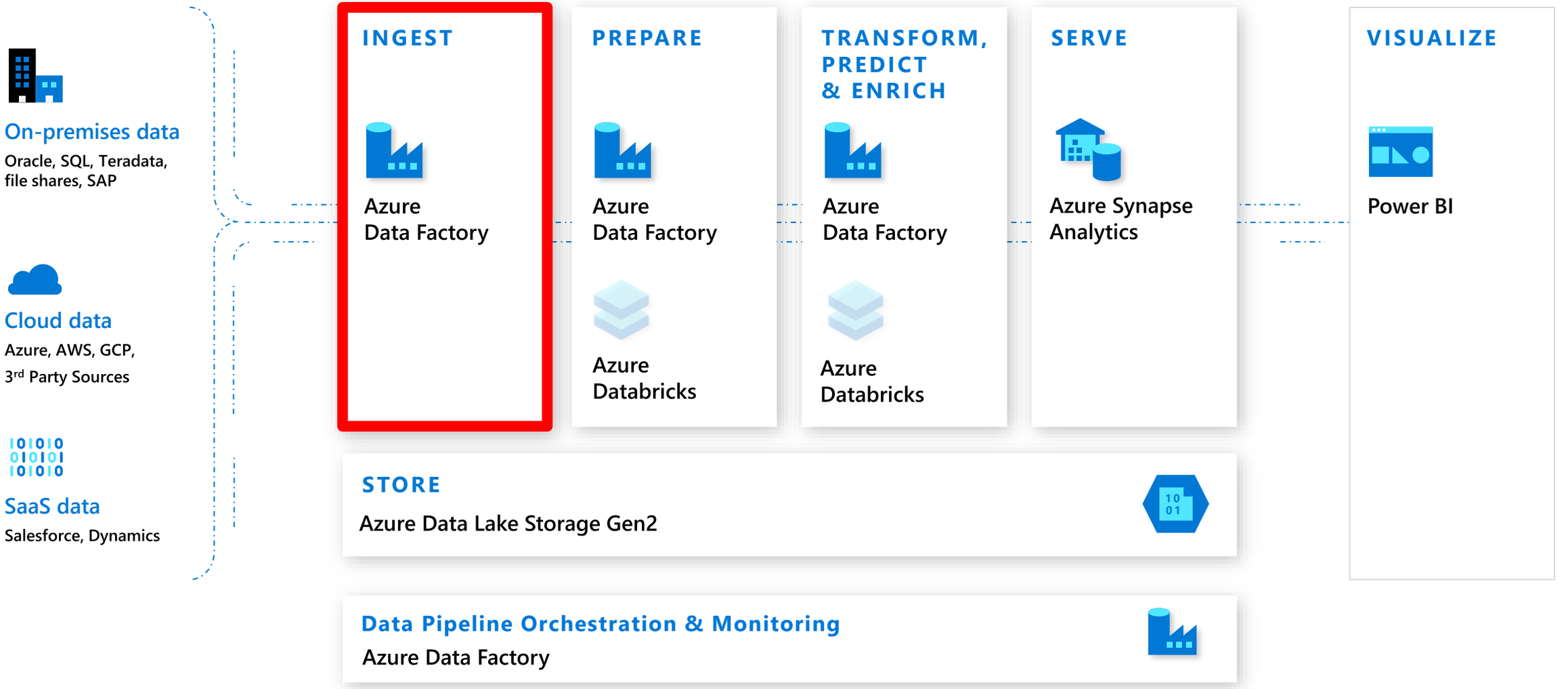
DATA() { EXPOSED;

Best practices of using ADF for data ingestion

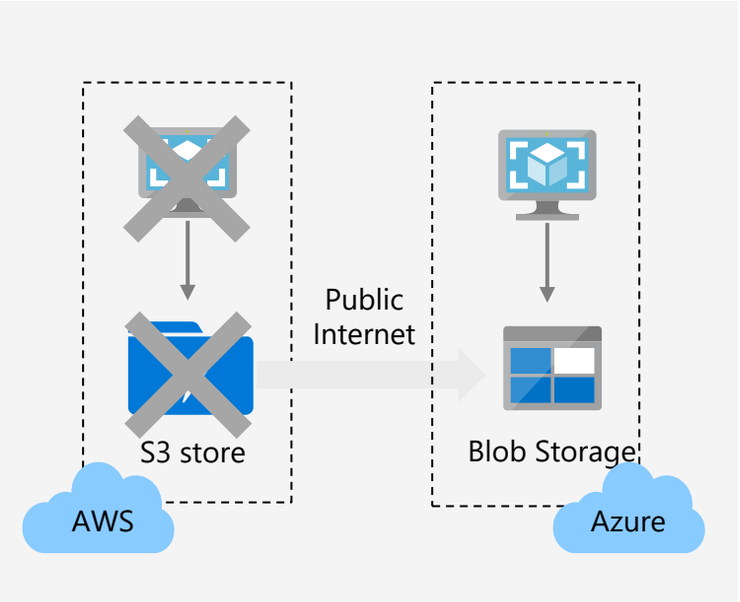
Ye Xu @AzDataFactory
ADF Program Manager

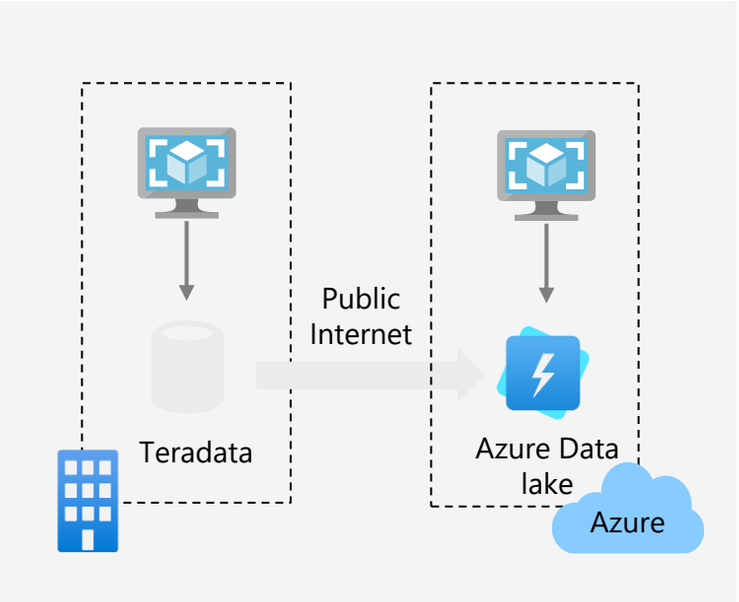
Data ingestion in Modern Data Warehouse (MDW)



Scenario #1: Data migration for data lake & EDW

- ☑ PB-scale big data workload migration from AWS S3 and on-prem Hadoop to Azure
- ☑ TB-size EDW migration from Oracle Exadata, Netezza, Teradata, and AWS Redshift to Azure





Customer example: migrate 2PB from S3→Blob in <11 days

- Tuned for perf & scale :**1.9PB initial load with 2.1GB/s** throughput
- Initial snapshot & incremental catch-up: **221TB incremental** load with **3.6GB/s throughput**
- Cost effective: **serverless**, **PAYG**

Customer example: migrated 25TB on-prem Netezza to ADLS

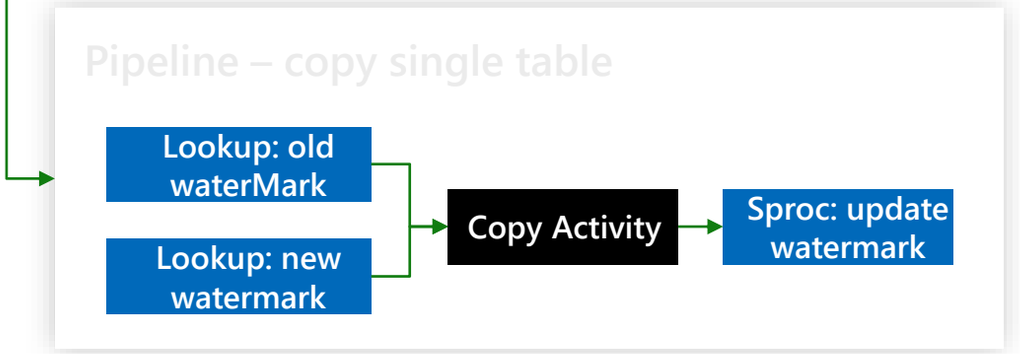
- **Flexible** migration plan**:** Designed nightly **migration window.**
- Control workload against data source**: Limited to 8 concurrent connections to cap DB overhead.**
- Completed migration in 3 weeks.

Metadata-Driven Pipeline for data migration

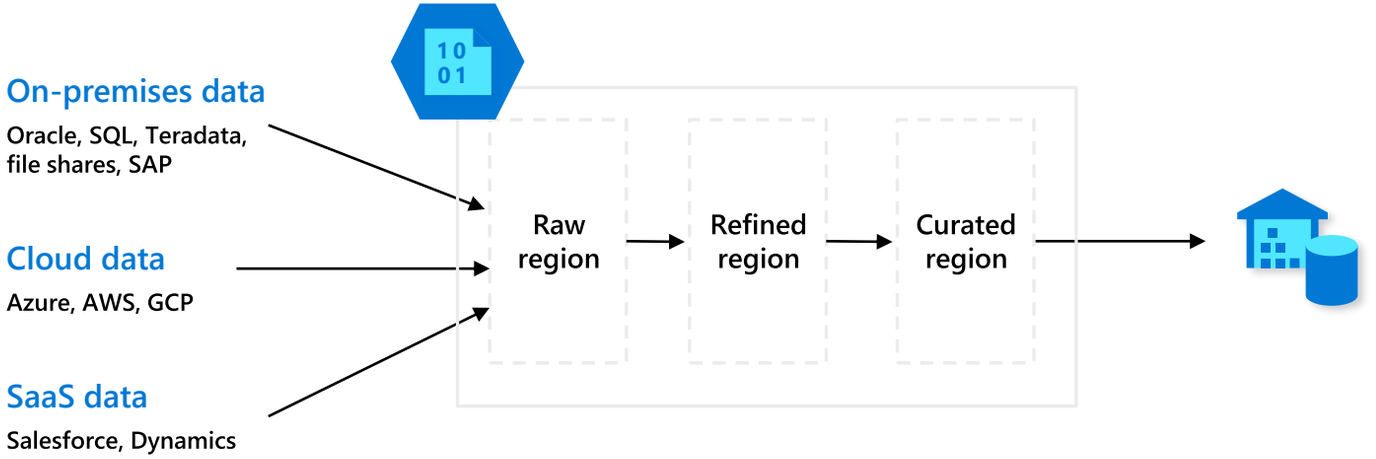
srcServerName	srcDatabaseName	SrcTableSchema	SrcTableName	DstFileName	WaterMarkColumnName	WatermarkValue
shwang-sql-server-east-us.da…	shwang_sqldb_adventureworksL…	[SalesLT]	[ProductModel]	[ProductModel]	[ModifiedDate]	2020-08-29 01:00:00.000
shwang-sql-server-east-us.da…	shwang_sqldb_adventureworksL…	[SalesLT]	[ProductDescription]	[ProductDescription]	[ModifiedDate]	2020-08-29 01:00:00.000
shwang-sql-server-east-us.da…	shwang_sqldb_adventureworksL…	[SalesLT]	[ProductModelProductDesc…	[ProductModelProduct…	[ModifiedDate]	2020-08-29 01:00:00.000
shwang-sql-server-west-us-2.…	shwang_sqldb_adventureworksL…	[SalesLT]	[Address]	[Address]	[ModifiedDate]	2020-08-29 01:00:00.000
shwang-sql-server-west-us-2.…	shwang_sqldb_adventureworksL…	[SalesLT]	[Customer]	[Customer]	[ModifiedDate]	2020-08-29 01:00:00.000
shwang-sql-server-west-us-2.…	shwang_sqldb_adventureworksL…	[SalesLT]	[CustomerAddress]	[CustomerAddress]	[ModifiedDate]	2020-08-29 01:00:00.000

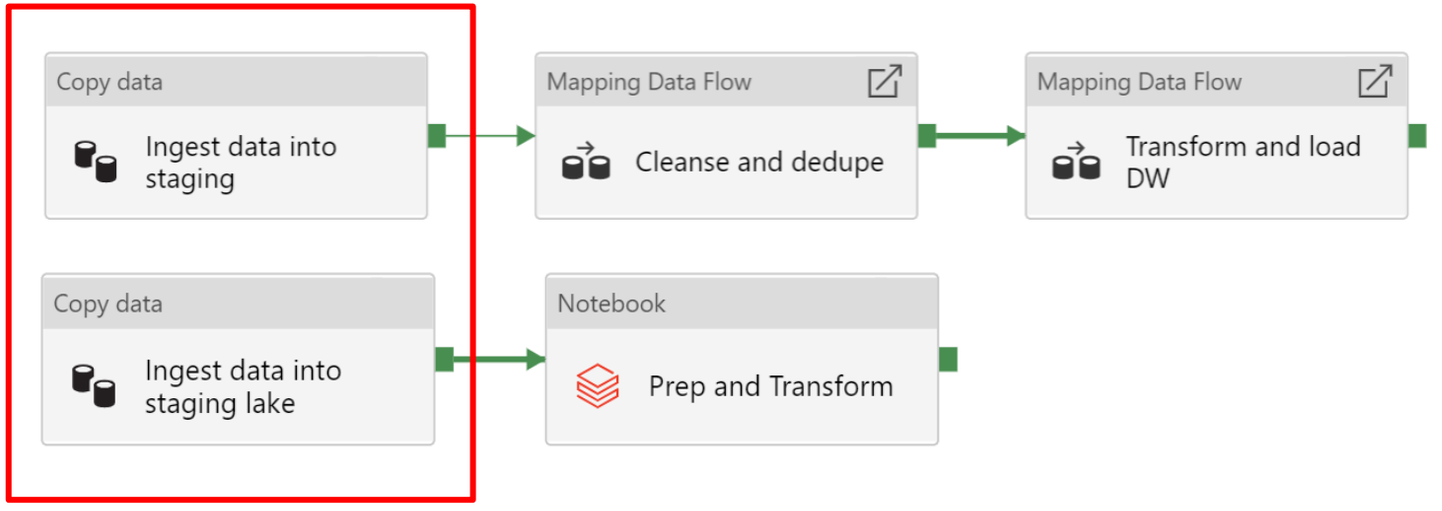
Tip:

- ✓ Do POC against sample data in your environment.
  1. To resolve the performance bottleneck.
  2. To capture your data structure for better partition plan for data migration.
  3. To estimate time required for entire data migration.
- ✓ Use metadata driven pipeline for data migration.
  1. To get flexibility to adjust job behaviors during data migration.
  2. To simply view the entire migration progress.



Scenario #2: Data ingestion for cloud ELT

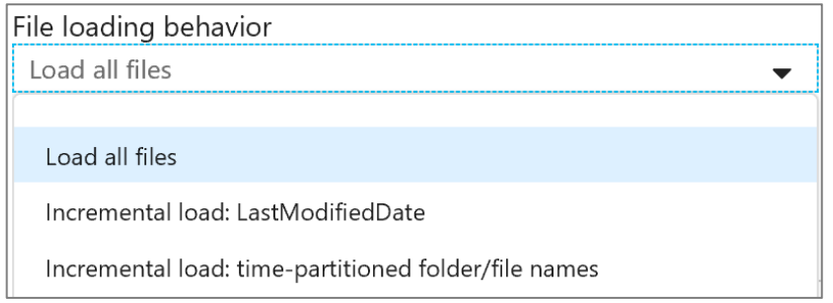




Benefits of using ADF

- **Rich built-in connectors:** file stores, RDBMS, NoSQL.
- **Hybrid connectivity:** on-prem, other public clouds, VNet/VPC
- **Enterprise grade security:** AAD auth, AKV integration
- **Developer productivity:** code-free authoring, CICD
- **Single-pane-of-class monitoring** & Azure Monitor integration

Incremental Extract from File Sources



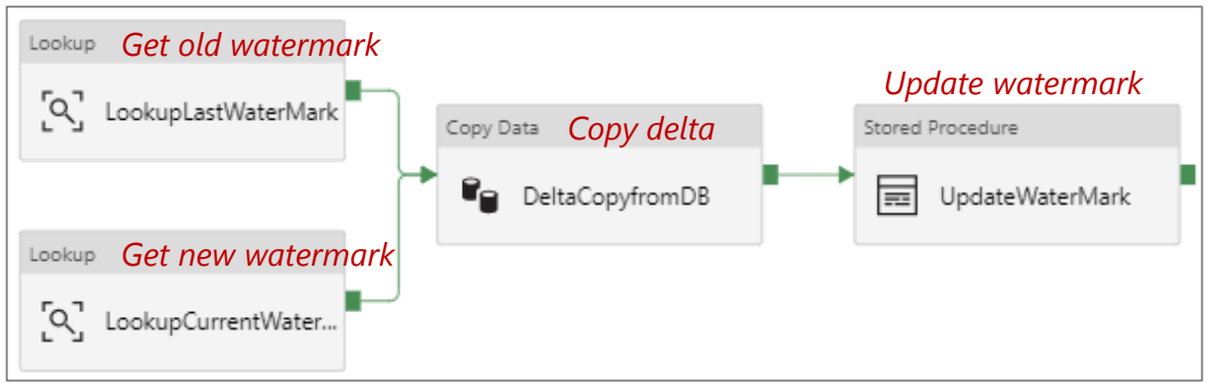
Tips:

- ✓ Determine which incremental pattern to use.
- ✓ Avoid enumerating large number of files but only copying few.
- ✓ Static path > data store's native filter > wildcard/last modified time.

	Incremental Data Pattern	Configuration
1	"**Move**" data – built-in copy + delete	Configure "Delete files after completion" in source 
2	**Date/time portioned folders/files** *e.g. path like root/folder/2021/01/10* *e.g. file name like <prefix>_20210111_<suffix>*	Dynamic folder/file/prefix/wildcard + tumbling window trigger *root/folder/@{formatDateTime(pipeline().parameters.windowStartTime, 'yyyy/MM/dd')* **_@{formatDateTime(pipeline().parameters.windowStartTime, 'yyyyMMdd')}_**
3	Identify newly created/updated files by **last modified time** *e.g. between 2021/01/10-2021/01/11*	Dynamic last modified time tumbling window trigger *modifiedDatetimeStart*: *@pipeline().parameters.windowStartTime* *modifiedDatetimeEnd*: *@pipeline().parameters.windowEndTime*
4	Custom way to get the list of new files, and let copy activity copy those files	In copy source, choose "List of files" 

Blog with more details: https://techcommunity.microsoft.com/t5/azure-data-factory/best-practices-of-how-to-use-adf-copy-activity-to-copy-new-files/ba-p/1532484

Retrieve changed data from Non-file Sources (DB/DW, NoSQL, SaaS)

	Incremental Data Pattern	Configuration
1	Data has **timestamp column** e.g. last modified time	Dynamic query + tumbling window trigger *SELECT * FROM MyTable* *WHERE LastModifiedDate >= @{formatDateTime(pipeline().parameters.windowStartTime, 'yyyy/MM/dd')* *AND LastModifiedDate < @{formatDateTime(pipeline().parameters.windowEndTime, 'yyyy/MM/dd')*
2	Data has **incremental column** e.g. ID (high watermark)	Control table/file + high watermark 
3	Data is **small in size** e.g. dimension data	Full copy and overwrite
4	Source has **change tracking**	Support for Azure SQL DB and SQL Server. Leverage SQL change tracking or CDC

DATA() {
EXPOSED;

Learn with us!

View our on-demand playlist:
aka.ms/azuresqlandadf

@AzureSQL
@AzDataFactory

Microsoft Azure