

Describe common practices for data loading

10 minutes

Data ingestion is the first part of any data warehousing solution. It is arguably the most important part. If you lose any data at this point, then any resulting information can be inaccurate, failing to represent the facts on which you might base your business decisions. In a big data system, data ingestion has to be fast enough to capture the large quantities of data that may be heading your way, and have enough compute power to process this data in a timely manner.

Azure provides several services you can use to ingest data. These services can operate with almost any source. In this unit, you'll examine some of the more popular tools used with Azure: Azure Data Factory, PolyBase, SQL Server Integration Services, and Azure Databricks.

Ingest data using Azure Data Factory

Azure Data Factory is a data ingestion and transformation service that allows you to load raw data from many different sources, both on-premises and in the cloud. As it ingests the data, Data Factory can clean, transform, and restructure the data, before loading it into a repository such as a data warehouse. Once the data is in the data warehouse, you can analyze it.

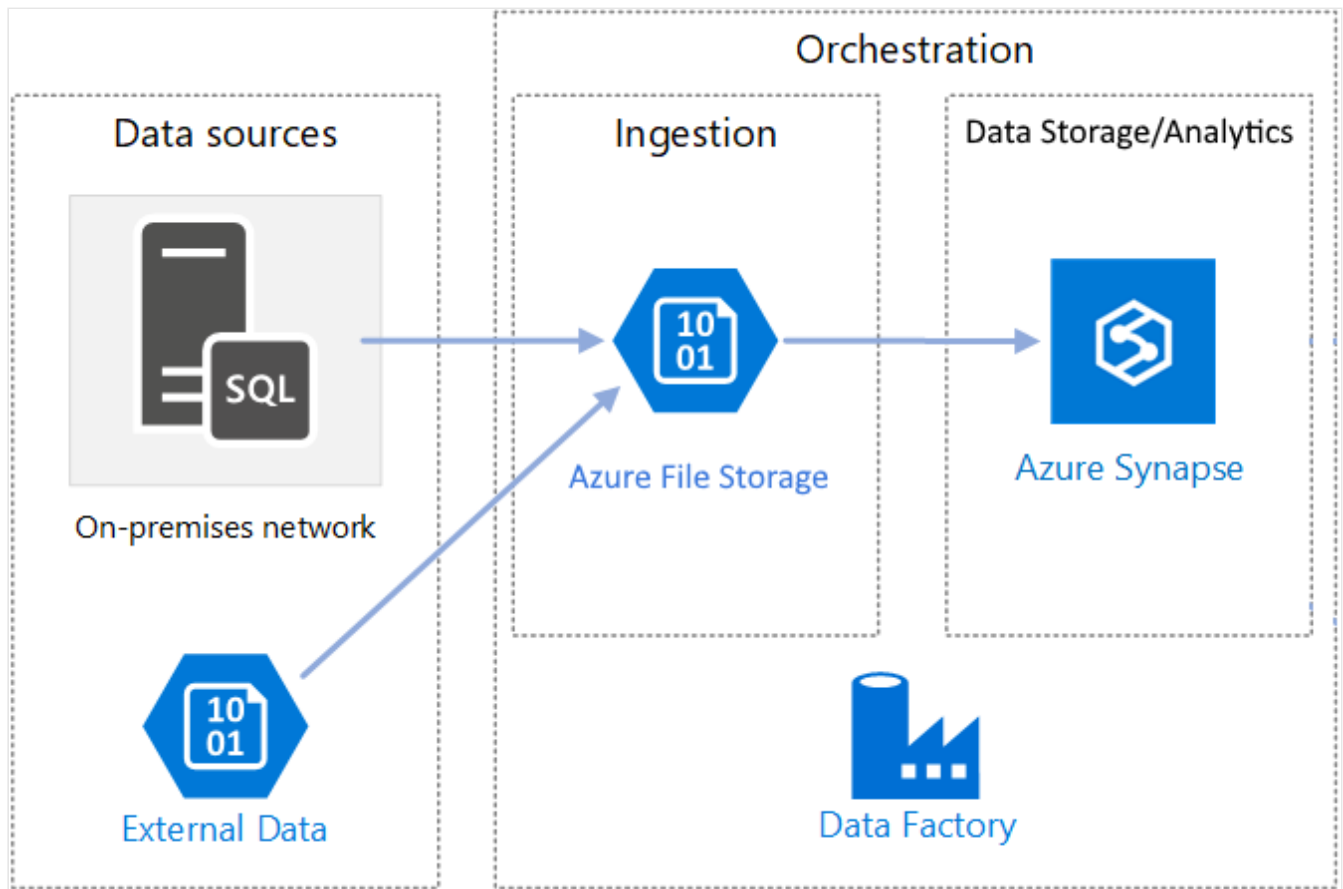
The data integration capabilities in Azure Synapse Analytics are based on Azure Data Factory, and can be used from within Azure Synapse Studio.

Note

To learn more about data integration capabilities, review [Data integration in Azure Synapse Analytics versus Azure Data Factory](#).

Data Factory contains a series of interconnected systems that provide a complete end-to-end platform for data engineers. You can load static data, but you can also ingest streaming data. Loading data from a stream offers a real-time solution for data that arrives quickly or that changes rapidly. Using streaming, you can use Azure Data Factory to continually update the information in a data warehouse with the latest data.

Data Factory provides an *orchestration* engine. Orchestration is the process of directing and controlling other services, and connecting them together, to allow data to flow between them. Data Factory uses orchestration to combine and automate sequences of tasks that use different services to perform complex operations.



Azure Data Factory uses a number of different resources: linked services, datasets, and pipelines. The following sections describe how Data Factory uses these resources.

Understand linked services

Data Factory moves data from a data source to a destination. A linked service provides the information needed for Data Factory to connect to a source or destination. For example, you can use an Azure Blob Storage linked service to connect a storage account to Data Factory, or the Azure SQL Database linked service to connect to a SQL database.


The information a linked service contains varies according to the resource. For example, to create a linked service for Azure Blob Storage, you provide information such as the name of the Azure subscription that owns the storage account, the name of the storage account, and the information necessary to authenticate against the storage account. To create a linked service to a different resource, such as Azure SQL Database, you specify the database server name, the database name, and the appropriate credentials.


The image below shows the graphical user interface provided by Azure Data Factory for creating linked services.


New linked service


Data store **Compute**


All **Azure** **Database** **File** **Generic protocol** **NoSQL** **Services and apps**



Azure Blob Storage



Azure Cosmos DB (MongoDB API)



Azure Cosmos DB (SQL API)



Azure Data Explorer (Kusto)


Azure Data Lake Storage


Azure Database for MariaDB


Azure Database for MySQL


Azure File Storage


Azure Synapse Analytics

Continue

New linked service (Azure Blob Storage)

i If the identity you use to access the data store only has permission to subdirectory instead of the entire account, specify the path to test connection. Please make sure your self-hosted integration runtime is higher than version 4.0 if connecting via self-hosted integration runtime.

Name *
AzureBlobStorageLinkedService

Description

Connect via integration runtime *
AutoResolveIntegrationRuntime

Authentication method
Account key

Connection string **Azure Key Vault**

Account selection method
☒ From Azure subscription ☐ Enter manually

Azure subscription

Storage account name *

Additional connection properties
[+ New](#)

Test connection
☐ To linked service ☒ To file path

/

Annotations
[+ New](#)

[Advanced](#)

Create **Back**

Connection successful
[Test connection](#) **Cancel**

Understand datasets

A dataset in Azure Data Factory represents the data that you want to ingest (input) or store (output). If your data has a structure, a dataset specifies how the data is structured. Not all datasets are structured. Blobs held in Azure Blob storage are an example of unstructured data.

A dataset connects to an input or an output using a linked service. For example, if you're reading and processing data from Azure Blob storage, you'd create an input dataset that uses a Blob Storage linked service to specify the details of the storage account. The dataset would specify which blob to ingest, and the format of the information in the blob (binary data, JSON, delimited text, and so on). If you're using Azure Data Factory to store data in a table in a SQL database, you would define an output dataset that uses a SQL Database linked service to connect to the database, and specifies which table to use in that database.

New dataset

In pipeline activities and data flows, reference a dataset to specify the location and structure of your data within a data store. [Learn more](#)

Select a data store

Search

All **Azure** Database File Generic protocol NoSQL Services and apps

Azure Data Explorer (Kusto) Azure Data Lake Storage Gen1 Azure Data Lake Storage Gen2

Azure Database for MariaDB Azure Database for MySQL Azure Database for PostgreSQL

Azure File Storage **Azure SQL Database** Azure SQL Database Managed Instance

Azure Search Azure Synapse Analytics (formerly SQL DW)

Continue

Set properties

Name
ProductTableDataSet

Linked service *
AzureSqlDatabaseLinkedService

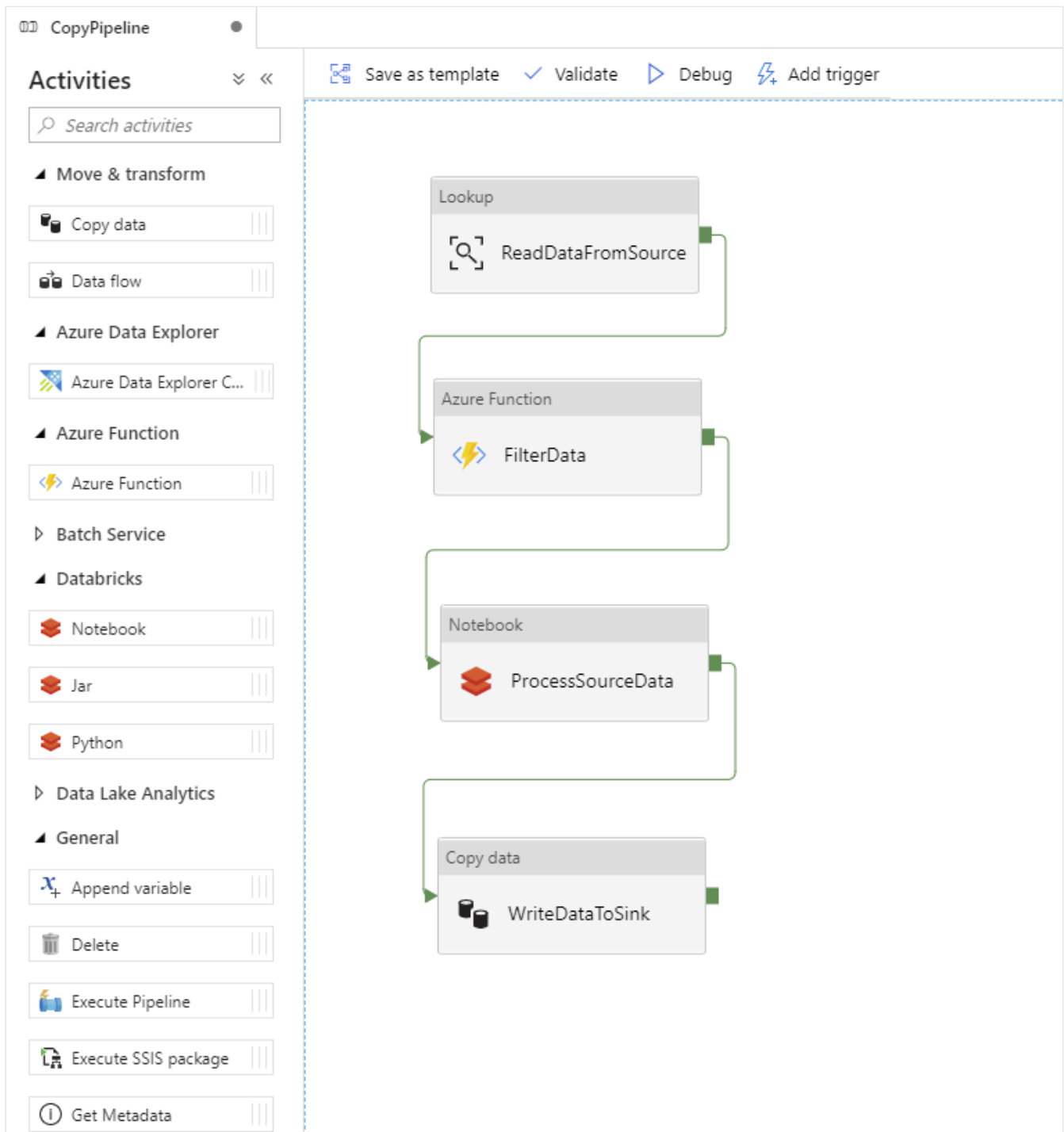
Table name
SalesLT.Product

Import schema
☒ From connection/store ☐ None

Understand pipelines

A pipeline is a logical grouping of activities that together perform a task. The activities in a pipeline define actions to perform on your data. For example, you might use a copy activity to transform data from a source dataset to a destination dataset. You could include activities that transform the data as it is transferred, or you might combine data from multiple sources

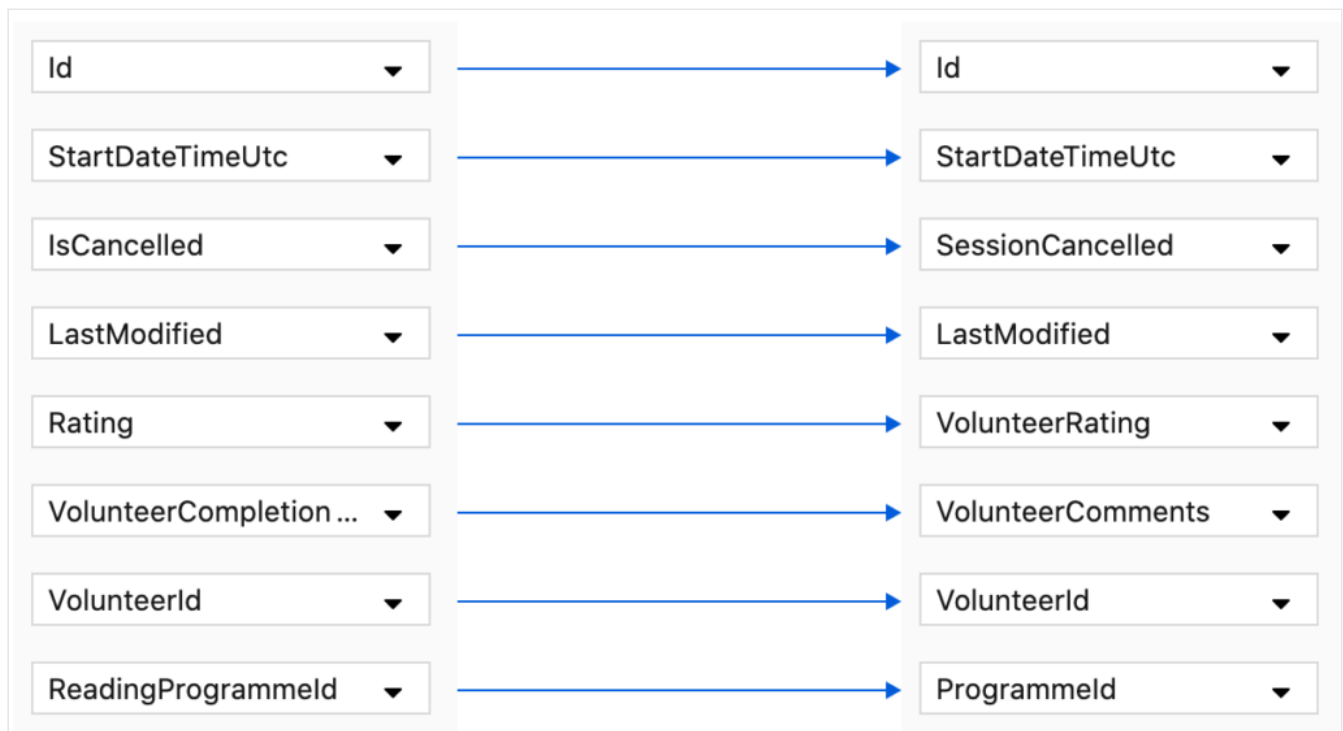
together. Other activities enable you to incorporate processing elements from other services. For example, you might use an *Azure Function* activity to run an Azure Function to modify and filter data, or an *Azure Databricks Notebook* activity to run a notebook that performs more advanced processing.



Pipelines don't have to be linear. You can include logic activities that repeatedly perform a series of tasks while some condition is true using a *ForEach* activity, or follow different processing paths depending on the outcome of previous processing using an *If Condition* activity.

Sometimes when ingesting data, the data you're bringing in can have different column names and data types to those required by the output. In these cases, you can use a mapping to

transform your data from the input format to the output format. The screenshot below shows the mapping canvas for the *Copy Data* activity. It illustrates how the columns from the input data can be mapped to the data format required by the output.



You can run a pipeline manually, or you can arrange for it to be run later using a trigger. A trigger enables you to schedule a pipeline to occur according to a planned schedule (every Saturday evening, for example), or at repeated intervals (every few minutes or hours), or when an event occurs such as the arrival of a file in Azure Data Lake Storage, or the deletion of a blob in Azure Blob Storage.

Ingest data using PolyBase

PolyBase is a feature of SQL Server and Azure Synapse Analytics that enables you to run Transact-SQL queries that read data from external data sources. PolyBase makes these external data sources appear like tables in a SQL database. Using PolyBase, you can read data managed by Hadoop, Spark, and Azure Blob Storage, as well as other database management systems such as Cosmos DB, Oracle, Teradata, and MongoDB.

Note

Spark is a parallel-processing engine that supports large-scale analytics.

PolyBase enables you to transfer data from an external data source into a table, as well as copy data from an external data source in Azure Synapse Analytics or SQL Server. You can also

run queries that join tables in a SQL database with external data, enabling you to perform analytics that span multiple data stores.

ⓘ Note

Azure SQL Database does not support PolyBase.

Azure Data Factory provides PolyBase support for loading data. For instance, Data Factory can directly invoke PolyBase on your behalf if your data is in a PolyBase-compatible data store.

Ingest data using SQL Server Integration Services

SQL Server Integration Services (SSIS) is a platform for building enterprise-level data integration and data transformations solutions. You can use SSIS to solve complex business problems by copying or downloading files, loading data warehouses, cleaning and mining data, and managing SQL database objects and data. SSIS is part of Microsoft SQL Server.

SSIS can extract and transform data from a wide variety of sources such as XML data files, flat files, and relational data sources, and then load the data into one or more destinations.

SSIS includes a rich set of built-in tasks and transformations, graphical tools for building packages, and the Integration Services Catalog database, where you store, run, and manage packages. A package is an organized collection of connections, control flow elements, data flow elements, event handlers, variables, parameters, and configurations, that you assemble using either the graphical design tools that SQL Server Integration Services provides, or build programmatically. You then save the completed package to SQL Server, the Integration Services Package Store, or the file system.

You can use the graphical SSIS tools to create solutions without writing a single line of code. You can also program the extensive Integration Services object model to create packages programmatically and code custom tasks and other package objects.

SSIS is an on-premises utility. However, Azure Data factory allows you to run your existing SSIS packages as part of a pipeline in the cloud. This allows you to get started quickly without having to rewrite your existing transformation logic.

The SSIS Feature Pack for Azure is an extension that provides components that connect to Azure services, transfer data between Azure and on-premises data sources, and process data stored in Azure. The components in the feature pack support transfer to or from Azure storage, Azure Data Lake, and Azure HDInsight. Using these components, you can perform large-scale processing of ingested data.

Ingest data using Azure Databricks

Azure Databricks is an analytics platform optimized for the Microsoft Azure cloud services platform. Databricks is based on Spark, and is integrated with Azure to streamline workflows. It provides an interactive workspace that enables collaboration between data scientists, data engineers, and business analysts.

Databricks can process data held in many different types of storage, including Azure Blob storage, Azure Data Lake Store, Hadoop storage, flat files, SQL databases, and data warehouses, and Azure services such as Cosmos DB. Databricks can also process streaming data. For example, you could capture data being streamed from sensors and other devices.

You write and run Spark code using *notebooks*. A notebook is like a program that contains a series of steps (called *cells*). A notebook can contain cells that read data from one or more data sources, process the data, and write the results out to a data store. The scalability of Azure Databricks makes it an ideal platform for performing complex data ingestion and analytics tasks.

Azure Data Factory can incorporate Azure Databricks notebooks into a pipeline. A pipeline can pass parameters to a notebook. These parameters can specify which data to read and analyze. The notebook can save the results, which the Azure Data Factory pipeline can use in subsequent activities.

Next unit: Describe data storage and processing

Continue >