Getting to know Azure Synapse Analytics Pipelines

In the previous chapter, we introduced you to Azure Synapse Analytics as Microsoft’s big data analytics service in the cloud. While touching very briefly on the primary features inside of Synapse, including Spark, SQL, and data pipelines, we focused on the important components of Synapse pipelines. In this chapter, we’ll begin diving into details of Synapse pipelines and how you can use them for data integration and ETL in your Synapse analytics projects.

Tour of Synapse Pipelines

Once you’ve instantiated your Azure Synapse workspace (Chapter 1 talks about using the Azure portal to spin-up a new Synapse workspace), you can open the Synapse Studio from the Azure portal (figure 2-1).

Graphical user interface, text, application

Description automatically generated

Figure 2-1. Open Synapse Studio tile from the Azure portal

The Synapse pipeline design interface

Inside the Synapse Studio, you’ll find data pipelines under the “Integrate” category in the left-hand navigation pane (figure 2-2).

Graphical user interface, application, Word

Description automatically generated

Figure 2-2. Click on “Integrate” in the Synapse Studio navigation pane to bring up the data pipelines interface

Let’s start by creating a new pipeline and we’ll build a very simple first pipeline so that you can get comfortable with the interface. Click on + (new) Pipeline (figure 2-3) to get started with a new pipeline.

Graphical user interface, text, application, Word

Description automatically generated

Figure 2-3. Create a new pipeline.

Name your pipeline “My First Pipeline” (figure 2-4). Throughout this chapter, we will work directly against the “live” version of your Azure Synapse workspace, which means that we are not setting up a “sandbox” environment via Git. Synapse integrates with Github and Azure Dev Ops (ADO) and that is the recommended working technique. Later chapters will dive into the configuration and the CI/CD process.

Without Git enabled, you’ll find that you cannot save your work while in progress of editing and designing inside the studio. To do this, follow the instructions later in this book to connect to Git. In order to publish your new pipeline, you must first add any activity and ensure that all of the required properties have been set so that the pipeline is in a “valid” state.

Graphical user interface

Description automatically generated

Figure 2-4. Name your pipeline “My First pipeline”

The objectives for building Synapse pipeline include ingesting data into Synapse, transforming data into shapes for analytics, orchestrating actions on the Synapse platform like Spark Notebooks and SQL jobs, and moving data around the Azure fabric. We’ll cover many of these scenarios in depth in coming chapters. For now, we’ll just explore the design surface.

Synapse pipeline activities

Pipelines are constructed by snapping together activities that act as a workflow to help you build big data analytics. On the left-hand side of the pipeline designer, you will see a docked toolbar (figure 2-5).

Graphical user interface, text

Description automatically generated with medium confidence

Figure 2-5. The pipeline toolbar in Synapse.

As a pipeline developer, you’ll have a rich set of activities available to you to build your ETL and data integration projects. You can think about the categories using this taxonomy:

Internal Synapse activities

The activities in the “Synapse” category allow you to orchestrate Synapse Spark jobs and Notebooks and SQL pool stored procedures (figure 2-6) that leverage the compute resources of your Synapse workspace.

Graphical user interface, text, application

Description automatically generated

Figure 2-6. Synapse activities

Move & transform

Use these activities to ingest data (Copy data) and transform data (Data flow) using the low-code designer in Synapse that are the same as those found in Azure Data Factory (figure 2-7). These are very good activities for you to start with here in your first pipeline to move data into your workspace.

Graphical user interface, text, application, email

Description automatically generated

Figure 2-7. Move & transform pipeline activities

External compute activities

There are a number of pipeline activities that you’ll find in the Synapse pipeline designer toolbox that provide access to Azure compute services that are external to your Synapse workspace (figure 2-8). When building a Synapse solution, you will likely use these activities in fewer scenarios than the Synapse activities above. You can call out to Azure Functions, Azure Data Explorer commands, custom code in Azure Batch, Azure Databricks jobs, and Azure Data Lake Analytics.

In addition to the Azure compute services listed in the screenshot below, you can also build pipelines that call out to HDInsight and Azure Machine Learning. We will not cover those in this book, but you can consider using these activities if you have Hive or MapReduce jobs for data transformation, or to use Synapse pipelines as a scheduled mechanism to refresh or train data models.

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Figure 2-8. Activities that are external to your Synapse workspace

General

“General” activities represent a set of miscellaneous pipeline commands that allow you to perform common actions that you’ll need when building orchestration processes (figure 2-9). You’ll likely find that you need these as you are constructing many of your data pipeline patterns. Let’s take a quick look at each of the general activities:

Graphical user interface, text

Description automatically generated

Figure 2-9. General activities

Append variable

Variables are used in pipeline to hold values that you can use throughout the lifecycle of the pipeline for counters, holding return values, and other typical patterns that require values to pass throughout activities in the pipeline. There is also a set variable activity below for scalar types. The append variable is specific to array variables and enables append values to your array.

Delete

The delete activity is used to delete files from file systems. You will choose a dataset that points to the files that you wish to delete. We’ll cover datasets in a bit more detail later in the chapter.

Execute Pipeline

Use this activity to create cascading parent-child patterns by adding an Execute Pipeline activity that can call other pipelines in your workspace. These are very common patterns in Synapse pipelines.

Execute SSIS Package

Although we will not cover SSIS in this book, it is important to know that you can lift and shift your existing SSIS packages into the cloud using Synapse. In your Synapse workspace is a compute resource object called an Integration Runtime (IR). One of the IR types available to you is an SSIS IR so that you can execute SSIS packages from a Synapse pipeline. We’ll talk more about the cloud (Azure IR) and on-premises (Self-hosted IR) integration runtimes in this chapter.

Fail

The Fail activity allows you, as a pipeline developer, to throw an error. You can customize the activity with your own custom error message and error code.

Get Metadata

The Get Metadata activity is very useful for querying metadata from datasets. The Get Metadata activity allows you to choose the meta fields that you wish to extract from the dataset. The field list changes based on the dataset type. The field list values can then be used later in your workflow instead control flow activities like “for each” and “if”.

Lookup

The Lookup activity is very useful when creating dynamic pipelines where the lookup is used as a means to get configuration from a dataset or to find values from a database or file that you will use in a later activity like a Copy activity or Data Flow. It is important to understand that the pipeline infrastructure is a shared platform service in Azure and therefore limits are applied to pipeline activities like lookup that limit the size of the data that can be returned. You should treat pipelines specifically as control flow and workflow mechanisms and use data flows for large data processing and large data lookups.

Stored Procedure

This is a very useful activity that you can use to call SQL stored procedures inline inside your pipelines. We’ll use the stored procedure activity later in this book to call SQL dedicated procs. To connect to your database, you’ll choose a linked service that contains your credentials and then select from the list of discovered stored procedures. We’ll talk more about linked services in just a bit.

Script

While the stored procedure activity is used to call existing code inside your database, the Script activity is very useful to call ad-hoc queries against your database, supporting both DML as well as DDL SQL statements. Whereas the stored proc activity is limited to Synapse SQL, SQL Server and Azure SQL DB, the Script activity supports many more database types including Oracle and Snowflake.

Set Variable

We talked briefly above about the append variable activity for array types. The Set Variable activity is where you will set values inside scalar variable types.

Validation

The validation activity is used to check for the existence of dataset references. It is useful when you only wish to continue your workflow if you can verify that the data required is available.

Web

The Web activity is very useful when you need to call into a REST API endpoint. GET, POST, and other REST methods are supported. The response from the API call can be captured in the activity output.

Webhook

The Webhook activity is very useful to extend your pipeline workflow by calling into a custom web endpoint.

Wait

Synapse pipelines are workflows with control flow, compute activities, data transformation, and data movement. In many case, you’ll want to wait for a period of time before checking on an operation or to provide time for a process to complete before continuing. The wait activity adds a delay in your pipeline using a configurable wait time property.

Iteration & conditionals

This is where you will find a rich set of control flow capabilities to add to your pipelines (figure 2-10).

Graphical user interface, text, application, email

Description automatically generated

Figure 2-10. Iteration & conditionals activities category

Filter

The filter activity will let you filter values from an input array. You’ll find it very useful when you have to filter out specific values from an array that you’ll use in a subsequent “for each” (below).

For Each

The “for each” activity is a general-purpose control flow activity for looping over a collection of values. Each iteration of the loop can execute another set of pipeline activities.

If Condition

This is a general purpose if condition control flow activity that will allow you to direct the flow of control down both a true and a false path.

Graphical user interface, application

Description automatically generated

Figure 2-11. If condition activity

Switch

For conditional control flow beyond just the true/false binary paths of the “if” activity, you can use the Switch activity and add more case paths as required.

Graphical user interface, text, application

Description automatically generated

Figure 2-12. The switch activity allows multiple case statements for flexible conditional flow execution

Until

Until is another looping control flow activity that will execute your activities inside the loop until the conditional expression that you define evaluates to true. The loop will execute minimum of one time with Until and will evaluate your expression on every iteration, completing until the condition evaluates to true.

Linked Services

Linked Services in Synapse work very much like Linked Services in Azure Data Factory (ADF). Linked Services define the connectivity to your data and compute services (figure 2-12).

Graphical user interface, application, table

Description automatically generated

Figure 2-12. Linked services are available from the Manage category

When creating linked services, you’ll navigate to the “Manage” category in the Synapse Studio explorer and click “+ New” to create linked services. You’ll need linked services to connect to your data sources, data destinations (sinks), and to connect to external compute resources like Databricks and HDI. The connection panel (figure 2-13) that flies in when creating linked services is similar to creating datasets (integration datasets below) that define the type of data or compute resource that you will connect to. Synapse provides over 100 connection types and includes default linked services to your internal Synapse workspace resources including the SQL database and data lake storage that are provisioned automatically when creating a new Synapse instance.

Graphical user interface, application

Description automatically generated

Figure 2-13. Linked Services connection type panel

Pipeline expression builder

Synapse pipelines includes the ADF pipeline expression builder experience that enables pipeline developers to extend the functionality of their workflows to include customizations based on parameters and custom logic using the pipeline expression language. The way that you will activate the pipeline expression builder is by clicking the blue “Add dynamic content” link on properties throughout the pipeline UI and the activity properties panels (figure 2-14).

Graphical user interface, text, application

Description automatically generated

Figure 2-14. Activate the pipeline expression builder from “Add dynamic content” links

When the expression builder launches (figure 2-15), you will have a text box available to enter your expression that will configure the behavior of that property. Synapse pipelines use the ADF / Logic Apps expression function syntax that begins with the “@” symbol. Typing "@” will activate Intellisense to display the list of available functions that match the text as you type.

Graphical user interface, text, application, email

Description automatically generated

Figure 2-15. Pipeline expression builder

In subsequent chapters, we’ll show many examples of using the expression language to configure custom logic in pipelines that can enable flexible and re-usable patterns. A key aspect of that reusability is leveraging pipeline parameters (figure 2-16). Parameters can be used throughout your pipeline activities and can be set from the pipeline designer by clicking on the graph designer whitespace. The Parameters option is in the bottom properties panel.

Graphical user interface, text, application, email

Description automatically generated

Figure 2-16. Pipeline parameters

I’ll touch on triggers before the end of this chapter, but for now, it is important to keep in mind that you will need to set the value of your parameters in the pipeline trigger or use a default value in your parameters. When you test your pipeline using the debug button (we’ll talk in depth about the CI/CD and testing process for pipelines in later chapters), you’ll be prompted to enter a value for your parameters. But pipelines operate in an unattended automated fashion using triggers. In order to create an operationalized pipeline, you’ll need to set parameter values in your triggers as well (figure 2-17).

Table

Description automatically generated

Figure 2-17. Setting parameters from a trigger

Integration Datasets

We touched on datasets above when we talked about the pipeline activities. Datasets are a concept that comes to Synapse from Azure Data Factory and represent the shape and location for your data. Datasets in Synapse do not hold any data, they are metadata. In your Synapse Studio, you will see datasets under the “Integration Datasets” heading when you select the Data navigation category (figure 2-18). Note that you will have click on the “Linked” section because the Data category has a separate section for your Synapse Workspace data resources, which are Synapse databases.

Graphical user interface, text, application

Description automatically generated

Figure 2-18. Integration datasets are available under the Data navigation category

When creating data pipelines in Synapse, you’ll need datasets to represent your sources and sinks in the Copy activity when moving data and use integration datasets or “inline” datasets to represent the data that you are transforming when building data flows.

Integration Runtimes

Integration runtimes (IRs) come from the Azure Data Factory application model and represent the pipeline compute being used either in the cloud (Azure IR) or on-prem (self-hosted IR). There is also an SSIS integration runtime that is used to lift and shift legacy SSIS packages which we will not explore in this book. You can manage IRs from the “Manage” category (figure 2-19). Your Synapse workspace will automatically create an “AutoResolveIntegrationRuntime” by default. This is the IR that will be used when you first create a pipeline.

Graphical user interface, application

Description automatically generated

Figure 2-19. Integration runtimes are available in the Manage category

When we talk about more advanced concepts in the next chapter around data transformations with data flows, you’ll learn how to create a new IR that sets specific compute size parameters for the Azure IR. When first getting started, you can use the default IR in your linked services and activities. That tells Synapse to use the compute resources identified in the integration runtime configuration to execute your processes. If you need to connect to on-premises resources, then you’ll create a new self-hosted IR which will require you to install the IR as an executable on your on-premises servers or VMs, or on a cloud VM. If your resources are protected inside a Vnet and are not accessible from public Azure networks, then you can enable the Vnet option on the Azure IR.

Triggers

The final step after testing your pipeline with debug, is to operationalize your pipeline by creating triggers that will schedule your pipelines for automated unattended execution. In later chapters, we’ll dive into a complete picture of end-to-end operationalization and testing of your pipelines using features like Git and CI/CD. There are two ways to create triggers in Synapse. There is an “Add trigger” button on the pipeline design surface in the top ribbon and the “Manage” section in the studio includes a “Triggers” section (figure 2-20). The manage section is where you will go to manage triggers that were already created previously. Triggers are shared objects in your workspace so that you can set common schedules across your pipelines.

Graphical user interface, application

Description automatically generated

Figure 2-20. Manage triggers from the “Manage” category in Synapse Studio.

Synapse pipelines operate on a batch-oriented mechanism meaning that triggers are necessary to automate your pipeline processes. There are four trigger types for Synapse pipelines:

Schedule trigger

This is the most common trigger type in pipelines. It enables pipeline execution based on a wall clock time, similar to setting schedules in SSIS using SQL Agent or a Linux cron job.

Tumbling window

Tumbling window will establish a set of time windows (i.e. 15 mins) that Synapse pipelines will establish from a start time that you choose.

Storage events

If you require your pipeline to execute when a file arrives or is removed, choose storage events.

Custom events

Custom events are useful when you wish to wire up the execution of your pipeline to your own events. You can do this by connecting the event trigger to Azure Event Grid such that when a specified event occurs, it can trigger your pipeline. Although this provides more an event-based trigger point for your pipeline executions, it does eliminate the batch-oriented principles of Synapse. There are service limits that are imposed that will limit the number of events that can be processed within a specific time period.

# Graphical user interface, text, application, email Description automatically generated

Figure 2-21. Create new trigger

Summary

In this chapter, we toured Synapse pipelines from the Synapse Studio. We talked about the purpose of data pipelines in Synapse projects and the design surface for creating pipelines. After introducing each activity type, you learned about datasets, linked services, and integration runtimes. When building and designing pipelines, you will use the pipeline expression builder for dynamic content that can utilize pipeline parameters, to make generic patterns which emphasize reuse, a very important pattern when building enterprise-grade data ingestions pipelines. Finally, we briefly touched on operationalization of your pipelines by creating triggers for unattended scheduling of your data pipelines. Understanding these concepts will be critically important for you as we progress through the rest of the book as we put these tools into practice when building our big data analytics data integration projects.