INTRODUCTION

Migrate On-premise Data to Azure SQL Server

Marcelo Pastorino
SOFTWARE DEVELOPER / SOLUTIONS ARCHITECT

@evangeloper softwaredeveloper.io/marcelo

Let's Start Migrating Data to Azure!

The startup started leveraging on-premise infrastructure

Migrate data from on-premise SQL Server to an Azure SQL database

Data Migration Assistant

Detect compatibility issues

Migrate schema

Migrate data

Azure Data Factory

Create a data migration pipeline to migrate on-prem to Azure SQL Database

CLIP 1

Azure Data Studio

Cross-platform database tool

Windows, Mac, and Linux

Azure Data Studio

https://docs.microsoft.com/en-us/sql/azure-data-studio

Azure SQL Database

Managed SQL database engine

Based on latest SQL Server Enterprise Edition

Azure SQL Database

https://azure.microsoft.com/en-in/services/sql-database

CLIP 2

Data Migration Assistant

- Helps upgrade infrastructure to a modern data platform
- Detects compatibility and migration issues
- Detects unsupported or partially supported features
- Provides guidance and support
- Move schema and data

Download the Data Migration Assistant

https://www.microsoft.com/en-us/download/details.aspx?id=53595

Learn More About the Data Migration Assistant

https://docs.microsoft.com/en-us/sql/dma/dma-overview

Data Migration Assistant Assessment

Data Migration Assistant Migration

CLIP 3

Azure Data Factory

Managed cloud-based data integration platform

Used to create data-driven workflows

Workflows orchestrate and automate data movement activities

You can transform and store data

Helps operationalize the process

Extract, Transform and Load scenarios

Data Factory Pipelines

Connect to data sources

Transform and enrich data

Store data

Monitor pipeline execution

About Azure Data Factory and GIT Integration

Handle Azure Data Factory infrastructure as code

Use ARM templates

Move pipelines between environments

Development, Staging, and Production

Treat ADF pipelines as you treat code

Upcoming Pluralsight course

Deploying Data Pipelines in Microsoft Azure

Release Date: December 2019

CLIP 4

Understanding ADF Pipelines

Azure Data Factory Pipelines

Data Factories can contain one or more Pipelines

A Pipeline is a logical grouping of Activities

It allows managing Activities as a set

You can have many Activities per Pipeline

Understanding ADF Pipelines

Azure Data Factory Activities

Represents a processing step in the Pipeline

They are actions to perform on data

- Ingest data
- Transform data
- Store data

Pipeline Activities can be linked

- Execute sequentially or run in parallel

Activity Types

Data movement activities

Data transformation activities

Control activities

Understanding ADF Pipelines

Activity Types

Data movement and transformation activities can copy and transform data

Control activities manage the control flow in a pipeline

Understanding ADF Pipelines

Data Movement Activities

Copy Data Activity

- On-premises
- In the cloud

Understanding ADF Pipelines

Data Movement Activities Connectors

- **Azure Blob Storage**
- **Azure Cosmos DB**
- **Amazon Redshift**
- **Google BigQuery**
- **Hive**
- **MariaDB**
- **Oracle**
- **SQL Server**
- **MongoDB**
- **Amazon S3**

Understanding ADF Pipelines

Data Transformation Activities

Transform and enrich data

Hive

Pig

MapReduce

Spark

Azure Databricks

Understanding ADF Pipelines

Control Activities

Control pipeline flow

- ForEach
- Web

Azure Data Factory Activities

https://docs.microsoft.com/en-us/azure/data-factory/concepts-pipelines-activities

Understanding ADF Pipelines

Azure Data Factory Datasets

Named view

References data used in an Activity

- Files
- Folders
- Documents
- Tables

Understanding ADF Pipelines

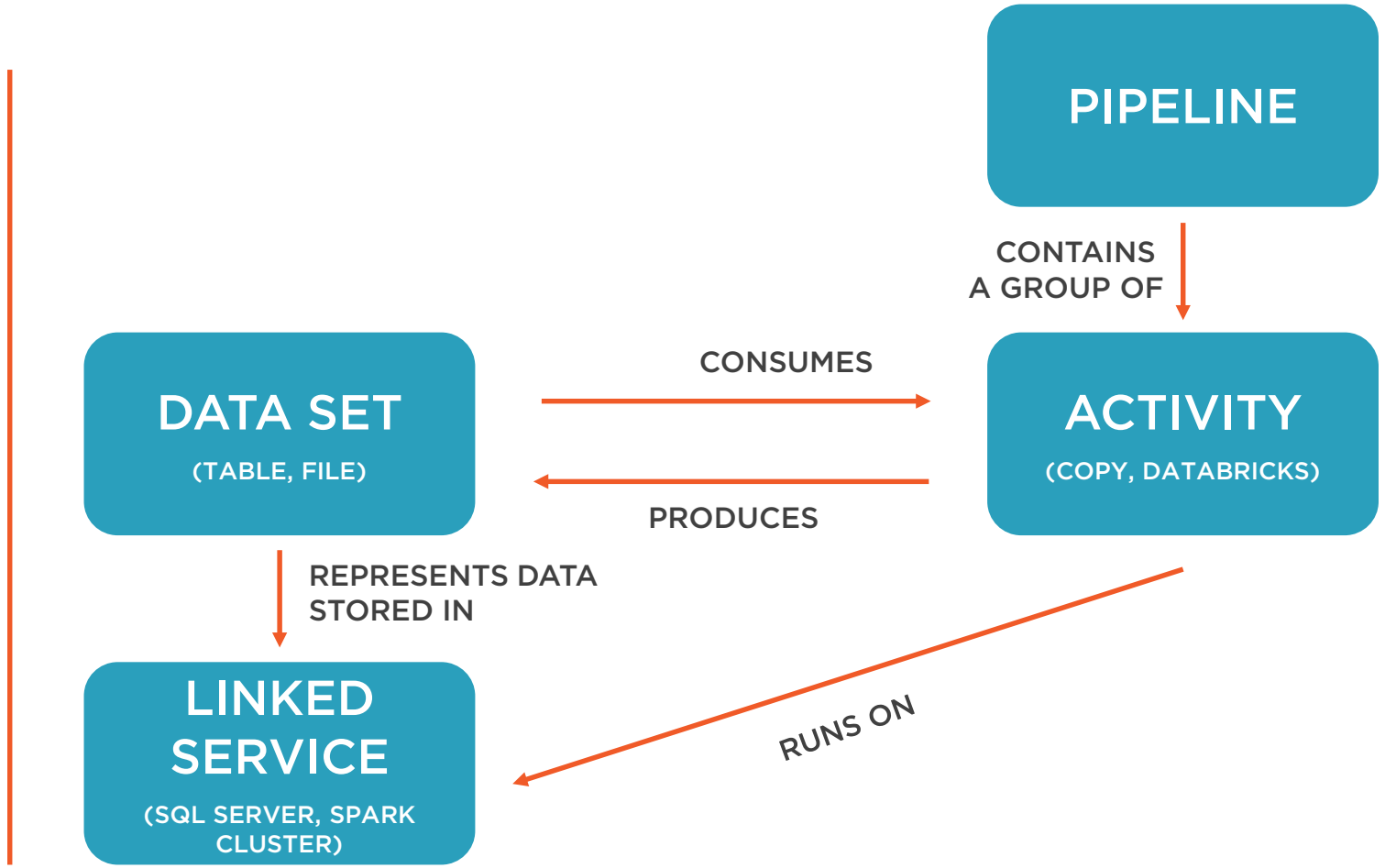
Azure Data Factory Datasets

Dataset

Col1	Col2	Col2	Col4
abc	123	abc	123
abc	123	abc	123

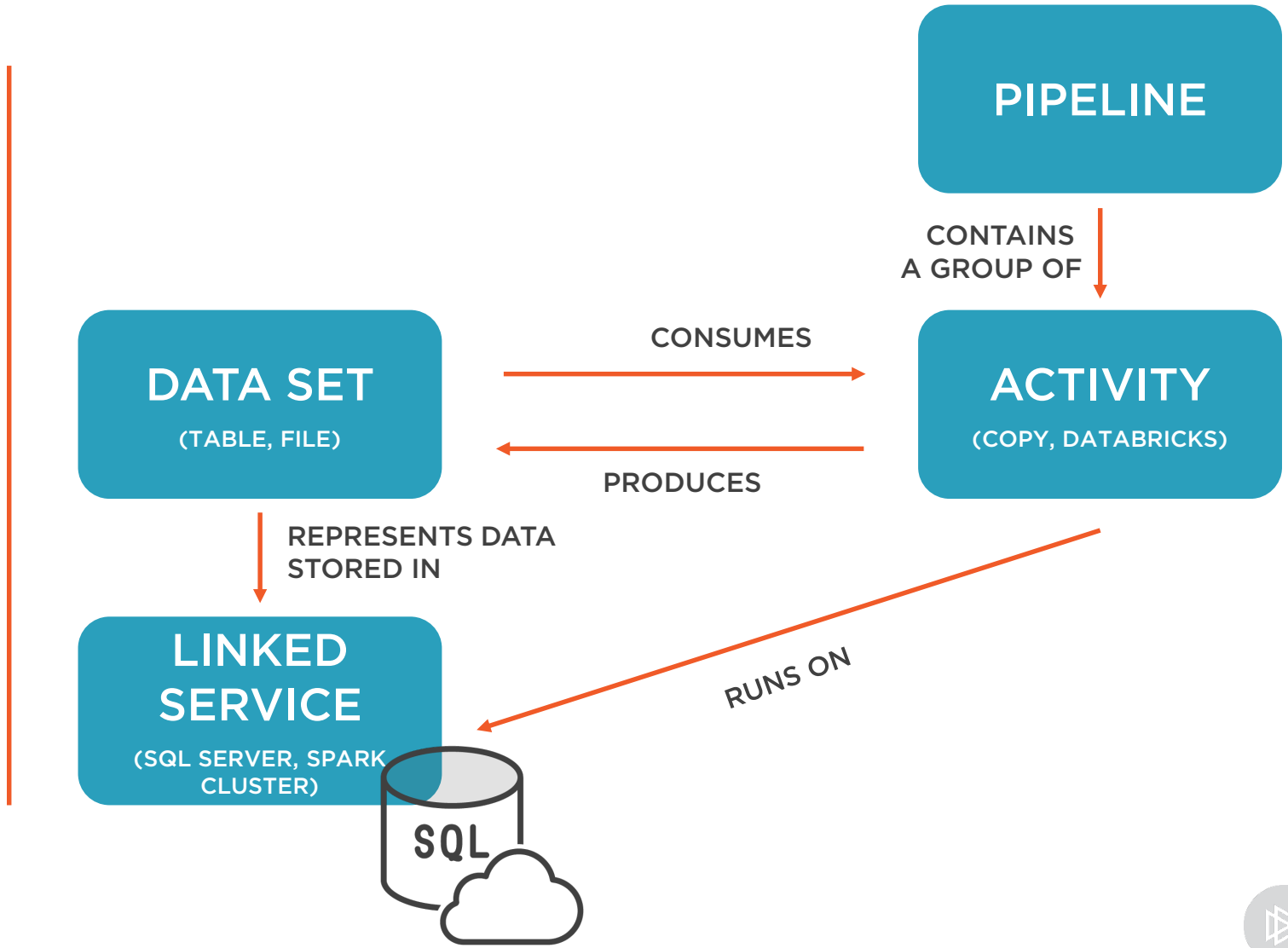
Understanding ADF Pipelines

Azure Data Factory Linked Services



Understanding ADF Pipelines

Azure Data Factory Linked Services



Understanding ADF Pipelines

Azure Data Factory Linked Services

Similar to connection strings

Represent the connection information to connect to external resources

- Datastores like Azure SQL Server
- Computer resource. E.g., Spark cluster

Data Factory Components

CLIP 5

Integration Runtimes

Compute infrastructure employed by Data Factory

Provides data integration capabilities across different network environments

Provides the bridge between activities and Linked Services

They operate in public and private networks

Integration Runtime Types

Azure Integration Runtime

Work on public networks

Responsible for data flows, data movements, and activity dispatches

Default Azure Integration Runtime

SSIS Integration Runtime

Supports SSIS package execution

Works on public and private networks

Self-hosted Integration Runtime

Work on public and private networks

Provide data movement and activity dispatch capabilities

Integration Runtimes

https://docs.microsoft.com/en-us/azure/data-factory/concepts-integration-runtime

CLIP 6

Azure SQL Tables

Legacy Sensor Readings table

Incremental Watermarks table

Sensor Readings table

Legacy Sensor Readings table

Azure Data Studio

```
create table LegacySensorReadings(
  Id int not null identity(1,1),
  ReadingId varchar(36) not null primary key,
  ReadingDateTime datetime not null default getdate(),
  LocationId smallint not null,
  PollutionLevel tinyint not null
)
```

Legacy Sensor Readings

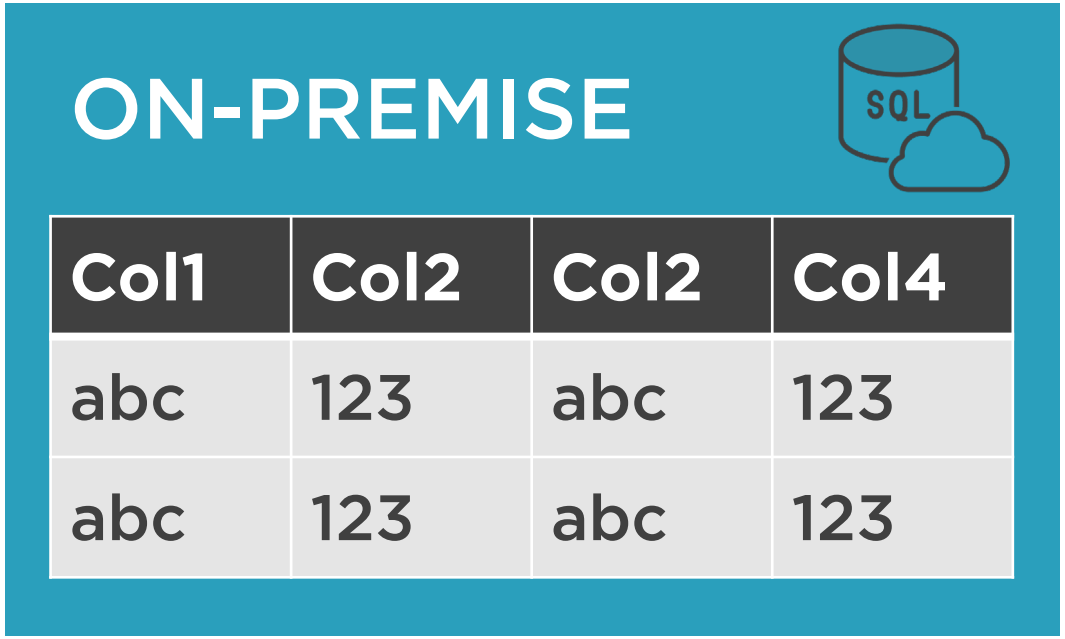
Backup table

Holds sensor reading events sent by the IoT devices

Target data to move

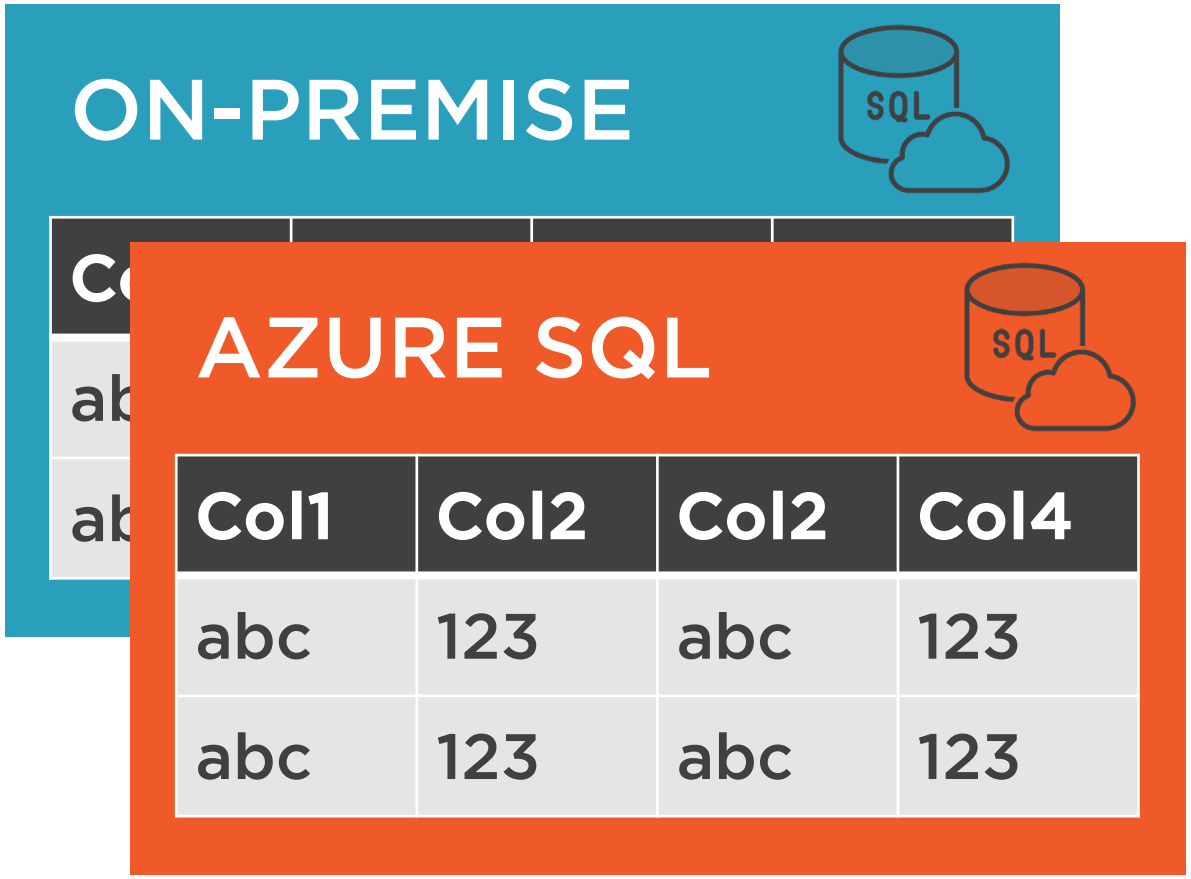
On-premise and Azure SQL Database Assets

Legacy Sensor Readings table



On-premise and Azure SQL Database Assets

Legacy Sensor Readings table



Legacy Sensor Readings

Receives events from some legacy sensors

We need to migrate data incrementally

By using a watermark table

Incremental Watermarks table

Azure Data Studio

```
create table IncrementalWatermarks(
  TableName varchar(50) not null primary key,
  LastInsertedId int not null
)
```

Incremental Watermarks table

Azure Data Studio

```
insert into IncrementalWatermarks(TableName,
LastInsertedId)

values('LegacySensorReadings', 0)
```

Incremental Watermarks table

Azure Data Studio

```
create proc spUpdateIncrementalWatermark
   @tableName varchar(50),
   @id int

as

   update IncrementalWatermarks
   set LastInsertedId = @id
   where TableName = @tableName

go
```

Sensor Readings table

Azure Data Studio

```
create table SensorReadings(
   ReadingId varchar(36) not null primary key,
   ReadingDateTime datetime not null,
   PollutionLevelId tinyint not null,
   PollutionLevel varchar(20) not null,
   LocationId smallint not null,
   City varchar(100) not null,
   Country varchar(100) not null,
   Population int not null,
   Latitude decimal(9,6) not null,
   Longitude decimal(9,6) not null
)
```

CLIP 7
NO SLIDES

CLIP 8

Azure Data Factory

Create a data migration pipeline to migrate on-prem to Azure SQL Database

Migrate IoT sensor data incrementally

CLIP 9

Incremental Watermarks table

Azure Data Studio

```
create table IncrementalWatermarks(
  TableName varchar(50) not null primary key,
  LastInsertedId int not null
)
```

CLIP 10

Azure Data Factory

Create new lookup activity to access a table in our on-premise SQL Server

Lookup MAX ID field value in the on-premise Legacy Sensor Readings table

Azure Pricing

https://azure.microsoft.com/en-us/pricing

CLIP 11

Add Dynamic Content

```
select * from LegacySensorReadings

where id > @{activity('LegacySensorReadingLastInserted').output.firstRow.LastInsertedId}

and id <= @{activity('MaxLegacySensorReadingId').output.firstRow.MaxId}
```

CLIP 12

Azure Data Factory

Update watermark table in Azure SQL database

Update MAX(ID) every time pipeline runs

The goal is to only import new data

Incremental Watermarks table

Azure Data Studio

```
create proc spUpdateIncrementalWatermark
   @tableName varchar(50),
   @id int

as

   update IncrementalWatermarks
   set LastInsertedId = @id
   where TableName = @tableName

go
```

Add Dynamic Content

```
select * from LegacySensorReadings

where id > @{activity('LegacySensorReadingLastInserted').output.firstRow.LastInsertedId}

and id <= @{activity('MaxLegacySensorReadingId').output.firstRow.MaxId}
```

CLIP 13
NO SLIDES

SUMMARY

Summary

Migrated data from on-premise SQL Servers to Azure using 2 different techniques

Learned how to detect compatibility issues between databases

Migrated schema from on-prem to the cloud servers

Summary

Learned about Azure Data Factory

Fully managed cloud-based data integration platform.

Learned key concepts such as pipelines, activities, data sets, linked services, integration runtimes

Created our first Data Factory pipeline

Move data incrementally