Azure Data Factory

Lab 02 – Building a Copy Data Pipeline

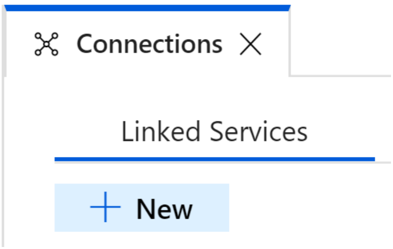
You will need:

- Access to our AdventureWorks Database (details below)
- An Azure subscription with a Data Factory & ADLS Gen 2 (from Lab 1)

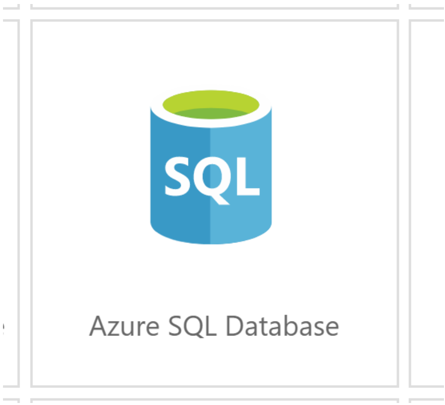
Lab 02.A – Create the Linked Services & Datasets

First things first, we'll get you linked up to our public copy of AdventureWorks. This is the OLTP version of the database, stored as an Azure SQLDB. We restrict the firewall to our events, but it's very easy to create your own version. There are some instructions for that as a separate lab!

1. Open up the Data Factory we created in Lab 1 and navigate to the connections tab. Click the new button to add a new linked service.



2. We've stored AdventureWorks as a SQL DB, so select that from the list and click continue.



3. Give the linked service a sensible name – we use the pattern LS_TYPE_NAME, so LS_SQL_Adventureworks.

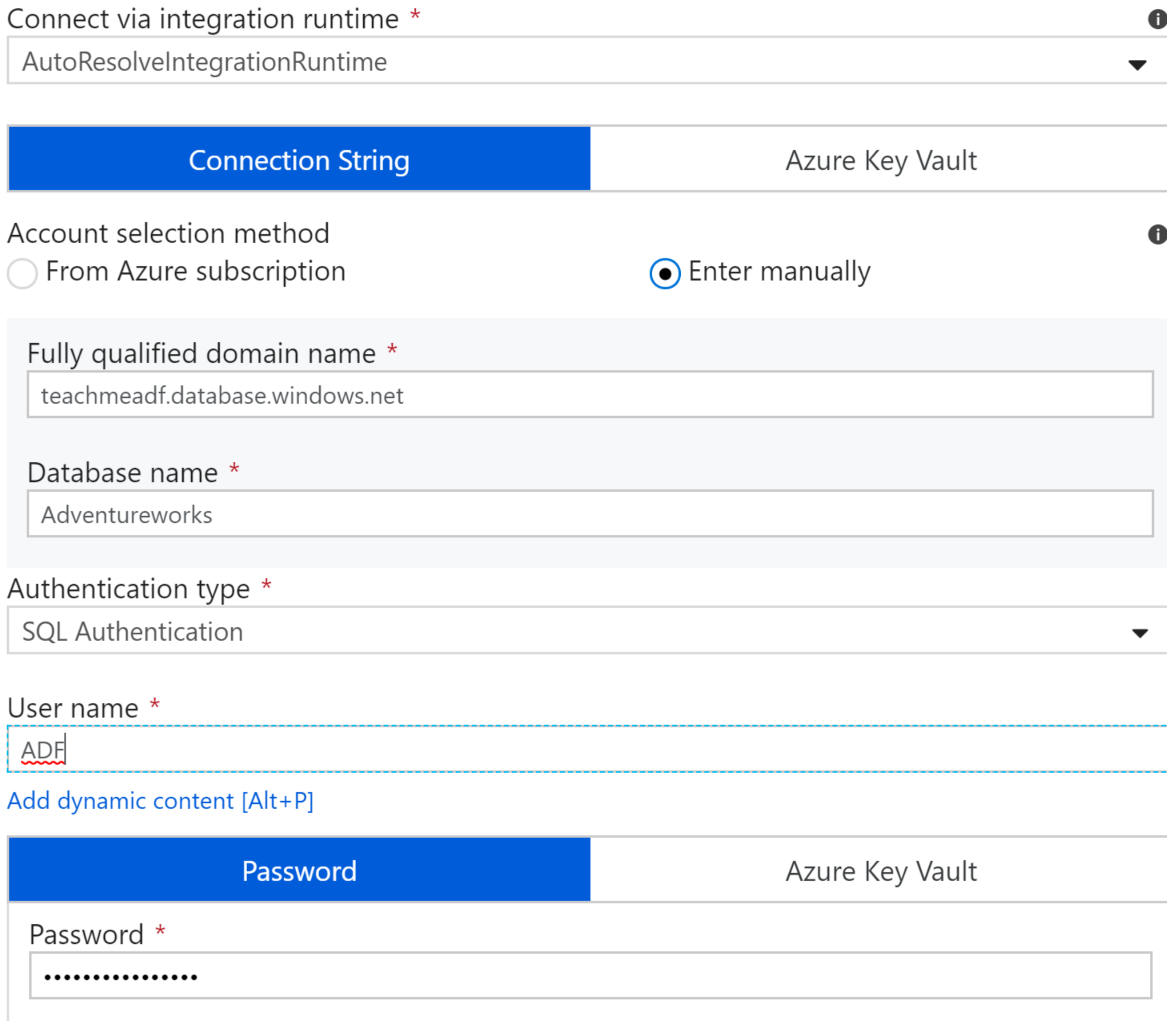
   The connection details to the shared database are as follows:

   Server: teachmeadf.database.windows.net

   Username: ADF

   Password: `@zureDataFactory`

Your completed linked service should look something like:



Connect via integration runtime *

AutoResolveIntegrationRuntime

Account selection method

○ From Azure subscription ● Enter manually

Fully qualified domain name *

teachmeadf.database.windows.net

Database name *

Adventureworks

Authentication type *

SQL Authentication

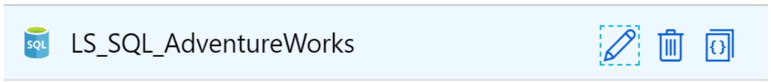
User name *

ADF

Password *

•••••••••••••••

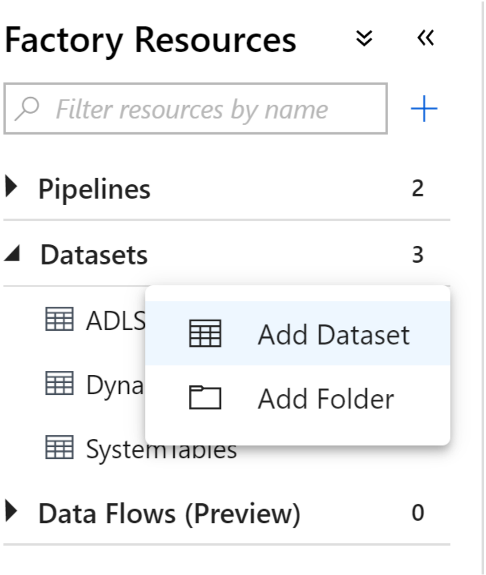
4. Test the connection, ensure it succeeds then click save. You should now see a completed Linked Service ready to be used.



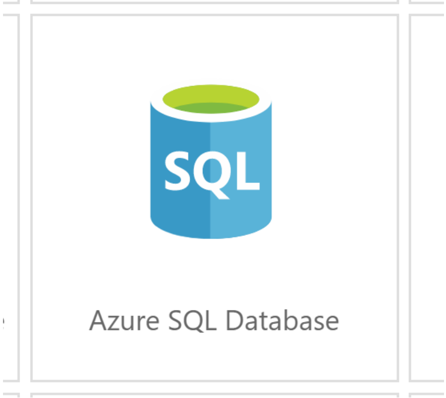
LS_SQL_AdventureWorks

However, this is simply a connection to the server. To perform a copy, we need a dataset that tells us which table to copy.

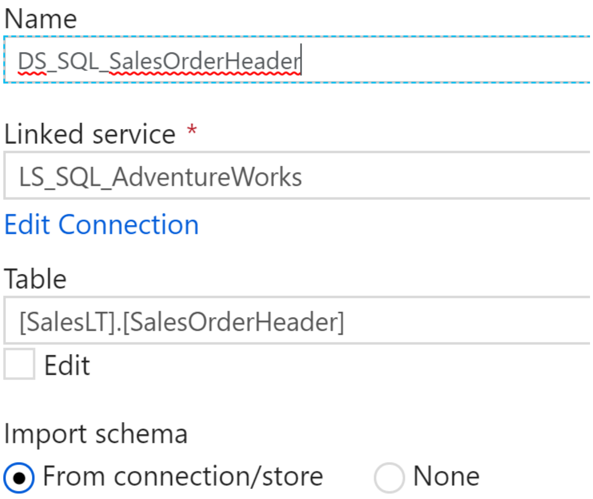
5. Expand the Datasets menu and click the “Add Dataset” button:



6. Select SQL Database once again



7. Complete the properties, select a table from adventureworks and click Continue



Name

DS_SQL_SalesOrderHeader

Linked service *

LS_SQL_AdventureWorks

Edit Connection

Table

[SalesLT].[SalesOrderHeader]

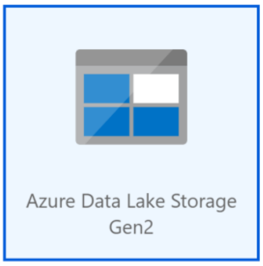
□ Edit

Import schema

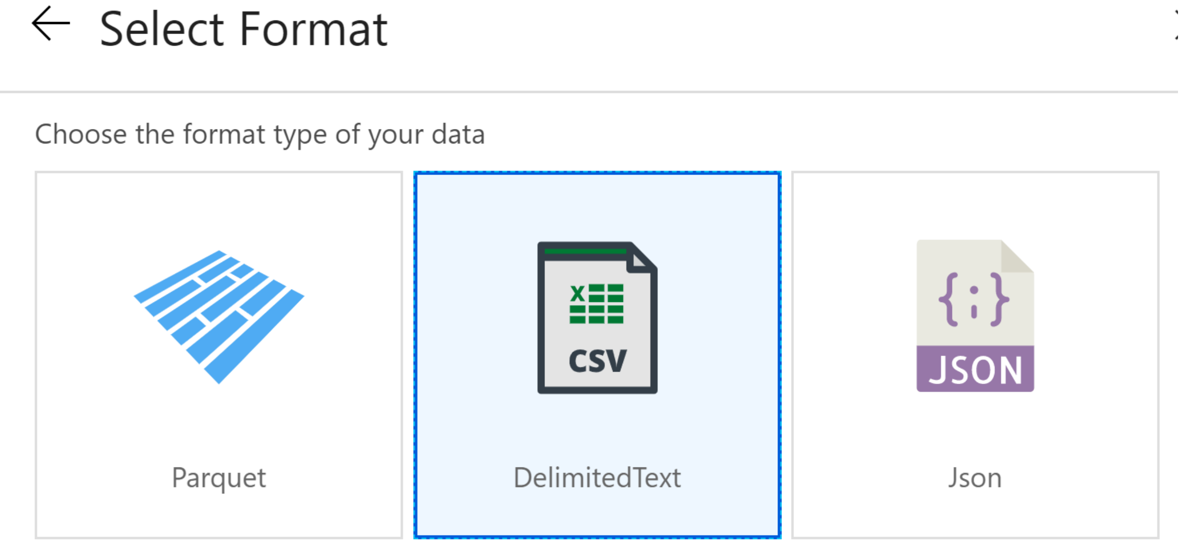
● From connection/store ○ None

And we now have a dataset created, pointing at a specific table within Adventureworks.

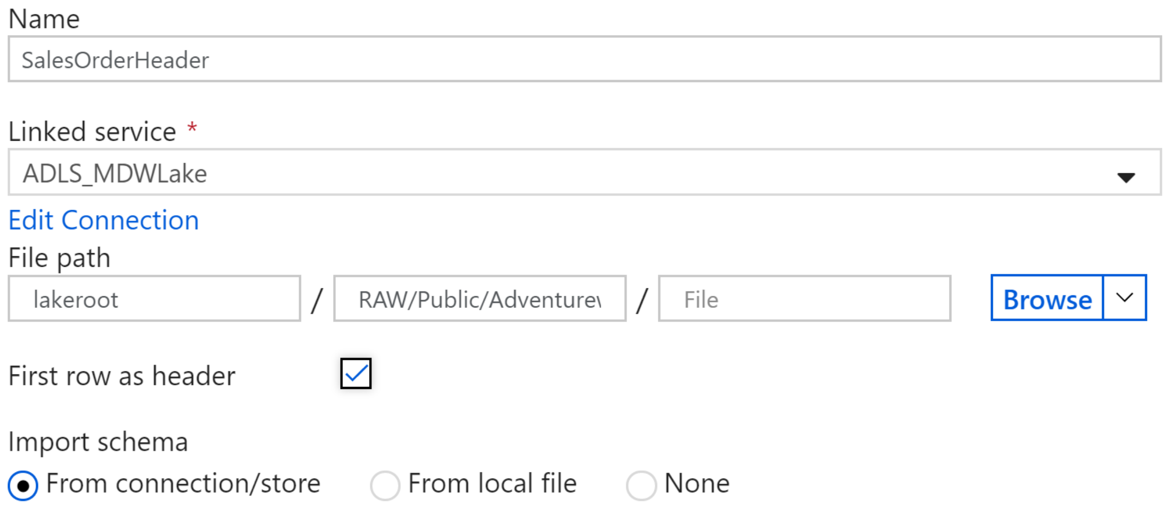
8. We need to do the same for the destination. Click and add a new Dataset, this time choosing "Data Lake Store Gen 2" as the type.



9. For flat formats, we also choose the file type. For now, select CSV:



10. Name the new dataset and select the ADLS Gen 2 linked service that we created in Lab 1. This will then open up file path – you can use the "browse" button to navigate your lake and select a destination file path



Name: SalesOrderHeader

Linked service *: ADLS_MDWLake

Edit Connection

File path: lakeroot / RAW/Public/Adventure\ / File

Browse

First row as header ☑

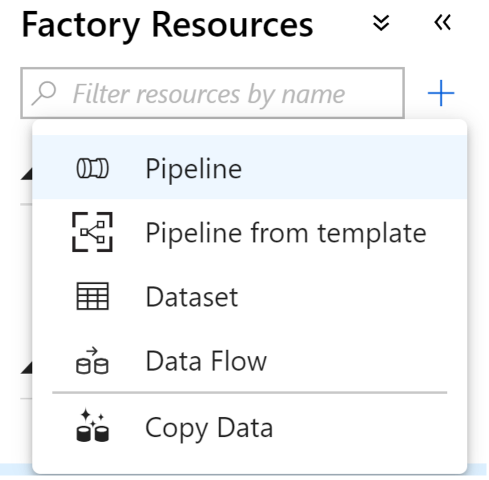
Import schema

● From connection/store ○ From local file ○ None

Lab 02.B – Create a Data Factory Pipeline

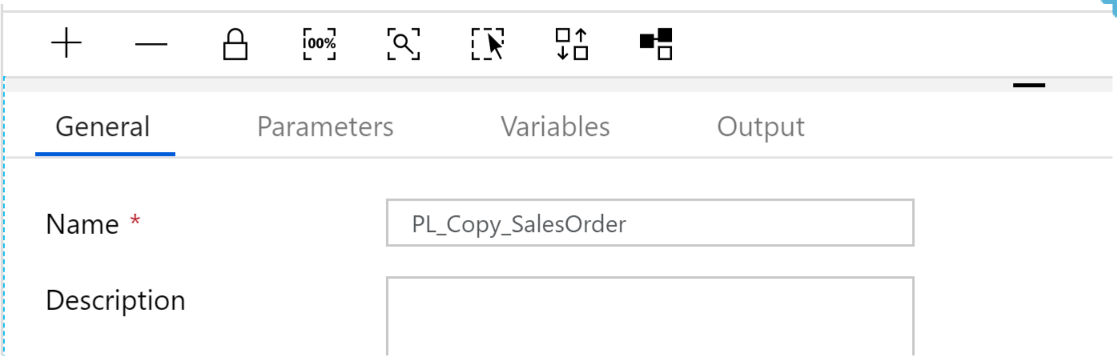
We now have our data lake ready to receive data, and a SQL database we want to get data from. Now we can create the data factory pipeline to extract the data from a table and land it into our lake.

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.

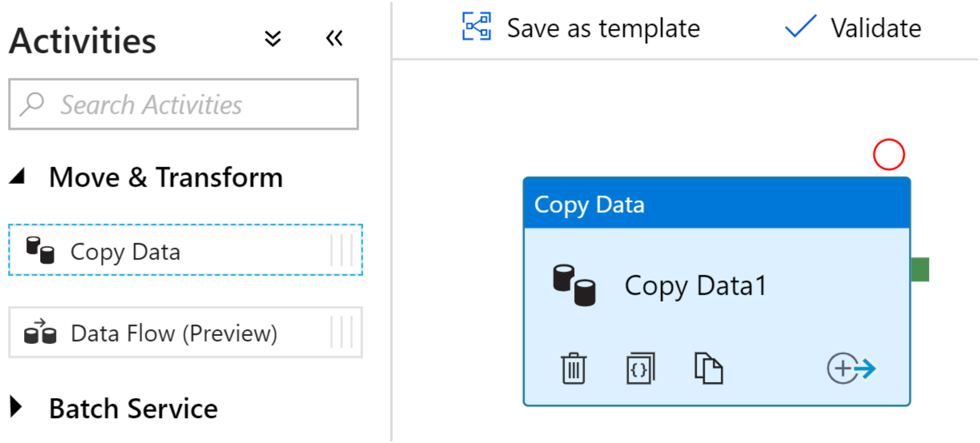
2. Give the pipeline a reasonable name:



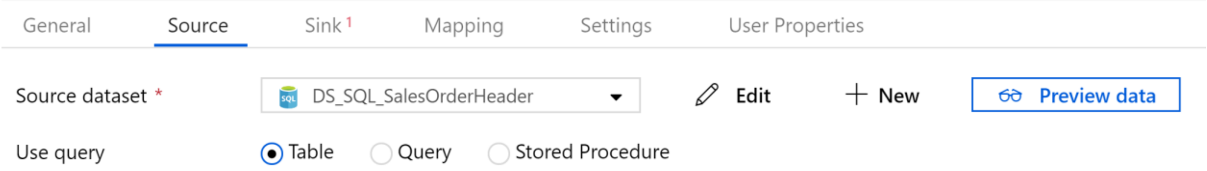
Name * PL_Copy_SalesOrder

Description

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



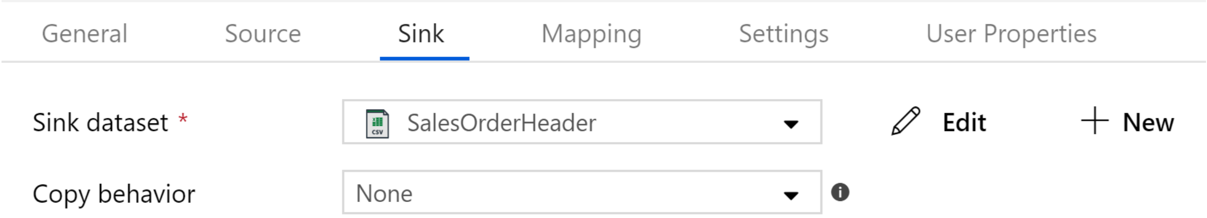
4. Give the activity a sensible name. Then click “Source” tab to set up where we’re copying data from. Just select the SQL dataset we created earlier.



Source dataset * DS_SQL_SalesOrderHeader

Use query ● Table ○ Query ○ Stored Procedure

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



Sink dataset * SalesOrderHeader

Copy behavior None

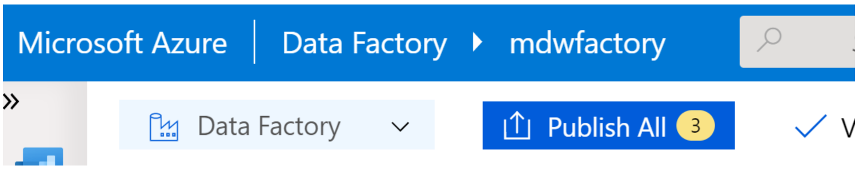
Lab 02 – Building a Copy Data Pipeline

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.

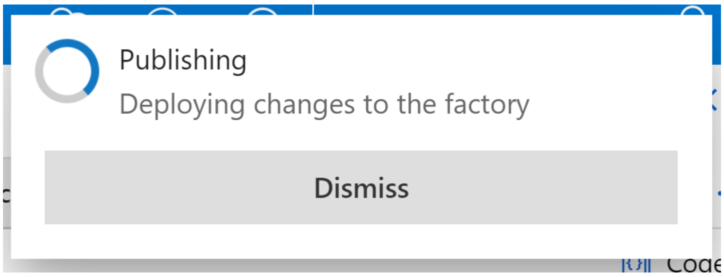
Lab 02.C – Trigger & Review Our Pipeline

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:

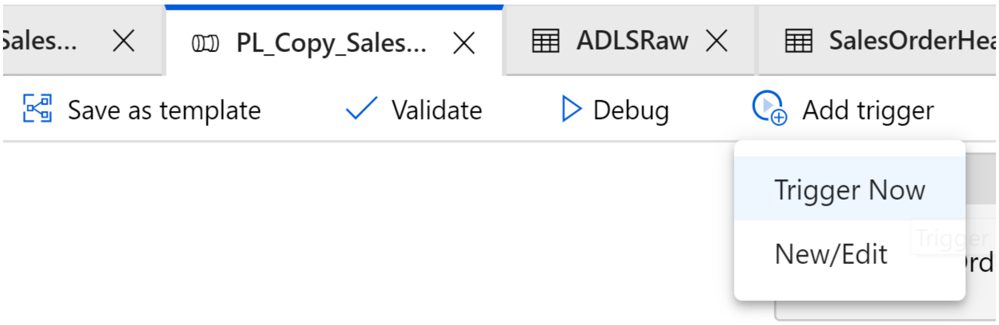


It'll take a moment to deploy your changes:

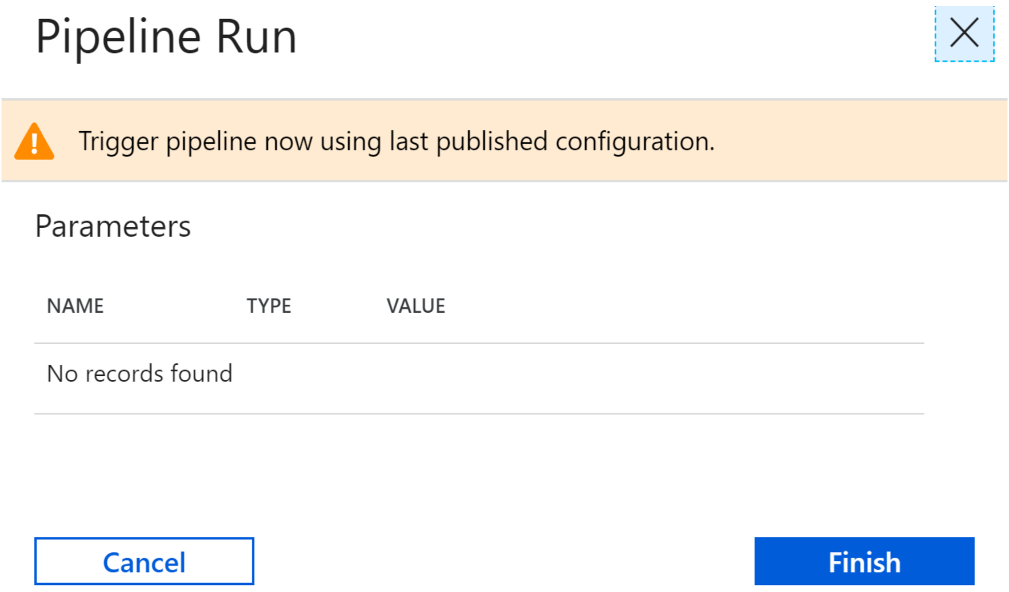

Publishing
Deploying changes to the factory

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

2. 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:



Pipeline Run

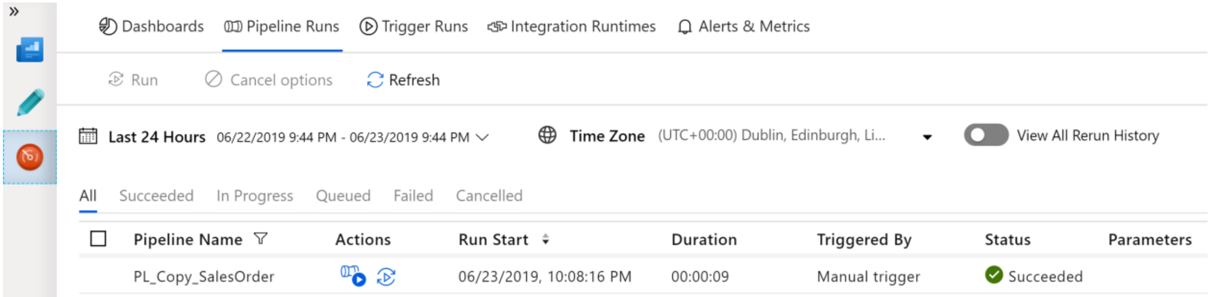
Trigger pipeline now using last published configuration.

Parameters

NAME	TYPE	VALUE
No records found		

3. Click Finish to complete the trigger.

4. 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:



Last 24 Hours 06/22/2019 9:44 PM - 06/23/2019 9:44 PM

Time Zone (UTC+00:00) Dublin, Edinburgh, Li...

View All Rerun History

□	Pipeline Name	Actions	Run Start	Duration	Triggered By	Status	Parameters
	PL_Copy_SalesOrder		06/23/2019, 10:08:16 PM	00:00:09	Manual trigger	✓ Succeeded	

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