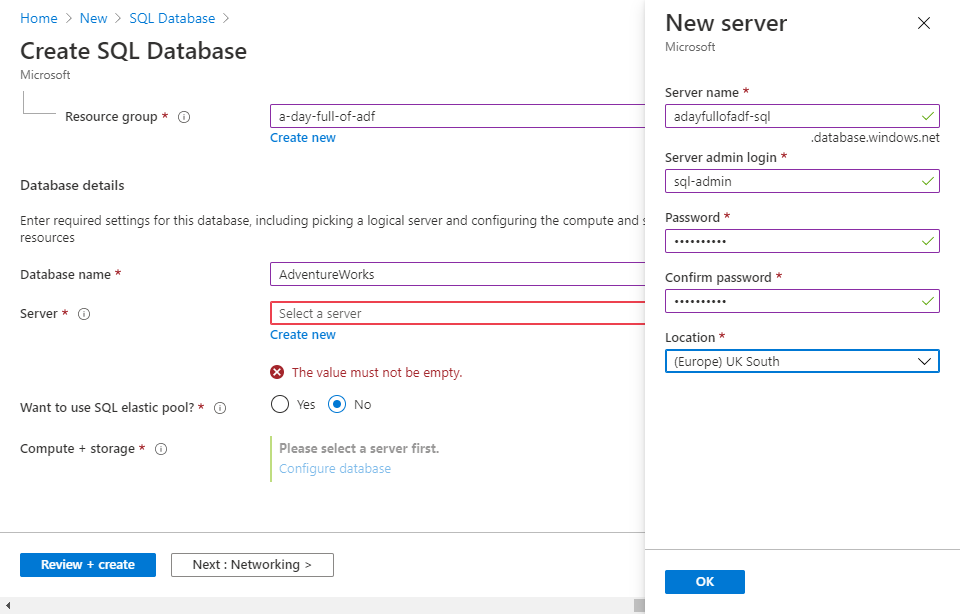
Lab 2 – Build a Copy data pipeline

In this lab you will create a Copy data pipeline to copy data from a source SQL database into your data lake.

# Lab 2.1 – Create a source SQL database

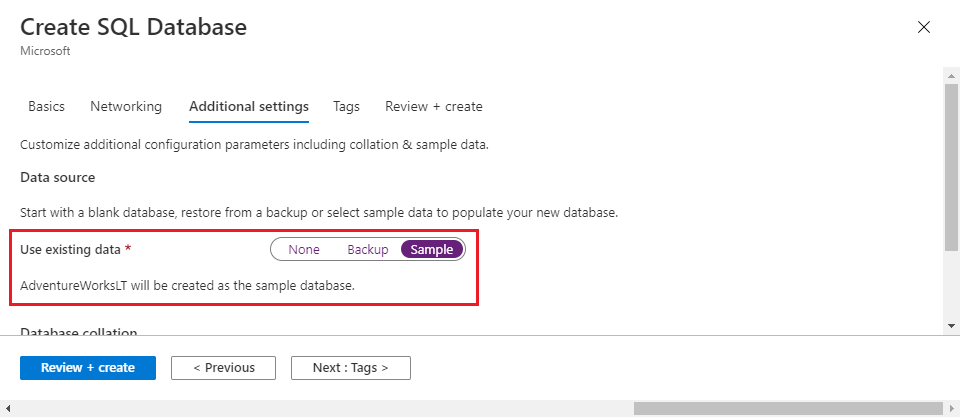
For this lab, you’ll be copying data from AdventureWorks OLTP database. Start by creating a copy in your resource group – this will play the role of a source business system for ADF data processing.

1. In the Azure portal, click “Create a resource” and search for “SQL Database”. Click “Create” on the overview screen.
2. Complete the **Basics** tab like this:
   * Choose your subscription and the resource group you created in Lab 1.1.
   * Enter a name for your database
   * Under the “Server” dropdown, click “Create new”
   * On the “New server” blade, provide a globally-unique server name, an admin login name and an admin password. **Make a note of the login and password**. Choose the same location you specified for your resource group, then click “OK”

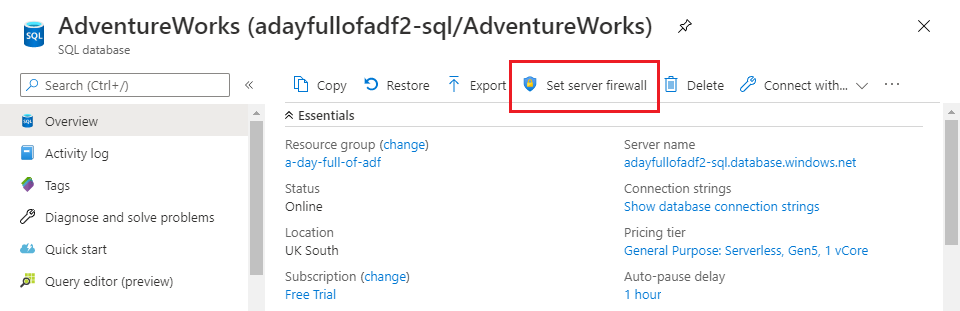


* + Under “Compute + storage”, click “Configure database”, then change the compute tier from “Provisioned” to “Serverless”. Serverless SQL services are automatically paused after a period of inactivity (by default 1 hour) – this is rarely appropriate for production environments but is fine for lab work, and a lot cheaper! Click “Apply”.

1. Skip forward past the Networking tab to the **Additional settings** tab. Under “Data source”, change the “Use existing data” option to “Sample” – the message “AdventureWorksLT will be created as the sample database” is displayed.



1. Click “Review + create”, then “Create”. When the server and database have finished deploying (this may take a few minutes), click on “Go to resource”.
2. In the database blade, click the menu bar’s “Set Server firewall” button.



1. On the “Firewall settings” blade:
   * Click “+ Add client IP” in the menu bar to add your computer’s IP address to the server firewall.
   * Set “Allow Azure services and resources to access this server” to **Yes**, to allow Azure Data Factory (and other Azure services) through the firewall.
   * Click “Save”.

# Lab 2.2 – Create linked service and datasets

To recap, you have created:

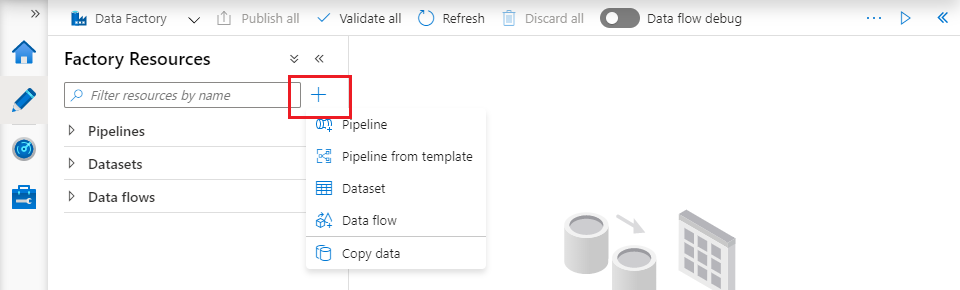
* A SQL database to provide a data source
* A data lake to act as your data sink (Azure Data Factory refers to copy targets/destinations as “sinks”)
* An instance of Azure Data Factory to copy data from source to sink

You’ve already created a linked service connection to your data lake in ADF – now you need to create one for the SQL database.

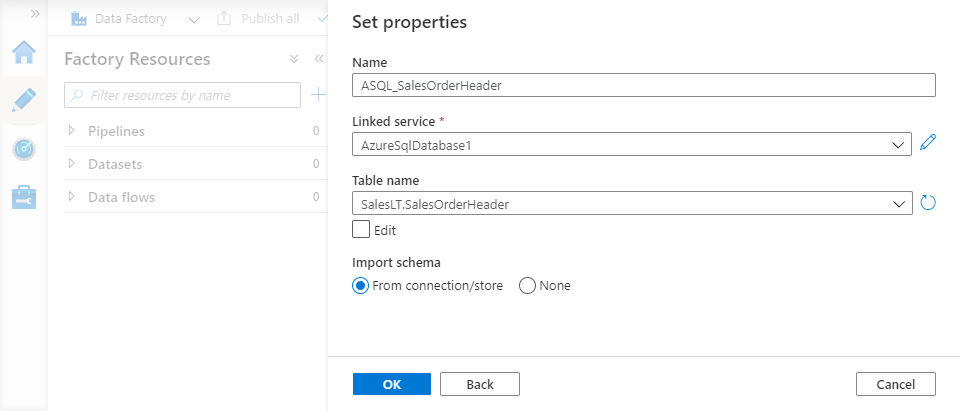
1. In the ADF UX, open the Management Hub and select “Linked services” from the “Connections” section of its sidebar.
2. In the main Linked services pane click “+ New”.
3. Search for “SQL”, then choose “Azure SQL Database” and click “Continue”.
4. Configure linked service details on the “New linked service (Azure SQL Database)” blade:
   * Give it a name.
   * Choose the Azure subscription containing your lab resources.
   * Choose your SQL Server from the “Server name” dropdown, then your new database from the “Database name” dropdown.
   * For “Authentication type”, choose “SQL authentication”, then enter the admin login name and password you noted down in lab 2.1
5. Click “Test connection” at the bottom of the blade. If you have configured your SQL Server and linked service correctly, you will receive a “Connection successful” message.
6. Click “Create” to create the linked service. The linked service is published automatically this time, so that the database username and password aren’t stored in your ADF UX session (or in your Git repo if you’re using one).

Linked services represent connections to external systems – in this example a SQL database. Data stored by those services (in this case as database tables) is represented by an ADF dataset.

1. Open the ADF authoring canvas by clicking on the pencil button (on the far left, two icons above the Management Hub button).
2. In the “Factory resources” sidebar, click the “+” button to the right of “Filter resources by name”, then choose “Dataset”.



1. Search for “SQL”, then choose “Azure SQL Database” and click “Continue”.
2. Give the dataset a name, then choose your Azure SQL Database from the “Linked service” dropdown. Finally, choose a table from the “Table name” dropdown and click “OK”.

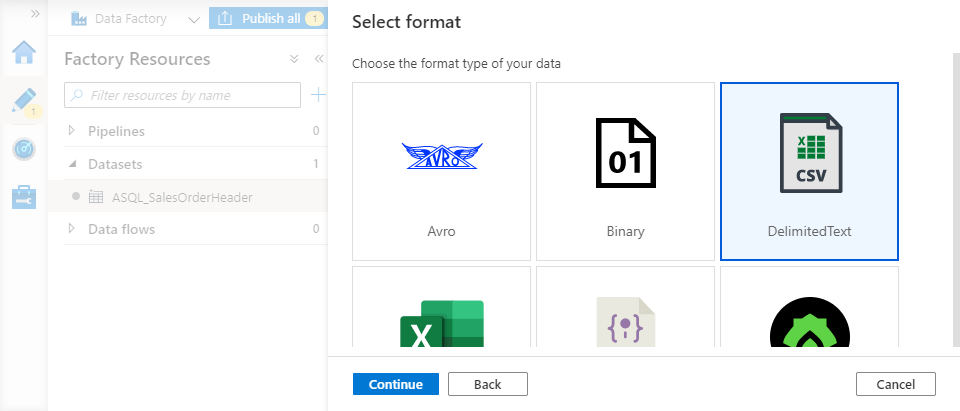


1. Save your changes:

* if you’re using Git, click “Save all”
* otherwise click “Publish all”.

You need a similar dataset to represent the sink file for the Copy data operation.

1. Use the “+” button to create a second dataset. This time search for “data lake”, then choose “Azure Data Lake Storage Gen2” and click “Continue”.
2. For file formats, you need also to specify a file type. Choose “DelimitedText” and click “Continue”.



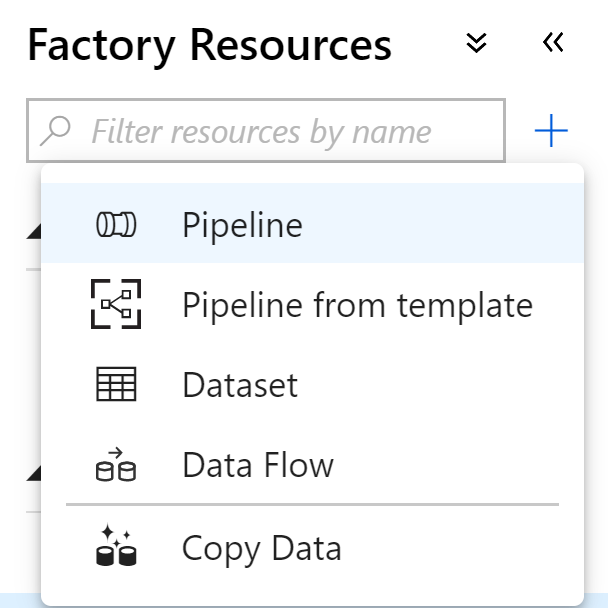
1. On the “Set properties” blade:
   * Give the dataset a name
   * Choose your Azure Data Lake Storage linked service
   * To the right of the three “File path” fields, click the folder icon and browse to your container’s “Raw” directory. Click “OK”.
   * Ensure that the “First row as header” checkbox is **ticked**, then click “OK”.
2. Save your changes.

# Lab 2.3 – Create a data factory pipeline

## Lab 02.B – Create a Data Factory Pipeline

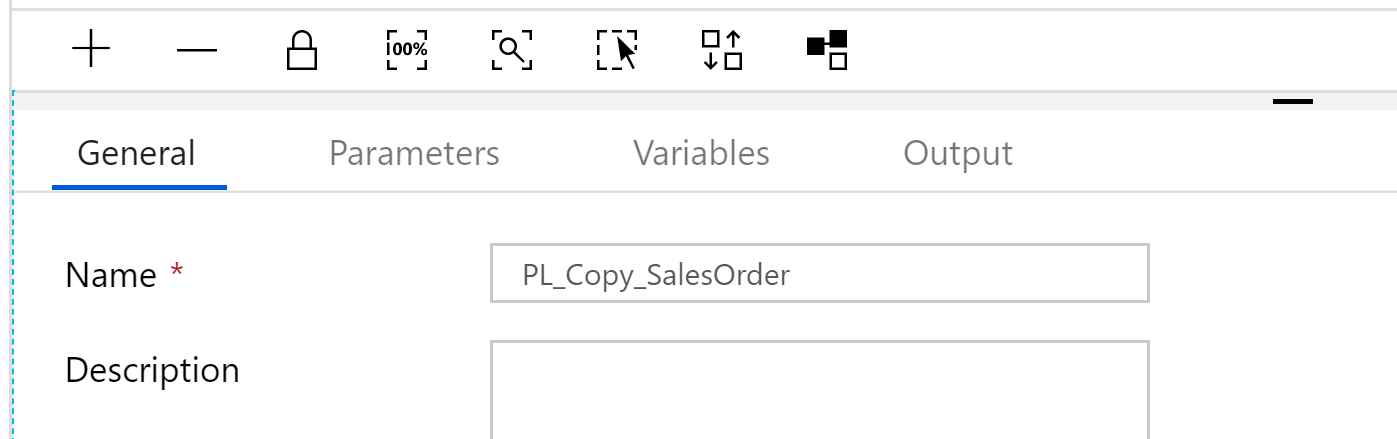
We now have our data lake ready to receive data, and a SQL database we want to get data from. Now we can create the data factory pipeline to extract the data from a table and land it into our lake.

1. Click on the “create pipeline” button to create a new data factory pipeline

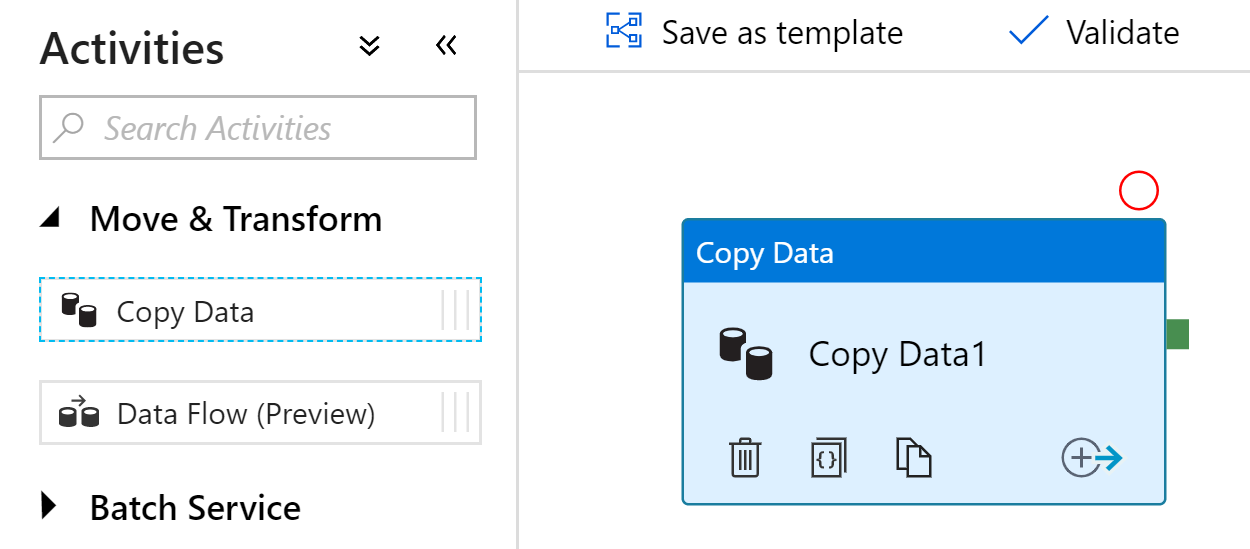


This creates a blank pipeline that we can start working with.

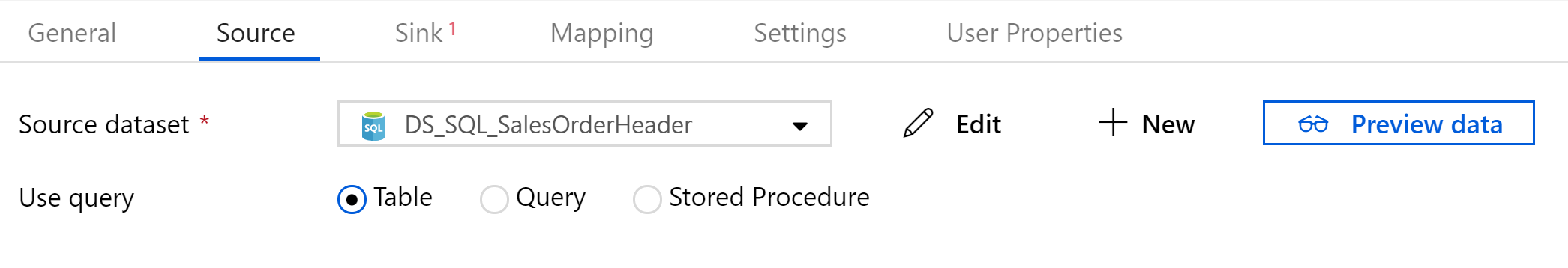
1. Give the pipeline a reasonable name:



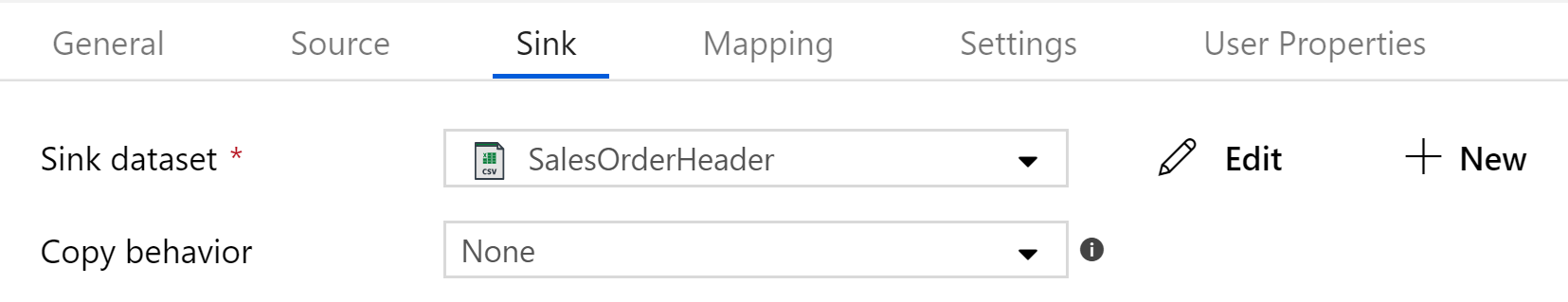
1. Open the “Move and Transform” activity menu and drag a “Copy Data” activity onto our pipeline canvas:



1. Give the activity a sensible name. Then click “Source” tab to set up where we’re copying data from. Just select the SQL dataset we created earlier.



1. Next click on “Sink” to set up the destination using the CSV dataset:

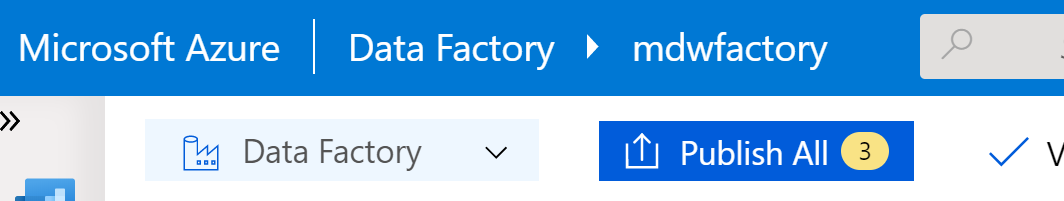


That’s everything set up for our very basic copy pipeline. We have told it where to get the data from, and how to connect. We have configured the destination and where it should place it, and we have created a workflow that to perform that work. Now, like any good developer, we just need to test it.

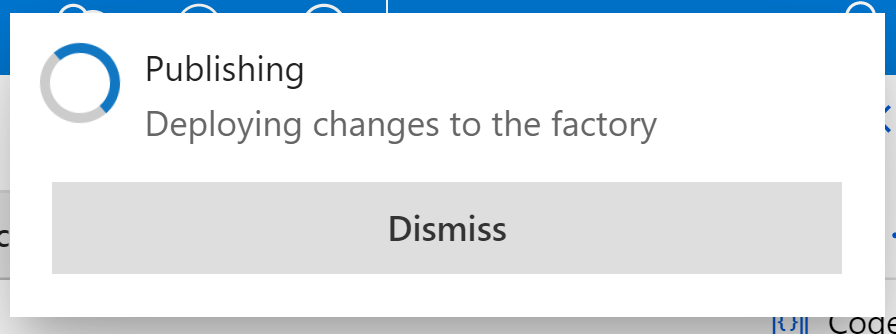
## Lab 02.C – Trigger & Review Our Pipeline

So far, the changes we’ve made have not yet been deployed. We want to try out the pipeline in the real world, so let’s publish our changes to our development data factory.

1. Click the “Publish” button to deploy our changes:

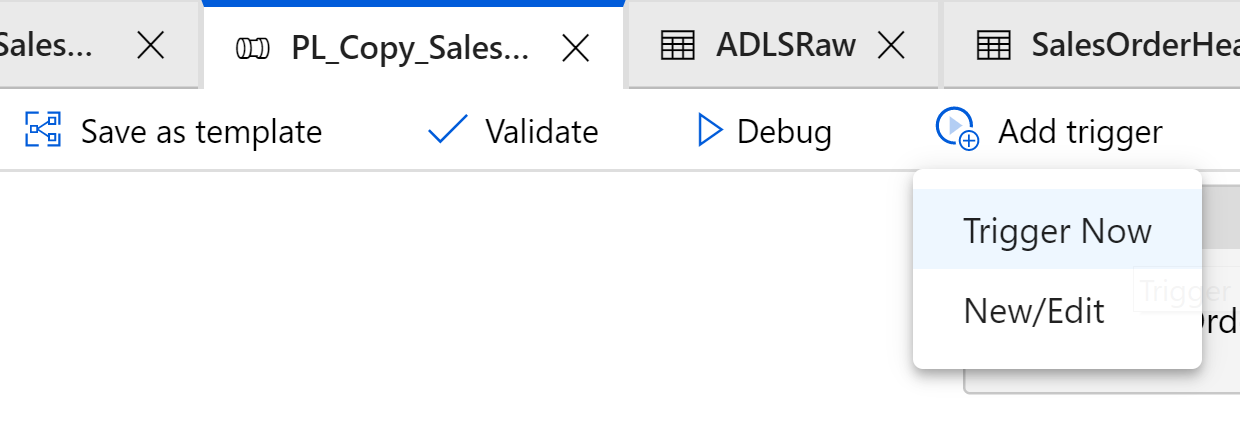


It’ll take a moment to deploy your changes:

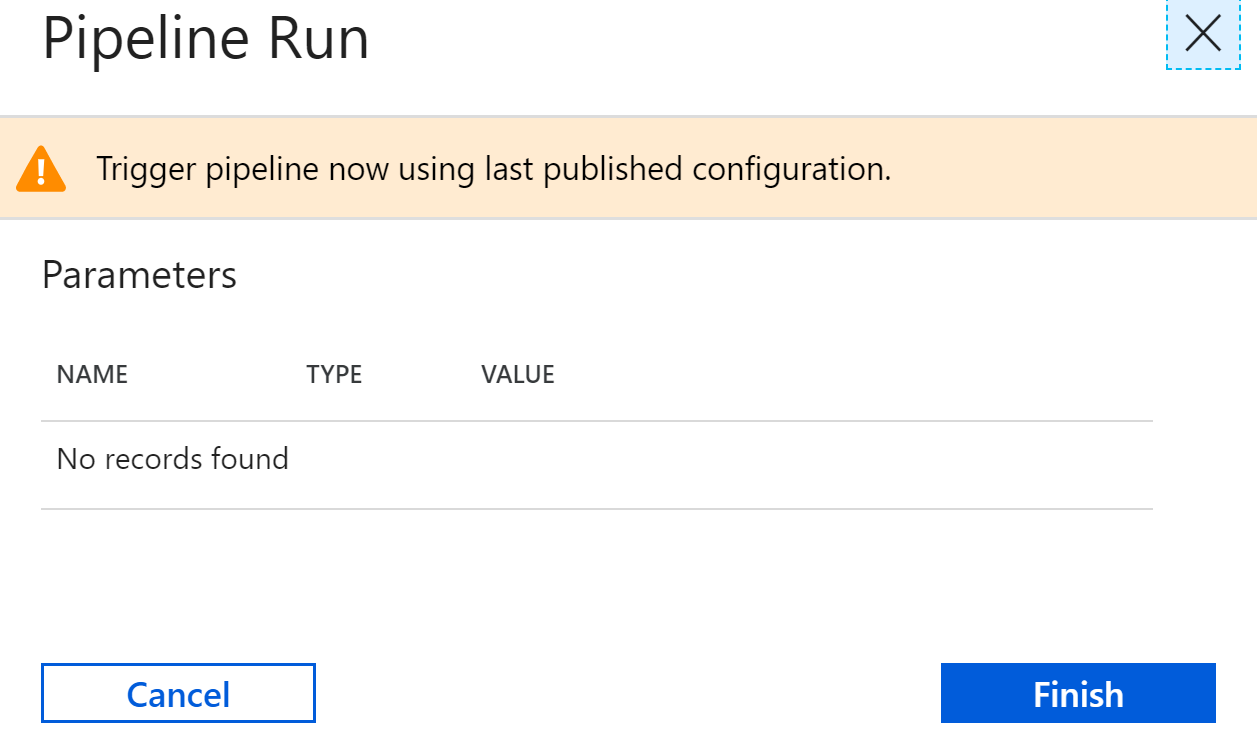


This is a useful step as it will also validate your configuration and make sure nothing has been done incorrectly.

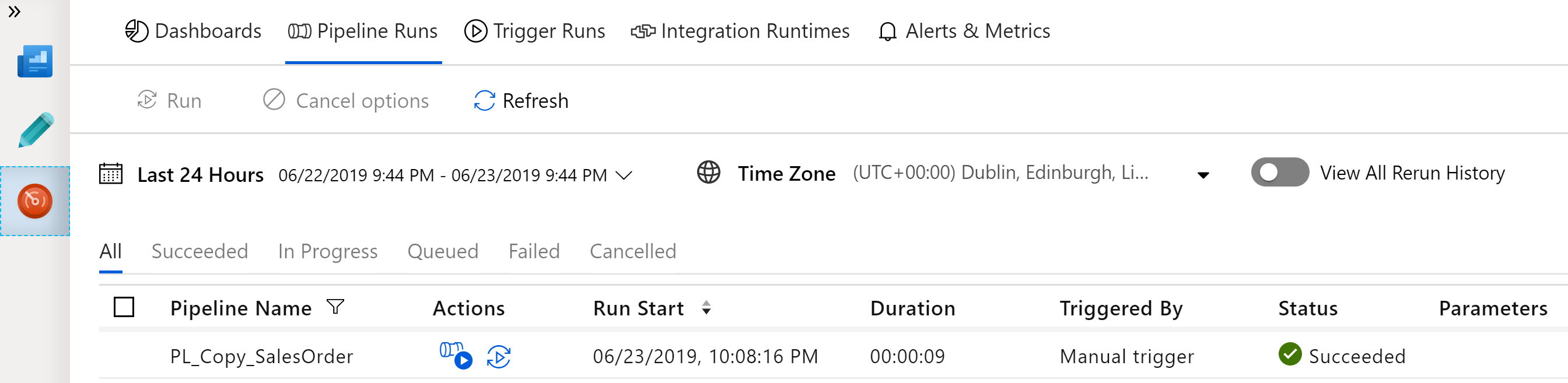
1. Next, let’s test out the pipeline and see if it works. Within our pipeline, click on the “Add Trigger” button and select “Trigger Now” to perform a manual execution of the pipeline.



This will pop up a confirmation screen:



1. Click Finish to complete the trigger.
2. Finally, we can click on the “Monitor” button and go and see our executing pipeline. After a minute or so, you should see a successful pipeline like this one:



You might also want to go and check your data lake store, where you should see a file ready and waiting!