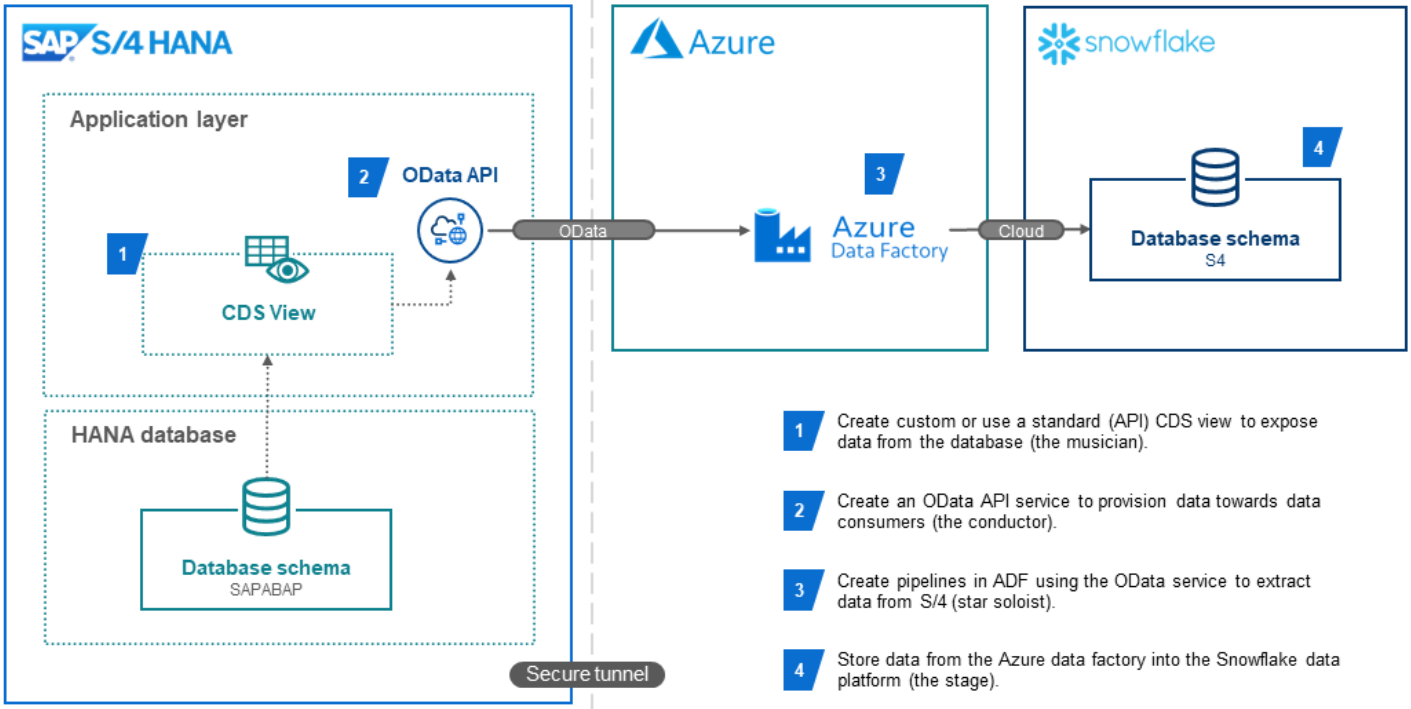
BU – BUSINESS INTELLIGENCE

Snowflake



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

McCOY

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

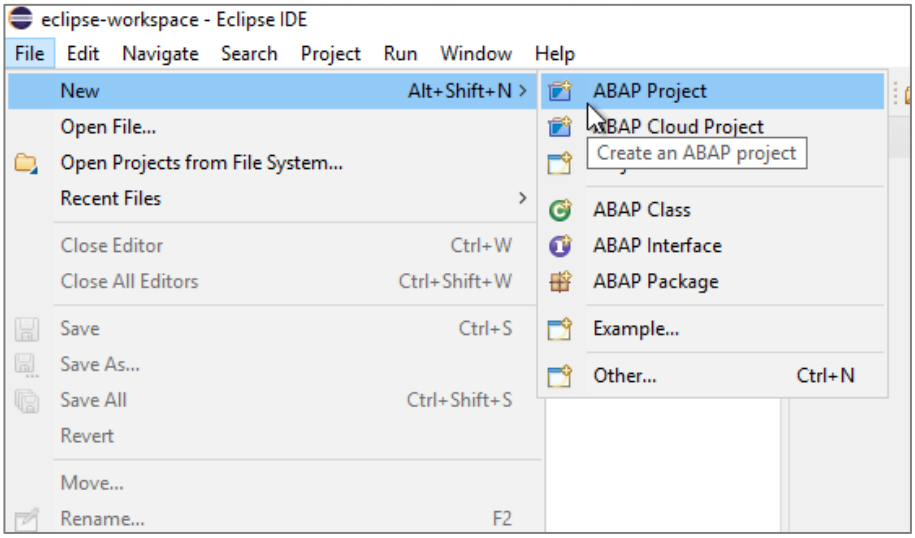
Note: This specific scenario assumes that the source data resides in a S/4 HANA on-premise server.

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:

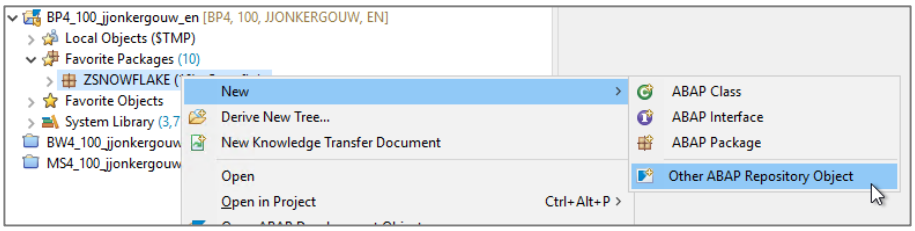
- Select "File" → "New" → "ABAP Project."
- Choose the right S/4 system and click "Next."
- Fill in your credentials and click "Finish."



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.

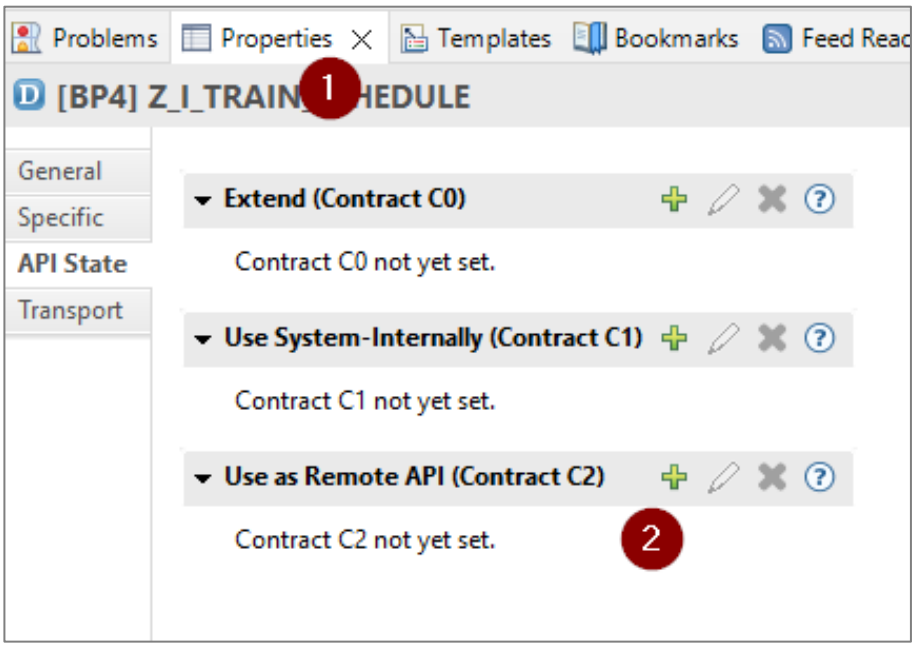
Below an example of a CDS with an additional “OData” annotation based on the table SFLIGHT.

```
@AbapCatalog.sqlViewName: 'ZITRAINSCHEDULE'
@AbapCatalog.compiler.compareFilter: true
@AccessControl.authorizationCheck: #NOT_REQUIRED
@OData.publish: true
define view Z_I_TRAIN_SCHEDULE as select from sflight
{
  key sflight.carrid as TrainOperator,
  key sflight.connid as TrainConnection,
  key sflight.fldate as TrainDate,
      sflight.price as Price,
      sflight.currency as Currency,
      sflight.planetype as TrainType,
      sflight.seatsmax as SeatsMax,
      sflight.seatsocc as SeatSocc,
      sflight.paymentsum as PaymentSum,
      sflight.seatsmax_b as SeatsMaxB,
      sflight.seatsocc_b as SeatSoccB,
      sflight.seatsmax_f as SeatsMaxF,
      sflight.seatsocc_f as SeatSoccF
}
```

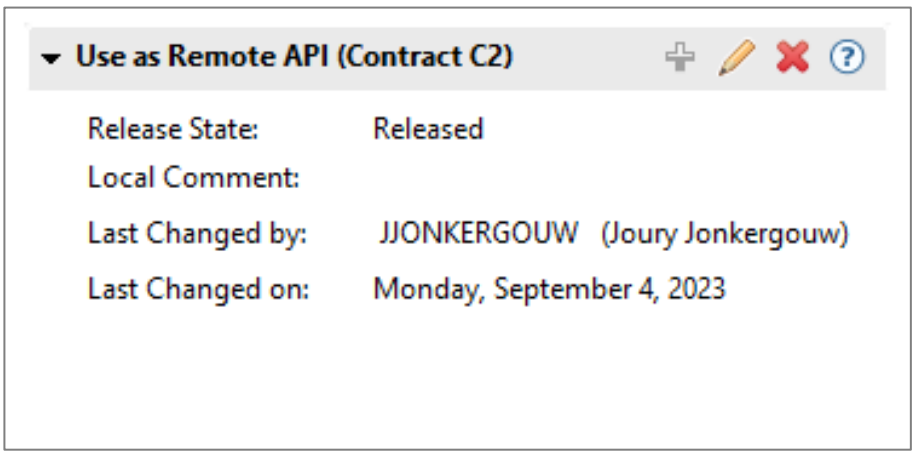
Notice that SFLIGHT is renamed to TRAIN_SCHEDULE as transportation by train is environmental friendly compared to travelling by plane 😉

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.



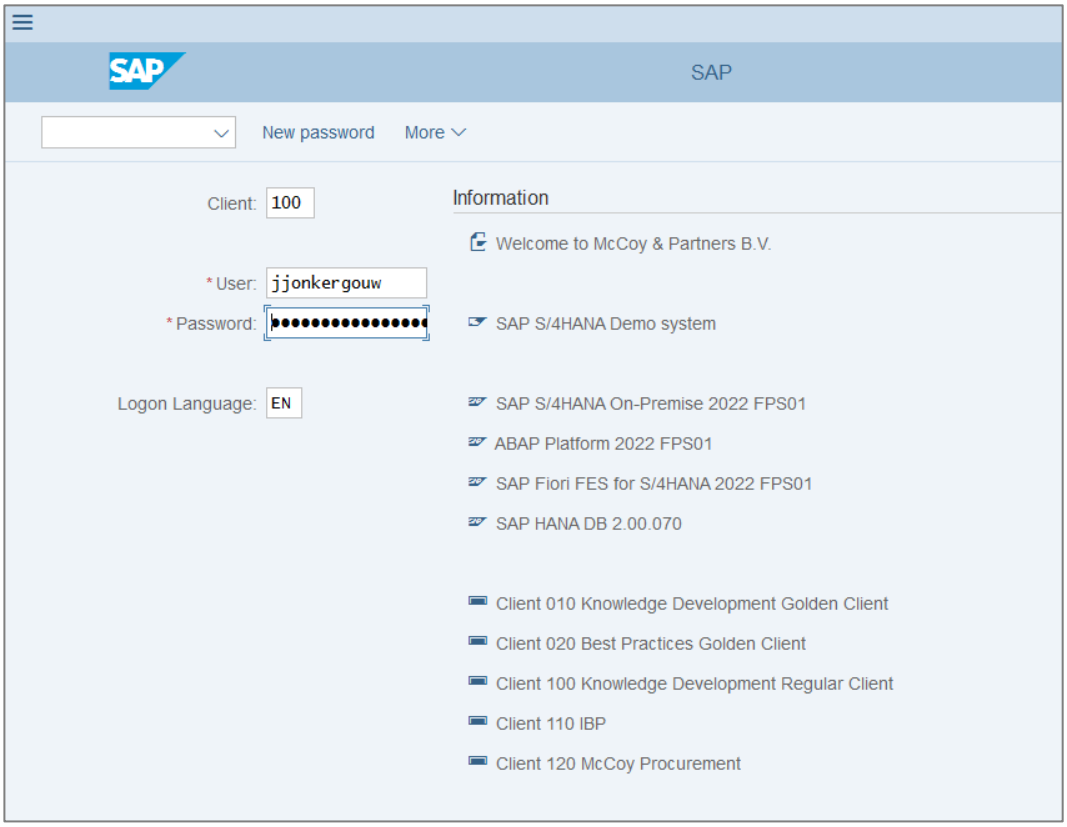
Once the contract has been added it will be shown in the properties tab.



Step 2: Create an OData API

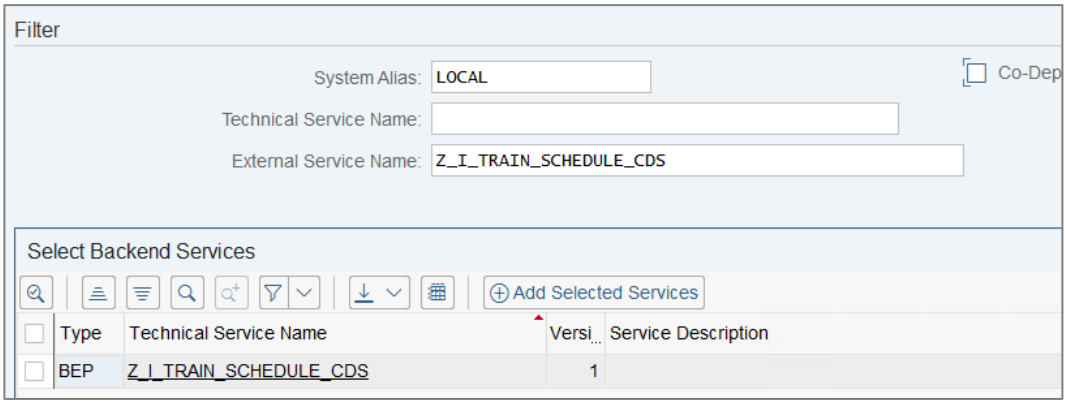
Step 2a: Access SAP S/4 HANA via SAP GUI

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

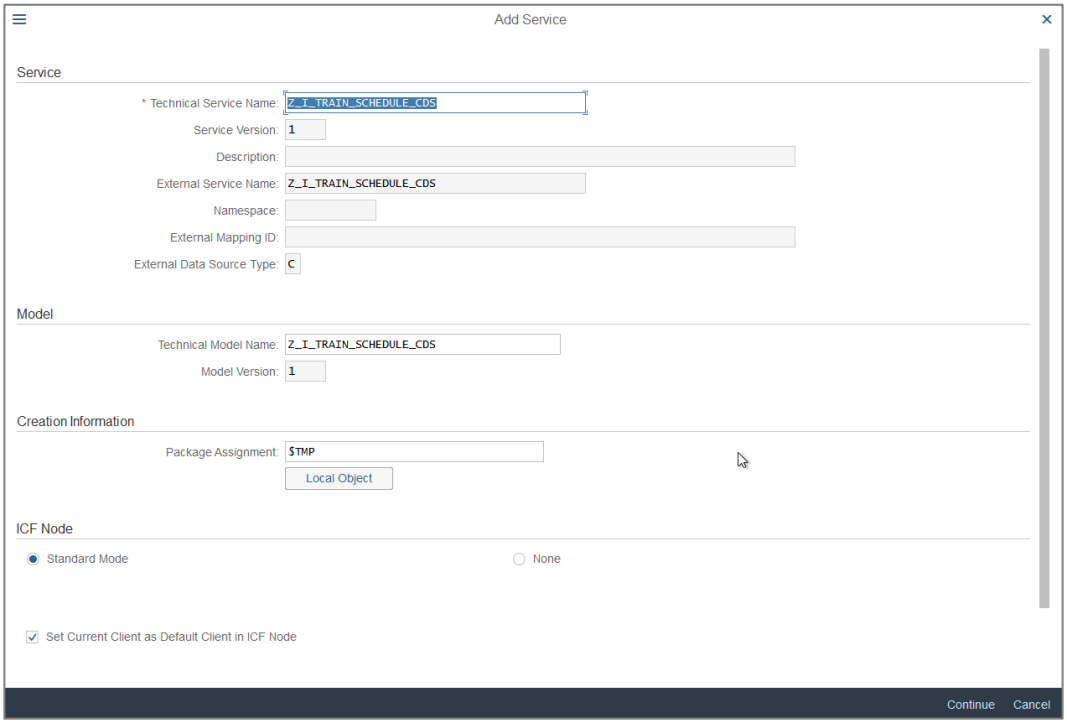


Step 2b: Enable OData Services

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"

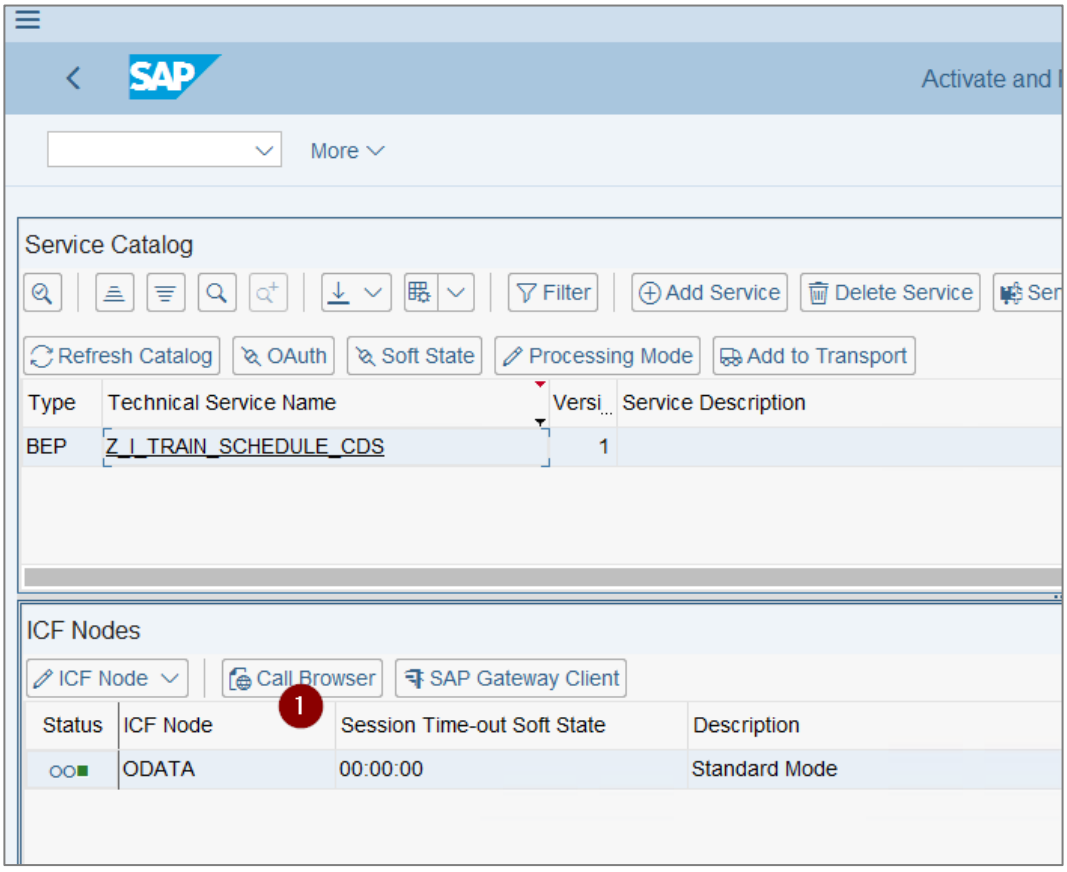


Assign "$TMP" as Package Assignment and click on "Continue"

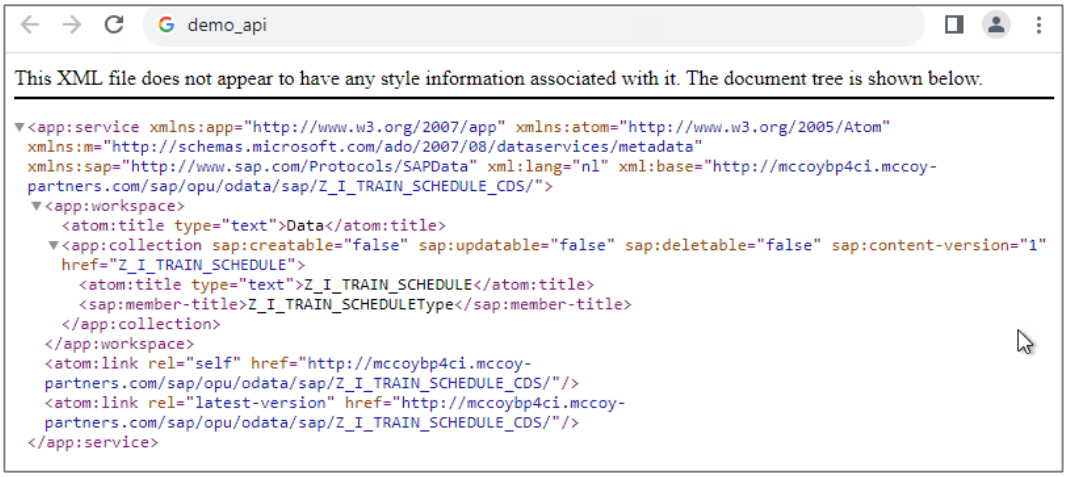


Step 2c: Test OData Service

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



Congratulations your OData API cannot be used for consumption.



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

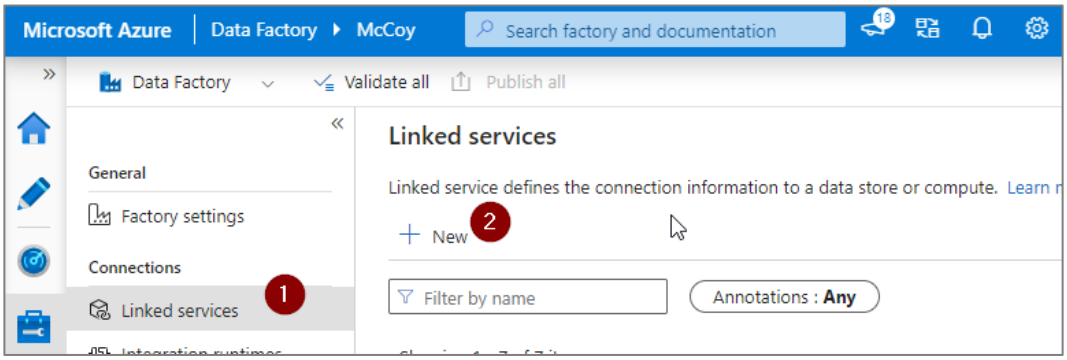
Log in to the Azure Portal with your credentials.

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

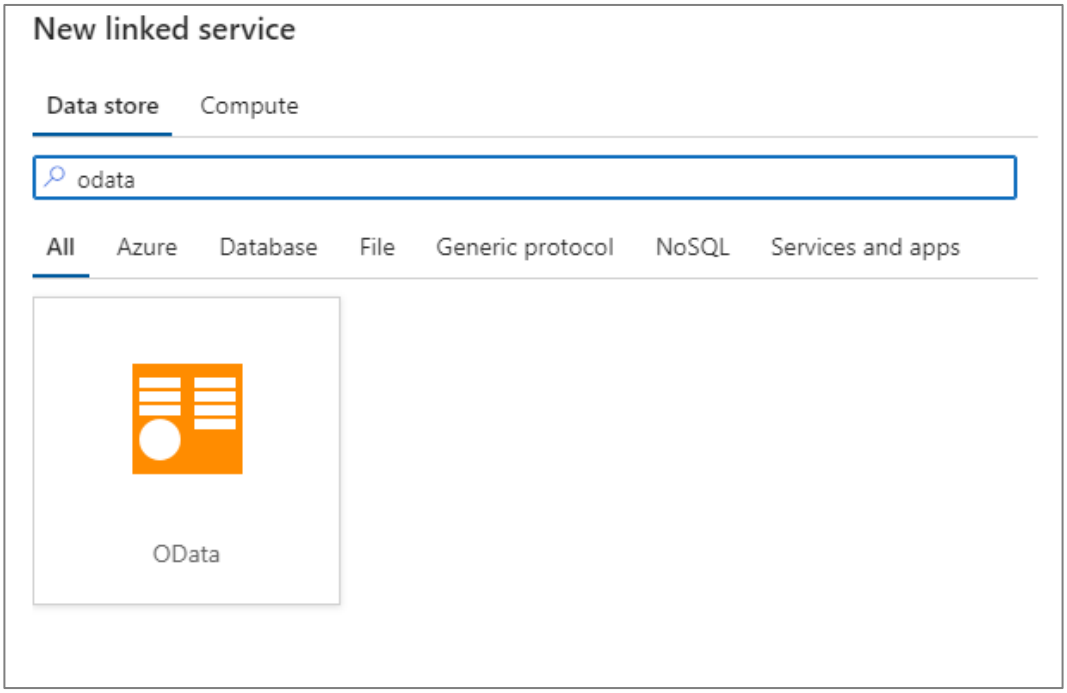
If you haven't already, create a new Azure Data Factory instance by following the Azure Portal's guided process.

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

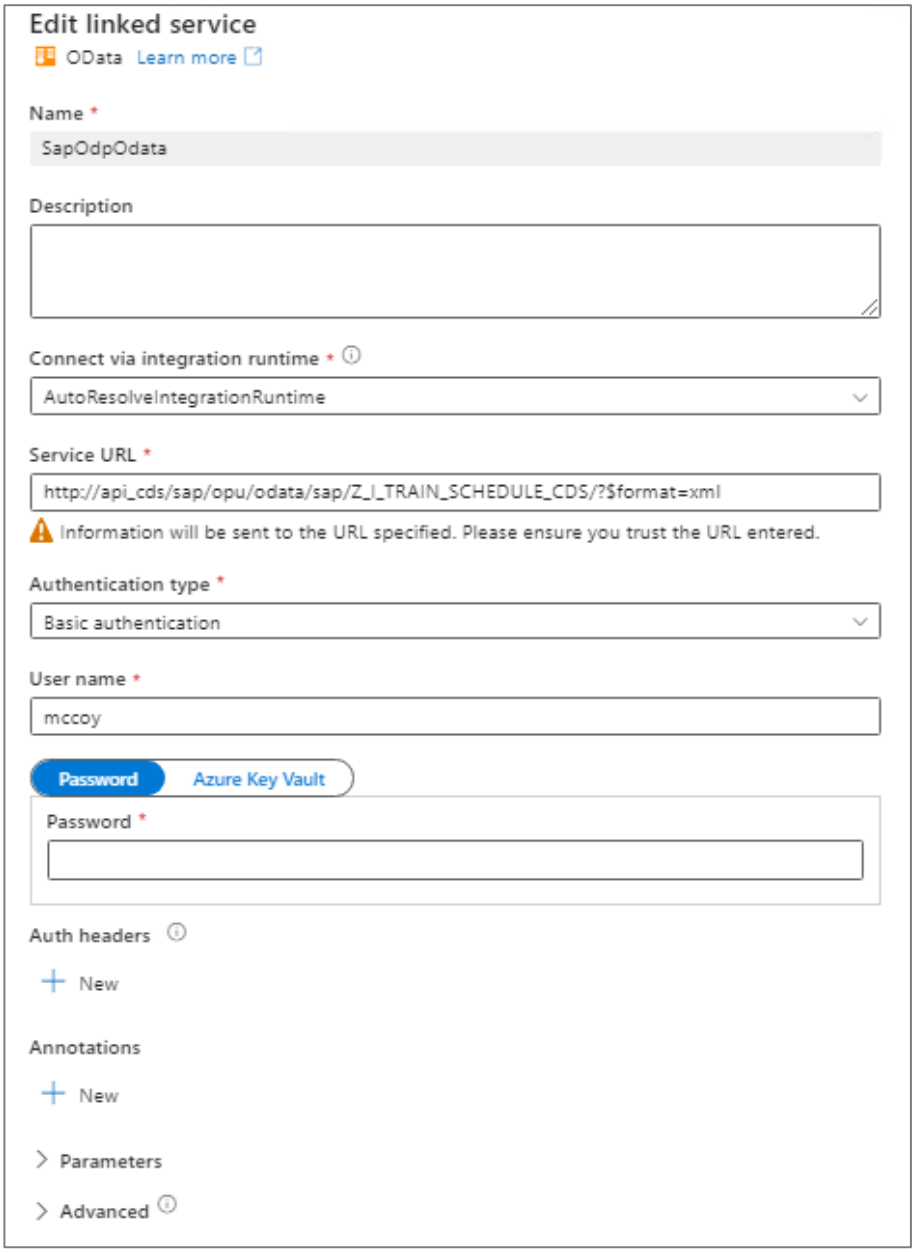
Within your Azure Data Factory, add a linked service for SAP S/4, providing the necessary connection details towards the OData API.



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



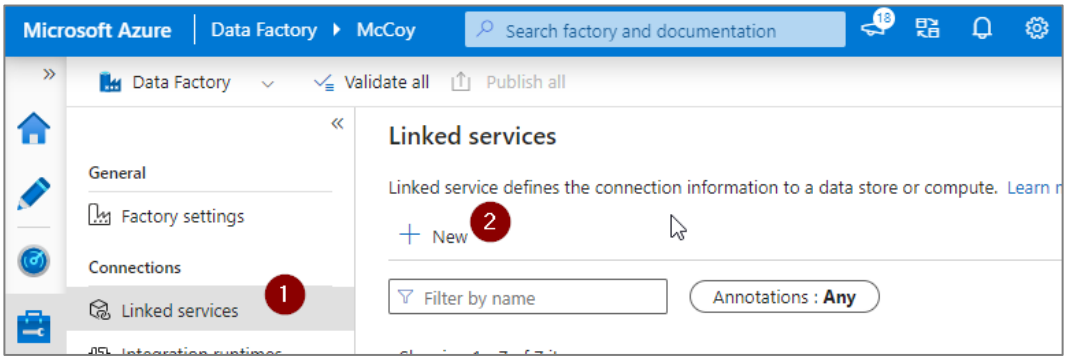
Fill in the required fields and test the connection.



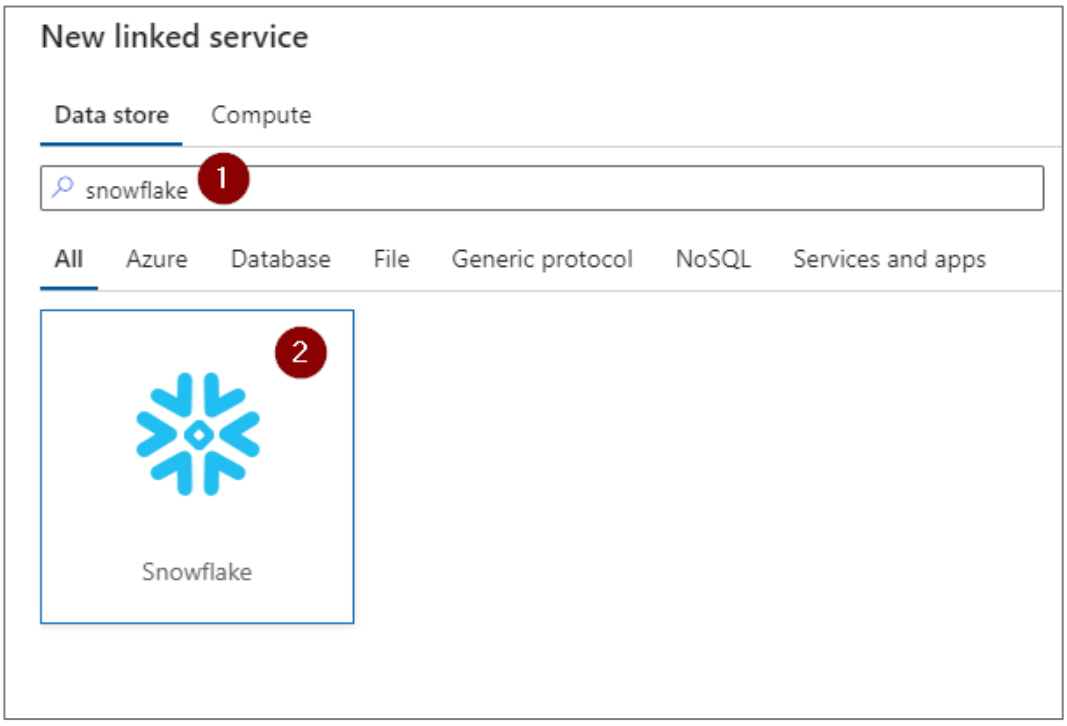
It's advisable to test the connection to ensure that the provided settings are correct and that ADF can connect to the OData API. Once the configuration is complete and the connection test is successful, save the linked service.

Step 4e: Add a Linked Service for Snowflake

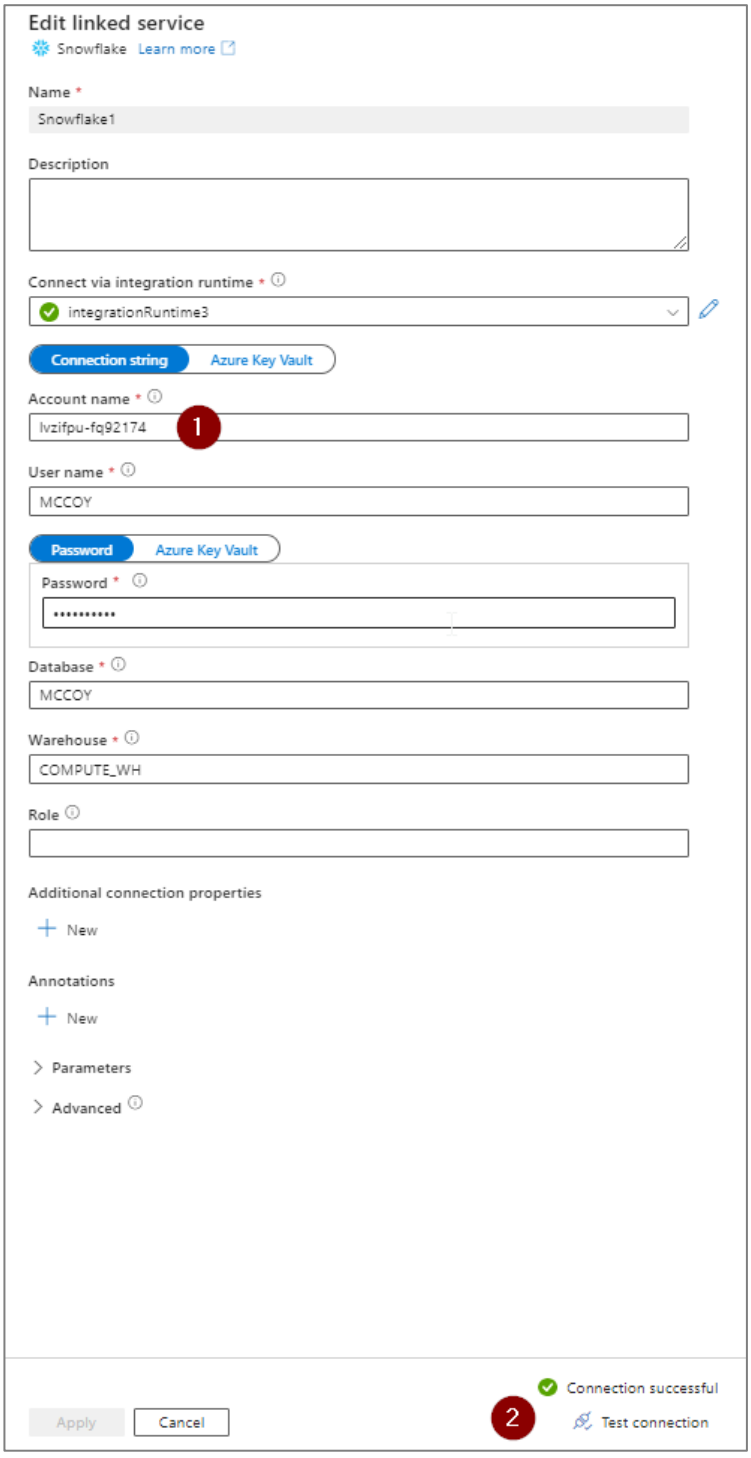
In your Azure Data Factory, add a linked service for Snowflake. Provide the Snowflake connection details, including credentials.



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



Fill in the required fields and test the connection.

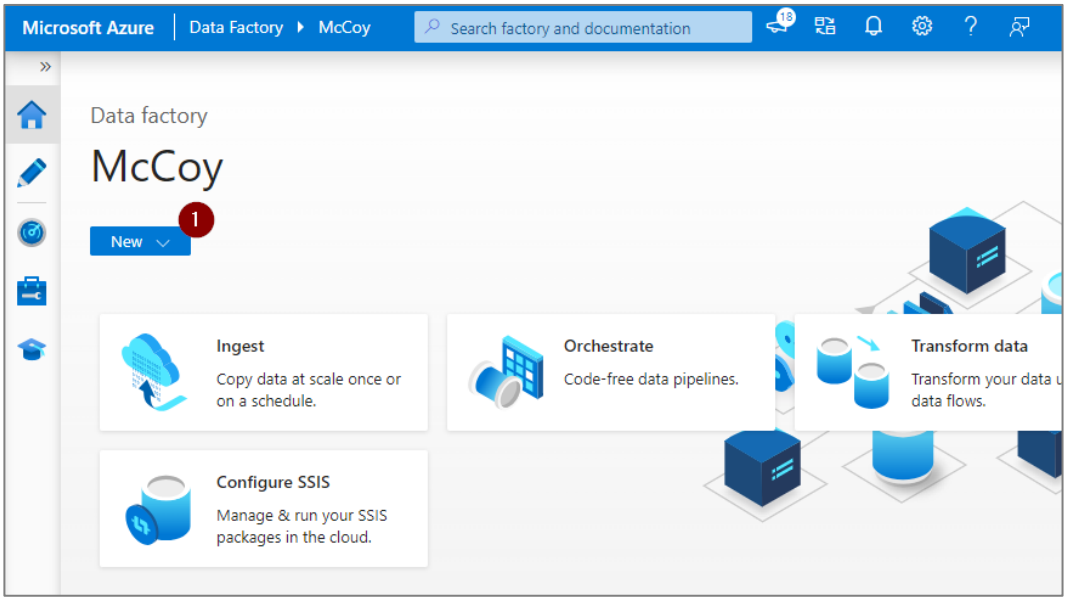


It's advisable to test the connection to ensure that the provided settings are correct and that ADF can connect to Snowflake. Once the configuration is complete and the connection test is successful, save the linked service.

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.



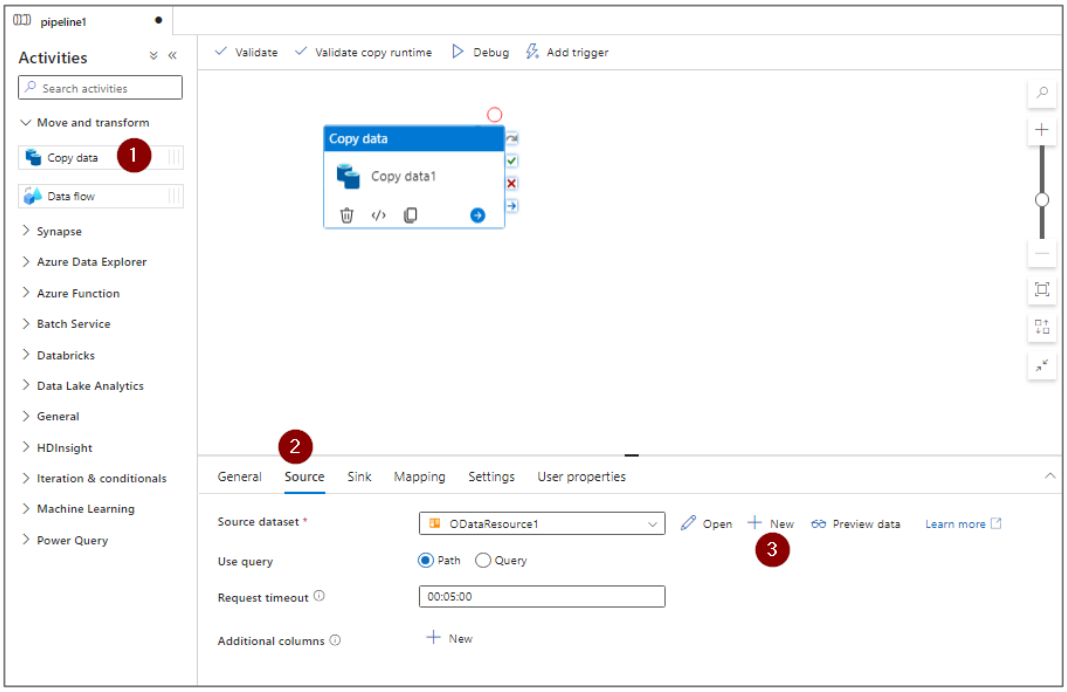
Step 5b: Add “Copy Data” activity

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

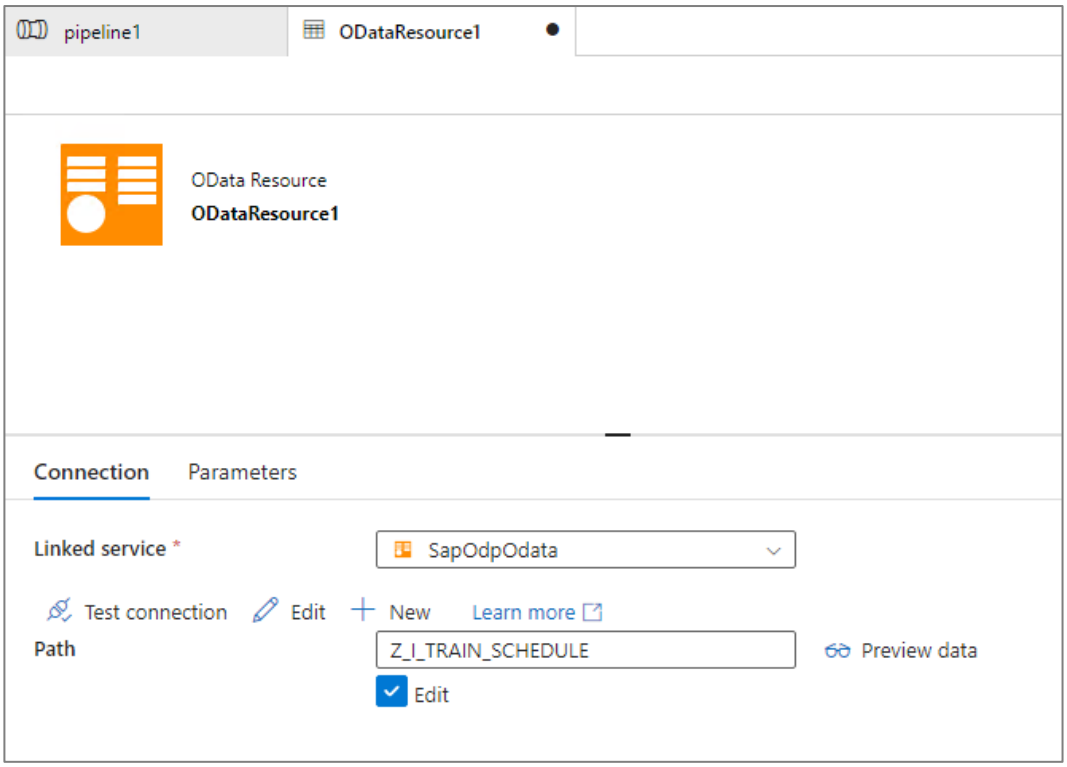
For this specific purpose a simple 1:1 is sufficient and therefore only add a “Copy Data” activity to the pipeline.

Step 5c: Define source data

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

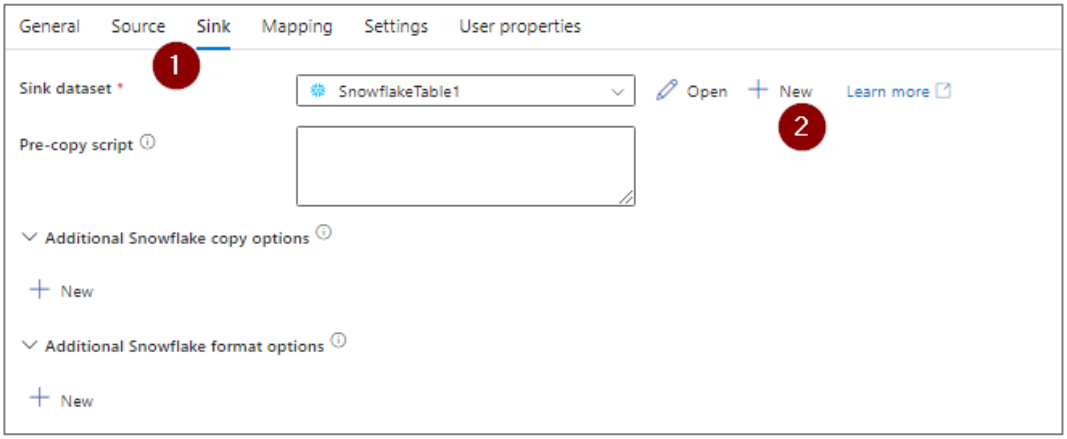


Now select your “Linked Services” and fill in your API technical name. Test the connection by previewing the data.

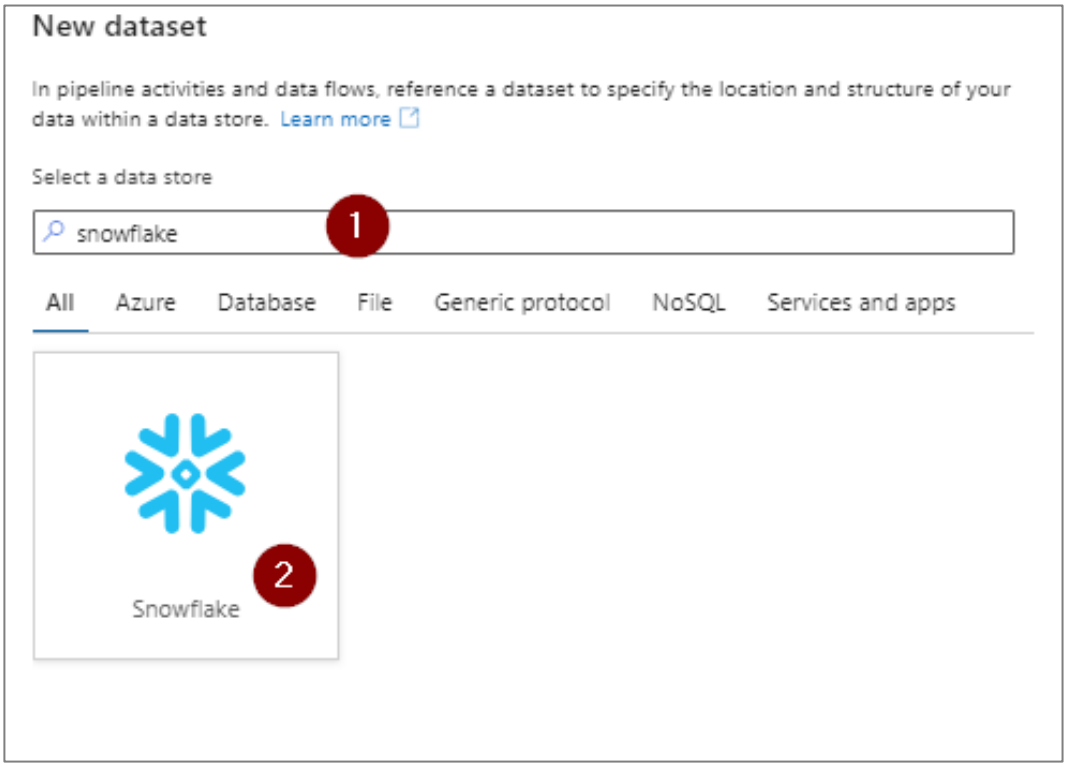


Step 5d: Define target data

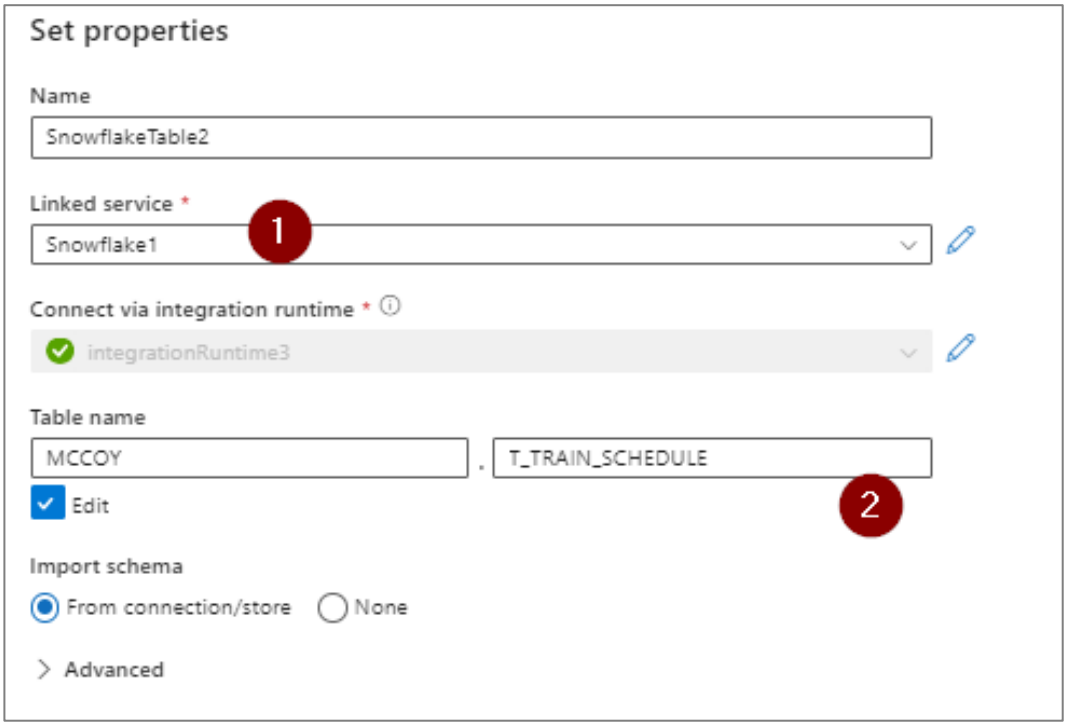
In the "Sink" tab of the "Copy Data" activity, configure the sink dataset. This is where the data will be copied to. Similar to the source, specify the sink data store and provide connection details and credentials. For this purpose we use the Snowflake “Linked Service”



Search for “Snowflake” and select the option.

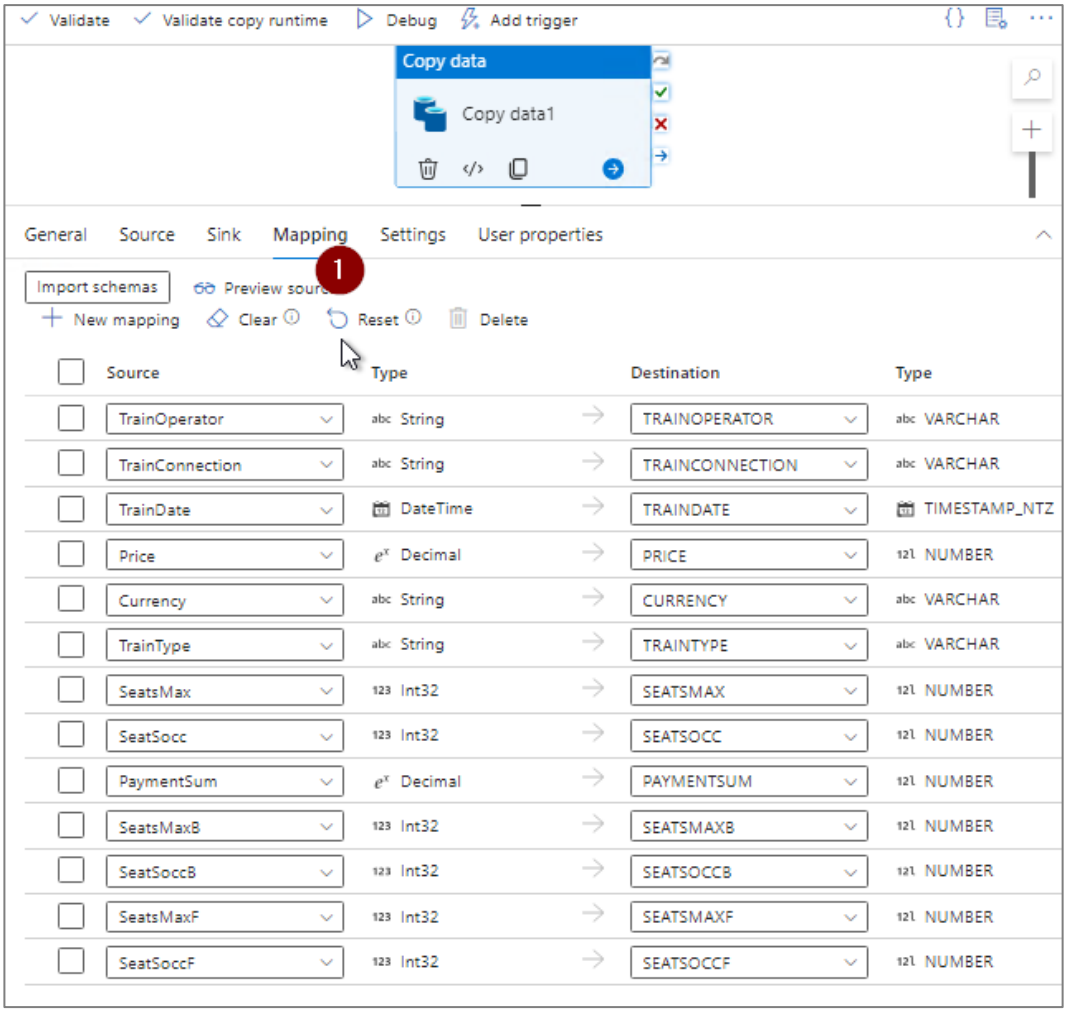


Define the target table of Snowflake as defined in Step 3



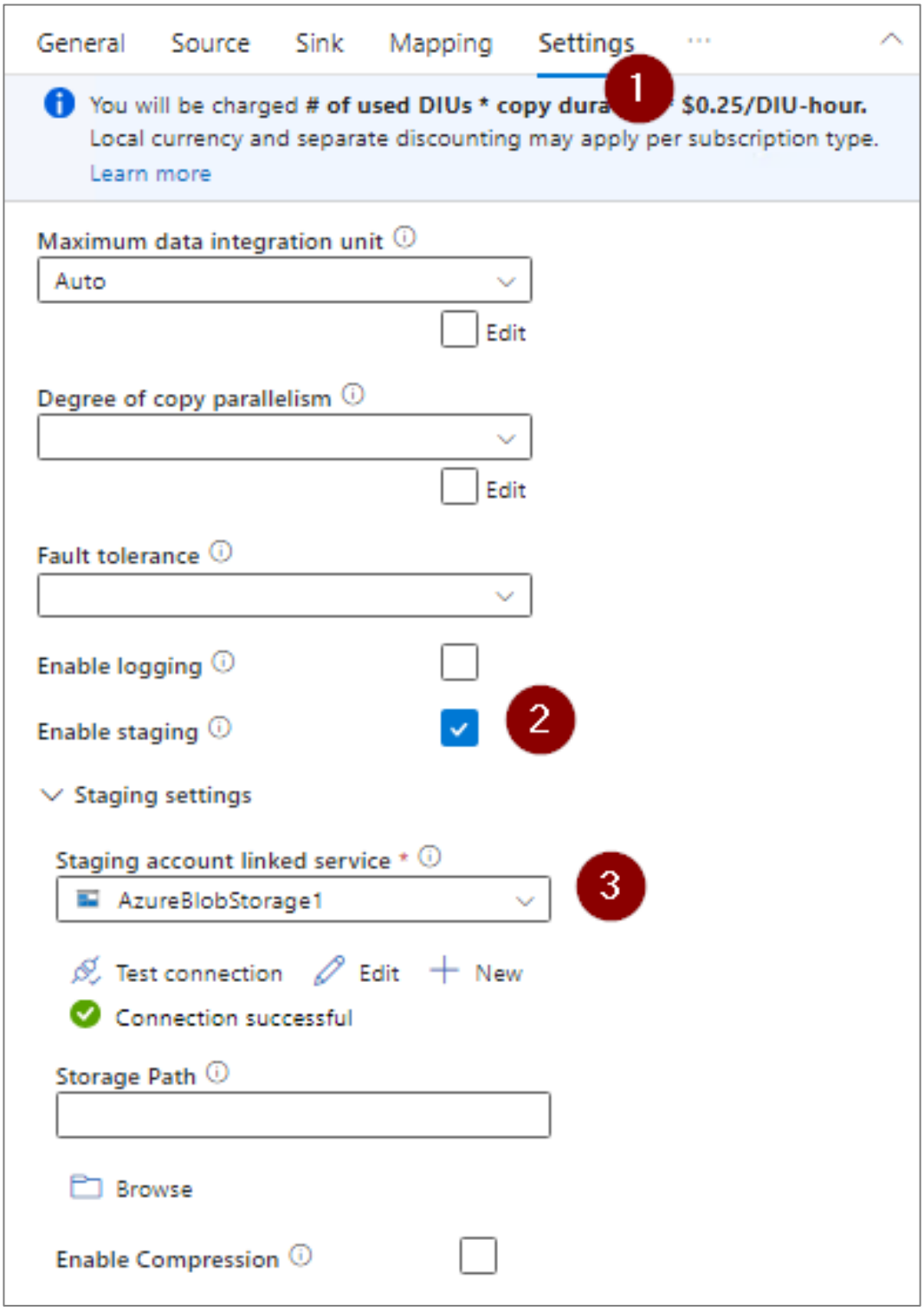
Step 5e: Mapping and transformation

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.



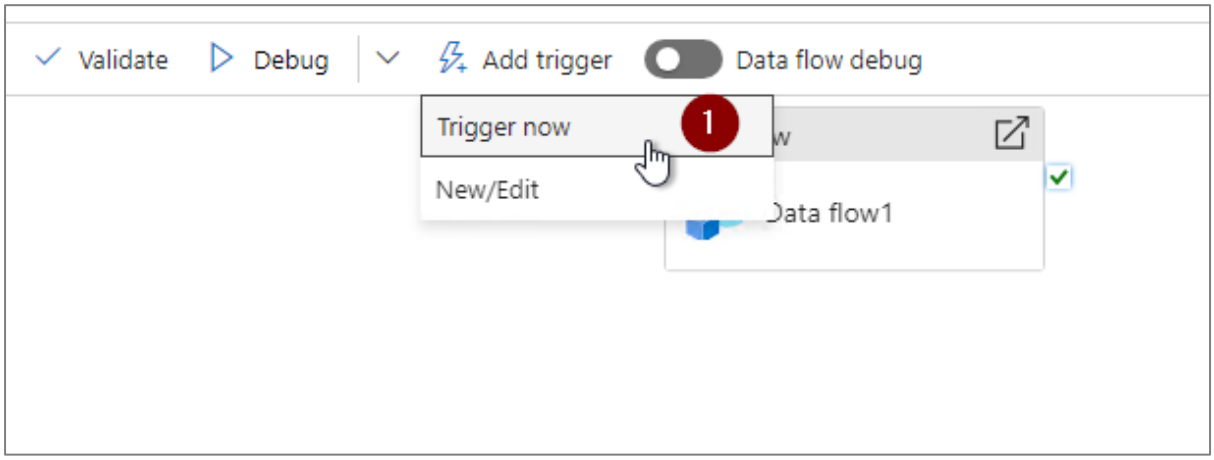
Step 5f: Additional settings

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

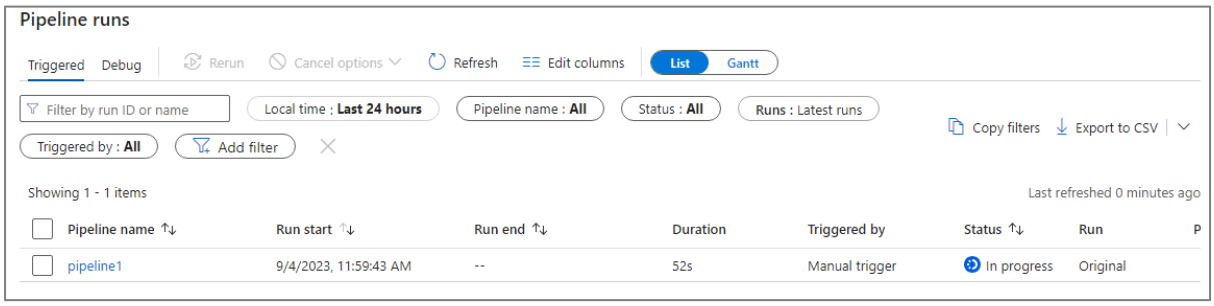


Step 6: Transfer the data from S/4 to Snowflake

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

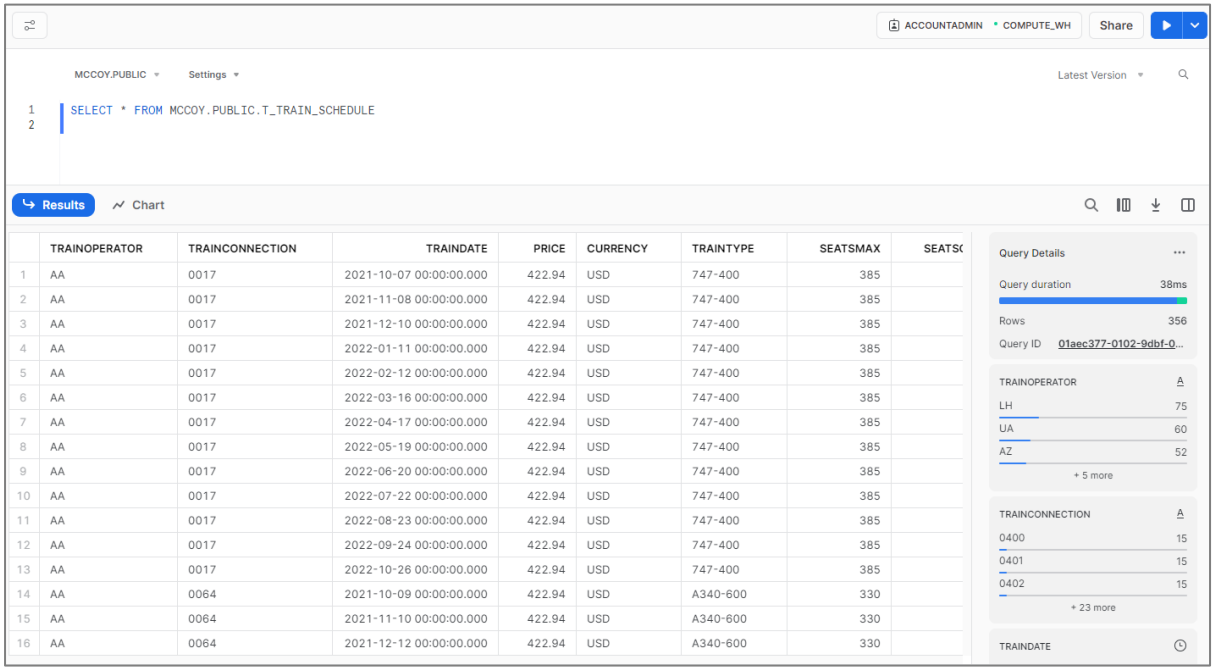


Monitor the pipeline's execution and validate the data transferred from SAP S/4 to Snowflake.



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.



	TRAINOPERATOR	TRAINCONNECTION	TRAINDATE	PRICE	CURRENCY	TRAINTYPE	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