SQL days konferenz

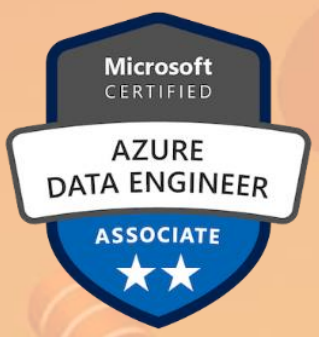
Azure Data Factory – what's new?

Neues in ADF & Synapse Pipelines

Stefan Kirner
scieneers GmbH

Stefan Kirner

- PASS Chapter Lead Karlsruhe ski@sqlpass.de & Beirat
- Director Business Intelligence scieneers GmbH
- Twitter: @KirnerKa

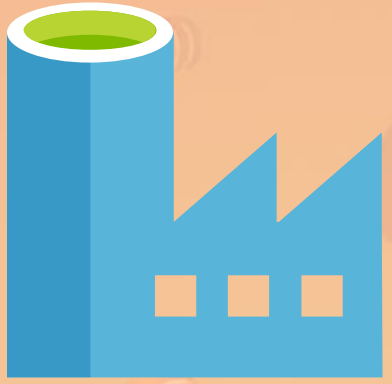


Agenda

- Azure Data Factory / Synapse Pipelines
- Flowlets und Data Libraries
- Activity & Data Flow updates
- Application Lifecycle Management
- Change Data Capture support extended

Azure Data Factory vs. Synapse Pipelines

Was ist was und wann benutze ich welches?



Was ist Azure Data Factory?

Ganz einfach...

WHAT IS AZURE DATA FACTORY

READ THE DOCS

A CLOUD-BASED DATA INTEGRATION SERVICE THAT ORCHESTRATES DATA MOVEMENT & TRANSFORMATION BETWEEN DIVERSE DATA SOURCES ⟷ CLOUD COMPUTE RESOURCES AT SCALE

DATA INTEGRATION CHALLENGE

WHAT PROBLEM DOES IT SOLVE?

BIG DATA

DATABASES, FILES, DATA STREAMS, DEVICES

DIVERSE DATA SOURCES, DURATIONS

DATA INGEST → DATA TRANSFORMATION → DATA MOVEMENT

STORED DATA, CLEANED & ENRICHED

ACTIONABLE INSIGHTS

DASHBOARDS, REPORTS

DATA OPS TEAM

HELP! We need to collect all this data!

DATA ANALYSTS

WE NEED DATA WITH RELEVANT CONTEXT TO DO OUR ANALYSIS

HOW CAN WE ORCHESTRATE (DATA MOVEMENT) AND OPERATIONALIZE (WORKFLOW)

AZURE DATA FACTORY

- Raw, unorganized data stored in relational, non-relational, and other storage systems
- We need actionable business insights but raw data lacks the correlating context (by itself) for analysis
- Big data needs a scalable service than can ORCHESTRATE the data movement and OPERATIONALIZE the data processing workflows!

THE SCENARIO YOU RECOGNIZE

GAME DEV COMPANY

- HAVE: Petabytes of game log data in the cloud
- WANT: Actionable insight into my customers
  - gaming preferences
  - player demographics
  - user behaviors
- USES
  - develop new features
  - improve player experience
  - upsell, cross-sell and drive business growth.

DATA ANALYST

WHAT I NEED..

- reference data from our on-prem services
  - customer info
  - marketing campaign
  - game metadata
- * combined with gameplay data logs from cloud
  - game play behaviors
  - success, failure rates etc.

WHAT I MUST DO

- collect data from sources
- process data to transform it
- publish data into warehouse for analysts.

AUTOMATE THIS WORKFLOW (TRIGGER IT BY AN EVENT OR MANUALLY — MONITOR & MANAGE DAILY SCHEDULES)

HOW DOES ADF WORK?

CAN YOU TELL ME WHAT ADF DOES AT A HIGH LEVEL?

LET'S BREAK IT DOWN

AZURE DATA FACTORY IS A SET OF INTERCONNECTED SYSTEMS ... THAT GIVE YOU AN END-TO-END PLATFORM ... FOR YOUR DATA ENGINEERING!

DESIGN WIZARD OR CODE + COMPUTE

CONNECT & COLLECT

1. Build an information production system to connect all data sources and adapt ingest to their diverse intervals and speeds! (CONNECTORS)
2. Collect all data in a centralized location to facilitate processing (e.g. transformation) next (COPY ACTIVITY)

TRANSFORM & ENRICH

3. Transform the collected data using actions to aggregate, filter, clean etc. Use code-free UI based mapping data transformation graph creation! (OR) use compute to transform data by hand. (MAPPING DATA FLOW)

CI/CD & PUBLISH

4. Manage data pipeline ops using Azure DevOps and GitHub! Incrementally develop and deploy your E-T-L process then publish processed data into Azure Data Warehouse, Azure SQL Database, Azure Cosmos DB – or your BI analytics engine! (PIPELINE OPS)

MONITOR & ALERT

Use Azure Monitor, API, PowerShell, health panels on Azure Portal!

5. Once your data pipeline is created and deployed, monitor it for the success/failure rates! (AZURE MONITOR)

LET'S TALK DATA INTEGRATION PATTERNS

The most common pattern is ETL

- E XTRACT = connect to sources & copy data to central store
- T RANSFORM = process the data for analysis & data transformation at scale
- L OAD = move the data to data warehouse or analytics engines for business insights

Another pattern is ELT where raw (native) data is itself loaded (stored) before the transformation phase ....

AZURE DATA FACTORY HAS CODE-FREE ETL AS A SERVICE!

You always have the option to do hand-coded transformation using Azure compute ... but mapping data flows provide a UI-based wizard to simplify your pipeline setup ..

5 ASPECTS IT COVERS!

1. INGEST DATA
   - * Copy data on-prem or in cloud!
   - * 90+ native connectors
   - * Serverless, auto-scaling!
2. CONTROL FLOW
   - * Design (via UI) or Create (via SDK) your data pipelines
   - * Utilize workflow constructs (loops, parameters, variables...) for efficiency
3. DATA FLOW
   - * Code-free data transforms (execute automagically in Spark)
   - * Scale out with Azure Integration Runtimes, generate data flows with SDK (code) or Designer (UI)
4. SCHEDULE OPS
   - * Build/Maintain ops schedules for your data pipelines
   - * Options include: Wall clock, Event based, Tumbling windows, chaining
5. MONITOR OPS
   - * View active executions and pipeline history – details at activity/pipeline
   - * Establish alerts for key events or progress notifications

7 THINGS TO KNOW ABOUT AZURE DATA FACTORY

1. ENTERPRISE READY — Data Integration At cloud scale!
2. ENTERPRISE DATA READY — 90+ connectors! It just works.
3. CODE-FREE TRANSFORMATION — UI driven mapping data flows
4. RUN CODE ON ANY AZURE COMPUTE — For hands-on data transformations
5. MANY SSIS PACKAGES RUN ON AZURE — (Reboot on-prem in ADF in 3 steps)
6. ADF CAN MAKE DATA OPS SEAMLESS — source control, automated deploy, simple templates
7. SECURE DATA INTEGRATION — Managed virtual networks protect against data exfiltration, simplify your networking!

6 INTEGRATION RUNTIME (IR)

INTEGRATED RUNTIME (IR) is COMPUTE INFRASTRUCTURE USED BY ADF TO PROVIDE FULLY MANAGED

- DATA FLOWS (transformation)
- DATA MOVEMENT (movement)
- ACTIVITY DISPATCH (route to service)
- SSIS PACKAGE EXECUTION (in managed compute)

IR Types:

AZURE-SSIS	* SSIS package execution	(public or private)	
SELF HOSTED	* Activity Dispatch * Data Movement	public or private	
AZURE	* Activity Dispatch * Data Movement * Data Flow	public or private	
CAPABILITY	N/W		

3 IR TYPES

- SELF-HOSTED
- AZURE-SSIS
- AZURE

POTENTIAL TO HAVE HYBRID DATA INTEGRATION SOLUTIONS

PICK WHAT WORKS FOR YOUR DATA NEEDS ... AND YOUR NETWORKING REQUIREMENTS!

5 ADDITIONAL CONCEPTS

1 CONTROL FLOW

REFERS TO THE ORCHESTRATION OF PIPELINE ACTIVITIES INCLUDING:

- CHAINING ACTIVITIES
- BRANCHING ACTIVITIES
- CUSTOM STATE PASSING
- DEFINING PARAMETERS
- PASSING ARGUMENTS (for pipeline run)
- LOOPING CONTAINERS

Think of ADF as the conductor for an orchestra of activities! (KNOWS THE BIG PICTURE)

2 PIPELINE RUN

AN INSTANCE OF PIPELINE EXECUTION

(arguments) A1 A2 A3 (parameters)

INSTANTIATE PIPELINE BY PASSING ARGUMENTS (values) TO PARAMETERS (placeholders) DEFINED IN IT.

PASS ARGUMENTS manually OR BY DEFINING triggers

3 TRIGGERS

TRIGGER EVENT

UNIT OF PROCESSING THAT DETERMINES WHEN TO KICK OFF PIPELINE RUNS.

DIFFERENT TRIGGER TYPES FOR VARIOUS EVENTS!

4 PARAMETERS

"READ-ONLY" KEY/VALUE PAIRS POPULATED FROM RUN CONTEXT AT EXECUTION

BOTH 'DATASET' AND 'LINKED SERVICE' ARE STRONGLY-TYPED, REUSABLE, REFERENCEABLE PARAMETER ENTITIES!!

ACTIVITY REFERENCES DATASET TO CONSUME DATA PROPERTIES, AND 'LINKED SERVICE' FOR THE CONNECTION INFO (to data store for copy, compute resource for flow)

5 VARIABLES

ARE USED INSIDE PIPELINES TO STORE TEMPORARY VALUES, STATE

USED WITH PARAMETERS TO PASS VALUES BETWEEN ACTIVITIES, DATA FLOWS, PIPELINES.

2 ACTIVITIES

REPRESENT A SINGLE PROCESSING STEP IN THE PIPELINE e.g.

- COPY ACTIVITY — copy data from one store (source) to other (destination)
- HIVE ACTIVITY — run a Hive query on an HDInsight cluster

THREE TYPES OF ACTIVITY SUPPORTED BY ADF

- Data Movement
- Data Transformation
- Control

1 PIPELINES

A1 A2 A3 PIPELINE STAGE

- A PIPELINE IS A LOGICAL GROUPING OF ACTIVITIES THAT PERFORM ONE UNIT OF WORK
- AN ADF INSTANCE CAN HAVE ONE OR MORE PIPELINES!

Example: Pipeline that ingests data from Azure Blob (A1), runs Hive query on HDInsight to partition it (A2) and moves result to next store (A3)

Benefit: Manage correlated activities as a single entity!

Execution: Activities may be chained (run sequentially) or be independent (run in parallel) within pipeline!

3 DATASETS

- TWO USES
  - REPRESENT DATA SOURCE
  - REPRESENT COMPUTE RESOURCE
- REPRESENT CONNECTION INFORMATION TO LINK ADF TO EXTERNAL SERVICES
- WORKS CLOSELY WITH DATASETS. THIS SPECIFIES HOW TO CONNECT & DATASET DEFINES WHAT DATA LOOKS LIKE...

data sources for INGEST

compute resources for TRANSFORM

DATA STORE

DATA SET (view)

- REPRESENTS DATA STRUCTURE GIVING SELECT VIEW INTO DATA STORE
- POINTS TO THE SPECIFIC DATA SUBSET TO USE IN ACTIVITY...
- .. FOR INPUT & OUTPUT

5 MAPPING DATA FLOWS

ADF EXECUTES LOGIC ON SPARK CLUSTER FOR YOU (automagically) & NO NEED TO MANAGE OR MAINTAIN YOUR OWN CLUSTER!

- CREATE & MANAGE DATA TRANSFORMATION GRAPHS THAT WORK ON ANY SIZE DATA
- BUILD UP REUSABLE LIBRARY OF DATA TRANSFORMATION ROUTINES!
- EXECUTE TRANSFORM ROUTINES IN PIPELINES (AS ACTIVITY) TO SCALE!

HEY PRESTO!

4 LINKED SERVICES

HERE ARE THE KEY TERMS AND CONCEPTS YOU NEED TO KNOW TO USE ADF!

AND A FEW KEY TERMS

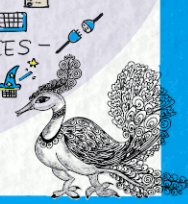
- CONTROL FLOW
- PIPELINE RUN
- TRIGGERS
- PARAMETERS
- VARIABLES

ADF TOOLKIT

LET'S TALK ABOUT COMPONENTS OF AZURE DATA FACTORY

6 MAIN CONCEPTS

- PIPELINES
- ACTIVITIES
- DATASETS
- LINKED SERVICES
- DATA FLOWS
- INTEGRATION RUNTIMES



Azure Data Factory Parts

Microsoft: „Code-free ETL as a service“

Ingest

- Multi-cloud and on-premise hybrid copy data
- 100+ native connectors
- Serverless and auto-scale
- Use wizard for quick copy jobs

Control Flow

- Design code-free data pipelines
- Generate pipelines via SDK
- Utilize workflow constructs: loops, branches, conditional execution, variables, parameters, …

Data Flow

- Code-free data transformations that execute in Spark
- Scale-out with Azure Integration Runtimes
- Generate data flows via SDK
- Designers for data engineers and data analysts

Schedule

- Build and maintain operational schedules for your data pipelines
- Wall clock, event-based, tumbling windows, chained

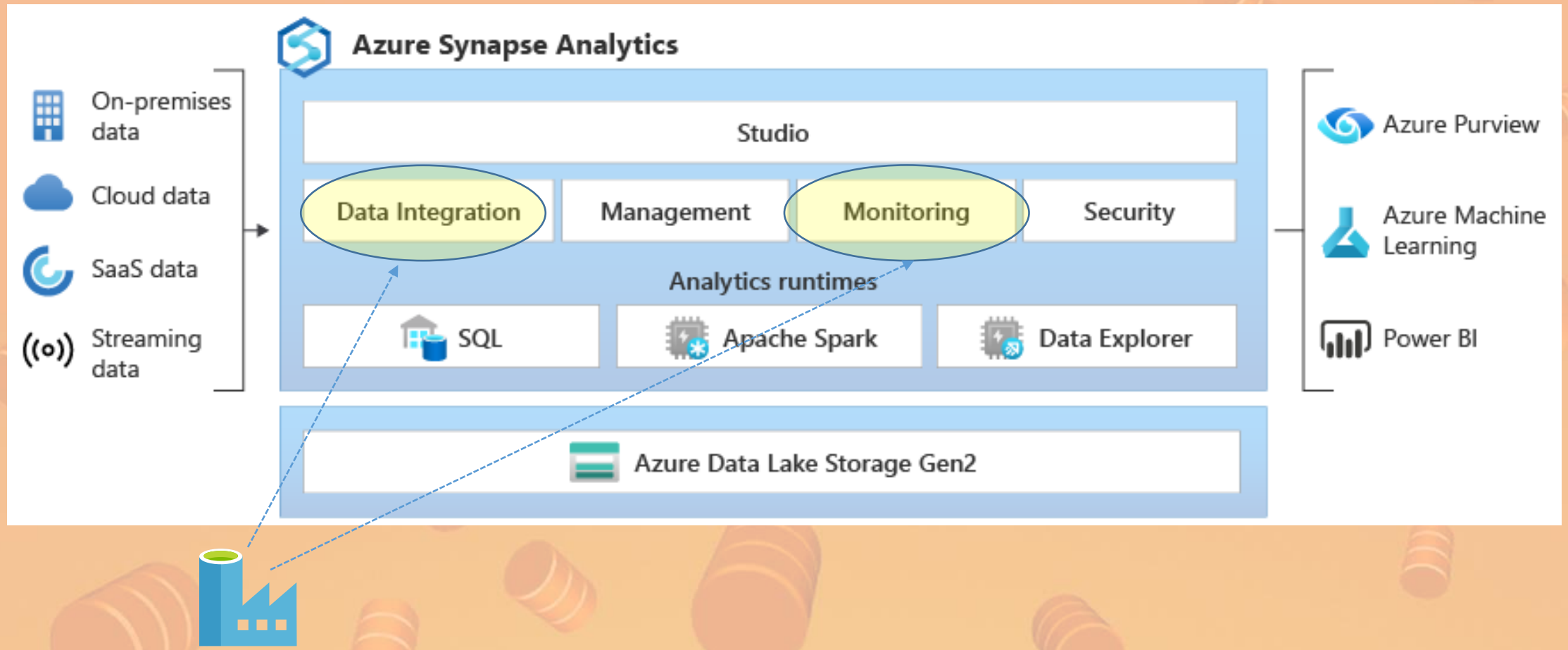
Monitor

- View active executions and pipeline history
- Detail activity and data flow executions
- Establish alerts and notifications

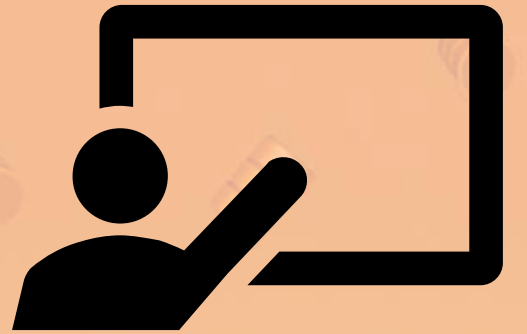
https://docs.microsoft.com/en-us/azure/data-factory/introduction

Was ist Azure Synapse Analytics?

Microsoft: “Azure Synapse is an enterprise analytics service"



https://docs.microsoft.com/en-us/azure/synapse-analytics/overview-what-is



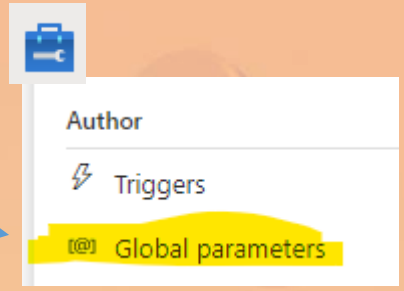
Demo

Überflug ADF vs. Synapse Pipelines

Azure Data Factory only stuff

Not available in Synapse

- Self-hosted Integration runtime sharing
- Support for Cross-region Integration Runtime (Data Flows)
- Power Query support aka Wrapping Data Flows
- Comfortable deployment parameters management
- *Update Snowflake source / destination*
- *Update: SSIS runtime, run SSIS packages -> in Preview for Synapse now*
- *Update: ADF Solution templates also available in Synapse*



https://docs.microsoft.com/en-us/azure/synapse-analytics/data-integration/concepts-data-factory-differences

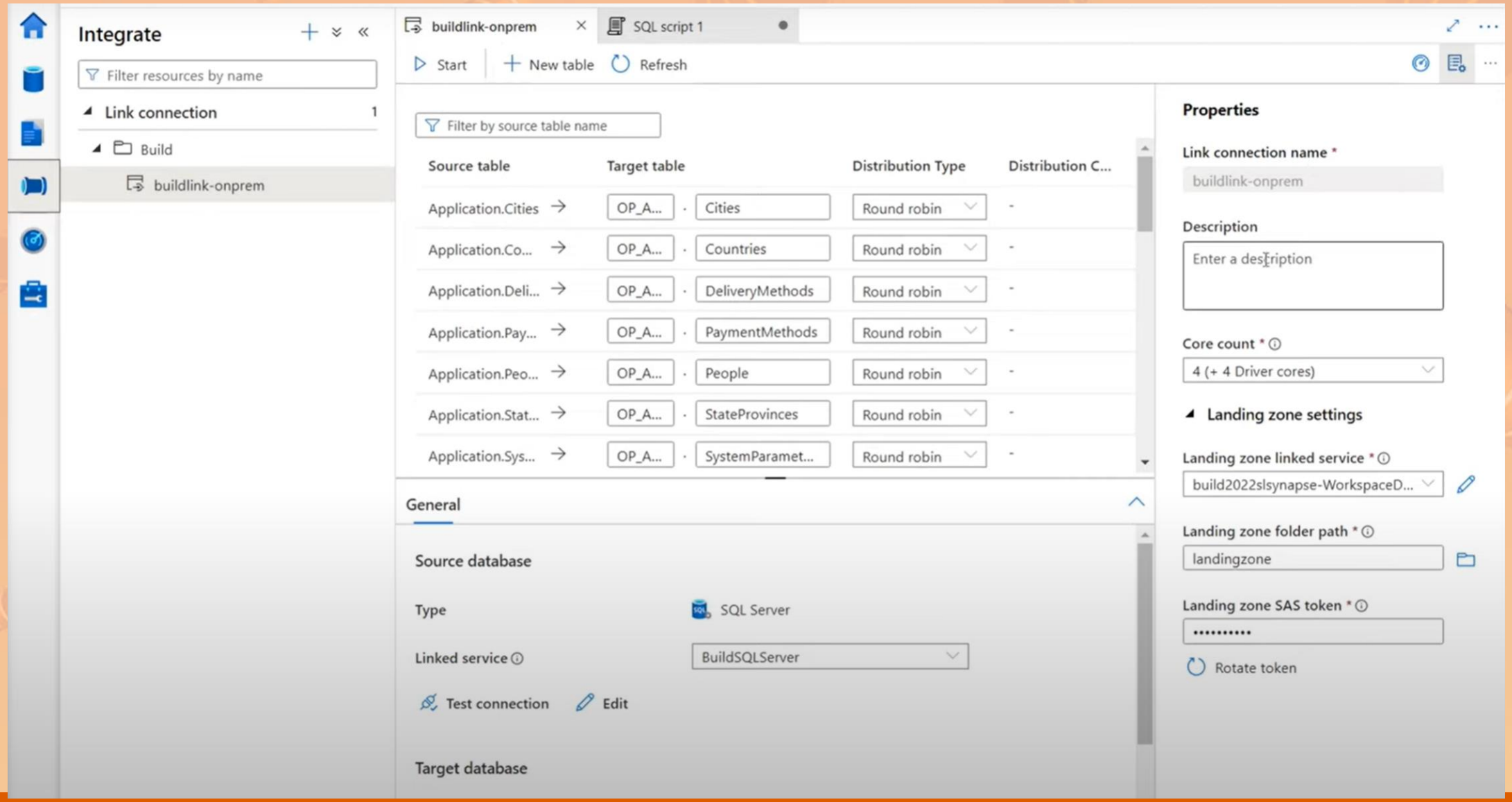
Synapse only stuff (not in ADF)

Not available in Azure Data Factory

- Spark Notebooks & Clusters
- SQL Serverless / dedicated pool (stored procedures for ETL)
- Detailed Monitoring Spark Jobs for Data Flows
- Azure Synapse Link for SQL, Dataverse, Cosmos DB

https://docs.microsoft.com/en-us/azure/synapse-analytics/data-integration/concepts-data-factory-differences

Kurzer Blick: Azure Synapse Link for SQL



Better use Azure Data Factory for

- Small projects (copy a bunch of tables every day)
- To keep a small footprint in infrastructure
- If having only permissions in dedicated Azure Resource Group
- As orchestration tooling for Databricks projects
- Migration from M-Code (Power BI)

Better use Synapse Data Integration if

- You plan to do data exploration on a data lake using SQL or connect Power BI
- Data platform projects including many data sources
- Spark Notebooks used as part of data integration
- Want to sync masses of different tables via Azure Synapse Link .. (having money for dedicated SQL pools)

Flowlets

In Mapping Data Flows

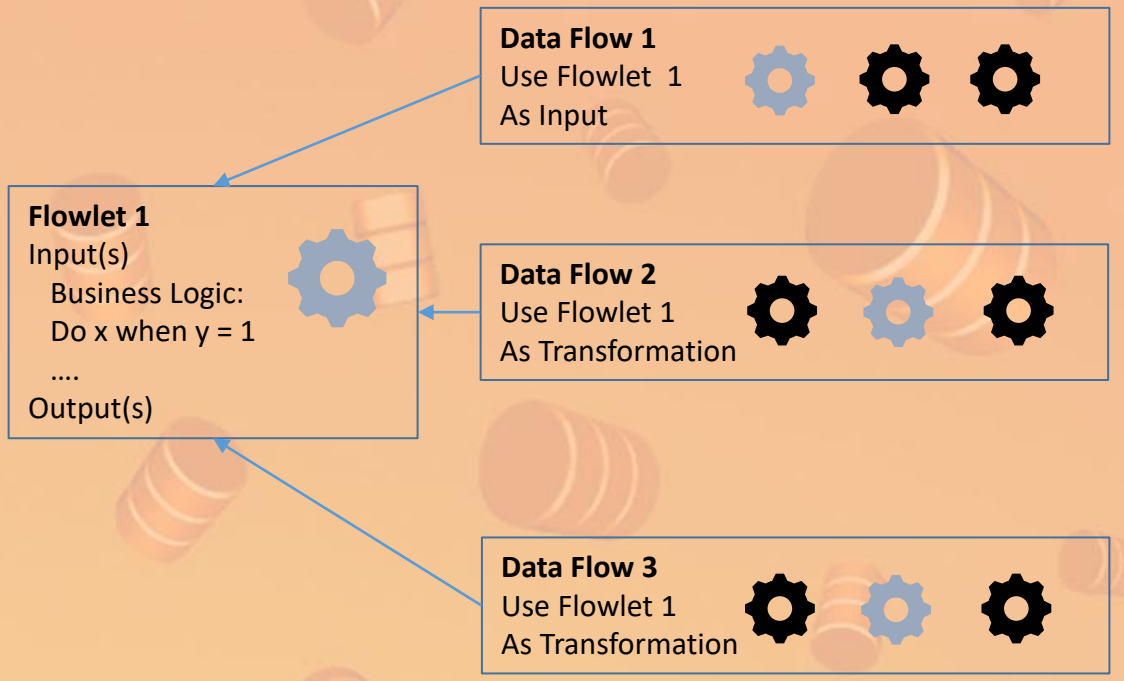
Flowlets in Mapping Data Flows

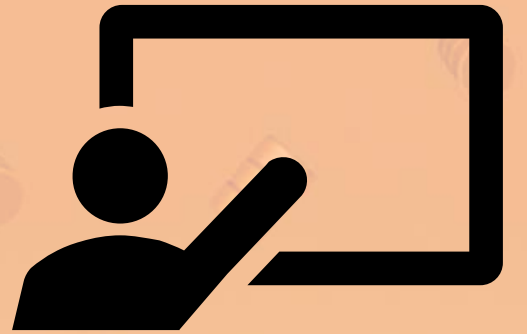
Wiederverwendbarer „Container“ von Aktivitäten

- **D**on’t **R**epeat **Y**ourself Prinzip (DRY) :

Durch Kapselung Duplikation von Code / Logik vermeiden

- Kann from scratch oder aus bestehenden Mapping Data Flows erstellt werden
- Einsatz als Quelle oder Transformation möglich
- Datentypen und Felder möglichst generisch und klein halten
- Schnittstellen-Vertrag zwischen Flowlets und den Data Flows die es verwenden





Demo

Flowlets

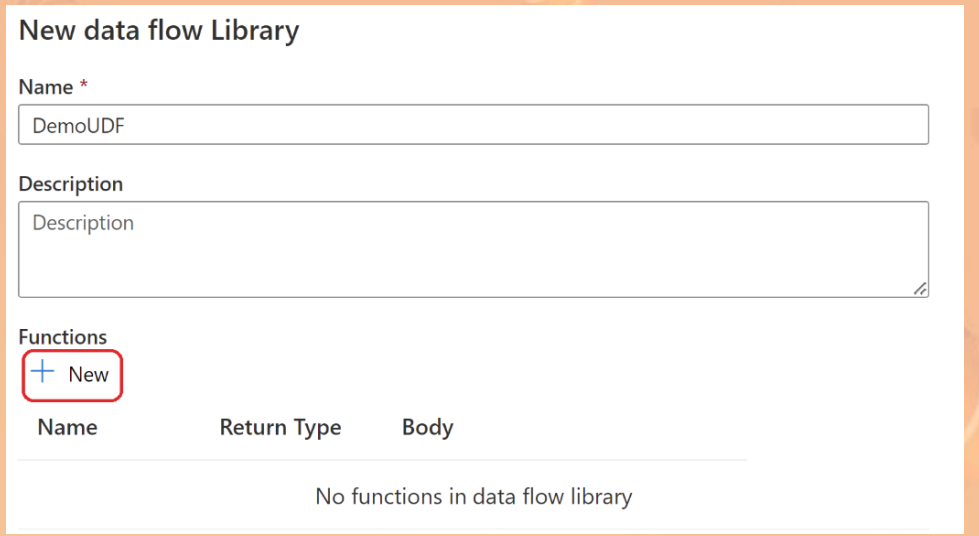
Data Flow Libraries

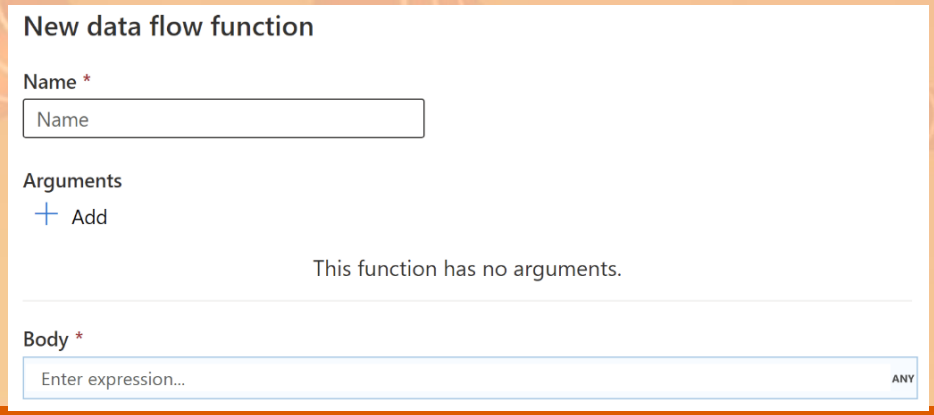
User Defined Functions

Data Flow Libraries – User defined functions

Noch mehr Kapselung – aber auf einer anderen Ebene

- Anlegen von Funktionen in der Expression Language die in allen Mapping Data Flows verwendet werden können
- Parametrisierung möglich
- Wiederverwendbarkeit von komplizierten Codeschnippseln
- DRY Prinzip

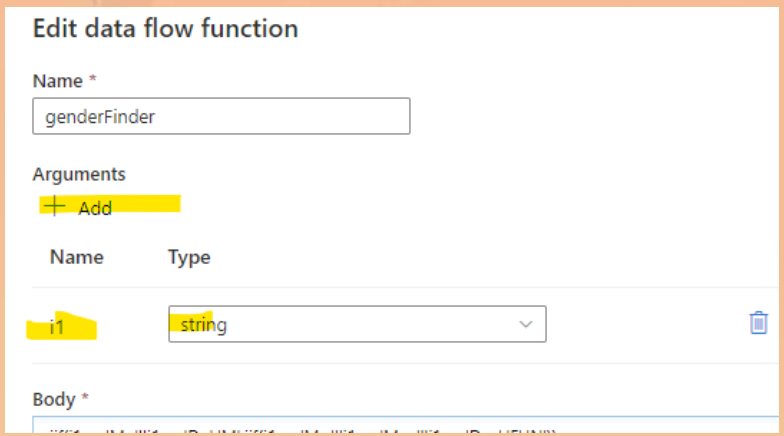


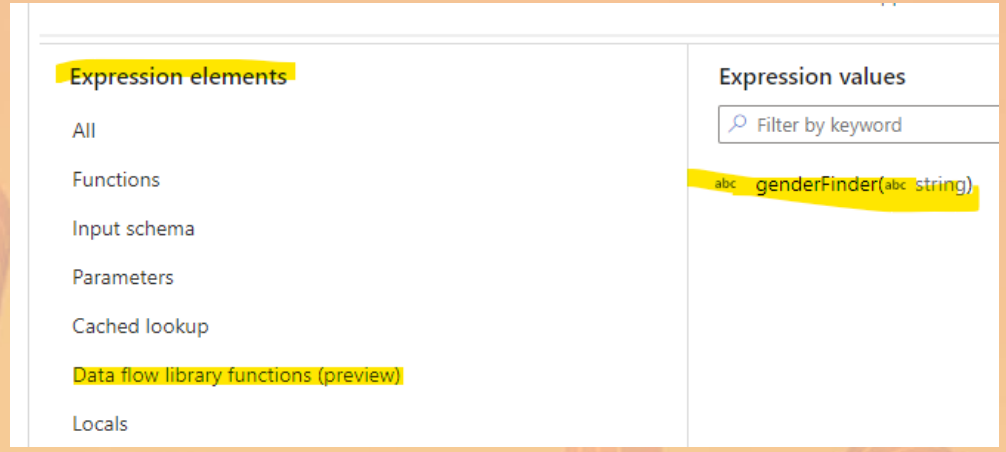


Data Flow Libraries – User defined functions

Definition von User Defined Functions mit Parametern

- Parameter als Arguments hinzufügen
- Namen sind vorgegeben als i1, i2 …
- Im bekannten Expression Editor entwickeln
- Einbinden dann im Data Flow im Expression Editor z.B. bei Derived Column unter Data flow library functions





Demo

Data Flow Libraries – User Defined Functions

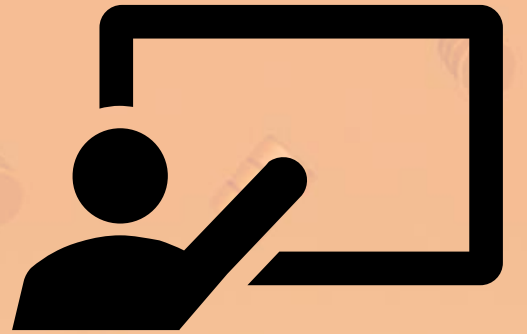
Activity & Data Flow updates

Script Activity

Besser mit Datenbanken können

- Kann einiges mehr als Lookup und Stored Procedure Activity (siehe rechts)
- Ein oder mehrere Statements
- Auch Rückgabe von Result sets sowie Abfrage-Meldungen

	Script Activity	Lookup Activity	SProc Activity
Supported data source	Database (SQL family, Snowflake, Oracle)	all data sources	SQL family
Supported Operations	Read / Modify	Read*	Modify
Multiple query support	Yes	No**	Yes***
Query parameter support	Input / Output	Not supported	Input
Output Result set support	One or more	One	No
Output query logs (PRINT)	Yes	No	No
Integrated CICD (ADF)	Yes	Yes	No



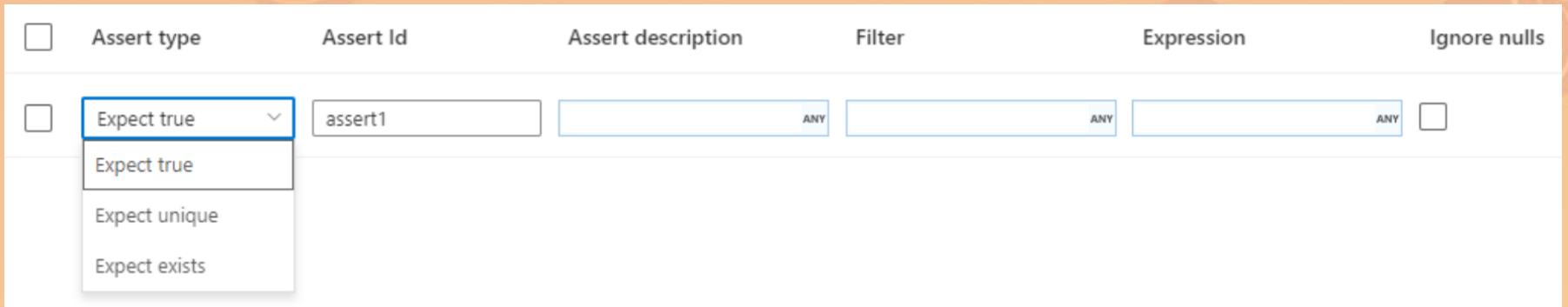
Demo

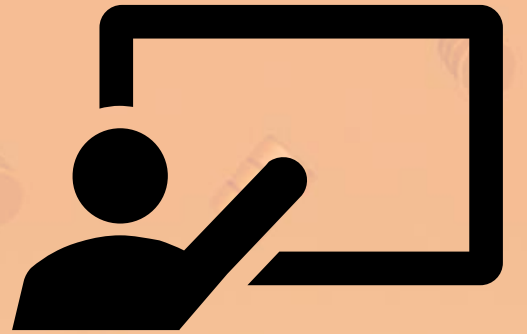
Script Task

Assert Transformation

Datenqualität inhaltlich checken im Data Flow

- Erstellen von Regeln die auf die Spalten im Flow angewendet werden
- Checken auf
  - Wertebereiche (Expect true)
  - Eindeutigkeit (Expect unique)
  - Vergleich zwischen Streams (Expect exists)
- Markieren der Zeilen oder Abbruch möglich





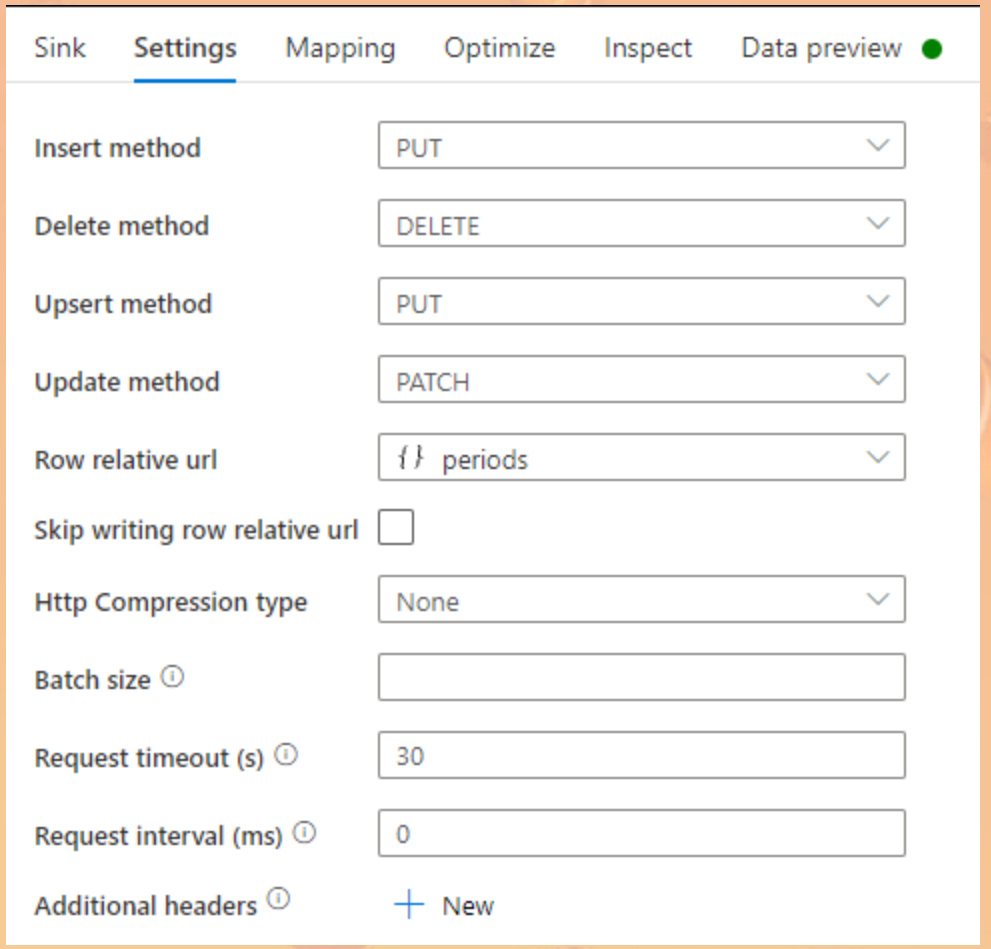
Demo

Assert Transformation

Rest Connector für Mapping Data Flows

Als Source und Sink einsetzbar

- requestMethod HTTP, Get & Post erlaubt
- integration datasets und inline datasets
- Relativ breiter Pagination Support
- Bei Verwendung als Sink ein Alter Row davorschalten um die Action zu bestimmen

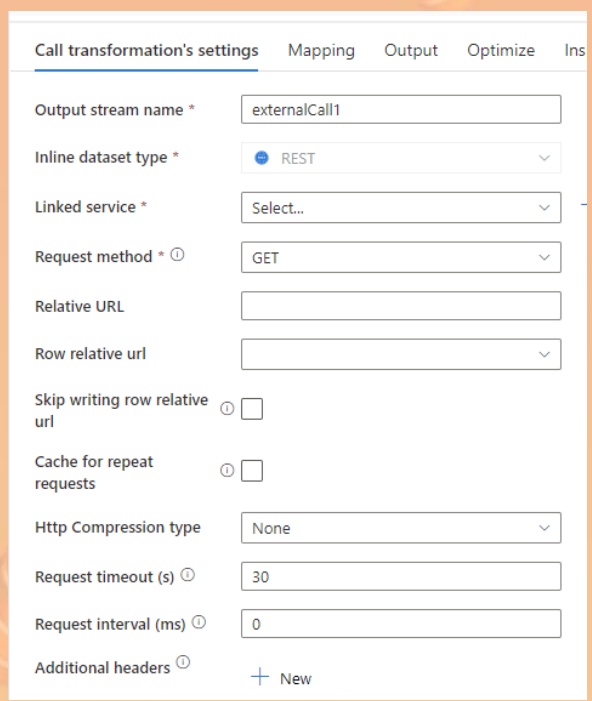


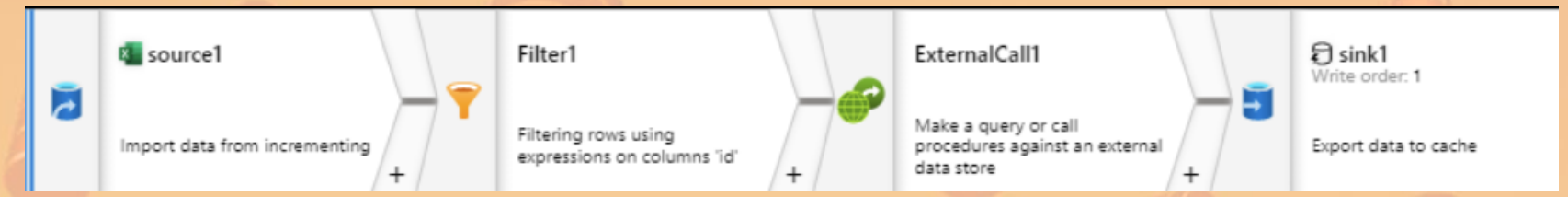
https://docs.microsoft.com/en-us/azure/data-factory/connector-rest?tabs=data-factory#mapping-data-flow-properties

External Call Transformation

Auch innerhalb eines Data Flows direkt Rest Services nutzen

- Einbinden externer Services im Zeilenmodus, z.B. Wetterdaten pro GPS Koordinate der Zeilen im Data Flow
- Wird noch erweitert auf Stored Procedures

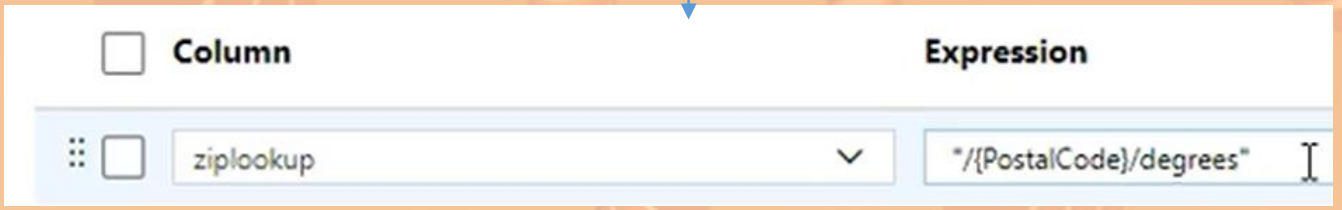




External Call Transformation

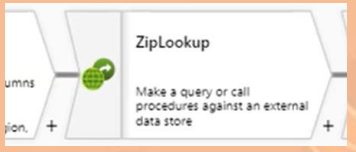
Am Beispiel Adressvalidierung

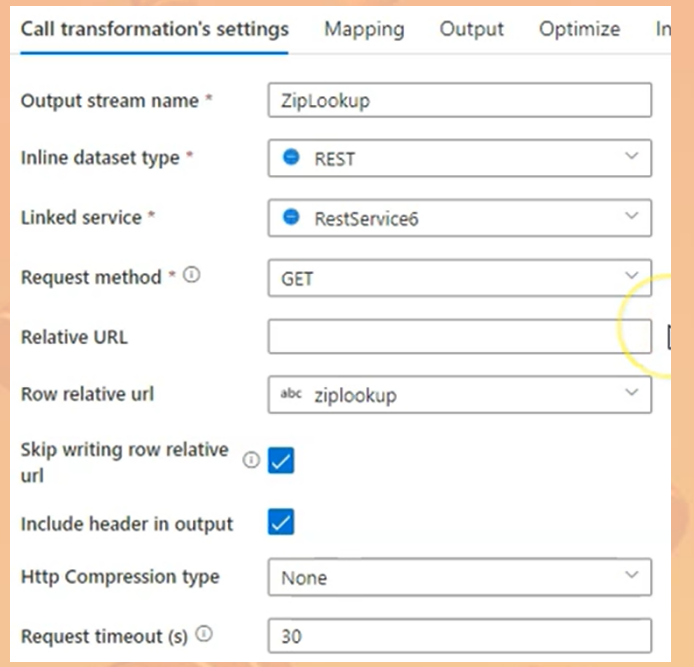
PostalCode abc
98011
98011
75201
85004
H1Y 2H5
98004

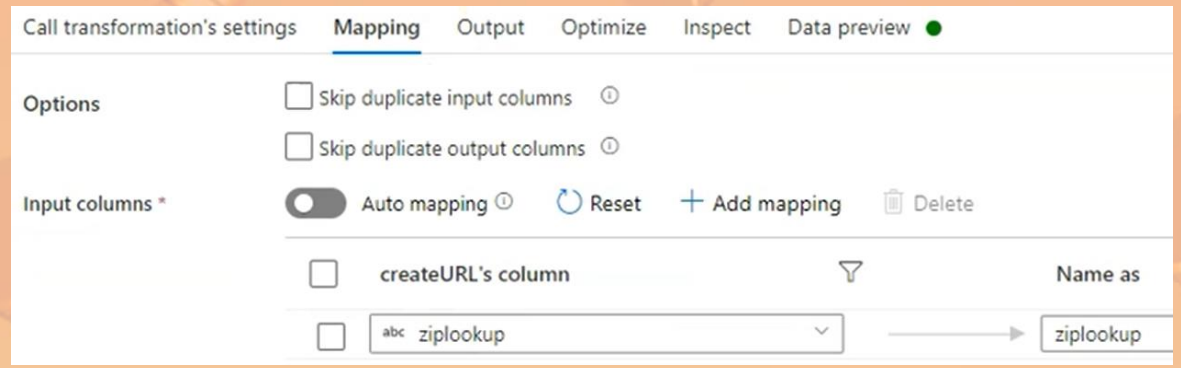


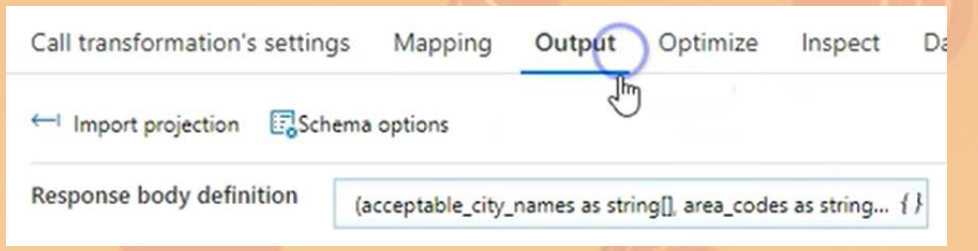
External Call Transformation

Konfiguration



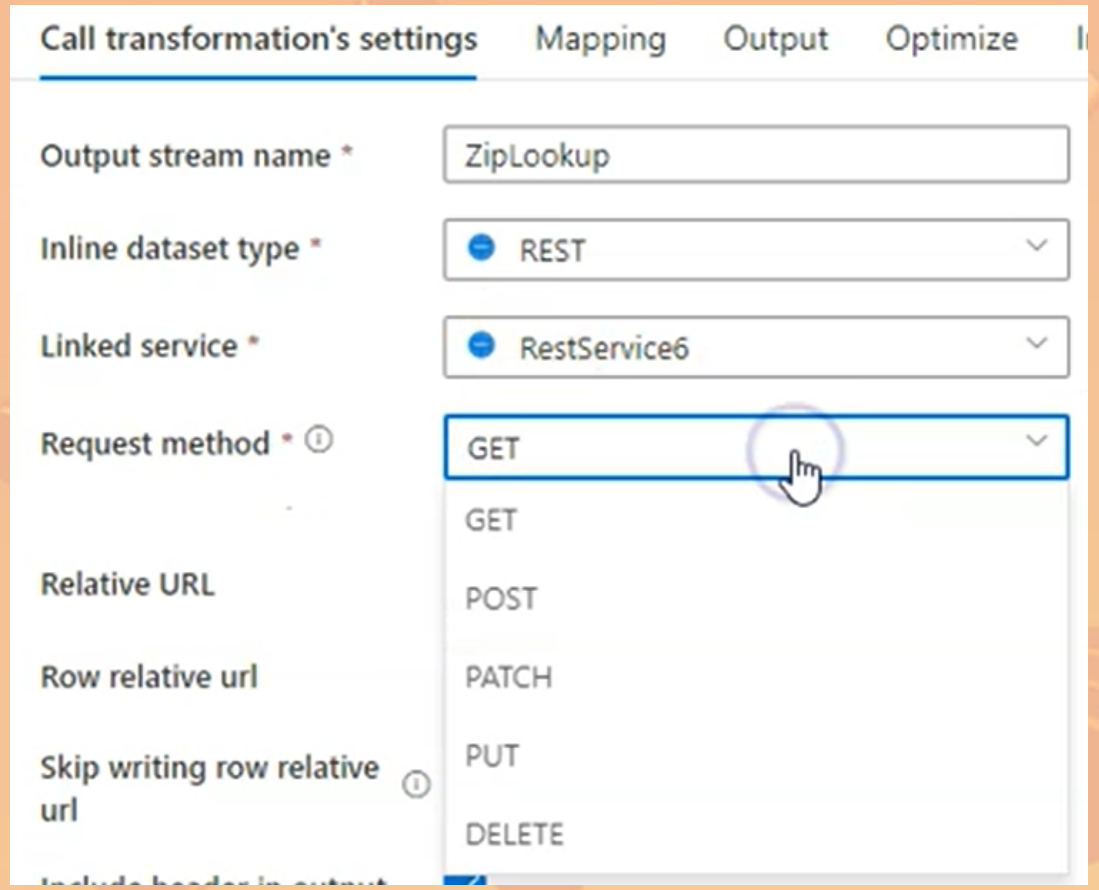






External Call Transformation

Unterstützte Rest Request Methoden



Achtung wird auch beim Preview ausgeführt (z.B. Delete)

Power Query

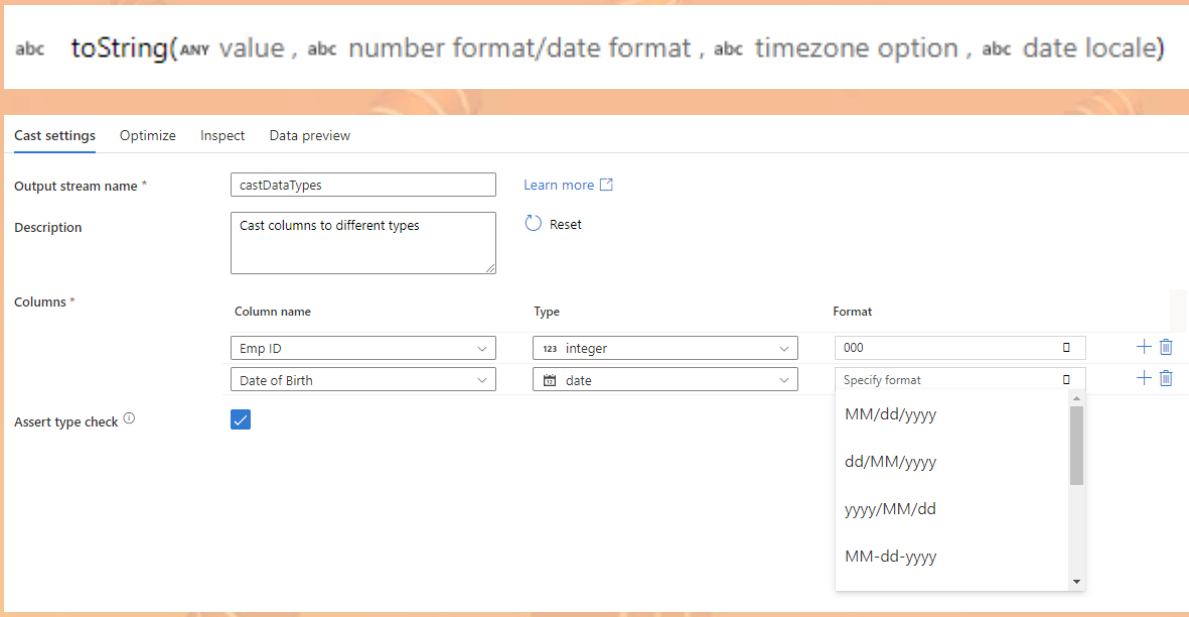
Aka Wrangling Data Flows

- Umbenannt, jetzt als „Power Query“ in ADF
- Wird Stück für Stück vervollständigt, Umstellung von Power BI Power Query bedeutet ja Spark hier im Backend
- Fühlt sich für mich trotzdem immer wie ein Fremdkörper an weil auch das Handling von Datasets anders ist usw.

Cast for Mapping Data Flows

Type conversions made easy

- Bislang über derived columns
- Bequemer in der GUI und expliziter
- Erfolgsprüfung über Assert möglich (type check)
- Komplexe Datentypen noch nicht möglich
- Anpassung ist nun auch direkt in der source projection möglich



https://learn.microsoft.com/en-us/azure/data-factory/data-flow-cast

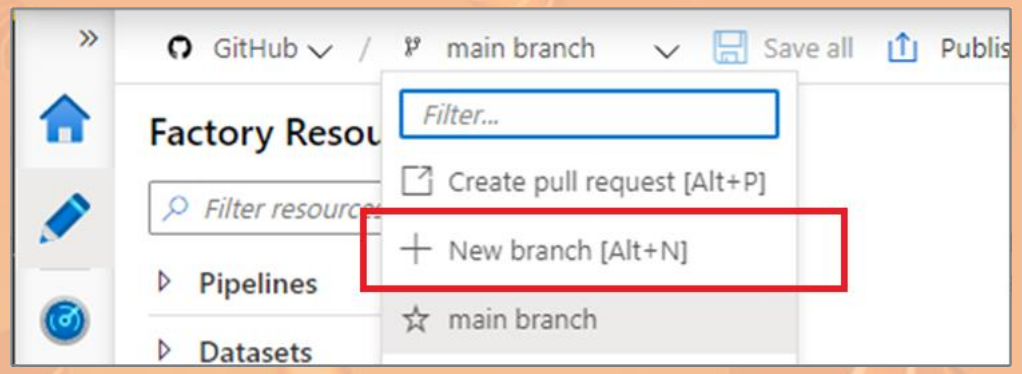
Application Lifecycle Management News

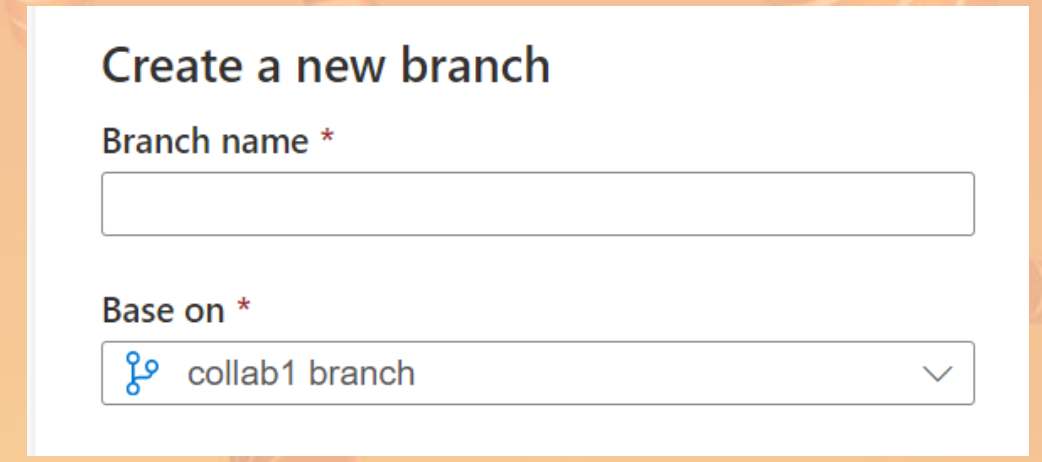
Build to survive

Azure Data Factory Github Integration

Limits entfallen & Branch of Branch

- Jetzt auch mehr als 1000 Data Factory Resourcen pro resource type möglich (gemeint sind datasets, pipelines & co.)
- Hilft bei der Arbeit mit Code Versioning & Deployments auch in Zusammenhang mit der GitHub API
- Es kann jetzt auch ein Branch basierend auf einem anderen Branch erstellt werden



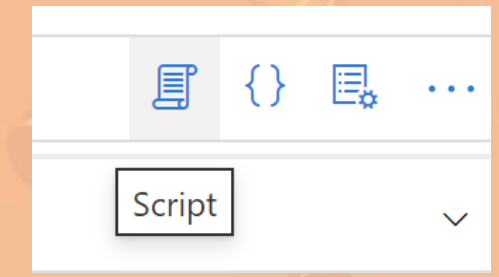


https://docs.microsoft.com/en-us/azure/data-factory/source-control#version-control

Intro Data Flow Script (DFS)

Programmiersprache hinter den Mapping Data Flows

- Data Flow Script auch Teil des Json Codes (ganz unten)
- Entsteht aber beides automatisch durch die grafische Oberfläche
- Editierbar
- Inhalte übertragbar in andere Data Flows / Projekte
- Verwenden in Powershell cmdlets
- Einfaches Find & Replace im Text
- Programmatisch Data Flows erzeugen aus Metadaten
- Einfacheres Finden von Abweichungen zwischen Versionen – vereinfacht auch in Json seit der Einführung von scriptline
- Viele Beispiele von MS oder Blog-Artikeln können einfach integriert werden




```
source(output(
        medallion as string,
        hack_license as string,
        vendor_id as string,
        rate_code as string,
        store_and_fwd_flag as string,
        pickup_datetime as string,
        dropoff_datetime as string,
        passenger_count as short,
        trip_time_in_secs as long,
        trip_distance as double,
        pickup_longitude as double,
        pickup_latitude as double,
        dropoff_longitude as double,
        dropoff_latitude as double
    ),
    allowSchemaDrift: false,
    validateSchema: false,
    ignoreNoFilesFound: false) ~> TripData
source(output(
        medallion as string,
        { hack_license} as string,
        { vendor_id} as string,
```

https://docs.microsoft.com/en-us/azure/data-factory/data-flow-script

Intro Data Flow Script (DFS)

Scriptline in Json Code Definition

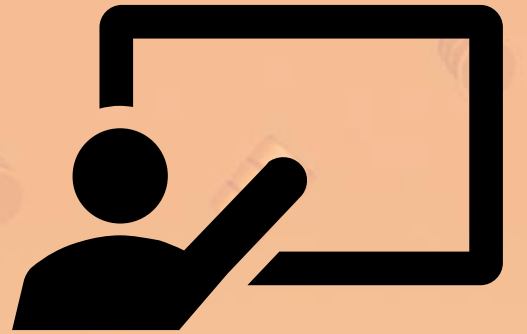
- JSON Definition in ADF für Data Flow früher: c

```
        }
      ],
      "script": "source(output(\n\t\tmedallion as string,\n\t\thack_license as string,\n\t\tvendor_id as string,\n\t\trate_code as string,\n\t\tstore_and_fwd_fl
```

- JSON Definition in ADF für Data Flow seit Einführung scriptline:

```
],
"scriptLines": [
    "source(output(",
    "          {Emp ID} as string,",
    "          {Name Prefix} as string,",
    "          {First Name} as string,",
    "          {Middle Initial} as string,",
    "          {Last Name} as string,",
    "          Gender as string,",
    "          {E Mail} as string,",
    "          {Father's Name} as string,",
    "          {Mother's Name} as string,",
    "          {Mother's Maiden Name} as string,",
    "          {Date of Birth} as string,",
    "          {Time of Birth} as string,",
```

https://techcommunity.microsoft.com/t5/azure-data-factory-blog/adf-mapping-data-flows-adds-scriptlines-and-link-service/ba-p/3249929#M589



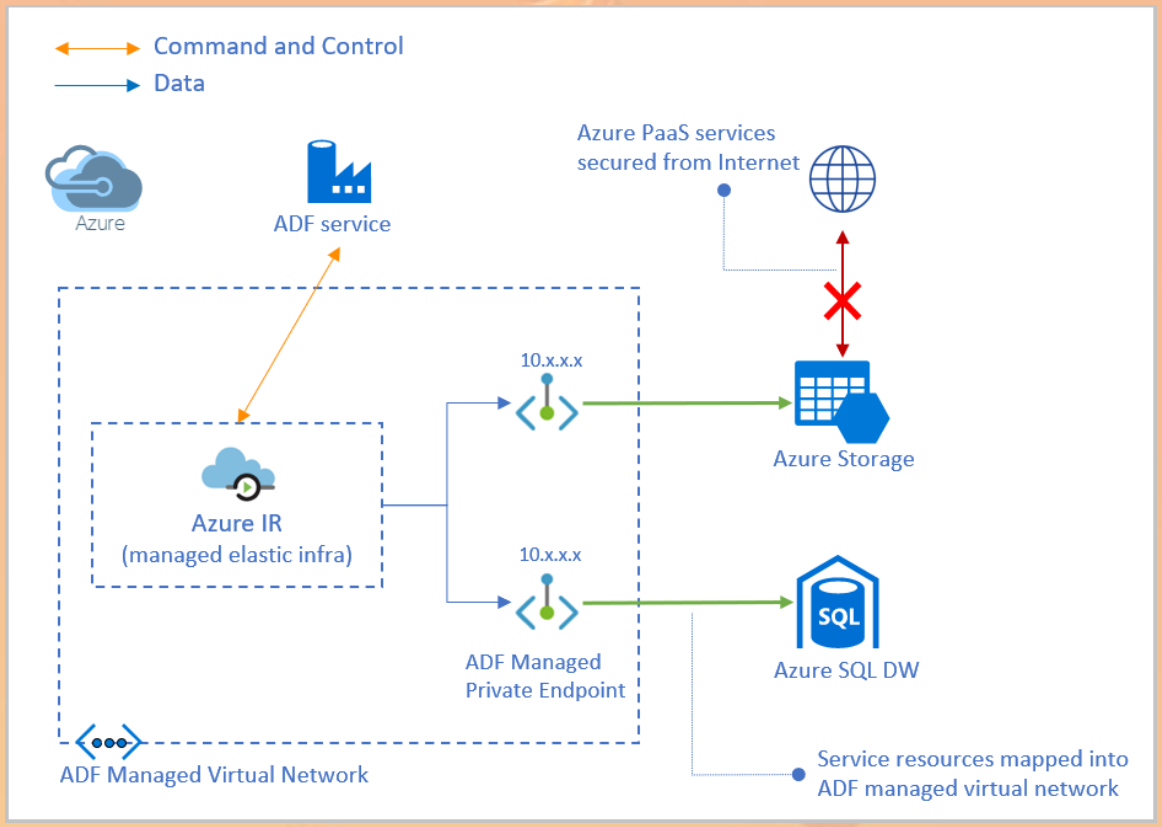
Demo

Data Flow Script

Definieren der Time to Live in managed VNET Integrated mode

und warum das in dem Fall so wichtig ist

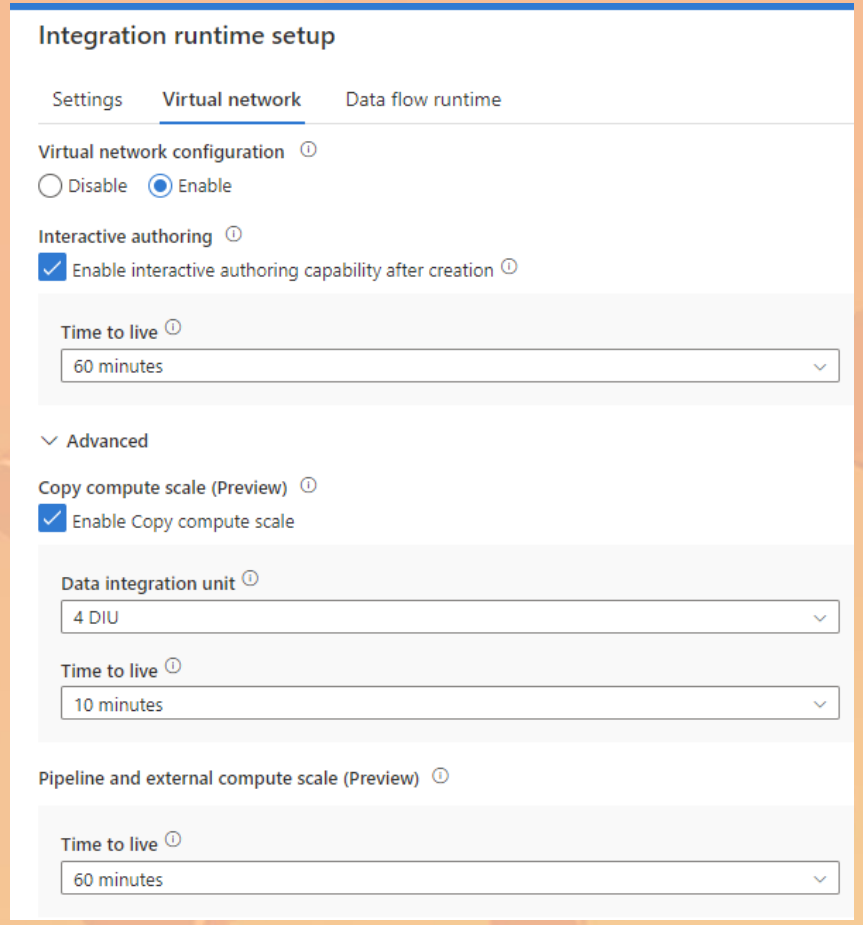
- Managed VNET kapselt die ADF vom öffentlichen Internet zur weiteren Sicherheit ab
- Aber: längere Ausführungszeiten weil keine Standby Resourcen nutzbar sondern extra welche hochgefahren werden, das dauert seine Zeit (Faktor 5)
- Wichtig die TTL für diese Resourcen zu definieren damit diese dann von Prozessketten wiederverwendet werden können



https://learn.microsoft.com/en-us/azure/data-factory/managed-virtual-network-private-endpoint

https://techcommunity.microsoft.com/t5/azure-data-factory-blog/azure-vnet-vs-azure-ir-inside-of-pipelines/ba-p/3606758
https://techcommunity.microsoft.com/t5/azure-data-factory-blog/announcing-public-preview-of-time-to-live-ttl-in-managed-virtual/ba-p/3552879

Definieren der Time to Live in managed VNET Integrated mode



Change Data Capture support extended

I want it (not) all – incremental sourcing

Mehr Quellen unterstützt

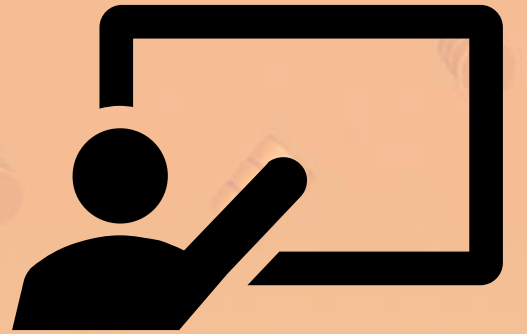
- CDC ist praktisch um nur Änderungen zu verarbeiten, verschiedene Quellen unterstützt
- Full CDC hat zusätzlich Infos zu for inserts, upserts, deletes, and updates für Datensätze
- Unterstützung jetzt auch nativ für SQL DBs in Mapping Data Flows (Ignite 2022)

CDC Connector support

Connector	Full CDC	Incremental CDC
ADLS Gen1		✓
ADLS Gen2		✓
Azure Blob Storage		✓
Azure Cosmos DB (SQL API)	✓	✓
Azure Database for MySQL		✓
Azure Database for PostgreSQL		✓
Azure SQL Database	✓	✓
Azure SQL Managed Instance	✓	✓
Azure SQL Server	✓	✓
Common data model		✓
SAP CDC	✓	✓

https://learn.microsoft.com/de-de/azure/data-factory/concepts-change-data-capture

https://techcommunity.microsoft.com/t5/azure-data-factory-blog/data-integration-news-from-ignite-2022-sap-cdc-m365-datasets-and/ba-p/3644476



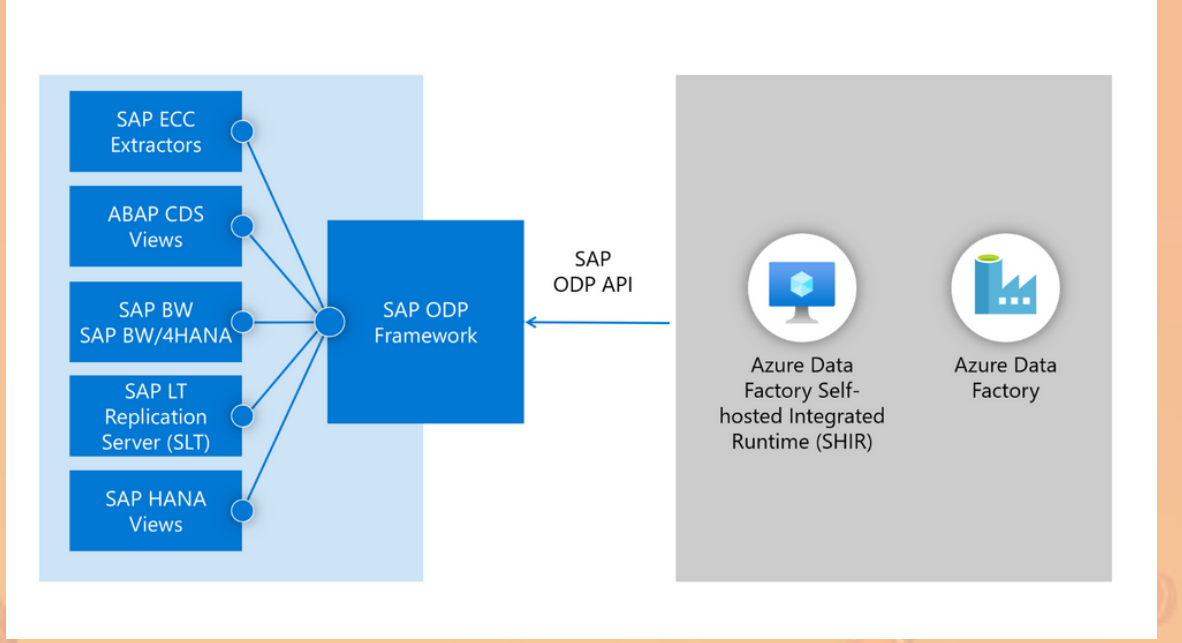
Demo

CDC auf Azure Data Lake

SAP CDC ist general available

Verwenden von SAP eigener Change Tracking Mechanismen für Delta Loads

- new SAP CDC connector leveraging SAP ODP framework
- can connect to all SAP systems that support ODP, such as R/3, ECC, S/4HANA, BW, and BW/4HANA
- directly at the application layer or indirectly using SAP Landscape Transformation (SLT) replication server as a proxy
- fully or incrementally extract SAP data
- physical tables & logical objects such as ABAP Core Data Services (CDS) views
- in GA since October 2022



https://techcommunity.microsoft.com/t5/azure-data-factory-blog/announcing-the-upcoming-preview-of-sap-cdc-in-azure-data-factory/ba-p/3420904

Vielen Dank!

Ich freue mich auf Feedback!