



DP-200T01: Orchestrating Data Movement with Azure Data Factory



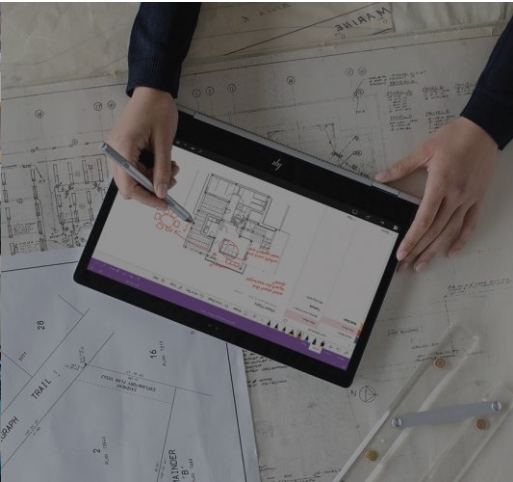
Agenda

- L01 - Introduction to Azure Data Factory
- L02 - Understand Azure Data Factory components
- L03 - Integrate Azure Data Factory with Databricks



Lesson 01

Introduction to Azure Data Factory



Lesson Objectives

- What is Azure Data Factory
- The Data Factory process
- Azure Data Factory components
- Azure Data Factory security

What is Azure Data Factory



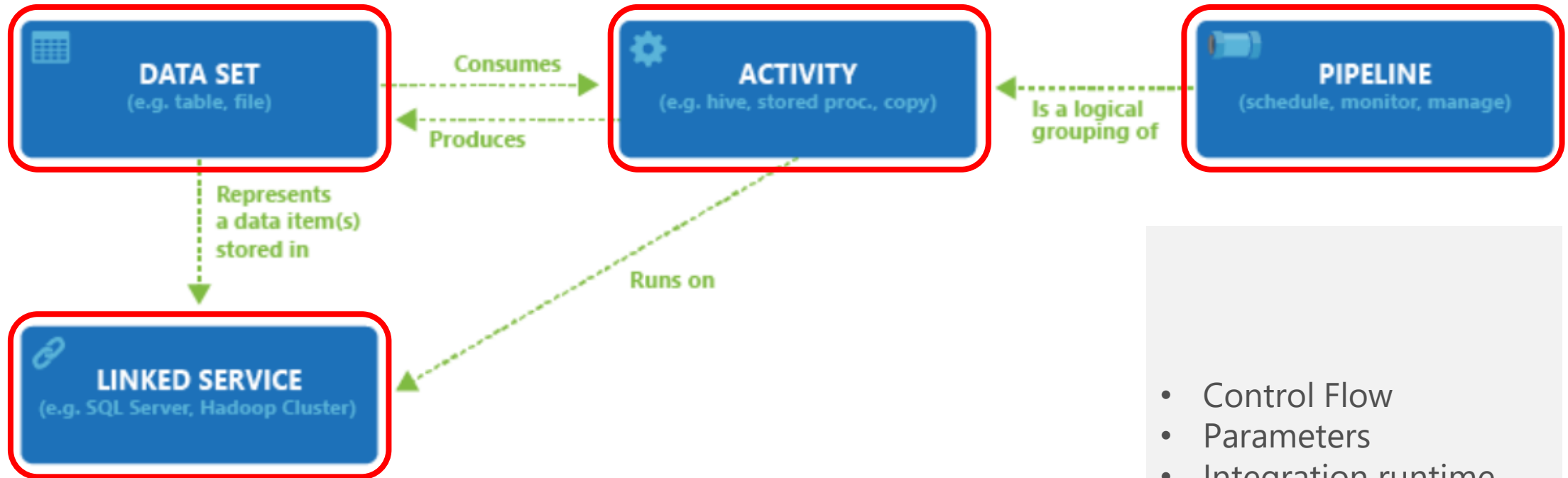
Creates, orchestrates, and automates the movement, transformation and/or analysis of data through in the cloud.



The Data Factory Process



Azure Data Factory Components



Azure Data Factory Security

Data Factory Contributor Role

1. Create, edit, and delete data factories and child resources including datasets, linked services, pipelines, triggers, and integration runtimes.
2. Deploy Resource Manager templates. Resource Manager deployment is the deployment method used by Data Factory in the Azure portal.
3. Manage App Insights alerts for a data factory.
4. At the resource group level or above, lets users deploy Resource Manager template.
5. Create support tickets.

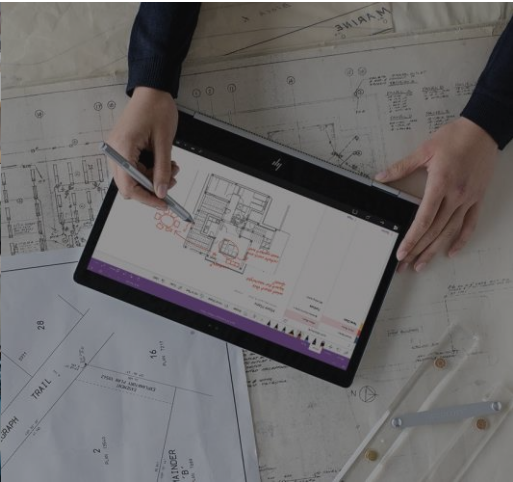
Review Questions

- Q01 – Which Azure Data Factory process involves using compute services to produce data to feed production environments with cleansed data?
- A01 – Transform and enrich
- Q02 – Which Azure Data Factory component contains the transformation logic or the analysis commands of the Azure Data Factory's work?
- A02 – Activities



Lesson 02

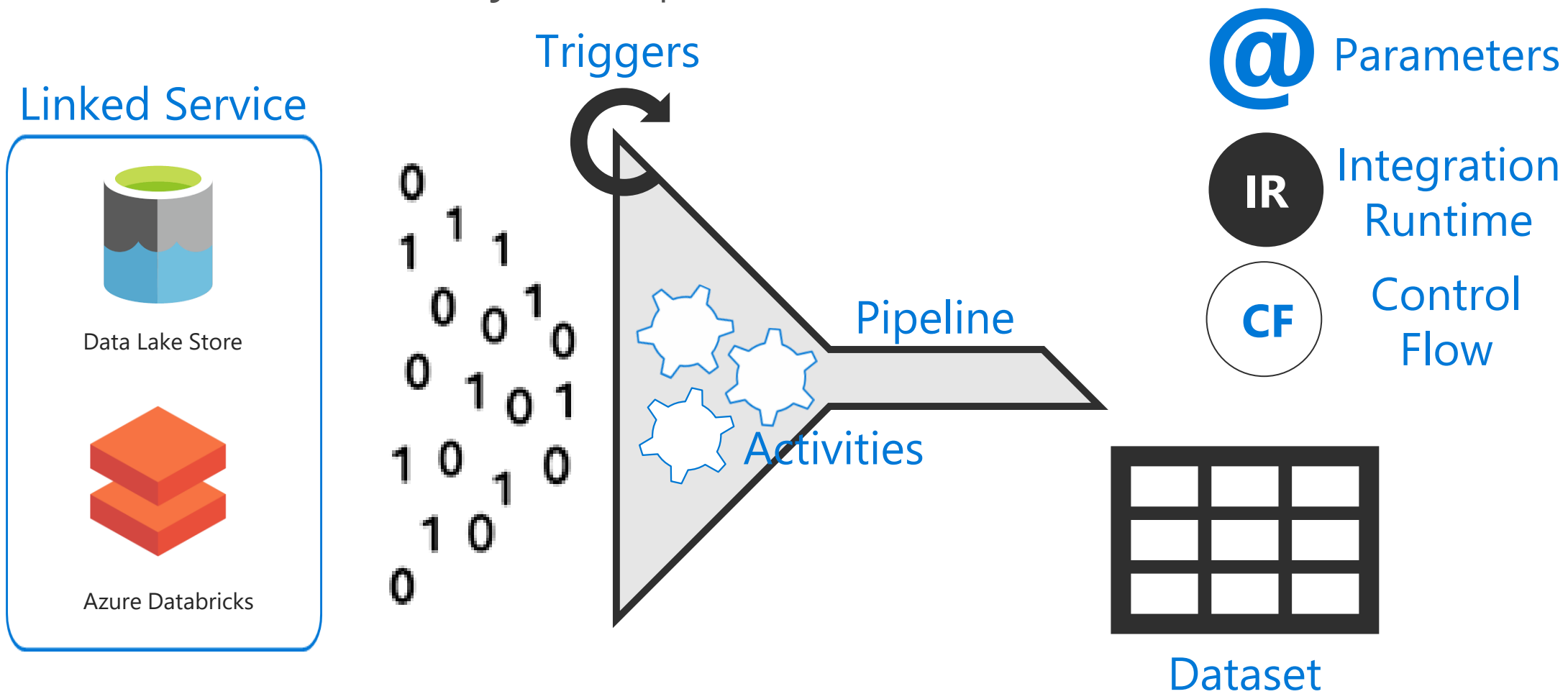
Azure Data Factory Components



Lesson Objectives

- Linked Services
- Datasets
- Data Factory activities
- Pipelines
- Pipeline example

Azure Data Factory components



Linked Services

AZURE SQL DATABASE EXAMPLE

```
{  
  "name": "AzureSqlLinkedService",  
  "properties": {  
    "type": "AzureSqlDatabase",  
    "typeProperties": {  
      "connectionString": "Server=tcp:ctosqldb.database.windows.net,1433;Database=EquityDB;User ID=ctestoneill;Password=P@ssw0rd;Trusted_Connection=False;Encrypt=True;Connection Timeout=30"  
    }  
  }  
}
```

AZURE BLOB STORE EXAMPLE

```
{  
  "name": "StorageLinkedService",  
  "properties": {  
    "type": "AzureStorage",  
    "typeProperties": {  
      "connectionString":  
"DefaultEndpointsProtocol=https;AccountName=ctostorageaccount;AccountKey=087ubp097guhB*(97g9879"  
    }  
  }  
}
```


Datasets

Dataset name

Properties

Type

External

LinkedServiceName

Structure

Name

Type

Availability

Policy

```
{
  "name": "<name of dataset>",
  "properties": {
    "type": "<type of dataset: AzureBlob, AzureSql etc...>",
    "external": "<boolean flag to indicate external data. only for input datasets>",
    "linkedServiceName": "<Name of the linked service that refers to a data store.>",
    "structure": [
      {
        "name": "<Name of the column>",
        "type": "<Name of the type>"
      },
      {
        "name": "AzureSqlLinkedService",
        "type": "StorageLinkedService"
      }
    ],
    "typeProperties": {
      "<type specific property>": "<value>",
      "<type specific property 2>": "<value 2>",
    },
    "availability": {
      "frequency": "<Specifies the time unit for data slice production. >",
      "interval": "<Specifies the interval within the defined frequency.>"
    },
    "policy": {
      {
        }
      }
    }
  }
}
```

Time Slicing Data

The diagram illustrates the mapping of JSON properties between two configurations: **AzureBlobInput** and **AzureBlobOutput**. The **AzureBlobInput** configuration on the left includes properties for **Availability**, **Offset**, and **Style**. The **AzureBlobOutput** configuration on the right includes properties for **Availability** and **AnchorDateTime**. Blue arrows indicate the mapping from the input properties to the output properties.

```
{
  "name": "AzureBlobInput",
  "properties": {
    "published": false,
    "type": "AzureBlob",
    "linkedServiceName": "StorageLinkedService",
    "typeProperties": {
      "fileName": "input.log",
      "folderPath": "datacontainer/inputdata",
      "format": {
        "type": "TextFormat",
        "columnDelimiter": ","
      }
    },
    "availability": {
      "frequency": "Day",
      "interval": "1",
      "offset": "06:00:00"
    },
    "external": true,
    "policy": {}
  }
}, {
  "availability": {
    "frequency": "Day",
    "interval": 1,
    "offset": "06:00:00",
    "style": "EndOfInterval"
  }
}, {
  "name": "AzureBlobOutput",
  "properties": {
    "published": false,
    "type": "AzureBlob",
    "linkedServiceName": "AzureStorageLinkedService",
    "typeProperties": {
      "folderPath": "datacontainer/partitioneddata",
      "format": {
        "type": "TextFormat",
        "columnDelimiter": ","
      }
    },
    "availability": {
      "frequency": "Hour",
      "interval": "1",
      "offset": "00:00:00",
      "style": "EndOfInterval"
    },
    "anchorDateTime": "2007-04-19T08:00:00"
  }
}
```

Data Factory Activities

Activities within Azure Data Factory defines the actions that will be performed on the data and there are three categories including:

Data movement activities

Data movement activities simply move data from one data store to another. A common example of this is in using the Copy Activity.

Data transformation activities

Data transformation activities use compute resource to change or enhance data through transformation, or it can call a compute resource to perform an analysis of the data.

Control Activities

Control flow orchestrate pipeline activities that includes chaining activities in a sequence, branching, defining parameters at the pipeline level, and passing arguments while invoking the pipeline on-demand or from a trigger

Pipelines

Pipeline is a grouping of logically related **activities**.

Pipeline can be **scheduled** so the activities within it get **executed**.

Pipeline can be **managed** and **monitored**.

Pipeline Example

Data transformation activity

Hive

Pig

MapReduce

Hadoop Streaming

Machine Learning activities: Batch Execution and Update Resource

Stored Procedure

DotNet

Compute environment

HDInsight [Hadoop]

HDInsight [Hadoop]

HDInsight [Hadoop]

HDInsight [Hadoop]

Azure VM

Azure SQL, Azure SQL DW,
or SQL Server

HDInsight [Hadoop] or
Azure Batch

```
{
  "name": "MyFirstPipeline",
  "properties": {
    "description": "My first Azure Data Factory pipeline",
    "activities": [
      {
        "type": "HDInsightHive",
        "typeProperties": {
          "scriptPath": "adfgetstarted/script/partitionweblogs.hql",
          "scriptLinkedService": "StorageLinkedService",
          "defines": {
            "inputtable": "wasb://adfgetstarted@ctostorageaccount.blob.core.windows.net/inputdata",
            "partitionedtable": "wasb://adfgetstarted@ctostorageaccount.blob.core.windows.net/partitioneddata"
          }
        },
        "inputs": [
          {
            "name": "AzureBlobInput"
          }
        ],
        "outputs": [
          {
            "name": "AzureBlobOutput"
          }
        ],
        "policy": {
          "concurrency": 1,
          "retry": 3
        },
        "scheduler": {
          "frequency": "Month",
          "interval": 1
        },
        "name": "RunSampleHiveActivity",
        "linkedServiceName": "HDInsightOnDemandLinkedService"
      }
    ],
    "start": "2016-04-01T00:00:00Z",
    "end": "2016-04-02T00:00:00Z",
    "isPaused": false,
    "hubName": "ctogetstartedddf_hub",
    "pipelineMode": "Scheduled"
  }
}
```

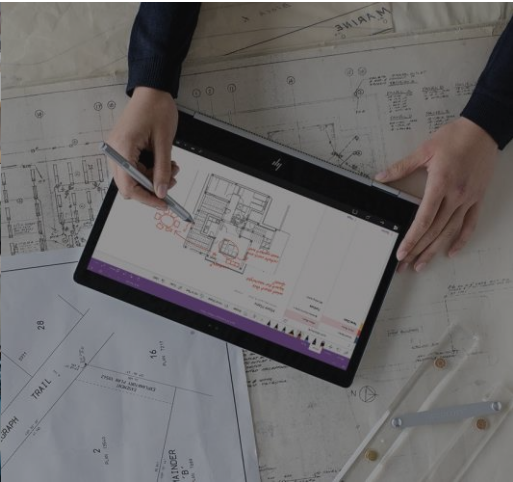

Review Questions

- Q01 - A pipeline JSON definition is embedded into an Activity JSON definition. True or False?
- A01 – False



Lesson 03

Ingesting and Transforming data



Lesson Objectives

- How to setup Azure Data Factory
- Ingest data using the Copy Activity
- Transforming data with the Mapping Data Flow

Create Azure Data Factory

[Home](#) > [New](#) > [Data Factory](#) > [New data factory](#)

New data factory

Name *

Version ⓘ

V2

Subscription *

chtestao

Resource Group *

Select existing...

[Create new](#)

Location * ⓘ

South Central US

Enable GIT ⓘ

☒

GIT URL * ⓘ

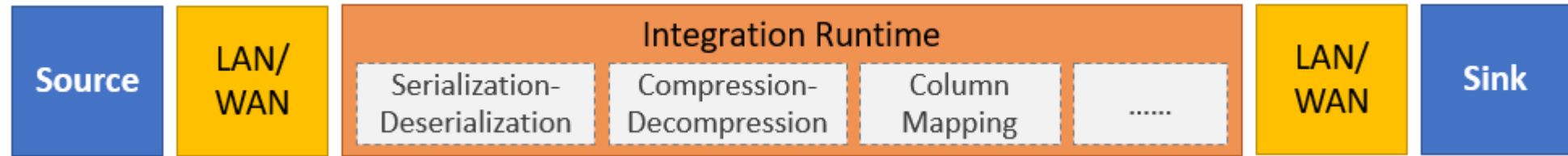
Repo name * ⓘ

Branch Name * ⓘ

Root folder * ⓘ

Create

Ingesting data with the Copy Activity

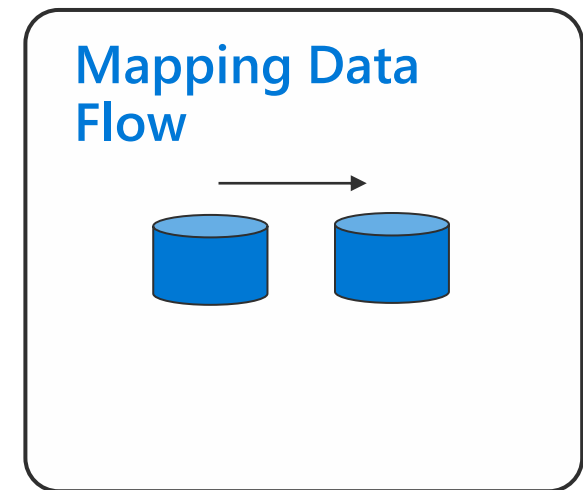


- Reads data from a source data store.
- Performs serialization/deserialization, compression/decompression, column mapping, and so on. It performs these operations based on the configuration of the input dataset, output dataset, and Copy activity.
- Writes data to the sink/destination data store

Transforming data with the Mapping Data Flow

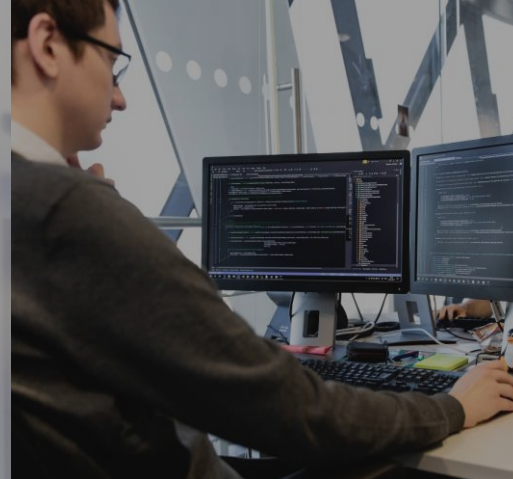
Code free data transformation at scale

- Perform data cleansing, transformation, aggregations, etc.
- Enables you to build resilient data flows in a code free environment
- Enable you to focus on building business logic and data transformation
- Underlying infrastructure is provisioned automatically with cloud scale via Spark execution



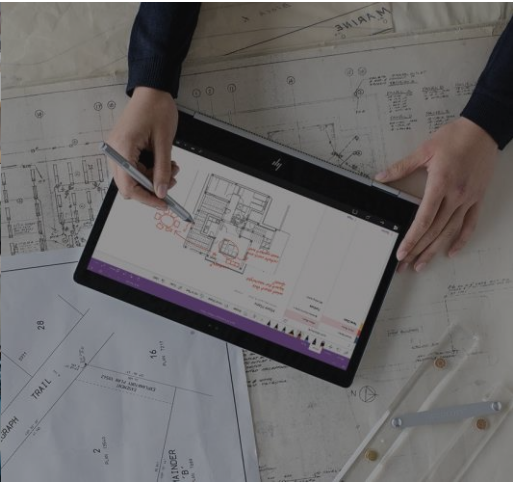
Review Questions

- Q01 - Which transformation in the Mapping Data Flow is used to route data rows to different streams based on matching conditions?
- A01 – Conditional Split
- Q02 - Which transformation is used to load data into a data store or compute resource?
- A02 – Sink



Lesson 04

Integrate Azure Data Factory with Azure Databricks



Lesson Objectives

- Use Azure Data Factory (ADF) to ingest data and create an ADF pipeline.
- Create Azure Storage account and the Azure Data Factory instance
- Use ADF to orchestrate data transformations using a Databricks Notebook activity.

Working with documents programmatically

1.
Create Storage
Account

2.
Create Azure
Data Factory

3.
Create data
workflow
pipeline

4.
Add Databricks
Workbook to
pipeline

5.
Perform analysis
on the data

Create Azure Storage account and the Azure Data Factory Instance

Home > New > Storage account > Create storage account

Create storage account

Basics

Advanced

Tags

Review + create

Azure Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, scalable, and redundant. Azure Storage includes Azure Blobs (objects), Azure Data Lake Storage Gen2, Azure Files, Azure Queues, and Azure Tables. The cost of your storage account depends on the usage and the options you choose below. [Learn more](#)

PROJECT DETAILS

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

* Subscription

* Resource group

Create new

INSTANCE DETAILS

The default deployment model is Resource Manager, which supports the latest Azure features. You may choose to deploy using the classic deployment model instead. [Choose classic deployment model](#)

* Storage account name ⓘ

* Location

West Europe

Performance ⓘ

Standard

Premium

Account kind ⓘ

StorageV2 (general purpose v2)

Replication ⓘ

Read-access geo-redundant storage (RA-GRS)

Access tier (default) ⓘ

Cool

Hot

Review + create

Previous

Next : Advanced >

Home > New > Data Factory > New data factory

New data factory

Name *

Version ⓘ

V2

Subscription *

chtestao

Resource Group *

Select existing...

Create new

Location * ⓘ

South Central US

Enable GIT ⓘ

☒

GIT URL * ⓘ

Repo name * ⓘ

Branch Name * ⓘ

Root folder * ⓘ

Create

Use ADF to orchestrate data transformations using a Databricks Notebook activity

Microsoft Azure

03-Data-Transformation (Python)

Detached File View: Code Permissions Run All Clear

Cmd 1

Data Transformation via Azure Data Factory

As you saw at the end of the previous lesson, different cities use different field names and values to indicate crimes, dates, etc. within their crime data.

For example:

- Some cities use the value "HOMICIDE", "CRIMINAL HOMICIDE" or "MURDER".
- In the New York data, the column is named `offenseDescription` while in the Boston data, the column is named `OFFENSE_CODE_GROUP`.
- In the New York data, the date of the event is in the `reportDate`, while in the Boston data, there is a single column named `MONTH`.

In the case of New York and Boston, here are the unique characteristics of each data set:

	Offense-Column	Offense-Value	Reported-Column	Reported-Data Type
New York	<code>offenseDescription</code>	starts with "murder" or "homicide"	<code>reportDate</code>	timestamp
Boston	<code>OFFENSE_CODE_GROUP</code>	"Homicide"	<code>MONTH</code>	integer

In this notebook, we will use an ADF Databricks Notebooks activity to perform transformations on and extract homicide statistics from the crime data being ingested.

In this lesson you:

1. Create Databricks Access Token.
2. Add Databricks Notebook activity to pipeline.
3. Connect Copy Activities to Notebook Activity.
4. Publish the updated pipeline.
5. Trigger and Monitor the pipeline run.
6. Verify transformations of data by looking at the generated table in Databricks.
7. Perform a simple aggregation of the data.

Review Questions

- Q01 – What is the DataFrame method call to create temporary view?
- A01 – `createOrReplaceTempView()`
- Q02 – How do you create a DataFrame object?
- A02 – An object is created by introducing a variable name and equating it to something like `myDataFrameDF =`
- Q03 – Why do you chain methods (operations) `myDataFrameDF.select().filter().groupBy()`?
- A03 – To avoid the creation of temporary DataFrames as local variables

Lab: Orchestrating Data Movement with Azure Data Factory



Lab overview

- In this module, students will learn how Azure Data factory can be used to orchestrate the data movement from a wide range of data platform technologies. They will be able to explain the capabilities of the technology and be able to set up an end to end data pipeline that ingests data from SQL Database and load the data into SQL Data Warehouse. The student will also demonstrate how to call a compute resource.

Lab objectives

After completing this lab, you will be able to:

1. Explain how Azure Data Factory works
2. Azure Data Factory Components
3. Azure Data Factory and Databricks

Lab scenario

- As part of the digital transformation project, you have been tasked by the CIO to help the marketing departments. After performing the initial population of the Data Warehouse into Azure SQL Data Warehouse, the information services department want to automate this process. You have been asked to support the information services department in developing a solution that can automate the movement of data from Azure SQL Database.
- Your solution should be able to perform full copy of [SalesLT].[ProductCategory] and [SalesLT].[ProductDescription] transaction table that act as dimension tables of the same name in your Azure SQL Data Warehouse. Furthermore, the solution should also follow best practices of loading into a Massively Parallel Processing (MPP) system using Azure Data Factory as the orchestrator of the data movements.
- In addition, the Data Scientists have asked to confirm if Azure Databricks can be called from Azure Data Factory. To that end, you will create a simple proof of concept Data Factory pipeline that calls Azure Databricks as a compute resource.

At the end of this lab, you will have:

1. Explain how Azure Data Factory works
2. Azure Data Factory Components
3. Azure Data Factory and Databricks

Lab review

- Q01 - Can you think of example of automating batch data loads with Azure Data Factory back at work?

Module Summary >

In this module, you have learned about:

- Learned Azure Data Factory
- Understood Azure Data Factory Components
- Integrate Azure Data Factory with Databricks

Next steps >

After the course, [read the white paper on data migration from on-premise relational data warehouse to Azure using Azure Data Factory](#)

