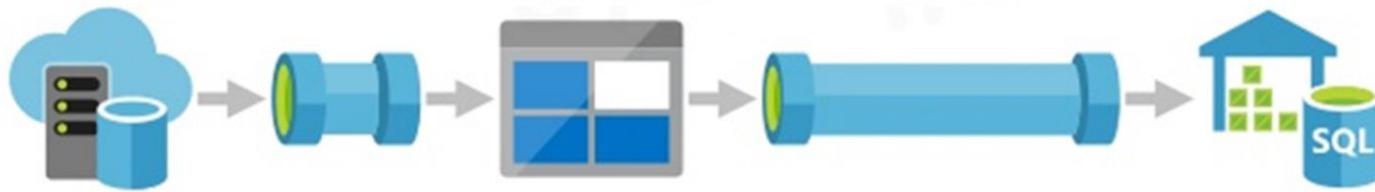




# Azure Data Factory

Cloud version of SSIS

# What can you do in Azure Data Factory?



## Copy Data

## Transform Data

Two PRIMARY TASKS



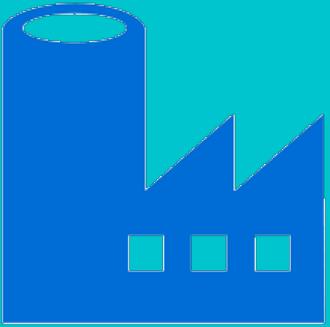
### Copy Data

More than 80 connectors to different services are available



### Transform Data

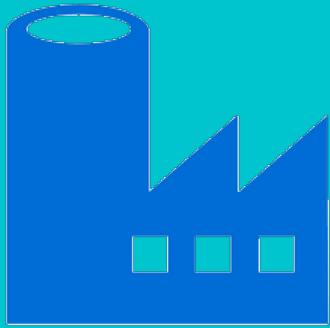
Using newly added Data Flow, now Data Factory is complete cloud based ETL tool.



Azure Data Factory

## Definition:

Azure Data Factory (ADF) is a hybrid data integration service that enables you to quickly and efficiently create automated data pipelines – without having to write any code!



## Azure Data Factory

- Hybrid Data Integration Service
- Simplifies ETL at scale
- Enables modern data integration
- Drag and drop interface
- Over 80 connectors available
- Move, transform and save data
- Managed Service
- Create Data Driver workflows
- Orchestrate and automate data movement
- Transform and store data
- Operationalize the process
- ETL or ELT scenarios

# Data Factory on Azure Ecosystem

01

Migration?

Data Factory excels in periodic data loads and transformation instead.



02

Streaming?

ADF can orchestrate, but there are other dedicated services for streaming



03

Transformations?

Data flows for simple ones, but you can use Databricks or HDInsight for more complex transforms



# SSIS vs Data Factory

## SSIS

More code-free transformations  
On Premises connectors (e.g excel)

## Data Factory

Much higher **scalability**  
Cloud and SaaS Connectors  
**Event based Triggers**  
Can use SSIS Packages

# Data Factory considerations

## Two versions

ADF V2 is the current and improved version

## Build options

PowerShell, .Net, Python, REST, ARM

## Highly integrated

AZURE  
GIT

DevOps, Key Vault, Monitor, Automation

## No data storage

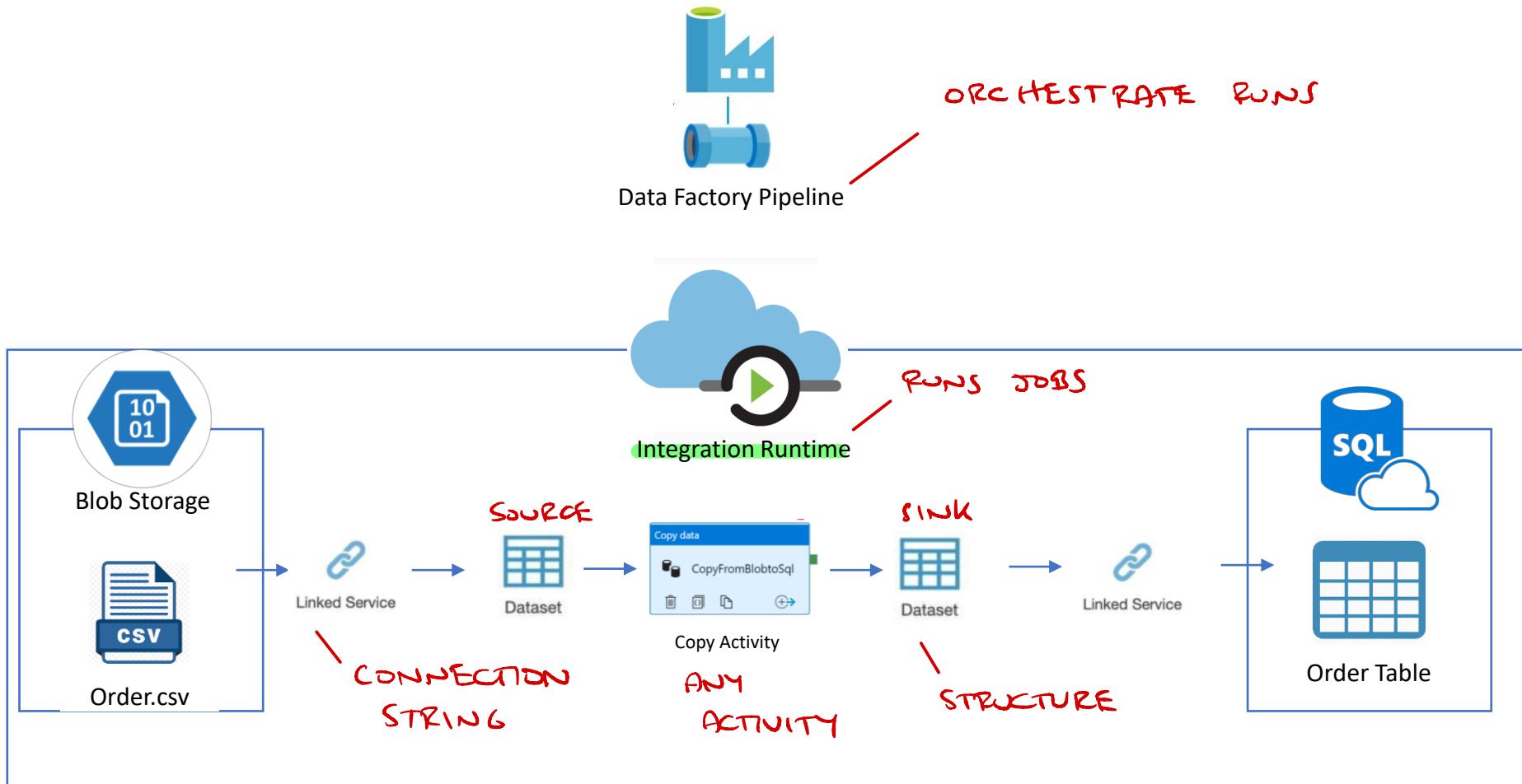
Need to persist data by the end.

## Security standards

HTTP/TLS whenever possible



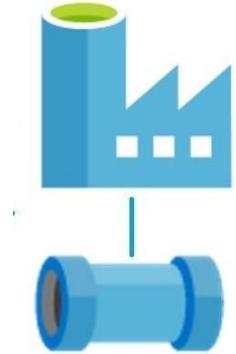
# Azure Data Factory Components



# Data Factory vs SSIS

## COMPARISONS

Azure Data Factory	SSIS
Pipeline	Package
Linked Service	Connection manager
Source	Source
Sink	Destination
Activity	Control flow task
Data Flow	Data flow



Data Factory  
Pipeline

- Data Factories can contain one or more pipelines
- Logical group of Activities
- Manage Activities as a set
- One Pipeline can have one or more activities

## Azure Data Factory Activities

- Represents a processing step in the pipelines
- Actions to perform on data
  - Ingest data
  - Transform data
  - Store data
- Can be linked
  - Execute sequentially or
  - Run in parallel

# Activity types

01

## Data movement activities

Copy data amongst data stores located on-premises and in the cloud

Data stores – Blob storage, Cosmos DB, Amazon Redshift, Google BigQuery Hive, Maria DB...etc.



02

## Data transformation activities

Transform and enrich data

e.g. Hive, Pig, MapReduce, Spark or Databricks



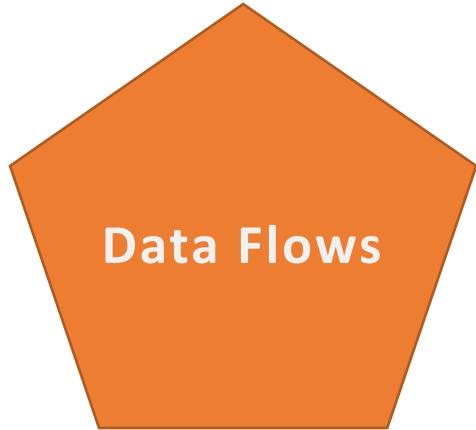
03

## Control activities

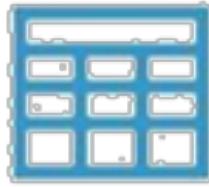
Control pipeline flow

e.g. ForEach, Web





- Data Flow is a new feature of Azure Data Factory (ADF) that allows you to develop graphical data transformation logic that can be executed as activities within ADF pipelines.
- Two types:
  - Mapping **DATA TRANSFORMATION**
  - Wrangling **DATA PREPARATION**



## Dataset

STRUCTURE / SCHEMA  
DOES NOT STORE DATA

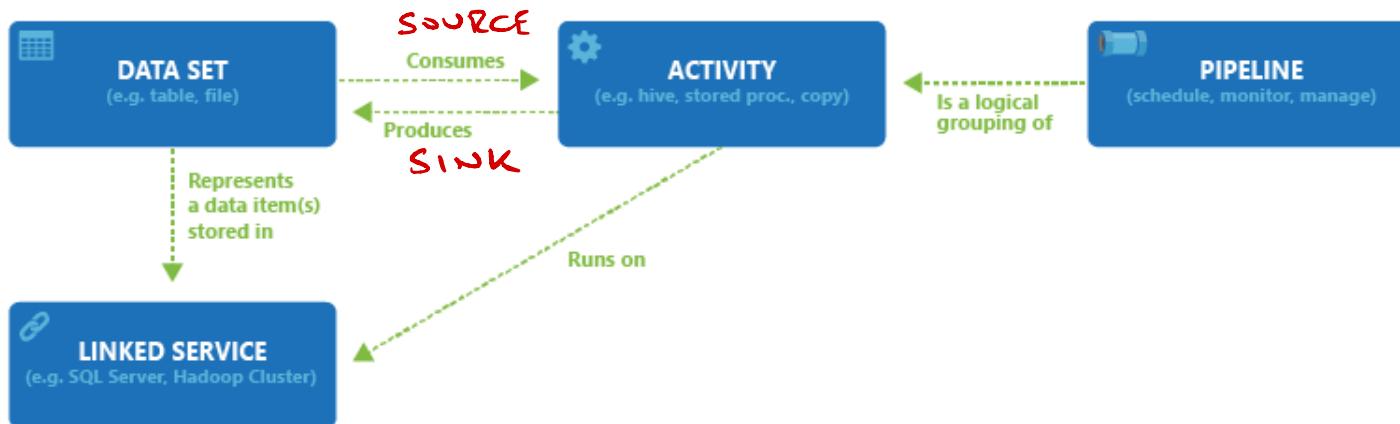
- Simply point or reference the data
- Reference data used in an Activity
  - Files
  - Folders
  - Documents
  - Tables



## Linked service

- Similar to connection string
- Represent the connection information to connect to external resources
  - Datastores like Azure SQL Server
  - Compute resource e.g. Spark Cluster

# ADF Components



## Integration Runtimes

PERFORMS THE ACTION/TASK

- Provides fully managed, serverless compute infrastructure
- You don't have to worry about infrastructure provision, software installation, patching, or capacity scaling.
- Pay only for duration of actual use
- Bridges between the activity and linked service
- Activity defines the action
- Linked service define the location

## Integration Runtimes

### FOUR OPERATIONS PERFORMED BY IR

#### ➤ Data Integration Capabilities

##### 1 ➤ Data Flow

##### 2 ➤ Data Movement

- Format conversion, column mapping, serialization/  
deserialization etc.
- Provides the native `compute` to move data between  
cloud data stores in a secure, reliable, and high-  
performance manner.

##### 3 ➤ Activity dispatch (e.g. Databricks Notebook, HDInsight Hive, pig, spark activity, SP, ADL Analytics U-SQL activity)

##### 4 ➤ SSIS Package execution

# Integration Runtimes

Specify the infrastructure to run activities

## AUTO RESOLVE INTEGRATION RUNTIME

### Azure Integration Runtime

Work on public networks

Responsible for data flows, data movements, and activity dispatches

### Self-hosted Integration Runtime

Work on public and private networks

Provide data movement and activity dispatch capabilities

Need to install on on-premises machine or a virtual machine inside private network

### SSIS Integration Runtime

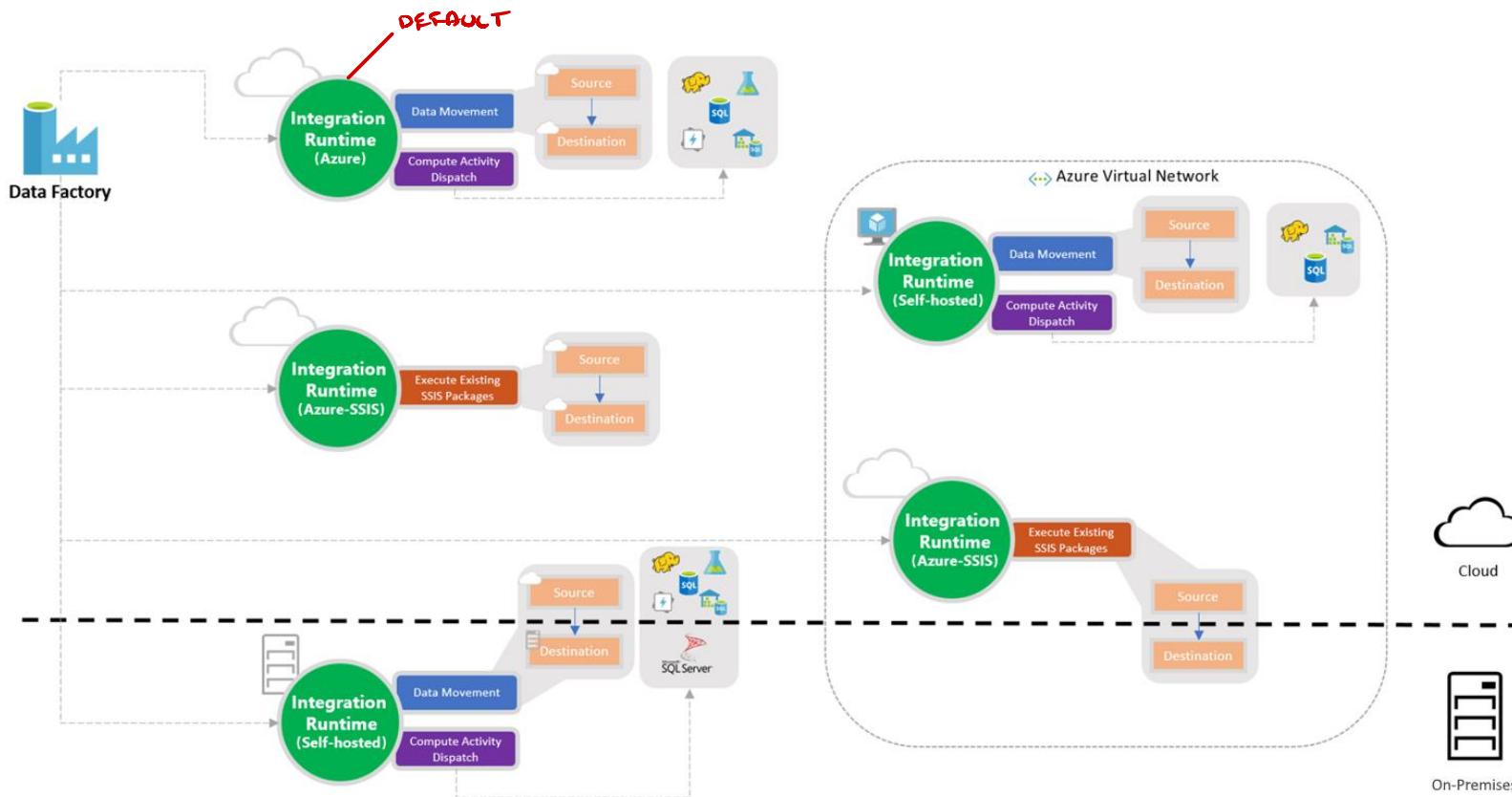
Supports SSIS package execution

Works on public and private networks

# Integration Runtimes

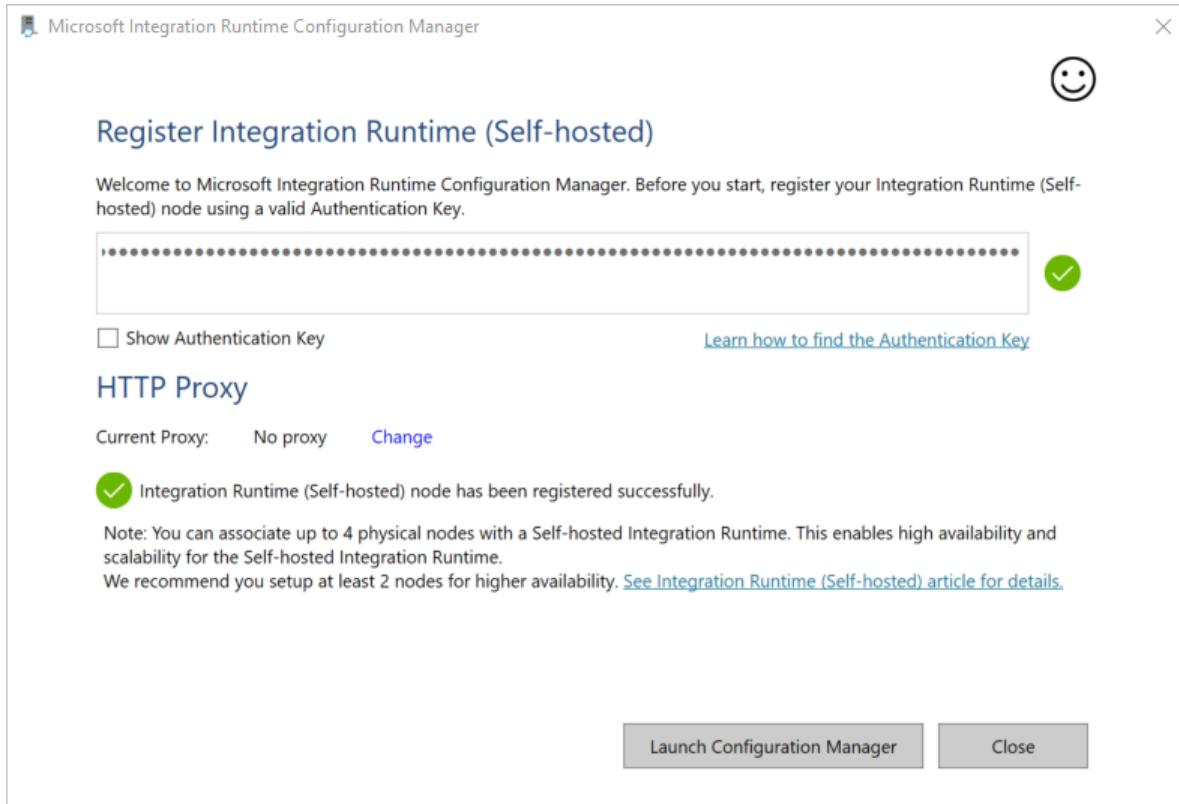
## IR OPERATIONS

IR type	Public network	Private network
Azure	Data Flow Data movement Activity dispatch	
Self-hosted <b>PRIVATE NET</b>	Data movement Activity dispatch	Data movement Activity dispatch
Azure-SSIS	SSIS package execution	SSIS package execution



→ Data Transfer

→ Command Dispatch

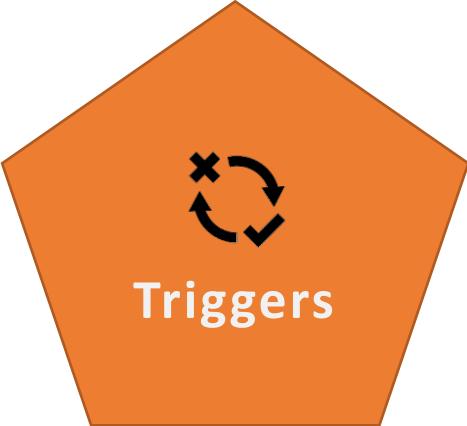


INSTALL IR  
LOCALLY

# Integration Runtimes

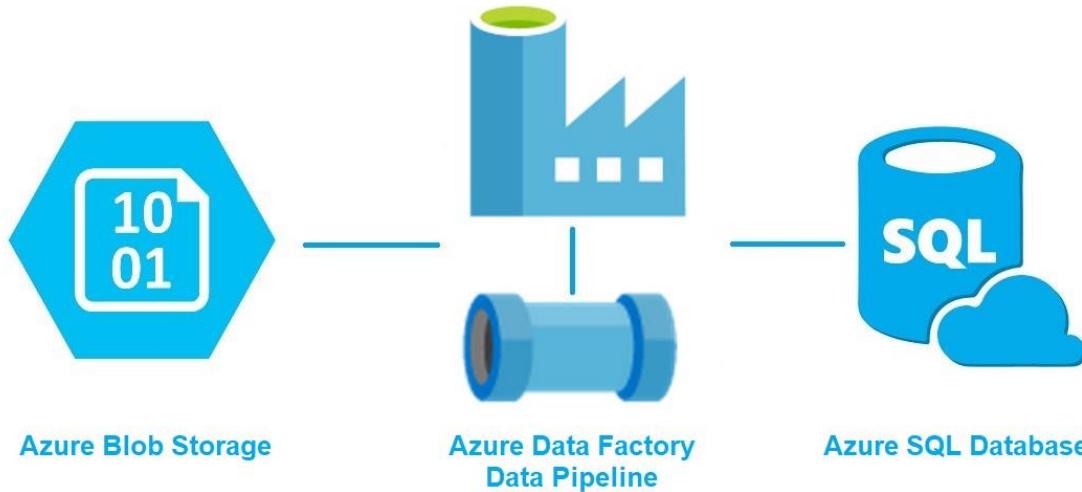
AUTO RESOLVE REFERS  
TO LOCATION

- Default IR – AutoResolveIntegrationRuntime
- Create Azure IR
  - When you want to explicitly define the location of IR
  - Virtually group the activities executions on different IR for management purpose

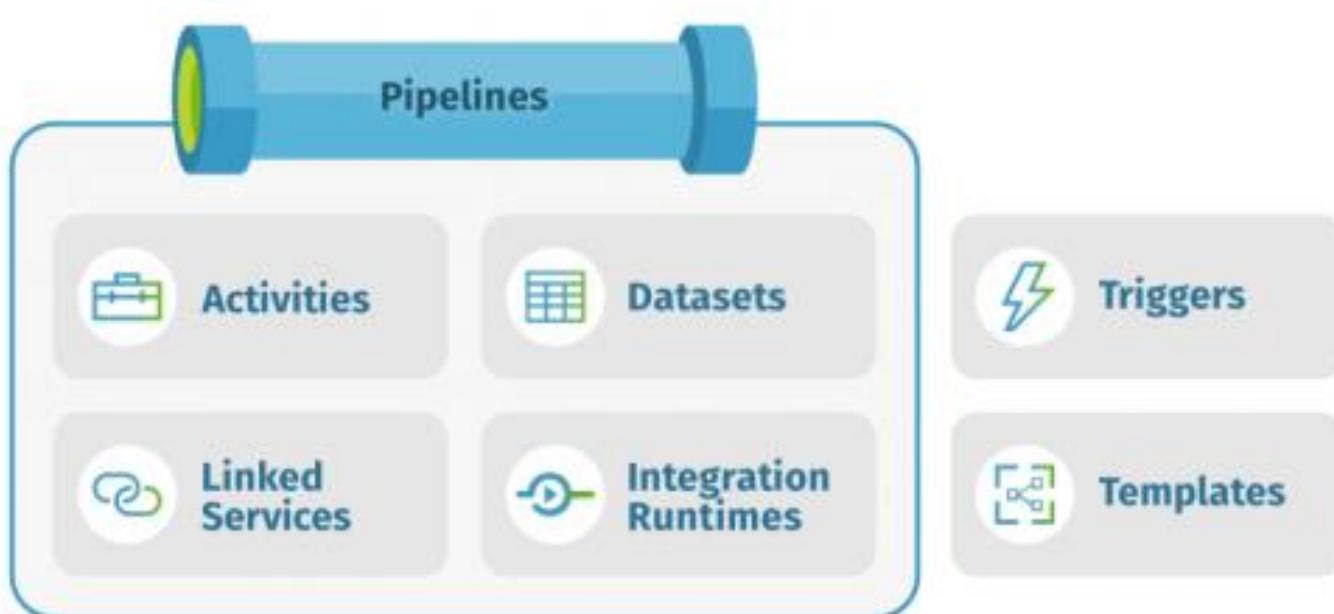


- Execute pipeline
- Many to many relationship b/w pipeline and trigger
- Three types of Trigger CAN ALSO RUN NOW
  - Schedule Trigger – Invoke pipeline on a wall-clock schedule
  - Tumbling Window Trigger – Operates on a periodic interval, also retain state
  - one-to-one relationship
  - Advance configuration options - Dependencies, delay, retry, concurrency
  - Properties - trigger().outputs.WindowStartTime/WindowEndTime
- Event-based Trigger – trigger pipeline in response to an event
  - e.g. Arrival/deletion of file in Blob storage
  - Event trigger with Azure Event Grid Service
  - Properties – triggerBody().FolderPath/fileName

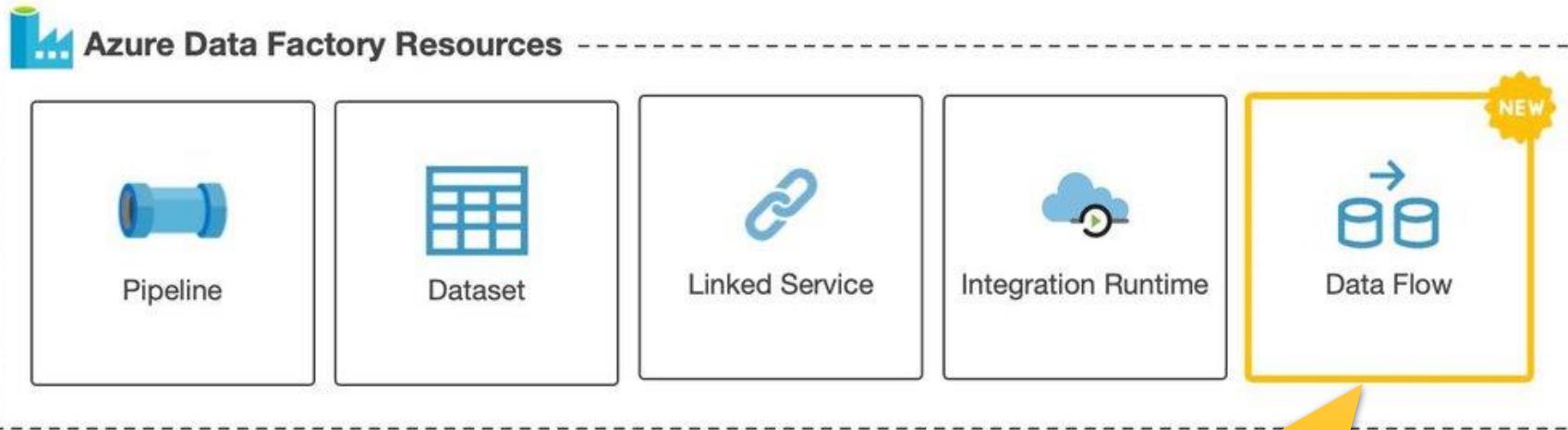
# Demo: Copy Activity



# Summary

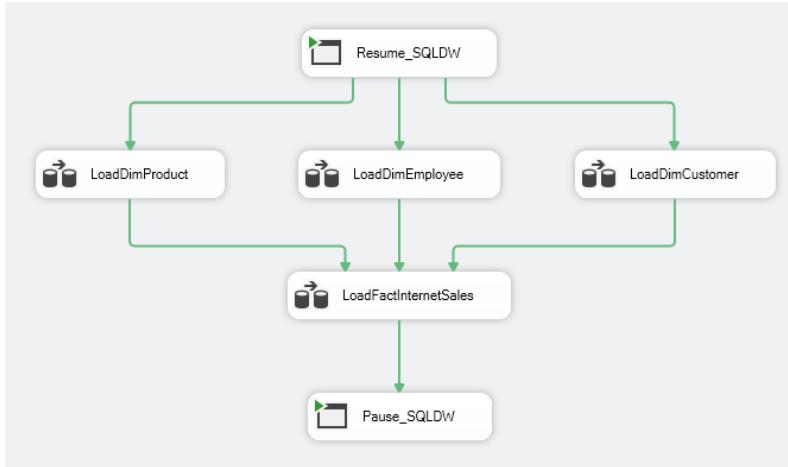


# Data Flows

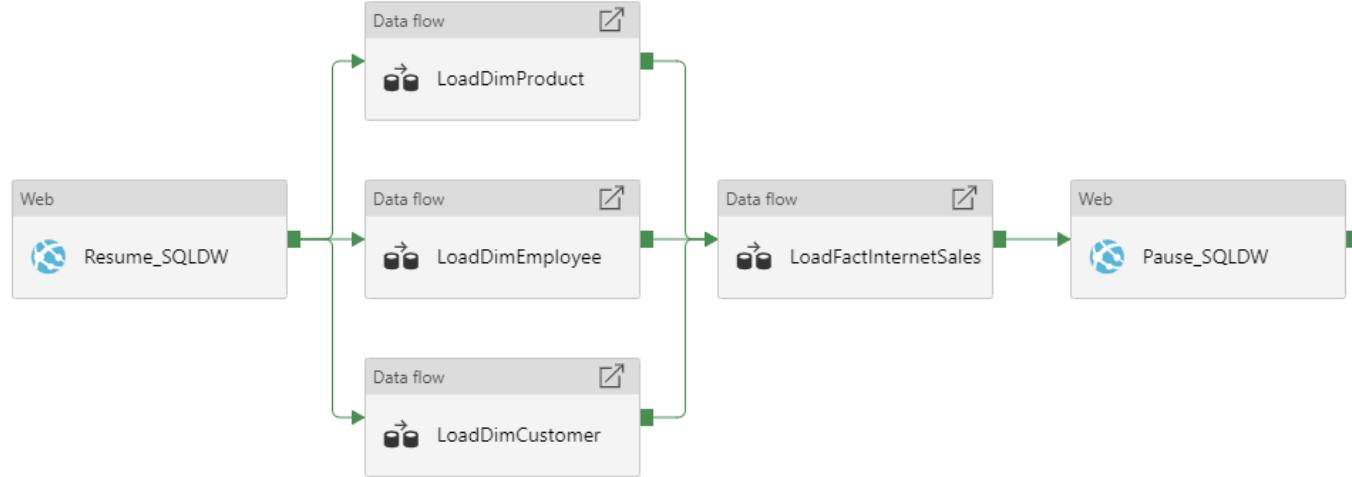


Allows you to develop graphical data transformation logic

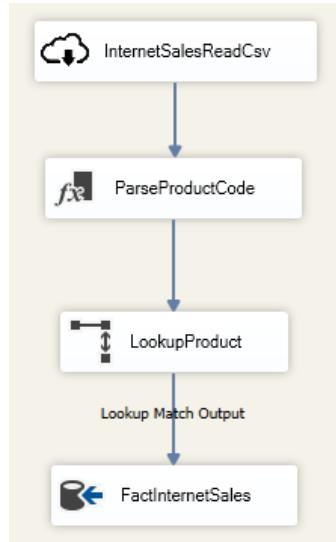
*Example of the SSIS Control Flow tab for loading our data mart tables:*



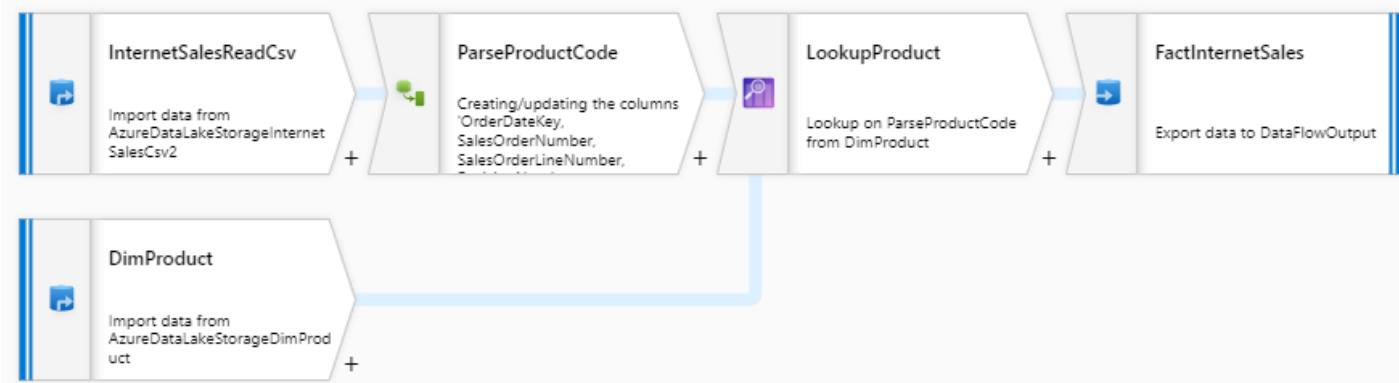
*Example of the ADF Pipeline for loading our data mart tables:*

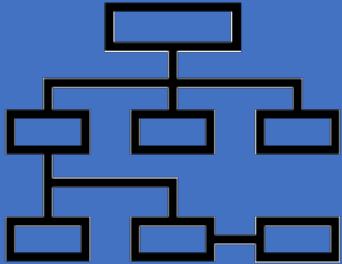


*Example of SSIS Data Flow tab for loading the FactInternetSales table:*



*Example of ADF Mapping Data Flows for loading the FactInternetSales table:*





# Data Flow

Mapping Data flow – Transform Data

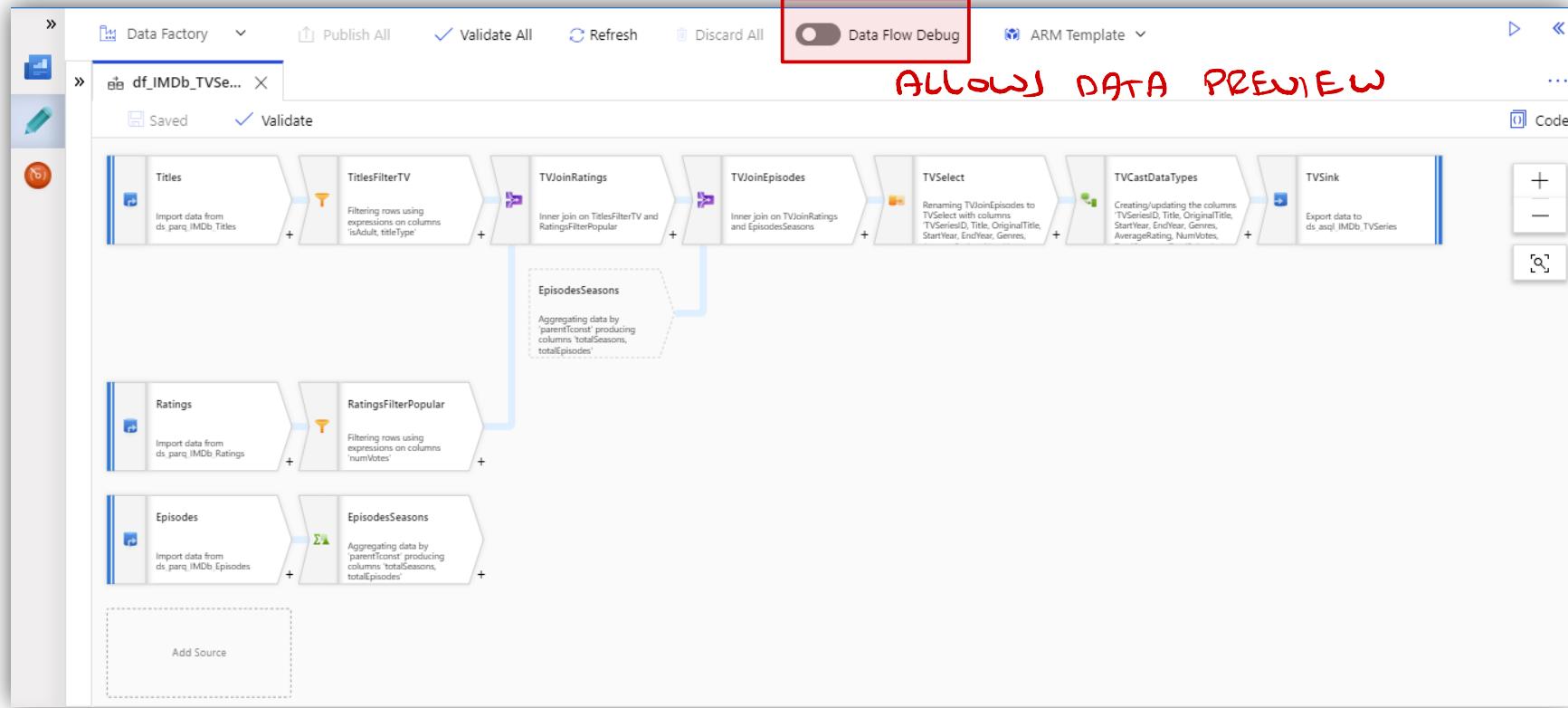
(Known data and schema)

GUI  
DRAG / DROP

Wrangling Data flow – Prepare and explore

data using power query (known or unknown  
datasets)

# Mapping Data Flows



# Mapping Data Flow Actions

## Multiple Inputs/outputs

Join  
Conditional Split  
Exists  
Union  
Lookup

## Schema Modifiers

Derived Columns  
Select  
Aggregate  
Surrogate key  
Pivot  
Unpivot  
Window

## Row Modifiers

Filter  
Sort  
Alter Row

# Wrangling Data Flows

POWER QUERY

Power BI Data Factory

Lego\_Prep\_Leg...

Refresh Options Manage columns Transform table Reduce rows Add column Combine tables Settings Code

ADFResource [2]

UserQuery

Table.RenameColumns(#"Removed columns", {{"set\_num", "SetID"}, {"name", "SetName"}, {"year", "ReleaseYear"}, ...})

	A <sup>B</sup> C SetID	A <sup>B</sup> C SetName	A <sup>B</sup> C ReleaseYear	A <sup>B</sup> C NumParts	A <sup>B</sup> C Theme1	A <sup>B</sup> C Theme2	A <sup>B</sup> C Theme3
1	001-1	Gears	1965	43	Technic	(null)	(null)
2	002-1	4.5V Samsonite Gears Motor Set	1965	3	Technic	(null)	(null)
3	1030-1	TECHNIC I: Simple Machines Set	1985	191	Technic	(null)	(null)
4	1038-1	ERBIE the Robo-Car	1985	120	Technic	(null)	(null)
5	1039-1	Manual Control Set 1	1986	39	Technic	(null)	(null)
6	0016-1	Castle Mini Figures	1978	15	Castle	(null)	(null)
7	10000-1	Guarded Inn	2001	256	Castle	(null)	(null)
8	10039-1	Black Falcon's Fortress	2002	431	Castle	(null)	(null)
9	024119931...	DC Super Heroes: Character Encyclopedia	2016	5	Books	(null)	(null)
10	024135752...	Star Wars: The Visual Dictionary, New Editi...	2019	0	Books	(null)	(null)
11	075666853...	Atlantis: Brickmaster	2010	157	Books	(null)	(null)
12	075667280...	Pirates: Brickmaster	2009	162	Books	(null)	(null)
13	075667281...	Castle: Brickmaster	2009	141	Books	(null)	(null)
14	075668276...	Ninjago Brickmaster	2011	154	Books	(null)	(null)

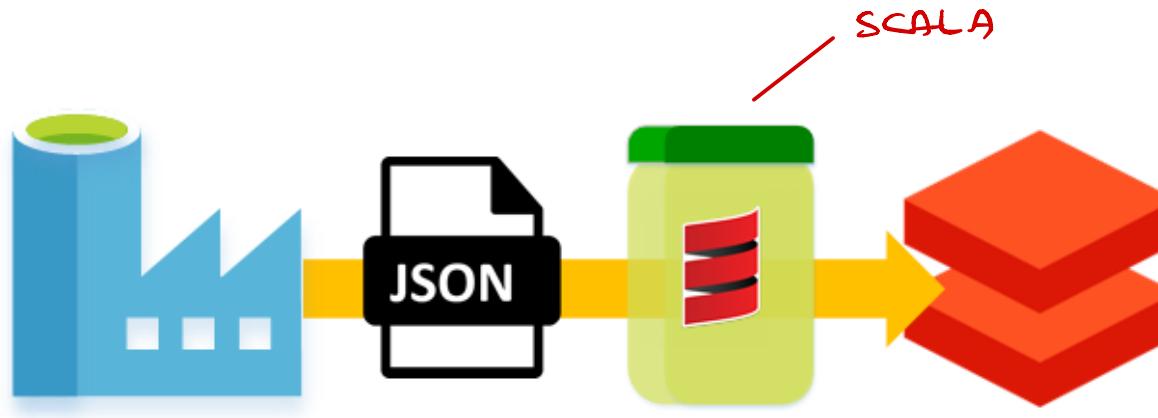
Name: UserQuery

Applied steps:

- Source
- Expanded CSV\_ADLS\_Leg...
- Merged queries
- Expanded CSV\_ADLS\_Leg...
- Merged queries 1
- Expanded CSV\_ADLS\_Leg...
- Removed columns
- Renamed columns

Reset Done

# Data flows behind the scene



Behind the scene Data flow will execute on Azure Databricks using Spark



ADF internally handles all the code translation, spark optimization and execution of transformation



# Spark

Basics

# Why Map reduce introduced



## Tradition system were failing

Centralized server to store and process data  
which creates bottleneck



## Google introduced Map Reduce

Divide the task into small parts and assign  
them to many computers



# Challenges with Map Reduce



## Data-sharing abstraction

Concurrent data access to memory across the cluster



## Inefficient use of resources

Poor memory utilization by spilling to disk after each job

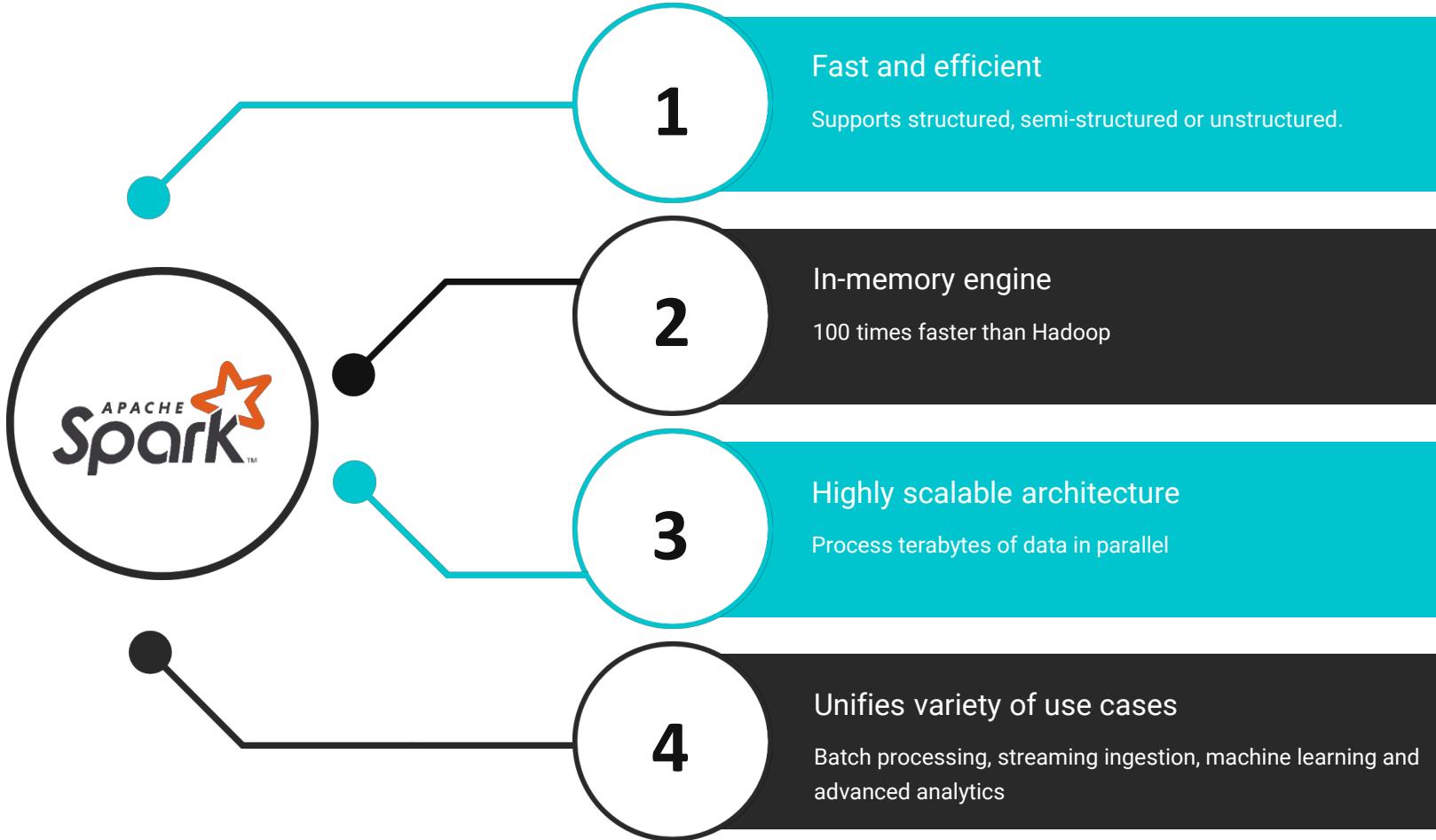




## Apache Spark

Big Data Tool

"Spark is an open source unified analytics engine for large-scale data processing."



# Hadoop

## HDFS

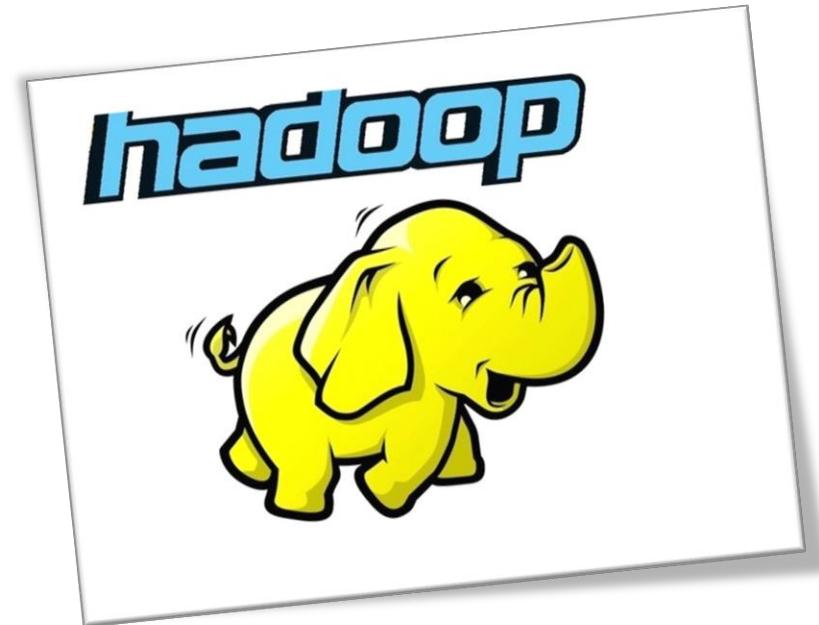
A file system to manage the storage of data

## MapReduce

A framework to define a data processing task

## YARN

A framework to run the data processing task



# Co-ordination between Hadoop Blocks



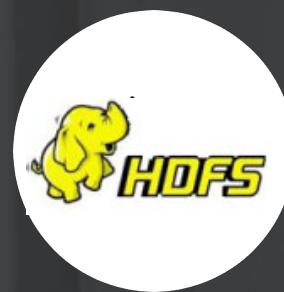
## 1. Step One

User defines map and reduce tasks using the MapReduce API



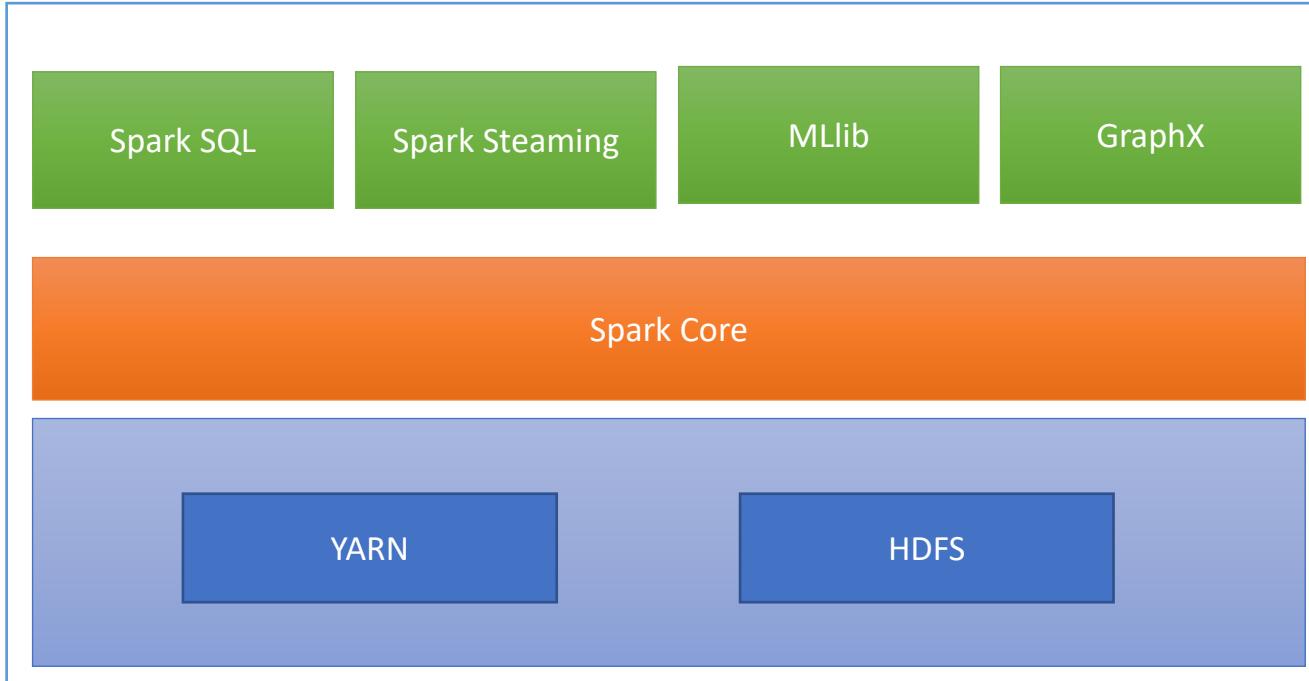
## 2. Step Two

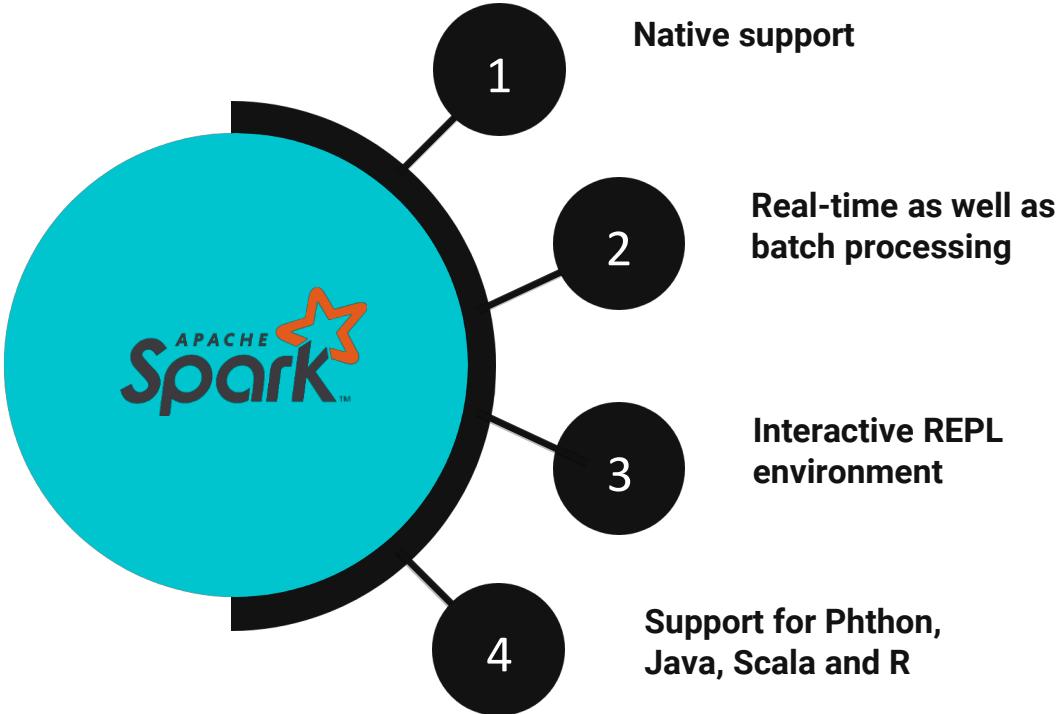
YARN takes care of resource allocation and figures out where and how to run the job



## 3. Step Three

YARN stores the result in HDFS





## RESILIENT DISTRIBUTED DATASET

### RDDs: Basic Building Blocks of Spark



||

RDD are still the fundamental building blocks of Spark

An RDD is a collection of entities – rows, records

# RDDs: Basic Building Blocks of Spark



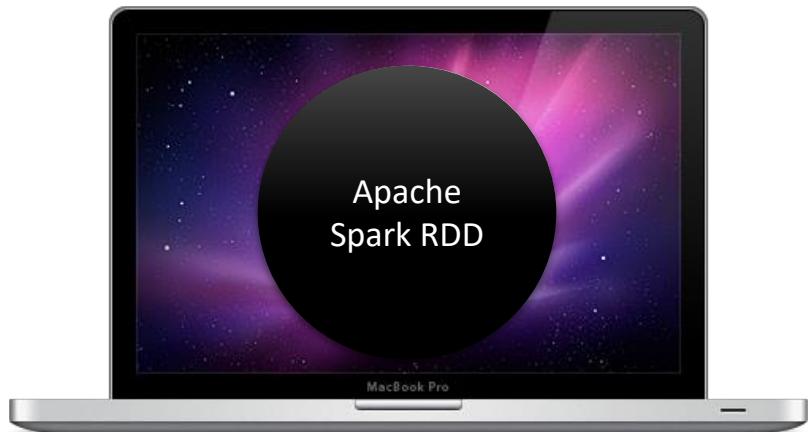
||

All operations in Spark are performed  
on in-memory objects *OBJECTS = RDD*

Two types of methods:

- Action - Return value
- Apply Transformation *CHANGE CONTENT*

# Characteristics of RDDs



01

## Partitioned

Split across data nodes in cluster

02

## Immutable

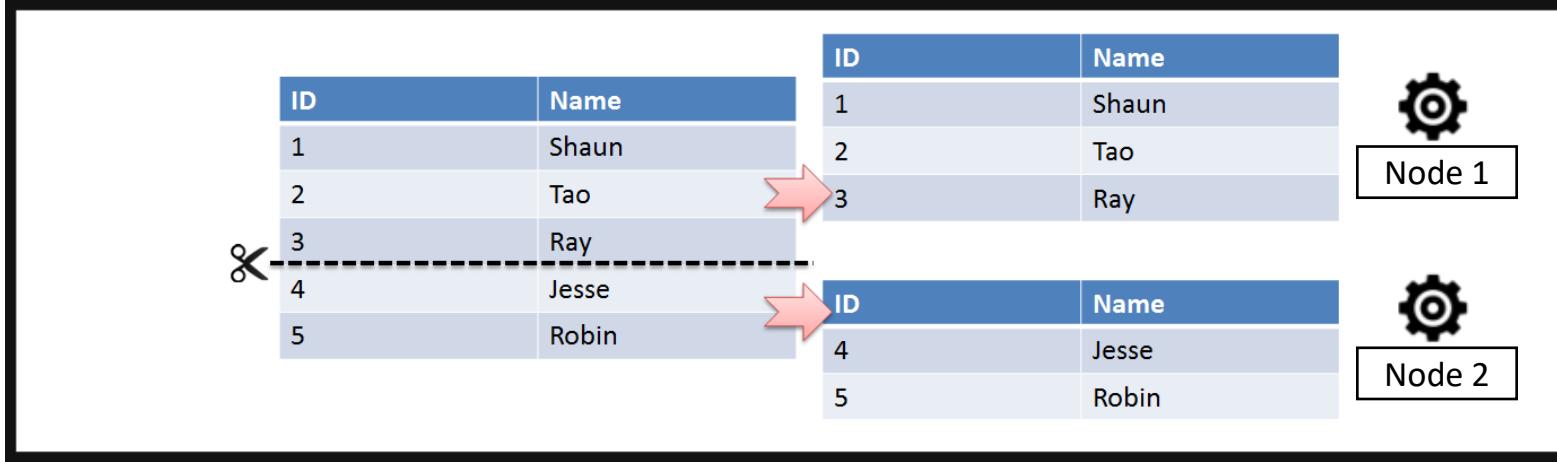
RDDs, once created, cannot be changed

03

## Resilient TO NODE FAILURES

Can be reconstructed even if a node crashes

# Partitioned



## Parallel processing

The main reason for the splitting of data across multiple nodes is parallelization. Data can be processed in parallel on all of these individual nodes.

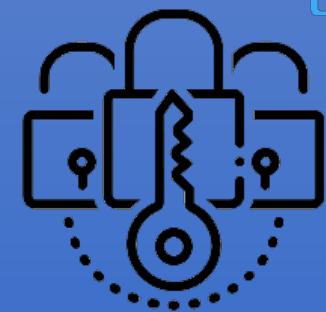


## Data stored in memory for each node in cluster

The most important thing about Spark is that the contents of this RDD are kept entirely in memory across multiple cluster nodes.

- An RDD cannot be mutated
- Only **two types of operations** are permitted on RDD
  - **Transformation:** Transform in to another RDD
  - **Action:** Request a result

\  
CHANGE  
CONTENT



RDD: Immutable

# RDD: Immutable

- A data set loaded in to RDD
- The user may define a chain of transformations on the dataset.

First Name	Last Name	Address	City	Age
Mickey	Mouse	123 Fantasy Way	Anaheim	73
Bat	Man	321 Cavern Ave	Gotham	54
Wonder	Woman	987 Truth Way	Paradise	39
Donald	Duck	555 Quack Street	Mallard	65
Bugs	Bunny	567 Carrot Street	Rascal	58
Wiley	Coyote	999 Acme Way	Canyon	61
Cat	Woman	234 Purrfect Street	Hairball	32
Tweety	Bird	543	Itotltaaw	28

## Example: Transformation

1. Load Data
2. Pick only the 3<sup>rd</sup> column
3. Sort the values

## Example: Action

1. The first 10 rows
2. A count
3. A Sum

Request a result using an action

Transformation is executed only when a result is requested

# RDD: Lazy Evaluation

- Spark keeps a record of the series of transformations requested by the user.
- It groups the transformations in an efficient way when an Action is requested





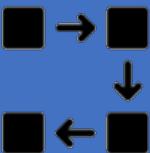
## Lazy Evaluation

Spark keeps a record of the series of transformations requested by the user.

It groups the transformations in an efficient way when an Action is requested.

PROCESSED IN PARALLEL

# Resilient



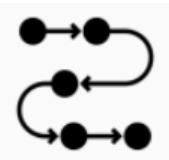
Spark keeps a record of the series of transformations requested by the user

RDD can be created in 2 ways

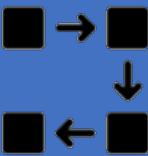
- Reading a file
- Transforming another RDD

Every RDD keeps track of where it came from

RDD'S CAN BE RECONSTRUCTED



# Resilient

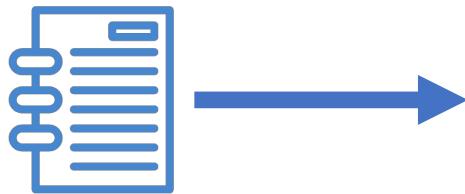


Allows RDDs to be **reconstructed** when nodes crash

Allows RDDs to be Lazily instantiated (**materialized**) when accessing the results

# DataFrame: Data in Rows and Columns

*SIMILAR RELATIONAL DATA*



First Name	Last Name	Address	City	Age
Mickey	Mouse	123 Fantasy Way	Anaheim	73
Bat	Man	321 Cavern Ave	Gotham	54
Wonder	Woman	987 Truth Way	Paradise	39
Donald	Duck	555 Quack Street	Mallard	65
Bugs	Bunny	567 Carrot Street	Rascal	58
Wiley	Coyote	999 Acme Way	Canyon	61
Cat	Woman	234 Purrfect Street	Hairball	32
Tweety	Bird	543	Itotltaw	28

DataFrame

# Apache Spark API



# RDDs to Dataset

## RDDs

Primary abstraction since initial versions

Immutable and distributed

Strong typing, use of Lambda

No optimized execution

Available in all languages

## Datasets

Added to Spark in 1.6

Also immutable and distributed

Also support strong typing, lambdas

Leverage optimizers in recent versions

Present in Scala and Java, not python or R

# Datasets to DataFrames

Datasets	DataFrames
Added to Spark in 1.6	Added to Spark in 1.3
Immutable and distributed	Also immutable and distributed
No named columns	Named columns, like Pandas or R
Extension of DataFrames – OOP interface	Conceptually equal to a table in an RDBMS <i>RELATIONAL TABLE</i>
Compile time type safety	No type safety at compile time
Present in Scala, Java, not Python, R	Available in all languages

Starting Spark 2.0  
APIs for Datasets and DataFrames have merged

# Datasets to DataFrames

## Datasets

Scala and Java

Datasets of the Row() object in Scala/Java  
often called DataFrames

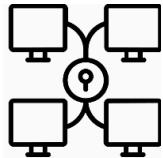
## DataFrames

Python, R, Scala, Java

Equivalent to Dataset<Row> in Java or  
Dataset[Row] in Scala

DATA FRAMES BUILT ON TOP OF RDD'S

# What makes Apache Spark difficult to use?



Infrastructure  
Management



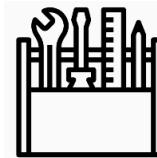
Upgrade Challenge



User  
Interface



Manual  
Configuration



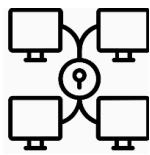
Tooling & Integration  
Complexity



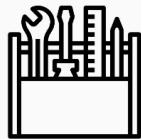
Difficult to  
Collaborate on  
Projects



# databricks®



Efficient and Interactive Platform



Tools are available



Integrated and Interactive workspace



User Interface to manage Infrastructure (Scalability, failure recovery, upgrades)



Distributed processing of data

In-memory

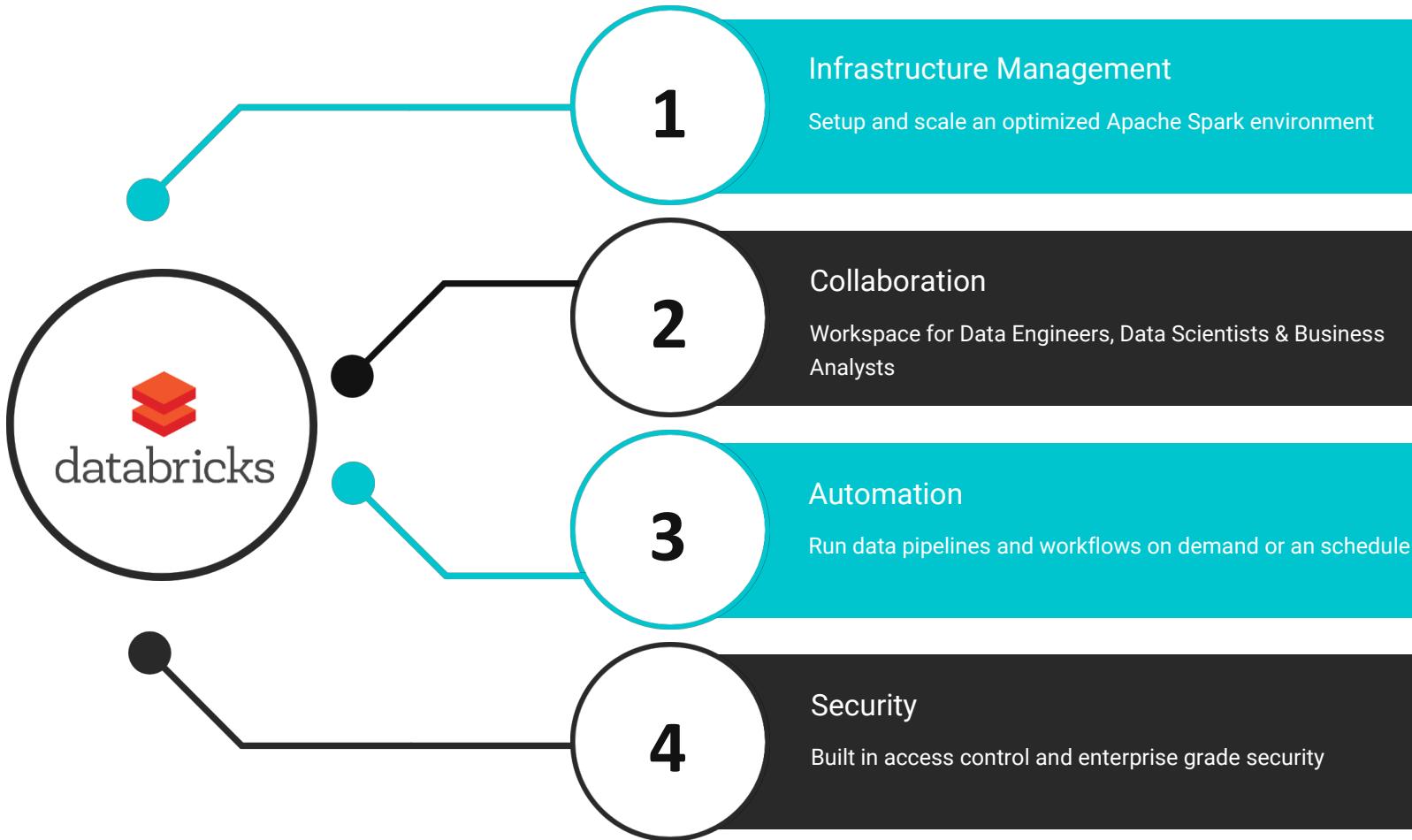
Language support

- Scala, Python, SQL, R & Java

Use cases

- Batch & Stream processing
- Machine learning
- Advanced Analytics

An Apache Spark based Unified  
Analytics Platform, optimized for the  
cloud



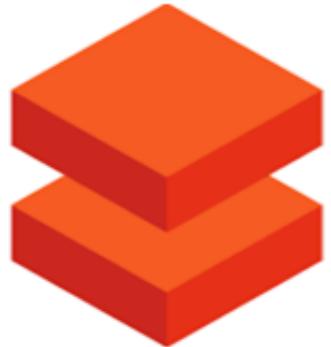
# Microsoft Azure Databricks

A fast, easy, and collaborative Apache Spark™ based analytics platform optimized for Azure



Azure Databricks





Azure Databricks

## Managed 1<sup>st</sup> Party Azure Service

Native integration with Azure & Its services;  
Azure SLA and support

## Transparency

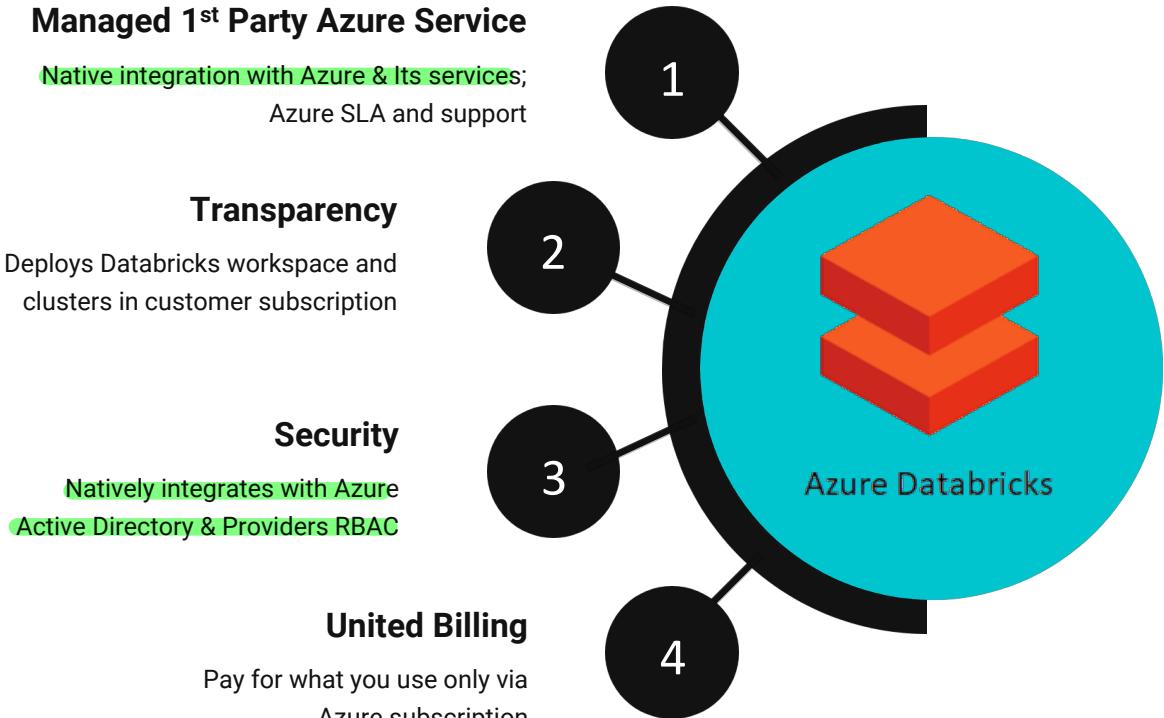
Deploys Databricks workspace and  
clusters in customer subscription

## Security

Natively integrates with Azure  
Active Directory & Providers RBAC

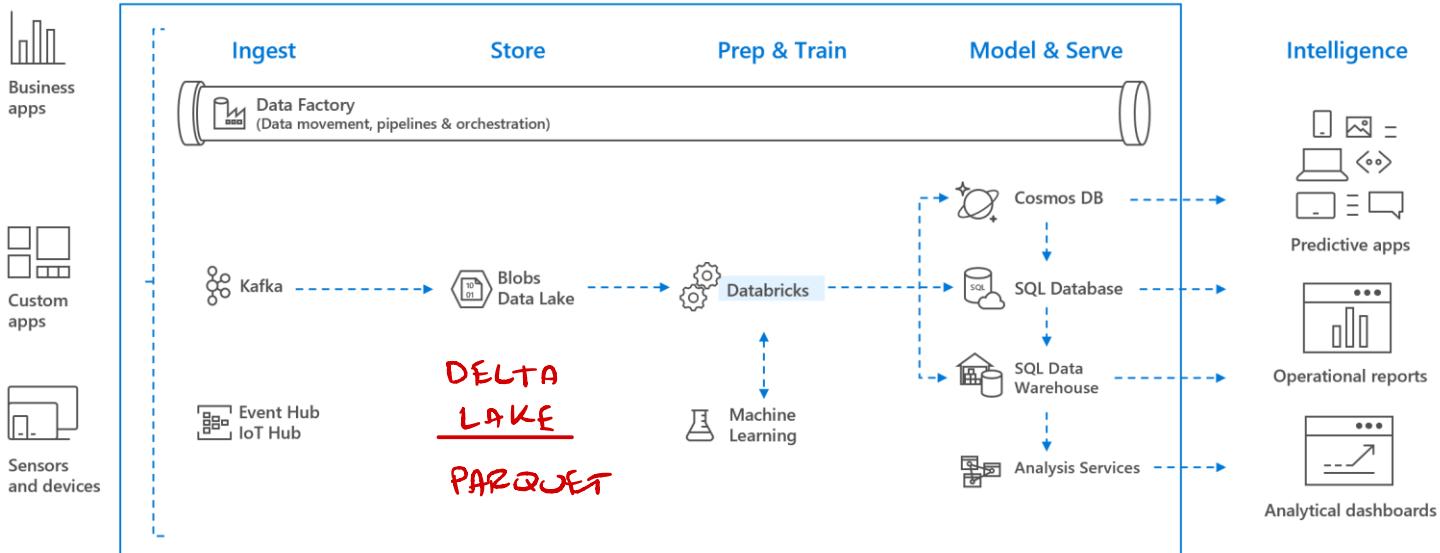
## United Billing

Pay for what you use only via  
Azure subscription



# Azure Databricks Architecture

DB CAN RUN END-TO-END, OR USED IN THE PREP/TRAIN



# Cluster Types



## Interactive Cluster

Multiple users interactively analyze the data together



## Job Cluster (AUTOMATED)

Created and terminated for running automated jobs

# Cluster Types

## Interactive Cluster

Interactively analyze the data

**Created by users**

Manually terminate

Option to auto terminate, if inactive

Low execution time

Auto scale on demand

Comparatively costly

## Job Cluster

Run automated jobs

**Auto created when job starts**

Terminates when the job ends

Option to auto terminate not applicable

High throughput

Auto scale on demand

Comparatively cheaper

RUNNING CLUSTERS WILL ACCUMULATE COMPUTE COSTS

# Cluster ~~Types~~ MODES

## Standard Mode

Single user

No fault isolation

No task preemption

Each user require separate cluster

Supports Scala, Python, SQL, R % Java

## High Concurrency Mode

Multiple users

Fault isolation

Task preemption – fair resource sharing

Maximum cluster utilization

Only supports Python, SQL & R

# Cluster

There are two types of nodes



## Worker Nodes

Multiple nodes perform data processing task

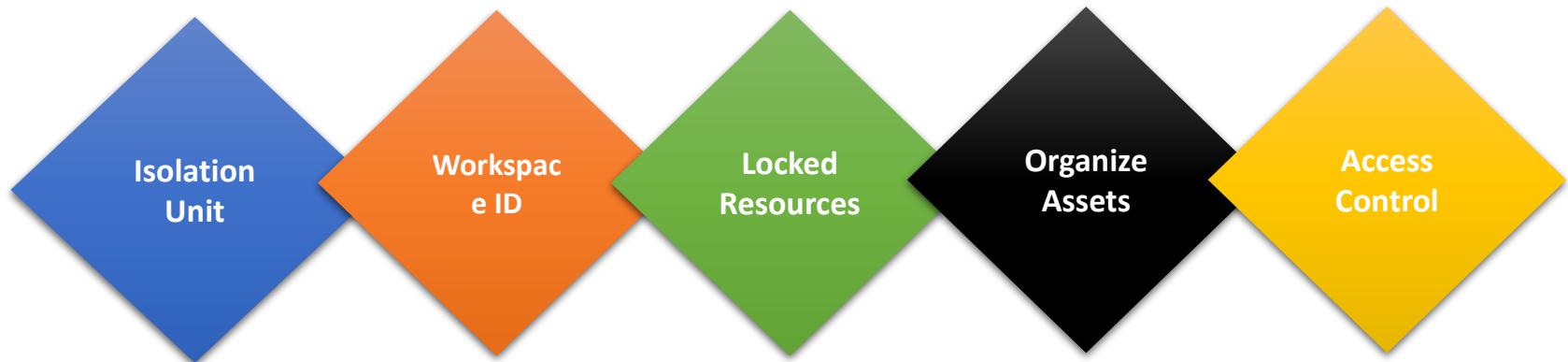


## Driver Node

Distributes task to workers and coordinates execution

MORE NODES → FASTER PROCESSING , more cost

# Workspace



MULTIPLE USERS CAN WORK FROM A WORKSPACE

# Notebooks

Languages

Workflows

Execution

Visualization

Collaboration

Code in any  
Spark supported  
Languages

Invoke notebook  
from others &  
pass data

Run directly on  
clusters or via  
jobs

Turn data into  
graphs or build  
dashboards

Multiple users  
can edit and  
share  
comments

SUPPORTS MULTIPLE LANGUAGES IN A NOTEBOOK

% SQL

% SCALA

% PYTHON



## Jobs

- Execution of a notebook or JAR
- It can run immediately or on schedule
- Create job clusters to run jobs
- Each job can have different cluster configuration
- Monitor job runs and setup alerts

- Install 3<sup>rd</sup> party libraries
- Can be in any supported language
- Import the library into notebook to work
- Scoped at:
  - Cluster
  - Notebook



## Libraries

- Create databases and tables inside them
- Table:
  - Collection of structured data
  - Equivalent to DataFrame – perform same operations on table
  - Created using files lying on storage
  - Directly query or write to tables



## Database & Tables