Overview of Data Pipelines and Azure Synapse Analytics

Azure Synapse Analytics is Microsoft's data analytics service in the cloud. A single Synapse workspace provides data engineers with immediate access to Apache Spark, the SQL data warehouse via SQL pools, data pipelines for data integration and ETL, the Azure Data Lake for data storage, and a single pane of glass experience for data developers and administrators. If you are a data engineer responsible for building big data solutions in Azure, Synapse is the service that you are going to use to integrate your heterogenous data sources and process that data for analysis, data science, and business intelligence.

In this book, we are going to explore building enterprise-grade data pipelines to load massive amounts of data into your Synapse instance for big data analytics and decision-making solutions. This initial chapter is going to provide an overview of the components of Azure Synapse Analytics and dive into our primary focus of data pipelines.

Components of Azure Synapse Analytics

Synapse Analytics brings together several Azure big data services into a single workspace with a single web browser studio experience for developers and administrators. Before we take a brief look at each component, let’s start by creating a new Synapse instance from the Azure portal (figure 1-1).

Graphical user interface, text, application, email

Description automatically generated

Figure 1-1. Create new Synapse instance from Azure portal

From the Azure portal (<https://portal.azure.com>), your new Synapse Analytics instance will provide access to the Azure Synapse Studio in your browser, which is where we will spend most of our time in this book. As you go through the series of prompts from the Azure portal to create your Synapse instance, you will provide login information to use for your SQL pools and security settings for your data. We’ll cover those options in the next chapter when we explore building Synapse pipelines.

Synapse Studio

In figure 1-2 below, you’ll see a sample data pipeline from the designer surface in Synapse Studio. This is available from the Integrate category which I’ve chosen on the left-hand navigator panel. The other categories from the Studio include develop, data, monitor, and manage. We’ll walk through how to use those other features later in this book. For now, we’ll explore the different aspects of your new Synapse workspace.

Graphical user interface, text, email

Description automatically generated

Figure 1-2. The Azure Synapse Studio

Enterprise data warehouse

From the Synapse Studio, you can spin-up database instances to use for your data warehouse workloads. You can choose between serverless and dedicated SQL pool options (see figure 1-3).

Graphical user interface, text, application

Description automatically generated

Figure 1-3. Choose between serverless and dedicated SQL pools.

To create a new database, click on the Data section from the left-hand navigation pane and add a new resource by selecting + SQL database.

Serverless SQL pools are generally used for Lake databases where your data is stored in Azure Data Lake Store, whereas dedicated SQL pools (see figure 1-4) are the more traditional next-generation of the SQL DW database.

Graphical user interface, text, application, email

Description automatically generated

Figure 1-4. This is the dedicated SQL option when creating a new SQL pool.

Data lake

Each Synapse instance provisions an Azure Data Lake store. This is where you will collect data files for processing. You can view the default (primary) data lake container by clicking the “Data” icon on the left-hand navigation pane, select “Linked” and then choose your primary account under “Azure Data Lake Storage Gen2” (figure 1-5).

Graphical user interface, text, application

Description automatically generated

Figure 1-5. This is the ADLS Gen2 container that is included in your Synapse instance.

The file explorer that is included in Synapse Studio (figure 1-6) can also show files from other storage locations outside of Synapse and has quick-action buttons on the top ribbon for creating a new data flow, integration dataset, and other features that we’ll discuss in future chapters.

Graphical user interface, text, application, email

Description automatically generated

Figure 1-6. The file explorer in Synapse Studio.

Data pipelines

The primary Synapse artifact we will explore in this book (data pipelines) can be found from the Integrate pipeline icon on the left-hand navigation pane. This is where you’ll go to begin building ETL and data integration pipelines to ingest, transform, and process data as well as building automated workflows. Synapse data pipelines use Azure Data Factory behind the scenes with essentially the same UI with some minor differences. I’ll save the deep-dive on this topic for the following chapters.

Apache Spark

Each Synapse workspace includes Apache Spark pools. You can manage your Spark pools and spin-up new Spark pools from the Manage section of the Studio (figure 1-7). Data engineers and data scientists will utilize Spark pools for analytics, data wrangling, data prep, and transformation. We’ll dig more into using Spark in pipelines via the Notebook and Spark Job activities later in this book.

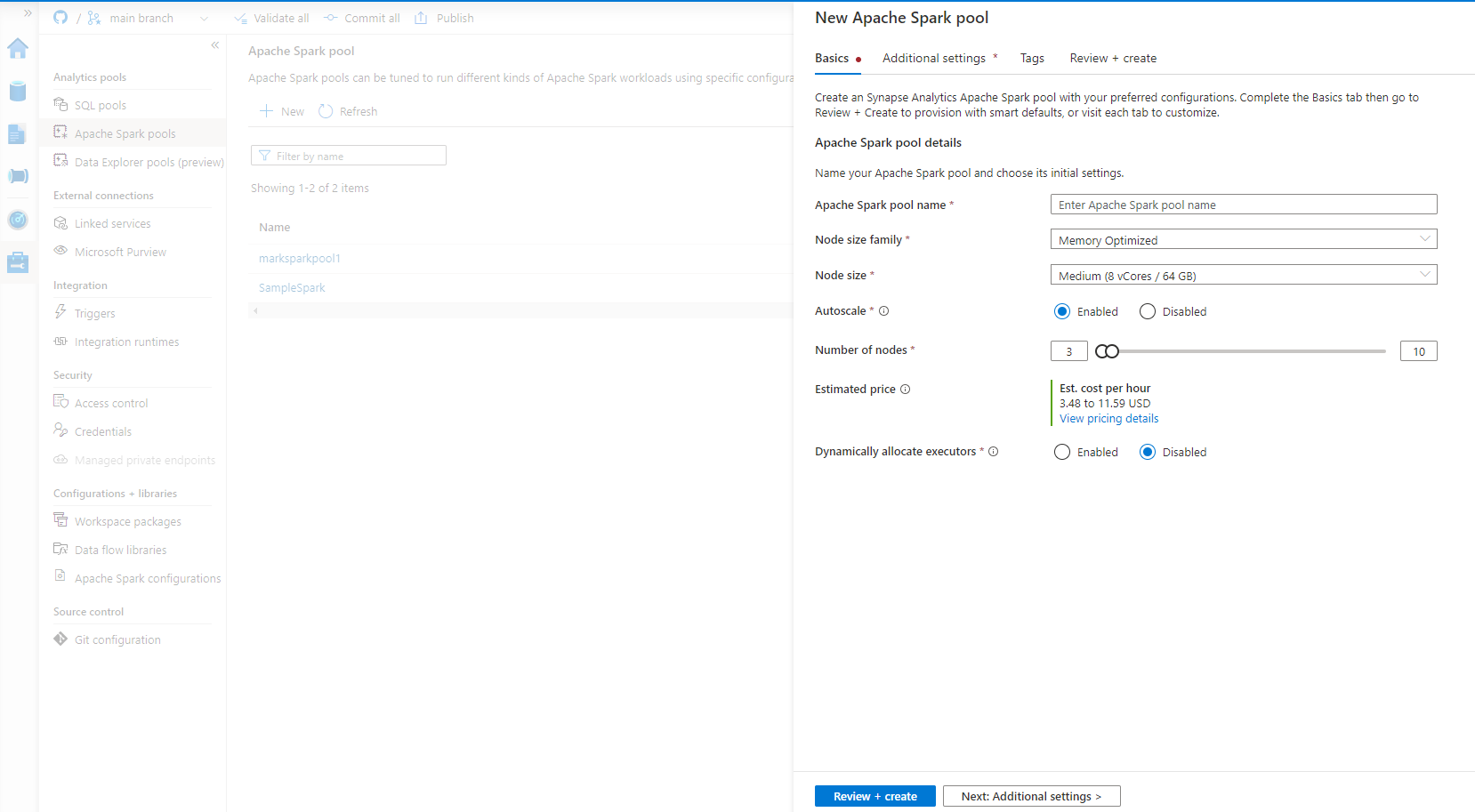


Figure 1-7. Configure Spark pools.

Managing Spark pools is shown in figure 1-8, which is available by clicking on the “Manage” section of the navigation pane on the left. In figure 1-9, you can see how to find the Notebook surface to interact with your data in the lake using Spark by clicking on the “Develop” section in the navigation pane.

Graphical user interface, text, application

Description automatically generated

Figure 1-8. Manage Spark pools from the Manage section.

Graphical user interface, application, Word

Description automatically generated

Figure 1-9. Create a new Notebook from the Develop section.

Synapse Link (Preview)

Synapse Link is a continuous data replication technology that is also part of the Integrate category in Synapse Studio. Use this feature if you need to move data on a continuous, incremental basis, into your data lake from SQL Server.

Graphical user interface, text, application

Description automatically generated

Figure 1-8. Synapse link is available under “Link connection” in the Integrate category, where you will also find data pipelines.

Data flows

Just like data pipelines, data flows come from Azure Data Factory and is the code-free data transformation at Spark-scale feature that is found in Data Factory known as “Mapping data flows” (figure 1-9). Data flows provides a code-free design surface to design and build complex data transformation patterns that execute natively on Synapse Spark clusters that are serverless and managed by Synapse.

A picture containing chart

Description automatically generated

Figure 1-9. Data flows in Synapse

We will use data flows in several chapters in this book to build ETL pipelines. Other patterns we’ll explore will also use Spark natively via Notebooks and Spark Job Definition activities in data pipelines. You’ll find data flows in the Develop category (figure 1-10).

Graphical user interface, text, application

Description automatically generated

Figure 1-10. Create a data flow from the “Develop” navigation section.

Copy data tool

A very common tool that you’ll use to ingest data into Synapse (whether that be the data lake or SQL database) is to use the Copy data tool (figure 1-11). This is the same tool that is found in Azure Data Factory and in Synapse you’ll find it in “Ingest”. The copy data tool provides a step-by-step configuration approach to building data ingestion pipelines. With this tool, you also have the option to build metadata-driven pipelines that provide external configurations to define dynamic pipeline ingestion patterns.

Graphical user interface, application

Description automatically generated

Figure 1-11. The Copy data tool.

What are data pipelines?

The focus of the rest of our book will be on building enterprise data pipelines using the components described above in Azure Synapse Analytics. We’ll dive deeply into data integration and ETL sample scenarios in the next chapter. But before we do that, let’s review what data pipelines mean for data engineers in big data analytics projects.

The fundamental goal of what data engineers need to achieve is to ingest all of the raw data needed to form intelligent data models that can be used by business users and data scientists to make better business decisions. In Synapse, you’ll achieve that goal by building data pipelines that will be automated as background processes. Data pipelines are a top-level artifact in the Synapse Studio that can be found in the “Integrate” navigation category.

Let’s walk through the reference architecture in figure 1-12. From a persona perspective, the primary participants in this data journey are data engineers, data scientists, and citizen integrators. Synapse data pipelines are used by all of those personas, while the final product is typically consumed by business analysts and decision makers in Power BI. Along the way, you will also encounter database administrators (serving layer) and storage administrators (staging layer).

Graphical user interface, application

Description automatically generated

Figure 1-12. This is a reference architecture for data integration in Synapse Analytics

We are going to essentially refine raw data from the left into actionable intelligence on the right. You will ingest data from many varied sources with many different sizes and shapes of data. In cloud-first big data projects, data will change often and can grow to be very large. You will find it challenging to work from the basic assumption that each data source will always abide by a given contract. Instead, expect data schemas to change and data quality will need to be always a top-of-mind issue for you.

In terms of the data lifecycle described above, you can first apply a level of data quality control and cleaning as you land the data in the lake (ADLS Gen2) for staging. But it’s the Transform step that typically cleans all of the raw data in Staging and prepares it for data warehouse models in SQL Pools and preps the data for reporting models. In this book, we’ll focus on data transformation using data flows, Notebooks, and SQL.

In the Serving layer, data will be stored in SQL database tables (data warehouse) and can be served directly to users from there, typically consumed there by data scientists and data analysts. Typical business users will consume data through reporting in Power BI as part of the Presentation layer.

Synapse data pipelines provide the UI for designing these processes using a no-code designer. The data pipeline will include activities like Copy and Data Flows to ingest and transform the data, clean it, and validate schemas which you can snap together in workflows that also include control flow and other orchestration options. After testing and completing your design, your final step will be to set schedules and other trigger types that will define the cadence and frequency in which these pipelines need to run to process data that matches up with source data availability and business requirements. Synapse pipelines also include a rich monitoring capability so that you can observe the processes while they are executing as well as observe history of successful and failed runs which can be used with notifications and re-runs in order to maintain a healthy data ecosystem.

Synapse data pipeline components

Let’s now take those concepts and apply them to the data pipeline components in Synapse that you’ll use to build these projects.

Pipelines

The pipeline is the most important fundamental component of a data pipeline in Synapse. You’ll typically start here when building an ETL or data integration project. All triggers, monitoring, and actions depend on the pipeline component. Figure 1-13 shows the new menu for creating a new pipeline.

Graphical user interface, text, application

Description automatically generated

Figure 1-13. Create a new pipeline.

Linked services

Linked services define connections to your data. You will typically define a linked service to your source and target data that is outside of Synapse using your credentials and a location (i.e. URL, database connection string, etc.). Data that is already in Synapse or that originates inside of Synapse from the default storage container or the default SQL database will already have Linked Services defined (figure 1-14). Linked Service types are available for all of the connectors present in Synapse which range from on-premises data like sFTP, Oracle, and SQL Server, to cloud-first data like Salesforce, Dynamics, Azure SQL DB, and Blob store. You can manage linked services from the “Manage” section.

Graphical user interface, application

Description automatically generated

Figure 1-14. Linked services are in “Manage” in Synapse Studio.

Datasets

In Synapse data pipelines, datasets are a representation of the location and shape of your data. Datasets use the associated linked service type from above and, just linked services, work exactly the same as datasets in Azure Data Factory. You’ll note that there is slight name difference in Synapse where they are called “Integration datasets” (figure 1-15).

Graphical user interface, application

Description automatically generated

Figure 1-15. Integration datasets in Synapse can be found in the “Data” section.

Integration runtimes

Integration runtimes are configurations that represent the location and power of the compute you wish to use inside your pipeline. There are 3 types of integration runtimes in Synapse (same as in ADF) that can be managed in the Synapse Studio (figure 1-16):

Azure integration runtime

Synapse is a cloud-first Azure service and the Azure integration runtime represents the cloud-based serverless compute for pipelines. This provides auto-scale compute for your pipelines in the cloud. Azure IR is the most common type of integration runtime and every Synapse instance includes a default integration runtime in the region that you chose when you created your instance. The Azure IR also includes an option for running the compute inside of a virtual network which can be used for securing network traffic between your data sources and targets in your pipeline. You can also configure the Azure IR with the Vnet option to reach on-premises data sources.

Self-hosted integration runtime

The self-hosted integration runtime is essentially an on-premises version of the Synapse pipeline compute resources and uses a downloaded executable that you will install and run from your own virtual machine or servers on premises or in a cloud-hosted VM. Instead of the pipeline compute resources executing in the cloud managed by Synapse, the self-hosted IR is fully user-managed.

SSIS Integration Runtime

Use this integration runtime for lift-and-shift of your SQL Server Integration Services packages. We will not cover this IR type in this book.

Graphical user interface, text, application, email

Description automatically generated

Figure 1-16. Configure integration runtimes from "Manage".

Triggers

After you’ve completed the design and testing of your data pipeline, you can set a trigger to automate the execution of your pipeline. Synapse pipelines are batch-oriented data pipelines as opposed to real-time, so you will need to configure a trigger to tell Synapse when your pipeline should run in the background. There 4 types of triggers your pipeline can use, same as in Azure Data Factory:

Schedule

Schedules are the most common trigger type used. You’ll set the pipeline recurrence based on frequency (minutes, hours, days, etc.) with the lowest interval being 1 minute.

Tumbling window

If your data arrives at regular intervals, you can set a cadence, or what are known as “time slices” as low as 15 minutes. Pipelines using this trigger type will automatically execute at that recurrence interval. This trigger type is also very good for running “back-fill” jobs when data was not previously received for some of those time slices.

Storage events

This trigger type is a good option when you want the arrival or deletion of a file to trigger your pipeline. The use of that file trigger should be thought of as a “semaphore” signal as opposed to using this trigger type as an event-based trigger where you may receive 1000s of files in a short period of time. Instead, treat Synapse pipeline triggers as a batch-oriented scheduler.

Custom events

Custom events have similar functional properties to storage events, except that you can define your own event handler. You can hook up event grid to the pipeline trigger using a custom event trigger to signal Synapse to execute your pipeline using a custom event.

Graphical user interface, application, email

Description automatically generated

Figure 1-17. Adding a new trigger to a pipeline.

Data flows

Data flows are the code-free data transformation feature in Synapse Studio that allow you to build powerful scale-out ETL jobs and then execute them from a data flow activity inside of your pipeline. A complete ETL pipeline in Synapse is typically including a data flow using basically the same engine as is included in ADF’s Mapping Data Flows. To design and develop data flows, go to the Develop category in the navigation pane.

Graphical user interface, application, Word

Description automatically generated

Figure 1-18. Data flows are in the Develop section of the navigator pane.

Summary

In this first chapter, we covered the basics of Azure Synapse Analytics and data pipelines. You learned about the primary components of an Azure Synapse Analytics instance and the accompanying data engineer tool “Synapse Studio”. We briefly touched on the components of a data pipeline to help provide and understanding of how and when you’ll use some of these components. In the next chapter, we’re going to explore several scenarios specifically intended for Synapse data pipelines.