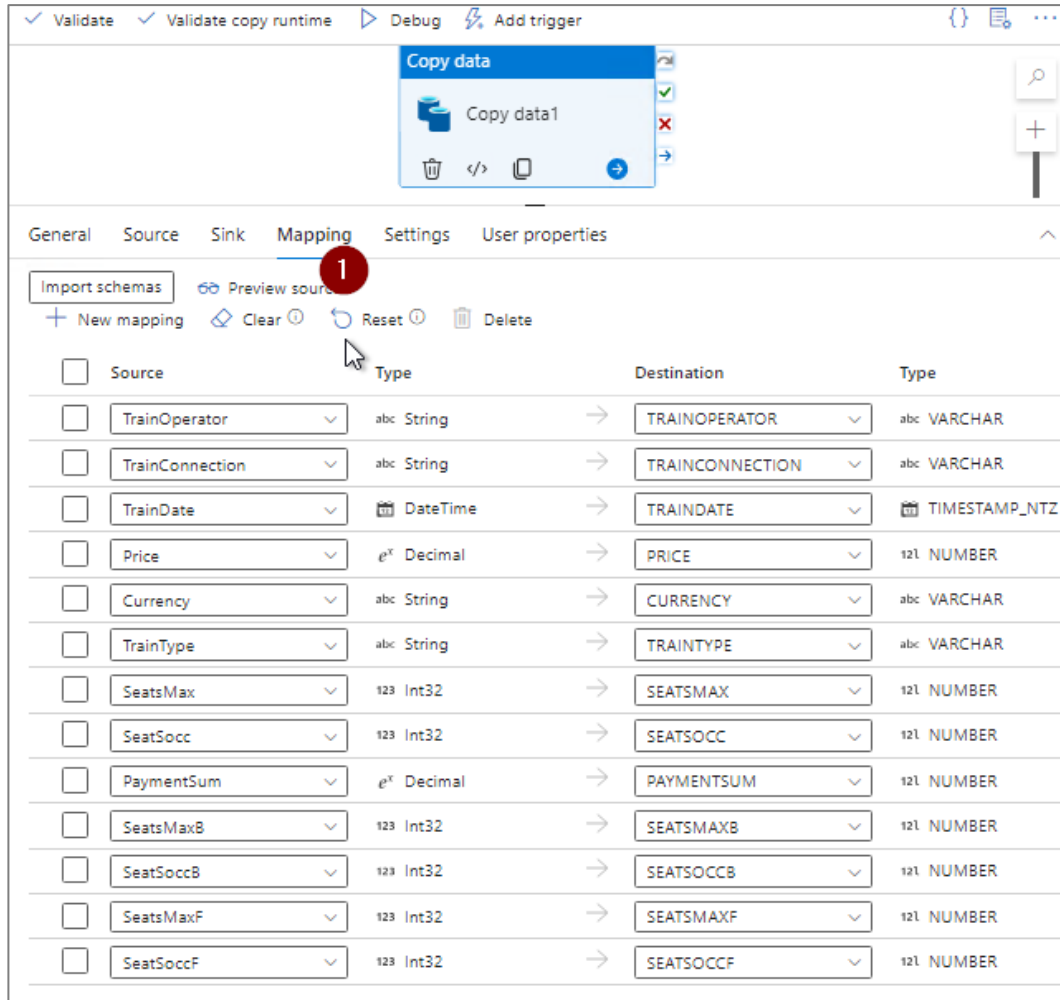
Edit

Import schema
 From connection/store None

> Advanced

Step 5e: Mapping and transformation

In the "Mapping" tab, you can define transformations or mapping rules if the source and sink schemas are different. This step is optional but crucial for data transformation scenarios.



The screenshot shows the 'Copy data' tool interface. At the top, there are buttons for 'Validate', 'Validate copy runtime', 'Debug', and 'Add trigger'. Below these is a 'Copy data' dropdown menu with 'Copy data1' selected. The main interface has tabs for 'General', 'Source', 'Sink', 'Mapping', 'Settings', and 'User properties'. The 'Mapping' tab is active, showing a table with columns for 'Source', 'Type', 'Destination', and 'Type'. A red circle with the number '1' is placed over the 'Preview source' button. Below the table, there are buttons for 'New mapping', 'Clear', 'Reset', and 'Delete'.

Source	Type	Destination	Type
<input type="checkbox"/> TrainOperator	abc String	→ TRAINOPERATOR	abc VARCHAR
<input type="checkbox"/> TrainConnection	abc String	→ TRAINCONNECTION	abc VARCHAR
<input type="checkbox"/> TrainDate	📅 DateTime	→ TRAINDATE	📅 TIMESTAMP_NTZ
<input type="checkbox"/> Price	e ^x Decimal	→ PRICE	121 NUMBER
<input type="checkbox"/> Currency	abc String	→ CURRENCY	abc VARCHAR
<input type="checkbox"/> TrainType	abc String	→ TRRAINTYPE	abc VARCHAR
<input type="checkbox"/> SeatsMax	123 Int32	→ SEATSMAX	121 NUMBER
<input type="checkbox"/> SeatSocc	123 Int32	→ SEATSOCC	121 NUMBER
<input type="checkbox"/> PaymentSum	e ^x Decimal	→ PAYMENTSUM	121 NUMBER
<input type="checkbox"/> SeatsMaxB	123 Int32	→ SEATSMAXB	121 NUMBER
<input type="checkbox"/> SeatSoccB	123 Int32	→ SEATSOCCB	121 NUMBER
<input type="checkbox"/> SeatsMaxF	123 Int32	→ SEATSMAXF	121 NUMBER
<input type="checkbox"/> SeatSoccF	123 Int32	→ SEATSOCCF	121 NUMBER

Step 5f: Additional settings

In order to stage towards Snowflake it is required to enable "Staging" and adding a "Staging account",

General Source Sink Mapping **Settings** ...

1 You will be charged # of used DIUs * copy duration * \$0.25/DIU-hour. Local currency and separate discounting may apply per subscription type. [Learn more](#)

Maximum data integration unit ⓘ
Auto Edit

Degree of copy parallelism ⓘ
 Edit

Fault tolerance ⓘ

Enable logging ⓘ

Enable staging ⓘ **2**

Staging settings

Staging account linked service * ⓘ **3**
AzureBlobStorage1

[Test connection](#) [Edit](#) [New](#)

Connection successful

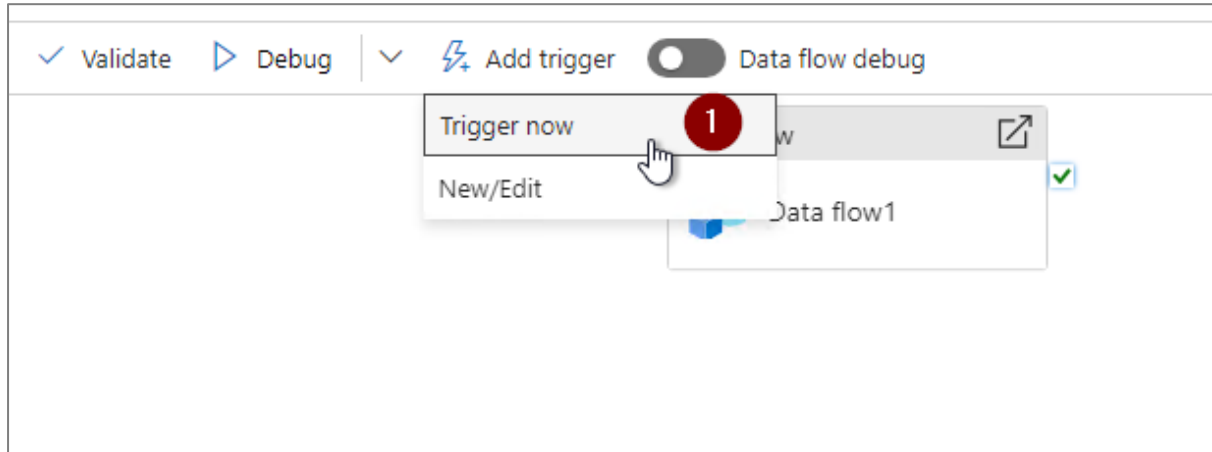
Storage Path ⓘ

[Browse](#)

Enable Compression ⓘ

Step 6: Transfer the data from S/4 to Snowflake

Now the complete pipeline is ready to be executed. By clicking on "Trigger now" the data transfer process is initiated.



Monitor the pipeline's execution and validate the data transferred from SAP S/4 to Snowflake.

Pipeline runs

Triggered Debug Rerun Cancel options Refresh Edit columns List Gantt

Filter by run ID or name Local time: Last 24 hours Pipeline name: All Status: All Runs: Latest runs Copy filters Export to CSV

Triggered by: All Add filter

Showing 1 - 1 items Last refreshed 0 minutes ago

<input type="checkbox"/>	Pipeline name ↑↓	Run start ↑↓	Run end ↑↓	Duration	Triggered by	Status ↑↓	Run	P
<input type="checkbox"/>	pipeline1	9/4/2023, 11:59:43 AM	--	52s	Manual trigger	In progress	Original	

Step 7: Validate the data in Snowflake

Now it's time to finally validate the data in Snowflake. This can be done by executing a SQL select statement directly in Snowflake on the target table.

The screenshot displays the Snowflake query results interface. At the top, the SQL query is shown: `SELECT * FROM MCCOY.PUBLIC.T_TRAIN_SCHEDULE`. Below the query, the results are presented in a table with the following columns: TRAINOPERATOR, TRAINCONNECTION, TRINDATE, PRICE, CURRENCY, TRRAINTYPE, SEATSMAX, and SEATSC. The table contains 16 rows of data. On the right side, the Query Details panel shows a query duration of 38ms and 356 rows returned. The Query ID is 01aec377-0102-9dbf-0... The results are also displayed in a chart view.

	TRAINOPERATOR	TRAINCONNECTION	TRINDATE	PRICE	CURRENCY	TRRAINTYPE	SEATSMAX	SEATSC
1	AA	0017	2021-10-07 00:00:00.000	422.94	USD	747-400	385	
2	AA	0017	2021-11-08 00:00:00.000	422.94	USD	747-400	385	
3	AA	0017	2021-12-10 00:00:00.000	422.94	USD	747-400	385	
4	AA	0017	2022-01-11 00:00:00.000	422.94	USD	747-400	385	
5	AA	0017	2022-02-12 00:00:00.000	422.94	USD	747-400	385	
6	AA	0017	2022-03-16 00:00:00.000	422.94	USD	747-400	385	
7	AA	0017	2022-04-17 00:00:00.000	422.94	USD	747-400	385	
8	AA	0017	2022-05-19 00:00:00.000	422.94	USD	747-400	385	
9	AA	0017	2022-06-20 00:00:00.000	422.94	USD	747-400	385	
10	AA	0017	2022-07-22 00:00:00.000	422.94	USD	747-400	385	
11	AA	0017	2022-08-23 00:00:00.000	422.94	USD	747-400	385	
12	AA	0017	2022-09-24 00:00:00.000	422.94	USD	747-400	385	
13	AA	0017	2022-10-26 00:00:00.000	422.94	USD	747-400	385	
14	AA	0064	2021-10-09 00:00:00.000	422.94	USD	A340-600	330	
15	AA	0064	2021-11-10 00:00:00.000	422.94	USD	A340-600	330	
16	AA	0064	2021-12-12 00:00:00.000	422.94	USD	A340-600	330	

Congratulations, your data is loaded into Snowflake!

SUMMARY

As we conclude this step-by-step walkthrough, you've witnessed how a well-orchestrated combination of Core Data Services (CDS) views, OData API's, Azure Data Factory pipelines and Snowflake database connectivity can seamlessly work together to easily expose SAP S/4 data into the powerful data platform Snowflake.

The architecture we described in our blog provided the vision and through these practical steps, we've made it a reality. Remember, this journey may vary for each organization, and challenges might arise along the way.

WANT TO KNOW MORE?

If you have any questions or need further guidance, don't hesitate to reach out. Stay tuned for more insights and solutions on <https://www.mccoy-partners.com/>.

McCOY