DATA() {
EXPOSED;

Data ingestion performance optimization

Linda Wang @AzDataFactory
ADF Program Manager

Agenda

- Data ingestion scenarios
- Breath of connectivity
- Copy performance features and best practice
- Performance tuning steps

Data movement scenarios

Ingest data using ADF to bootstrap your analytics workload

KEY SCENARIO	WHY ADF
Data migration for data lake & EDW 1. Big data workload migration from AWS S3, on-prem Hadoop File System, etc. 2. EDW migration from Oracle Exadata, Netezza, Teradata, AWS Redshift, etc.	• **Tuned for perf & scale:** PBs for data lake migration, tens of TB for EDW migration • **Cost effective:** serverless, PAYG • Support for **initial snapshot & incremental catch-up**
Data ingestion for cloud ETL 1. Load as-is from a variety of data stores 2. Stage for code-free or code-based transformation 3. Publish to DW for reporting or OLTP store for app consumption	• **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

Access all your data - 90+ built-in connectors & growing

Azure	Database & DW		File Storage	File Formats	NoSQL	Services & Apps		Generic
🟦 Blob Storage	🟪 Amazon Redshift	🟪 Phoenix	🟪 Amazon S3	🟦 Avro	🟪 Cassandra	🟪 Amazon MWS	🟪 PayPal	🟪 HTTP
🟦 Cosmos DB – SQL API	🟪 DB2	🟪 PostgreSQL	🟦 File System	🟦 Binary	🟪 Couchbase	🟦 CDS for Apps	🟪 QuickBooks	🟪 OData
🟦 Cosmos DB – MongoDB API	🟪 Drill	🟪 Presto	🟪 FTP	🟦 CDM	🟪 MongoDB	🟪 Concur	🟦 Salesforce	🟦 ODBC
🟦 ADLS Gen1	🟪 Google BigQuery	🟪 SAP BW Open Hub	🟪 Google Cloud Storage	🟦 Delimited Text	🟪 MongoDB Atlas	🟦 Dynamics 365	🟦 SF Service Cloud	🟦 REST
🟦 ADLS Gen2	🟪 Greenplum	🟪 SAP BW MDX	🟪 HDFS	🟦 Delta		🟪 Dynamics AX	🟪 SF Marketing Cloud	
🟦 Data Explorer	🟪 HBase	🟦 SAP HANA	🟦 SFTP	🟪 Excel		🟦 Dynamics CRM	🟦 SAP C4C	
🟪 Database for MariaDB	🟪 Hive	🟪 SAP Table		🟦 JSON		🟪 Google AdWords	🟪 SAP ECC	
🟦 Database for MySQL	🟪 Impala	🟦 Snowflake		🟦 ORC		🟪 HubSpot	🟪 ServiceNow	
🟦 Database for PostgreSQL	🟪 Informix	🟪 Spark		🟦 Parquet		🟪 Jira	🟪 SharePoint List	
🟦 Databricks Delta Lake	🟪 MariaDB	🟦 SQL Server		🟪 XML		🟪 Magento	🟪 Shopify	
🟦 File Storage	🟪 Microsoft Access	🟪 Sybase				🟪 Marketo	🟪 Square	
🟦 SQL Database	🟪 MySQL	🟪 Teradata				🟪 Office 365	🟪 Web Table	
🟦 SQL Database MI	🟪 Netezza	🟪 Vertica				🟪 Oracle Eloqua	🟪 Xero	
🟦 Synapse Analytics	🟦 Oracle					🟪 Oracle Responsys	🟪 Zoho	
🟦 Search Index						🟪 Oracle Service Cloud		
🟦 Table Storage						🟪		

🟦 Support read & write

🟪 Support read only

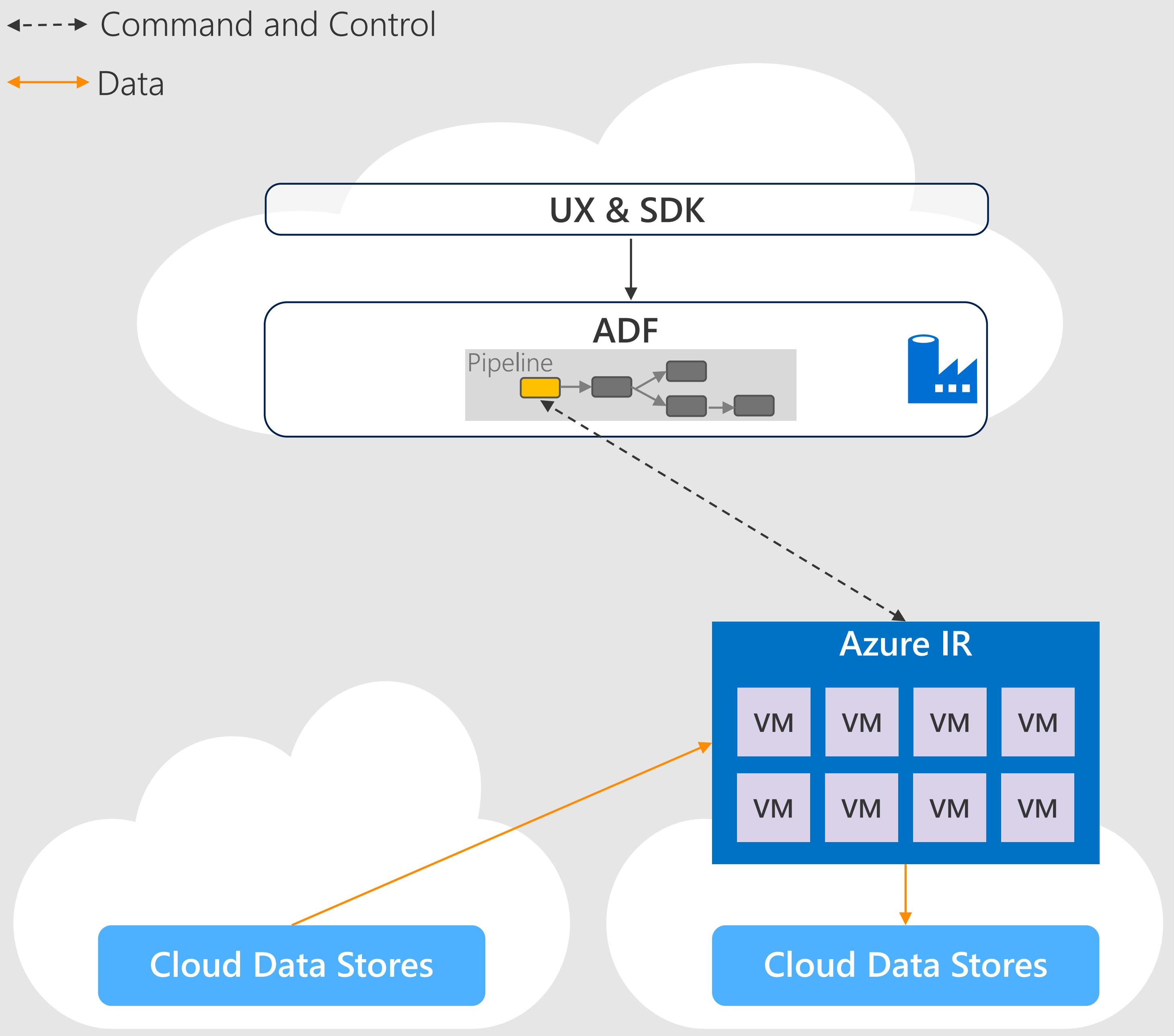
Connectivity to additional data stores

Not in the supported list? No worries:

Database/DW	✓ Use **generic ODBC** connector	ODBC
SaaS apps	✓ If it provides RESTful APIs, use **generic REST** connector ✓ If it provides SOAP APIs, use **generic HTTP** connector ✓ If it has OData feed, use **generic OData** connector	REST OData
Custom	✓ Check if you can load data to or expose data as ADF **supported data stores**, e.g. Azure Blob/File/FTP/SFTP/Amazon S3. ✓ Invoke **custom data loading mechanism** via Azure Function, Custom activity, Databricks/HDInsight, Web activity, etc.	File System SFTP .....

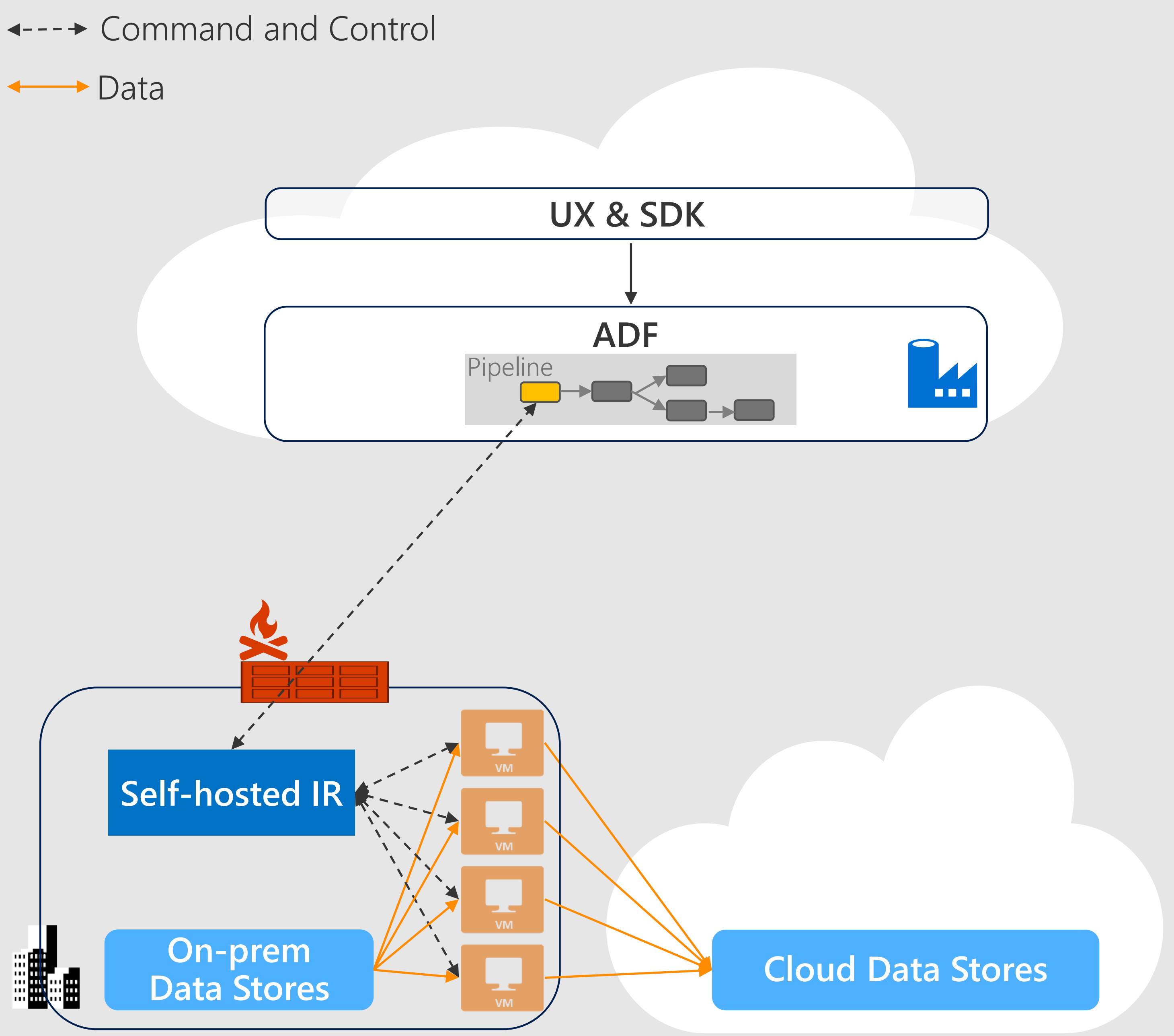
Fully-managed runtime

- **Azure Integration Runtime:** managed, serverless, and pay-as-you-go
- Support **managed virtual network**.
- Specify how much horsepower to use for each copy by **Data Integration Units (DIUs)**
- DIU is a combination of CPU, memory, and network resource allocation.
- Default behavior based on your data pattern – larger file size & file count, larger DIUs.
- You can set DIU = 2, 4, 8, ..., 256.



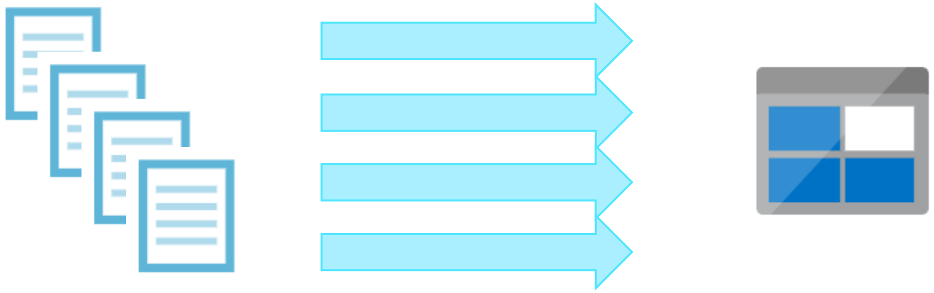
Self-hosted runtime

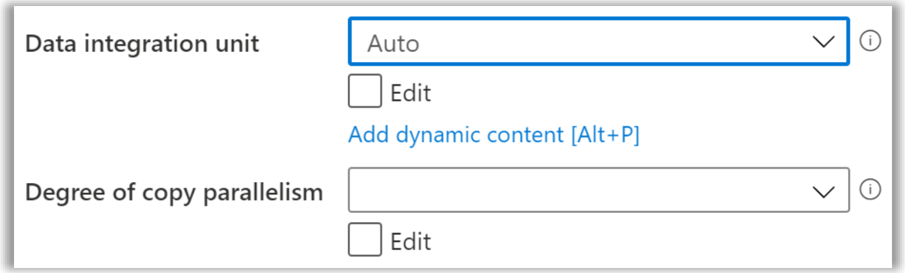
- **Self-hosted Integration Runtime:** component installed on machine on-prem or VM in cloud
- **Touchless:** latest version automatically pushed down to machine during downtime
- **HA and scale-out:** register up to 4 nodes for each self-hosted IR.
- **Active-active mode:** requests are dispatched to each node.
- **Single-node concurrency:** # of concurrent activity runs on each node, determined based on IR CPU/memory, can be tuned.



Extract data from file sources

Copy activity **in parallel** copy/parse multiple files (determined by “parallel copy”) **across Azure IR Data Integration Units (DIUs) or Self-hosted IR nodes**; within each file, data is handled by chunks concurrently.





Data integration unit: Auto

□ Edit

Add dynamic content [Alt+P]

Degree of copy parallelism:

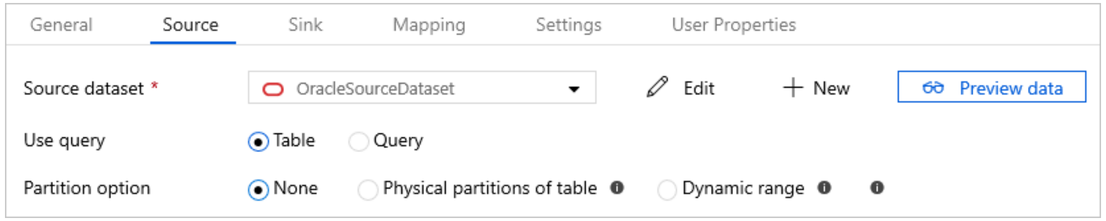
□ Edit

Tips:

- Start with default DIUs. Increase the DIU if your source has a number of files and with large size or file count.
- When using file/folder filter, avoid enumerating large number of files but only copying few.
- To copy to file-based sink:
    - Use default “degree of parallelism”, ADF auto determine based on your file pattern – size/count.
    - To copy files as-is, use Binary dataset to avoid unexpected SerDes.
- To copy to non-file sink, you can increase the “degree of parallelism” to enable parallel write.

Extract data from non-file sources (database, NoSQL, SaaS)

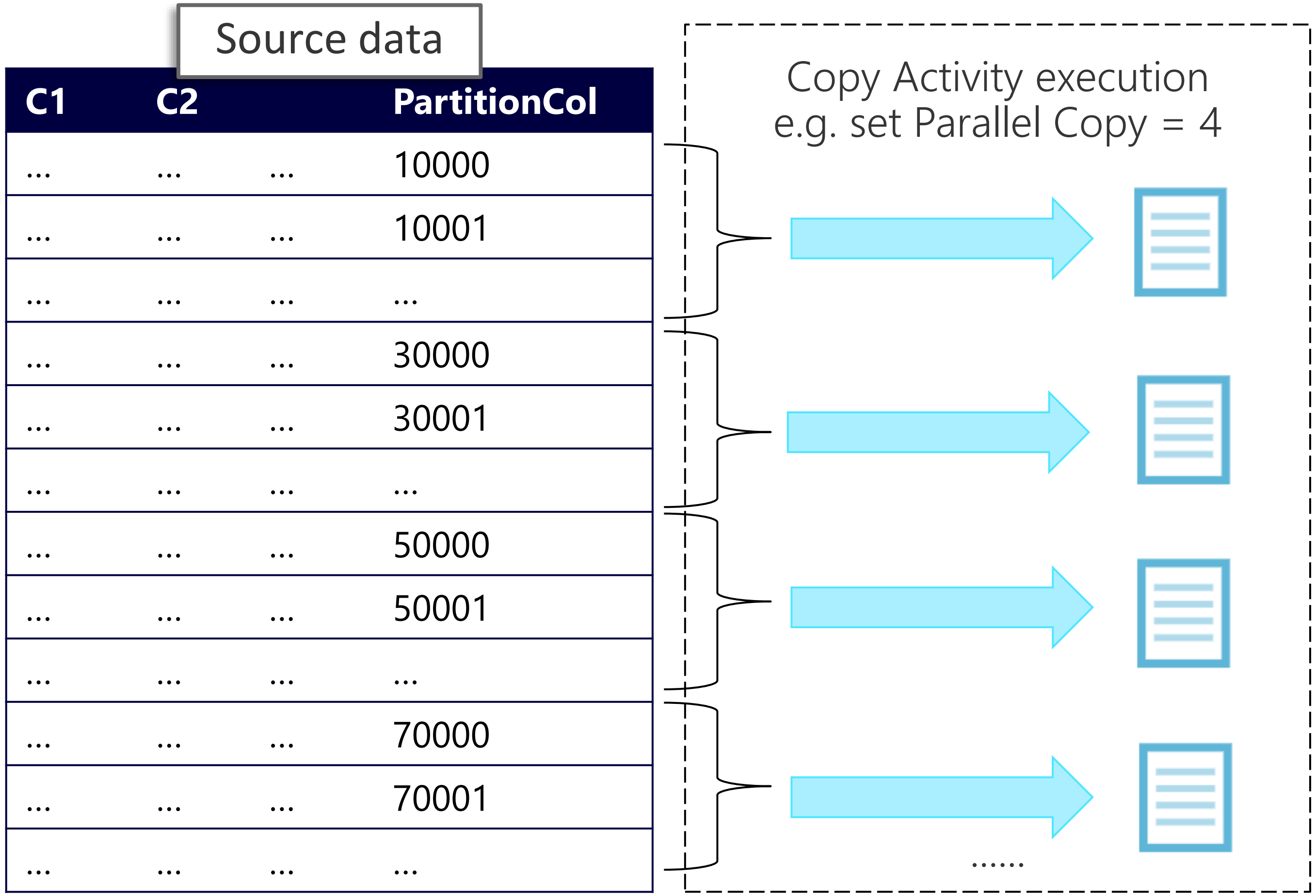
Out-of-box optimization for Azure SQL Database, Azure SQL MI, Azure Synapse Analytics, SQL Server, Oracle, Teradata, Netezza, SAP HANA, SAP Table & SAP BW via Open Hub



Source dataset * OracleSourceDataset

Use query ● Table ○ Query

Partition option ● None ○ Physical partitions of table ○ Dynamic range



Source data

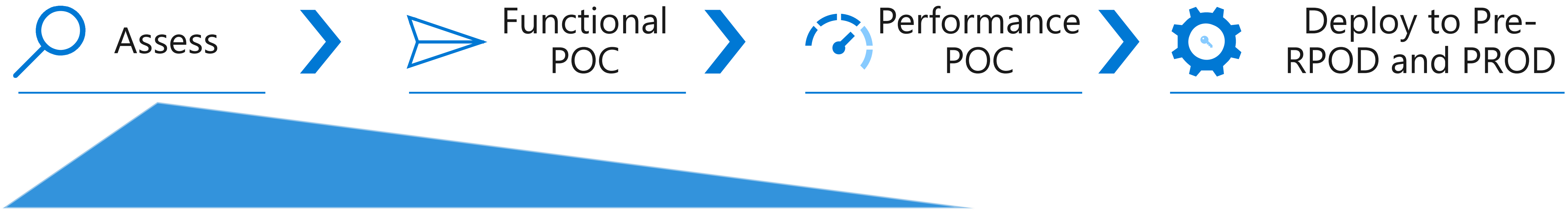
C1	C2		PartitionCol
...	...	...	10000
...	...	...	10001
...	...	...	...
...	...	...	30000
...	...	...	30001
...	...	...	...
...	...	...	50000
...	...	...	50001
...	...	...	...
...	...	...	70000
...	...	...	70001
...	...	...	...

Copy Activity execution
e.g. set Parallel Copy = 4

Tips:

- Enable built-in parallel copy by partitions to boost perf for large migration/ingestion.
- Options of range partition and native partition mechanism per data store.
- Tune the parallel copy count and DIUs/# of Self-hosted IR nodes to increase parallelism.
- For data sources without built-in partition, for very large copy, manually partition to multiple copy activities to divide and conquer.

Build your solution – process



Assess

Functional POC

Performance POC

Deploy to Pre-RPOD and PROD

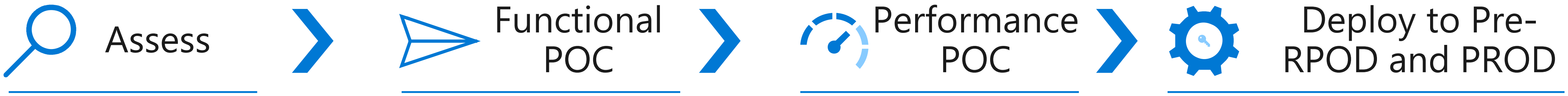
Understand the scenario and the workload:

- **Connectivity:** What are the source and sink stores? Which format?
- **Network:** What is the network requirement?
- **Data loading pattern:** One-time historical or incremental copy?
- **Scale:** What is the data volume, # of objects (tables/files) and size distribution?

Identify key criteria:

- **Security** requirement
- **Performance** expectation
- Special need

Build your solution – process



Assess

Functional POC

Performance POC

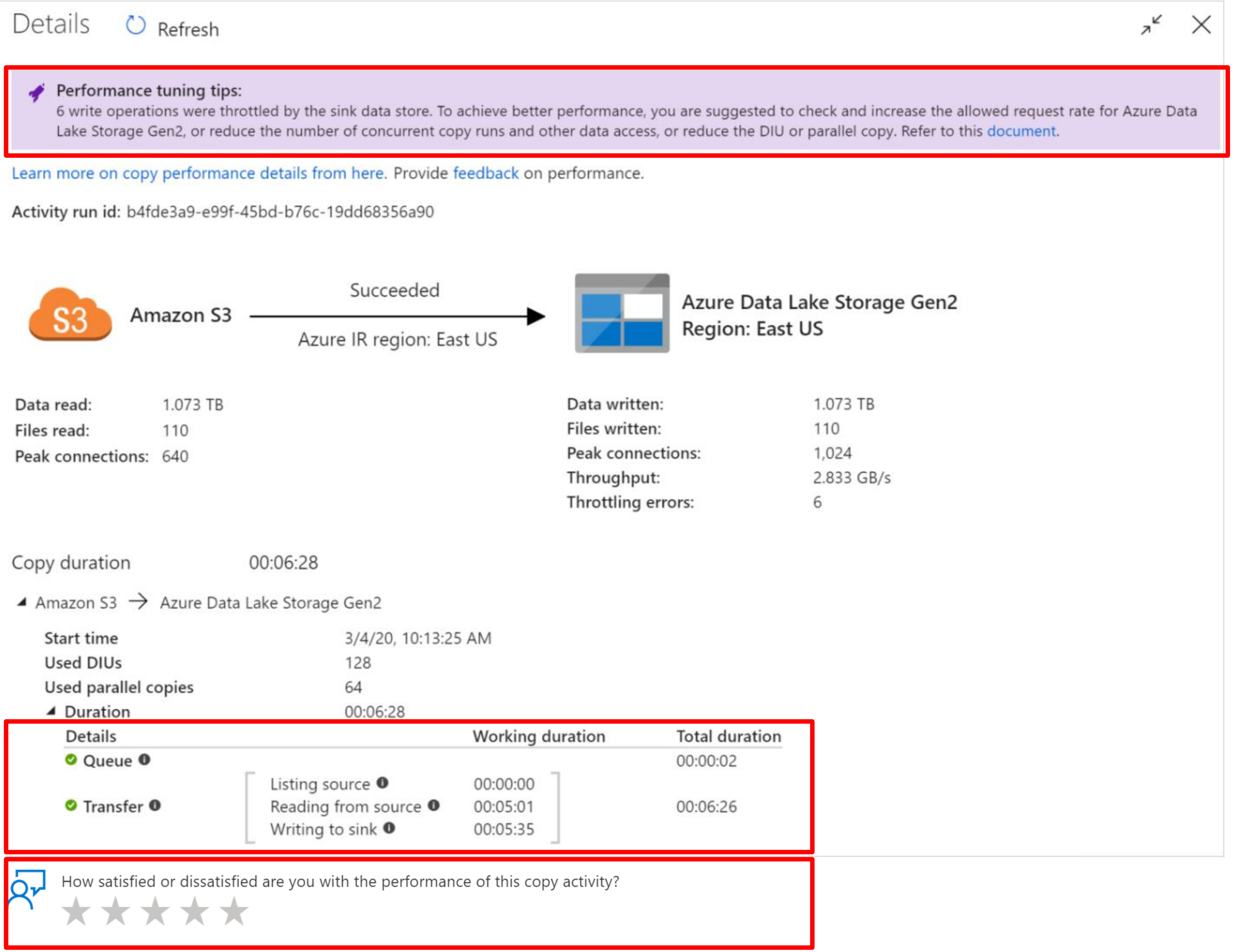
Deploy to Pre-RPOD and PROD

Performance test and tuning to handle data at scale:

1. Pick a representative workload
2. Obtain throughput baseline and compare to the performance expectation
3. Identify performance bottlenecks
4. Tune configurations to optimize
5. Scale to entire dataset

Consideration: maximize performance of a single copy activity
VS maximize aggregate throughput across objects/data stores

Easily identify bottlenecks



Performance tuning tips:
6 write operations were throttled by the sink data store. To achieve better performance, you are suggested to check and increase the allowed request rate for Azure Data Lake Storage Gen2, or reduce the number of concurrent copy runs and other data access, or reduce the DIU or parallel copy. Refer to this document.

Learn more on copy performance details from here. Provide feedback on performance.

Activity run id: b4fde3a9-e99f-45bd-b76c-19dd68356a90

Amazon S3 → Azure Data Lake Storage Gen2
Succeeded
Azure IR region: East US
Azure Data Lake Storage Gen2
Region: East US

Data read:	1.073 TB
Files read:	110
Peak connections:	640

Data written:	1.073 TB
Files written:	110
Peak connections:	1,024
Throughput:	2.833 GB/s
Throttling errors:	6

Copy duration 00:06:28

◢ Amazon S3 → Azure Data Lake Storage Gen2

Start time	3/4/20, 10:13:25 AM
Used DIUs	128
Used parallel copies	64
◢ Duration	00:06:28

Details		Working duration	Total duration
✓ Queue			00:00:02
✓ Transfer	Listing source	00:00:00	00:06:26
	Reading from source	00:05:01	
	Writing to sink	00:05:35	

How satisfied or dissatisfied are you with the performance of this copy activity?
☆ ☆ ☆ ☆ ☆

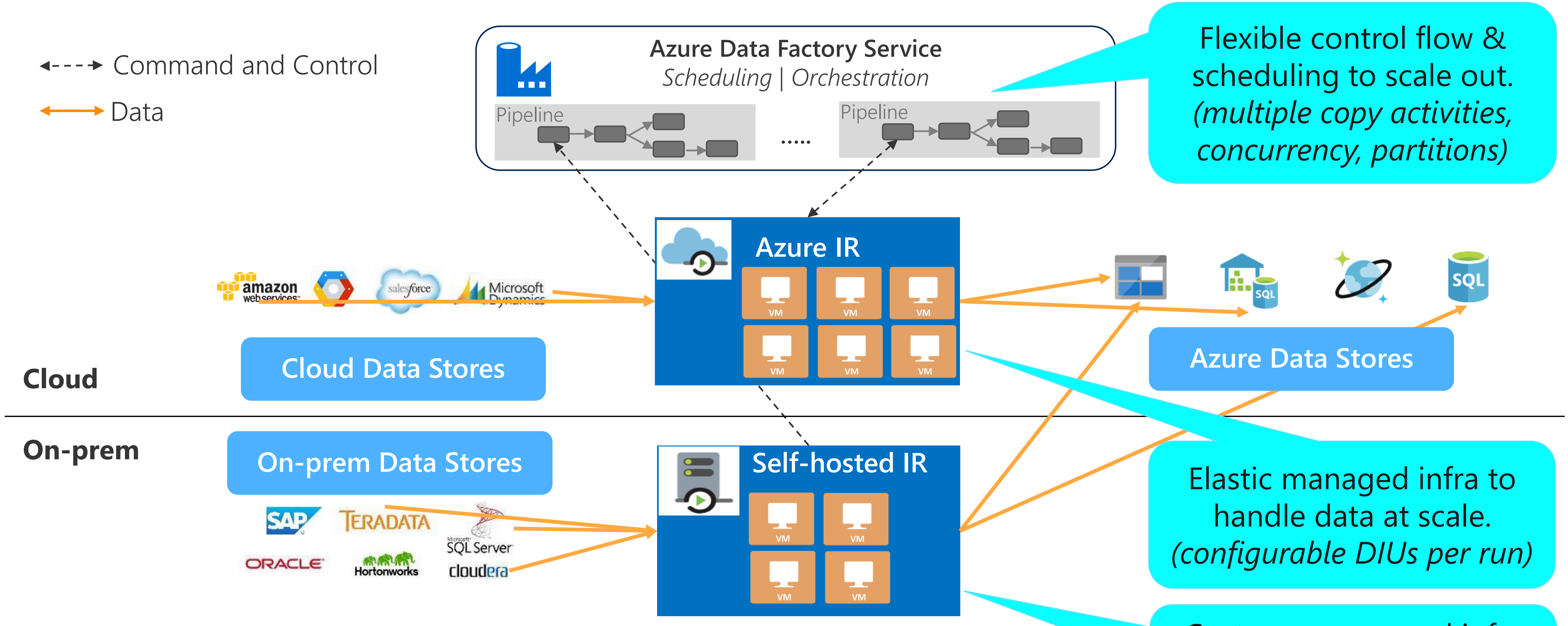
❶ Follow the performance tuning tips, which is auto generated based on the execution status.

❷ Check the execution details to identify bottlenecks and refer to *performance and scalability guide*.

E.g. Queue time – scale IR;
Time to first byte – tweak query;
Read/write – scale out or add parallelism

❸ Provide us feedback on the perf you experienced

Understand how ADF copy scales



Flexible control flow & scheduling to scale out. *(multiple copy activities, concurrency, partitions)*

Elastic managed infra to handle data at scale. *(configurable DIUs per run)*

Customer managed infra with scaling options. *(powerfulness, concurrency)*

Addition: parallelism within each copy activity run

Reference

- Connector overview
- Copy data using Copy Activity
- Performance and scalability guide
- Troubleshoot performance
- Performance features

DATA( ) {
EXPOSED;

Learn with us!

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

@AzureSQL
@AzDataFactory

Microsoft Azure