Microsoft Azure Solutions Business

Data processing dans Azure

Venez découvrir les différentes options qui s'offrent à vous pour la manipulation des données dans le cloud Azure.
Nous présenterons les différents services vous permettant de déplacer, transformer, appliquer des calculs sur vos données, en prenant en compte les notions bien connues de variétés, vélocités, volumes.

Agenda (2h00)

- Revue des options de processing des données dans le cloud Azure
- Azure Synapse (Pipeline, Data Flow)
- Azure Batch
- processing Spark et serverless
- Compute Azure ML
- Orchestration

Microsoft

GPS Data/AI Strategy FY23

Delivered by CSA Team

Franck Gaillard
Cloud Solution Architect
Data AI
frgail@microsoft.com

Narjes Majdoub
Cloud Solution Architect
Data AI
nmajdoub@microsoft.com

Ali Bouhaddou
Cloud Solution Architect
Data Analytics
albouhad@microsoft.com

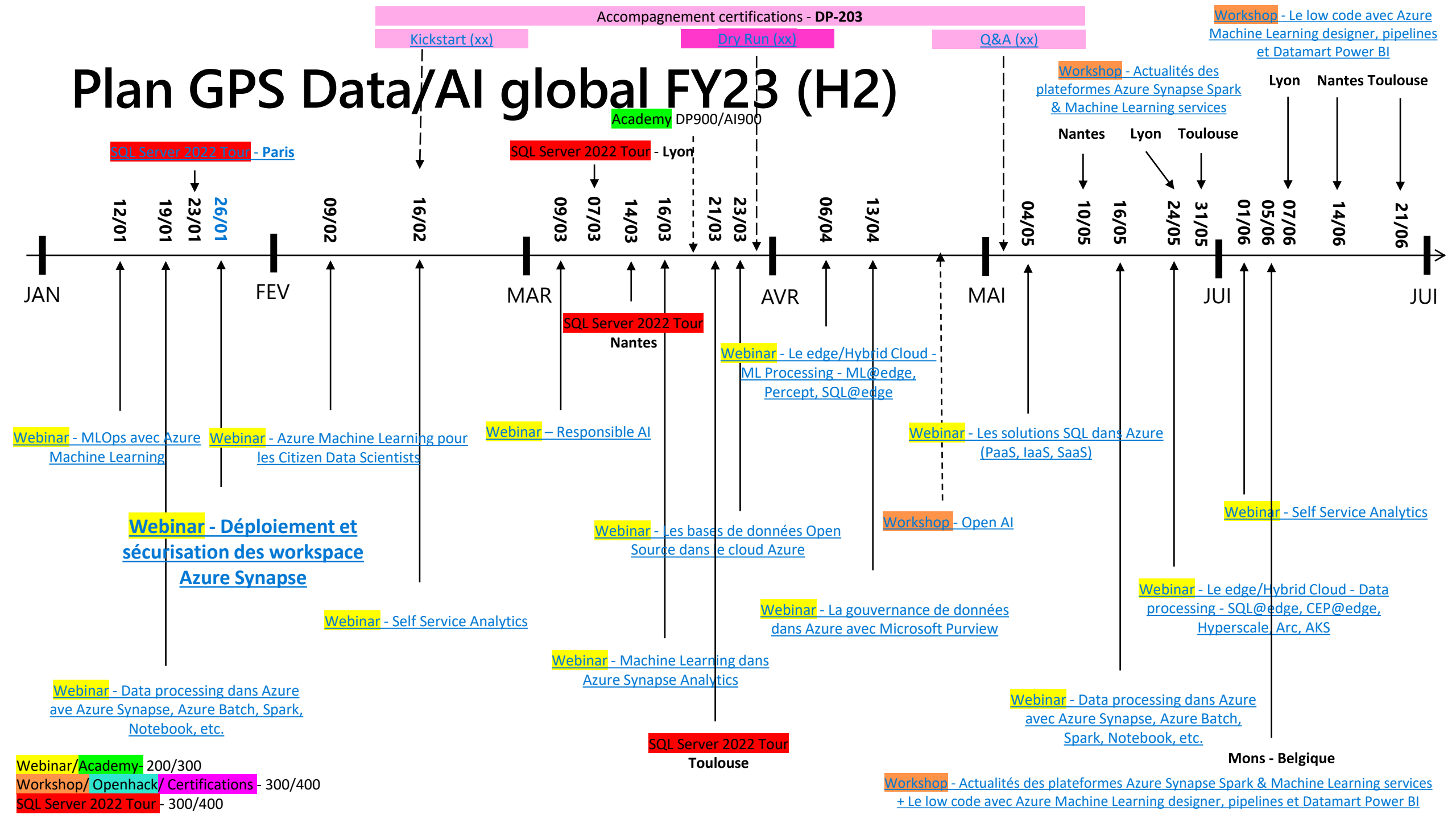
Frederic Gisbert
Cloud Solution Architect
Data Analytics
frgisber@microsoft.com

Azure Data & AI technical intensity plan

- From June 2022 to June 2023
- Focus on "**Azure Data & AI**" tech intensity

- Many content, from **L100** Beginner to **L400** Expert level:
  - **Academy** L100
  - **Webinar** L200/L300
  - **Workshop** L300/L400
  - **Certification kickstart** L300/L400
  - **Openhack / Microhack** L400

Plan GPS Data/AI global FY23 (H2)



Liste des évènements de type Webinar 2H

Event Webinar (Les jeudis de la Data & AI) - L200/300	Date	Duration (min)	Link
Azure Machine Learning pour les Data Scientists	15/09/2022	120	https://msevents.microsoft.com/event?id=2454281594
Azure Synapse	22/09/2022	120	https://msevents.microsoft.com/event?id=857781749
Les solutions SQL dans Azure (PaaS, IaaS, SaaS)	29/09/2022	120	https://msevents.microsoft.com/event?id=502366997
Déploiement et sécurisation des workspaces Azure Machine learning	06/10/2022	120	https://msevents.microsoft.com/event?id=1505714138
Azure Scale Analytics - Architectures Data Mesh dans Azure avec Azure Synapse, Microsoft Purview et Azure Data Share	13/10/2022	120	https://msevents.microsoft.com/event?id=139685175
MLOps avec Azure Machine Learning	20/10/2022	120	https://msevents.microsoft.com/event?id=1245885767
SQL Server 2022 et hybridation native avec Azure SQL Managed Instance	10/11/2022	120	https://msevents.microsoft.com/event?id=145826476
Machine Learning dans Azure Synapse Analytics	17/11/2022	120	https://msevents.microsoft.com/event?id=3637723312
Azure Cosmos DB et IA	24/11/2022	120	https://msevents.microsoft.com/event?id=2646013445
Azure et les Services Cognitifs	08/12/2022	120	https://msevents.microsoft.com/event?id=3772037220
La gouvernance de données dans Azure avec Microsoft Purview	15/12/2022	120	https://msevents.microsoft.com/event?id=1499560981
MLOps avec Azure Machine Learning	12/01/2023	120	https://msevents.microsoft.com/event?id=4115194515
Data processing dans Azure ave Azure Synapse, Azure Batch, Spark, Notebook, etc.	19/01/2023	120	https://msevents.microsoft.com/event?id=1537241181
Déploiement et sécurisation des workspace Azure Synapse	26/01/2023	120	https://msevents.microsoft.com/event?id=1806467748
Azure Machine Learning pour les Citizen Data Scientists	09/02/2023	120	https://msevents.microsoft.com/event?id=2367521229
PowerBI - Self Service Analytics	16/02/2023	120	https://msevents.microsoft.com/event?id=1401519679
L'IA responsable avec Azure machine learning	09/03/2023	120	https://msevents.microsoft.com/event?id=2072953112
Machine Learning dans Azure Synapse Analytics	16/03/2023	120	https://msevents.microsoft.com/event?id=3413014857
Les bases de données Open Source dans le cloud Azure	23/03/2023	120	https://msevents.microsoft.com/event?id=2727487131
Hybridation des services de Machine Learning Azure	06/04/2023	120	https://msevents.microsoft.com/event?id=1624914222
La gouvernance de données dans Azure avec Microsoft Purview	13/04/2023	120	https://msevents.microsoft.com/event?id=3909342839
Les solutions SQL dans Azure (PaaS, IaaS, SaaS)	04/05/2023	120	https://msevents.microsoft.com/event?id=1162207895
Data processing dans Azure ave Azure Synapse, Azure Batch, Spark, Notebook, etc.	16/05/2023	120	https://msevents.microsoft.com/event?id=3517068442
Hybridation des services de données Azure	24/05/2023	120	https://msevents.microsoft.com/event?id=2996507398
Self Service Analytics	01/06/2023	120	https://msevents.microsoft.com/event?id=2387937988
Total	**25**		

Liste des évènements de type Workshop/Prepa Cert/Academy

Event Workshop L300/400	Date	Duration (min)	Link
Synapse et Lakehouse Database	18/10/2022	120	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHbR3zwJTO3s11AuaqpNnBbrwdURE1RMVgwTDNISTE1TDFYSDVLR0cyS1kwWS4u
Le NoSQL dans Azure - Migration CosmosDB / OSS	08/11/2022	120	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHbR3zwJTO3s11AuaqpNnBbrwdURE1RMVgwTDNISTE1TDFYSDVLR0cyS1kwWS4u
Lab Lyon - Data & AI	22/11/2022	240	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHbR3zwJTO3s11AuaqpNnBbrwdUMlZZOURET0RSWjcyTERYRkJGTlFFUjJaUi4u
Lab Nantes - Data & AI	29/11/2022	240	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHbR3zwJTO3s11AuaqpNnBbrwdUMlZZOURET0RSWjcyTERYRkJGTlFFUjJaUi4u
L'IA dans les services de données Azure	12/12/2022	120	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHbR3zwJTO3s11AuaqpNnBbrwdURE1RMVgwTDNISTE1TDFYSDVLR0cyS1kwWS4u
Open AI	H2	120	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHbR3zwJTO3s11AuaqpNnBbrwdURE1RMVgwTDNISTE1TDFYSDVLR0cyS1kwWS4u

Event Academy, kickstart certifications, workshop certifications	Date	Duration (min)	Link
MCPP - DP-420	10/10/2022	420	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHbR3zwJTO3s11AuaqpNnBbrwdUMkJISlRKSU1RRFA0OVgzSFdTSTY0RE9WQy4u
Micro Hack CosmosDB	20/10/2022	420	H1 - Inscriptions PTA
Academy DP900	17-21/10/2022	300	https://msevents.microsoft.com/event?id=3250818161
Academy AI900	17-21/10/2022	300	https://msevents.microsoft.com/event?id=2717528090
Kickstart DP-500	17/10/2022	60	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHbR3zwJTO3s11AuaqpNnBbrwdUNEk3WFQ1TEdNNTQ2Uk85V0cxQzM3TE9ZRS4u
Dry Run DP-500	14/11/2022	120	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHbR3zwJTO3s11AuaqpNnBbrwdUNEk3WFQ1TEdNNTQ2Uk85V0cxQzM3TE9ZRS4u
Q&A DP-500	05/12/2022	90	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHbR3zwJTO3s11AuaqpNnBbrwdUNEk3WFQ1TEdNNTQ2Uk85V0cxQzM3TE9ZRS4u
Kickstart DP-100	17/10/2022	60	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHbR3zwJTO3s11AuaqpNnBbrwdUNDAxV0hSN0FHM1YzUzI3OUNMUFYxSkRIMi4u
Dry Run DP-100	14/11/2022	120	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHbR3zwJTO3s11AuaqpNnBbrwdUNDAxV0hSN0FHM1YzUzI3OUNMUFYxSkRIMi4u
Q&A DP-100	05/12/2022	90	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHbR3zwJTO3s11AuaqpNnBbrwdUNDAxV0hSN0FHM1YzUzI3OUNMUFYxSkRIMi4u
Kickstart DP-203	17/10/2022	60	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHbR3zwJTO3s11AuaqpNnBbrwdUOVFWOUVCNFcyQk5SVjFBUFczNktCUFpLMi4u
Dry Run DP-203	14/11/2022	120	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHbR3zwJTO3s11AuaqpNnBbrwdUOVFWOUVCNFcyQk5SVjFBUFczNktCUFpLMi4u
Q&A DP-203	05/12/2022	90	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHbR3zwJTO3s11AuaqpNnBbrwdUOVFWOUVCNFcyQk5SVjFBUFczNktCUFpLMi4u

Liste des évènements de type Workshop/Prepa Cert/Academy

SQL Server 2022 Tour L300/400	Date	Duration (min)	Link
Session Paris	23/01/2023	480	https://msevents.microsoft.com/event?id=1969809187
Session Lyon	07/03/2023	480	
Session Nantes	14/03/2023	480	
Session Toulouse	21/03/2023	480	

Tour de France - Data/AI services L300/400	Date	Duration (min)	Link
Nantes - Session Azure Synapse SQL-Spark-ML / Azure Machine Learning	10/05/2023	480	https://forms.office.com/r/P6fUq2VJmv
Nantes - Session low code - Azure Machine Learning, Pipelines, Datamarts Power BI	07/06/2023	480	https://forms.office.com/r/P6fUq2VJmv
Toulouse - Session Azure Synapse SQL-Spark-ML / Azure Machine Learning	31/05/2023	480	https://forms.office.com/r/P6fUq2VJmv
Toulouse - Session low code - Azure Machine Learning, Pipelines, Datamarts Power BI	21/06/2023	480	https://forms.office.com/r/P6fUq2VJmv
Lyon - Session Azure Synapse SQL-Spark-ML / Azure Machine Learning	24/05/2023	480	https://forms.office.com/r/P6fUq2VJmv
Lyon - Session low code - Azure Machine Learning, Pipelines, Datamarts Power BI	14/06/2023	480	https://forms.office.com/r/P6fUq2VJmv

Data as a strategic asset

Data-driven transformations yield significant benefits

54%
increase in revenue performance

44%
faster time to market

62%
improvement in customer satisfaction

54%
increased profit results

Source: How to lead a data-driven digital transformation by Harvard Business Reveiw

Today's data realities

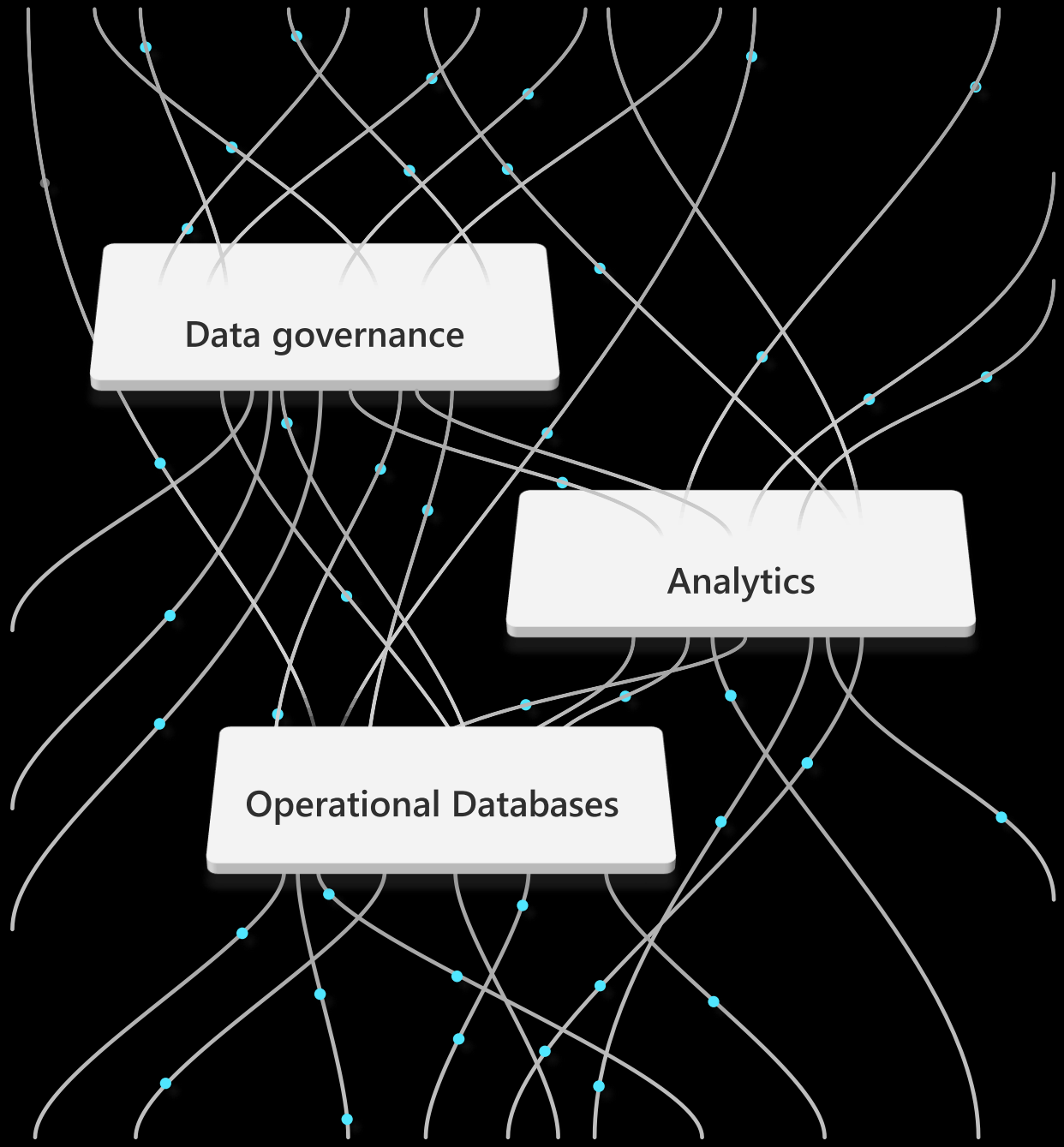
What data do I have?

Is it trustworthy?

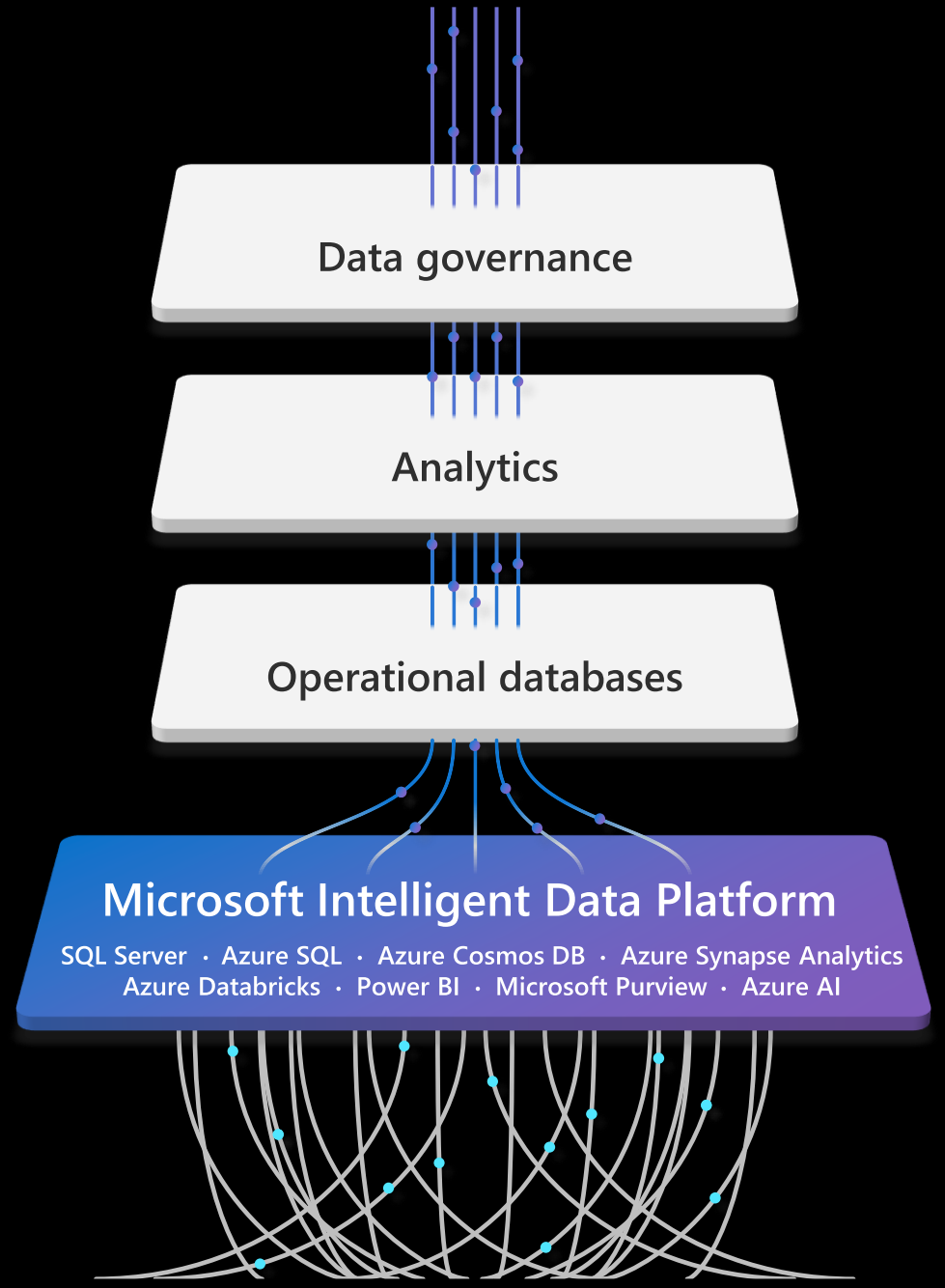
Can people access the data needed to make the right decisions?

How can I enable faster business insights?

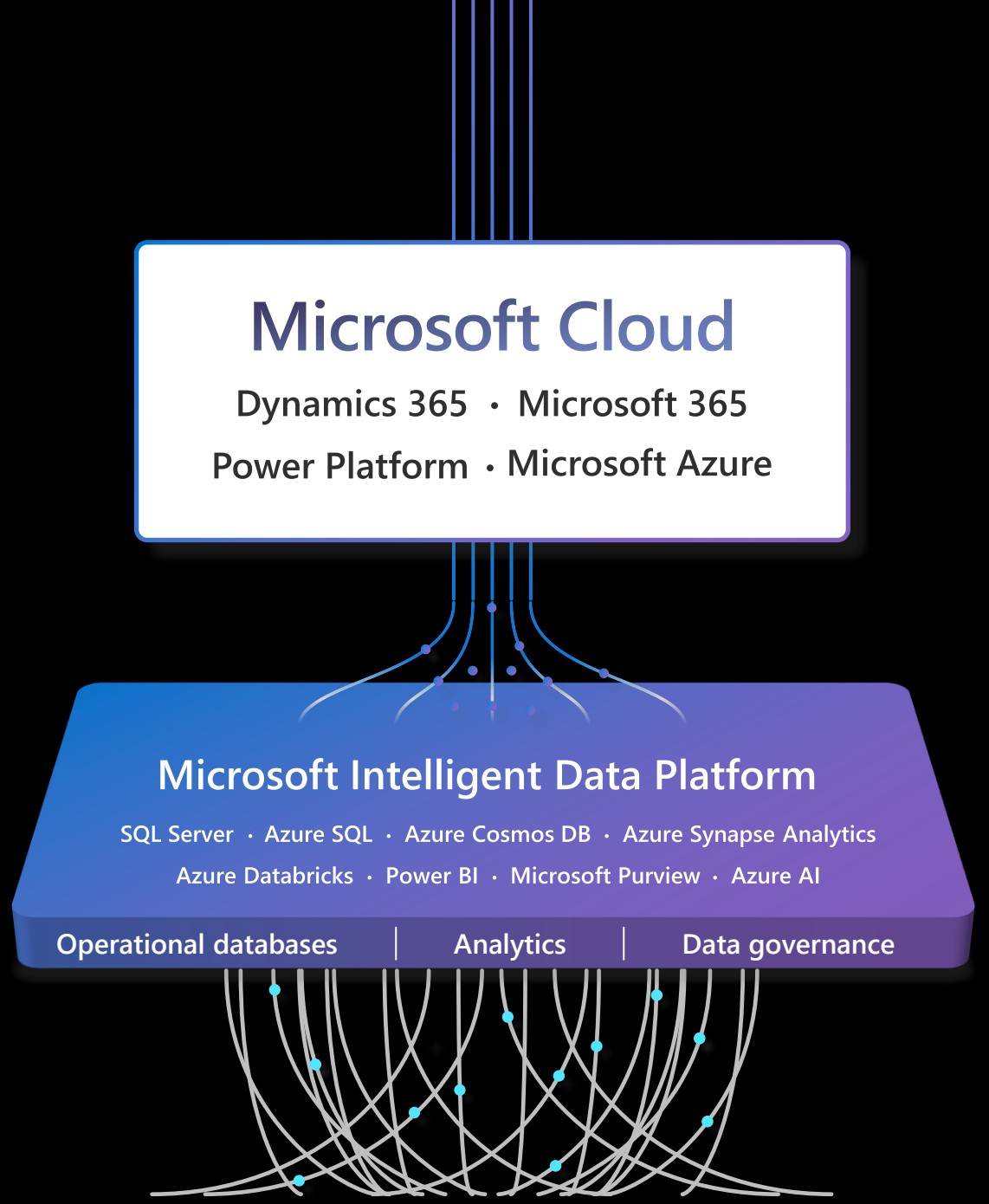
What's my compliance exposure?



Introducing Microsoft Intelligent Data Platform



Transform with the Microsoft Cloud



Processing de données dans Azure

- Data Integration
- Data Engineering
- Data Warehousing
- Data Science
- Observational Analytics
- Business Intelligence
- Governance

Data Engineering

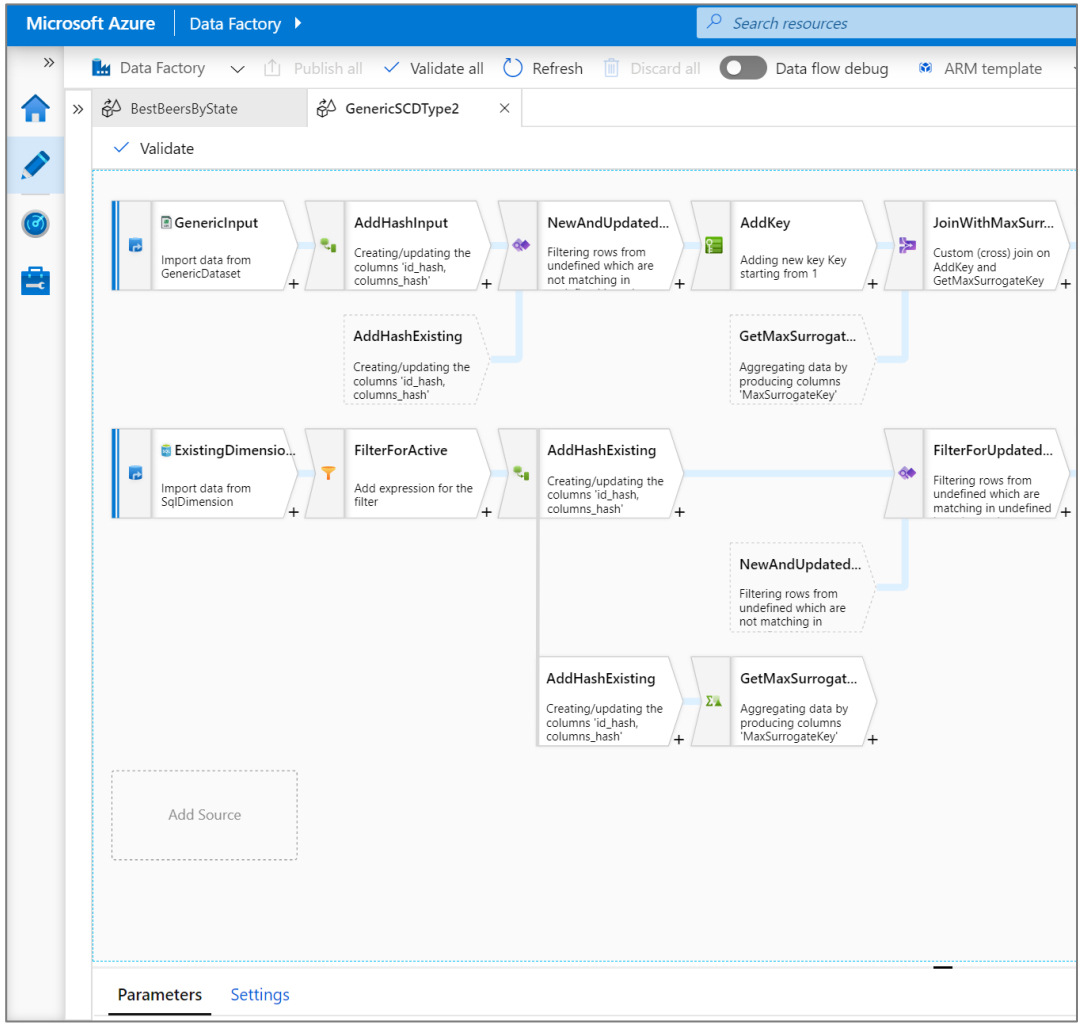
Microsoft

Mapping Data Flows

Code-free data transformation at scale with Azure Data Factory and Synapse Analytics

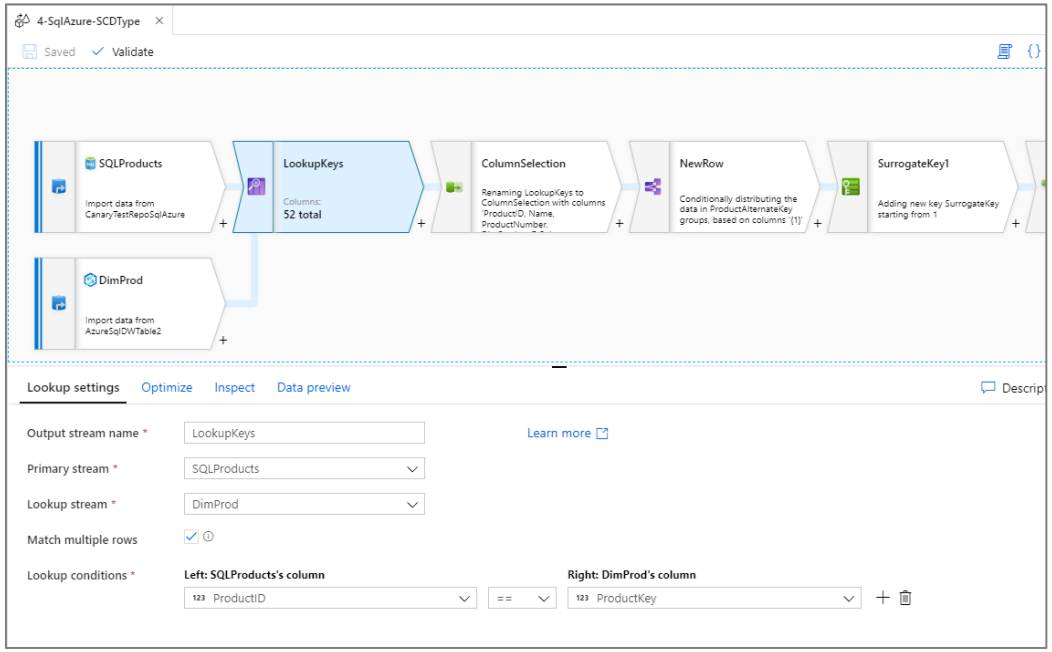
What are mapping data flows?

- Code-free data transformation at scale
- Serverless, scaled-out, ADF/Synapse-managed Apache Spark™ engine
- Resilient flows handle structured and unstructured data
- Operationalized as a data pipeline activity

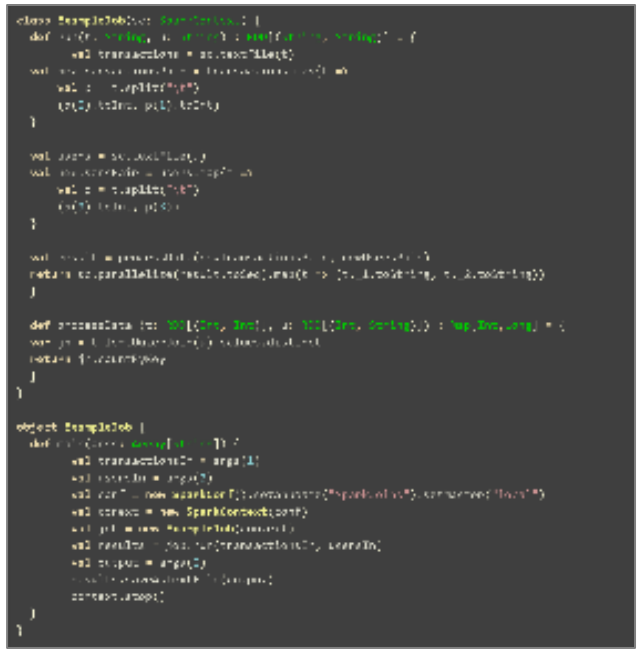


Code-free data transformation at scale

- Intuitive UX lets you focus on building transformation logic
  - Data cleansing
  - Data validation
  - Data aggregation
- No requirement of knowing Spark, cluster management, Scala, Python, etc



vs



ADF Data Flows Service Architecture

ADF Data Flow UI Designer

Design, debug, manage data transform logic in browser UI

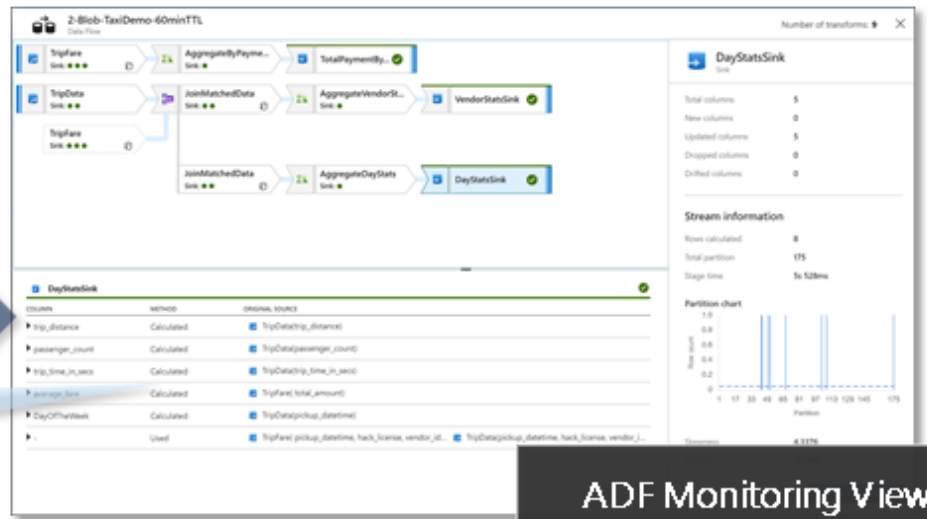
```
Data Flow Name  metadataChecker

1  parameters{
2    filename as string ('SampleData/moviesDB.csv'),
3    columns as string ('movie,title')
4  }
5  source(allowSchemaDrift: true,
6    validateSchema: false,
7    limit: 1,
8    wildcardPaths:[($filename)] ~> source1
9  DerivedColumn1 foldDown(unroll(cols),
10   mapColumn(
11     cols
12   ),
13   skipDuplicateMapInputs: false,
14   skipDuplicateMapOutputs: false) ~> Flatten1
15 Flatten1 filter(instr($columns,cols) > 0) ~> Filter1
16 source1 derive(cols = columnNames()) ~> DerivedColumn1
17 Filter1 sink(allowSchemaDrift: true,
18   validateSchema: false,
```

Data Flow Script

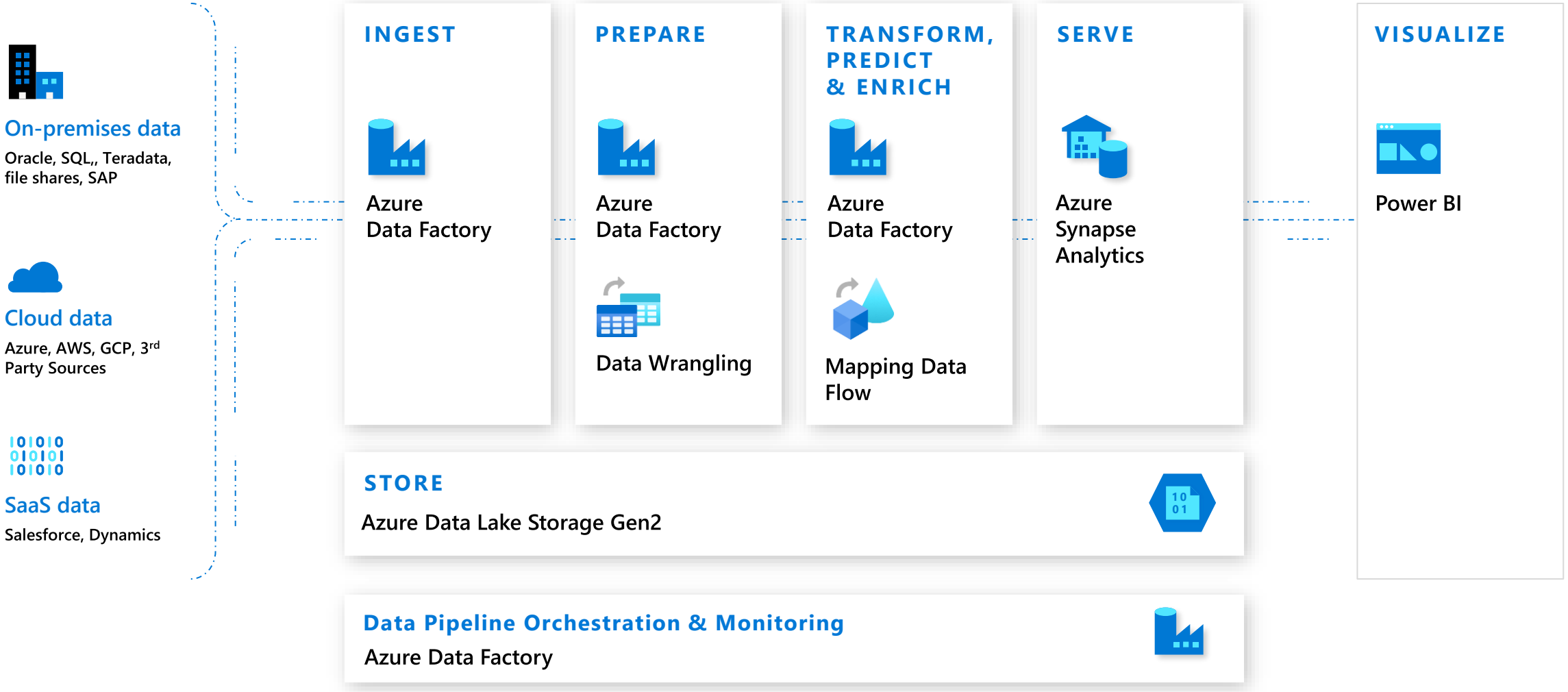
UI builds data transformation scripts that contain metadata defining your logic. Script payload is combined with the ADF JSON definition of your pipeline activities. ADF spins-up a JIT on-demand Spark cluster and builds an execution plan for your data flow in Spark.

Using your Azure IR configs, ADF will spin-up Spark clusters and an executor job will execute your data transformation logic. The results of the data ingest, transformation, data partitioning, data egress, and timings will all appear in the monitoring view.



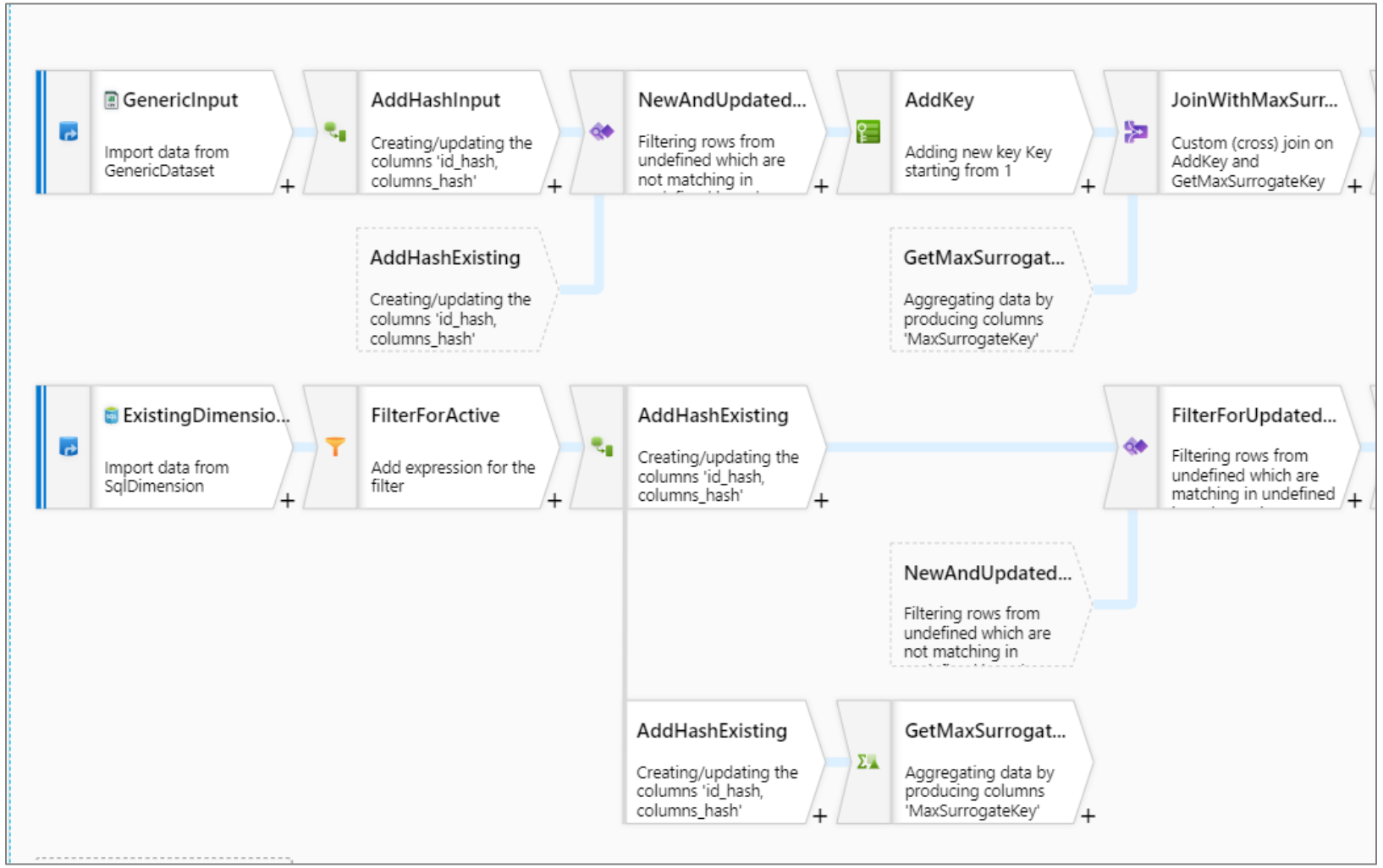
ADF Monitoring View

Modern Data Warehouse (MDW)

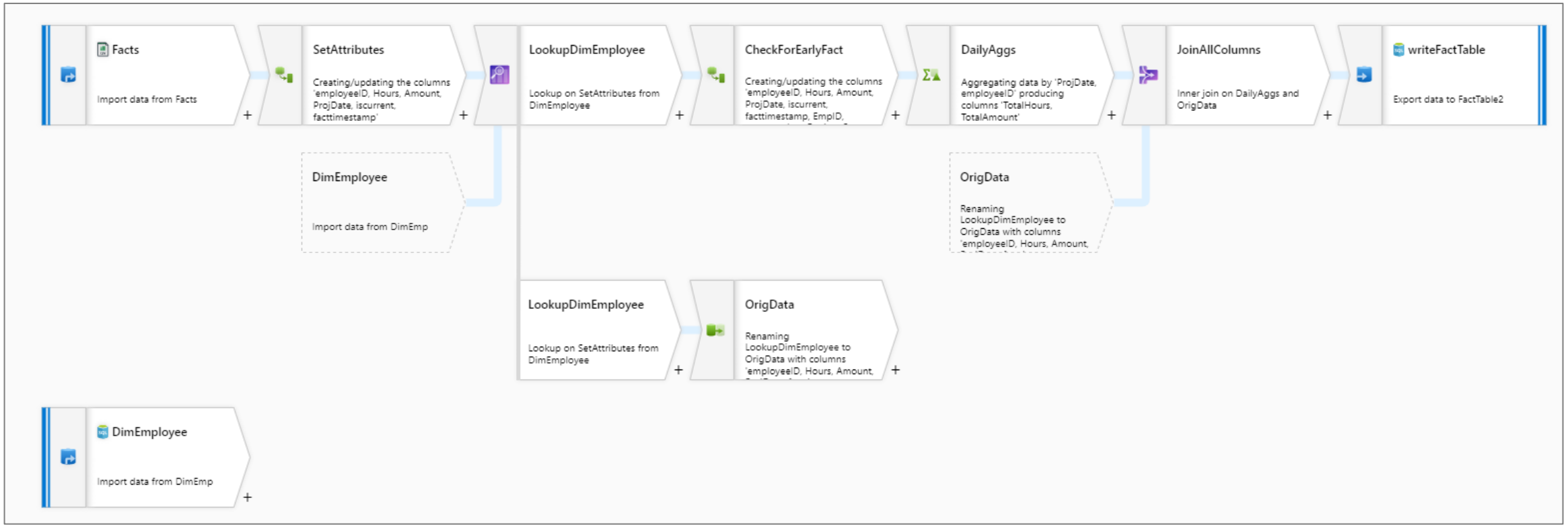


Common data flow scenarios

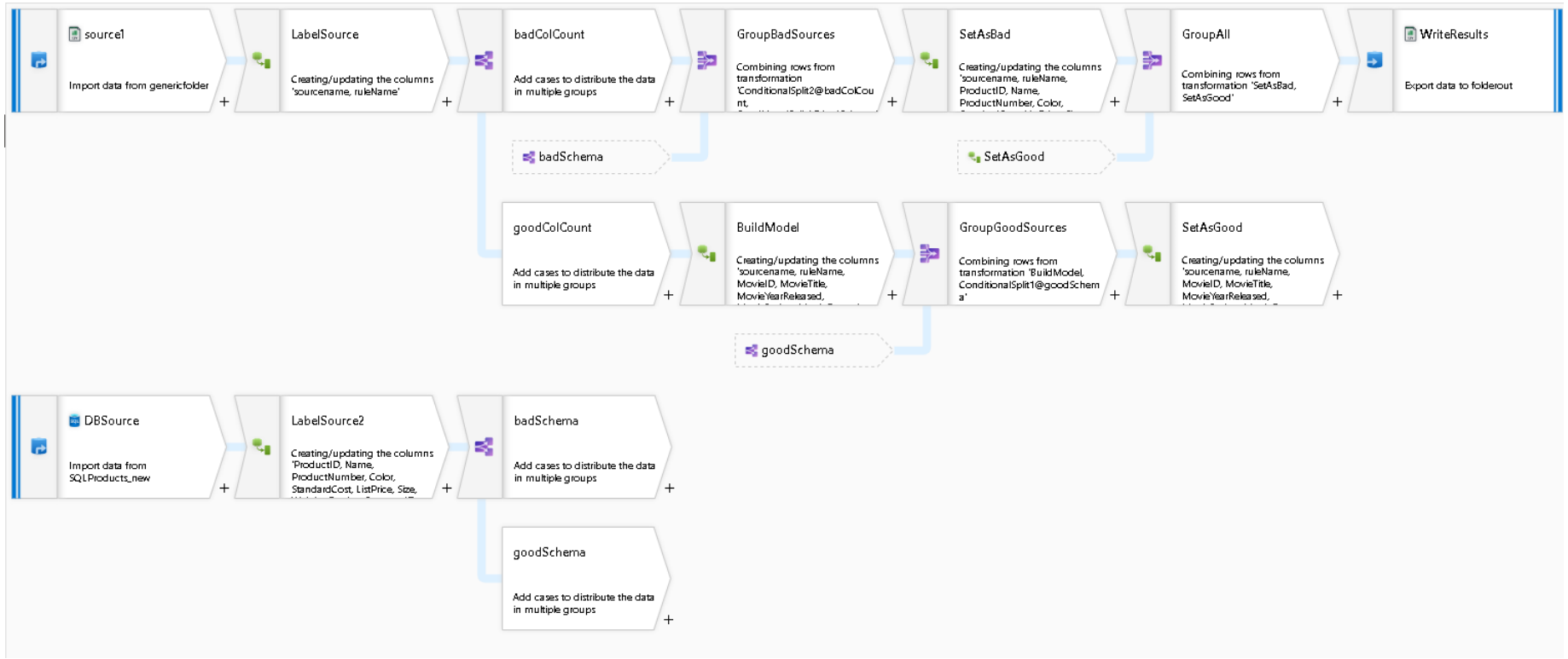
Slowly changing dimensions



Fact loading into a data warehouse

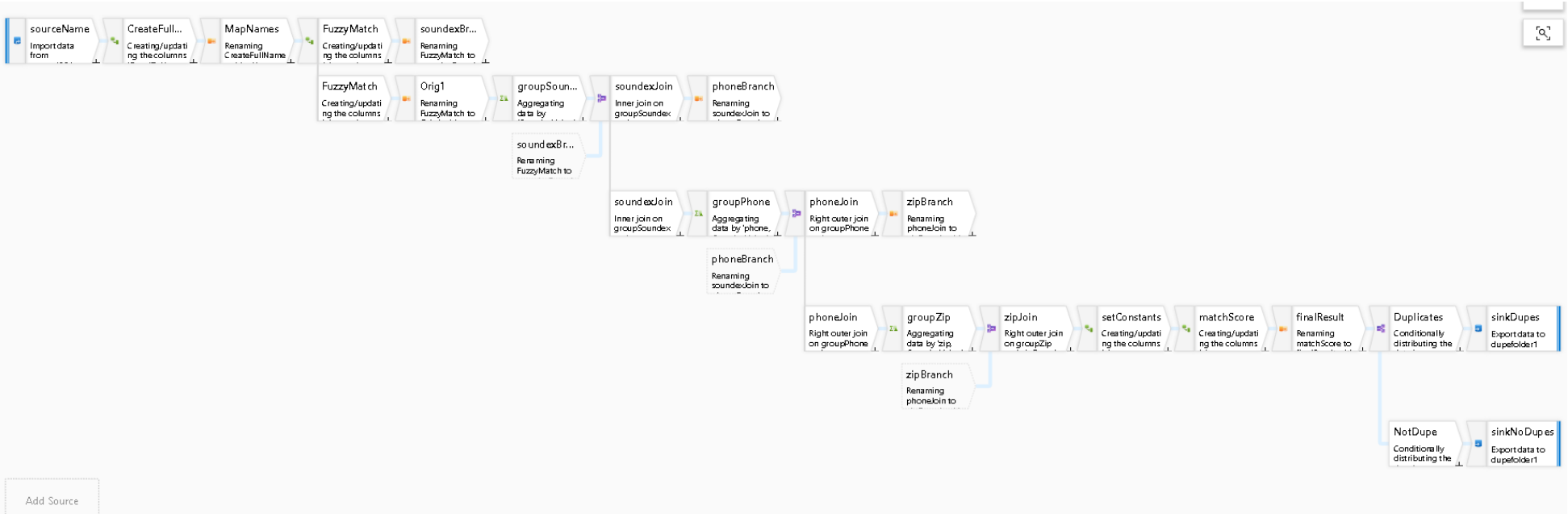


Metadata Validation Rules

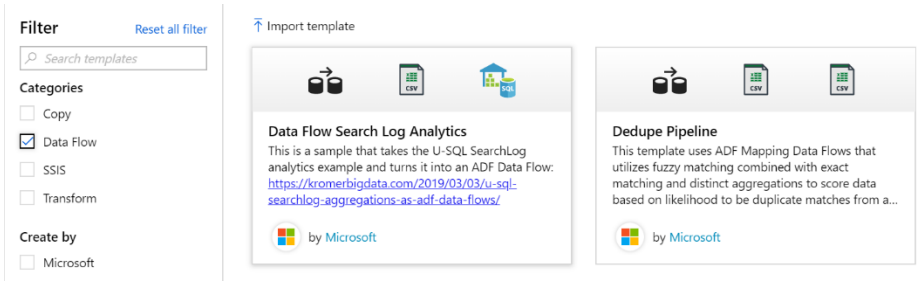


- Transform data conditionally based on metadata traits
- Create and manage metadata quality rules
- Manipulate column properties of source data
- https://www.youtube.com/watch?v=E_UD3R-VpYE

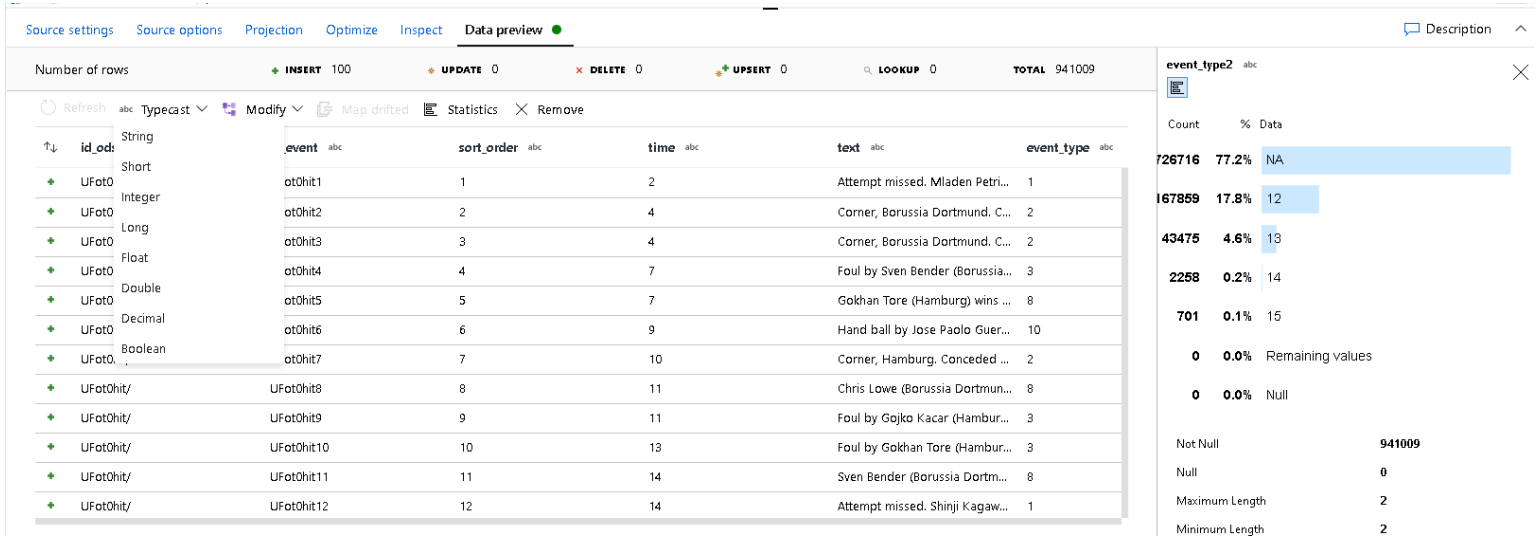
Data De-Duplication and Distinct Rows

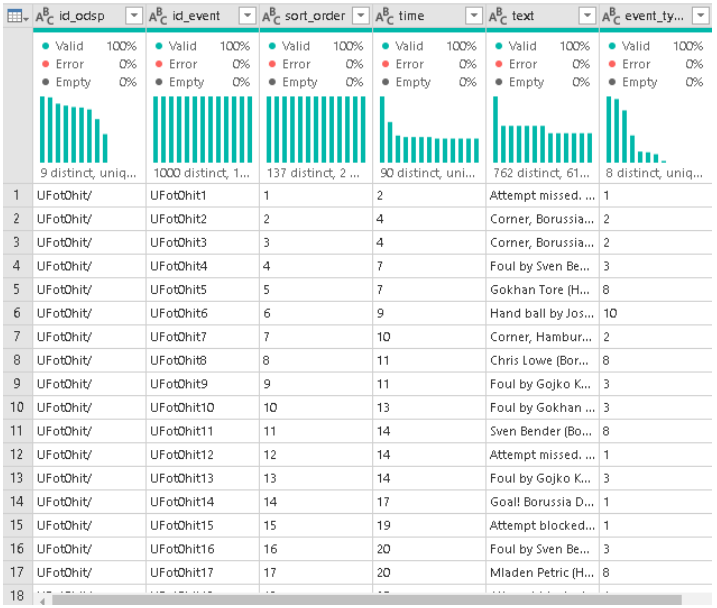


- Use this pattern to eliminate common rows from your data
- You pick a heuristic to use during duplicate matching
- You can tag rows and/or remove duplicate rows
- Use exact matching and/or fuzzy matching
- Available as pipeline template *Dedupe Pipeline*



Data Profiling

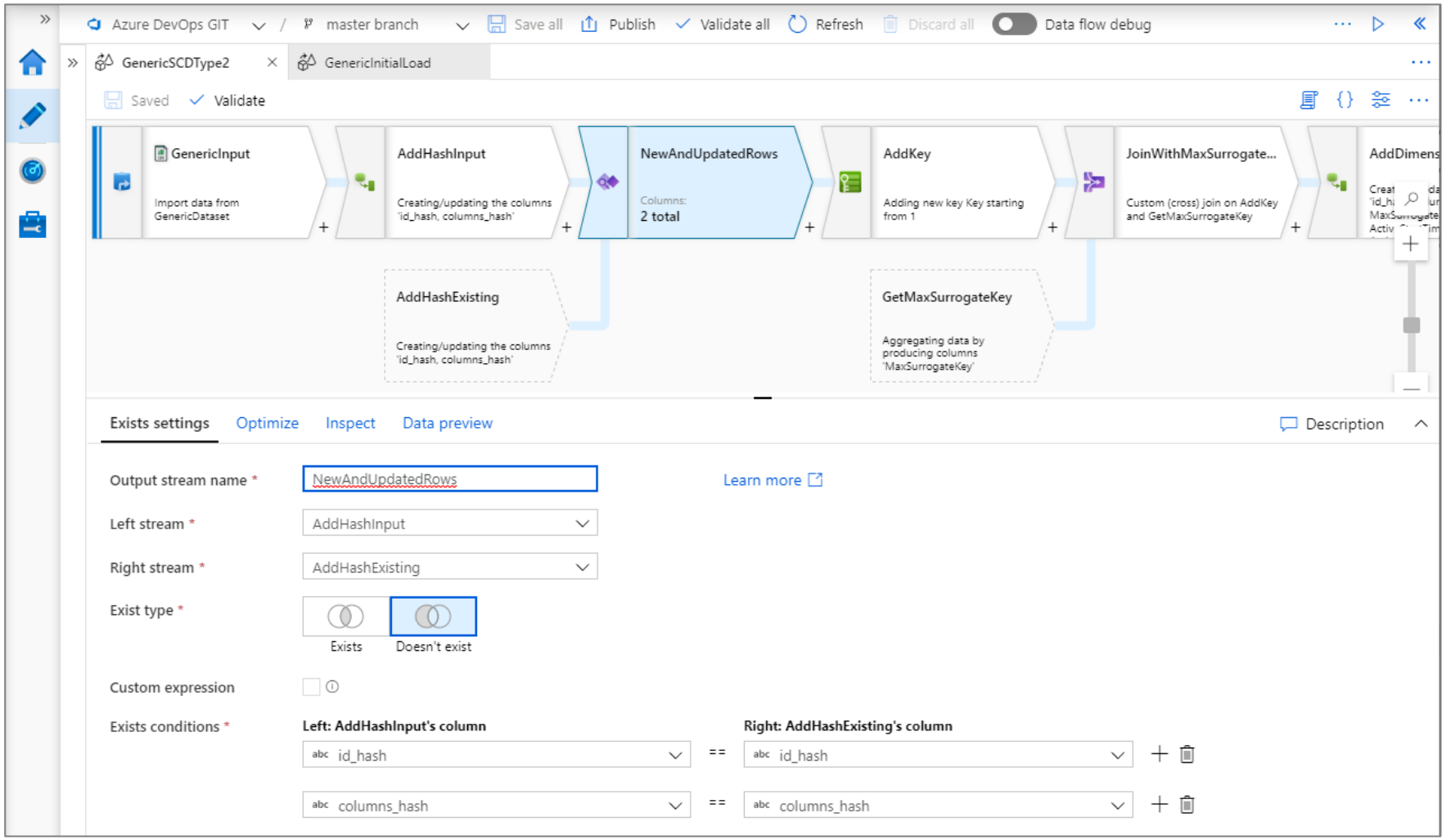




- Summary statistics describing the shape, size, content of your data
- Helps you to understand what is inside your data
- As a Data Engineer, you have the responsibility to provide proper data for analytics and models
- For big data sets, use sampling / good enough approach
- View data statistics at any step in your data transformation
- Here is how to persist your data profile stats: https://techcommunity.microsoft.com/t5/azure-data-factory/how-to-save-your-data-profiler-summary-stats-in-adf-data-flows/ba-p/1243251

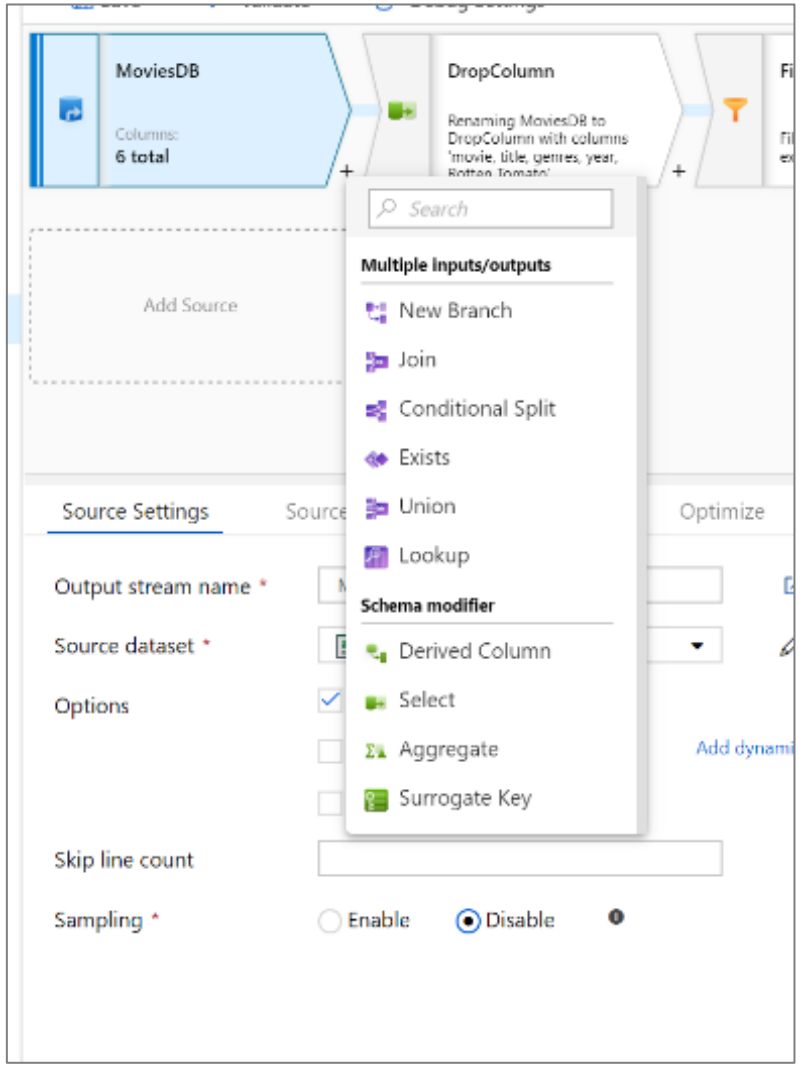
Authoring mapping data flows

Dedicated development canvas



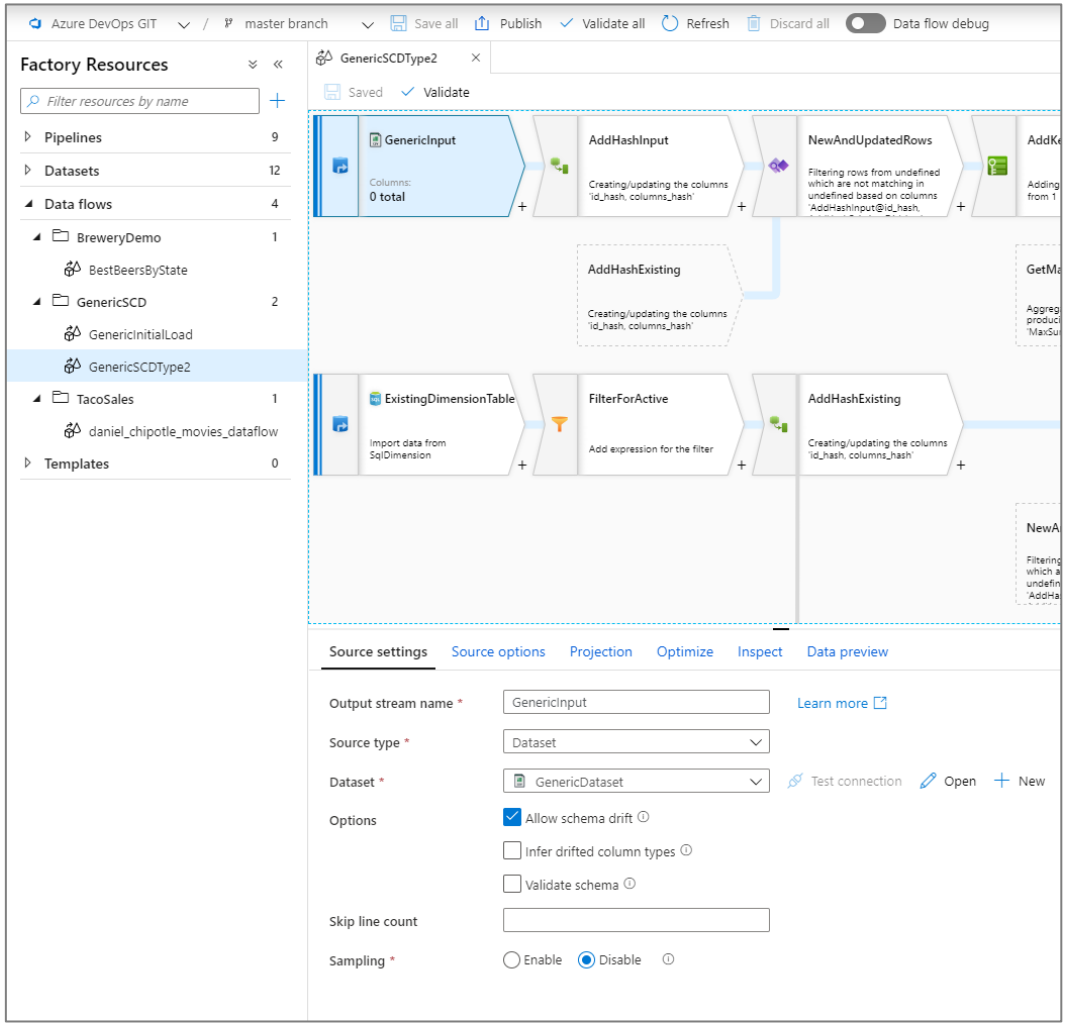
Building transformation logic

- Transformations: A ‘step’ in the data flow
  - Engine intelligently groups them at runtime
  - 19 currently available
- Core logic of data flow
  - Add/Remove/Alter Columns
  - Join or lookup data from datasets
  - Change number or order of rows
  - Aggregate data
  - Hierarchal to relational



Source transformation

- Define the data read by your data flow
  - Import projection vs generic
  - Schema drift
  - Connector specific properties and optimizations
- Min: 1, Max: ∞
- Define in-line or use dataset



Source: In-line vs dataset

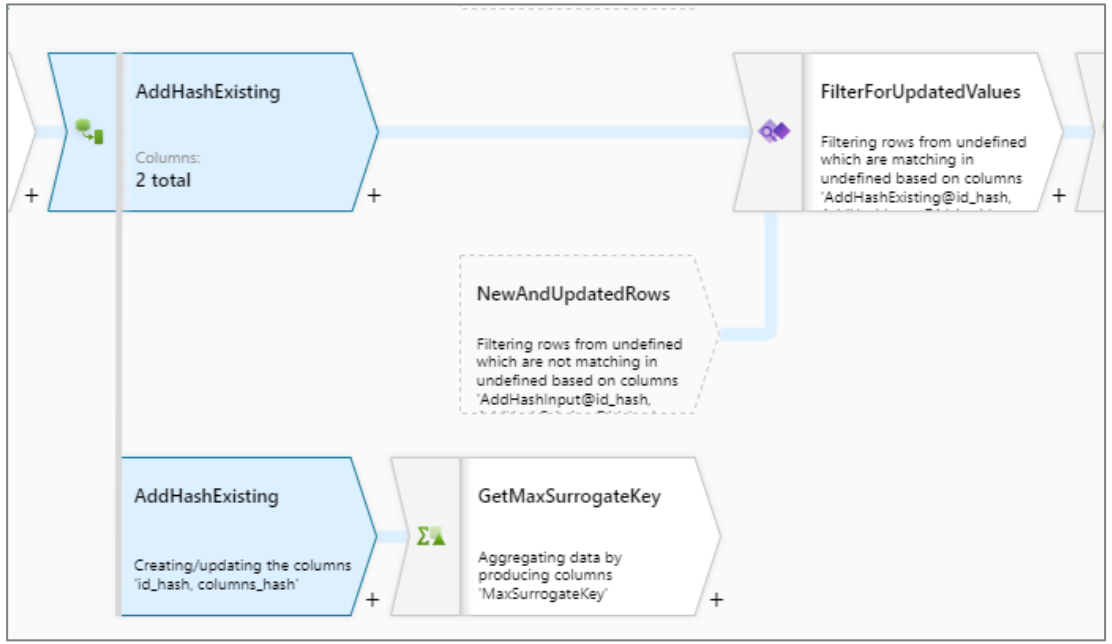
- Define all source properties within a data flow or use a separate entity to store them
- Dataset:
  - Reusable in other ADF activities such as Copy
  - Not based in Spark -> some settings overridden
- In-line
  - Useful when using flexible schemas, one-off source instances or parameterized sources
    - Do not need “dummy” dataset object
  - Based in Spark, properties native to data flow
- Most connectors only in available in one

Supported connectors

- File-based data stores (ADLS Gen1/Gen2, Azure Blob Storage)
  - Parquet, JSON, DelimitedText, Excel, Avro, XML
  - In-line only: Common Data Model, Delta Lake
- SQL tables
  - Azure SQL Database
  - Azure Synapse Analytics (formerly SQL DW)
- Cosmos DB
- Coming soon: Snowflake
- **If not supported, ingest to staging area via Copy activity**
  - 90+ connectors supported natively

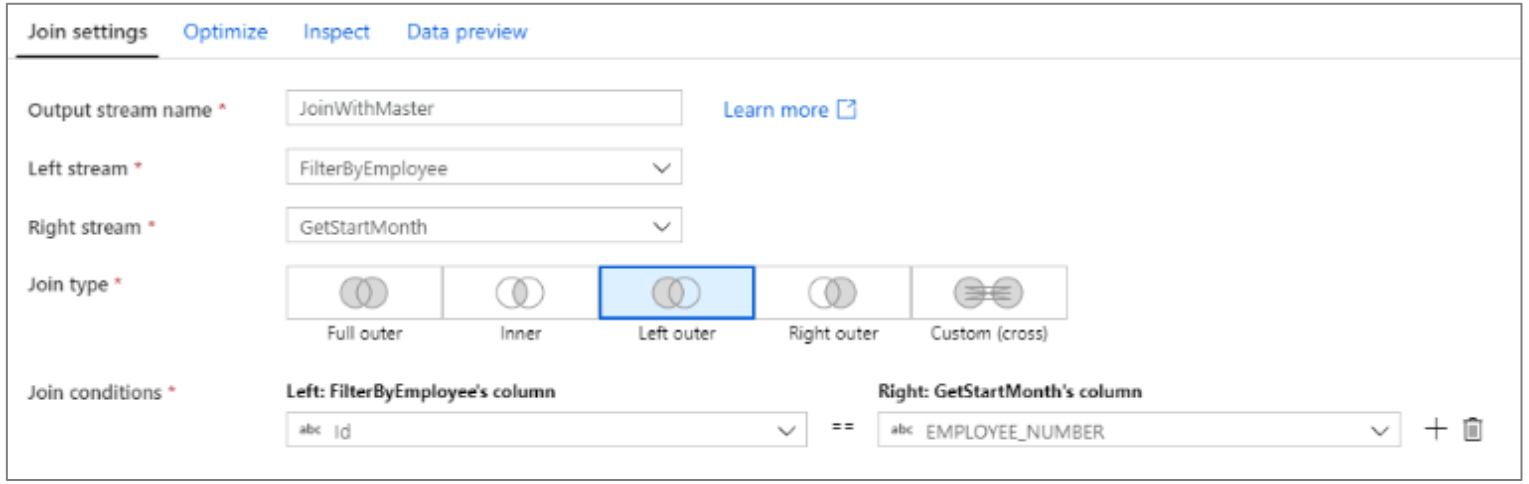
Duplicating data streams

- Duplicate data stream from any stage of your data flow
  - Select ‘New branch’
- Operate on same data with different transformation requirements
  - Self-joins
  - Writing to different sinks
  - Aggregating in one branch



Joining two data streams together

- Use *Join transformation* to append columns from incoming stream to any stream in your data flow
  - Join types: full outer, inner, left outer, right outer, cross
  - SQL Join equivalent
- Match on computed columns or use non-equality conditions
- Broadcast small data streams to cache data and improve performance



Lookup transformation

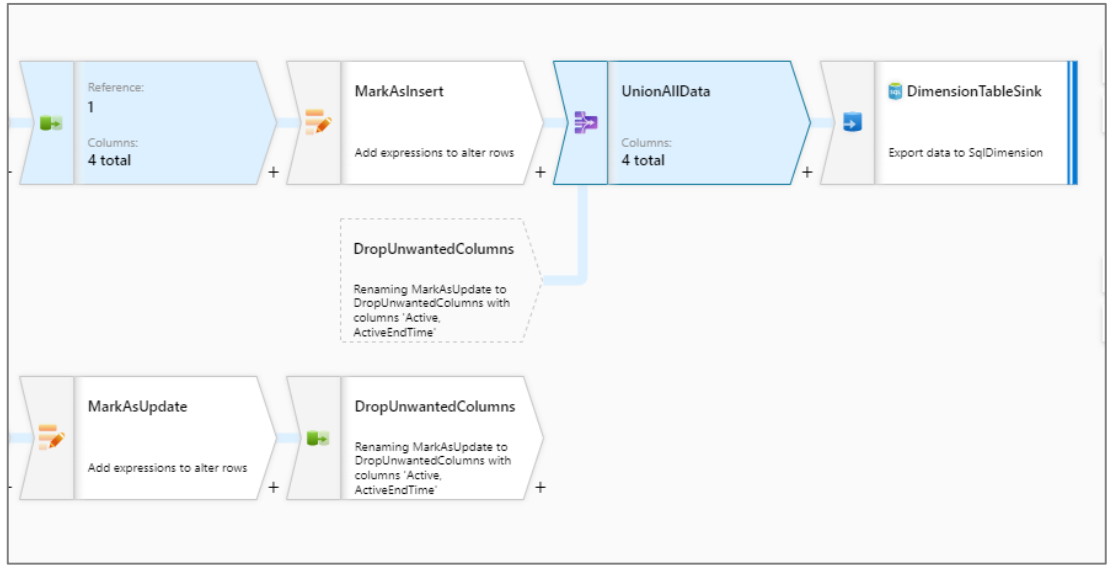
- Similar to left outer join, but with more functionality
  - All incoming rows are passed through regardless of match
- Matching conditions same as a join
- Multi or single row lookup
  - Match on all, first, last, or any row that meets join conditions
- *isMatch()* function can be used in downstream transformations to verify output

Exists transformation

- Check for existence of a value in another stream
  - SQL Exists equivalent
  - See if any row matches in a subquery, just like SQL
- Filter based on join matching conditions
- Choose **Exist** or **Not Exist** for your filter conditions
- Can specify a custom expressoin

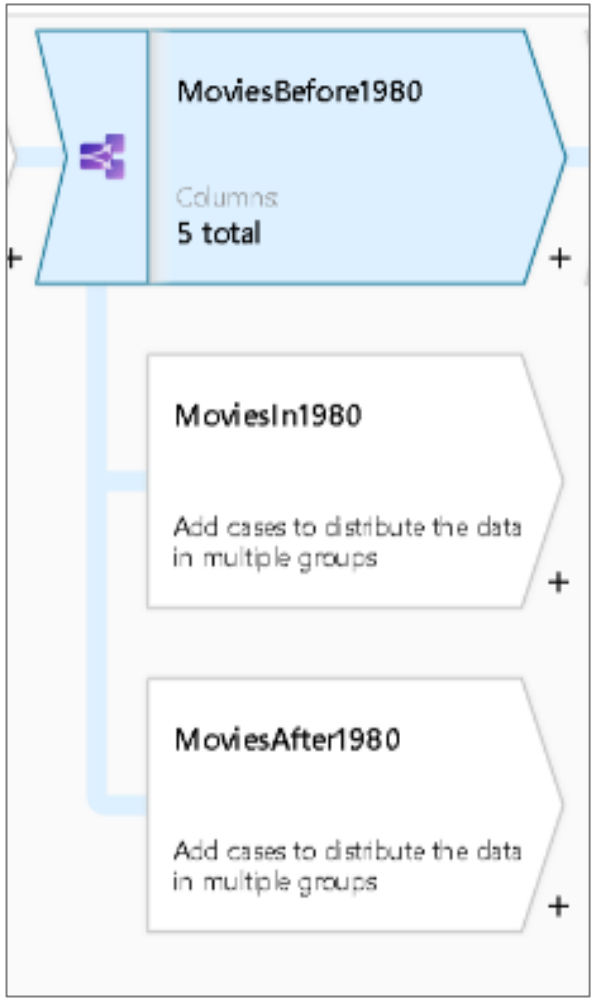
Union transformation

- Combine rows from multiple streams
- Add as many streams as needed
- Combine data based upon column name or ordinal column position
- Use cases:
  - Similar data from different connection that undergo same transformations
  - Writing multiple data streams into the same sink



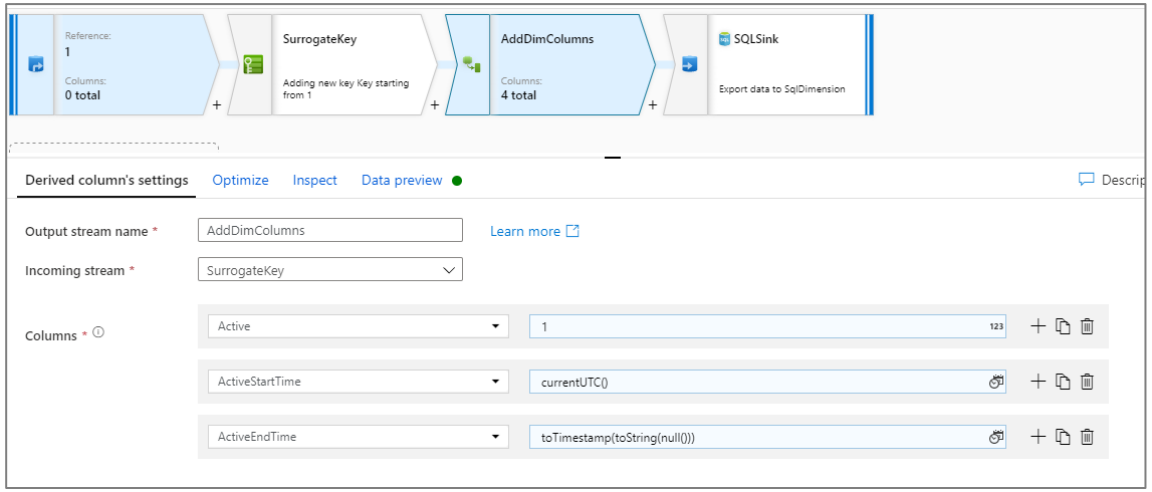
Conditional split

- Split data into separate streams based upon conditions
  - Use data flow expression language to evaluate boolean
- Use cases:
  - Sinking subset of data to different locations
  - Perform different calculations on data depending on a set of values



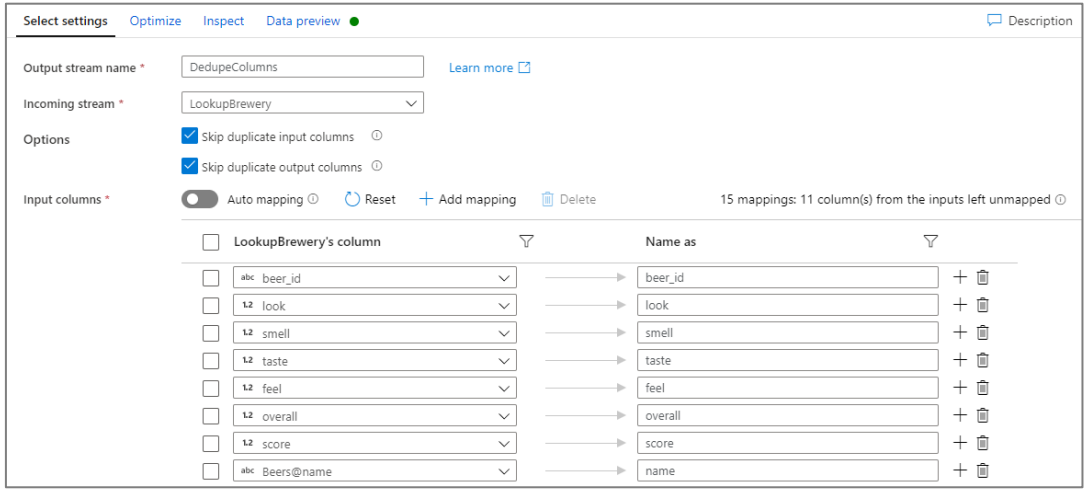
Derived column

- Transform data at row and column level using expression language
  - Generate new or modify existing columns
  - Build expressions using the expression builder
- Handle structured or unstructured data
  - Use column patterns to match on rules and regular expressions
  - Can be used to transform multiple columns in bulk
- Most heavily used transformation



Select transformation

- Metadata and column maintenance
  - SQL Select statement
- Alias or renames data stream and columns
- Prune unwanted or duplicate columns
  - Common after joins and lookups
- Rule-based mapping for flexible schemas, bulk mapping
  - Map hierarchal columns to flat structure

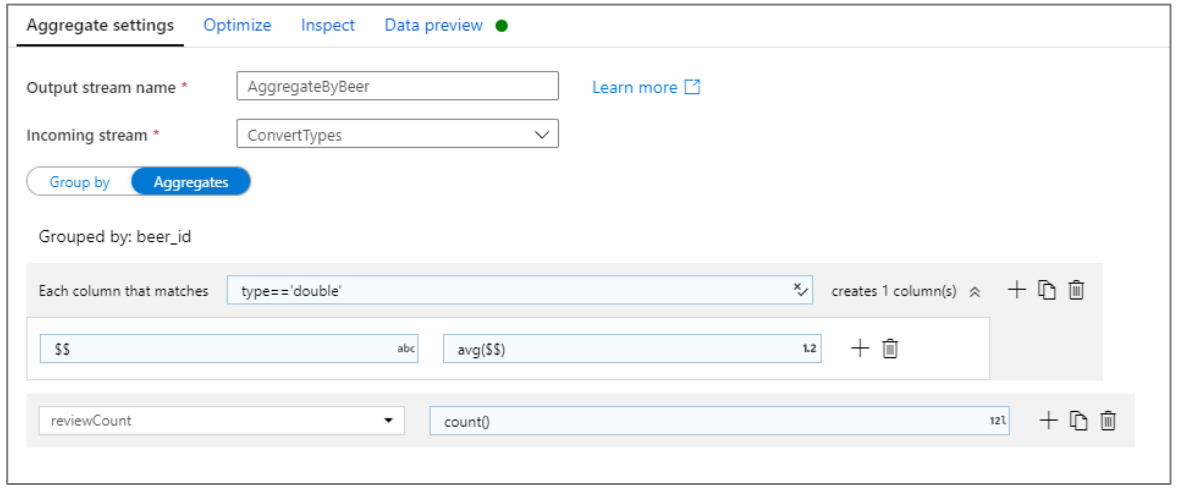


Surrogate key transformation

- Generate incrementing key to use as a non-business key in your data
- To seed the starting value of your surrogate key, use derived column and a lookup from an existing table
  - Examples are in documentation
- Useful for generating keys for star schema dimension tables

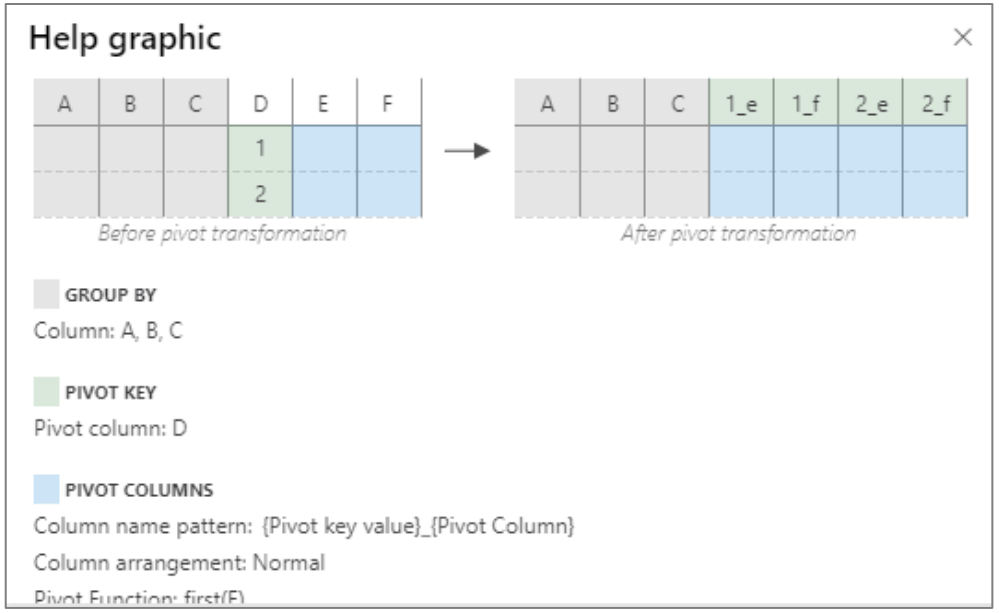
Aggregate transformation

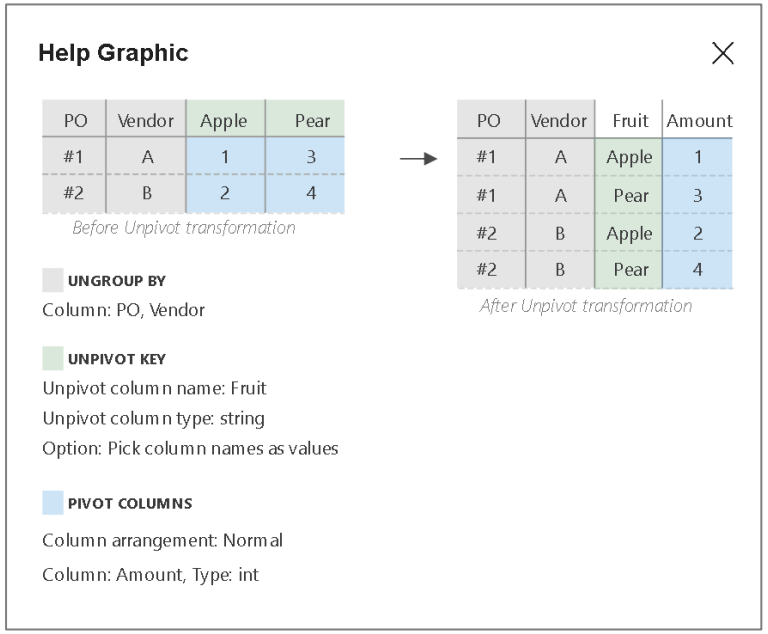
- Aggregate data into groups using aggregate function
  - Like SQL GROUP BY clause in a Select statement
  - Aggregate functions include *sum()*, *max()*, *avg()*, *first()*, *collect()*
- Choose columns to group by
  - One row for each unique group by column value
- Only columns used in transformation are in output data stream
  - Use self-join to append to existing data
- Supports pattern matching



Pivot and unpivot transformations

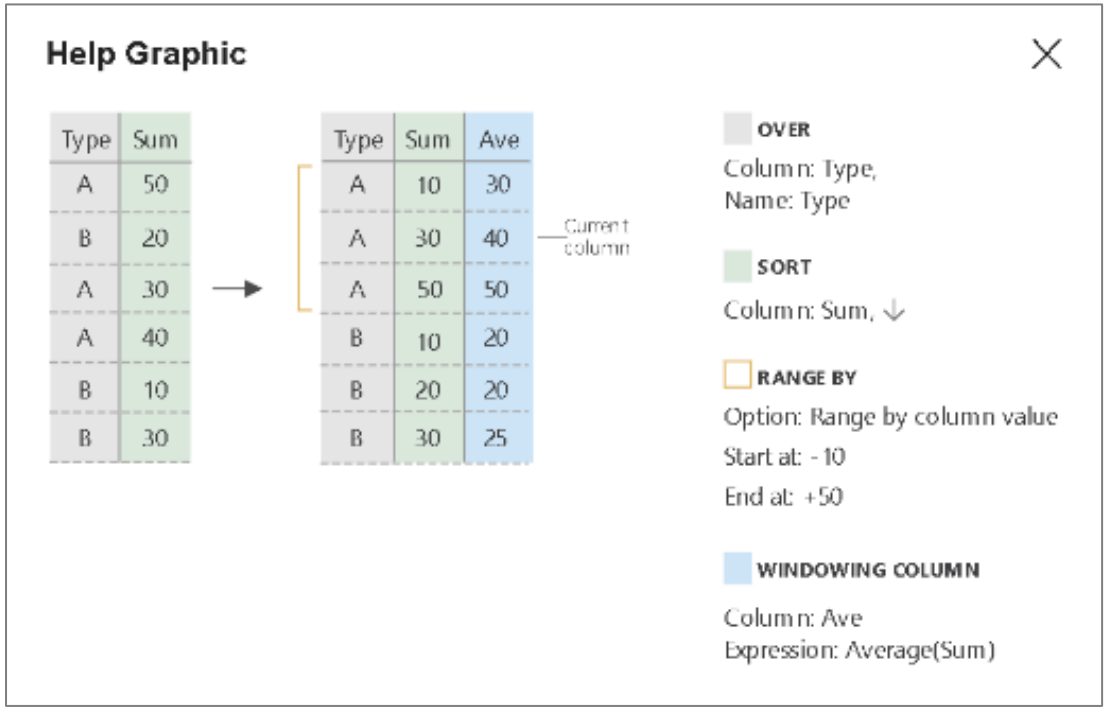
- Pivot row values into new columns and vice-versa
- Both are aggregate transformations that require aggregate functions
- If pivot key values not specified, all columns become drifted
  - Use map drifted quick action to add to schema quickly





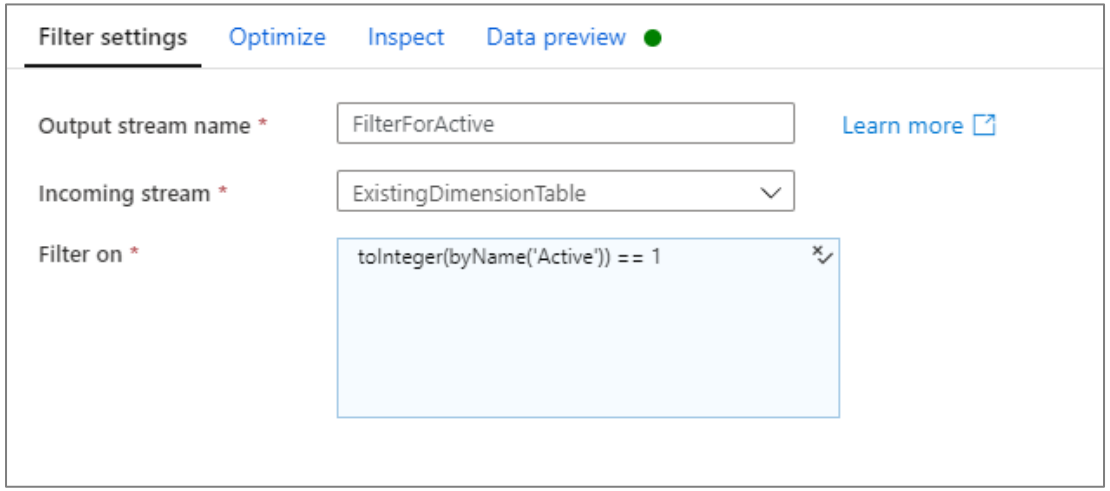
Window transformation

- Aggregate data across “windows” of data partitions
  - Used to compare a row of data against others in its ‘group’
- Group determined by group by columns, sorting conditions and range bounds
- Used for ranking rows in a group and getting lead/lag
- Sorting causes reshuffling of data
  - “Expensive” operation



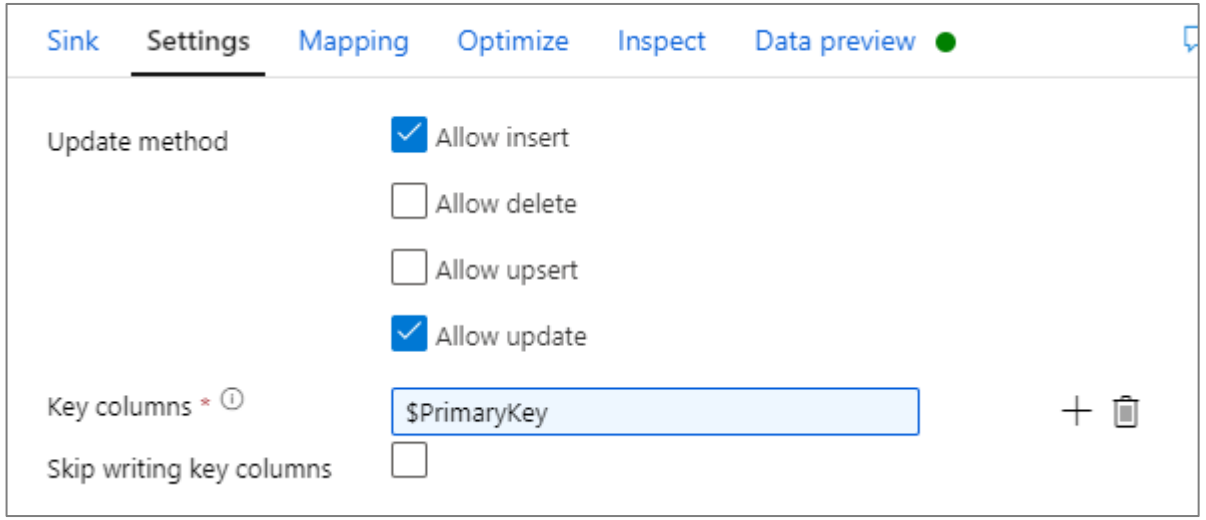
Filter transformation

- Filter rows based upon an expression
  - Like SQL WHERE clause
- Expressions return true or false



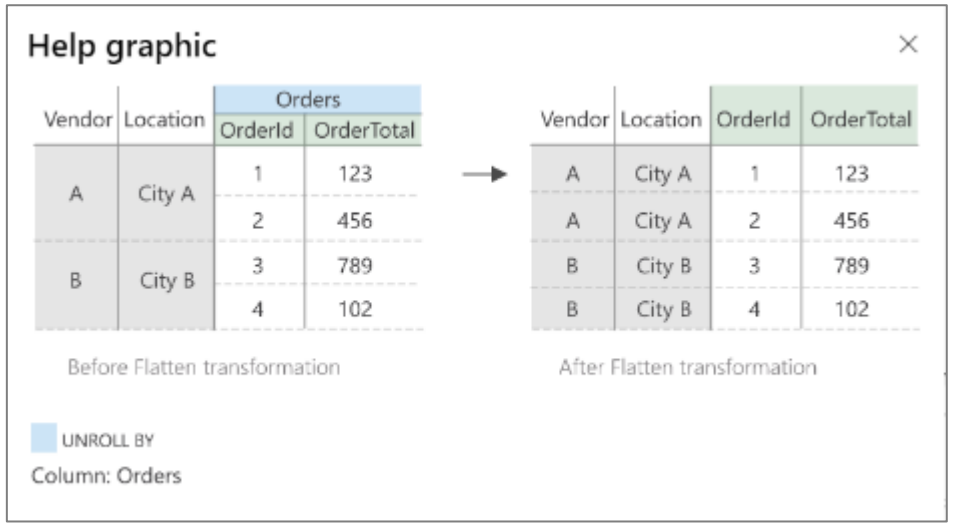
Alter row transformation

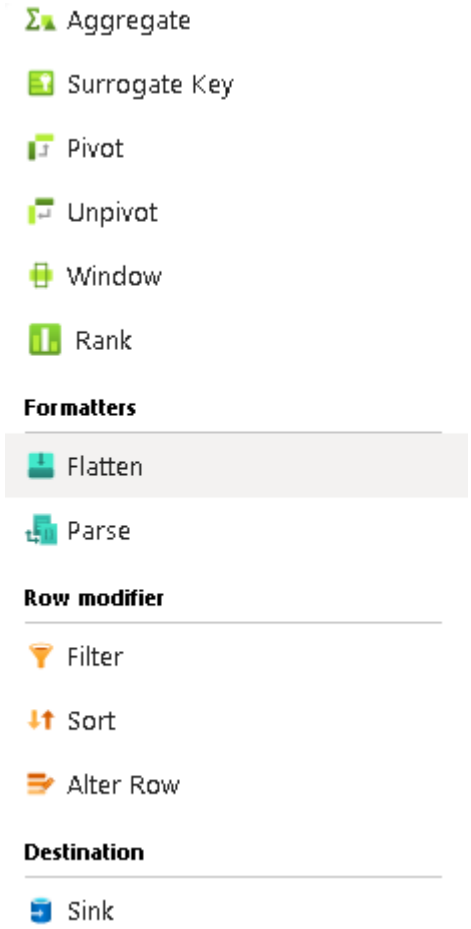
- Mark rows as Insert, Update, Delete, or Upsert
  - Like SQL MERGE statement
  - Insert by default
- Define policies to update your database
  - Works with SQL DB, Synapse, Cosmos DB, and Delta Lake
- Specify allowed update methods in each sink



Flatten transformation

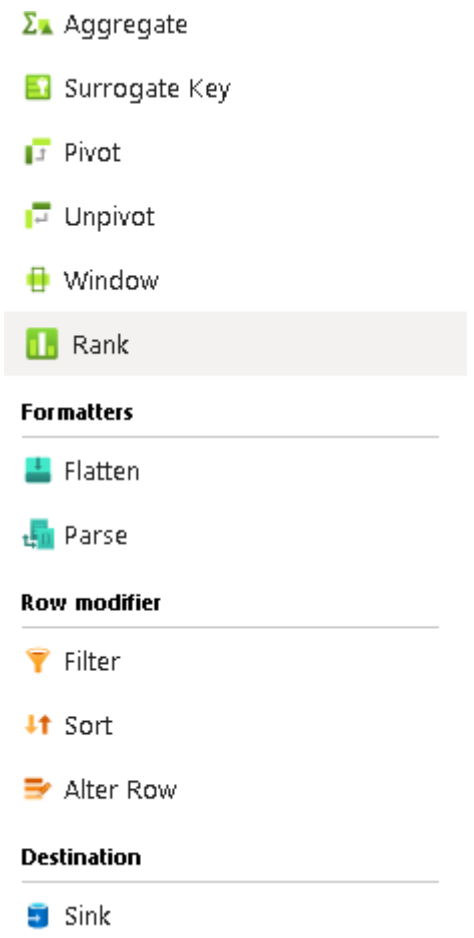
- Unroll array values into individual rows
  - One row per value
- Used to convert hierarchies to flat structures
- Opposite of collect() aggregate function





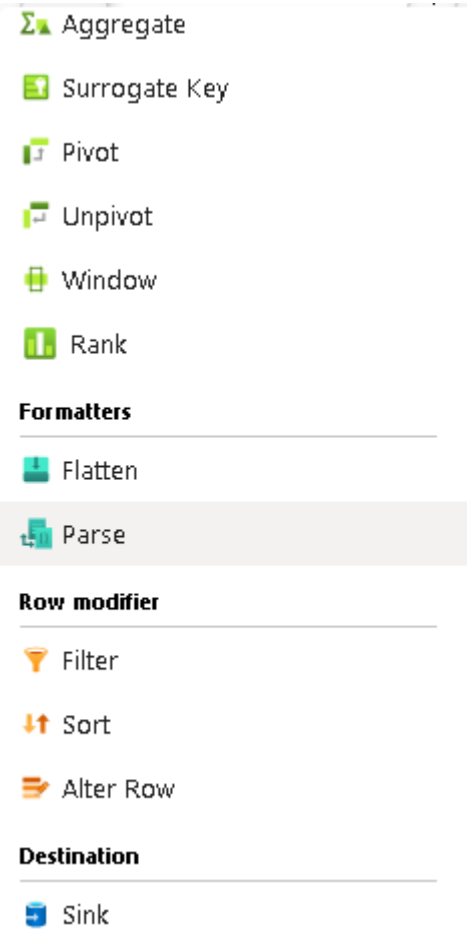
Rank transformation

- Rank data across an entire dataset
- Same as Rank() function in Window transformation,but scales better for a ranking an entire dataset
- For ranking of data partitions, use Window rank()
- For ranking entire dataset, use Rank transformation



Parse transformation

- Parses string data from columns that are formatted text
- Currently supported formats: JSON, text delimited
- Ex: Turn plain text JSON strings from a source file in a single column into formatted JSON output

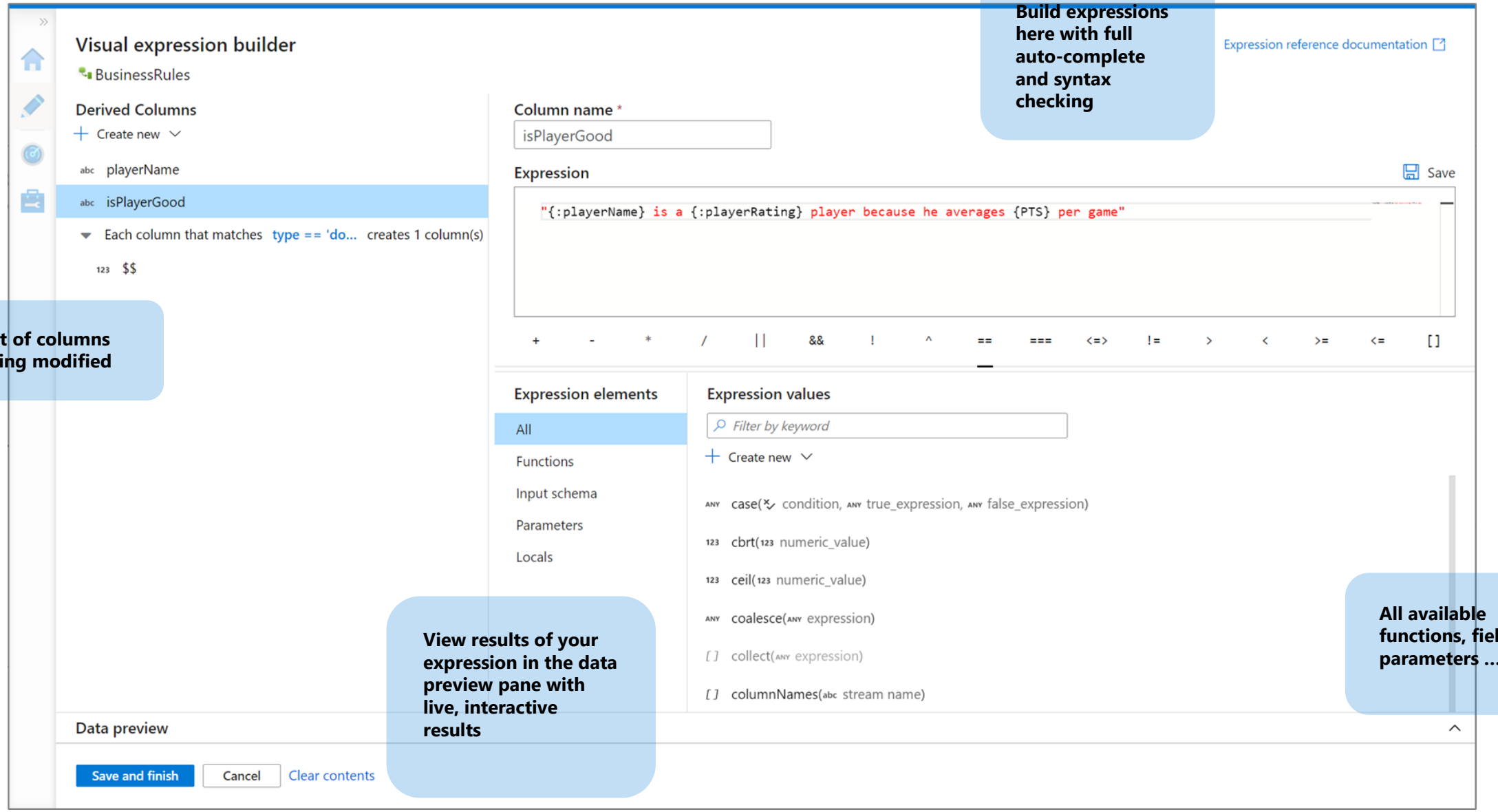


Sink transformation

- Define the properties for landing your data in your destination target data store
  - Define using dataset or in-line
- Can map columns similar to select transformation
  - Import schema definition from destination
- Set actions on destinations
  - Truncate table or clear folder, SQL pre/post actions, database update methods
- Choose how the written data is partitioned
  - Use current partitioning is almost always fastest
- **Note: Writing to single file can be very slow with large amounts of data**

Mapping data flow expression language

Visual expression builder



Build expressions here with full auto-complete and syntax checking

List of columns being modified

View results of your expression in the data preview pane with live, interactive results

All available functions, fields, parameters ...

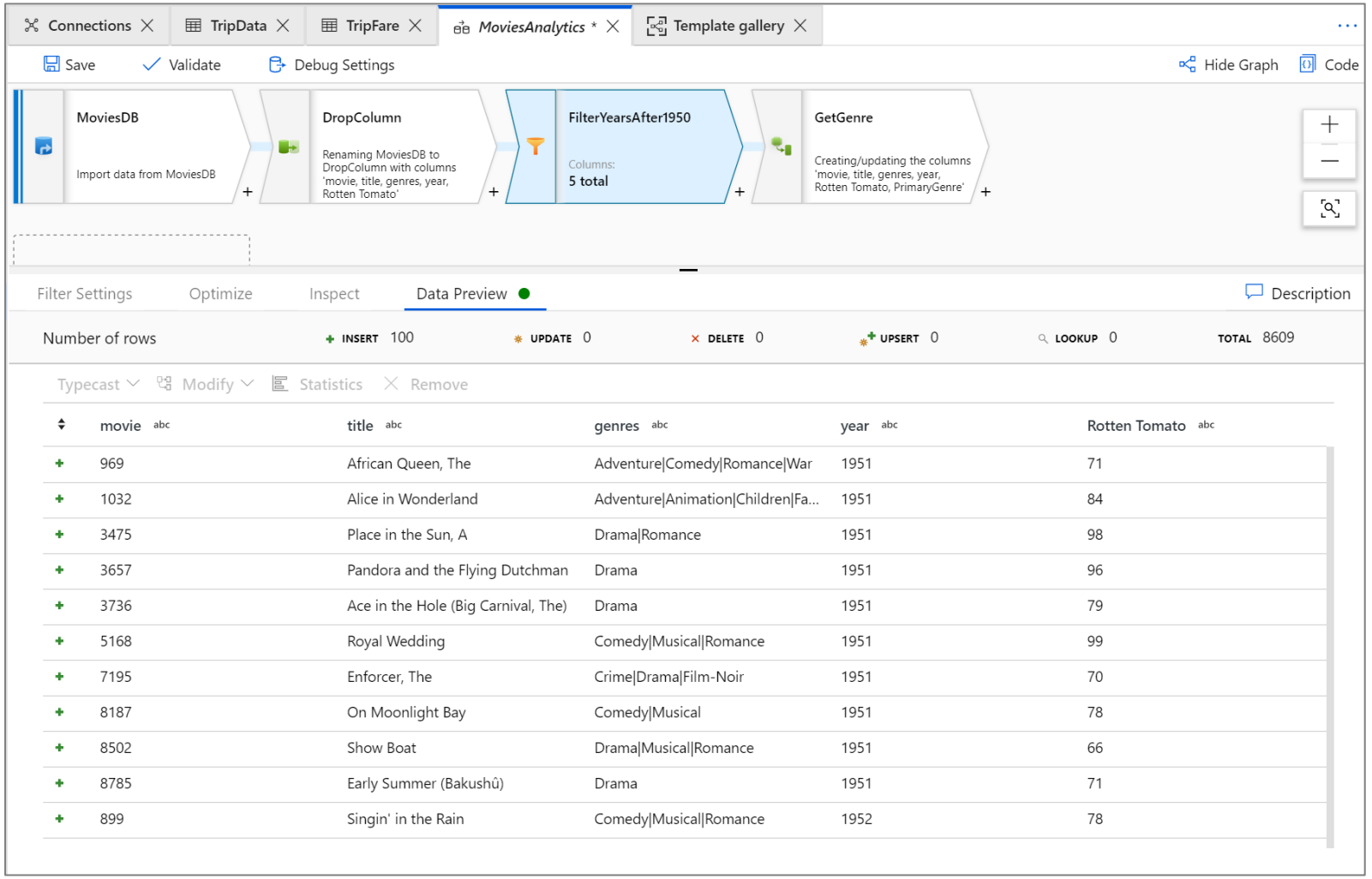
Expression language

- Expressions are built using the data flow expression language
- Expressions can reference:
  - Built-in expression functions
  - Defined input schema columns
  - Data flow parameters
  - Literals
- Certain transformations have unique functions
  - Count(), sum() in Aggregate, denseRank() in Window, etc
- Evaluates to spark data types
- 100s of transformation, analytical, array, metadata, string, and mathematical functions available

Debug mode

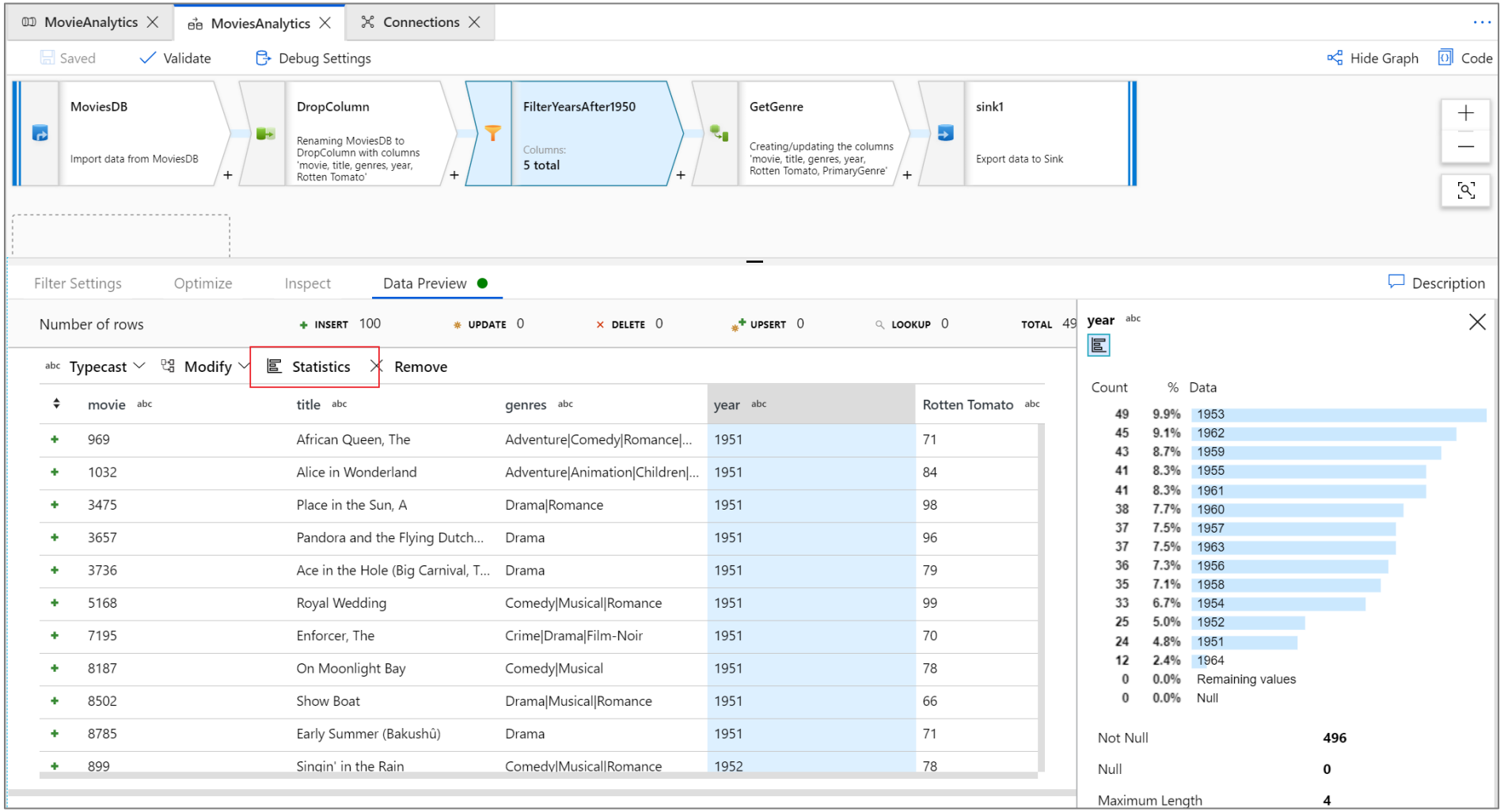
- Quickly verify logic during development on small interactive cluster
  - 4 core, 60-minute time to live
- Enables the following:
  - Get data preview snapshot at each transformation
  - Preview output of expression in expression builder
  - Run debug pipeline with no spin up
  - Import Spark projection of source schema
- **Rule of thumb**: If developing Data Flows, turn on right away
  - Initial 3-5-minute start up time

Debug mode: data preview

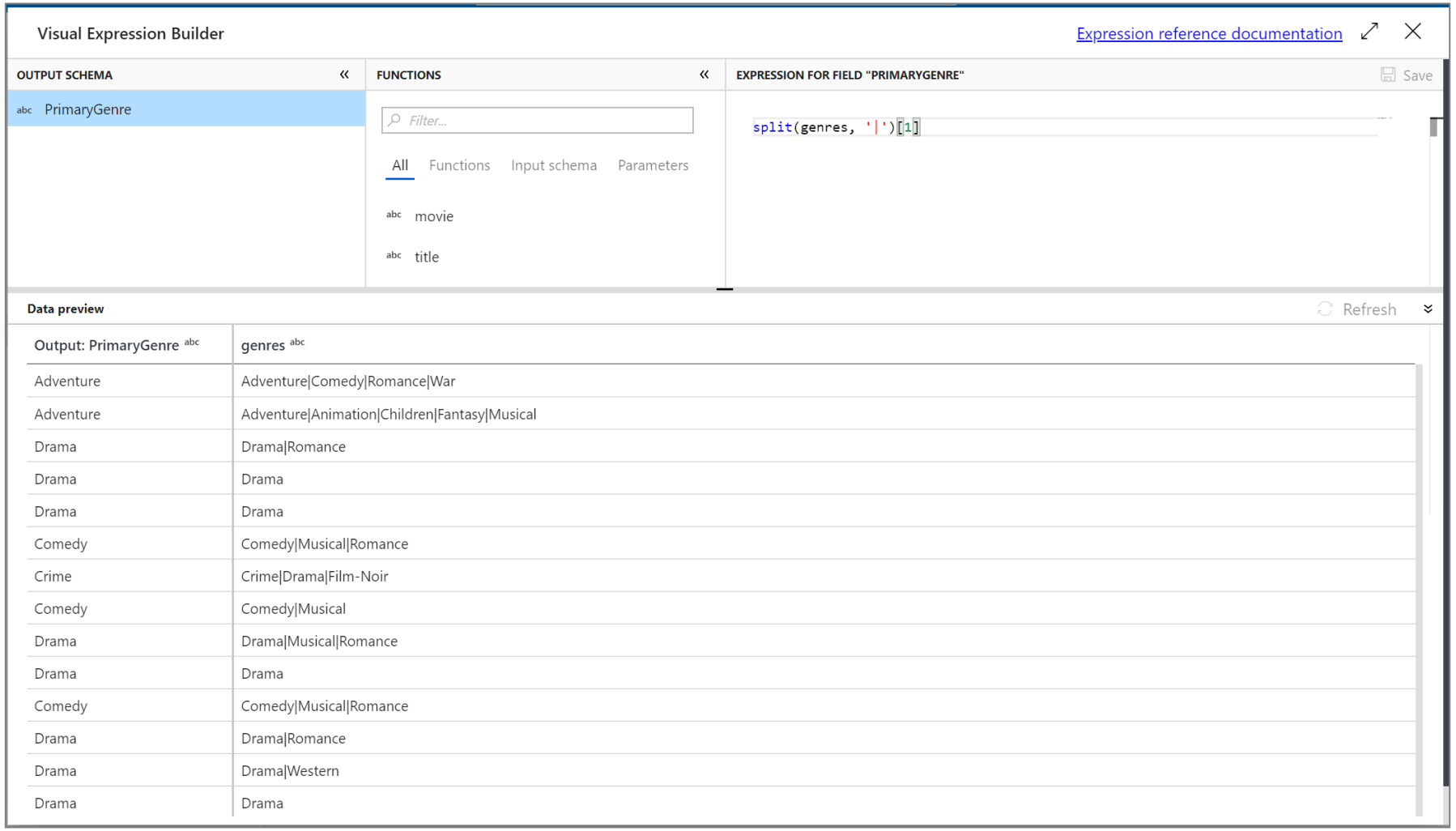


movie	title	genres	year	Rotten Tomato
969	African Queen, The	Adventure|Comedy|Romance|War	1951	71
1032	Alice in Wonderland	Adventure|Animation|Children|Fa...	1951	84
3475	Place in the Sun, A	Drama|Romance	1951	98
3657	Pandora and the Flying Dutchman	Drama	1951	96
3736	Ace in the Hole (Big Carnival, The)	Drama	1951	79
5168	Royal Wedding	Comedy|Musical|Romance	1951	99
7195	Enforcer, The	Crime|Drama|Film-Noir	1951	70
8187	On Moonlight Bay	Comedy|Musical	1951	78
8502	Show Boat	Drama|Musical|Romance	1951	66
8785	Early Summer (Bakushû)	Drama	1951	71
899	Singin' in the Rain	Comedy|Musical|Romance	1952	78

Debug mode: data profiling



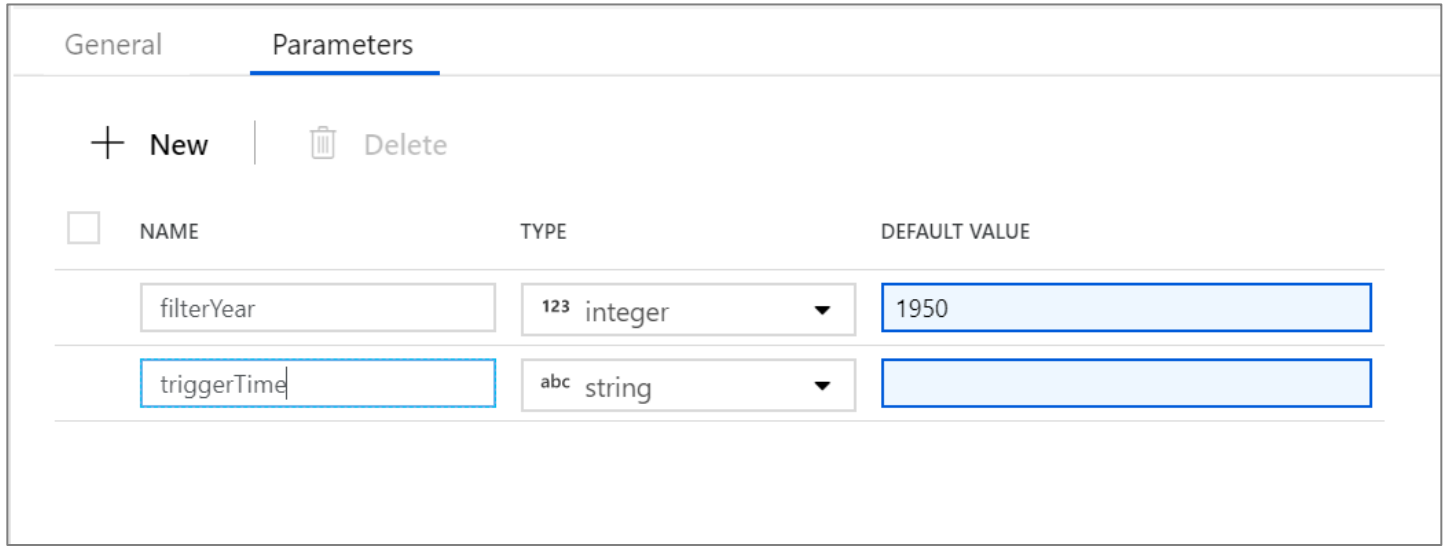
Debug mode: expression output



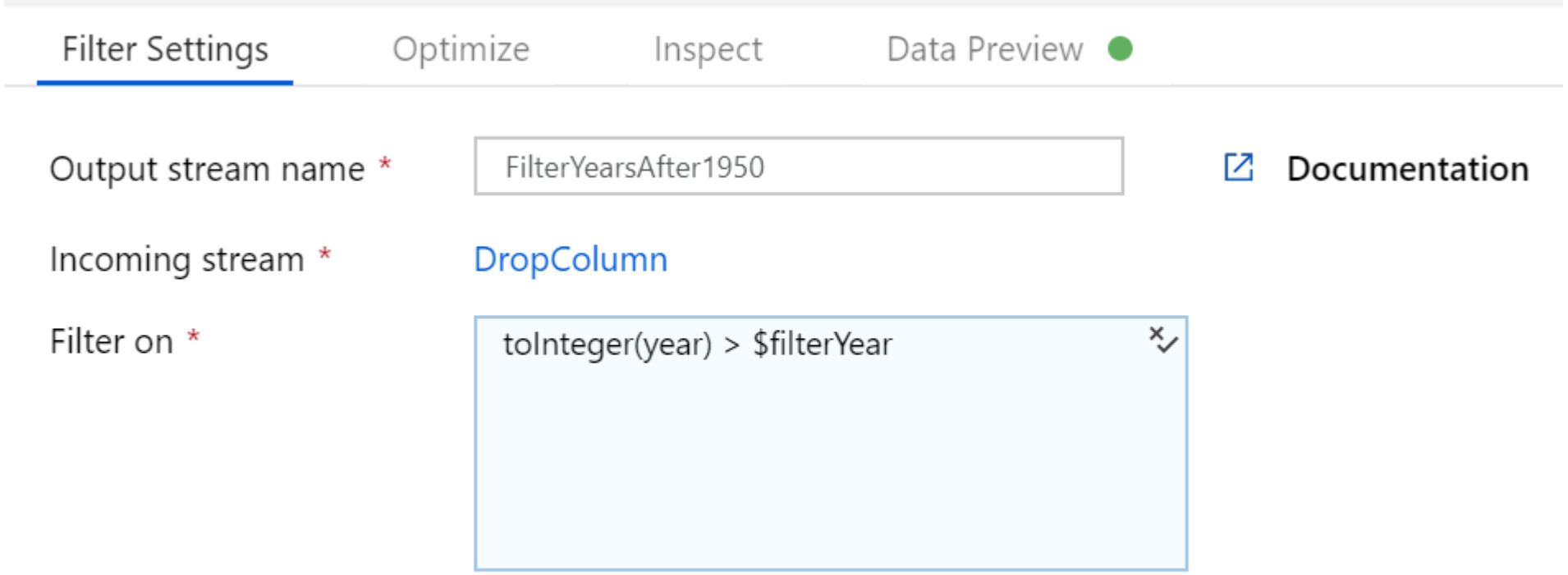
Output: PrimaryGenre abc	genres abc
Adventure	Adventure|Comedy|Romance|War
Adventure	Adventure|Animation|Children|Fantasy|Musical
Drama	Drama|Romance
Drama	Drama
Drama	Drama
Comedy	Comedy|Musical|Romance
Crime	Crime|Drama|Film-Noir
Comedy	Comedy|Musical
Drama	Drama|Musical|Romance
Drama	Drama
Comedy	Comedy|Musical|Romance
Drama	Drama|Romance
Drama	Drama|Western
Drama	Drama

Parameterizing data flows

- Both dataset properties and data-flow expressions can be parameterized
  - Passed in via data flow activity
    - Can use data flow or pipeline expression language
  - Expressions can reference *$parameterName*
- Can be literal values or column references



Referencing data flow parameters



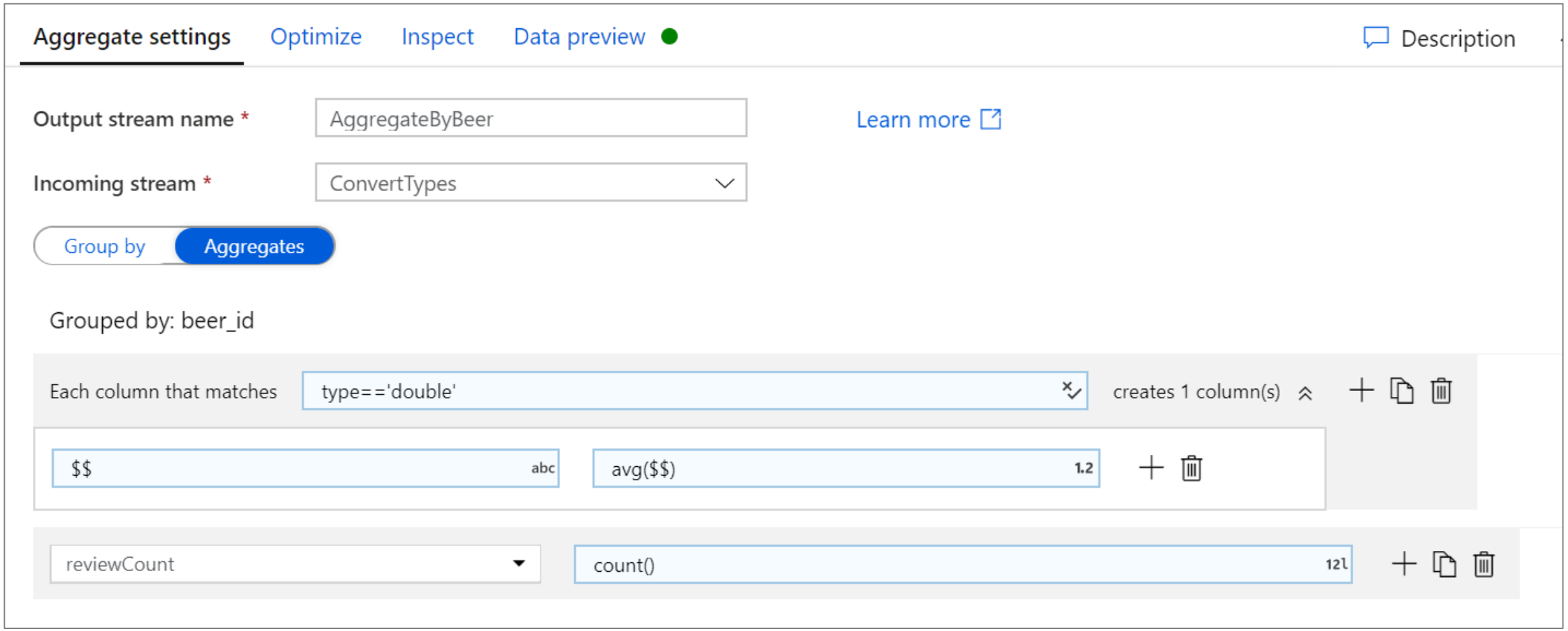
Working with flexible schemas

Schema drift

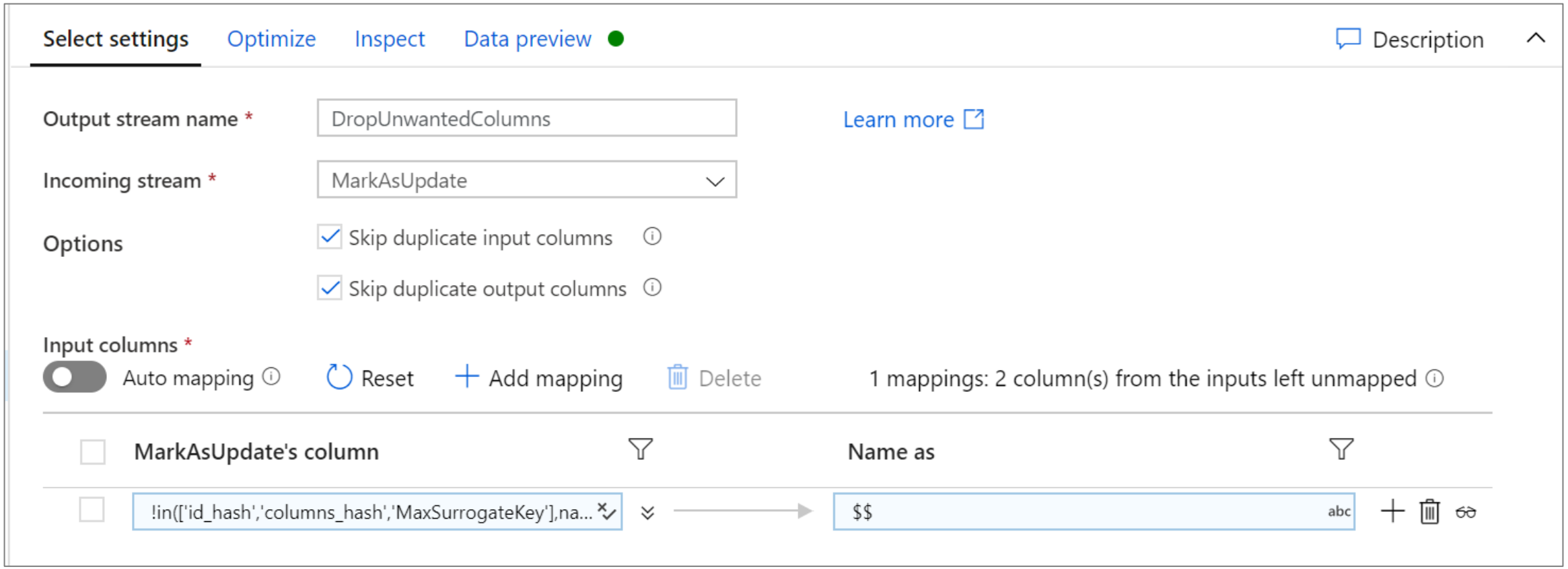
- In real-world data integration solutions, source/target data stores change shape
  - Source data fields can change names
  - Number of columns can change over time
- Traditional ETL processes break when schemas drift
- Mapping data flow has built-in handling for flexible schemas
  - Patterns, rule-based mappings, byName(s) function, etc
  - Source: Read additional columns on top of what is defined in the source schema
  - Sink: Write additional columns on top of what is defined in the sink schema

Column pattern matching

- Match by name, type, stream, position



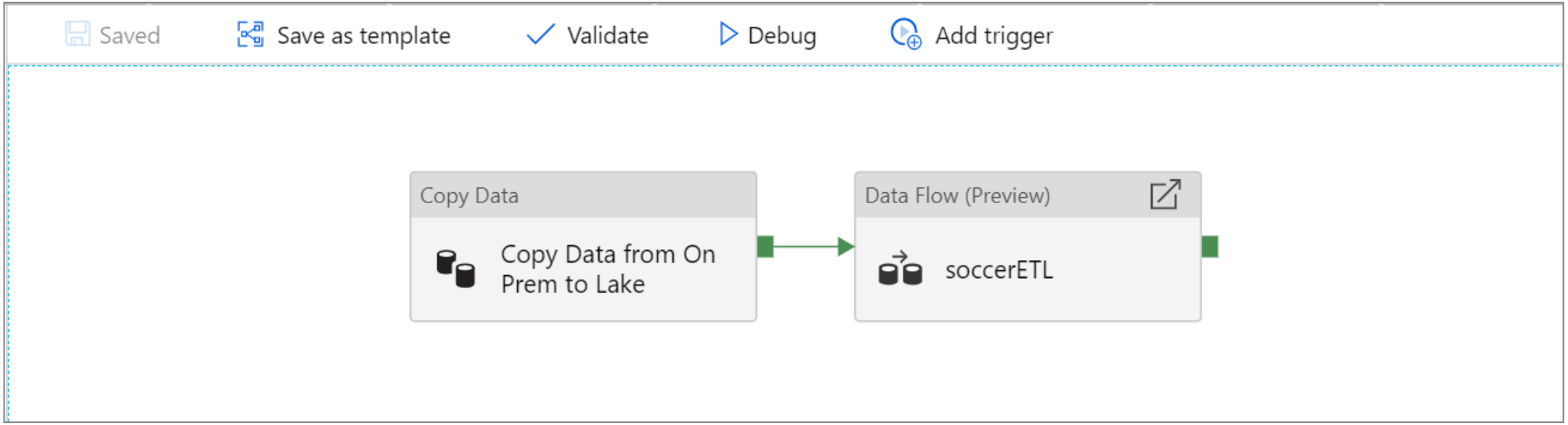
Rule-based mapping



Operationalizing and monitoring data flows

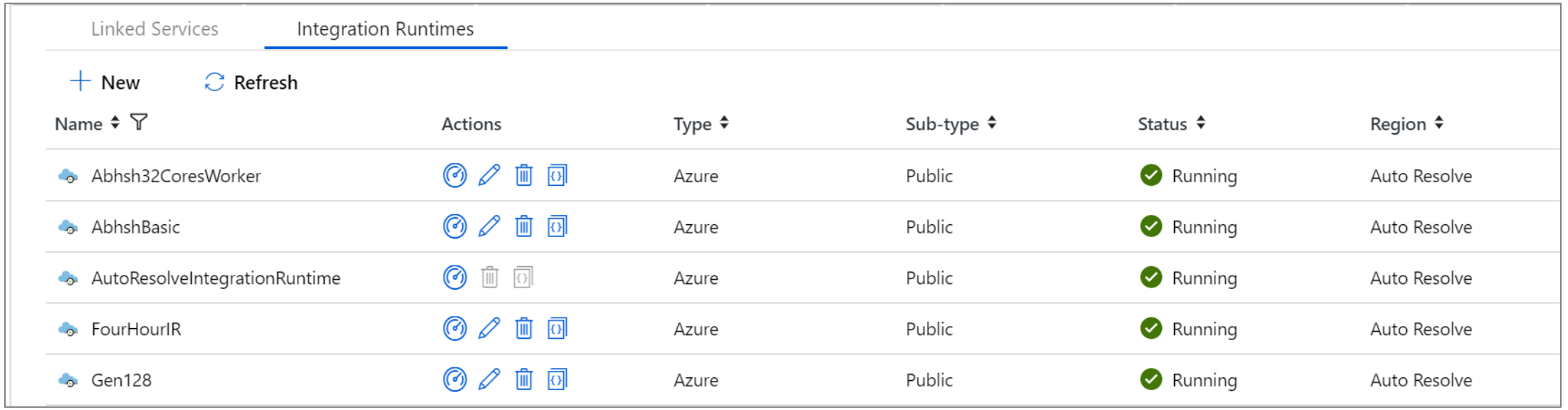
Data flow activity

- Run as activity in pipeline
  - Integrated with existing ADF control flow, scheduling, orchestration, montoring, C/ICD
- Choose which integration runtime (IR) to run on
  - # of cores, compute type, cluster time to live
- Assign parameters

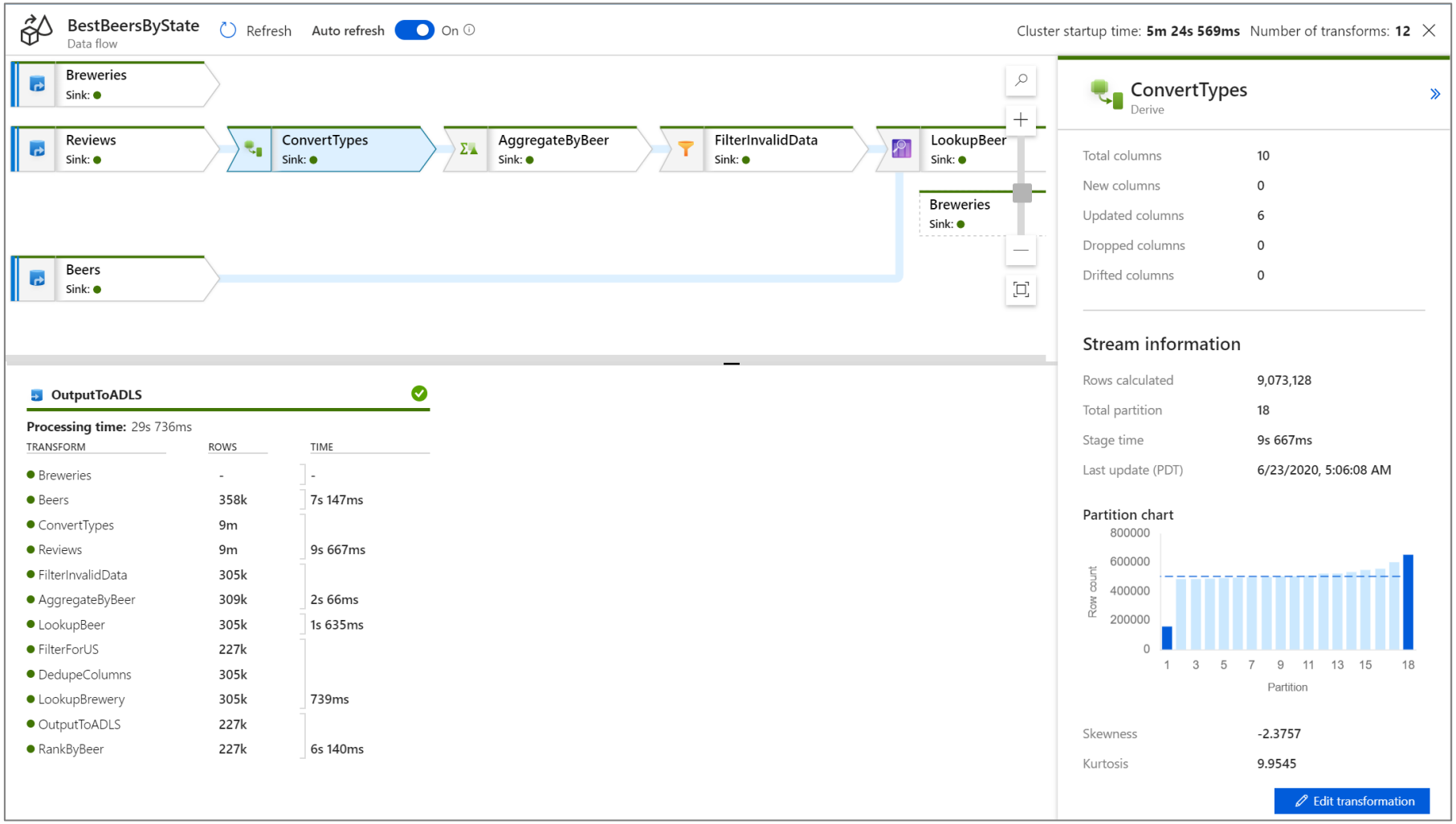


Data flow integration runtime

- Integrated with existing Azure IR
  - Choose compute type, # of cores, time to live
- **Time to live:** time a cluster is alive after last execution concludes
  - Minimal start up time for sequential data flows
- Parameterize compute type, # of cores if using *Auto Resolve*



Monitoring data flows



Data flow security considerations

- All data stays inside VMs that run the Databricks cluster which are spun up JIT for each job
  - Azure Databricks attaches storage to the VMs for logging and spill-over from in-memory data frames during job operation. These storage accounts are fully encrypted and within the Microsoft tenant.
  - Each cluster is single-tenant and specific to your data and job. This cluster is not shared with any other tenant
- Data flow processes are completely ephemeral. Once a job is completed, all associated resources are destroyed
  - Both cluster and storage account are deleted
- Data transfers in data flows are protected using certificates
- Active telemetry is logged and maintained for 45 days for troubleshooting by the Azure Data Factory team

Data flow best practices and optimizations

Best practices – Lifecycle

1. Test your transformation logic using debug mode and data preview
   - Limited source size or use sample files
2. Test end-to-end pipeline logic using pipeline debug
   - Verify data is read/written correctly
   - Used as smoke test before merging your changes
3. Publish and trigger your pipelines within a Dev Factory
   - Test performance and cluster size
4. Promote pipelines to higher environments such as UAT and PROD using CI/CD
   - Increase size and scope of data as you get to higher environments

Best practices – Debug (Data Preview)

- Data Preview
  - Data preview is inside the data flow designer transformation properties
  - Uses row limits and sampling techniques to preview data from a small size of data
  - Allows you to build and validate units of logic with samples of data in real time
  - You have control over the size of the data limits under Debug Settings
  - If you wish to test with larger datasets, set a larger compute size in the Azure IR when switching on “Debug Mode”
  - Data Preview is only a snapshot of data in memory from Spark data frames. This feature does not write any data, so the sink drivers are not utilized and not tested in this mode.

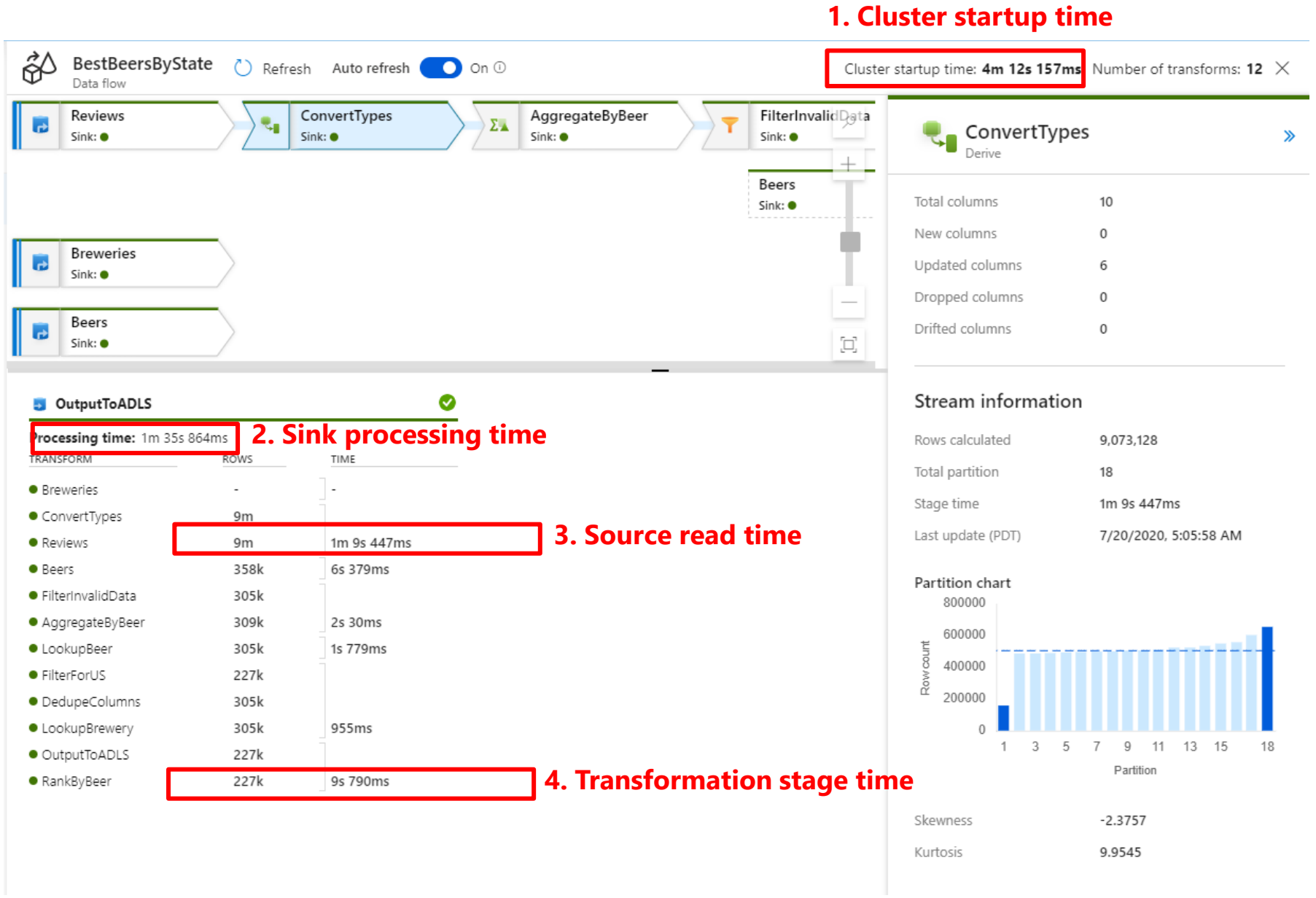
Best practices – Debug (Pipeline Debug)

- Pipeline Debug
  - Click debug button to test your data flow inside of a pipeline
  - Default debug limits the execution runtime so you will want to limit data sizes
  - Sampling can be applied here as well by using the “Enable Sampling” option in each Source
  - Use the debug button option of “use activity IR” when you wish to use a job execution compute environment
    - This option is good for debugging with larger datasets. It will not have the same execution timeout limit as the default debug setting

Optimizing data flows

- Transformation order generally does not matter
  - Data flows have a Spark optimizer that reorders logic to perform as best as it can
  - Repartitioning and reshuffling data negates optimizer
- Each transformation has ‘Optimize’ tab to control partitioning strategies
  - Generally do not need to alter
- Altering cluster size and type has performance impact
- Four components
  1. Cluster startup time
  2. Reading from sources
  3. Transformation time
  4. Writing to sinks

Identifying bottlenecks



1. Sequential executions can lower the cluster startup time by setting a TTL in Azure IR
2. Total time to process the stream from source to sink. There is also a post-processing time when you click on the Sink that will show you how much time Spark had to spend with partition and job clean-up. Write to single file and slow database connections will increase this time
3. Shows you how long it took to read data from source. Optimize with different source partition strategies
4. This will show you bottlenecks in your transformation logic. With larger general purpose and mem optimized IRs, most of these operations occur in memory in data frames and are usually the fastest operations in your data flow

Best practices - Sources

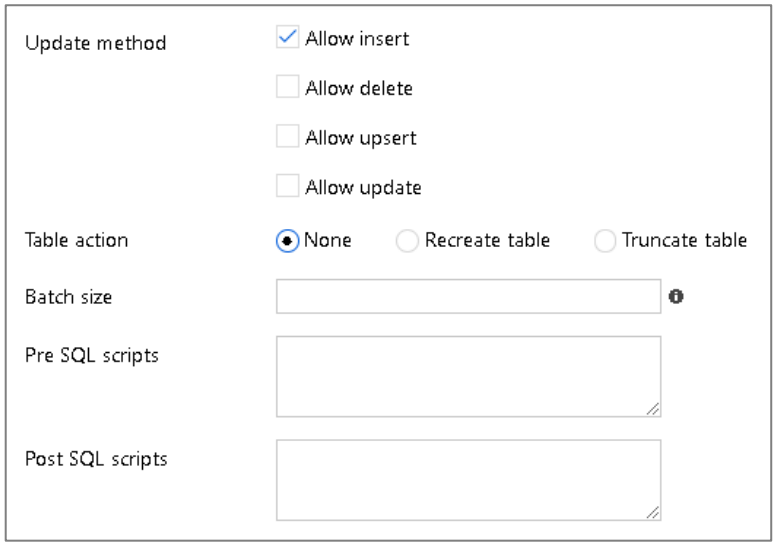
- When reading from file-based sources, data flow automatically partitions the data based on size
  - ~128 MB per partition, evenly distributed
  - Use current partitioning will be fastest for file-based and Synapse using PolyBase
    - Enable staging for Synapse
- For Azure SQL DB, use Source partitioning on column with high cardinality
  - Improves performance, but can saturate your source database
- Reading can be limited by the I/O of your source

Optimizing transformations

- Each transformation has its own optimize tab
  - Generally better to not alter -> reshuffling is a relatively slow process
- Reshuffling can be useful if data is very skewed
  - One node has a disproportionate amount of data
- For Joins, Exists and Lookups:
  - If you have a lot, memory optimized greatly increases performance
  - Can ‘Broadcast’ if the data on one side is small
    - Rule of thumb: Less than 50k rows
- Increasing integration runtime can speed up transformations
- Transformations that require reshuffling like Sort negatively impact performance

Best practices - Sinks

- SQL:
  - Disable indexes on target with pre/post SQL scripts
  - Increase SQL capacity during pipeline execution
  - Enable staging when using Synapse
- File-based sinks:
  - Use current partitioning allows Spark to create output
  - Output to single file is a very slow operation
    - Combines data into single partition
    - Often unnecessary by whoever is consuming data
  - Can set naming patterns or use data in column
  - Any reshuffling of data is slow
- Cosmos DB
  - Set throughput and batch size to meet performance requirements



Azure Integration Runtime

- Data Flows use JIT compute to minimize running expensive clusters when they are mostly idle
  - Generally more economical, but each cluster takes ~4 minutes to spin up
  - IR specifies what cluster type and core-count to use
    - Memory optimized is best, compute optimized doesn't generally work for production workloads
- When running Sequential jobs utilize *Time to Live* to reuse cluster between executions
  - Keeps cluster alive for TTL minutes after execution for new job to use
  - Maximum one job per cluster
- **Rule of thumb**: start small and scale up

Data flow script

Data flow script (DFS)

- DFS defines the logical intent of your data transformations
- Script is bundled and marshalled to Spark cluster as a job for execution
- DFS can be auto-generated and used for programmatic creation of data flows
- Access script behind UI via “Script” button

- Click “Copy as Single Line” to save version of script that is ready for JSON

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

Data flow script (DFS)

Movie Analytics: Split and aggregate genres , then unpivot column values to rows


```
source(output(
                movie as string,
                title as string,
                genres as string,
                year as string,
                Rating as integer,
                {Rotton Tomato} as string
        ),
        allowSchemaDrift: true,
        validateSchema: false) ~> source1
Aggregate1 unpivot(output(
                avg_genre as string,
                avgrating as double
        ),
        ungroupBy(year),
        lateral: false,
        ignoreNullPivots: true) ~> Unpivot1
source1 aggregate(groupBy(year),
        avg_comedy = round(avgIf(split(lower(genres),'|')[1]=='comedy',Rating),1,2),
                avg_drama = round(avgIf(split(lower(genres),'|')[1]=='drama',Rating),1,2),
                avg_action = round(avgIf(split(lower(genres),'|')[1]=='action',Rating),1,2)) ~> Aggregate1
Unpivot1 sort(desc(year, true)) ~> Sort1
Sort1 sink(allowSchemaDrift: true,
        validateSchema: false) ~> sink1
```

Source projection (1)

Source properties

Unpivot transformation (3)

Aggregate Transformation (2)

Sort (4)

Sink (5)

- Syntax: input_name transform_type(properties) ~> stream_name

Data flow script (DFS)

Eliminate duplicate rows and only keep distinct values

```
source(output(
                movie as integer,
                title as string,
                genres as string,
                year as string,
                Rating as string,
                {Rotton Tomato} as string
        ),
        allowSchemaDrift: true,
        validateSchema: false) ~> MoviesCSV
MoviesCSV aggregate(groupBy(movie),
        each(match(name!='movie'), $$ = first($$))) ~> DistinctRows
DistinctRows aggregate(rowcount_agg = count(1)) ~> RowCountDistinct
MoviesCSV select(mapColumn(
                movie,
                title,
                genres,
                year,
                Rating,
                {Rotton Tomato}
        ),
        skipDuplicateMapInputs: false,
        skipDuplicateMapOutputs: false) ~> OriginalData
OriginalData aggregate(rowcount_orig = count(1)) ~> RowCountOrig
DistinctRows sink(allowSchemaDrift: true,
        validateSchema: false,
        partitionBy('hash', 1)) ~> OutputDistinctData
```

Source projection (1)

Source properties

Distinct Aggregate (3)

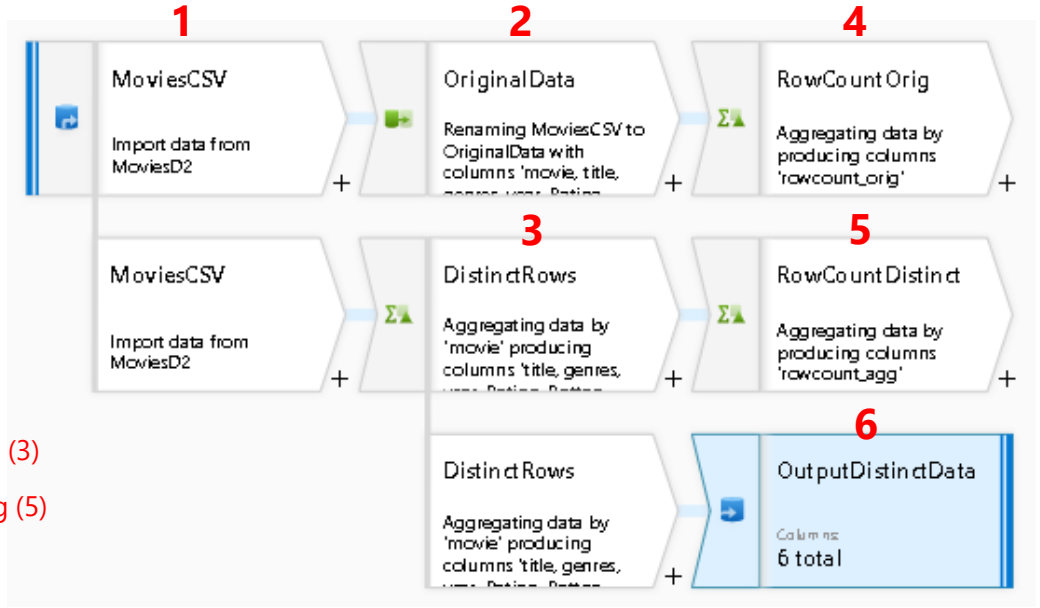
Row Count Agg (5)

Select transformation mappings (2)

Select properties (2)

Row Count Agg (4)

Sink transformation (6)



- ~> name_of_transform
- New branch does not require any script element

ETL Migrations

ETL Tool Migration Overview

- Migrating from an existing large enterprise ETL installation to ADF and data flows requires adherence to a formal methodology that incorporates classic SDLC, change management, project management, and a deep understanding of your current data estate and ETL requirements.

- Successful migration projects require project plans, executive sponsorship, budget, and a dedicated team to focus on rebuilding the ETL in ADF.

- For existing on-prem ETL estates, it is very important to learn basics of Cloud, Azure, and ADF generally before taking this Data Flows training.

Sponsorship

- Migration to another platform requires buy-in and sponsorship at the executive level and budget for this effort
- A team must be dedicated to this work
- That team can be internal or external, but a commitment to fund milestone-based project work is necessary

Discovery

- How many mappings are in the current ETL tool?
- Understand the high-level objectives, business rules, and logic of each mapping
- List out all required data source, sinks, and sample data for testing
- Capture current schedules and data SLAs
- Capture all required developer, user, and tester credentials as well as required credentials to access data sources and destinations
- Think about how you want to categorize the ETL mappings for migration
  - By complexity
  - By workflow
  - By business unit

Training

- On-prem to Cloud, Azure general training, ADF general training, Data Flows training
- A general understanding of the different between legacy client/server on-prem ETL architectures and cloud-based Big Data processing is required
- ADF and Data Flows execute on Spark, so learn the fundamentals of the different between row-by-row processing on a local server and batch/distributed computing on Spark in the Cloud

Execution

- Start with the top 10 mission-critical ETL mappings and list out the primary logical goals and steps achieved in each
- Use sample data and debug each scenario as new pipelines and data flows in ADF
- UAT each of those 10 mappings in ADF using sample data
- Lay out end-to-end project plan for remaining mapping migrations
- Plan the remainder of the project into quarterly calendar milestones
- Except each phase to take around 3 months
- Majority of large existing ETL infrastructure modernization migrations take 12-18 months to complete

ETL System Integrator Partners

- Bitwise Global
  - https://www.bitwiseglobal.com/webinars/automated-etl-conversion-to-adf-for-accelerated-data-warehouse-migration-on-azure/
- Next Pathway
  - https://blog.nextpathway.com/next-pathway-adds-ground-breaking-capability-to-translate-informatica-and-datastage-etl-pipelines-to-the-cloud-in-latest-version-of-shift

Microsoft

Azure HPC/Batch

HPC and the Cloud

HPC in the cloud: Market snapshot

47%

of HPC customers are testing workloads in the cloud

Drivers

- Faster time-to-solution
- Integration of AI
- Broad awareness

Challenges

- Complex workflows
- Software licensing
- Long sales cycle
- Perceived costs

New workloads

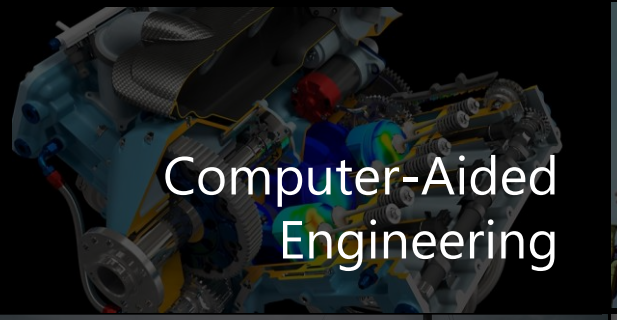
- Predictive analytics
- Autonomous driving
- Deep learning

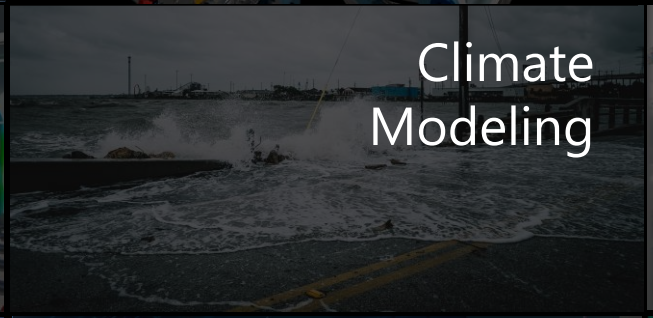
By 2022, global HPC market expected to grow to

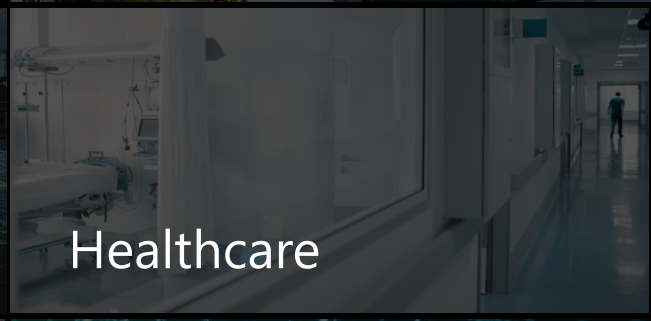
~$45B

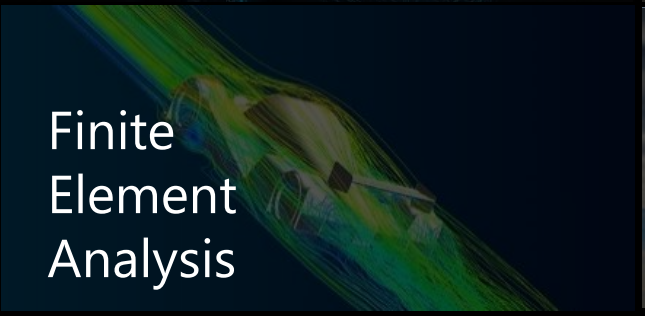
at a compound annual growth rate of 7%

Innovation without boundaries

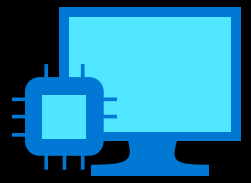








~~HPC in the Cloud~~ A cloud built for HPC



Purpose-built HPC

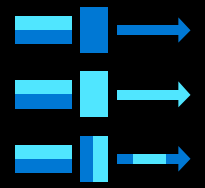
A full range of CPU and GPU capabilities that help applications scale to 80K+ cores

Fast, Secure Networking

Fast InfiniBand inter-connects as well as edge-to-cloud connectivity

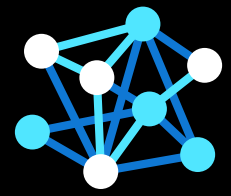
High Performing Storage

A range of storage capabilities to support simple-to-complex storage needs



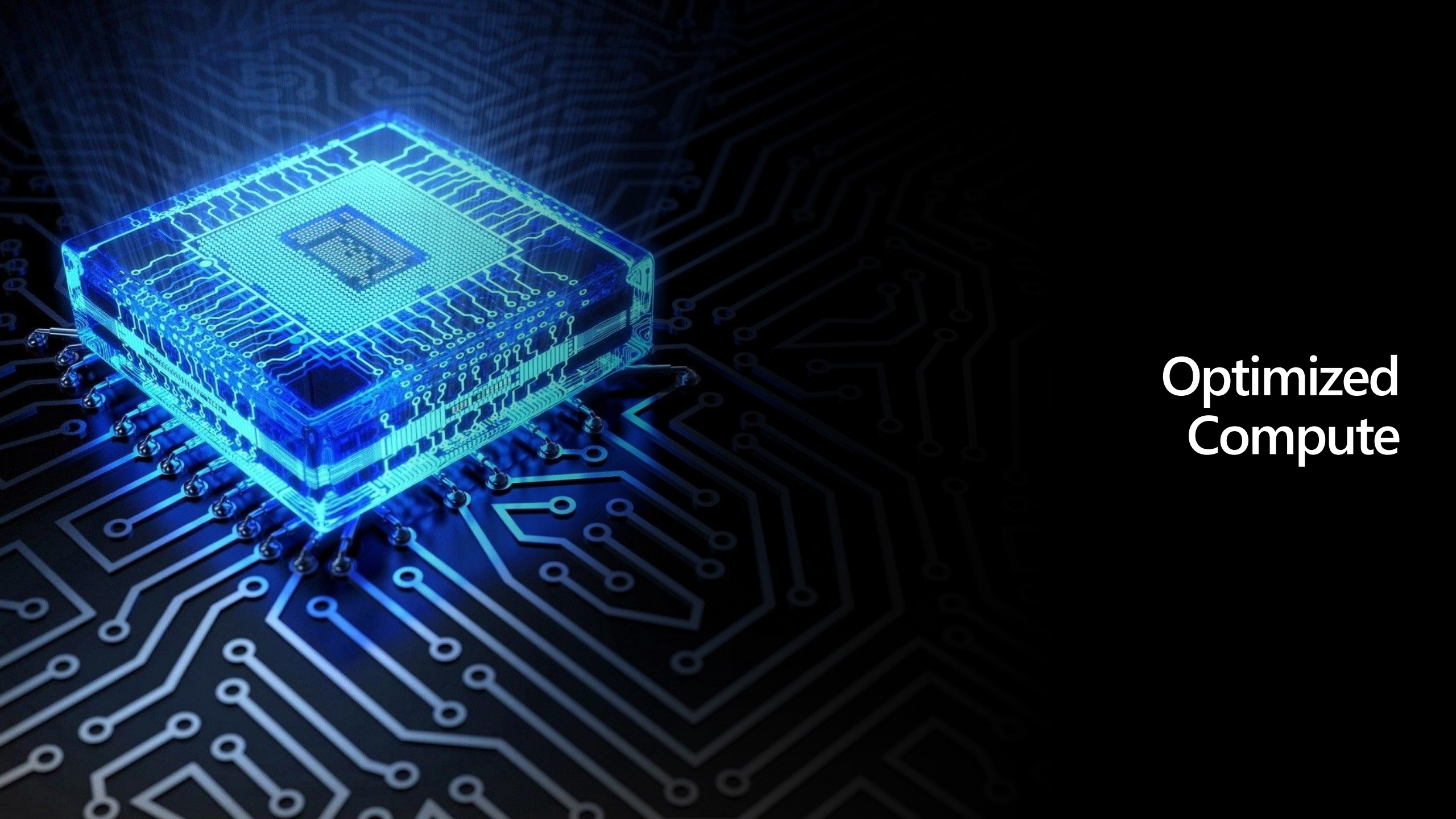
Workload Orchestration

End–to–end workflow agility using known, familiar tools & processes



Intelligence Services

AI, machine learning, and deep learning at supercomputer scale



Optimized Compute

HPC Resource Stack on Azure

Transformative Services

Azure Machine Learning | Azure Data Lake | Azure ML Compute

Workload Orchestration

VM Scale Sets | Azure Batch | Azure CycleCloud

Fast, Secure Networking

ExpressRoute | InfiniBand

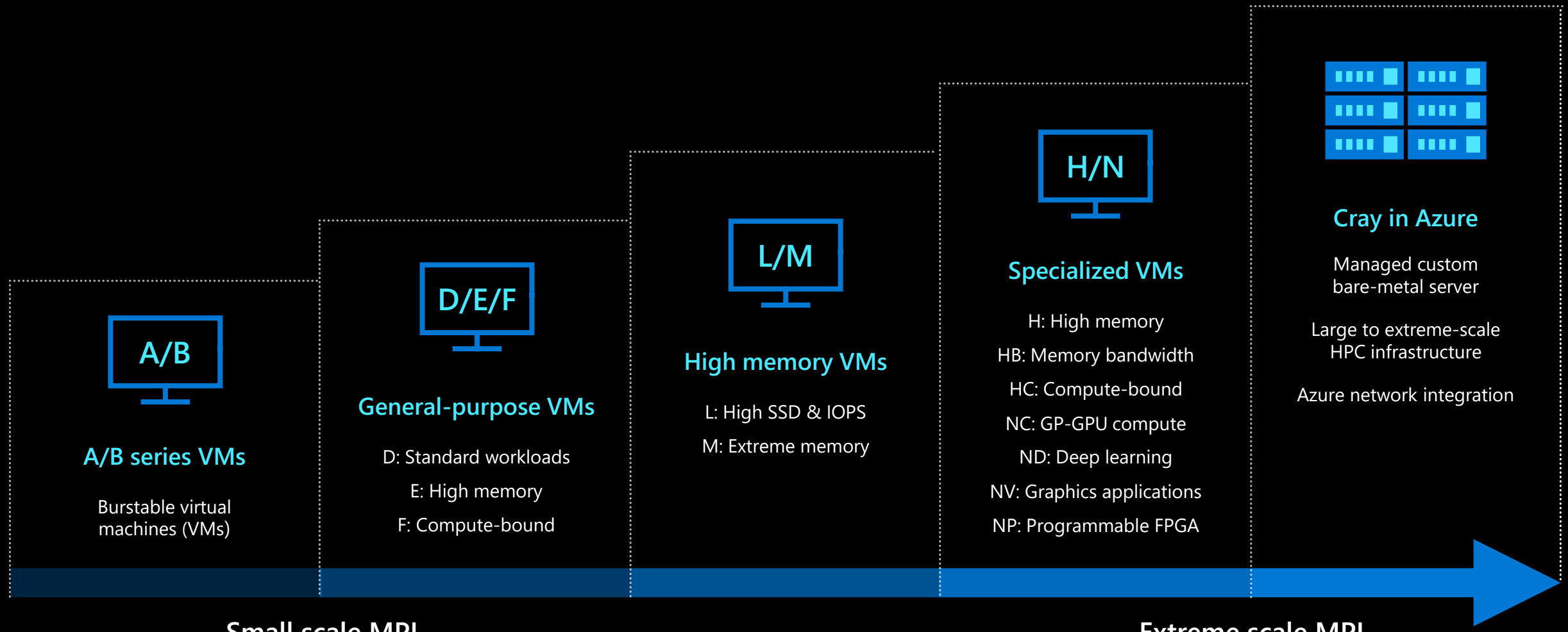
High Performing Storage

Azure HPC Cache | Azure NetApp Files | Clusterstore

Optimized Compute

H-Series | N-Series | Cray

Solve any HPC, AI workload—at any scale



High-Performance Computing VMs (H-Series)

	HBv3	HBv2	HB	HC	H
Workload Optimized	Memory Bandwidth	Memory Bandwidth	Memory Bandwidth	Dense Compute	Large-Memory HPC
CPU	AMD EPYC 3rd Gen w/ 3D V-Cache "Milan-X"	AMD EPYC 2nd Gen "Rome"	AMD EPYC 1st Gen "Naples"	Intel Xeon Platinum 1st Gen "Skylake"	Intel Xeon E5 v3 "Haswell"
Cores/VM	120	120	60	44	16
TeraFLOPS/VM (FP64)	4T	4 TF	.9 TF	2.6 TF	.7 TF
Memory Bandwidth	350 GB/sec	350 GB/s	263 GB/s	191 GB/s	80 GB/s
Memory	448 GB total	4 GB/core, 480 total	4 GB/core, 240 total	8 GB/core, 352 GB	14 GB/core, 224 GB
Local Disk	2 * 960 GB NVMe	960 GB NVMe	700 GB NVMe		2 TB SATA
InfiniBand	200 Gb HDR	200 Gb HDR	100 Gb EDR		56 Gb FDR
Network	50 GbE	50 GbE	50 GbE		16 GbE

**All cores, non-AVX, peak Boost/Turbo frequencies*

NVIDIA in Azure

Broad offering with the N-series

GPUs from M60 to V100 to support ML & DL training & inference, HPC across industries and Graphics workloads

Integration with key Microsoft solutions

like AML and ONNX for end-to-end acceleration for the most complex workloads, reducing burden of producing world-class solutions

Versatility across applications and frameworks

to support engineering on one platform and solution for the entire team, regardless of the preferences of particular engineers

All NVIDIA acceleration software housed in NGC

via Azure marketplace to ensure users have the easiest experience packaging what they need to achieve best performance in no time.

Optional InfiniBand interconnect

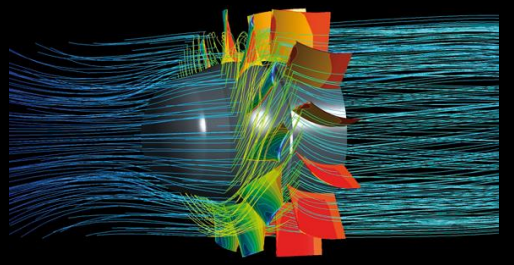
enables scale-up performance

GPU Computing

Visualization

Rendering

HPC/Simulation



Deep-Learning/AI

NV

Workload centric PCs & Workstations

Variable configuration for GPU workstations for content consumption

Modern workspace for interactive collaboration

NC

Scale out using IB for multimode HPC and ML workloads on any MPI stack

Scale-up multi GPU VMs with fast NVLINK interconnect for high-density single box training and HPC workloads

ND

GPUs for Deep Learning (ND Series)

	ND	NDv2	ND A100 v4	NDm A100 v4
CPU Cores	6,12,24	40	96	96
GPU	1x, 2x, or 4x P40 GPUs	8x V100 32 GB (NVLink) GPUs	8x A100 40 GB GPUs	8x A100 80 GB GPUs
Memory	12/224/448 GB	672 GB	900 GB	1900 GB
Local Disk	736/1474/2948 GiB SSD	2948 GiB SSD	6 TB SSD	6.4 TB SSD
Network	Azure Network + InfiniBand EDR	Azure Network + InfiniBand EDR + NVLink GPU Interconnect	Azure Network + InfiniBand EDR + NVLink GPU Interconnect	Azure Network + InfiniBand EDR + NVLink GPU Interconnect

deep learning

Workload Orchestration

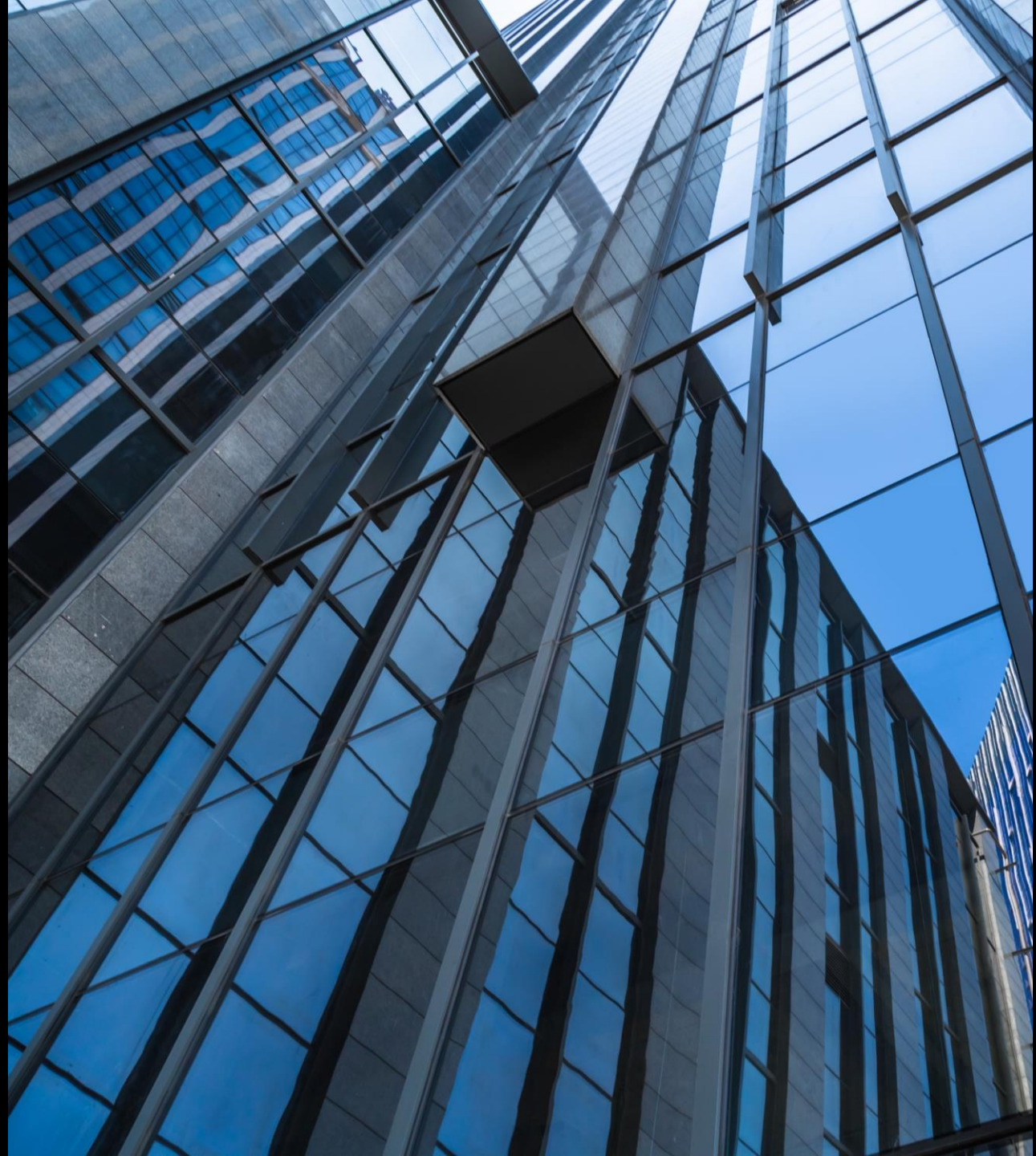
Azure Batch

Batch pools

- Configure and create VMs to cater for any scale: tens to thousands
- Automatically scale the number of VMs to maximize utilization
- Easy low-priority and VM sizing, suited to your application

Batch jobs and tasks

- Task is a unit of execution; task = application command line (EXE, BAT, CMD, PS1, etc.)
- Jobs are created and tasks are submitted to a pool. Next, tasks are queued and assigned to VMs
- Any application, any execution time; run applications unchanged
- Automatic detection and retry of frozen or failing tasks



Azure Batch Capabilities

Job scheduling

Supports both EP & MPI jobs

Run > 1 task in parallel per node

Detect and retry failed tasks

Task dependencies

Job prep and cleanup tasks

Rich app management

Get apps from blobs, Batch app packages, package managers, custom VM images

Docker container images

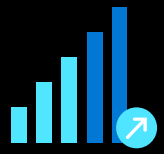
Choice of VMs

Windows or Linux

Standard or custom images

Windows pool can use AHUB

Use low-priority VMs



Monitoring

VM monitoring and auto-recover

Metrics and logs available via Portal and API

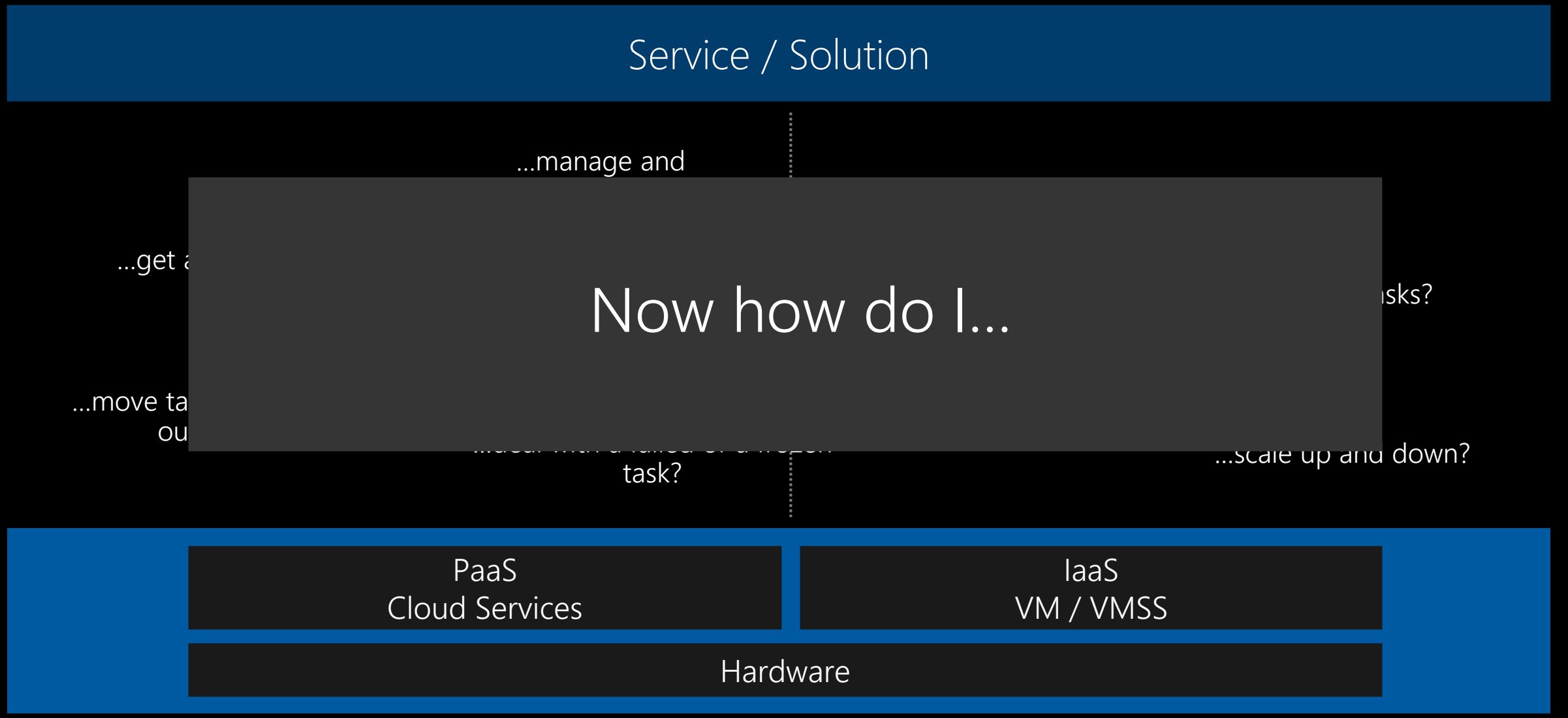
Developer access

.NET, Java, Node.js, Python, REST

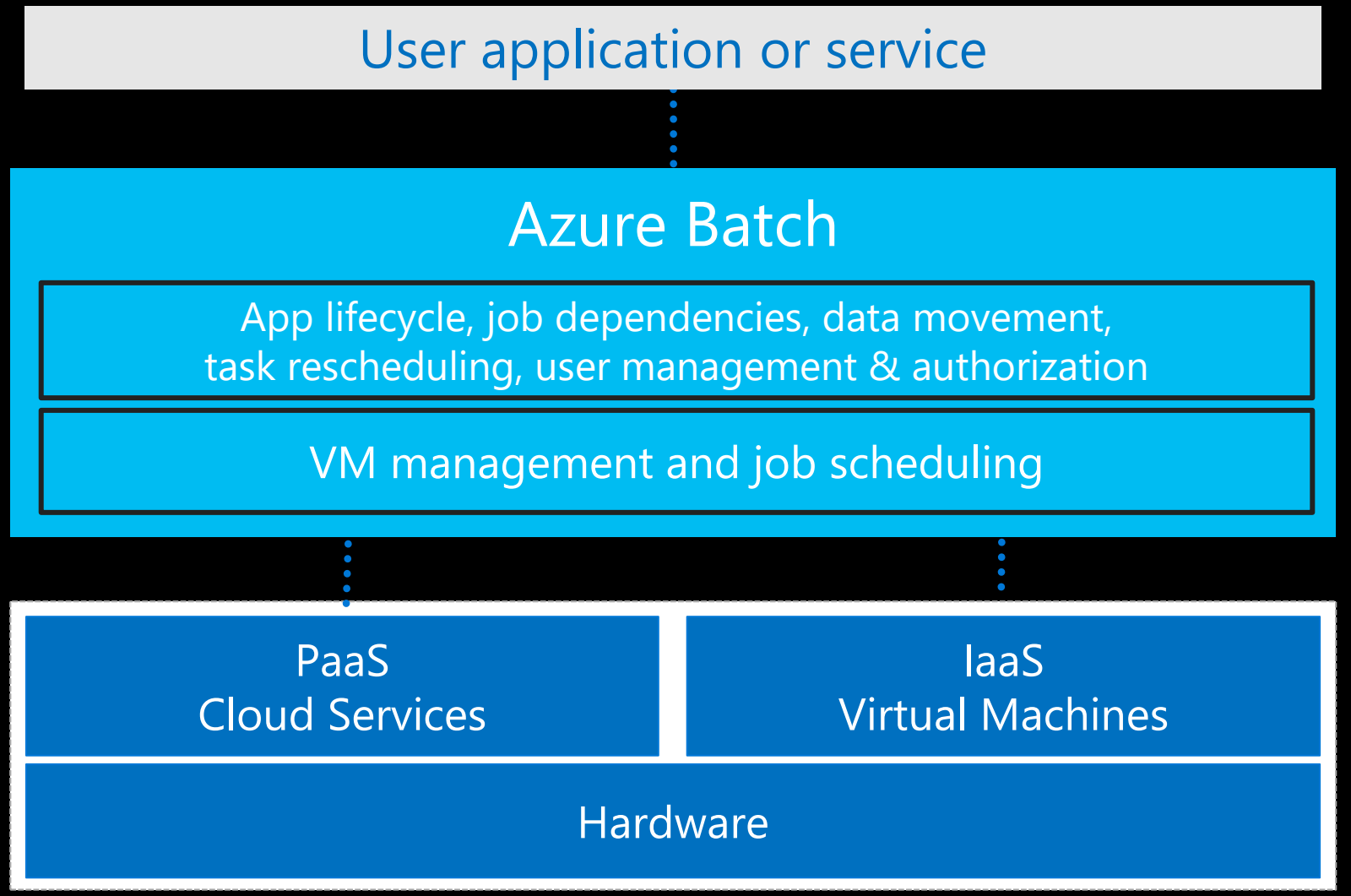
PowerShell, x-plat Azure CLI

Azure Portal, Batch Labs x-plat client UI

What is Azure Batch?



What is Azure Batch?



Provided by the cloud platform

- Don't worry about the "plumbing"
- Focus on the workload/app
- Access higher-level capabilities
- Minimize the required cloud or Azure experience

Azure Batch Capabilities

Job scheduling

- Supports both embarrassingly parallel and tightly-coupled MPI jobs
- Run > 1 task in parallel per node
- Detect and retry failed tasks
- Can set max execution time for jobs and tasks
- Task dependencies
- Job prep and cleanup tasks

Rich app management

- Get apps from blobs, Batch app packages, package managers, custom VM images
- Docker container images

Choice of VMs

- Windows or Linux
- Standard or custom images
- Windows pool can use AHUB
- Use low-priority VMs

Monitoring

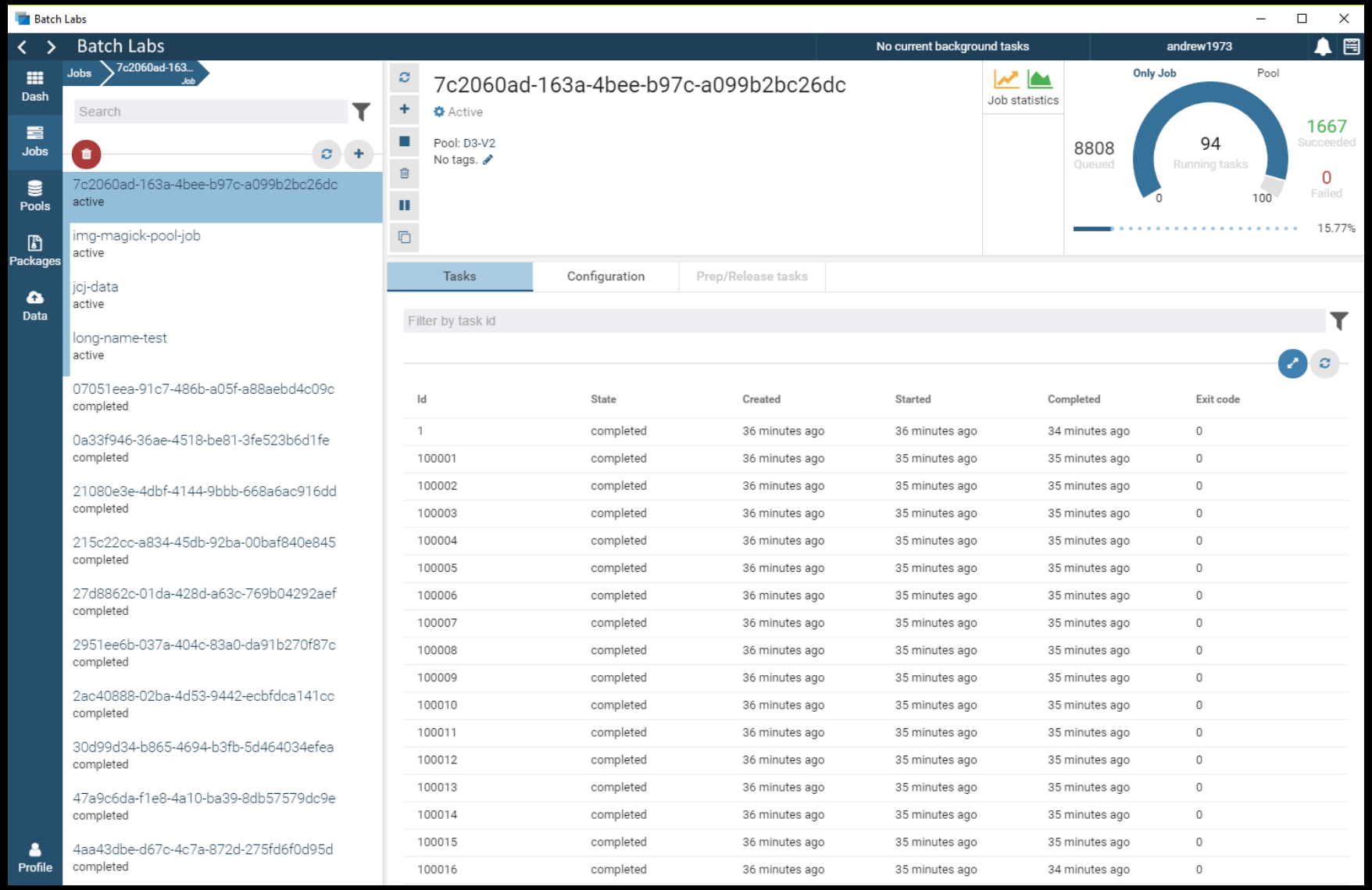
- VM monitoring and auto-recover
- Metrics and logs available via Portal and API

Access via APIs, CLIs, and UIs

- .NET, Java, Node.js, Python, REST
- PowerShell, Azure CLI
- Azure Portal, Batch Explorer client UI

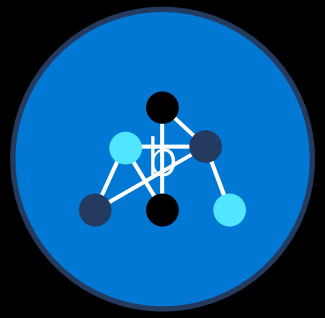
Batch Explorer

https://github.com/Azure/BatchExplorer



Incorporating Intelligence

Azure AI



Machine learning

AI apps & agents

Knowledge mining

Machine Learning on Azure

Domain-specific pretrained models

Vision | Speech | Language | Search

Familiar data science tools

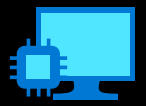
Visual Studio Code | Azure Notebooks | Jupyter | Command Line

Popular frameworks

PyTorch | TensorFlow | Scikit-Learn | ONNX

Productive services

Azure Data Bricks | Azure Machine Learning | ML VMs



Powerful infrastructure

CPU | GPU | FPGA

Azure Machine Learning Compute

Addresses the challenges of training Machine Learning models

Challenges with training models

Managing dependencies

Securing access

Scheduling jobs

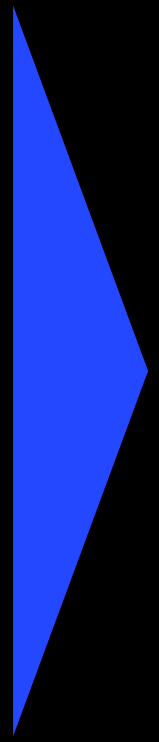
Feeding data

Provisioning VMs

Scaling resources

Handling failures

Gathering results



Solutions with Azure ML Compute

Integrates Batch AI to provide managed compute clusters and job scheduling

Work with clusters of GPU or CPU to run experiments in parallel and at scale to reduce training time

Enables data scientists to easily train, test, and score Deep Learning and other AI/ML models without managing infrastructure

Use with Azure Machine Learning service and integrate with your data science tools and pipeline

Microsoft

Azure Synapse - Spark

Data Engineering with Spark

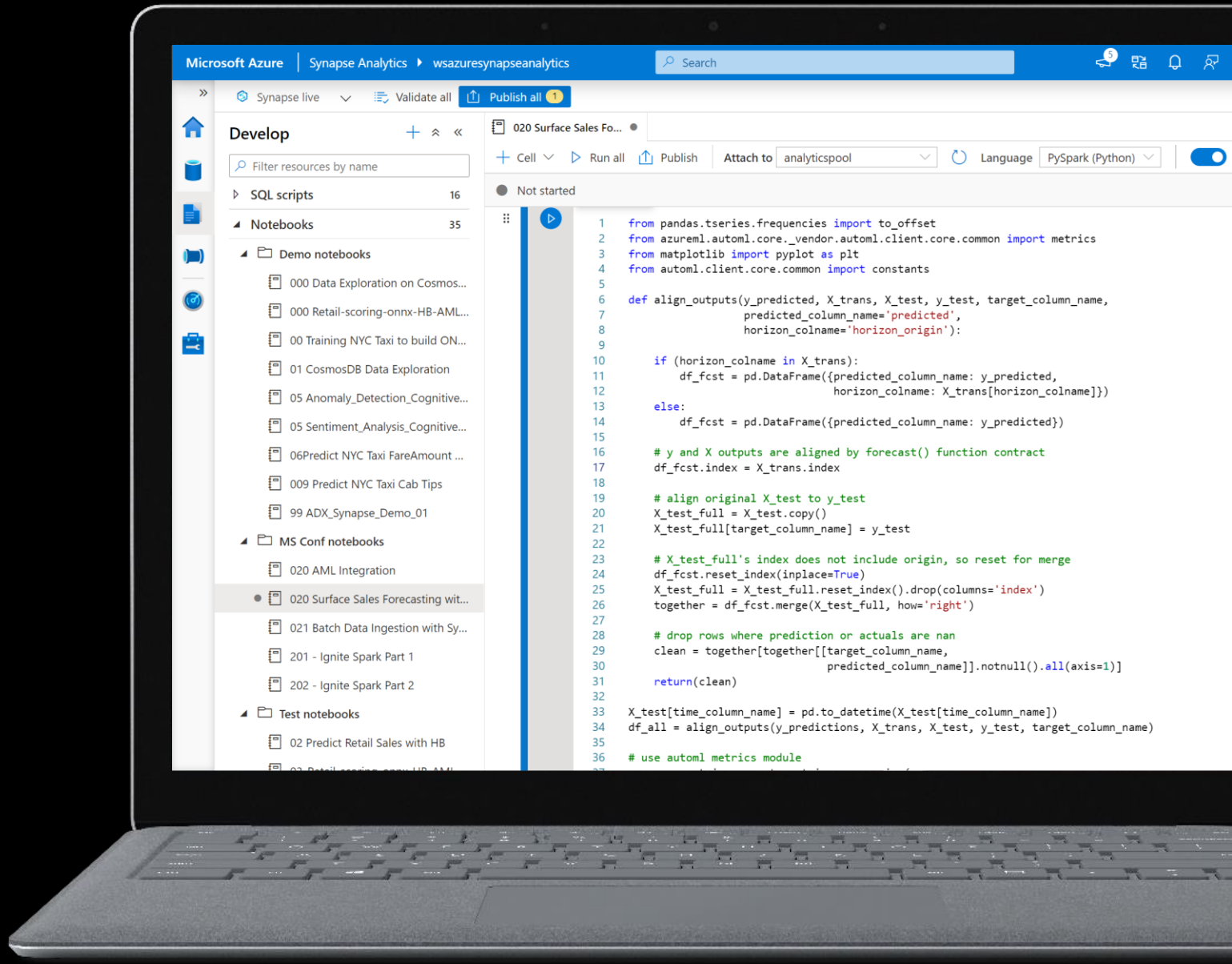
Code-first Data Engineering

PySpark, Scala, SQL and C# languages supported

Author multiple languages in a single notebook

Analyze & transform data from the data warehouse, data lake, and real-time operational data from one place

Synapse Spark uses Spark 3.2 runtime, which includes Delta Lake 1.0



Spark 3.3

Enables developers can leverage the latest innovations in the Spark ecosystem

Pandas (Koalas) integration

A highly popular and flexible library with broad industry adoption

Adaptive Query Execution (AQE) enabled by default

Significant improvements in query performance out-of-the-box

Small Query execution improvements

Small queries run faster due to reduced initialization overhead

[Azure Synapse Runtime for Apache Spark 3.3 - Azure Synapse Analytics | Microsoft Learn]

What is Apache Spark?

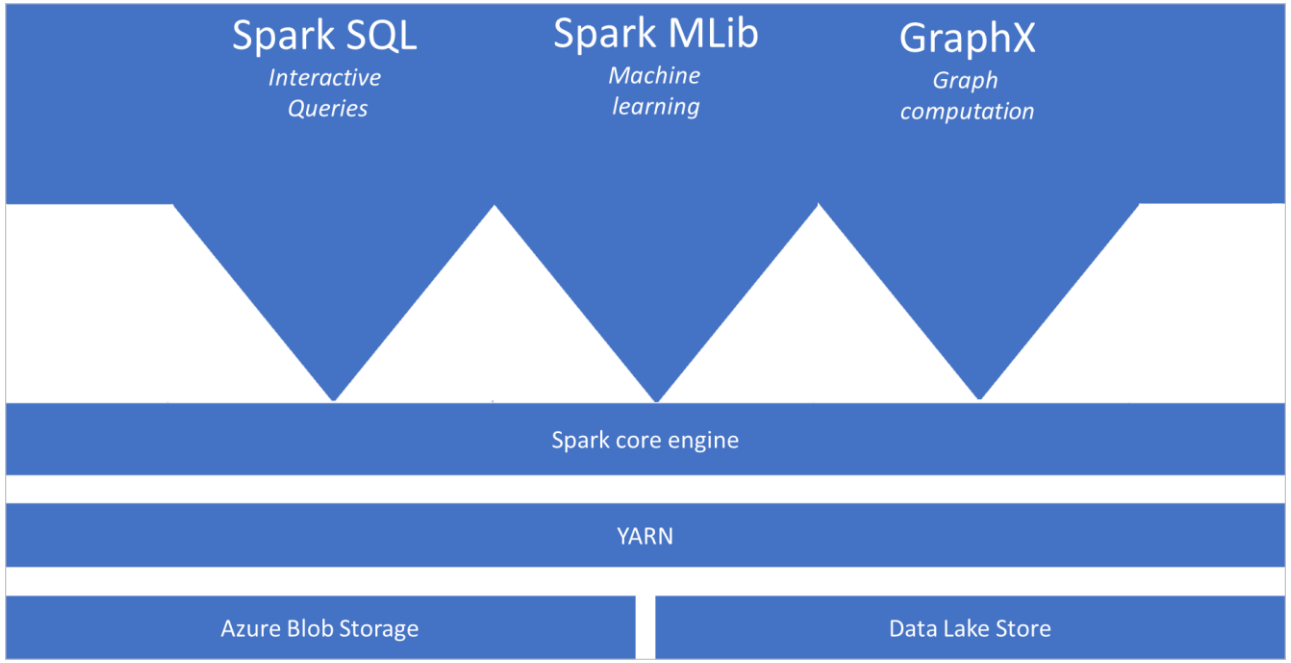
Parallel processing framework

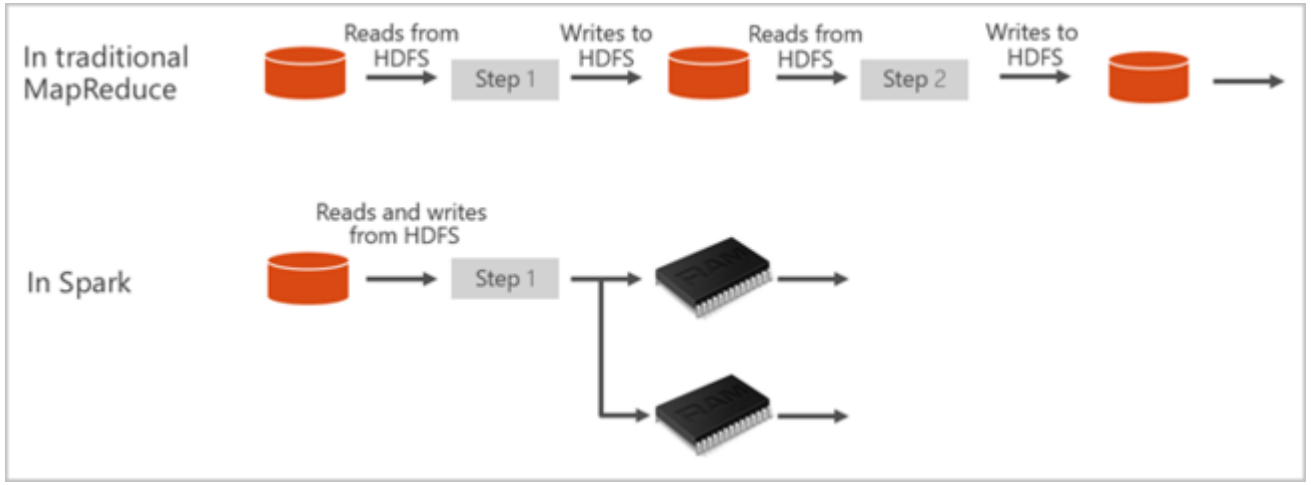
In-memory processing engine to boost the performance of big-data analytic applications

Much faster than disk-based applications

Integrates with multiple programming languages

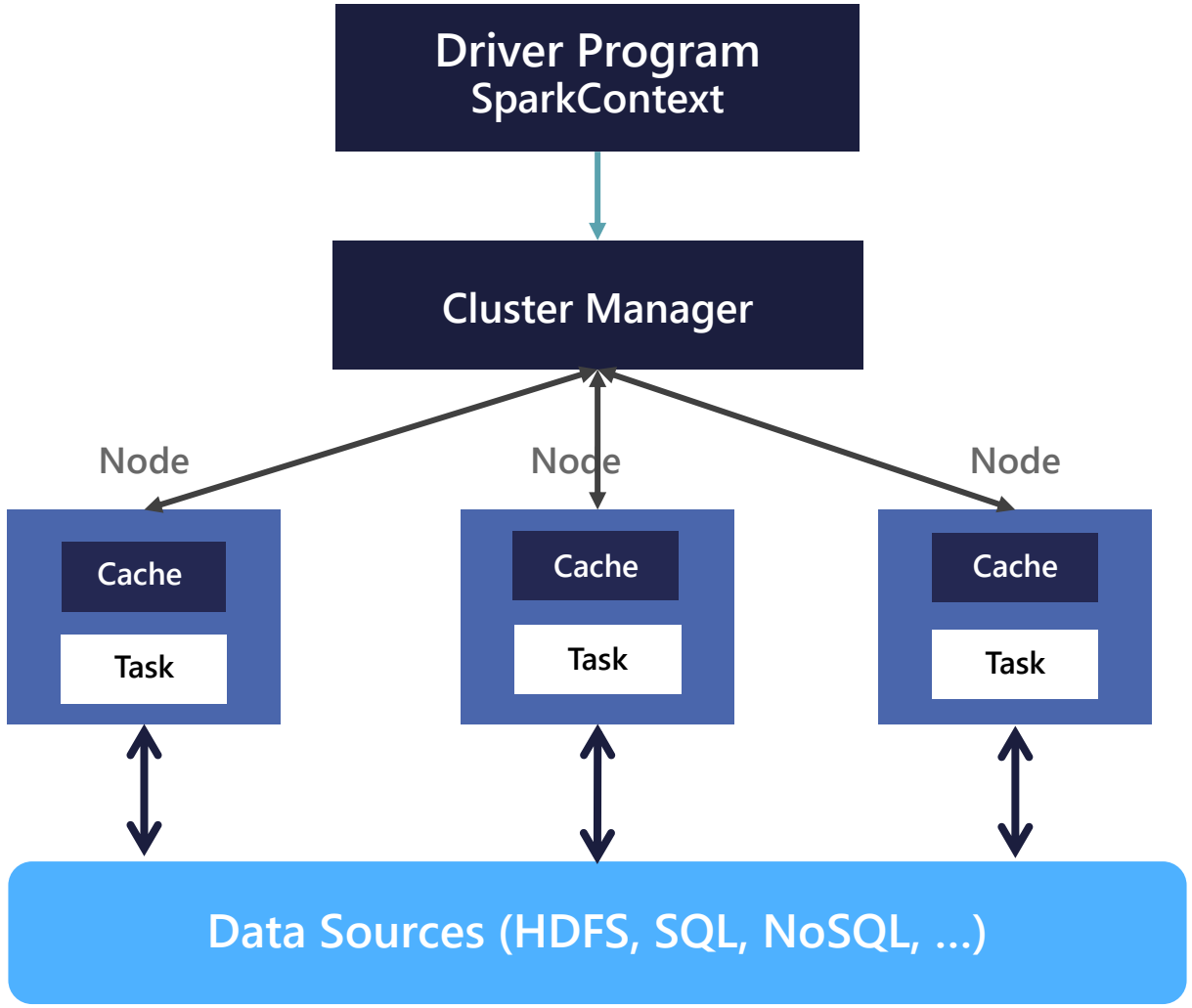
Supports many workloads: Data Engineering, SQL, ML, Graphs., etc.





General Spark Cluster Architecture

- 'Driver' runs the user's 'main' function and executes the various parallel operations on the worker nodes.
- The results of the operations are collected by the driver
- The worker nodes read and write data from/to Data Sources including HDFS.
- Worker node also cache transformed data in memory as RDDs (Resilient Data Sets).
- Worker nodes and the Driver Node execute as VMs in public clouds (like Azure).
- These resources come in the form of *executors* on the worker nodes (logical blocks of CPU/memory). These executors are what read and write data for your application.
- The *SparkContext* breaks down the application code into *tasks* that are operated on in parallel by the executors.



Motivation for Spark Pool in Synapse

Speed and efficiency	• Spark instances start in approximately 2 minutes for fewer than 60 nodes and approximately 5 minutes for more than 60 nodes. • The instance shuts down, by default, 5 minutes after the last job executed unless it is kept alive by a notebook connection.
Ease of creation	Quickly create a new Spark pool in Azure Synapse using: • Azure portal • Azure PowerShell • Synapse Analytics .NET SDK.
Ease of use	Synapse Analytics includes a custom notebook derived from Nteract. You can use these notebooks for interactive data processing and visualization.

Motivation for Spark Pool in Synapse

REST APIs	Spark in Azure Synapse Analytics includes Apache Livy, a REST API-based Spark job server to remotely submit and monitor jobs.
Support for Azure Data Lake Storage Generation 2	Spark pools in Azure Synapse can use: • Azure Data Lake Storage Generation 2 • BLOB storage.
Integration with third-party IDEs	Azure Synapse provides an IDE plugin for JetBrains' IntelliJ IDEA that is useful to create and submit applications to a Spark pool.

Motivation for Spark Pool in Synapse

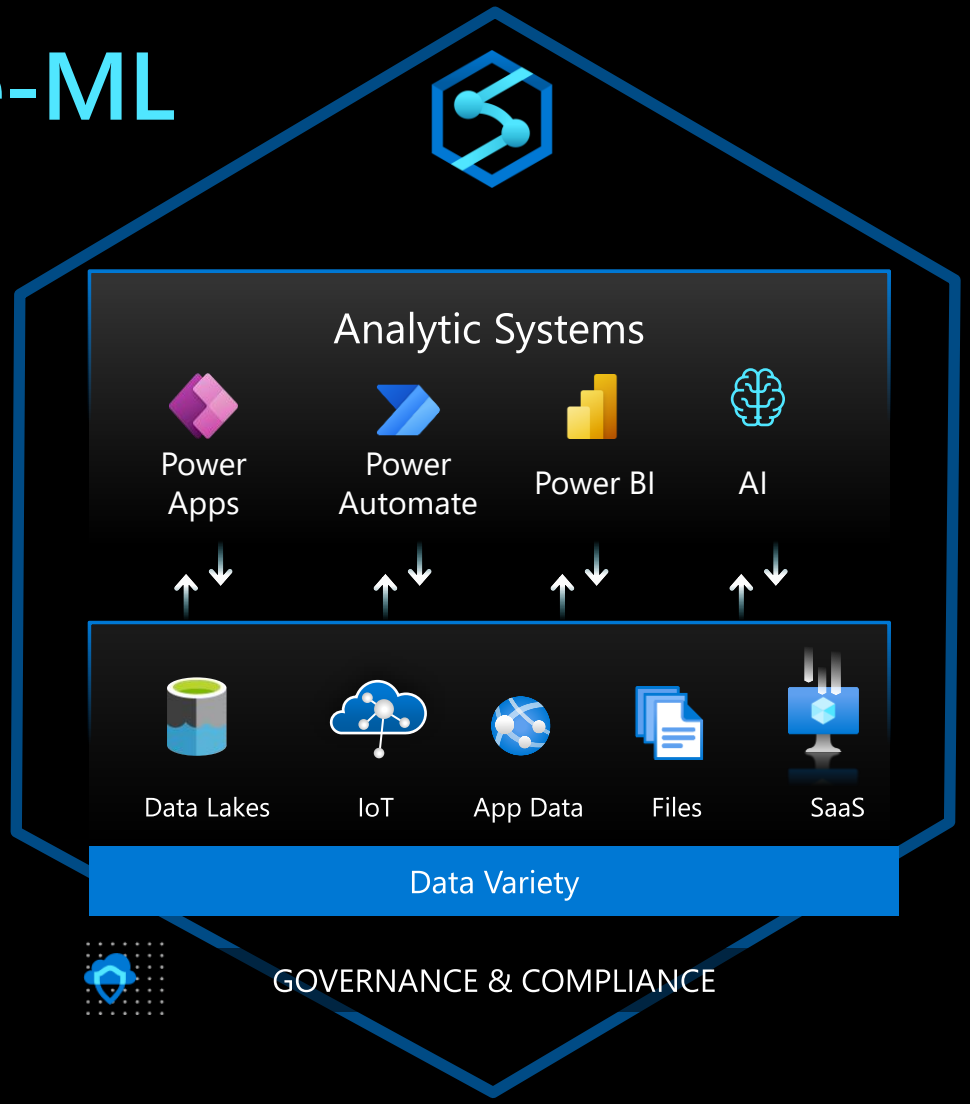
Pre-loaded Anaconda libraries	• Spark pools in Azure Synapse come with Anaconda libraries pre-installed. • Providing close to 200 libraries for machine learning, data analysis, visualization, etc.
Scalability	• Can have Auto-Scale enabled, so that pools scale by adding or removing nodes as needed. • Spark pools can be shut down with no loss of data since all the data is stored in Azure Storage or Data Lake Storage.

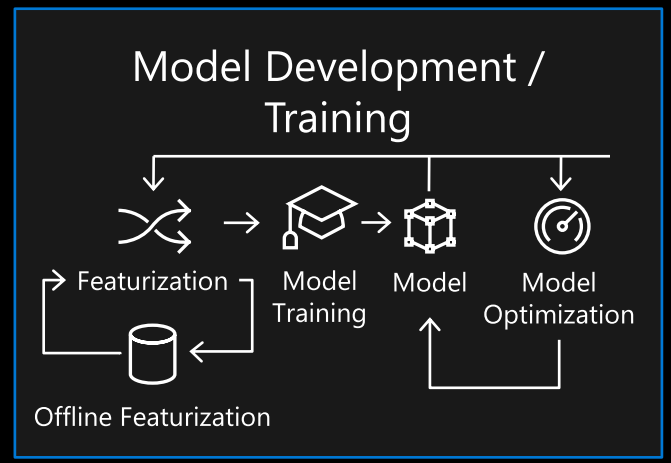
Modeling capabilities

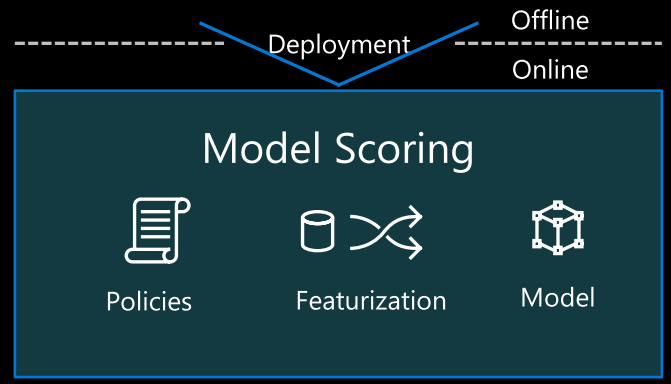
Why Synapse
Enterprise Grade-ML

Productizing AI

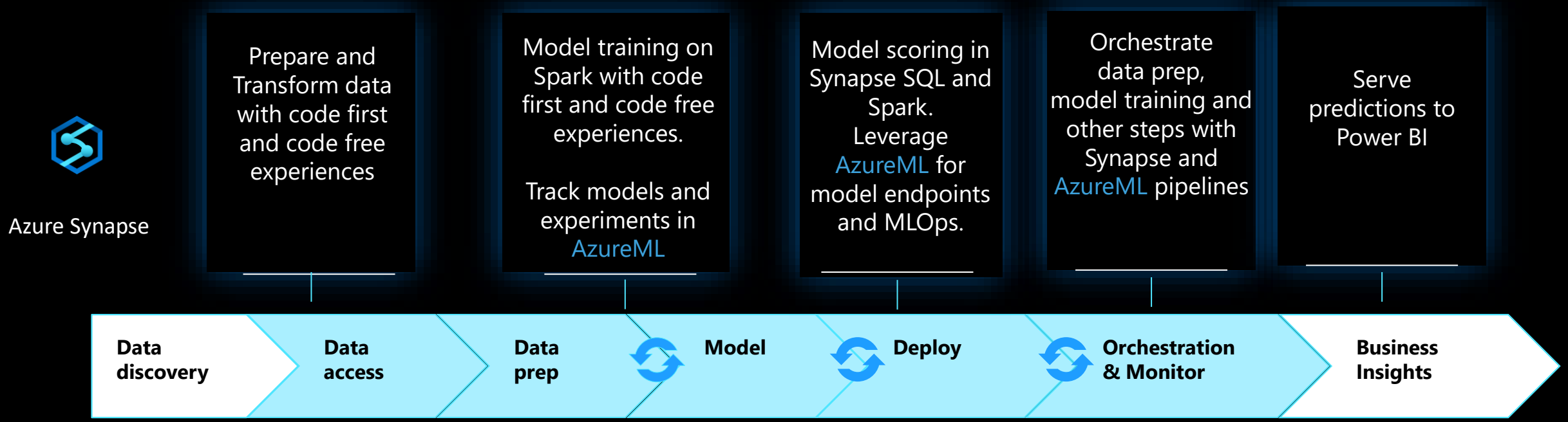
- Train in the cloud within the Hub
- Scoring with operational systems
- Governance everywhere (models, lineage)
- Ethical AI
- Control over deployment
- Deployment across Apps, BI, Processes
- Exchange of models (ONNX)
- Enabling Reinforcement learning







Synapse & Azure ML: Supporting the full Data & AI lifecycle



Data wrangling with Apache Spark pools (preview) - Azure Machine Learning | Microsoft Learn

MachineLearningNotebooks/how-to-use-azureml/azure-synapse at master · Azure/MachineLearningNotebooks (github.com)

Synapse Apache Spark in Azure Machine Learning

Link workspace and attach compute

Link and attach compute via ARM, SDK and UI

Authentication via Identity

Permission control based on Role-Based Access Control (RBAC)

Leverage Spark magic for data prep

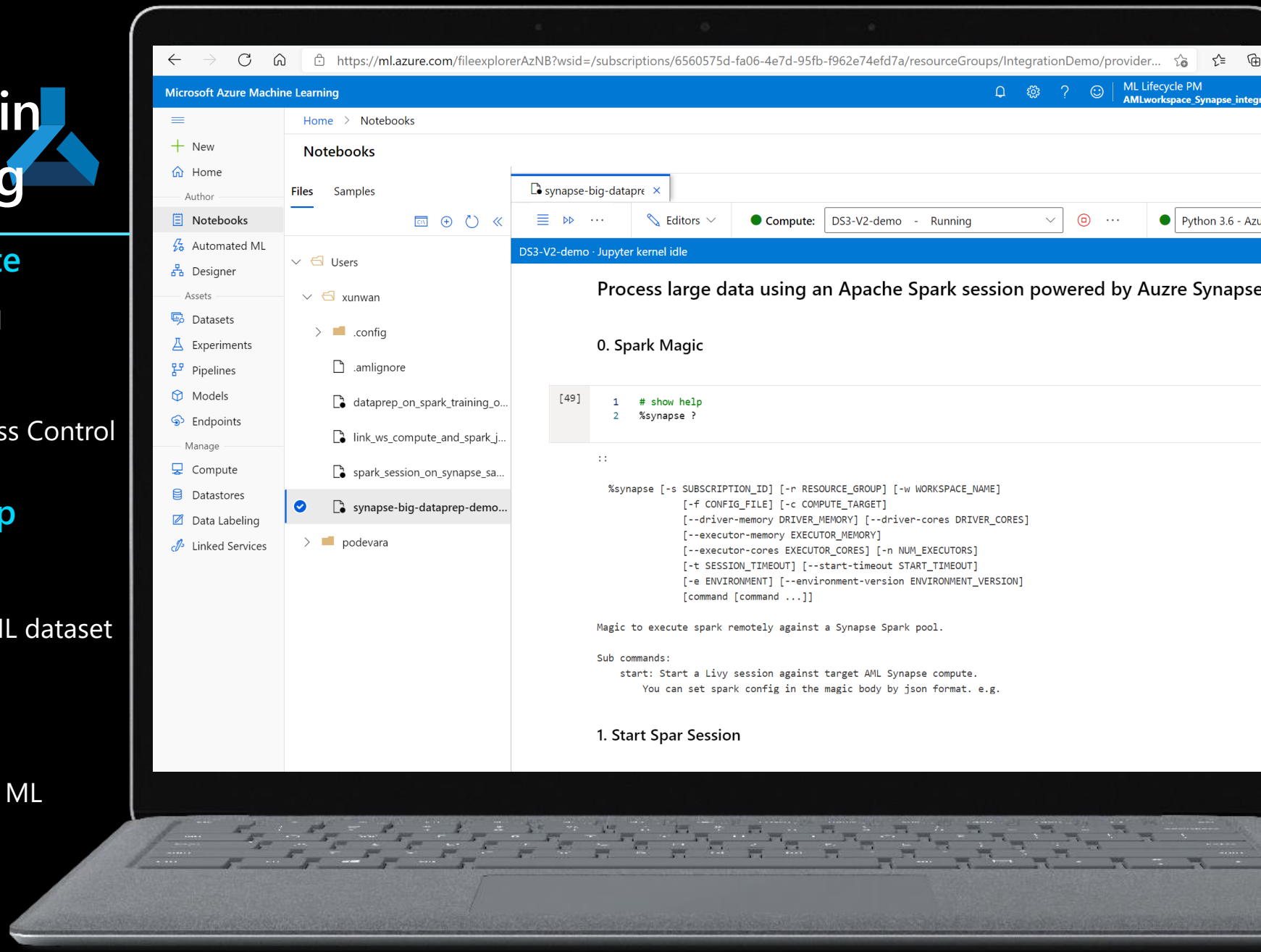
Run spark magic on cell to do data prep

Data access via HDFS data loading or Azure ML dataset

Pick up the data via dataset for training

Orchestration

Orchestrate data prep and ML steps via Azure ML pipeline



Spark Pool - Configuration

- **Nodes**
  - Apache Spark pool instance consists of one head node and two or more worker nodes with a minimum of three nodes in a Spark instance.
  - The head node runs additional management services such as Livy, Yarn Resource Manager, Zookeeper, and the Spark driver.
  - All nodes run services such as Node Agent and Yarn Node Manager.
  - All worker nodes run the Spark Executor service.

Spark Pool - Configuration

- **Node Sizes**
  - Node sizes can be altered after pool creation, instance may need to be restarted.

Size	vCore	Memory
Small	4	32 GB
Medium	8	64 GB
Large	16	128 GB
XLarge	32	256 GB
XXLarge	64	432 GB
XXX Large (Isolated Compute)	80	504 GB

ML Platform with Azure Machine Learning

Machine Learning Platform Architecture

Azure ML

Open Datasets

Structured Data

Unstructured Data

All other ML scenarios; NLP, Vision, IoT, etc.

Seamless studio experience

Notebooks | Automated ML | Designer

Responsible ML tools

Comprehensive MLOps

Reproducibility | Automation | Deployment | Re-training

Security and Governance

Unified management across clouds and on-premises

Azure

Serverless Compute | Managed Kubernetes

Edge & Hybrid

Azure Arc-enabled Kubernetes | Edge/IoT Devices

Business Apps

Models: AKS

Batch predictions: SQL DB

Analytics

Data Bricks | Synapse | Power BI

Prepare Data | Build & Train | Deploy | Manage & Monitor

ML Lifecycle workflows

Re-training

Data Preparation

Data Engineer
Data Modeler

Build & Train

Data Scientist
Citizen Data Scientist
ML Engineer

Deploy

ML Engineer
DevOps Engineer
Citizen Data Scientist

Manage & Monitor

ML Engineer
DevOps Engineer

Design solution architecture

IT Pro Security Engineer AI Architect

Compute Options for Experiment Runs

Local Compute

- Compute where the control code for the experiment is running
- Often a development workstation or Azure Machine Learning compute instance

Compute Cluster

- Cloud-based cluster managed in an Azure Machine Learning workspace
- Starts, stops, and scales on-demand

Attached Compute

- Azure compute resource outside of a workspace
- For example:
  - Virtual Machine
  - Azure Databricks
  - Azure HDInsight

What are Compute Targets in Azure ML?

A *compute target* is a designated compute resource or environment where you run your training script or host your service deployment.

In a typical model development lifecycle, you might:

1. Start by developing and experimenting on a small amount of data. At this stage, use your local environment, such as a local computer or cloud-based virtual machine (VM), as your compute target.
2. Scale up to larger data, or do distributed training by using one of these training compute targets.
3. After your model is ready, deploy it to a web hosting environment with one of these deployment compute targets.

Training Compute Targets

Training targets	Automated machine learning	Machine learning pipelines	Azure Machine Learning designer
Local computer	Yes		
Azure Machine Learning compute cluster	Yes	Yes	Yes
Azure Machine Learning compute instance	Yes (through SDK)	Yes	Yes
Azure Machine Learning Kubernetes	Yes	Yes	Yes
Remote VM	Yes	Yes	
Apache Spark pools (preview)	Yes (SDK local mode only)	Yes	
Azure Databricks	Yes (SDK local mode only)	Yes	
Azure Data Lake Analytics		Yes	
Azure HDInsight		Yes	
Azure Batch		Yes	

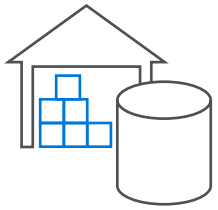
Inference Compute Targets

When performing inference, Azure Machine Learning creates a Docker container that hosts the model and associated resources needed to use it. This container is then used in a compute target

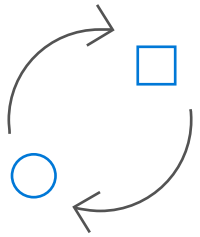
Compute target	Used for	GPU support	Description
Local web service	Testing/debugging		Use for limited testing and troubleshooting. Hardware acceleration depends on use of libraries in the local system.
Azure Machine Learning endpoints	Real-time inference Batch inference	Yes	Fully managed computes for real-time (managed online endpoints) and batch scoring (batch endpoints) on serverless compute.
Azure Machine Learning Kubernetes	Real-time inference Batch inference	Yes	Run inferencing workloads on on-premises, cloud, and edge Kubernetes clusters.
Azure Container Instances (SDK/CLI v1 only)	Real-time inference Recommended for dev/test purposes only.		Use for low-scale CPU-based workloads that require less than 48 GB of RAM. Doesn't require you to manage a cluster. Supported in the designer.

AZURE DATA FACTORY

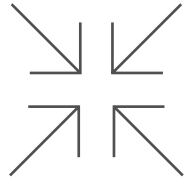
Fully-managed data integration service in the cloud



Flexible
Data integration

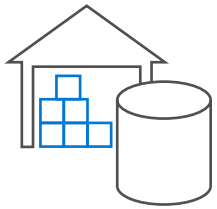


Hybrid
Data orchestration

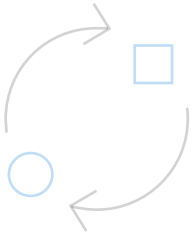


Data movement
As-a-service

Security and compliance



Flexible
Data integration



Hybrid
Data orchestration

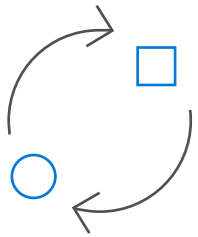
Data movement
As-a-service

Modernize your data warehouse with Azure big data and advanced analytics services such as HDInsight and Data lake Analytics

Build custom data-driven SaaS applications unique to your customer data using your language of choice

Bring together all your sources of data to understand your customers and drive impactful business decisions

Flexible
Data integration



Hybrid
Data orchestration

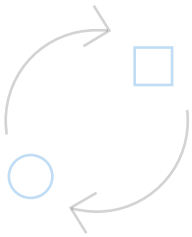
Data movement
As-a-service

Orchestrate your data pipeline wherever your data lives – in cloud or in self-hosted environment

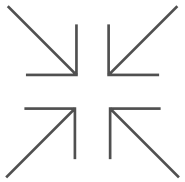
Meet your security and compliance needs while taking advantage of truly hybrid integration capabilities

Execute your SQL Server Integration Services (SSIS) packages in the cloud

Flexible
Data integration



Hybrid
Data orchestration



Data movement
As-a-service

Accelerate integration with managed data movement as-a-service

Improve your TCO with 30+ natively supported connectors across 18 global points of presence

Elastic data movement at scale

Serverless data movement with no infrastructure to manage

HYBRID DATA INTEGRATION AT SCALE

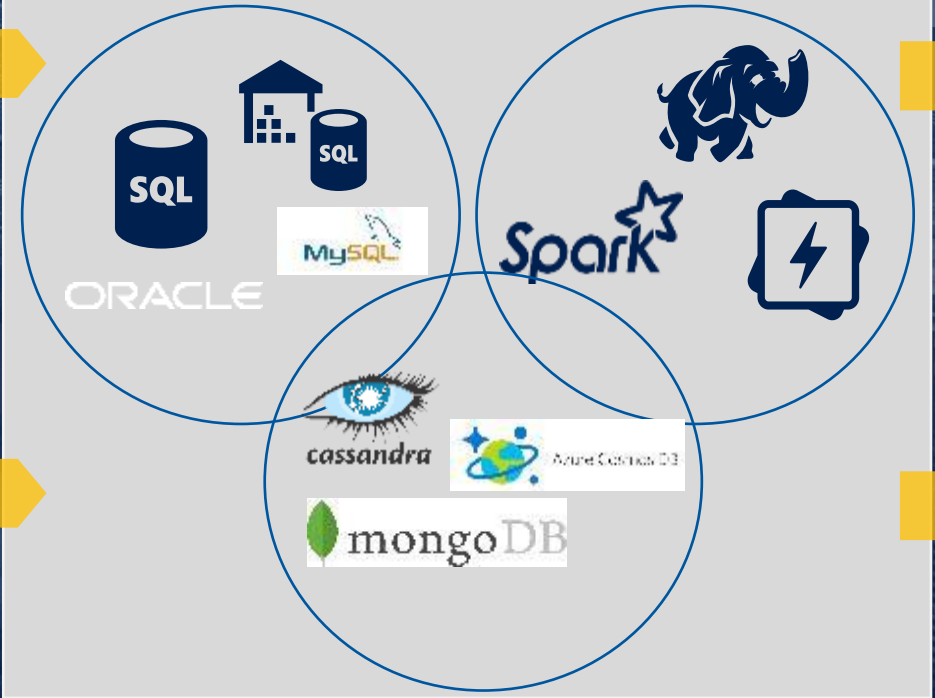
Relational data

OLTP | ERP | CRM | LOB

Non-relational data

Web | Media | Social media | Devices

Data Processing & Movement



Any BI tool

Dashboards | Reporting
Mobile BI | Cubes

Advanced Analytics

Machine Learning
Stream analytics Cognitive | AI

Any language

.NET | Java | R | Python
Ruby | PHP | Scala

CLOUD

V-NET

ON-PREMISE

AZURE DATA FACTORY ORCHESTRATES DATA PIPELINE ACTIVITY WORKFLOW & SCHEDULING

ADF: Cloud-First Data Integration Objectives

- Consume hybrid disparate data
  - On-prem + Cloud
  - Grow ADF ecosystem of structured, un-structured, semi-structured data connectors
- Calculate and format data for analytics
  - Transform, aggregate, join, normalize
  - Separate data flow (transformation) from control flow (orchestration)
- Address large-scale Big Data requirements
  - Scale-up or Scale-out data movement and transformation
  - Support multiple processing engines
- Operationalize
  - Support flexible scheduling and triggering mechanism for broad range of use cases
  - Manage & monitor multiple pipelines (via Azure Monitor & OMS)
  - Support secure VNET environments
- Enable SSIS package execution
  - Execute SSIS packages in ADF Integration Runtime

ADF: Cloud-First Data Integration Scenarios

Lift and Shift to the Cloud
- Migrate on-prem DW to Azure
- Lift and shift existing on-prem SSIS packages to cloud
- No changes needed to migrate SSIS packages to Cloud service

DW Modernization
- Modernizing DW arch to reduce cost & scale to needs of big data (volume, variety, etc)
- Flexible wall-clock and triggered event scheduling
- Incremental Data Load

Build Data-Driven, Intelligent SaaS Application
- C#, Python, PowerShell, ARM support

Big Data Analytics
- Customer profiling, Product recommendations, Sentiment Analysis, Churn Analysis, Customized offers, customer usage tracking, customized marketing
- On-demand Spark cluster support

Load your Data Lake
- Separate control-flow to orchestrate complex patterns with branching, looping, conditional processing

ADF V2 Improvements

- Integration Runtimes (IR) replace DMG, provide data movement and activity dispatch on-prem or in the cloud
- Supports resources within virtual networks
- Integration Runtime includes SSIS option to lift & shift SSIS packages to the Cloud
- Separation of “control flow” & “data flow” capabilities for more flexible pipeline management
  - Looping, conditionals, dependencies, parameters
- Python SDK
- On-Demand Spark support
- Flexible pipeline scheduling with wall-clock and triggered executions
- Expanded use cases: From primarily time window-oriented pipelines, to trigger-based on-demand for more flexible ETL and data integration orchestrations
- (Coming in GA) UX pipeline and data transformation builder code-free experience

New ADF V2 Concepts

Concept	Description	Sample
Control Flow	Orchestration of pipeline activities that includes chaining activities in a sequence, branching, conditional branching based on an expression, parameters that can be defined at the pipeline level and arguments passed while invoking the pipeline on demand or from a trigger. Also includes custom state passing and looping containers, I.e. For-each, Do-Until iterators.	{ "name":"MyForEachActivityName", "type":"ForEach", "typeProperties":{ "isSequential":"true", "items":"@pipeline().parameters.mySinkDatasetFolderPathCollection", "activities":[{ "name":"MyCopyActivity", "type":"Copy", "typeProperties": …
Runs	A Run is an instance of the pipeline execution. Pipeline Runs are typically instantiated by passing the arguments to the parameters defined in the Pipelines. The arguments can be passed manually or properties created by the Triggers.	POST https://management.azure.com/subscriptions/<subId>/resourceGroups/<resourceGroupName>/providers/Microsoft.DataFactory/factories/<dataFactoryName>/pipelines/<pipelineName>/createRun?api-version=2017-03-01-preview
Activity Logs	Every activity execution in a pipeline generates activity start and activity end logs event	
Integration Runtime	Replaces DMG as a way to move & process data in Azure PaaS Services, self-hosted or on prem or IaaS Works with VNETs Enables SSIS package execution	

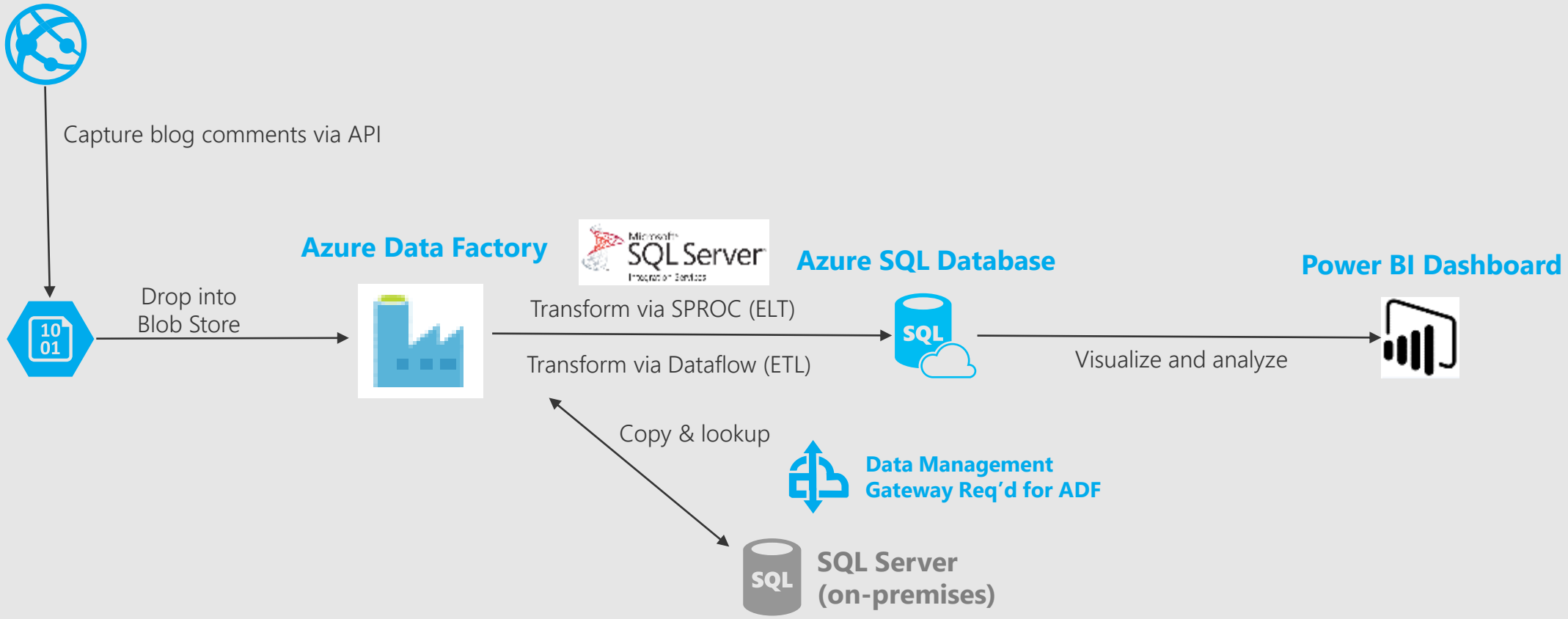
Scheduling	Flexible Scheduling
	Wall-clock scheduling
	Event-based triggers
		"type": "ScheduleTrigger",
		"typeProperties": {
		"recurrence": {
		"frequency": <<Minute, Hour, Day, Week, Year>>,
		"interval": <<int>>, // optional, how often to fire (default to 1)
		"startTime": <<datetime>>,
		"endTime": <<datetime>>,
		"timeZone": <<default UTC>>
		"schedule": { // optional (advanced scheduling specifics)
		"hours": [<<0-24>>],
		"weekDays": ": [<<Monday-Sunday>>],
		"minutes": [<<0-60>>],
		"monthDays": [<<1-31>>],
		"monthlyOccurences": [
		{
		"day": <<Monday-Sunday>>,
		"occurrence": <<1-5>>

New ADF V2 Concepts

Concept	Description	Sample
On-Demand Execution	Instantiate a pipeline by passing arguments as parameters defined in a pipeline and execute from script / REST / API.	Invoke-AzureRmDataFactoryV2PipelineRun -DataFactory $df -PipelineName "Adfv2QuickStartPipeline" -ParameterFile .\PipelineParameters.json
Parameters	Name-value pairs defined in the pipeline. Arguments for the defined parameters are passed during execution from the run context created by a Trigger or pipeline executed manually. Activities within the pipeline consume the parameter values. A **Dataset** is a strongly typed parameter and a reusable/referenceable entity. An activity can reference datasets and can consume the properties defined in the Dataset definition A **Linked Service** is also a strongly typed parameter containing the connection information to either a data store or a compute environment. It is also a reusable/referenceable entity.	Accessing parameters of other activities Using expressions @parameters("{name of parameter}") @activity("{Name of Activity}").output.RowsCopied
Incremental Data Loading	Leverage parameters and define your high-water mark for delta copy while moving dimension or reference tables from a relational store either on premises or in the cloud to load the data into the lake	

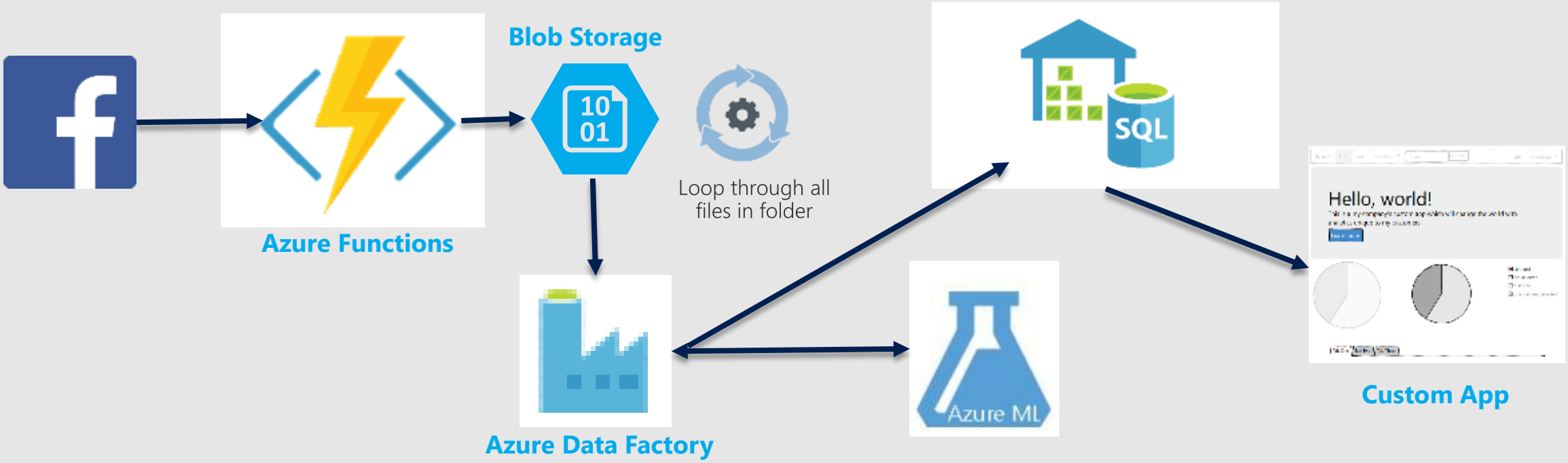
Patterns & Scenarios for ADF V2

Hybrid Data Integration Pattern 1:
Analyze blog comments

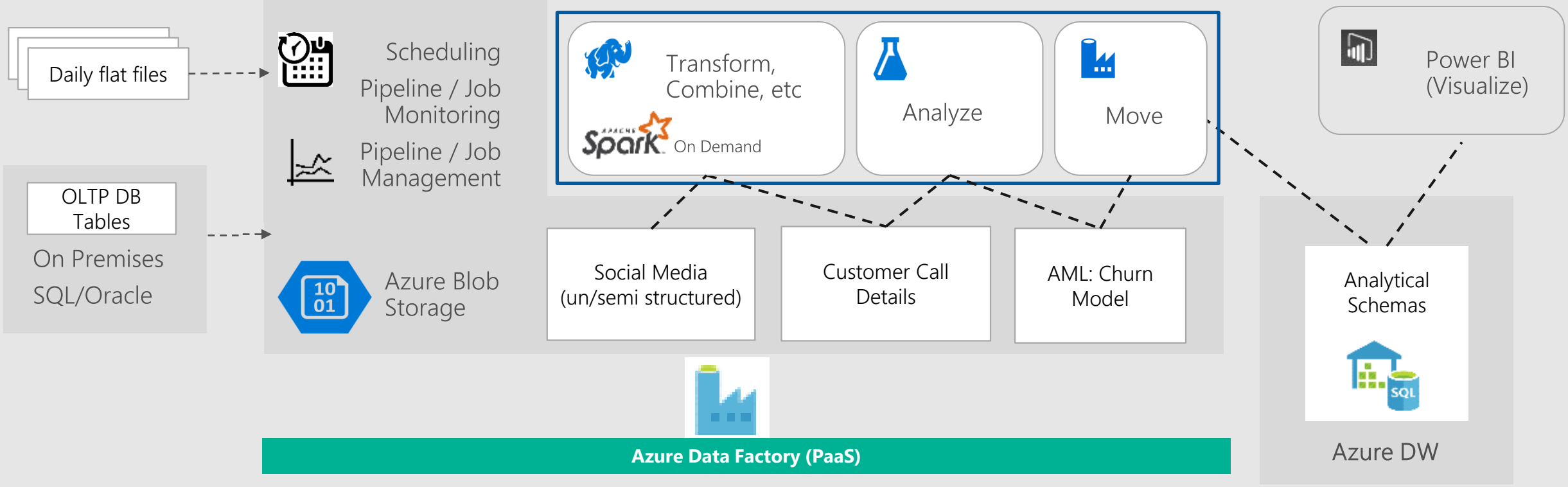


Hybrid Data Integration Pattern 2:

Data-Driven SaaS App: Sentiment Analysis w/Machine Learning

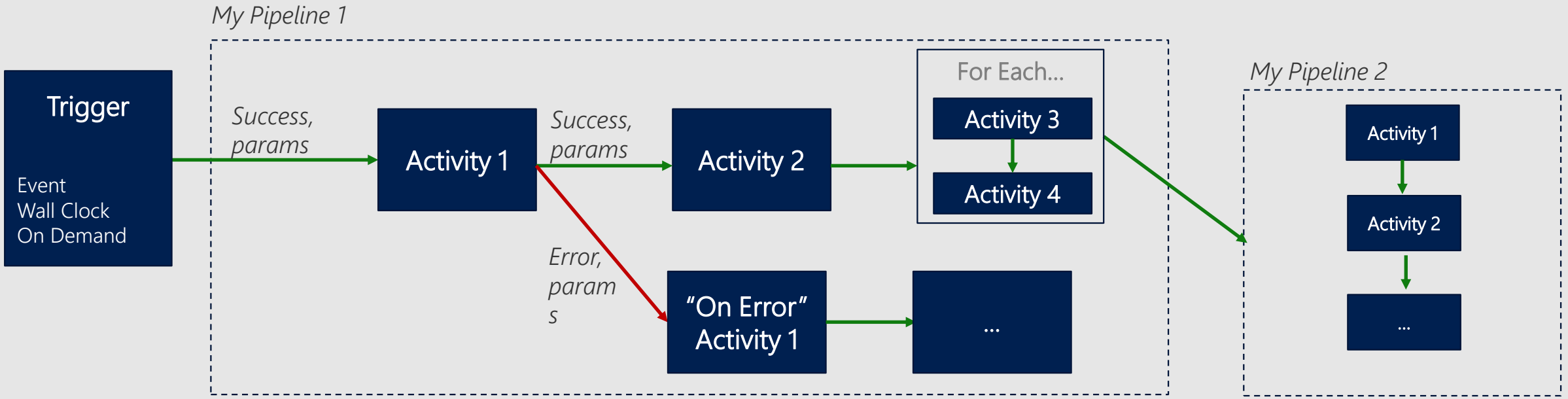


Hybrid Data Integration Pattern 3:
Modern Data Warehouse



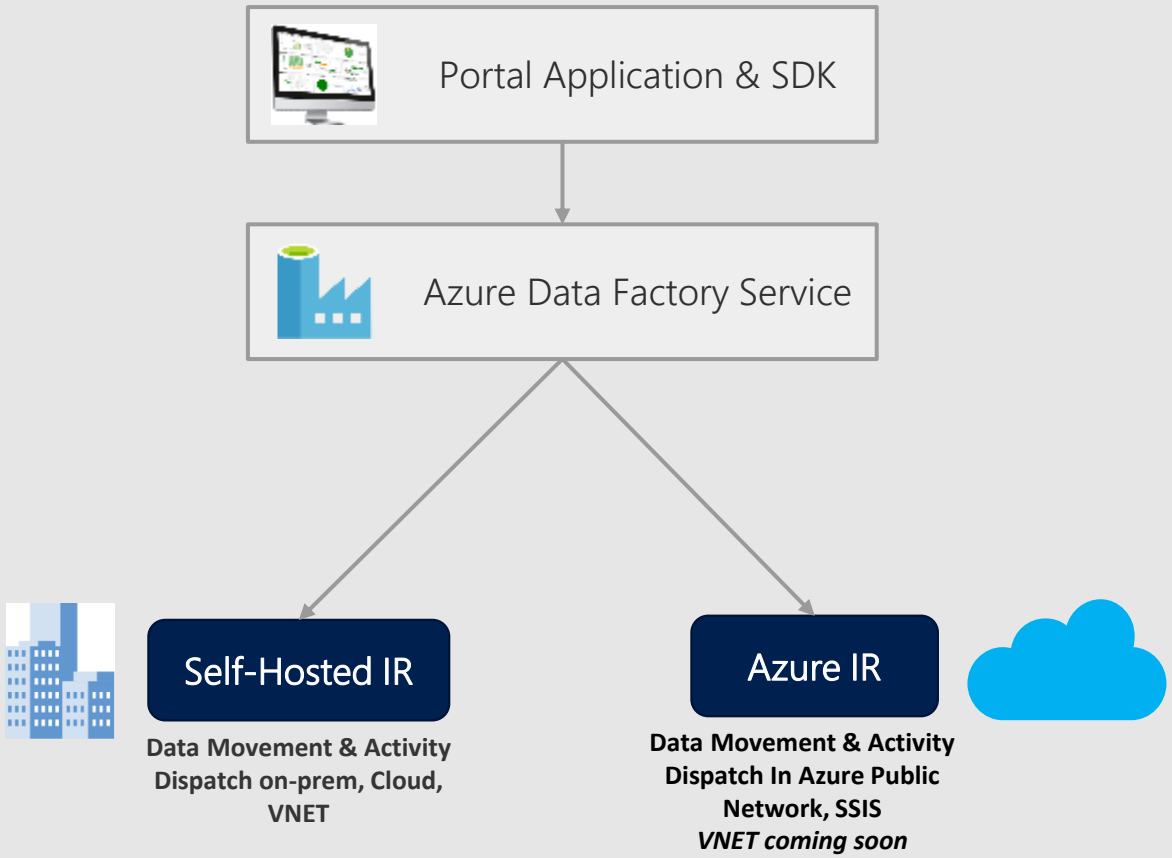
ADFv2: Control Flow

Coordinate pipeline activities into finite execution steps to enable looping, conditionals and chaining while separating data transformations into individual data flows



Integration Runtime

ADF Integration Runtime (IR)



- ADF compute environment with multiple capabilities:
  - Activity dispatch & monitoring
  - Data movement
  - SSIS package execution
- To integrate data flow and control flow across the enterprises' hybrid cloud, customer can instantiate multiple IR instances for different network environments:
  - On premises (similar to DMG in ADF V1)
  - In public cloud
  - Inside VNet
- Bring a consistent provision and monitoring experience across the network environments

Command & Control

Data Flow

UX & SDK

Authoring | Monitoring/Mgmt

Azure Data Factory Service

Scheduling | Orchestration | Monitoring

On Premises Apps & Data

SAP

TERADATA

Microsoft SQL Server

cloudera

Hortonworks

ORACLE

Cloud Apps, Svcs & Data

amazon web services

salesforce

Microsoft Dynamics

Adobe

Microsoft Azure

workday

Command & Control

Data Flow

UX & SDK

Authoring | Monitoring/Mgmt

Azure Data Factory Service

Scheduling | Orchestration | Monitoring

Azure Cloud

PaaS Cloud Host

Integration Runtime

On Premises Apps & Data

SAP

TERADATA

Microsoft SQL Server

cloudera

Hortonworks

ORACLE

Cloud Apps, Svcs & Data

amazon web services

salesforce

Microsoft Dynamics

Adobe

Microsoft Azure

workday

Command & Control

Data Flow

UX & SDK

Authoring | Monitoring/Mgmt

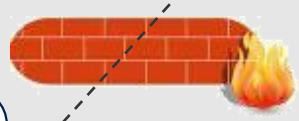
Azure Data Factory Service

Scheduling | Orchestration | Monitoring

Azure Cloud

PaaS Cloud Host

Integration Runtime



Installable Agent

Integration Runtime

On Premises Apps & Data

SAP

TERADATA

Microsoft SQL Server

cloudera

Hortonworks

ORACLE

Cloud Apps, Svcs & Data

amazon web services

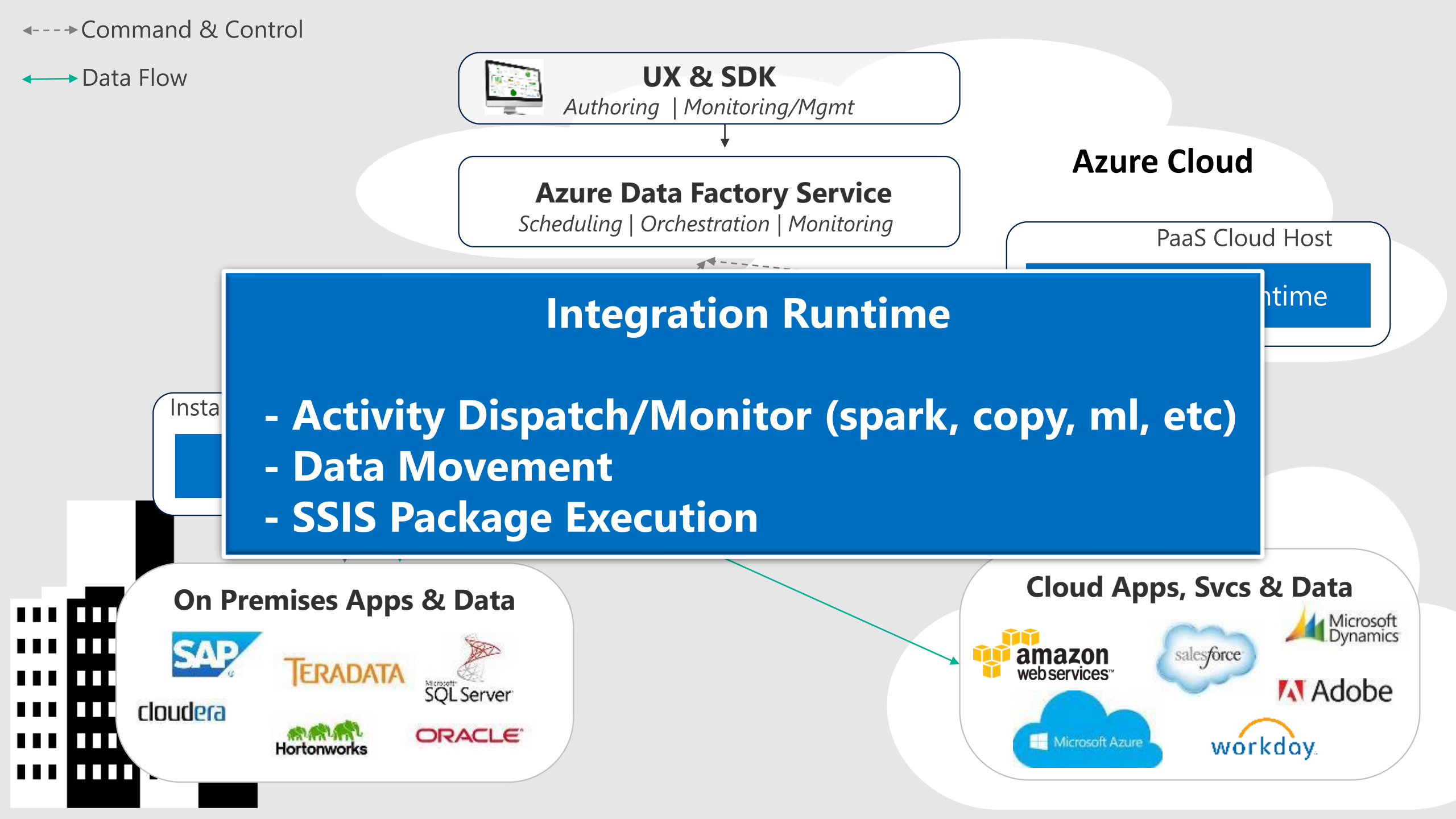
salesforce

Microsoft Dynamics

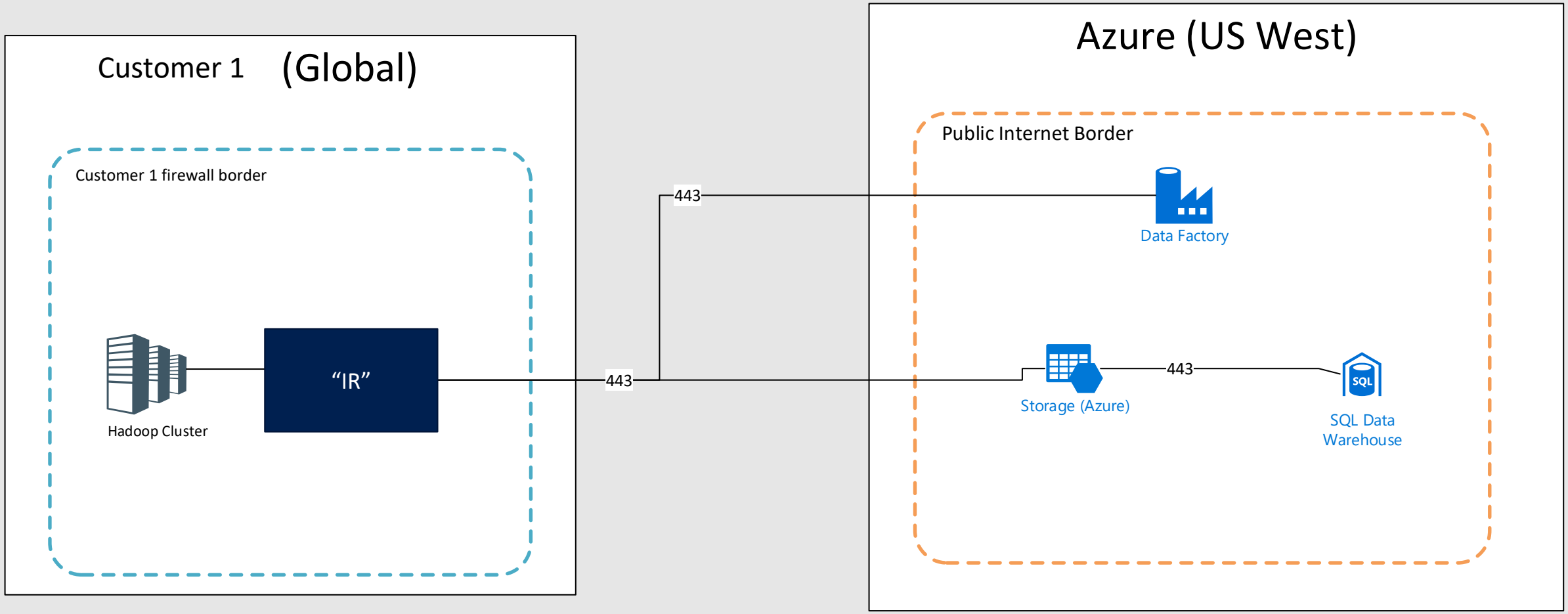
Adobe

Microsoft Azure

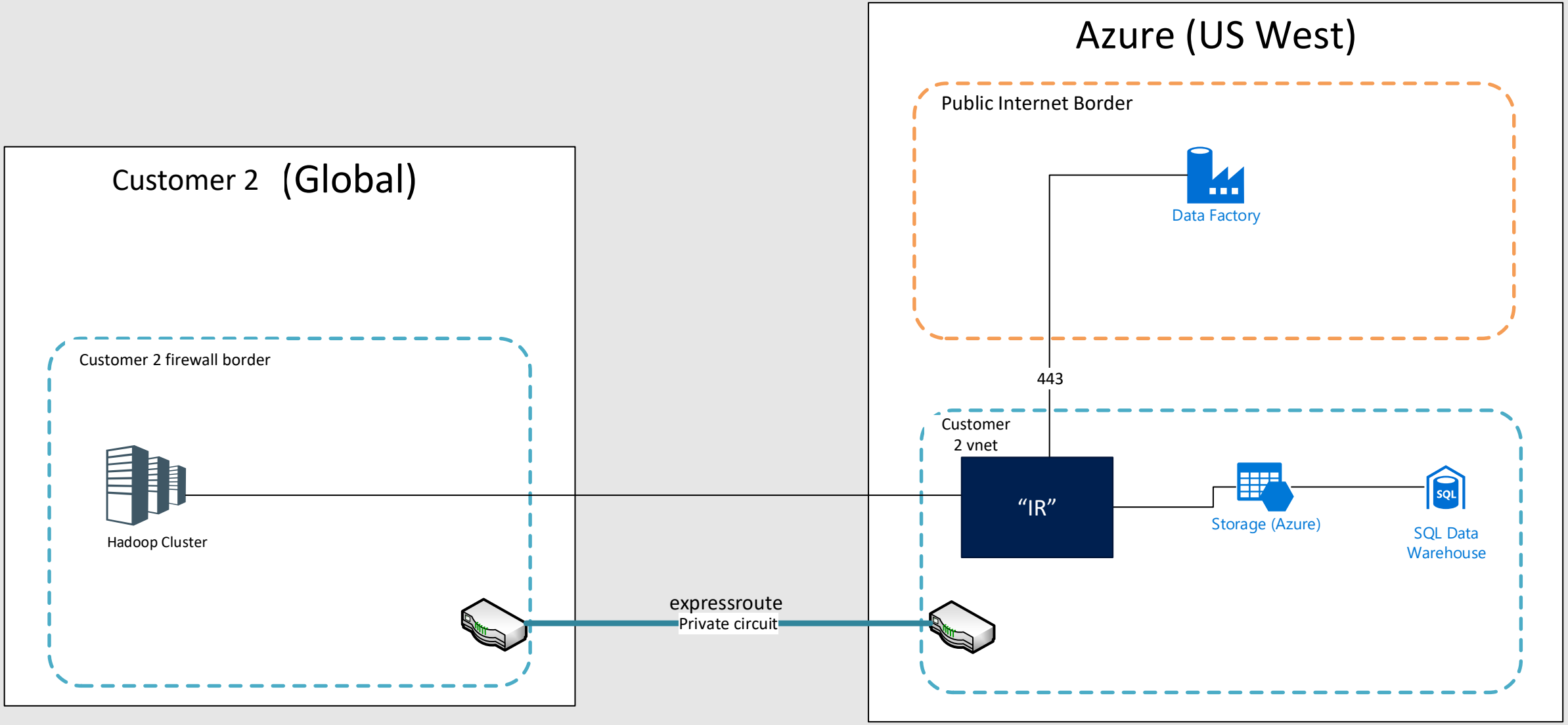
workday


Command & Control
Data Flow
UX & SDK
Authoring | Monitoring/Mgmt
Azure Data Factory Service
Scheduling | Orchestration | Monitoring
Azure Cloud
PaaS Cloud Host
ntime
Insta
Integration Runtime
- Activity Dispatch/Monitor (spark, copy, ml, etc)
- Data Movement
- SSIS Package Execution
On Premises Apps & Data
SAP
TERADATA
Microsoft SQL Server
cloudera
Hortonworks
ORACLE
Cloud Apps, Svcs & Data
amazon web services
salesforce
Microsoft Dynamics
Adobe
Microsoft Azure
workday.

Azure Data Factory “Integration Runtime” deployed on premises for transformation and then moved to cloud

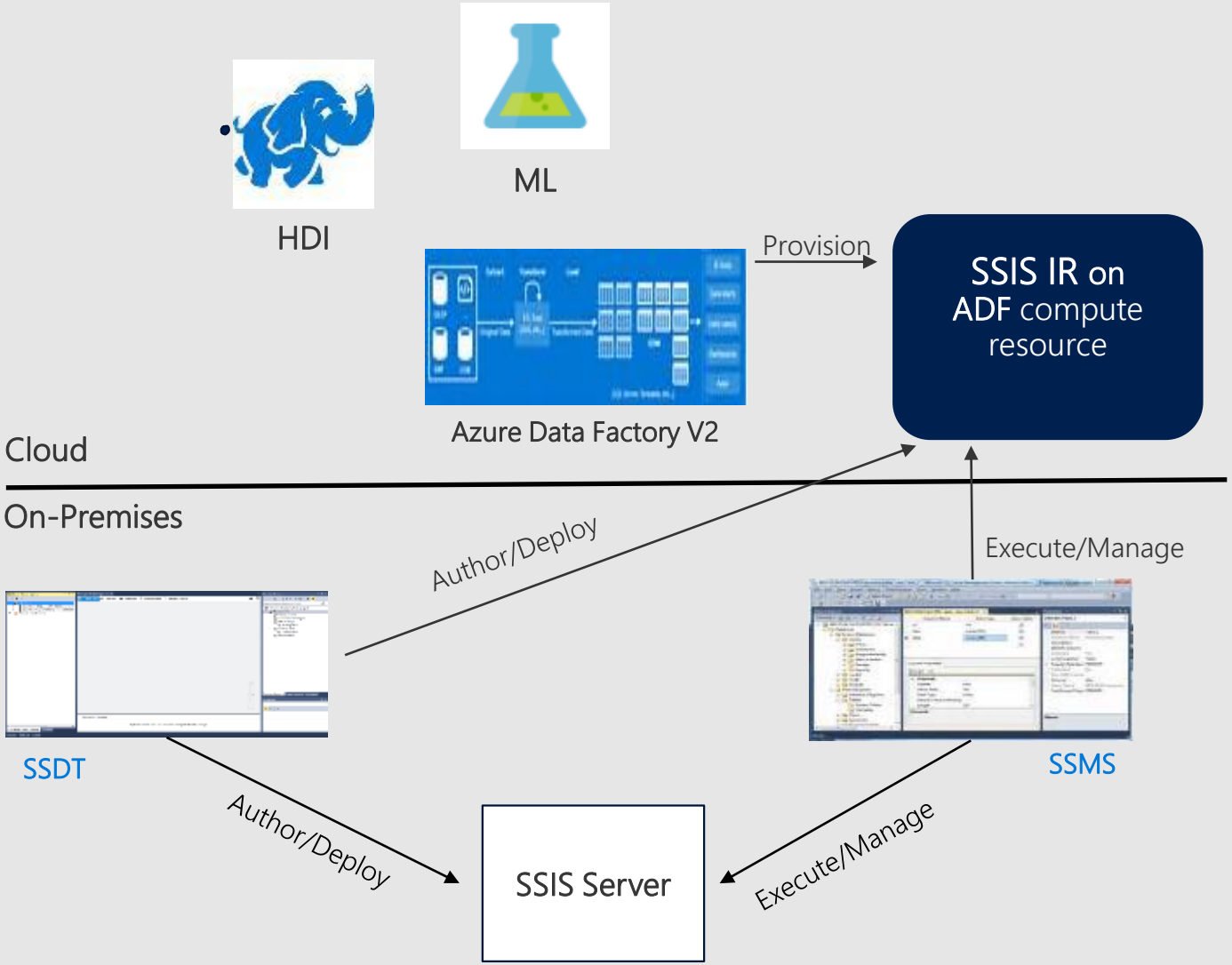


Azure Data Factory “Integration Runtime” deployed inside VNet



SSIS in ADF

SSIS as an Integration Runtime



- Provision SSIS IR via Data Factory
- Use SQL Server Data Tools (**SSDT**) to author/deploy SSIS packages
- Use SQL Server Management Studio (SSMS) to execute/manage SSIS packages
- Target SSIS customers who want to move all/part of their on-premises workloads and just "lift & shift" many existing packages to Azure
- Independent Software Vendors (ISVs) can build extensions/Software as a Service (**SaaS**) on SSIS Everest

Traditional ETL

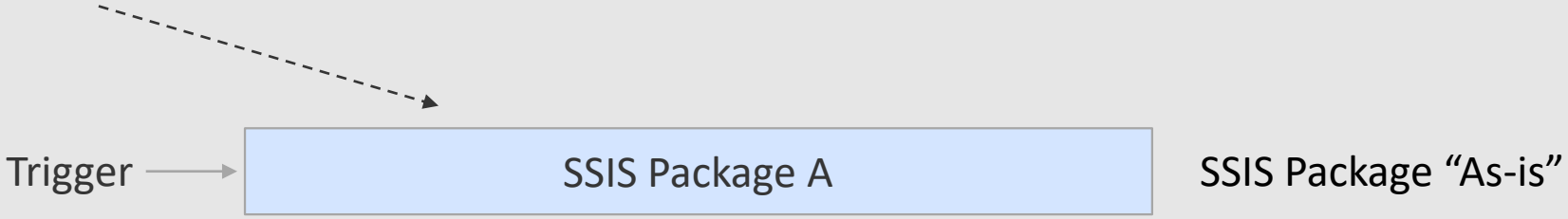
1. SSIS on Prem (to SQL Svr)

- *Lift & Shift w/ compatibility*

2. SSIS on IaaS (to SQL on IaaS | Az DB)

- *Want PaaS benefits (scale, no VM mgmt, etc)*
- *Mix reuse & modernization*

3. SSIS Runtime in ADF



Traditional ETL

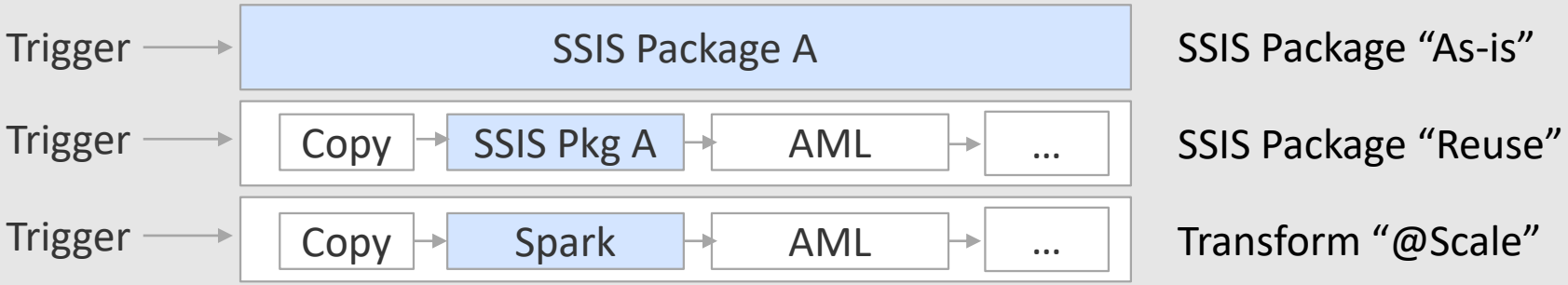
1. SSIS on Prem (to SQL Svr)

- *Lift & Shift w/ compatibility*

2. SSIS on IaaS (to SQL on IaaS | Az DB)

- *Want PaaS benefits (scale, no VM mgmt, etc)*
- *Mix reuse & modernization*

3. SSIS Runtime in ADF



Traditional ETL

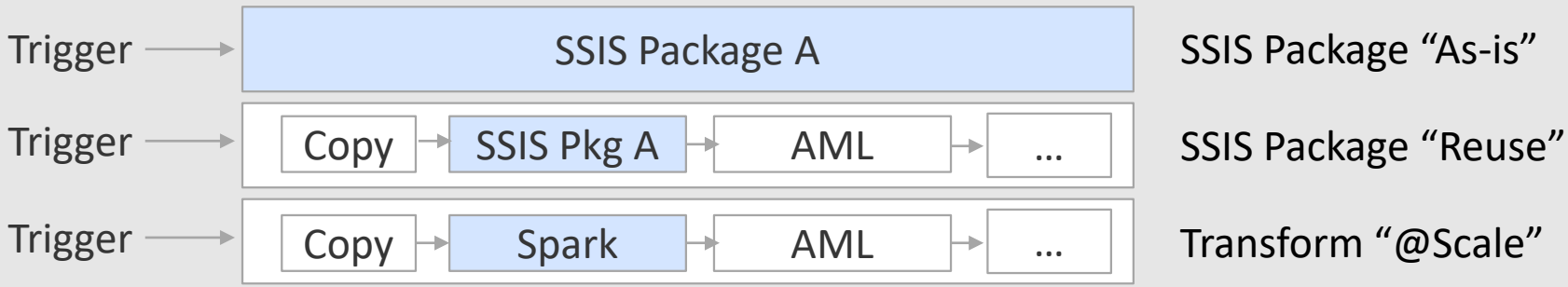
1. SSIS on Prem (to SQL Svr)
   - *Lift & Shift w/ compatibility*
2. SSIS on IaaS (to SQL on IaaS | Az DB)
   - *Want PaaS benefits (scale, no VM mgmt, etc)*
   - *Mix reuse & modernization*
3. **SSIS Runtime in ADF**

Modern DW & Data Driven SaaS

1. ADF V1 (Time-series, Tumbling Window)
   (scenarios: log processing, etc.)

Migrate v1 to v2:
- *Much more flexible model*
- *Similar or cheaper in many scenarios*

2. **ADF v2 (PaaS: Control Flow, Data Flow)**



ADF V2 Pricing (Preview Prices)

Orchestration

Units: Activity Runs

Activity Run: Single run or re-run of an activity or trigger

Where the activity is orchestrated	Price
Azure Integration Runtime (Public Network environment)	0 – 50K runs: **$.55/1k runs** 50K+ runs: **$.50/1k runs**
Self-Hosted Integration Runtime (Private Network environment)	**$.75/1K runs**

Data Movement

Units: DMU

DMU: Unit of measure used to copy data from source to sink

	Price
Azure Integration Runtime (Public Network environment)	$0.125/hr
Self-Hosted Integration Runtime (Private Network environment)	$0.05/hr

Inactive Pipelines (Pipelines not associated with a trigger with zero runs for a week) : $.20 / week

Azure Integration Runtime for SSIS

VM	Price for Standard Edition $/hr	*Price for Enterprise Edition $/hr*
A4 v2 (4core)	**$0.420**	***$0.956***
A8 v2 (8core)	**$0.862**	***$1.935***
D1 v2 (1core)	**$0.296**	***$0.832***
D2 v2 (2core)	**$0.397**	***$0.933***
D3 v2 (4core)	**$0.599**	***$1.136***
D4 v2 (8core)	**$1.199**	***$2.271***

The ADF Azure-SSIS IR is a fully-managed service to execute your SSIS packages. You can specify the VM type and the number of nodes for your dedicated IR environment. Note that you have to pay for a SQL Azure DB instance separately in order to host the SSIS catalog in the cloud. If you are using the Azure Data Factory to move data outside Azure network, you are required to pay for the amount of the data egress.

Note that these VM prices are discounted rates from the list prices of SQL Server VMs.

Enterprise SKUs for SSIS will not be available during public preview.

Scenario 1: Data movement across different data sources

Suppose you have a pipeline that has:

1. Copy Activity 1 moves data from On-Premises SQL Server to Azure Blob to be consumed later by a customized application
2. Copy Activity 2 moves data from Azure Blob to Azure SQL Database

Assume that the copy from On-Premises SQL Server to Azure Blob takes 2 hours. And the copy from Azure Blob to Azure SQL Database takes 1 hour with 2 DMUs. The pipeline is executed 30 times in a month.

Copy from On-Premises SQL Server to Azure Blob:

Data Movement: 30 activity runs X 2 hours duration for every run X $0.05 = $3

Activity Runs on Self-Hosted Integration Runtime (30 runs) + $.75

Subtotal = $3.75

Copy from Azure Blob to Azure SQL Database:

30 activity runs X 1 hour duration for every run X 2 DMUs x $.125 = $7.50

Activity Runs on Azure Integration Runtime (30 runs) + $0.55

Subtotal = $8.05

Total Price (per month) $3.75 + $8.05 = $11.80

Scenario 2: Orchestrating data transformation activities on compute services

Suppose you have a pipeline that has:

1. Spark Activity to run Spark application on an Azure HDInsight cluster in Azure Virtual Network
2. Azure Data Lake Analytics U-SQL Activity to execute U-SQL on Azure Data Lake Analytics

Assume the pipeline was executed for 30 times in one month.

Spark Activity Data Movement $0
Activity Runs on Self-Hosted Integration Runtime (30 runs) + $.75

Subtotal = $.75

Azure Data Lake Analytics U-SQL Activity Data Movement $0
Activity Runs on Azure Integration Runtime (30 runs) + $.55

Subtotal = $.55

Total Price (per month) **$1.30**

Scenario 3: Lift & Shift SSIS Packages

Suppose you have set up a SQL Server Standard and SSIS on a local 4-cores physical machine to run your daily ETL workload. Your ETL workload includes 100 SSIS packages and each of these packages takes about 5 mins to run. These packages are executed once daily and it requires about 8 hours each day to complete all the packages.

In order to lift and shift your SSIS package execution to Azure Data Factory, you need to provision the SSIS dedicated Integration Runtime via the Azure Data Factory portal

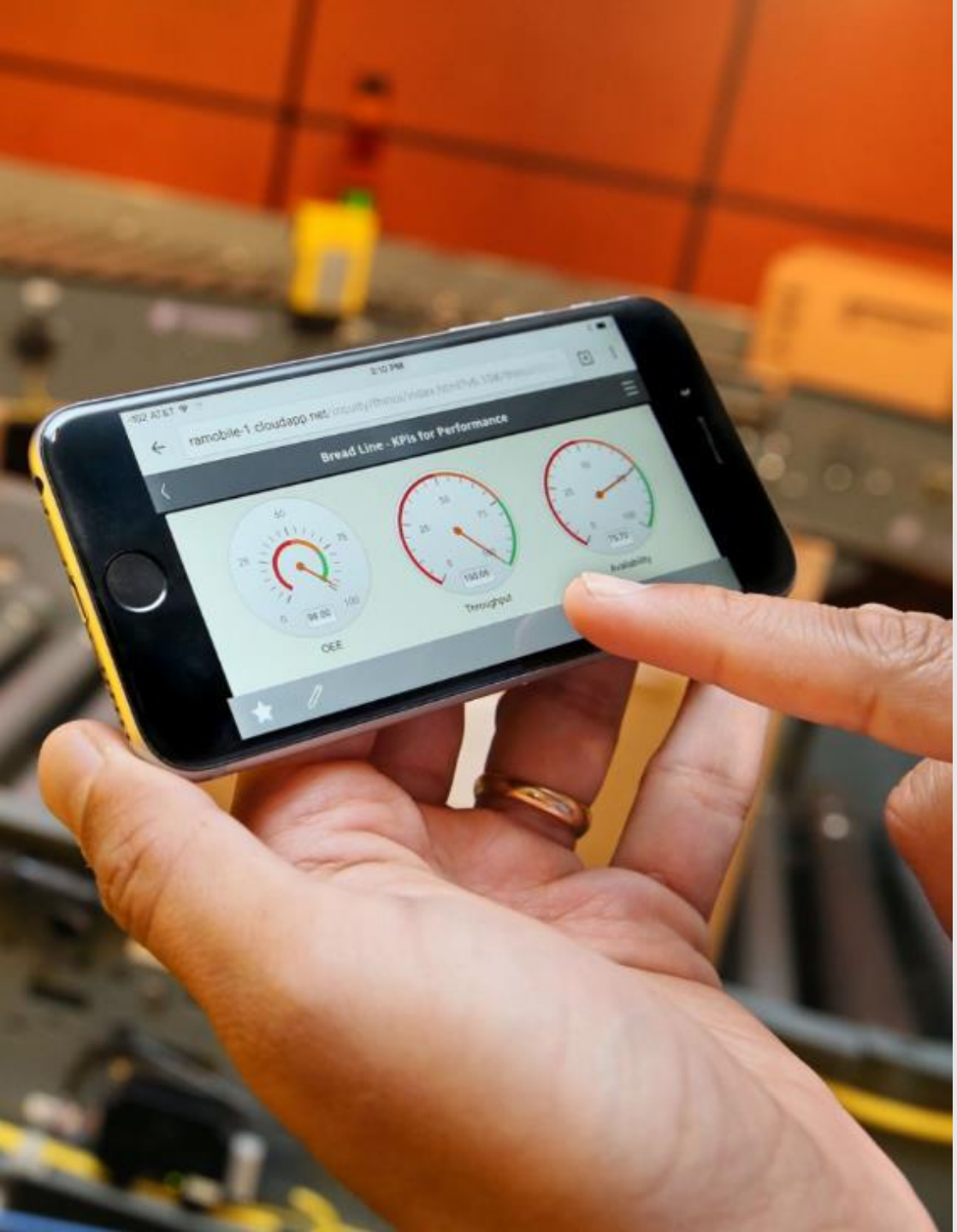
During the provision, you need to select the VM Type that match closely to the compute power you use on-premises. In this example, we will choose the A4 v2 type and 1 node to match the compute power on-premises. You are also required to create a SQL Azure Database instance for hosting the SSIS catalog on Azure.

We choose the S0 Standard for this example. SSIS Integration Runtime is a dedicated pool so the compute resource is dedicated only for you (not shared with other user). You can choose to keep the dedicated integration runtime pool 24/7 available or you can choose to spin it up and down

Integration Runtime Dedicated for 24/7	Cost
SSIS Integration Runtime – A4 v2	$0.42 / hr
X	1 node
X	24 hours / day (dedicated pool)
X	31 days / month
SSIS Integration Runtime 24/7 cost =	$312.48 / month
SQL Azure for SSIS Catalog – S0	$0.02 / hr
X	24 hours / day
X	31 days / month
SQL Azure for SSIS Catalog Cost =	$15.03 / month
Total monthly cost to run SSIS ETL workload	$327.51 / month

Integration Runtime Dedicated for the execution time only	Cost
SSIS Integration Runtime – A4 v2	$0.42 / hr
X	1 node
X	8 hours / day (dedicated pool)
X	31 days / month
SSIS Integration Runtime 24/7 cost =	$104.16 / month
SQL Azure for SSIS Catalog – S0	$0.0202 / hr
X	24 hours / day
X	31 days / month
SQL Azure for SSIS Catalog Cost =	$15.03 / month
ADF to orchestrate the start / stop SSIS Integration runtime	
2 custom activities – start / stop IR	2 activity runs / day
X	31 days / month
Total ADF Activity Cost	$.55 / month for 1 – 1000 runs
Total monthly cost to run SSIS ETL workload	$119.74 / month

Customer Evidence



Industrial automation firm cuts up to 90 percent of costs

Challenge

Manage data in excess of the existing data warehouse and systems capacity

Integrate data on-premises with cloud and batch processing with live streaming

Monitor a complex supply chain – undersea wells, refining facilities, gas stations

Solution

Integrated complex data systems with Azure Data Factory

Composed, orchestrated, and managed highly available data pipelines

Reduced up to 90 percent of costs

Rockwell Automation

Actuarial firm unlocks the potential of human capital to drive business

Challenge

Increasing complexity of insurance products and regulations

Diversion from firm's core purpose

Rise in on-premises hardware investments to handle high-performance grid computing

Solution

Built software-as-a-service on Azure to address modeling and reporting of large, compute-intensive workloads and used Azure Data Factory to automate extract, transform, and load (ETL) processes

Gained real-time insight into financial data

Milliman

INTEGRATE

Improved Health Outcomes by integrating healthcare data

Challenge

Integration of Healthcare data from multiple hospitals and heterogenous sources

Transformation and orchestration of dataflow at scale

Increasing costs of scaling on-premise data pipelines

Solution

Built end-to-end data integration with Azure Data Factory, leveraging big data and compute offerings in Azure

Enabled data ingestion from hybrid sources of data

Orchestrated complex data workflows at scale

LUMEDX