# What is ADF? Level 1 of the Stairway to Azure Data Factory

**By** [**Andy Leonard**](http://www.sqlservercentral.com/Authors/Articles/Andy_Leonard/185646/)**, 2018/10/31**

**FROM:** <http://www.sqlservercentral.com/articles/Azure+Data+Factory/179513/?utm_source=SSC&utm_medium=pubemail>

### The Series

This article is part of the Stairway Series: [Stairway to Azure Data Factory](http://www.sqlservercentral.com/stairway/179562/)

Learn about the Azure Data Factory service for moving data in ETL flows.

Azure Data Factory – or ADF – is a data integration service built in the cloud. Like other data integration solutions, ADF is designed to perform ETL (Extract, Transform, and Load), ELT (Extract, Load, and Transform), and similar integration work.

## History

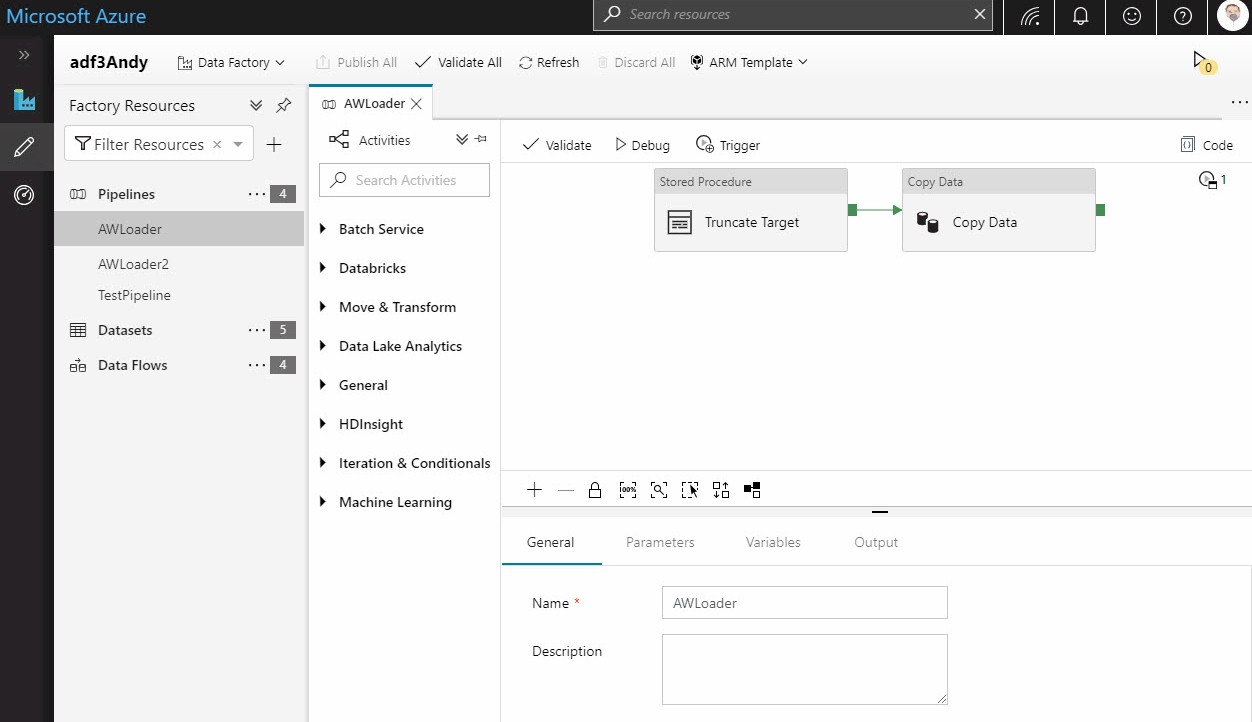
Introduced as a preview version in 2014 and general available in early 2015, Azure Data Factory version 1 initially supported a handful of Azure-hosted transformations:

* .Net
* Azure Machine Learning
* Hadoop
* Hive
* MapReduce
* Pig
* Spark
* Stored Procedures
* U-SQL (Data Lake)

Developers used [template-driven wizards](https://docs.microsoft.com/en-us/azure/data-factory/v1/data-factory-azure-copy-wizard) at the Azure Portal or [Microsoft Visual Studio](https://azure.microsoft.com/en-us/blog/azure-data-factory-visual-studio-extension-for-authoring-pipelines/) to build ADF v1 pipelines. A pipeline is a data-driven workflow comparable in function to a SQL Server Integration Services (SSIS) Control Flow. Pipelines supply orchestration.

## ADF v2

Azure Data Factory version 2 was released in preview in late 2017 and became generally available in Spring 2018. At the time of this writing, ADF v2 is the current version. ADF v2 includes an online editor for pipeline development, shown here:



The revision to Azure Data Factory between versions 1 and 2 are significant and some functionality available in ADF v1 is not available in ADF v2 (at the time of this writing). Differences in available functionality bely major changes to the architecture underlying Azure Data Factory.

See [Compare Azure Data Factory with Data Factory version 1](https://docs.microsoft.com/en-us/azure/data-factory/compare-versions) for a detailed comparison of Azure Data Factory versions 1 and 2.

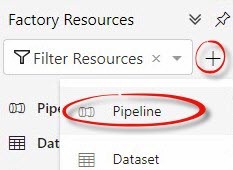
## What Can I Do with Azure Data Factory?

Azure Data Factory is designed to perform data integration or data engineering. “What is the difference between data integration and data engineering?” you may ask. That’s a fair question. As is often the case in technology, the answer lies with who you ask. Technologists of the NoSQL, Big Data, open source persuasion will likely use the term “data engineering” to describe the same functionality that SQL Server practitioners label “data integration.”

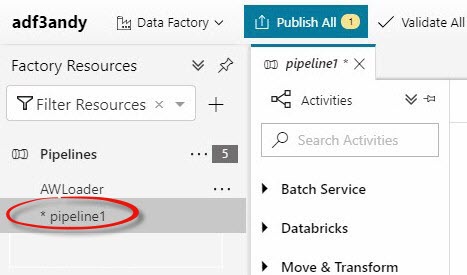
Regardless of semantics, ADF is built to move data. Since Azure Data Factory is a native cloud service, it is designed to move data between cloud-based sources and destinations.

## An Example

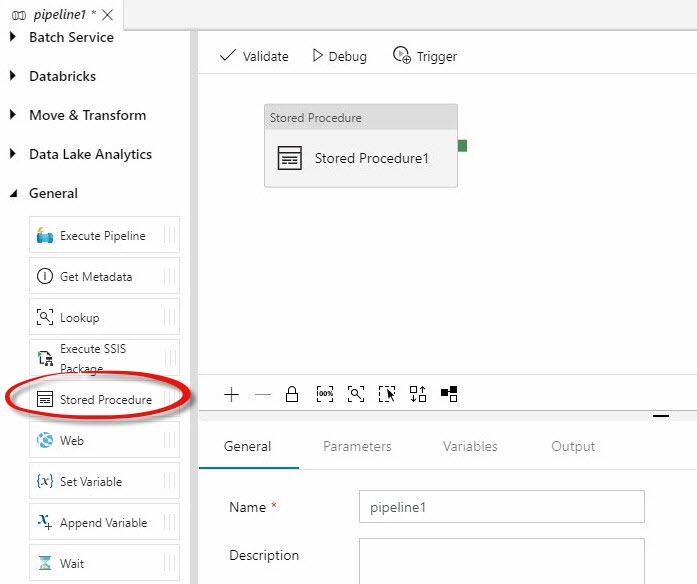
In this quick-start example (we will cover this example more thoroughly in an upcoming article), first click the plus sign (“+”) beside the Filter Factory Resources dropdown, then click Pipeline:



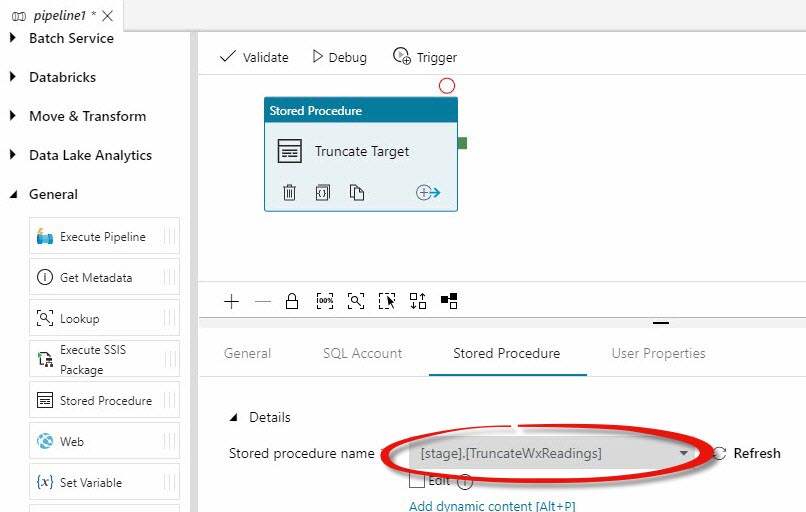
This creates a new pipeline. In this example, the new pipeline is named “pipeline1” in my data factory:



You can then add activities to the pipeline, such as the Stored Procedure activity:

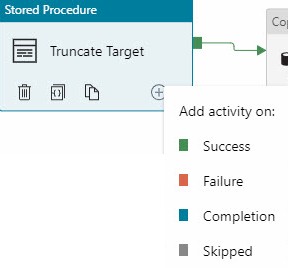


Click your new pipeline. Expand the General Activities group to the left of the pipeline editing surface. When you click the Stored Procedure activity, you can configure it using the tabbed configuration pane shown beneath the pipeline’s editing surface where you may rename the activity (I renamed mine “Truncate Target”), configure the SQL Account, and select the stored procedure:

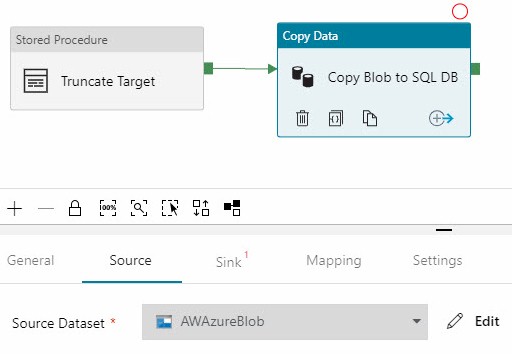


In my example, the [stage].[TruncateWxReadings] stored procedure truncates a table named [stage].[WxReadings] that is hosted in an Azure SQL DB instance. This is part of a data integration design pattern called Truncate and Load (or, sometimes, “Whack ‘N’ Load”).

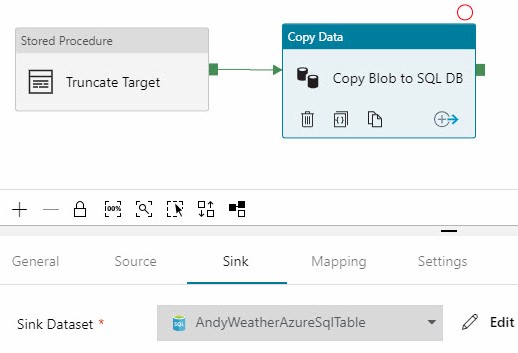
Expand the Move & Transform Activities group and drag a Copy Data activity onto the surface. Click the green rectangle on the Stored Procedure activity named “Truncate Target”, hold the mouse and drag it over the new Copy Data activity, and then release the mouse. Alternately, click the Add Output icon (http://www.sqlservercentral.com/Images/36179.png) on the Truncate Target stored procedure activity to add an output other than On-Success:



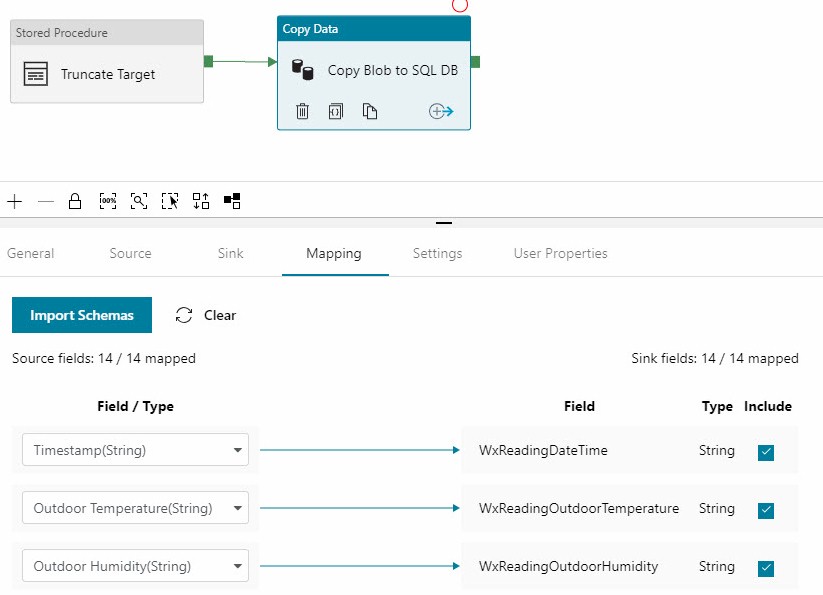
Rename the Copy Data activity, if desired. Configure the Source Dataset on the Source tab (mine is a CSV file stored in Azure Blob Storage:



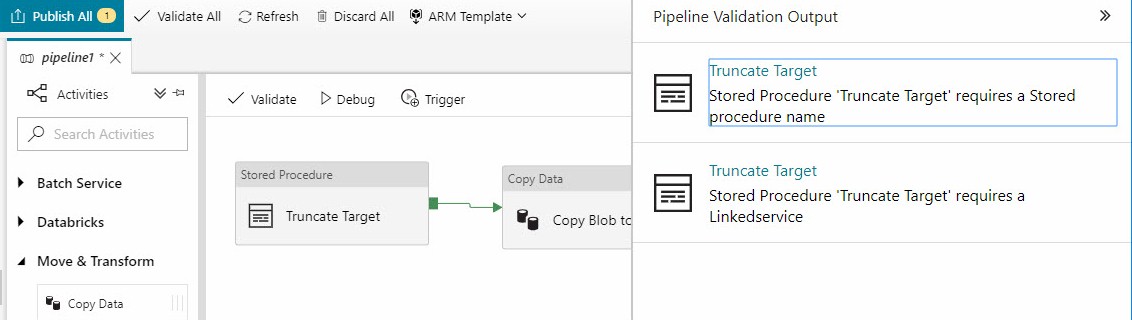
Configure the Sink – or Destination – next:



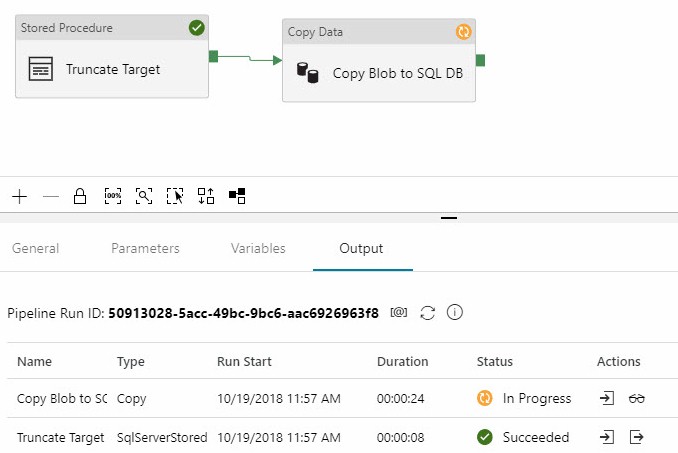
The next step in our quick-start example is to configure the Mapping. Importing the Schemas is handy and there’s a button for that:



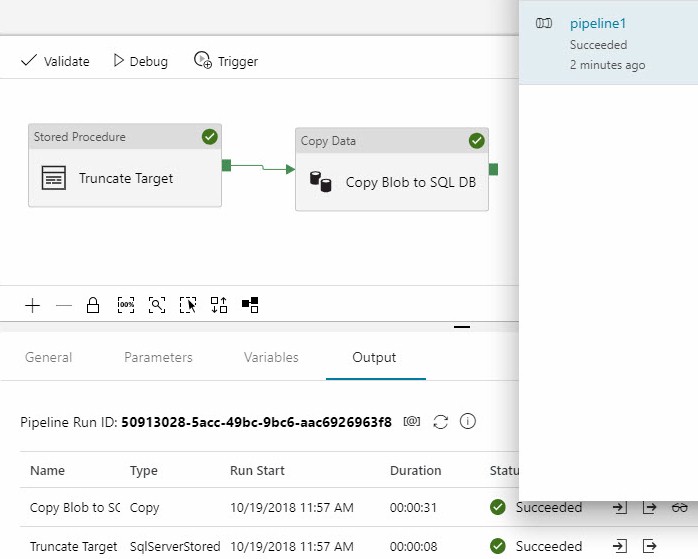
Click Validate in the pipeline menu to make sure the pipeline is ready to execute. Any errors or missing configurations are displayed:



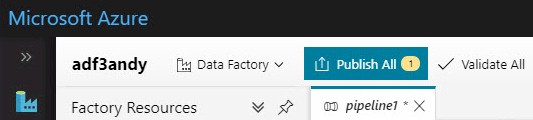
Once the pipeline is valid you may perform a test execution, which you accomplish by clicking Debug in the pipeline menu. The Output tab displays status of each activity:



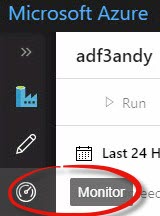
If all goes as planned, the pipeline succeeds:



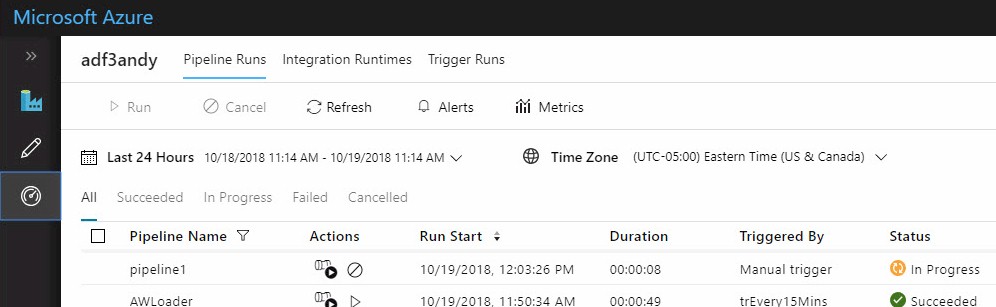
One way to publish (deploy) an ADF pipeline is to click the “Publish All” button:



Once published, pipeline execution may be triggered manually or via a Trigger (which we will cover in another article). You can view the progress of a manually-triggered pipeline execution on the Monitor page:



A manually-triggered pipeline execution will show “In Progress” until it completes and succeeds or fails:



# Conclusion

Azure Data Factory is an important component in the Azure cloud eco-system. Data integration / engineering remains the largest individual component of most data science and analytics efforts. Still relatively new, ADF offers compelling functionality.