PASS Regionalgruppe Bayern
12.03.2020

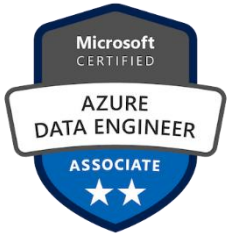
Data engineering pattern in der Azure Data Factory

Stefan Kirner

About me

Stefan Kirner

- PASS Chapter Lead Karlsruhe ski@sqlpass.de
- Teamlead BI Solutions @inovex
- Microsoft P-TSP Data Platform
- Twitter: @KirnerKa



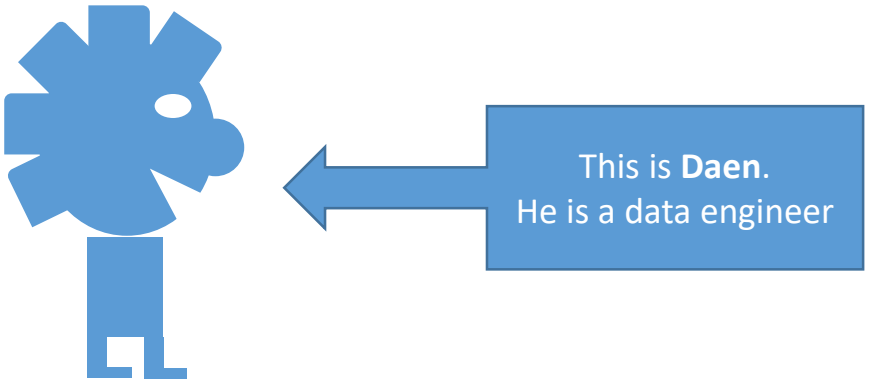
Agenda

- The data challenge
- The different data engineering pattern
- The control flow
- Best practices + Q&A

The data challenge

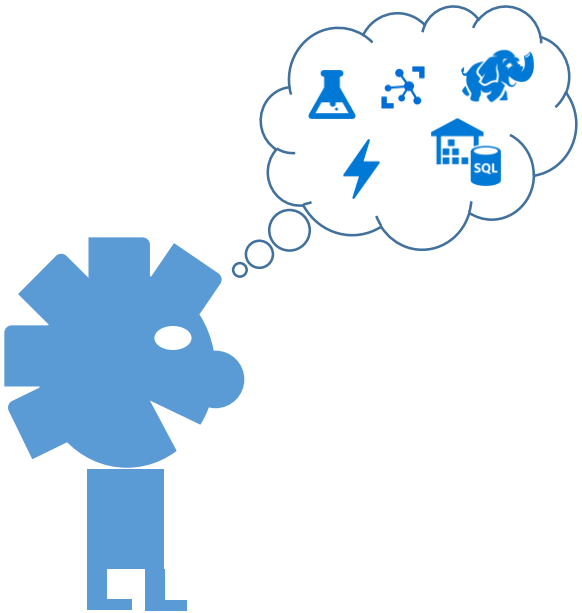
What is „data engineering“?

“**preparing data** for analytical or operational uses. The specific tasks [..] typically include building **data pipelines** to pull information from different source systems together; **integrating, consolidating and cleansing data**[..]



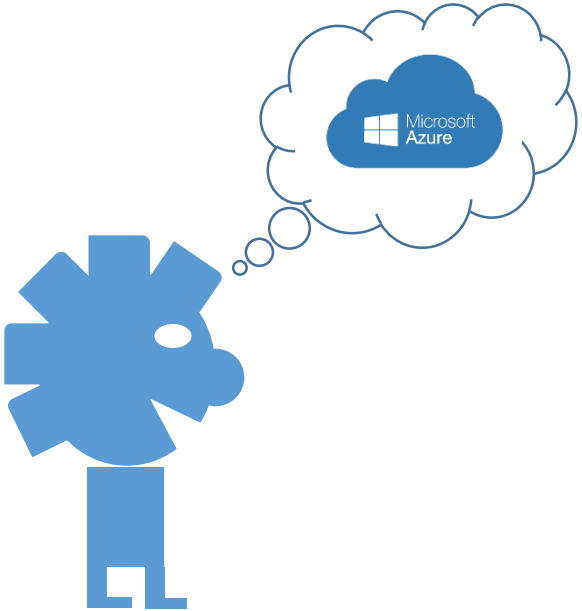
https://searchdatamanagement.techtarget.com/definition/data-engineer

Daen thinks about a new data platform



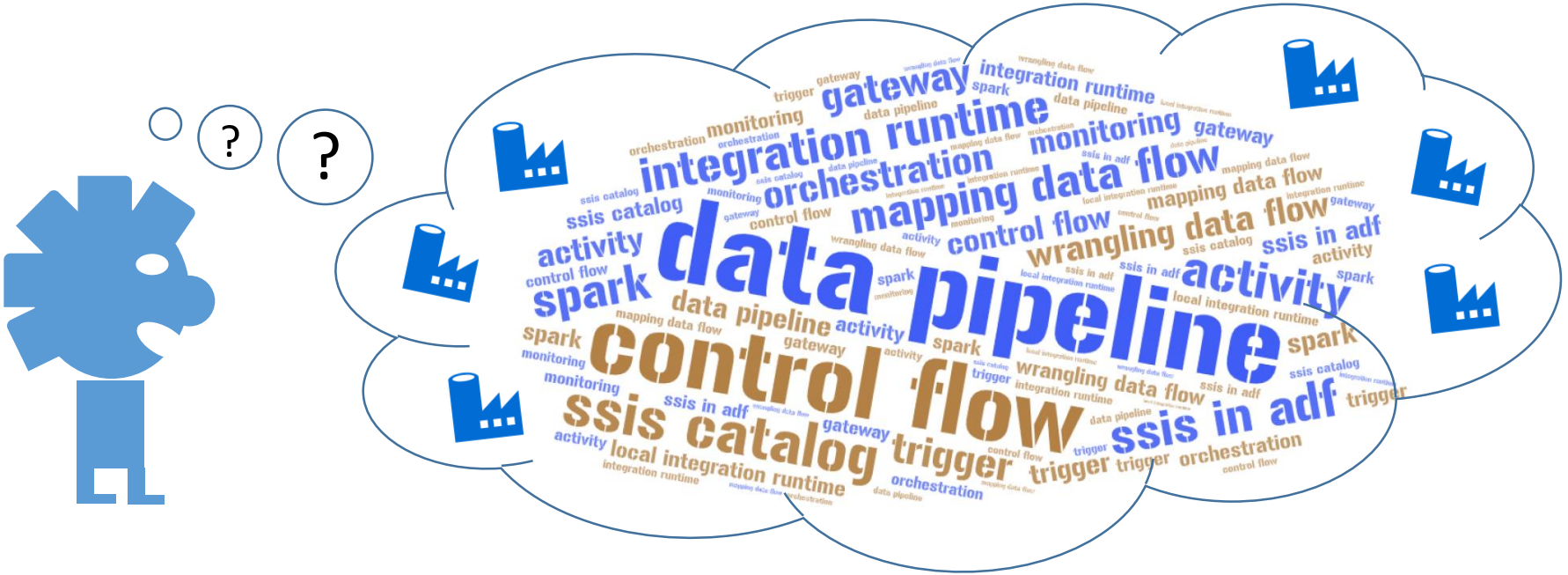
- structured & unstructured data
- mass data & messed up data
- integrate existing BI systems
- on-prem & cloud based source data
- less staff for IT system engineering

Why using managed services on a cloud platform?



- Scalability to infinity
- Cutting edge technologies
- Relaxed IT staff
- Agile Setup
- Increased reliability
- GO FAST!

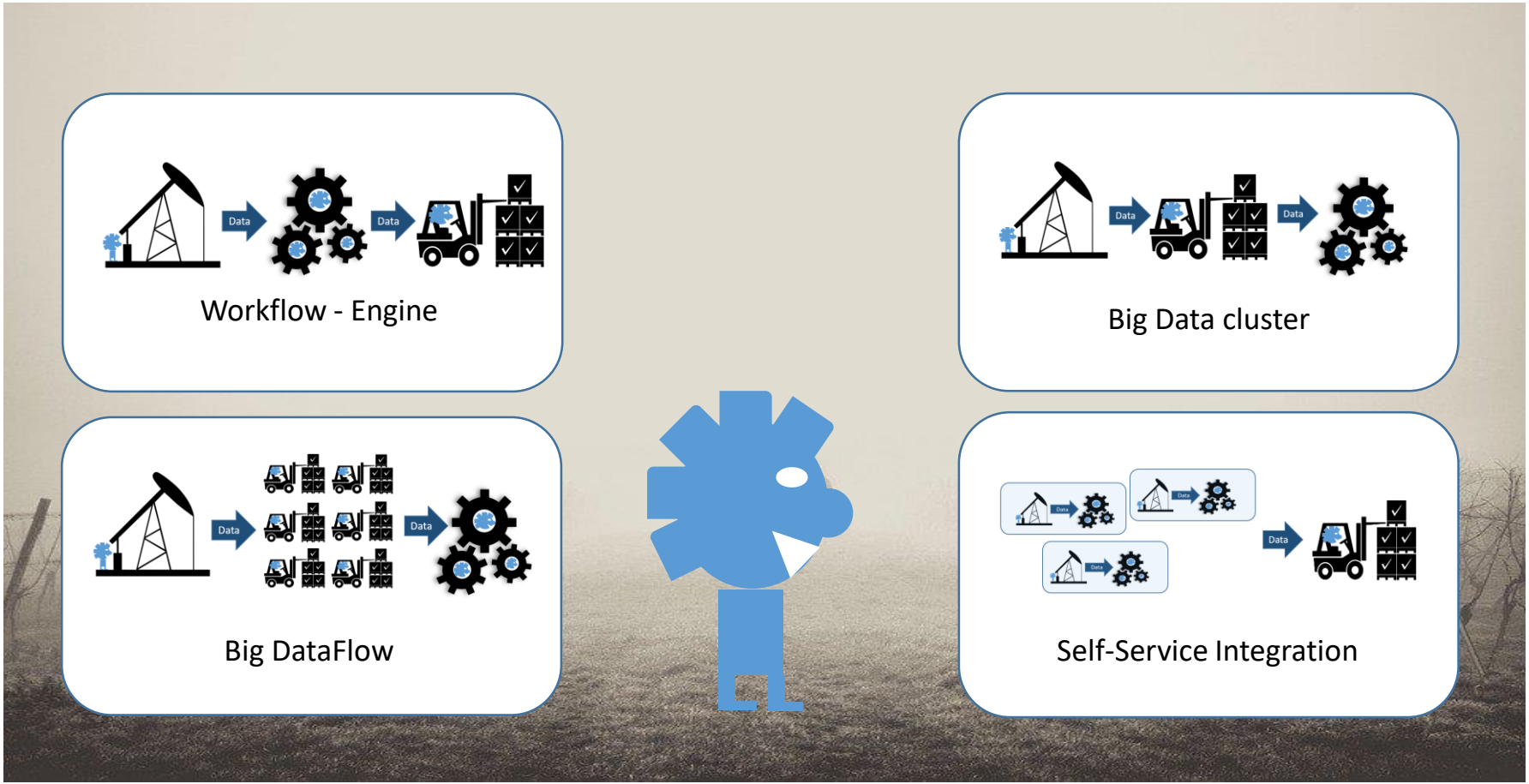
Azure Data Factory - Data Management on Azure



So many ways…

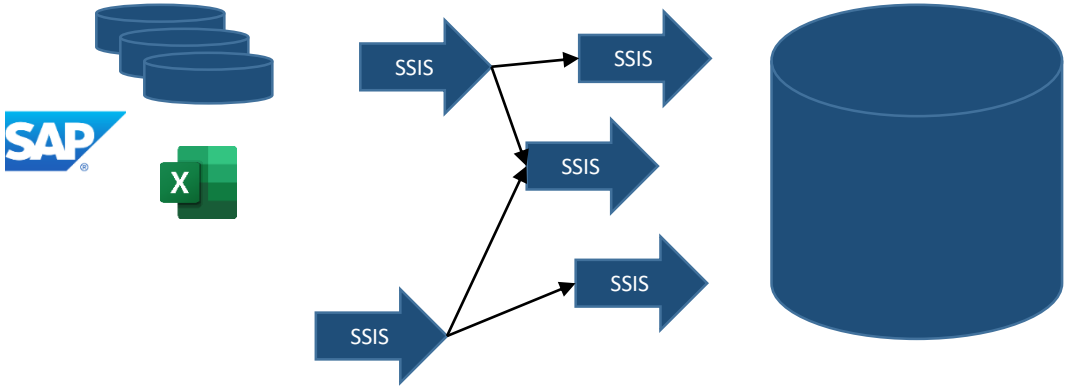
…which is the best?

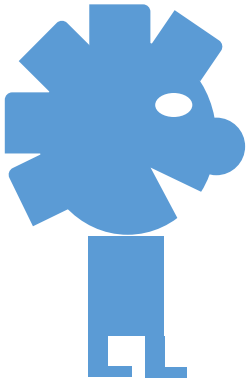
Some pattern visible in the fog



Workflow - Engine

Use Case 1: Migrate Existing Microsoft BI systems

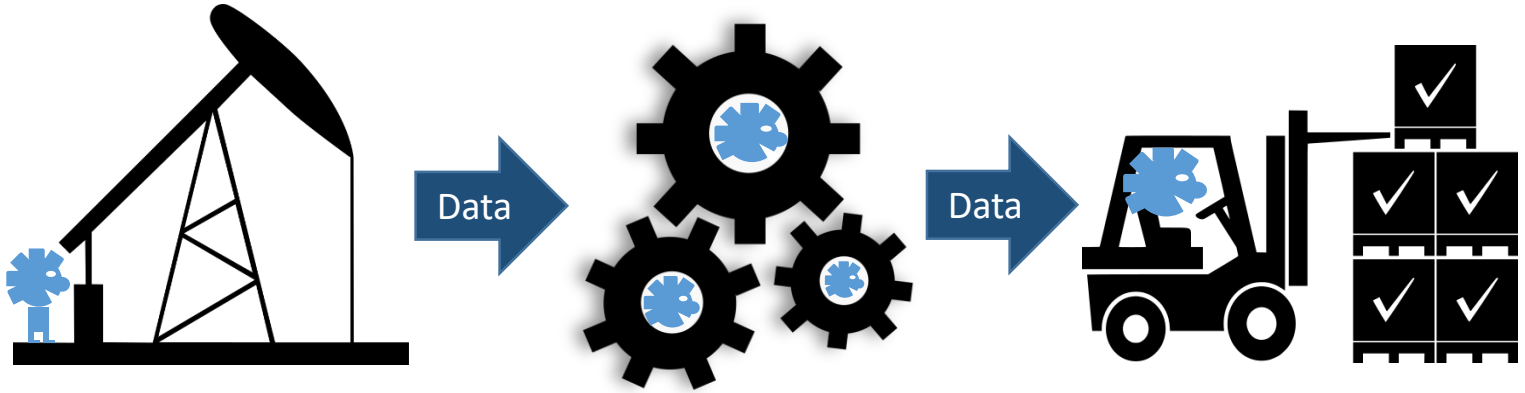




Daen has to:

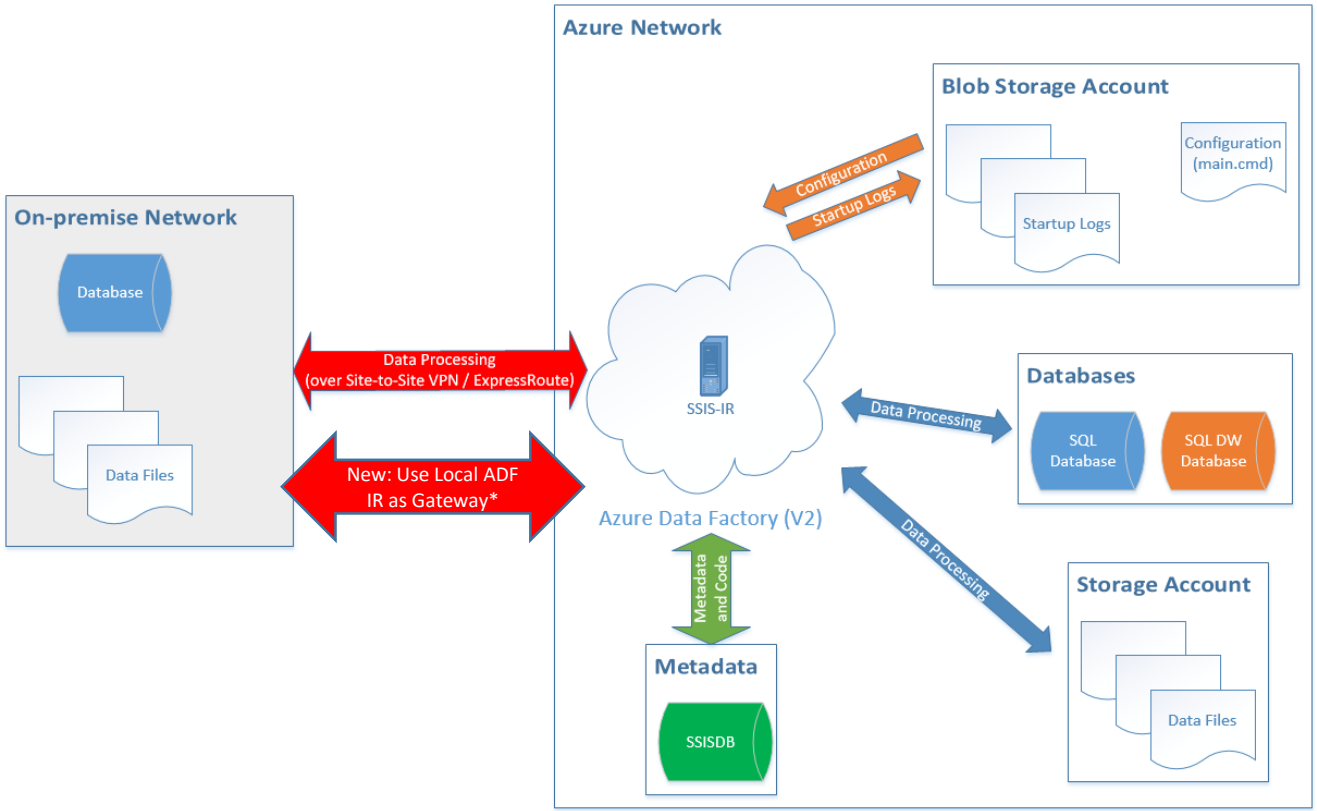
- Run logic of 1000+ SSIS packages
- Reuse deep SSIS Know-How of his team
- Bring workload to Azure with less effort

Workflow Engine = Integration Services on Azure



All logic inside Integration Services Workflow Engine
Extraction-Transformation-Load Scheme

Integration Services in ADF - components overview

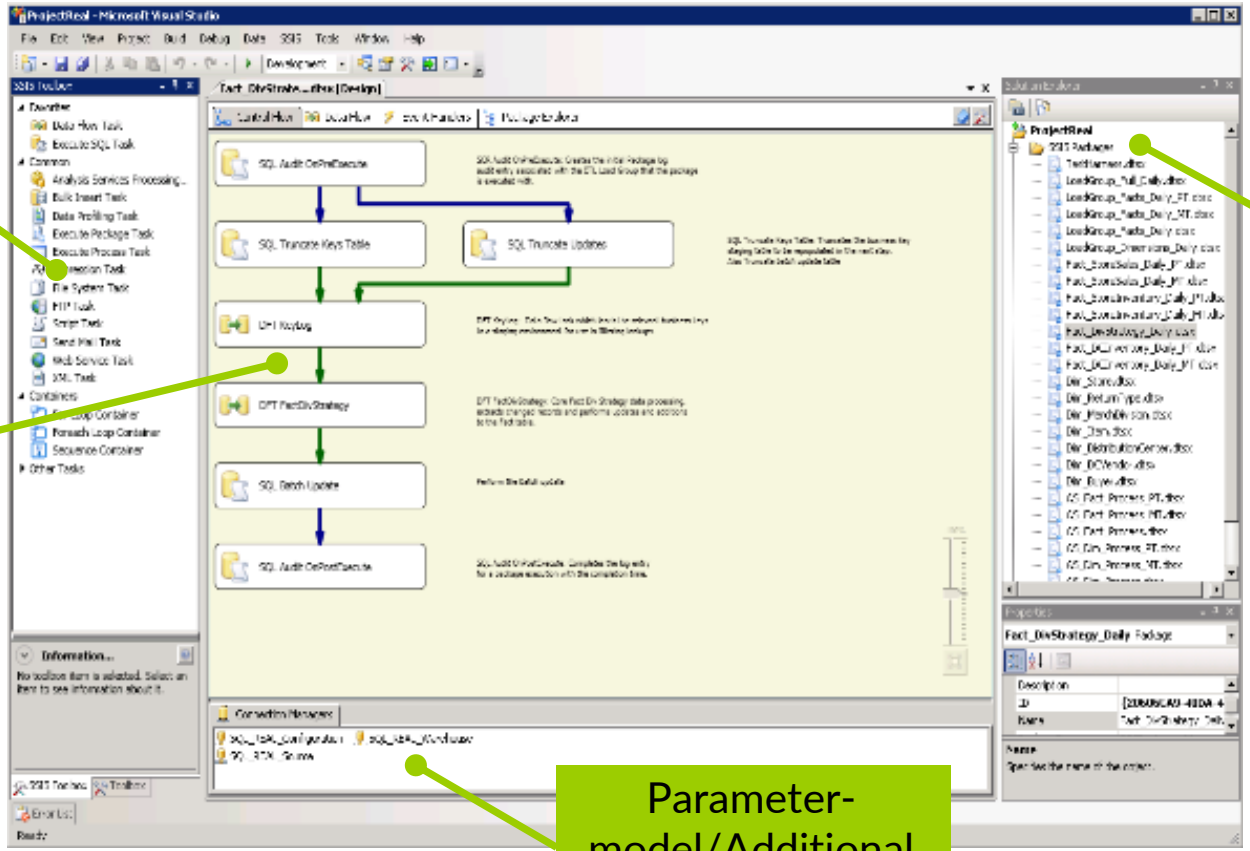


https://blog.westmonroepartners.com/the-new-azure-data-factory-now-with-ssis-compatibility/
* https://docs.microsoft.com/en-us/azure/data-factory/self-hosted-integration-runtime-proxy-ssis

Integration Services Development

Pre-defined elements for typical tasks

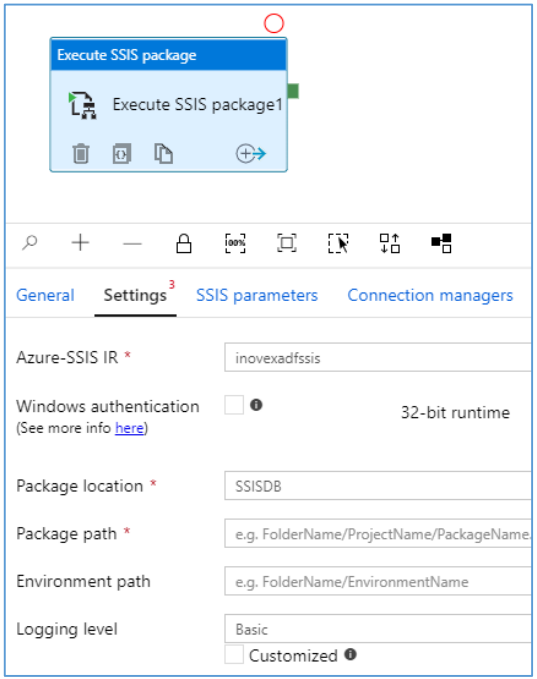
Canvas: Undo, Toolbox, Wizards, Autosave, Comments

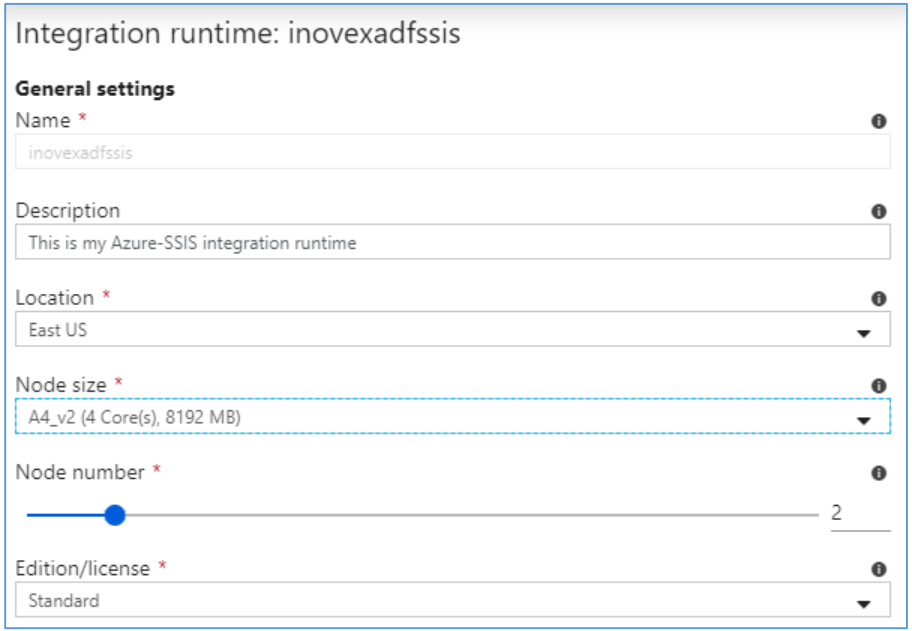


Project-model for easy deployment

Parameter-model/Additional to variables

Integration Services in Azure Data Factory





Good & bad about workflow engine pattern?

- Stability ★★
  - No childhood deseases (+)
  - Breaks on source data type changes (-)
- Maturity ★★★★
  - 15 years old, much used, many skilled developers (+)
- Supported data sources & sinks ★★★★
  - Many supported data sources and additional 3rd party components available (+)
  - Difficult to integrate semi-structured data (-)

Good & bad about workflow engine pattern?

- Lifecyle management support ★★★★
  - Visual Studio Projects for builds & source code management (+)
  - Useful environments (dev, test, prod) in SSMS (+)
  - Monitoring included in SSMS (+)
- Scalability ★★
  - Scale-out option only distributes packages (-)
  - Limited by resources of runtime computer (-)

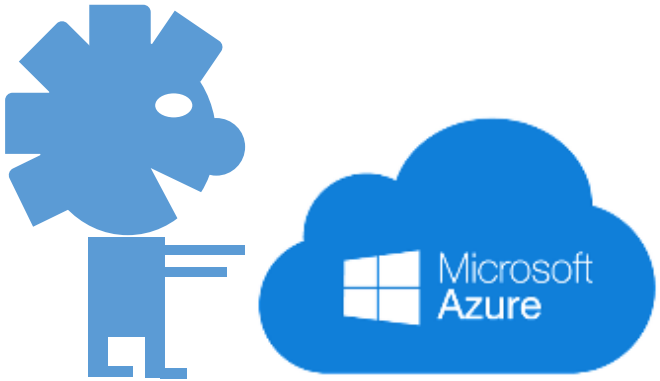
Good & bad about workflow engine pattern?

- Learning curve ★★★
  - GUI has high learn curve (+)
  - is self-documenting (+)
  - data viewers useable, but no real WYSIWYG mode (-)

Best use cases for workflow engine

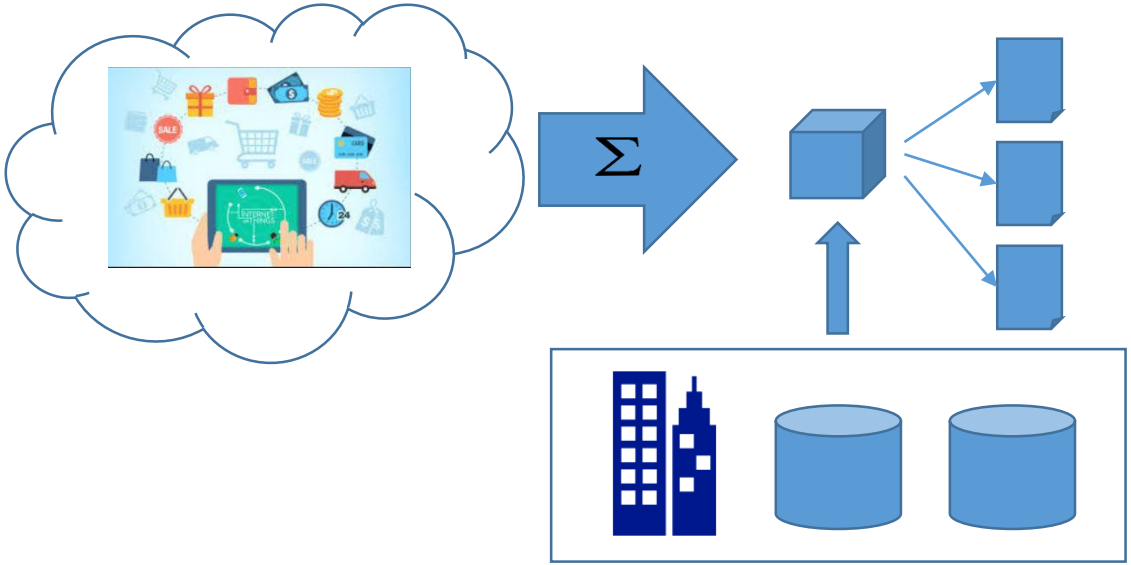
- Lift & Shift scenarios of MS BI on-prem systems using SSIS
- Small & medium data volumes
- Well structured (relational) data
- Minor schema changes
- Deterministic usage of data (e.g. for this report)

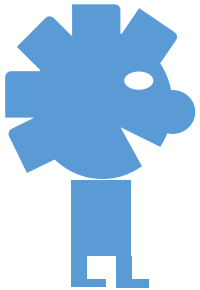
Demo – SSIS in ADF



Big data cluster pattern

Use Case 2: cloud born mass data

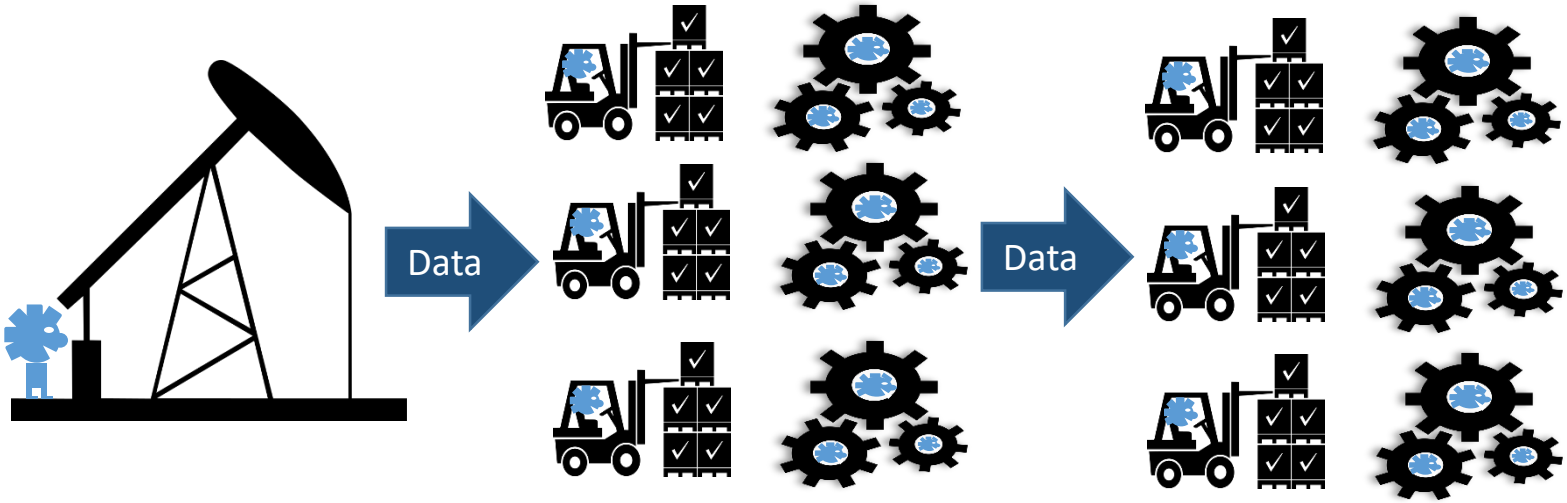




Daen has to:

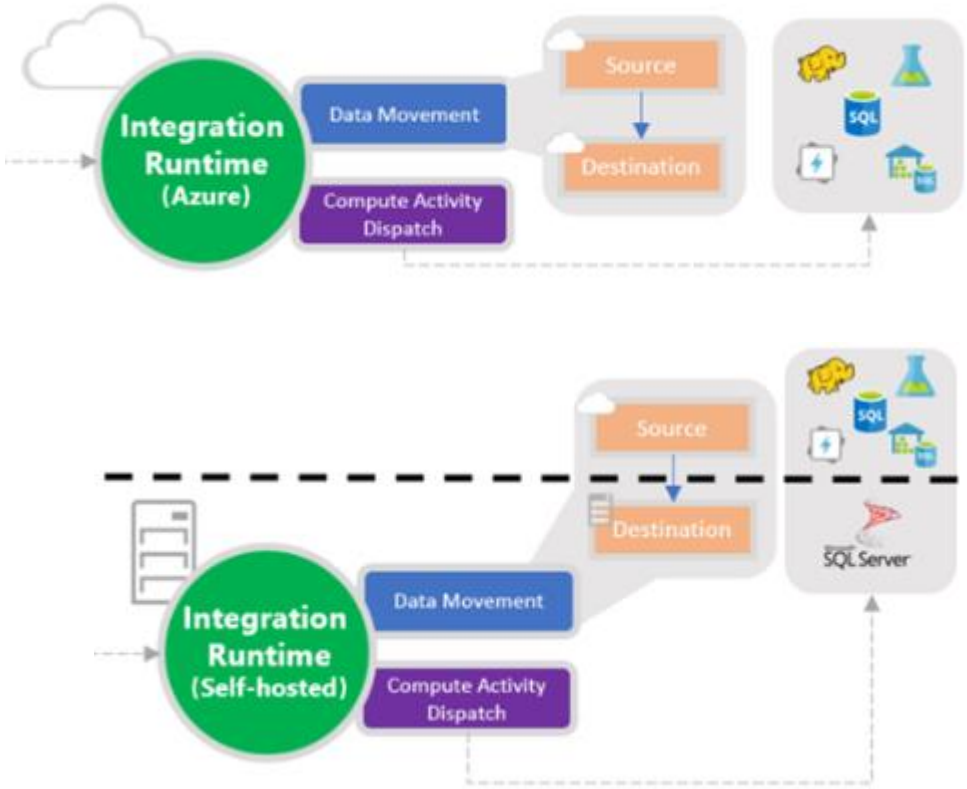
- Extract cloud-born mass data
- Aggregate data
- Combine with on-prem data
- Prepare data for analysis

Big data cluster – Let the engine do the work

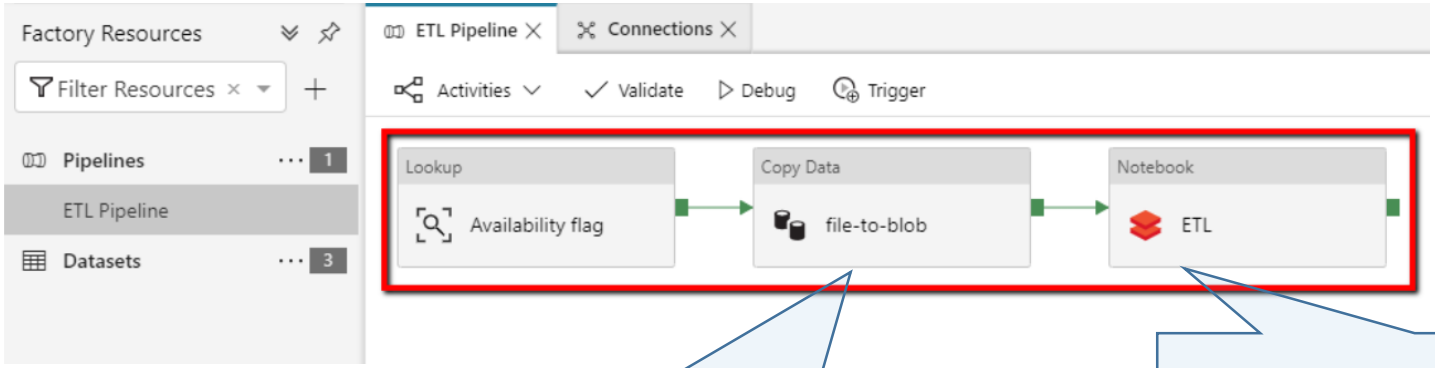


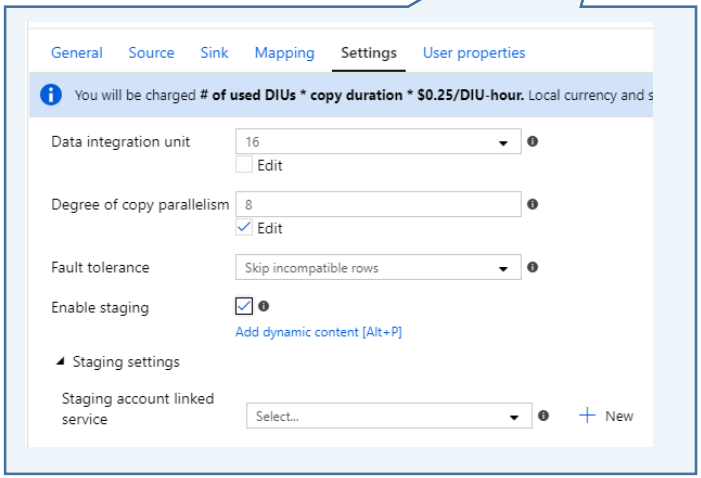
Copy to destination, Logic in external processing engine like Hadoop, Spark, SQL Server procedures…

Resources in Azure Data Factory



Big data cluster development





Good & bad about big data cluster pattern?

- Stability ★★★★
  - Data type changes do not break loads - schema on read (+)
- Maturity ★★★
  - Hadoop & spark clusters are kind of commodity today (+)
  - Azure data factory copy activity well known (+)
  - Less good skilled staff available than for SSIS (-)
- Supported data sources & sinks ★★★
  - Not as many as SSIS but many (+)

Good & bad about big data cluster pattern?

- Lifecyle management support ★★★
  - Using different tools for copy / transform (-)
  - No visual studio support for ADF, no builds (-)
  - GUI transformations easier to understand for operator (-)
  - More storage layers needed (-)
  - Source code integration in Azure portal (+)
- Scalability ★★★★
  - Easy scale-out for transformations (+)
  - and data copy tasks (+)

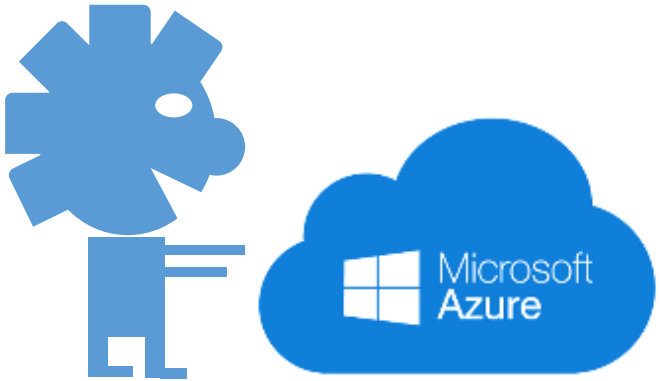
Good & bad about big data cluster pattern?

- Learning curve ★★
  - High for ADF Copy Tasks (+)
  - Low for programming / scripting in PySpark / Java etc. (-)
  - No WYSIWYG Editor (-)
  - Development in Notebooks could help documenting (+)

Use cases for Big Data clusters

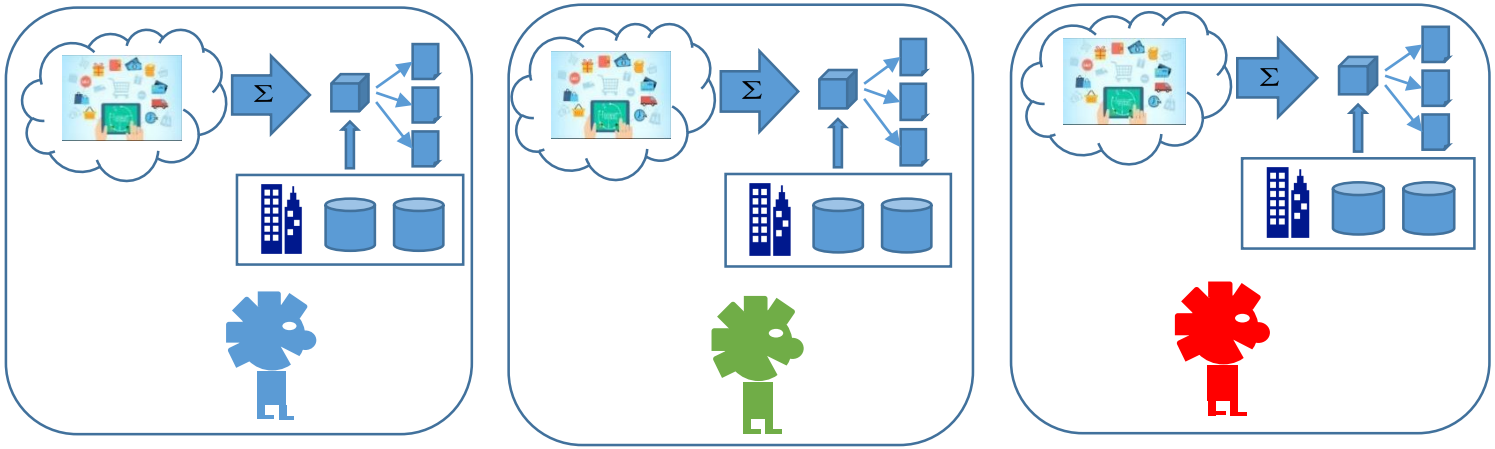
- High volume of data
- Partitioning possible by attributes
- Complex data sources
- Many different use cases for the data
- „Agile“ source systems with often changing schemas

Demo – Big Data Clusters in ADF



Big DataFlow

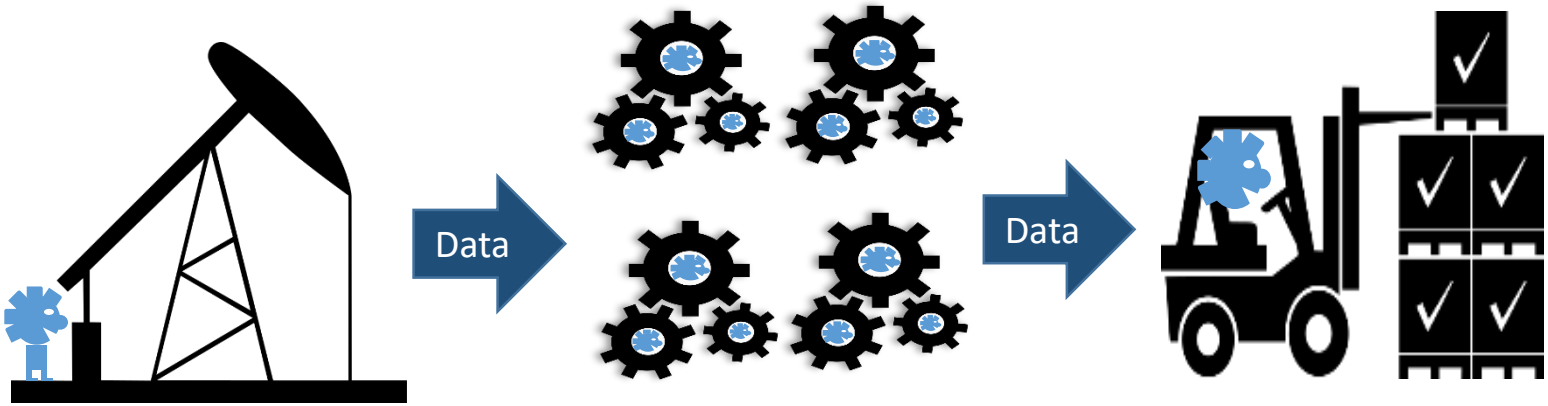
Use Case 3: **Many** cloud born mass data stores



Daen and his collegues have to:

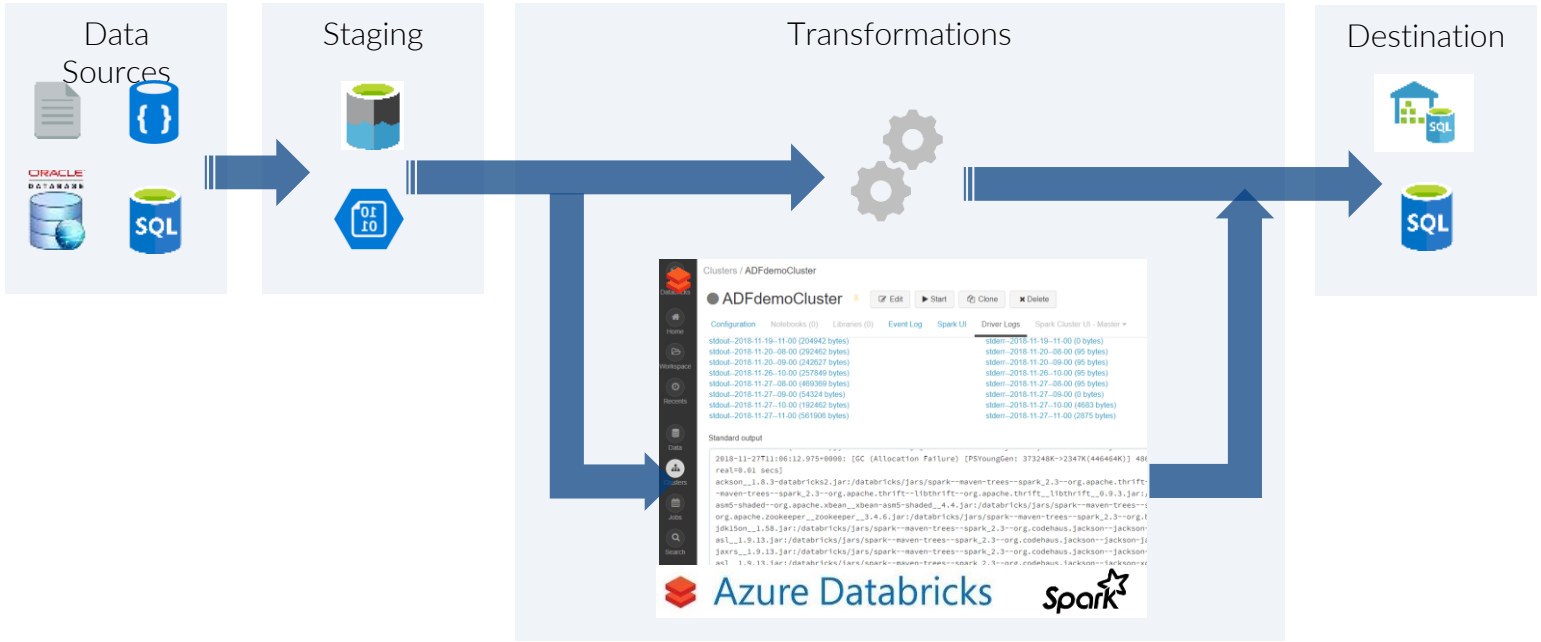
- Extract more external mass data
- Aggregate & Combine & Prepare
- **Less software engineering experience in staff**
- **Self-documenting and manageable**

Big Dataflow – Mapping dataflows in ADF



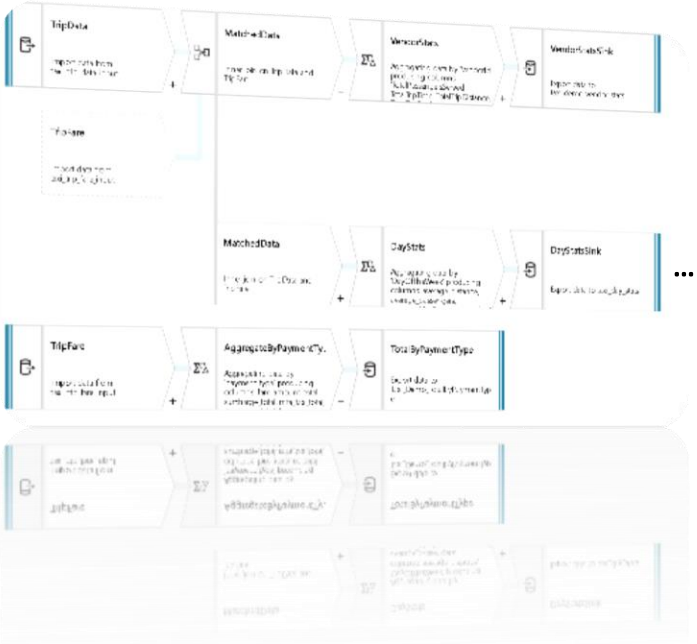
Business Logic complete in Azure Databricks external processing engine

ADF Data Flow Overview



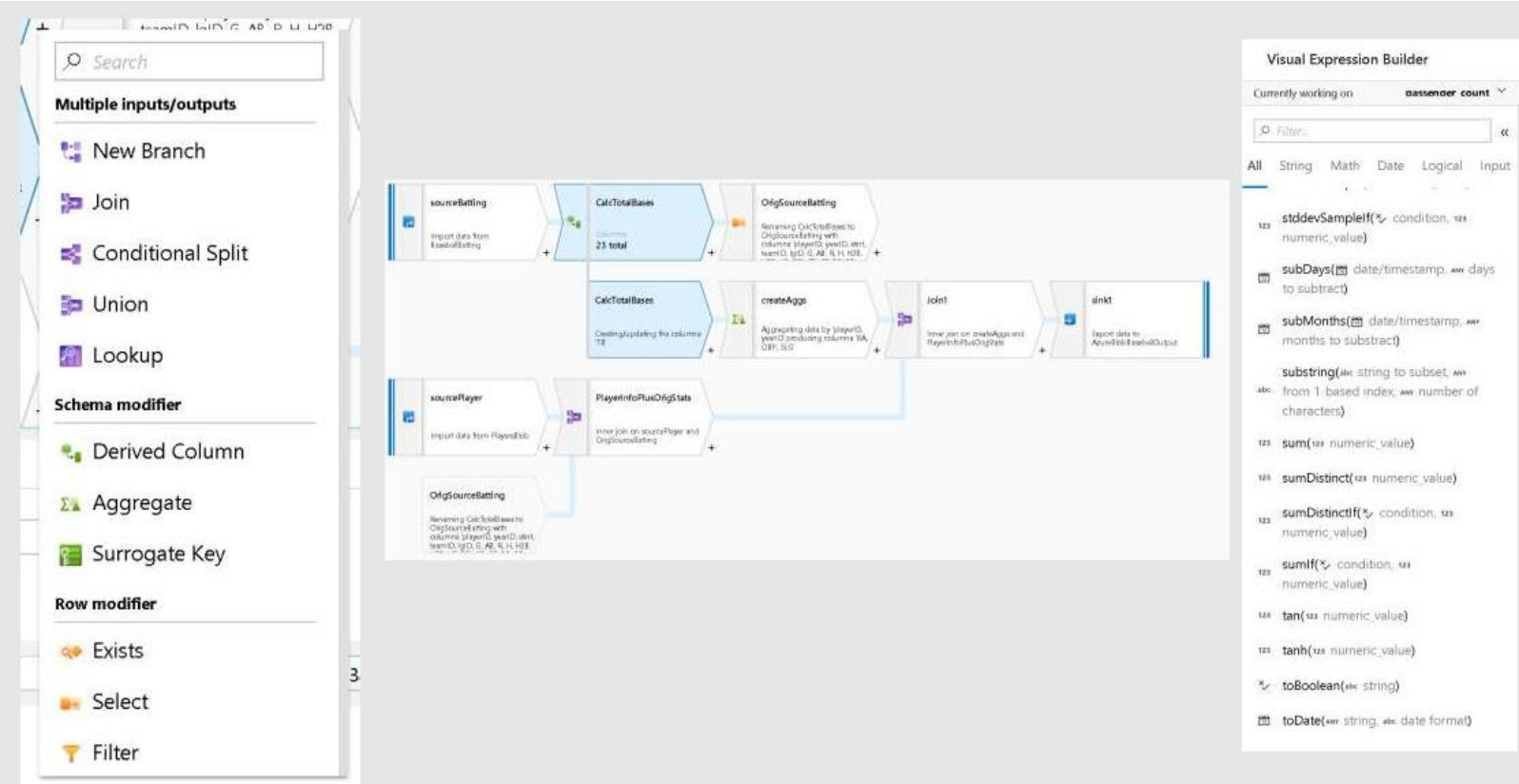
Code-free Data Transformation At Scale

- Does not require understanding of Spark, Big Data Execution Engines, Clusters, Scala, Python …
- Focus on building business logic and data transformation
  - Data cleansing
  - Aggregation
  - Data conversions
  - Data prep
  - Data exploration

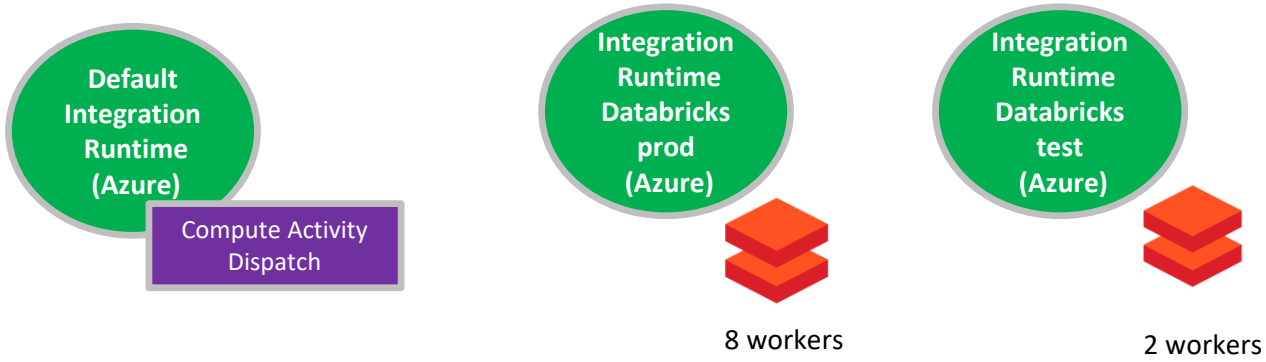


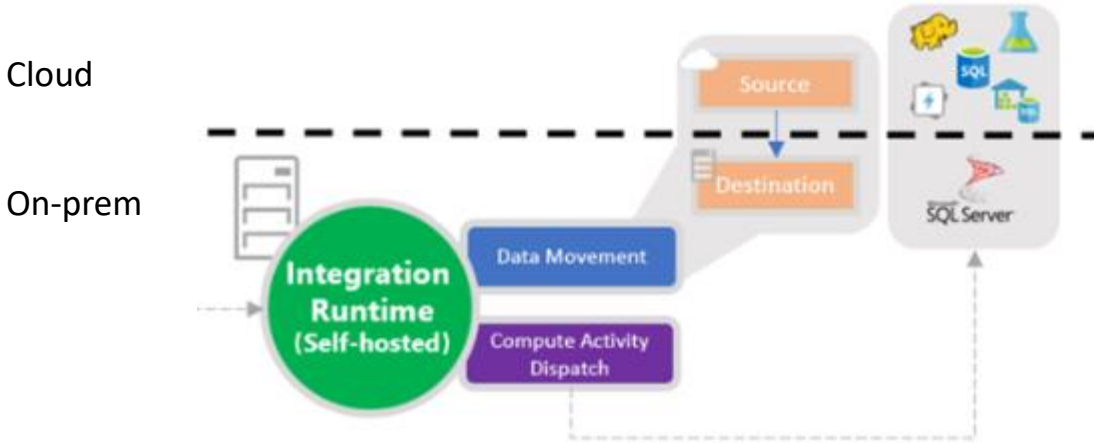
… not …

Development Mapping Data Flows



Mapping Data Flow Resources in Azure Data Factory





Good & bad about Big DataFlow pattern?

- Stability ★★
  - Mechanisms to prevent break caused by data type changes (+)
  - Still some work in progress…(-)
- Maturity ★★
  - Pretty new in the data world – minor trust
  - Vendor lock feared
- Supported data sources & sink ★★
  - Not as many data inputs/outputs components available as SSIS & ADF Copy

Good & bad about Big DataFlow pattern?

- Lifecyle management support ★★
  - Single data-flow pipeline for copy / transform leads (+)
  - Fewer additional storage layers needed (+)
  - No visual studio support for ADF, no builds (-)
  - Source code integration in Azure portal (+)
- Scalability ★★★
  - Easy scale out in same way integrated in ADF for copy/transform (+)
  - But could never scale as good as the optimized code of an expert (-)

Good & bad about Big DataFlow pattern?

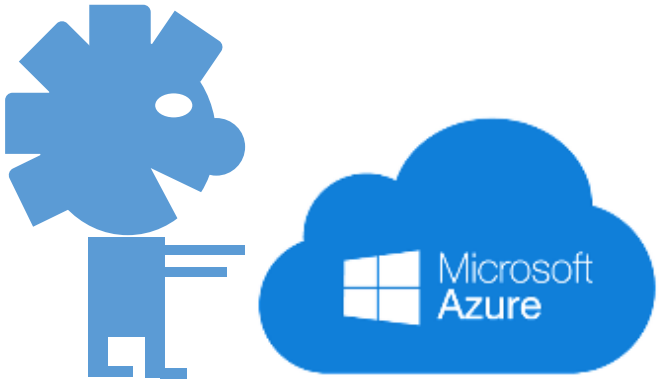
- Learning curve ★★★
  - WYSIWYG- GUI has high learn curve (+)
  - and is self-documenting (+)
  - No software engineering skills needed (+)

Typical use cases for Big Data workflow

- High volume of data
- Partitioning possible by attributes
- Complex data
- Many different use cases for the data
- „Agile“ source systems with often changing schemas
- Many different source systems / operators
- Less software engineering skills
- Less IT engineering skills for clusters

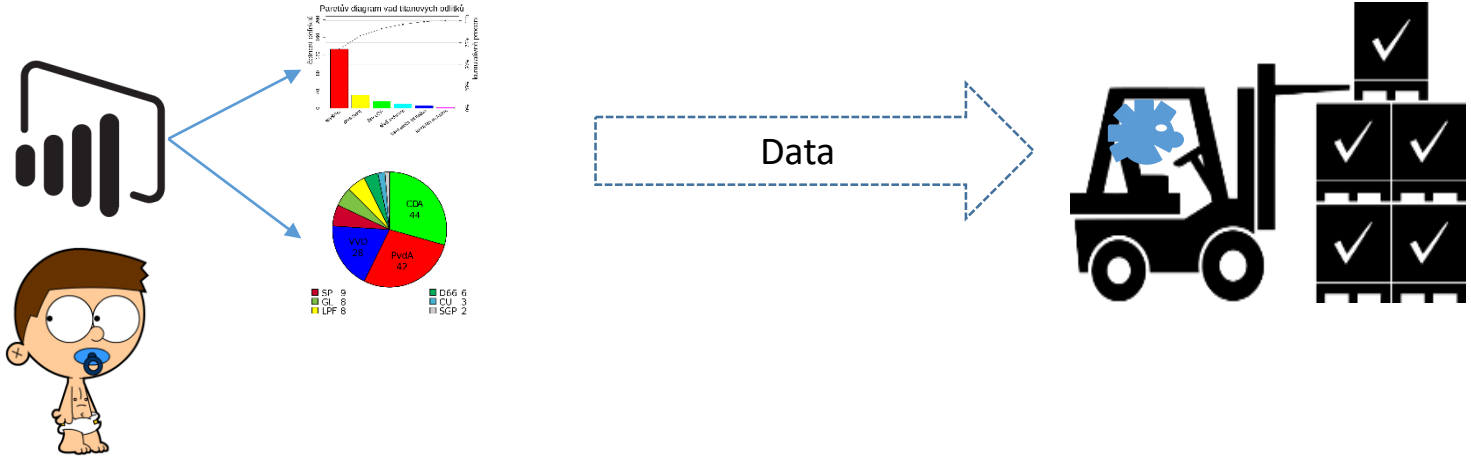
As big data cluster pattern

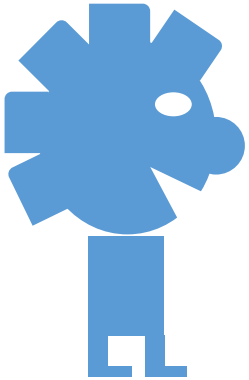
Demo – Mapping DataFlow in ADF



Self-Service Integration

Use Case 4: Reuse existing Power Query Code





Daen has to:

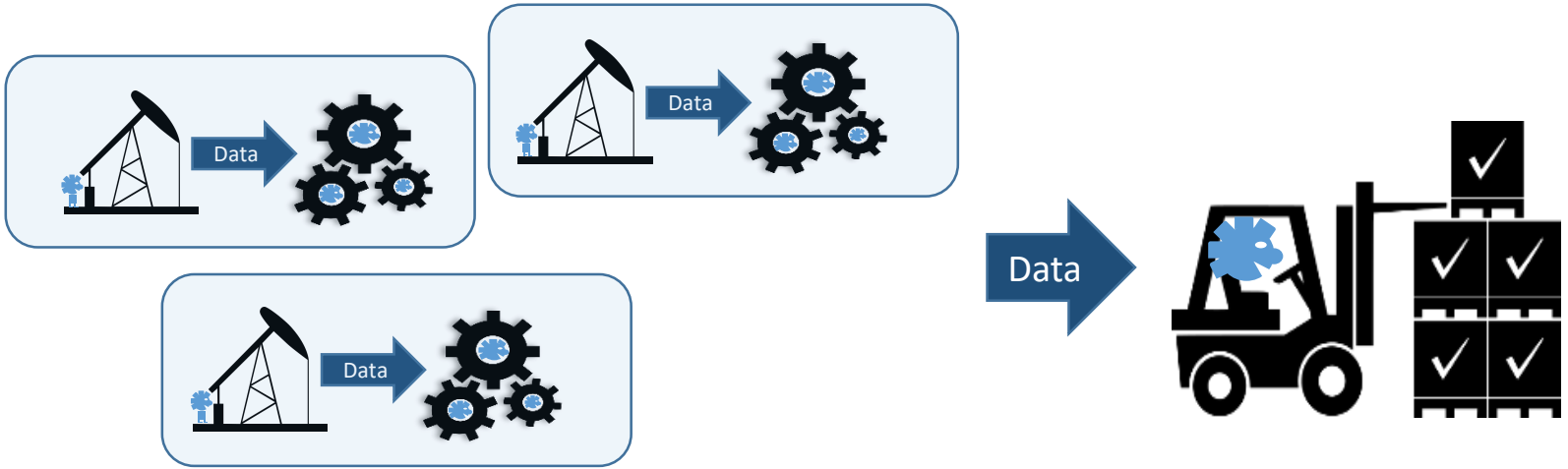
Reuse students Power Query work

Without rewriting code (complex)

Run this regularly & managed

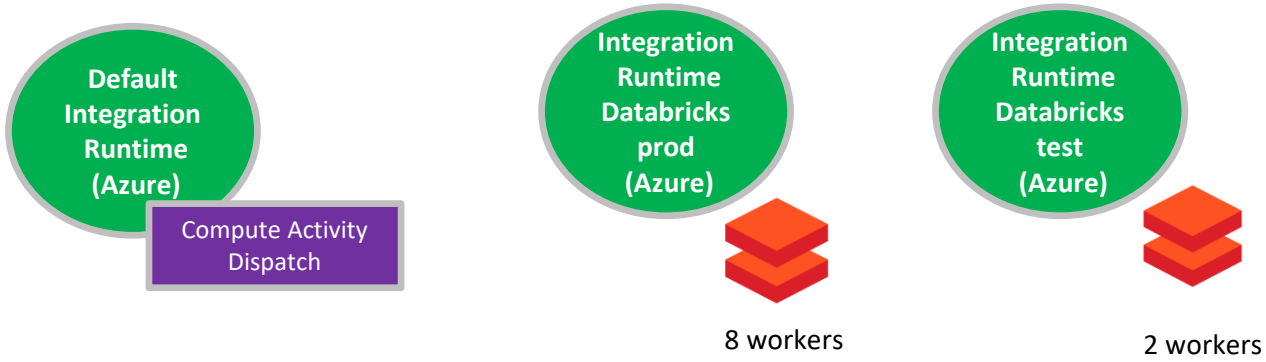
Integrate data on data platform

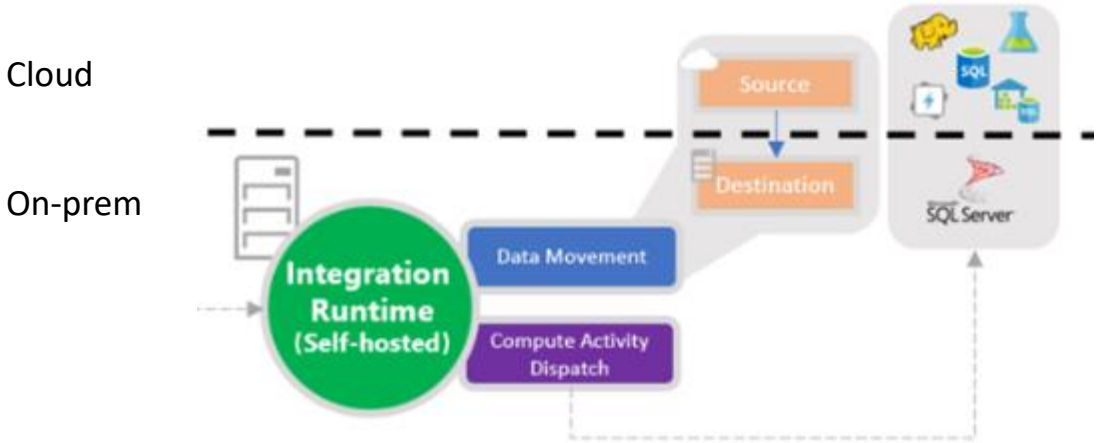
Self-Service Integration – Wrangling Dataflows



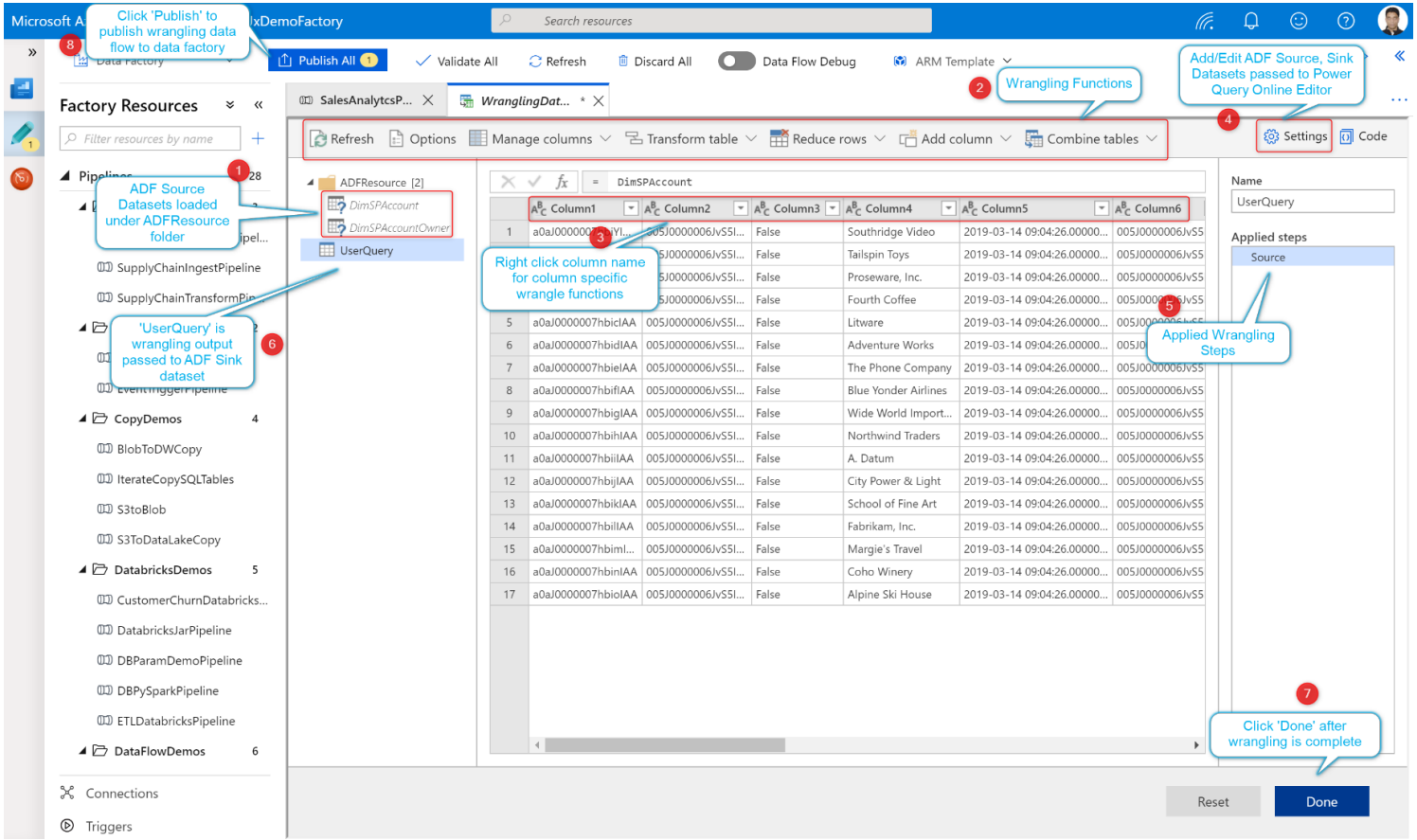
Logic in Azure Databricks external processing engine

Wrangling Data Flow Resources in Azure Data Factory





Development Wrangling Data Flows



https://github.com/gauravmalhot/wranglingdataflow/blob/master/GettingStarted/Wrangling%20Data%20Flow.md

Good & bad about self-service integration pattern?

- Stability ☆
  - Less mechanisms to prevent break caused by data type changes (-)
  - Still very much work in progress…(-)
- Maturity ☆
  - Still in preview (-)
  - Newer in the data world as Mapping Data Flow – minor trust (-)
  - Vendor lock feared (-)
- Supported data sources & sink ☆
  - Minor data sources (-)

Good & bad about self-service integration pattern?

- Lifecyle management support ★
  - *Same as Big DataFlow*
  - Self-service very focused approach e.g. build a report – not always suiteable for a data platform with multiple purposes for data (-)
  - Less expertise of users could leed to bad queries and minor data quality (-)
- Scalability ★★
  - *Same as Big DataFlow*
  - Double-interpreted -> M to Mapping Flow, Mapping Flow to Scala (-)

Good & bad about self-service integration pattern?

- Learning curve ★★★★
  - *Same as Big DataFlow*
  - Re-use skills from Power BI solutions (+)
  - Re-Use existing knowlegde of business users & analysts (+)

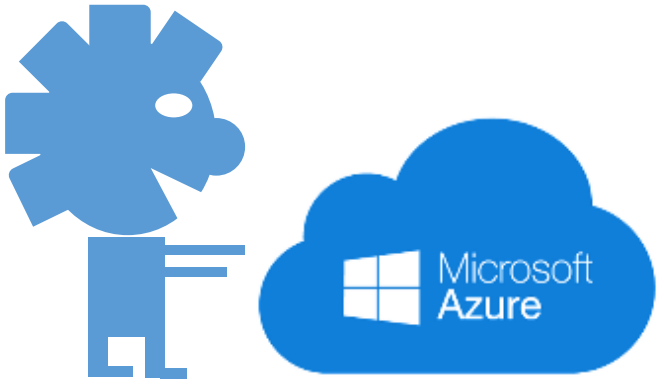
Typical self-service integration use cases

- Someone build a very complex Power Query
  - fastest way to integrate this to a managed production data platform
- Integration of data exploration

Results in overview

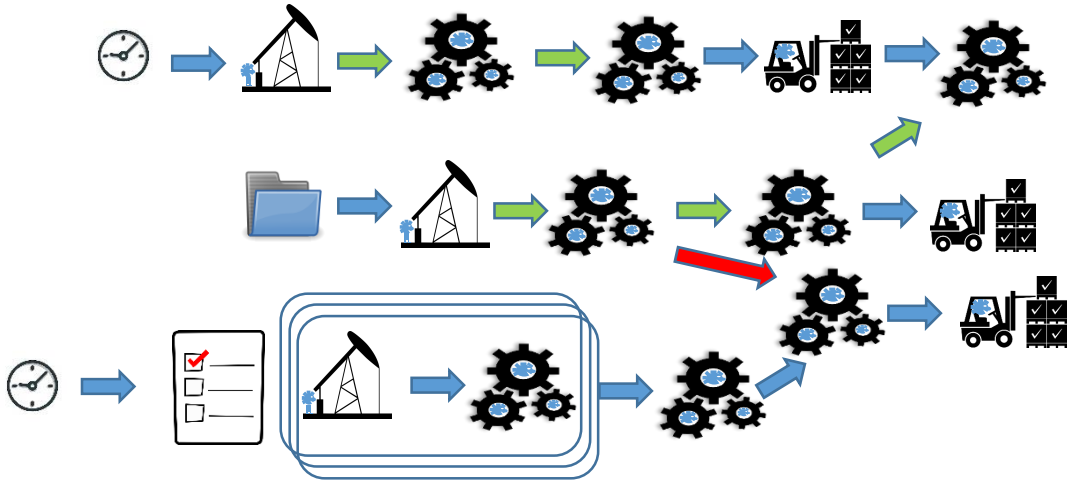
Criterion	Workflow engine	Big Data cluster	Big DataFlow	Self-Service
Stability				
Maturity				
Supported data sources & sinks				
Lifecycle management support				
Scalability				
Learning curve				

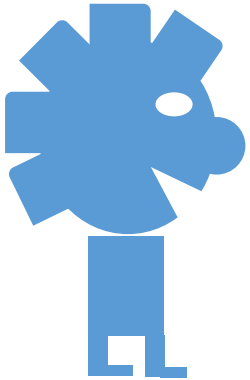
Demo – Wrangling DataFlow in ADF



Control Flow

Use Case 5: Run & Keep control





Daen has to:

Model dependencies

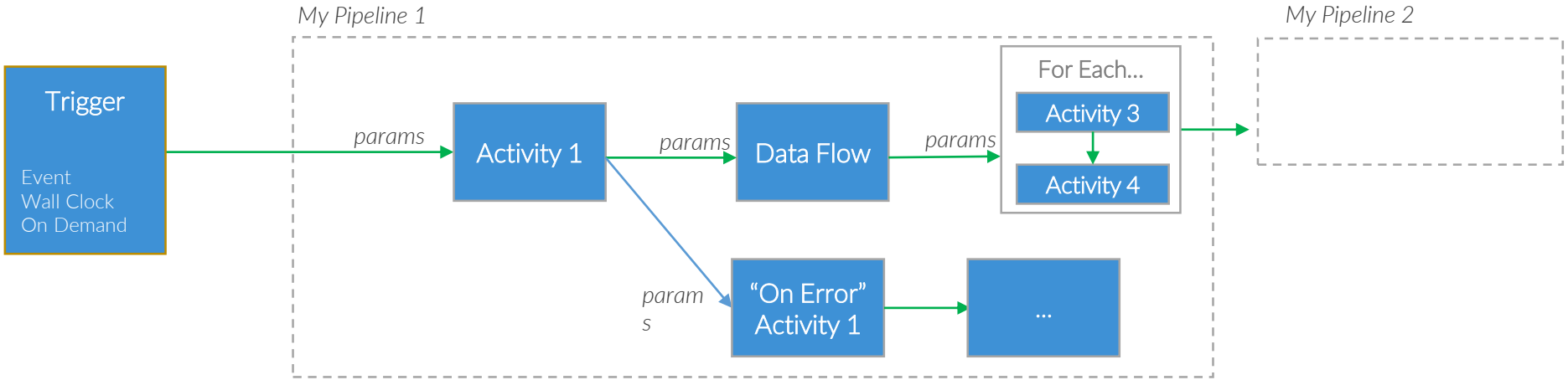
Implement error handling

Trigger loads by schedule & events

Run pipelines in a loop

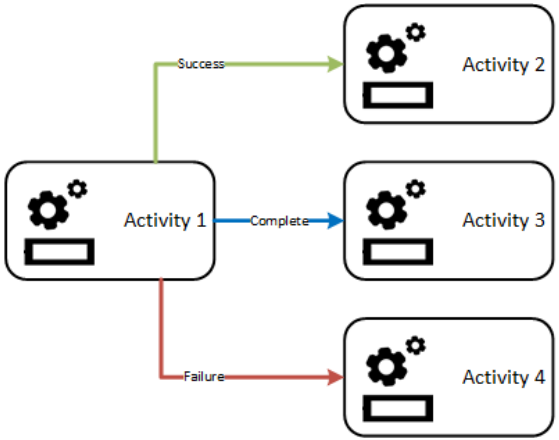
Monitor & Debug data pipelines

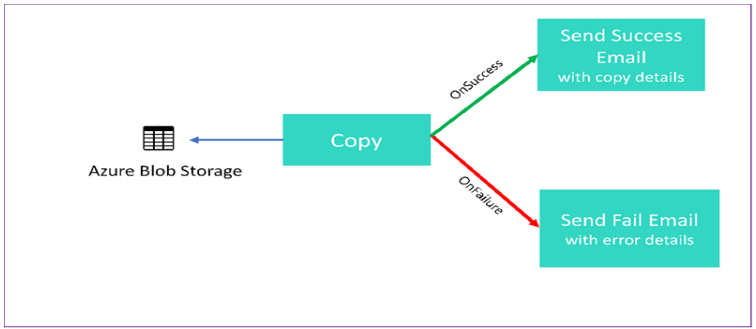
Control Flow – data pipelines, activities, triggers



Activities

Concepts





Branching

Dependencies of activities in a pipeline

Possible constraints:

- On success
- On failure
- On completion

Also custom 'if' conditions will be available for branching based expressions

Triggers

How do pipelines get started?

- on-demand
- Wall-clock Schedule
- Tumbling Window (aka time-slices in v1)
- Event on Blob Store

Example Control Flow

Get all data of a system by metadata

Pipeline check metadata

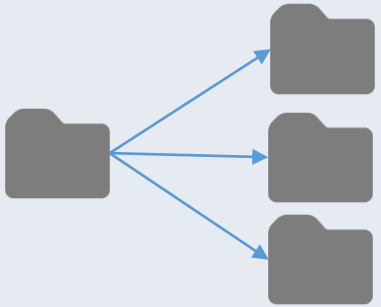
Activity: For each table in source

exec

exec

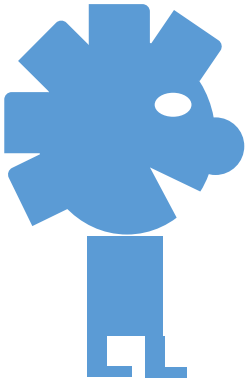
exec

Activity: Copy Data to Data Lake / Blob Store



Best practices + Q&A

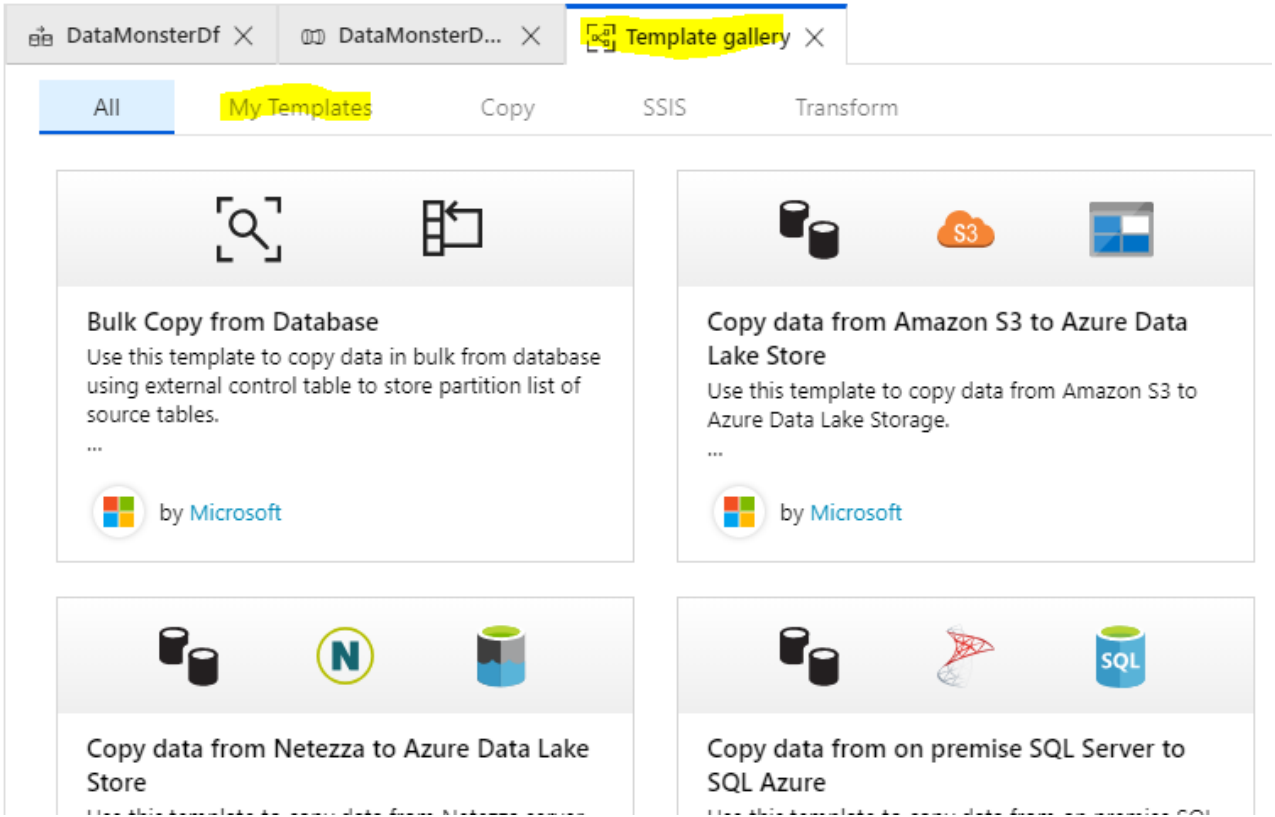
Best practices: Work, don‘t play!



Daen has to:

- Re-use logic in ADF
- Use secure mechanisms for credentials
- Deploy code to different stages w/o manual interference
- Develop with his team

Use Templates



Use Azure Key Vault for Configs

PASS DEUTSCHLAND e.V.

Edit linked service (Azure SQL Database)

Name *

AzSQLDB

Description

Connect via integration runtime *

AutoResolveIntegrationRuntime

Connection string | Azure Key Vault

AKV linked service *

InovexKeyVault

Edit connection

Secret name *

dbconnection

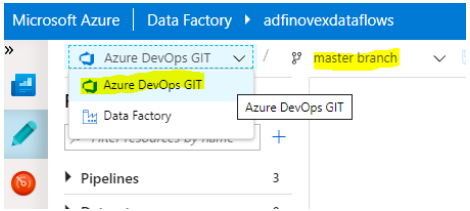
Secret version

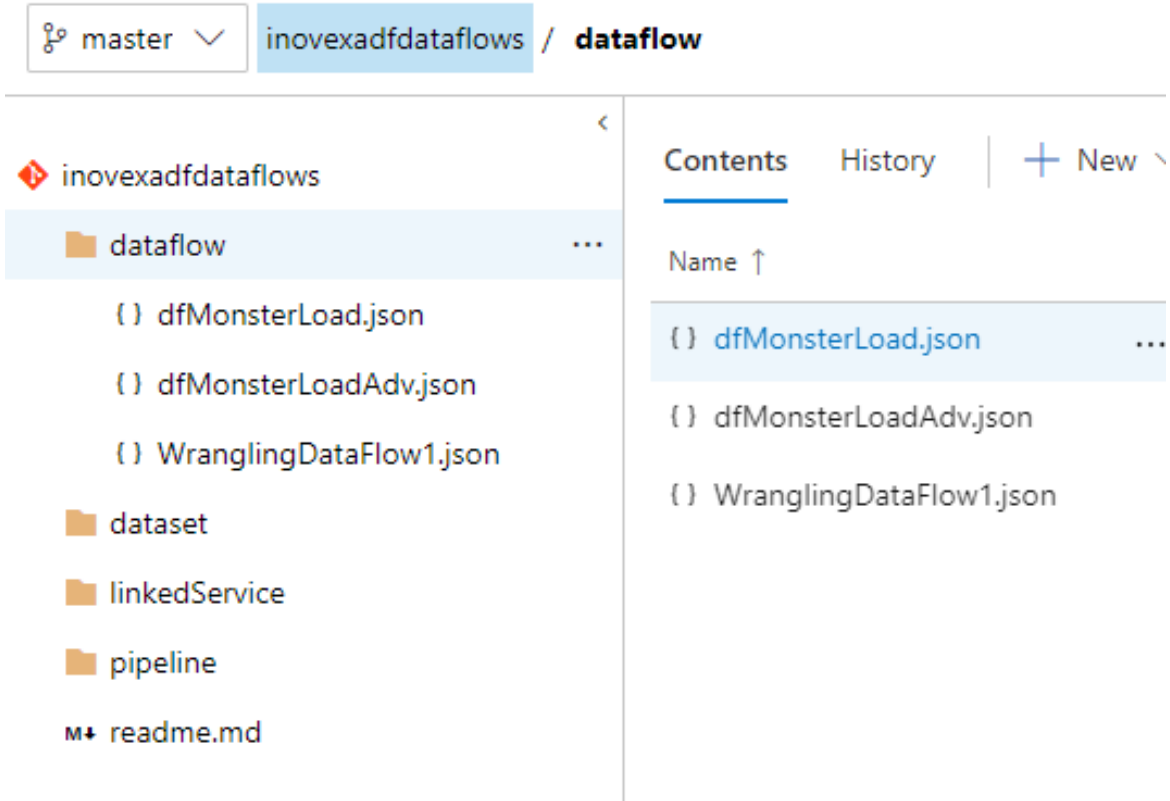
Use the latest version if left blank

Authentication type *

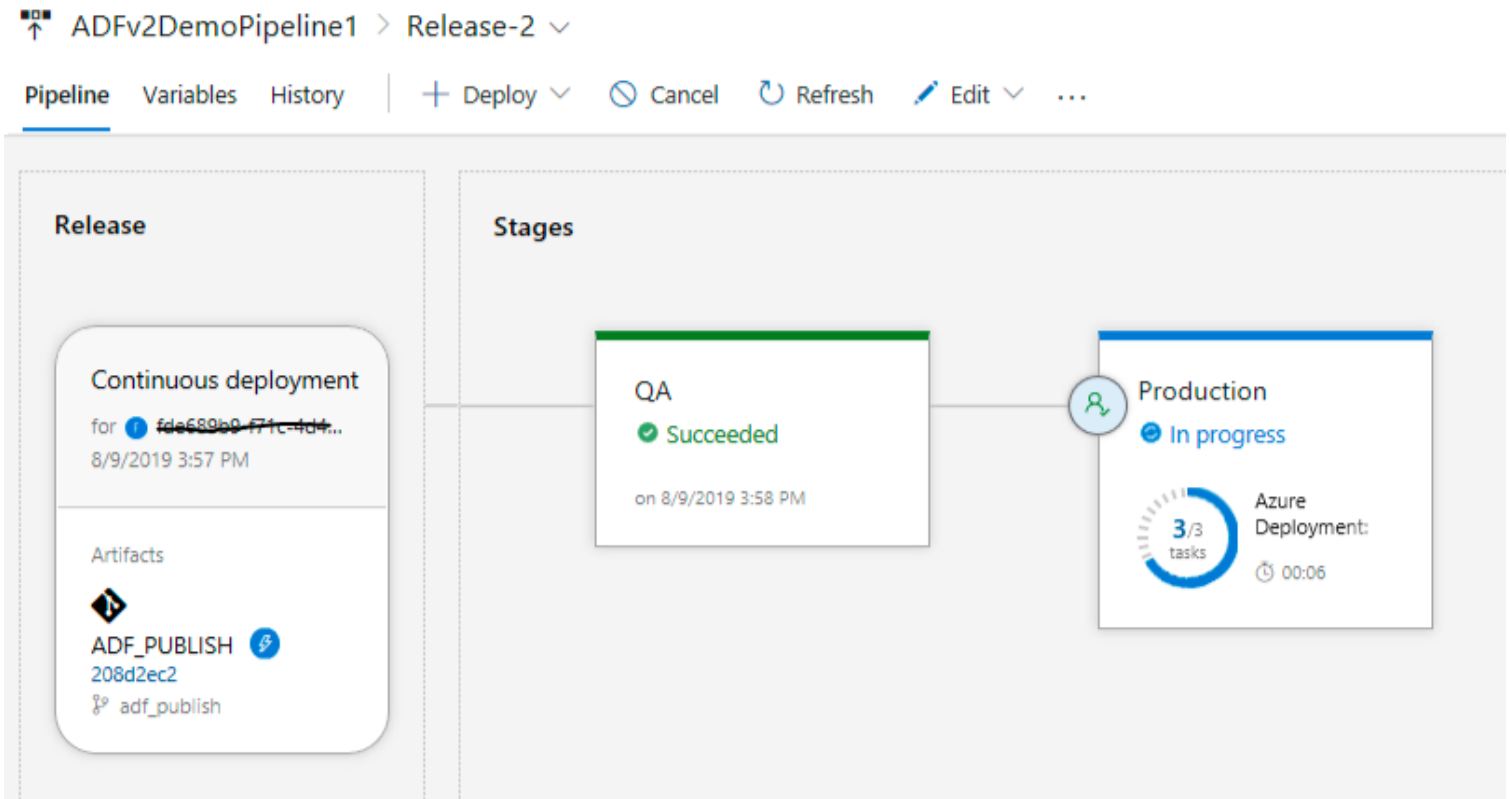
Sql Authentication or Managed Identity

Use a repository for your solution





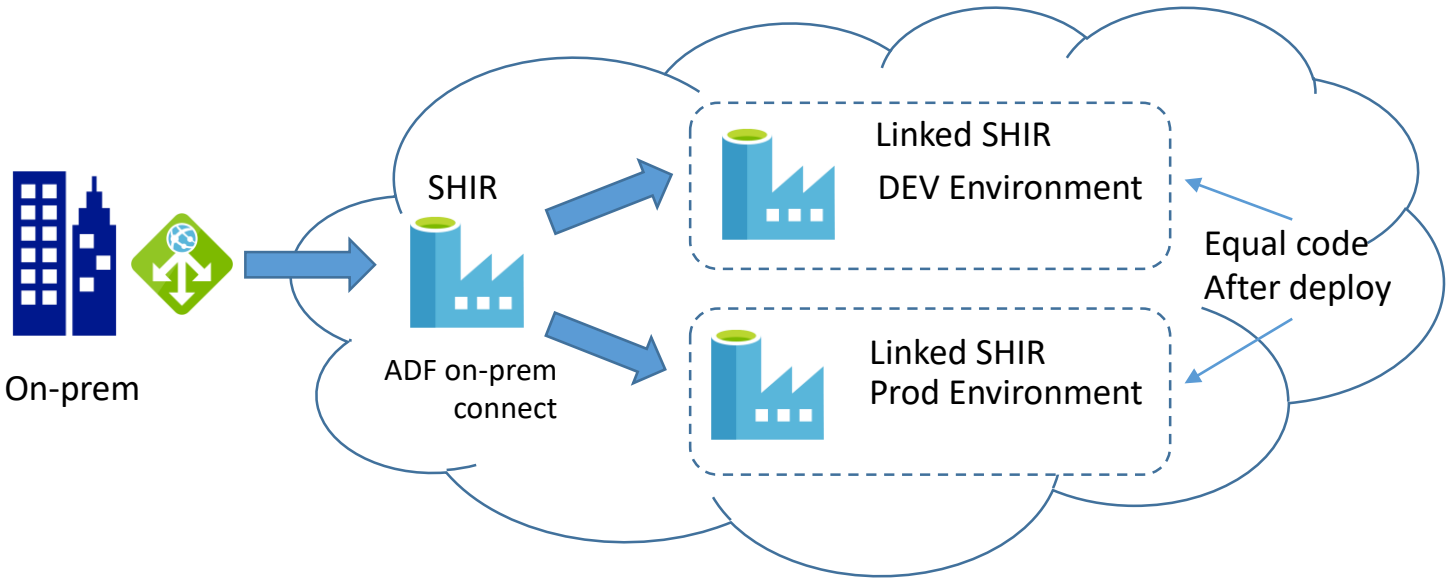
Use Build & Release pipelines in Azure Devops



Azure Devops Pipeline Setup for Azure Data Factory: https://medium.com/@cprosenjit/azure-devops-pipeline-setup-for-azure-data-factory-v2-8e957cd5141

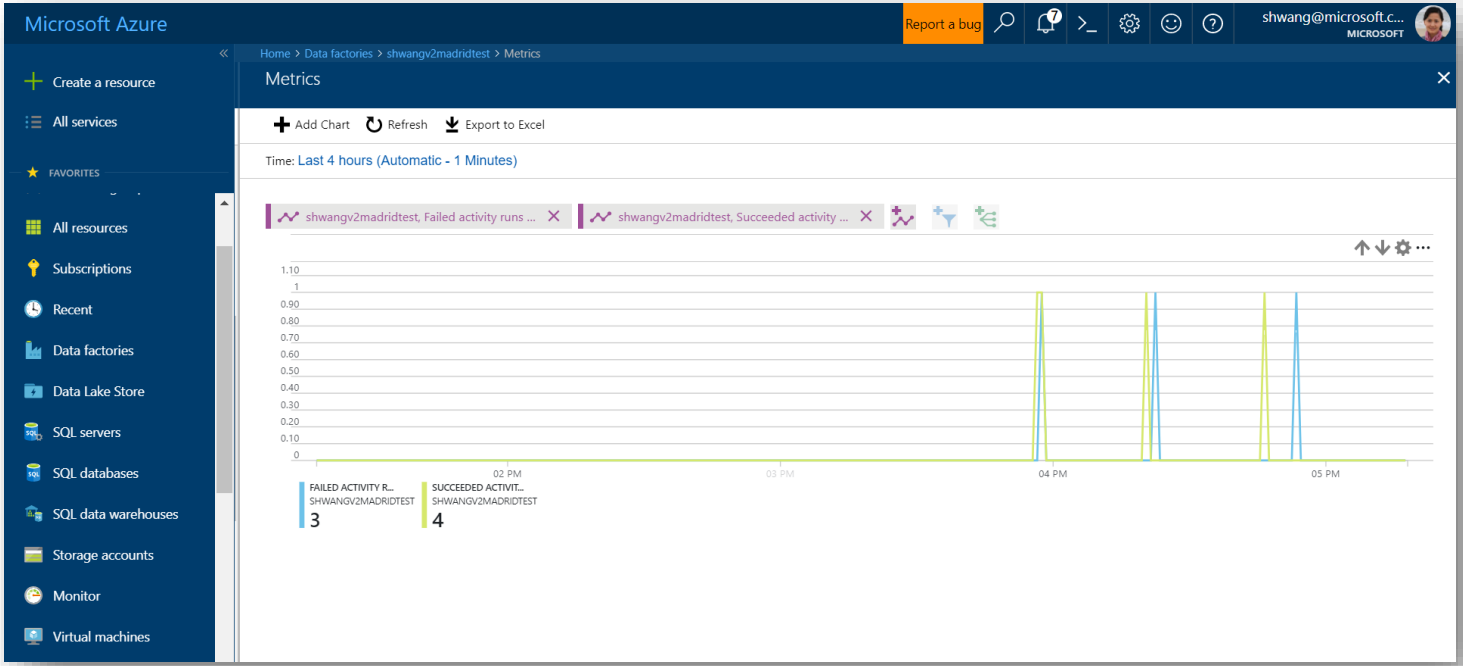
Using Linked Self-Hosted Integration Runtimes

Keep structure of data factories similar for different environments



Azure Monitoring integration

- Aggregated Metrics
- Alerts and actions
- Detailed diagnostic logs



Links and further informations

1. Microsoft documentation:
https://docs.microsoft.com/en-us/azure/data-factory/
2. Azure Data Factory – data flows preview documentation
https://github.com/kromerm/adfdataflowdocs
3. Cool screencasts about data flows
https://github.com/kromerm/adfdataflowdocs/tree/master/videos
4. Another good blogpost about ADF Data Flows
https://visualbi.com/blogs/microsoft/azure/azure-data-factory-data-flow-activity/
5. Comparison ADF Data Flows vs. SSIS vs. T-SQL
https://sqlplayer.net/2018/12/azure-data-factory-v2-and-its-available-components-in-data-flows/

Thank you for listening!

Daen is happy now and can better decide which pattern to use