

**Lab Manual- Azure Data Bricks Provisioning and Data Ingestion Part1**

**Prepared for**:

**Date:** 18th March 2022

**Prepared by:**

Document Name: Lab Manual **Document Number** AZLabn990

**Contributor:**

Contents

[1. Introduction 3](#_Toc102632905)

[2. Exercise 1 – Ingest Data from GitHub 3](#_Toc102632906)

[3. Exercise: Run code in the 1-Delta-Architecture notebook 5](#_Toc102632907)

[1. Datasets Used 6](#_Toc102632908)

[2. Define Schema 6](#_Toc102632909)

[3. Define Streaming Load from Files in Blob 7](#_Toc102632910)

[4. WRITE Stream using Delta Lake 7](#_Toc102632911)

[5. Load Static Lookup Table 7](#_Toc102632912)

[6. Create QUERY tables (aka "silver tables") 8](#_Toc102632913)

[7. Create QUERY tables (aka "silver tables") 8](#_Toc102632914)

[8. See list of active streams. 9](#_Toc102632915)

[9. Gold Table: Grouped Count of Events 9](#_Toc102632916)

[10. Gold Table: Grouped Count of Events 10](#_Toc102632917)

[11. Materialized View: Windowed Count of Hourly gt Events 10](#_Toc102632918)

# Introduction

Systems are working with massive amounts of data in petabytes or even more and it is still growing at an exponential rate. Big data is present everywhere around us and comes in from different sources like social media sites, sales, customer data, transactional data, etc

[Apache Spark](https://spark.apache.org/) is an open-source, fast cluster computing system and a highly popular framework for big data analysis. This framework processes the data in parallel that helps to boost the performance. It is written in [Scala](https://spark.apache.org/docs/0.9.1/scala-programming-guide.html), a high-level language, and also supports APIs for Python, SQL, Java and R.

Azure Databricks is the implementation of Apache Spark on Azure. With fully managed Spark clusters, it is used to process large workloads of data and also helps in data engineering, data exploring and also visualizing data using Machine learning.

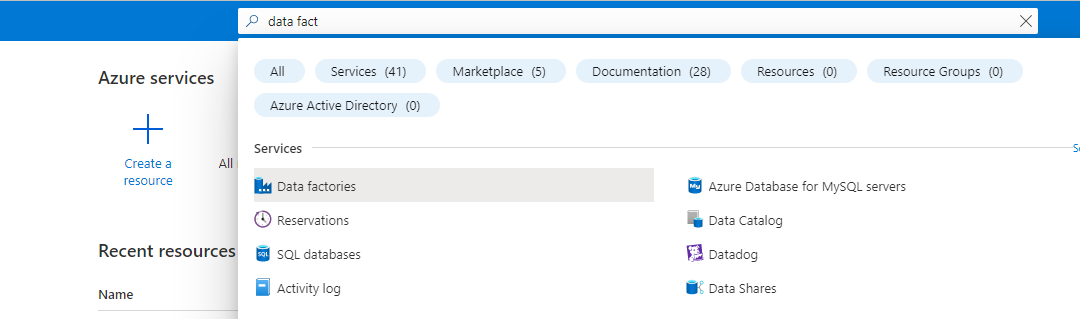
In this notebook, we will explore combining streaming and batch processing with a single pipeline. We will begin by defining the following logic:

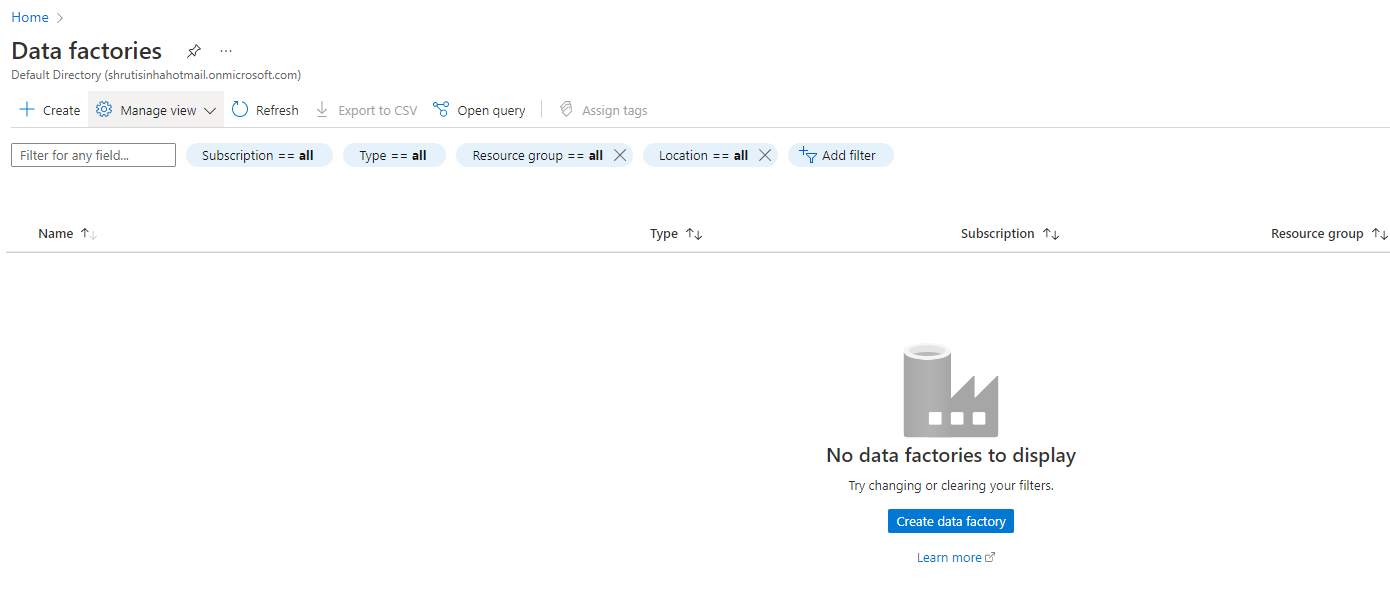
* Ingest streaming JSON data from disk and write it to a Delta Lake Table **/activity/Bronze**
* perform a Stream-Static Join on the streamed data to add additional geographic data
* transform and load the data, saving it out to our Delta Lake Table **/activity/Silver**
* summarize the data through aggregation into the Delta Lake Table /activity/Gold/groupedCounts
* materialize views of our gold table through streaming plots and static queries

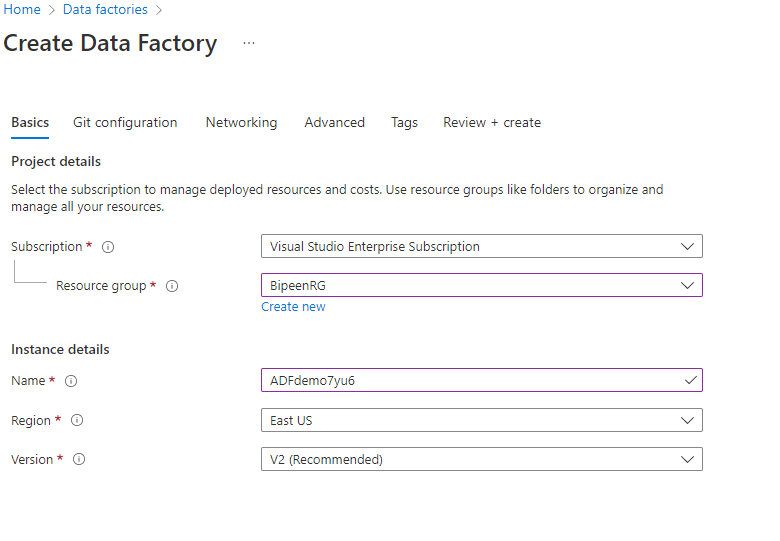
We will then demonstrate that by writing batches of data back to our bronze table, we can trigger the same logic on newly loaded data and propagate our changes automatically.

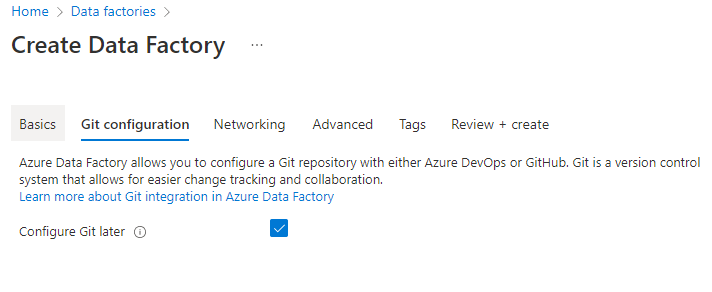
# Exercise 1 – Ingest Data from GitHub

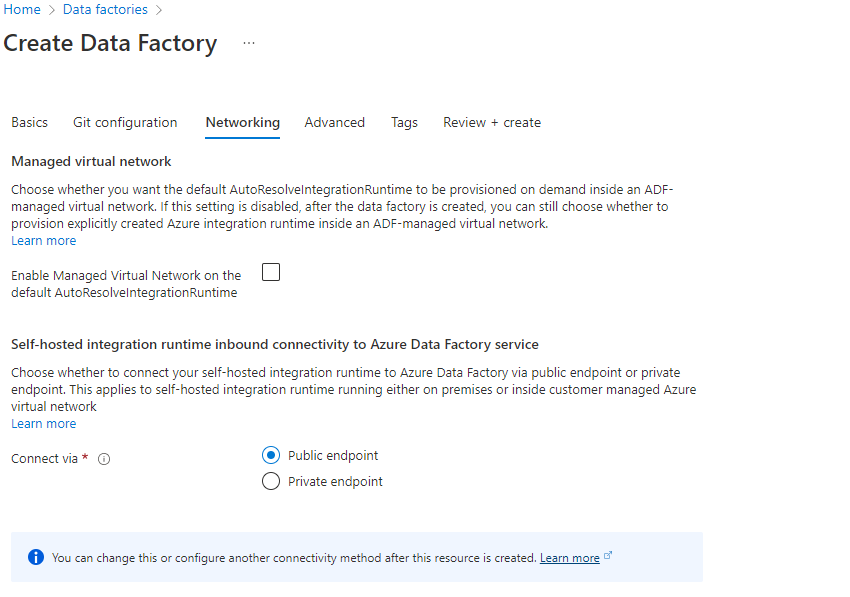
* In the Azure Databricks Workspace, in the left pane, select **Workspace >** Users, and select your username (the entry with the house icon).

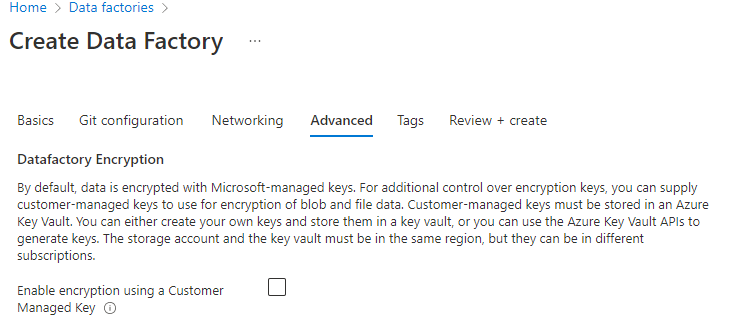


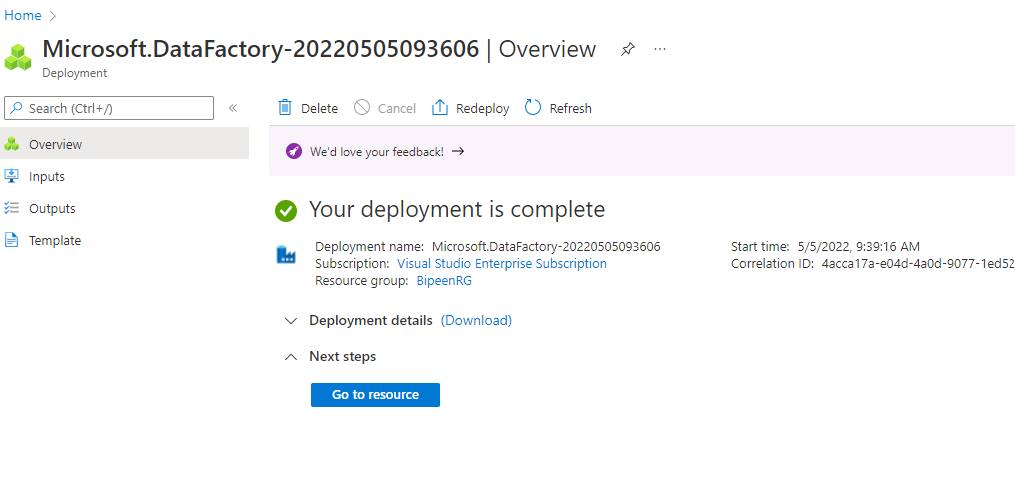






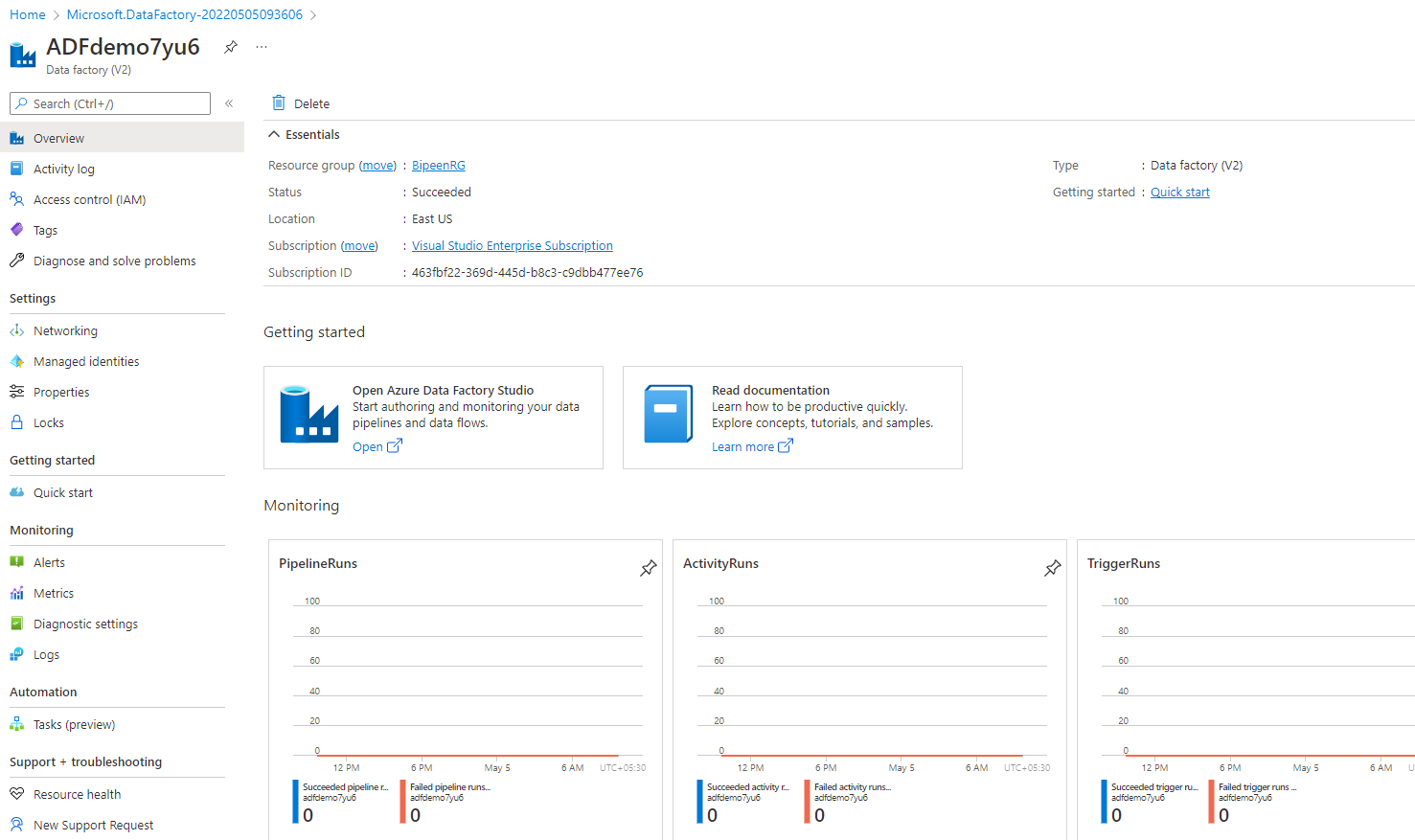


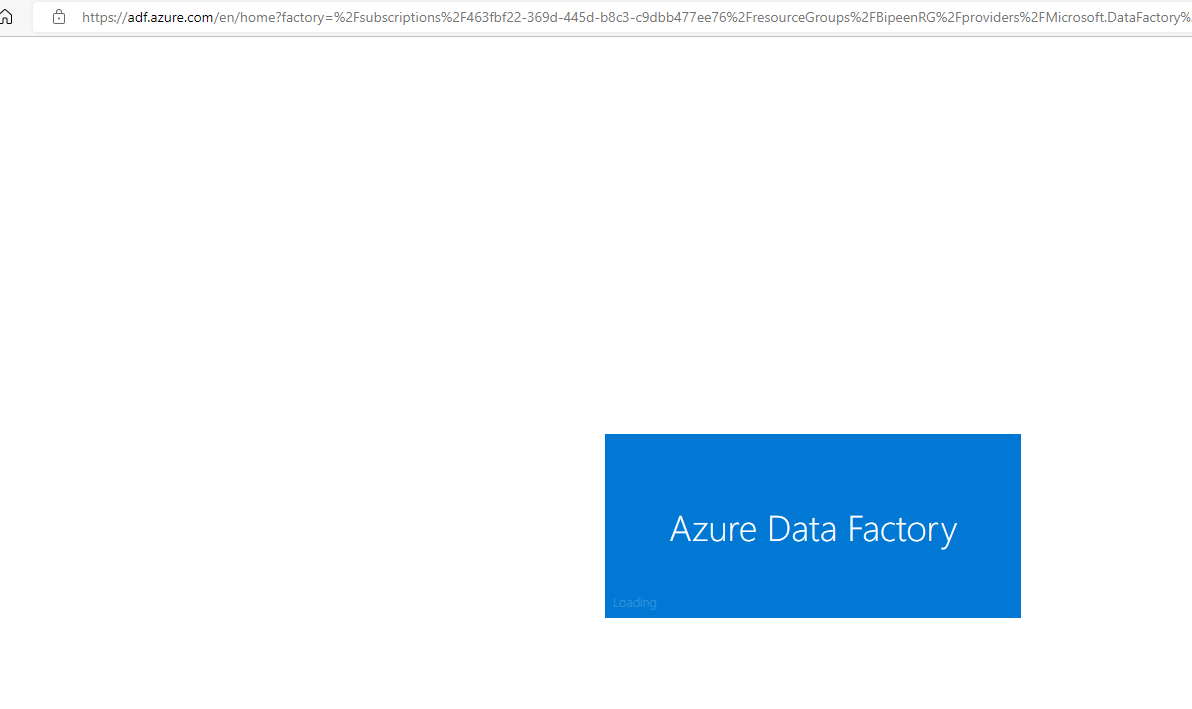


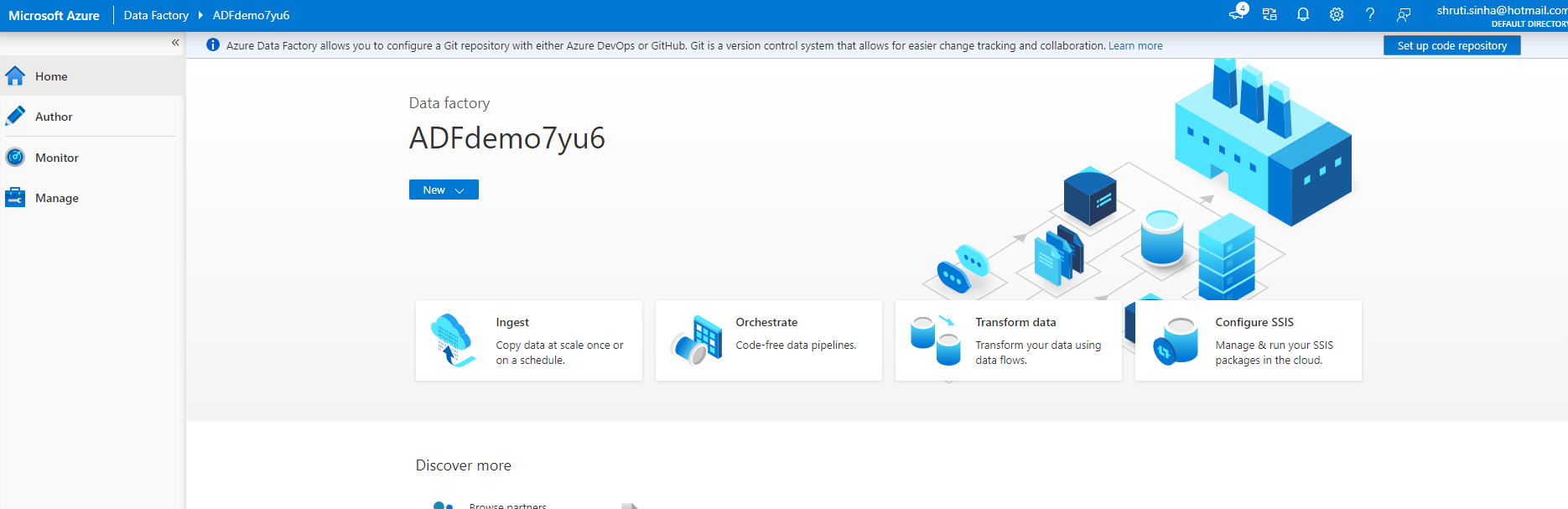


# Exercise 1 – Ingest Data from GitHub

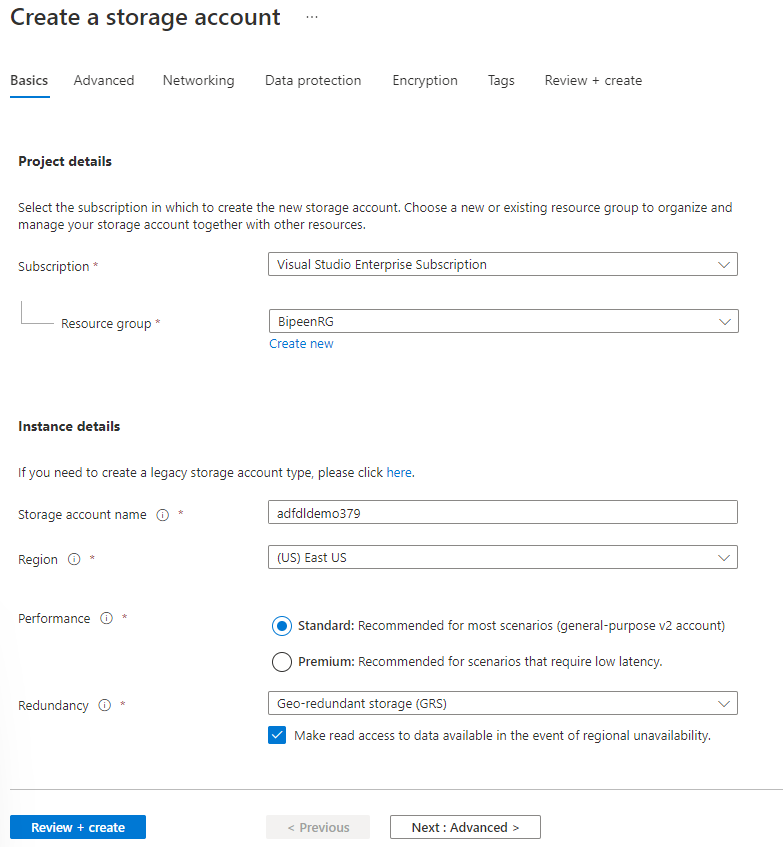
\

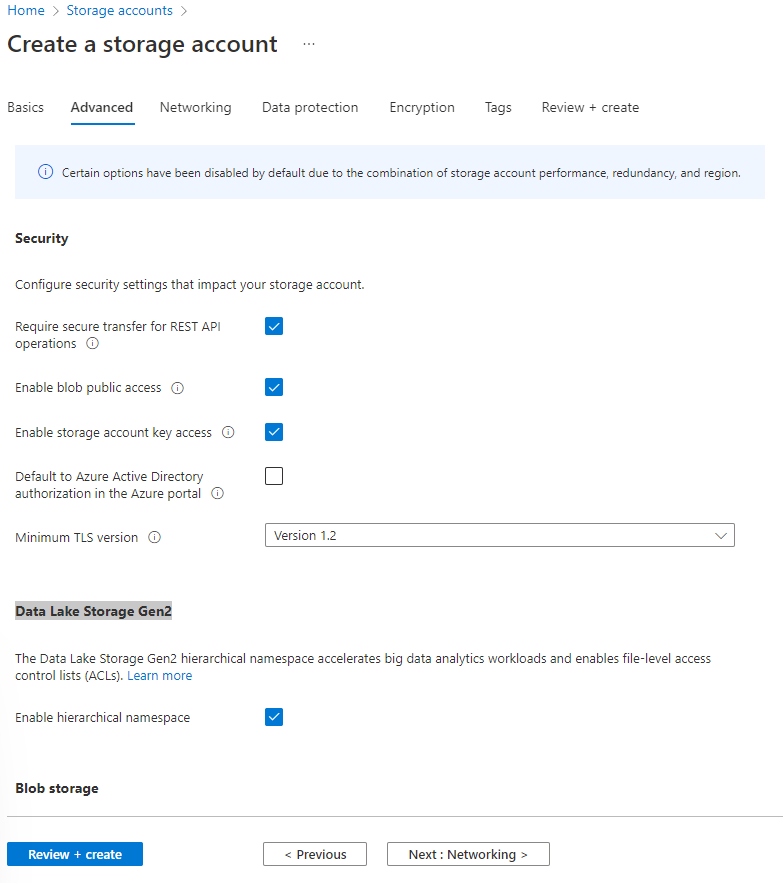


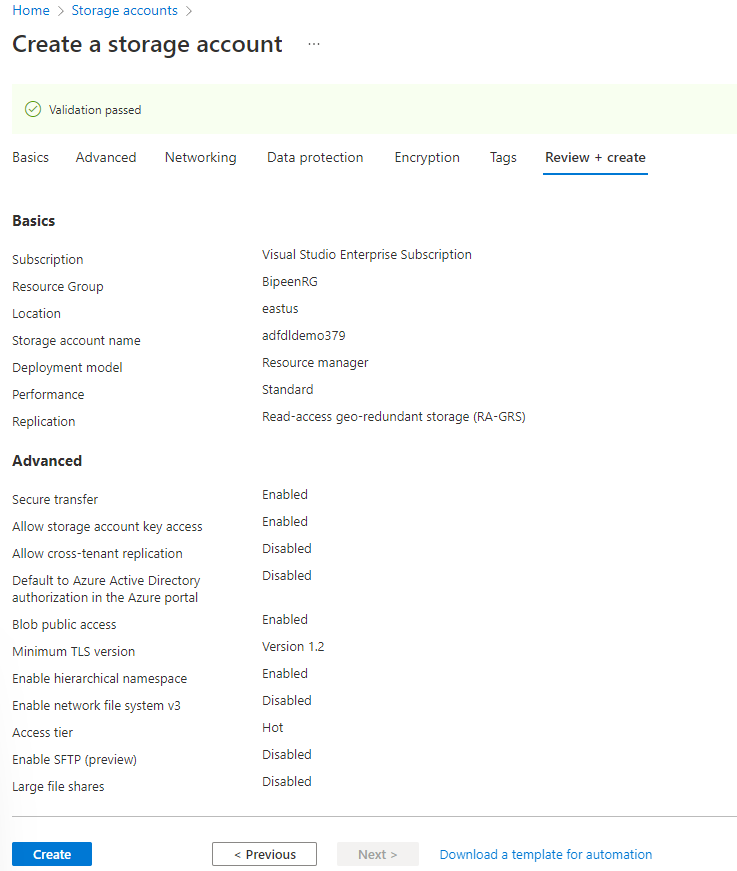


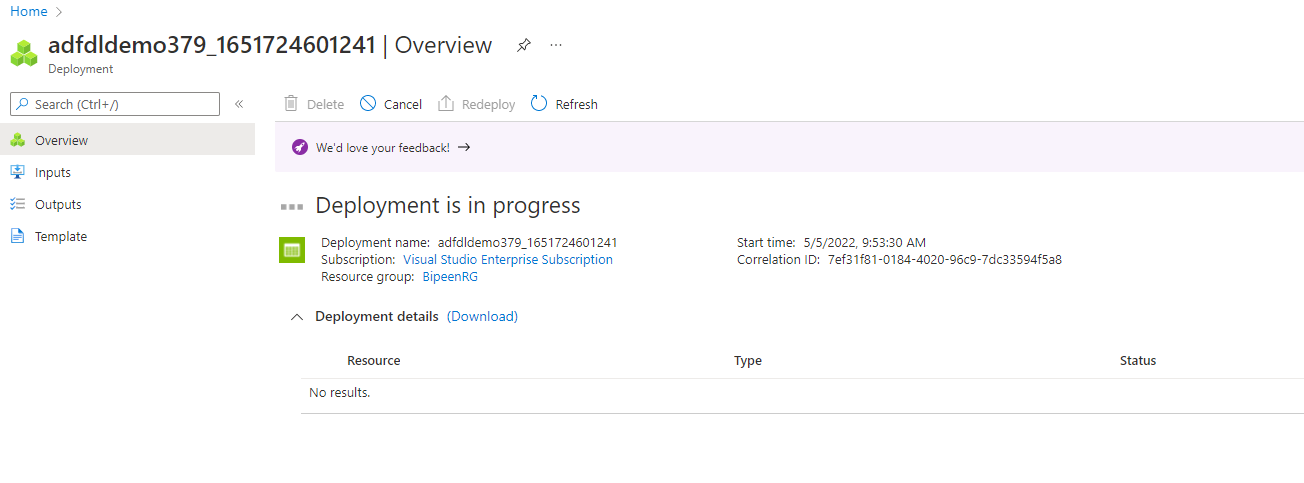


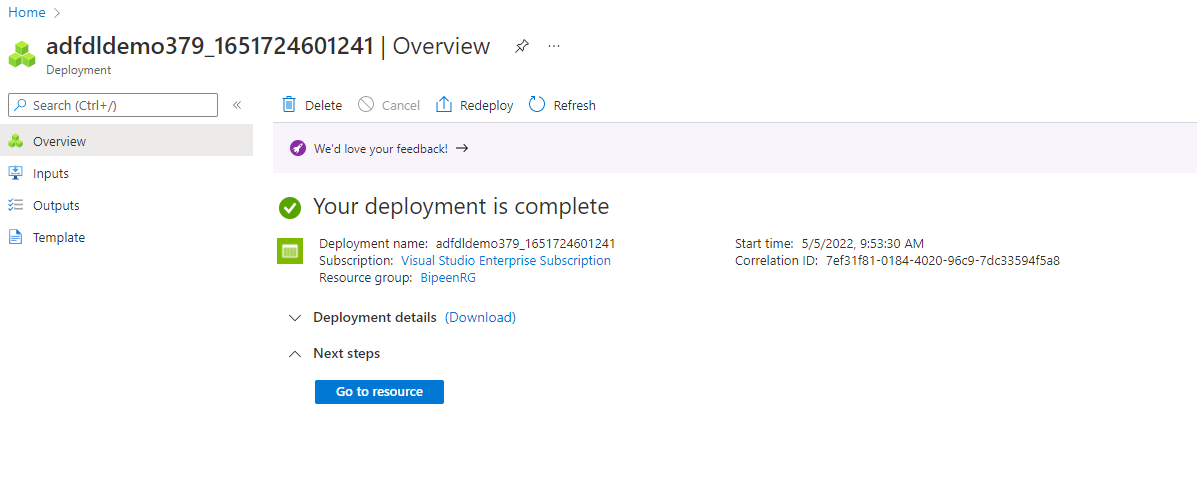
# Exercise 1 – Ingest Data from GitHub



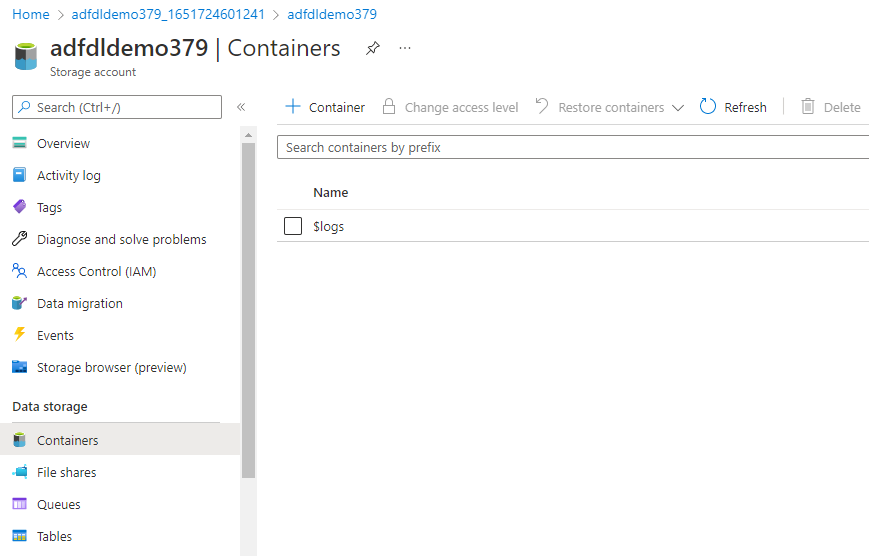


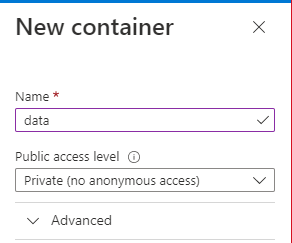


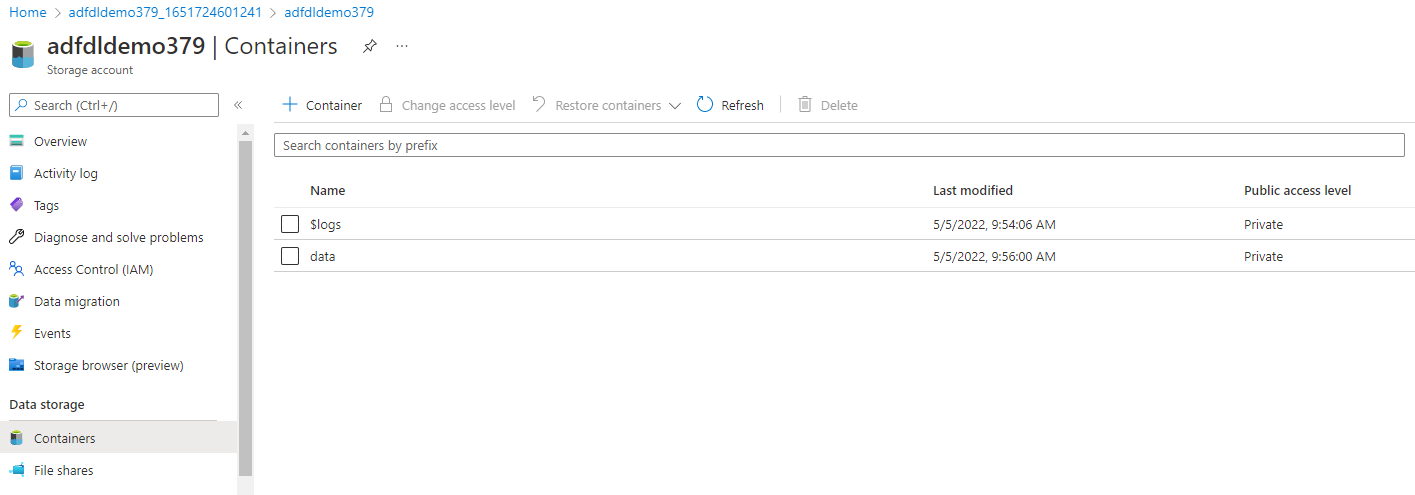




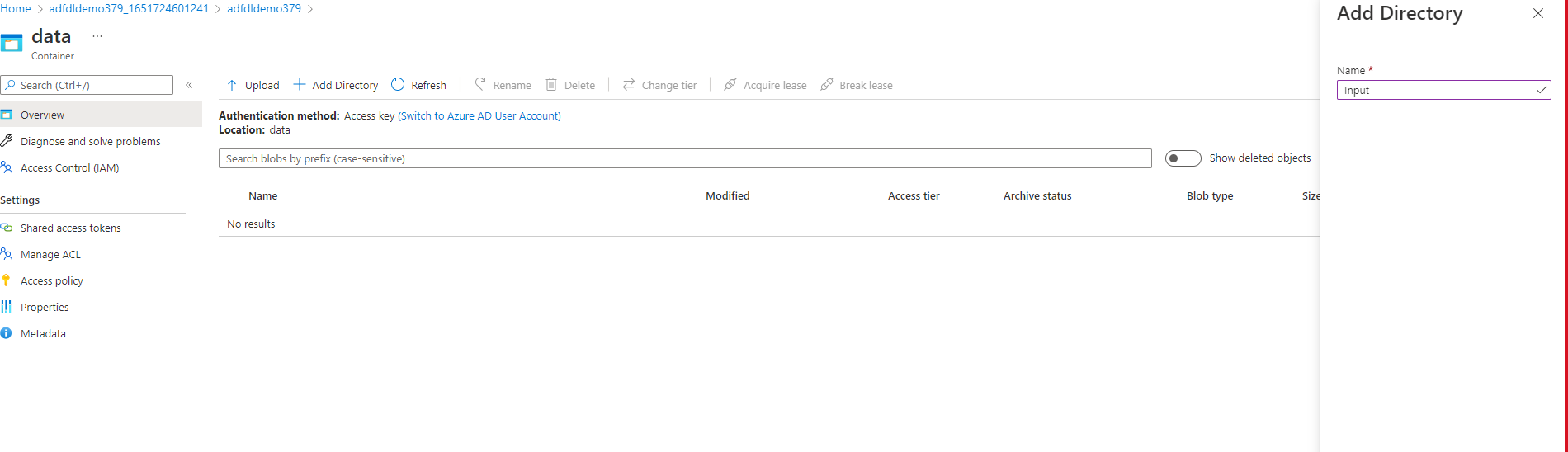
# Exercise 1 – Ingest Data from GitHub

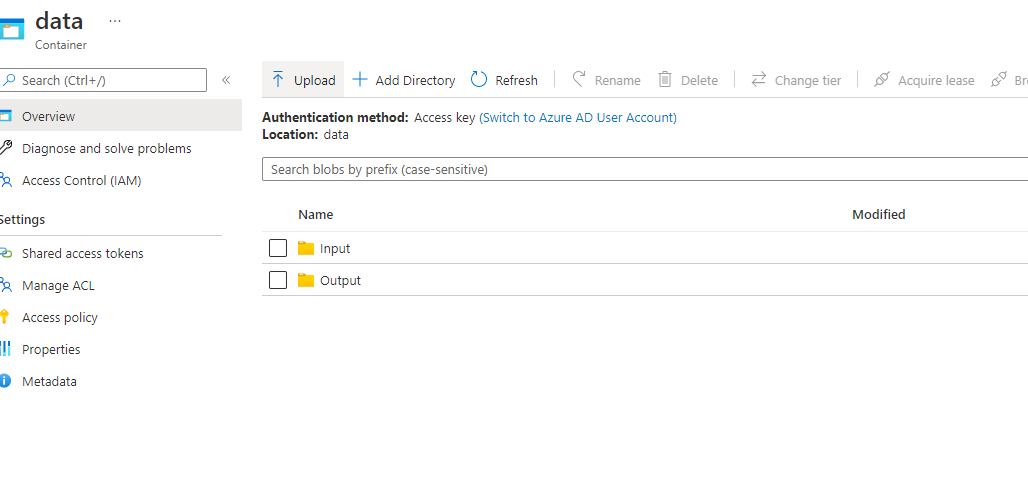


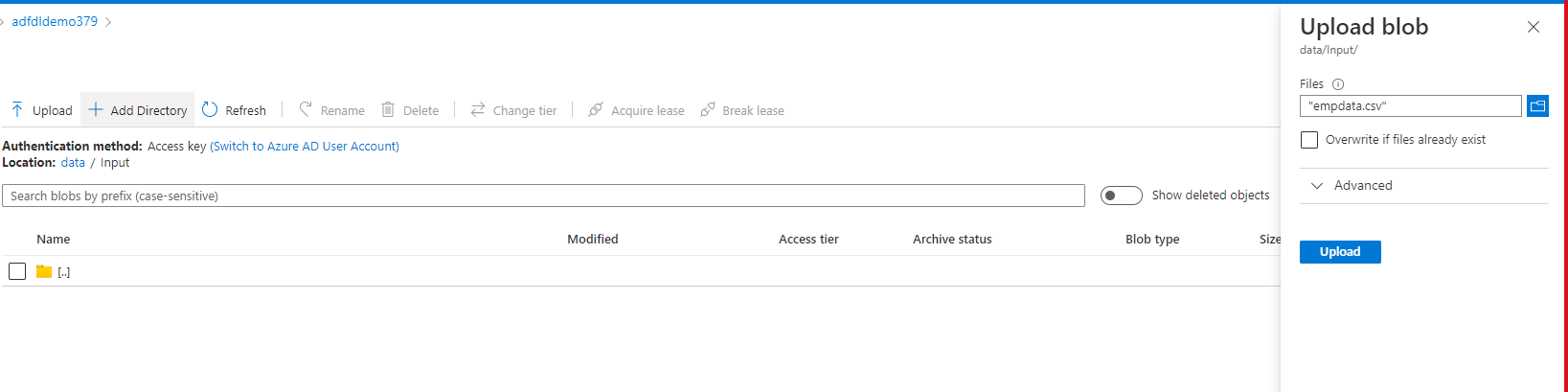


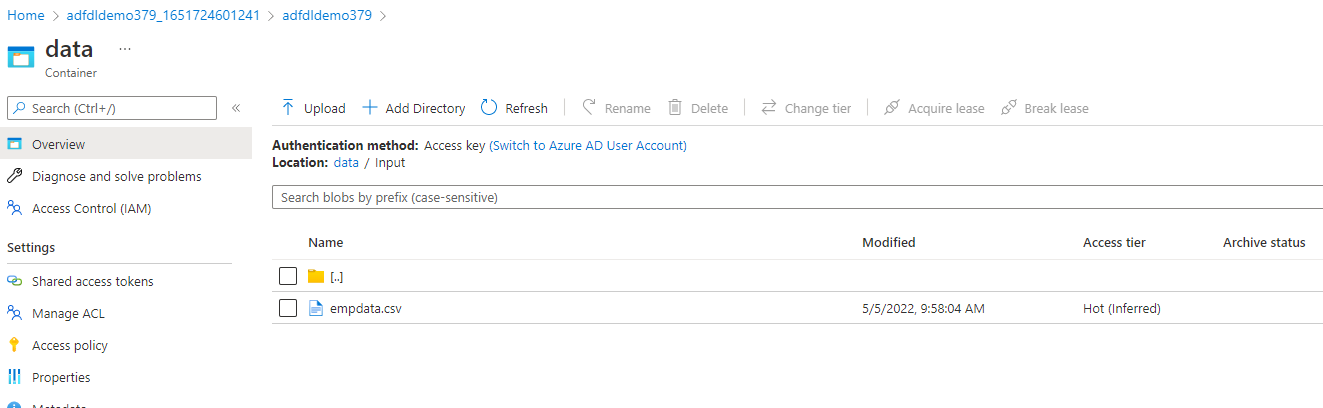


**Click Save**

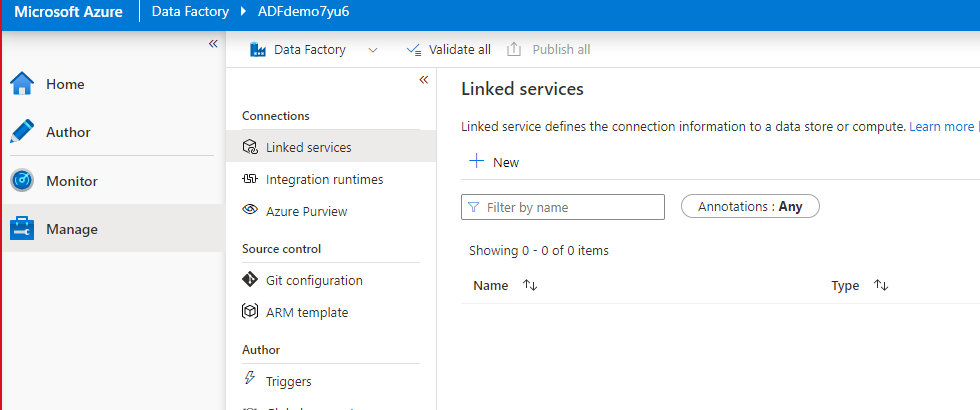
****

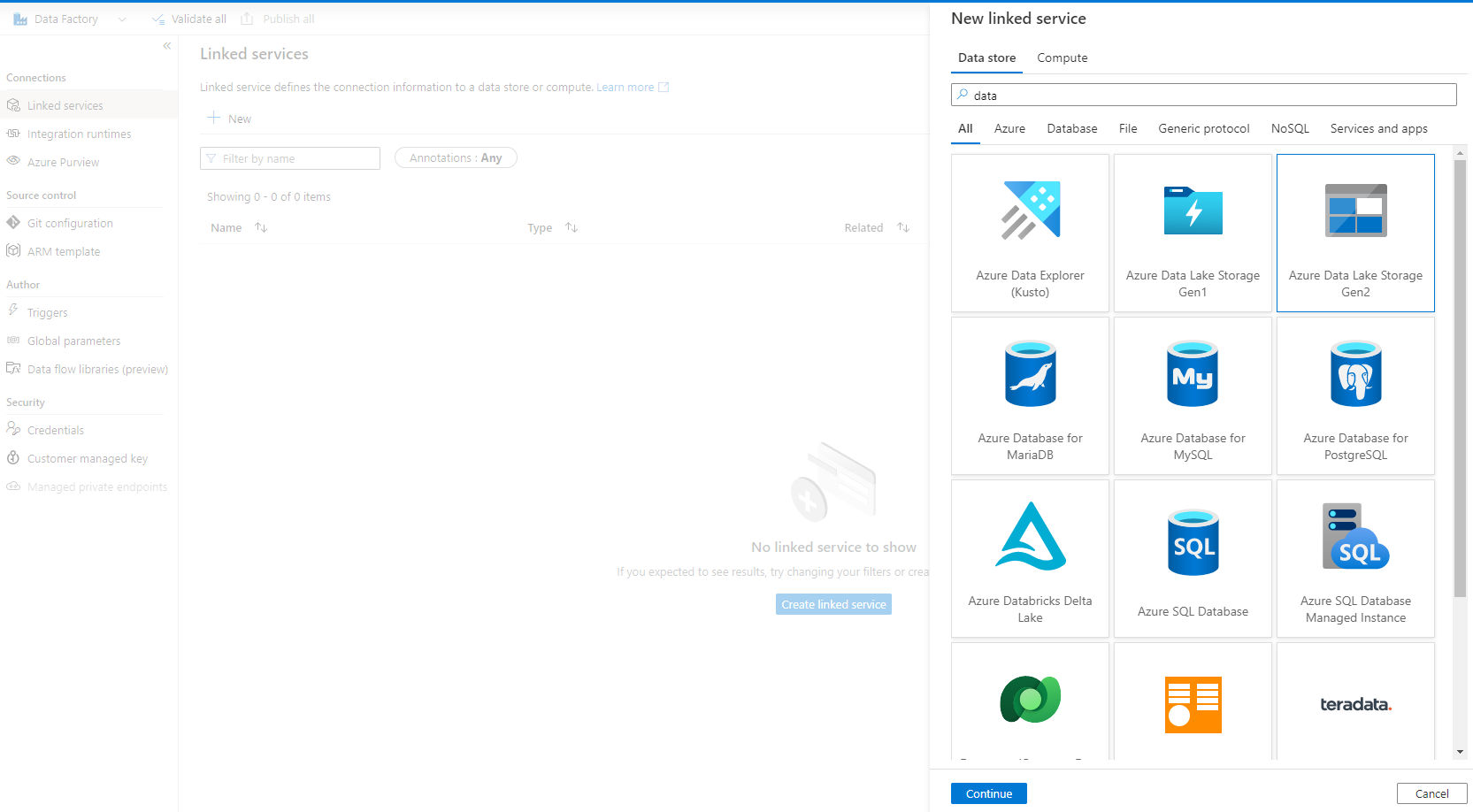
****

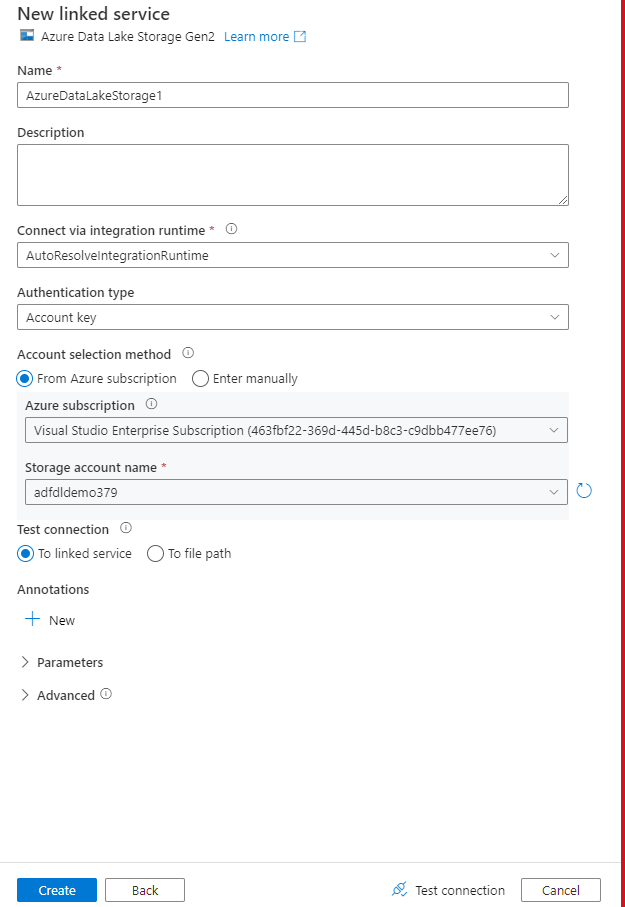
****

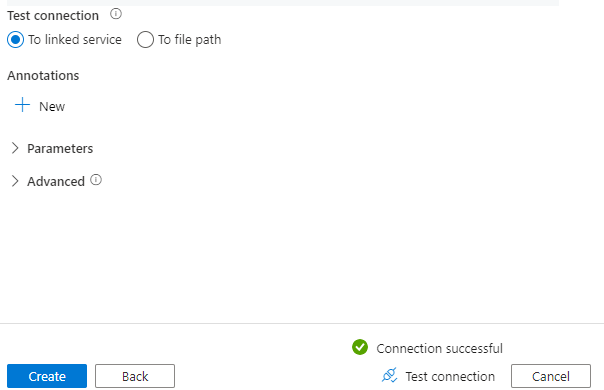
****

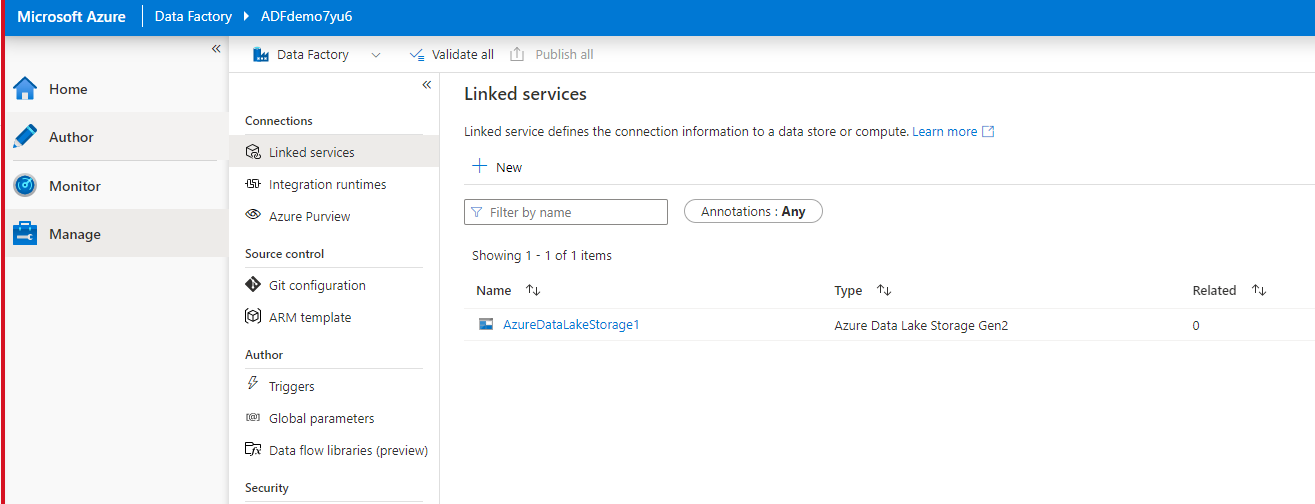
# Exercise 1 – Ingest Data from GitHub

****

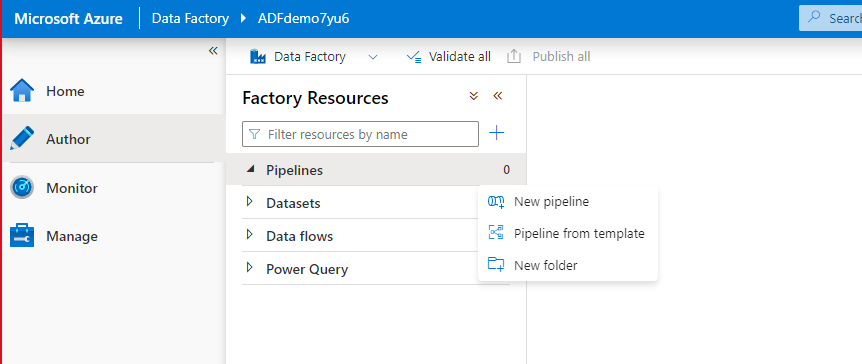
****

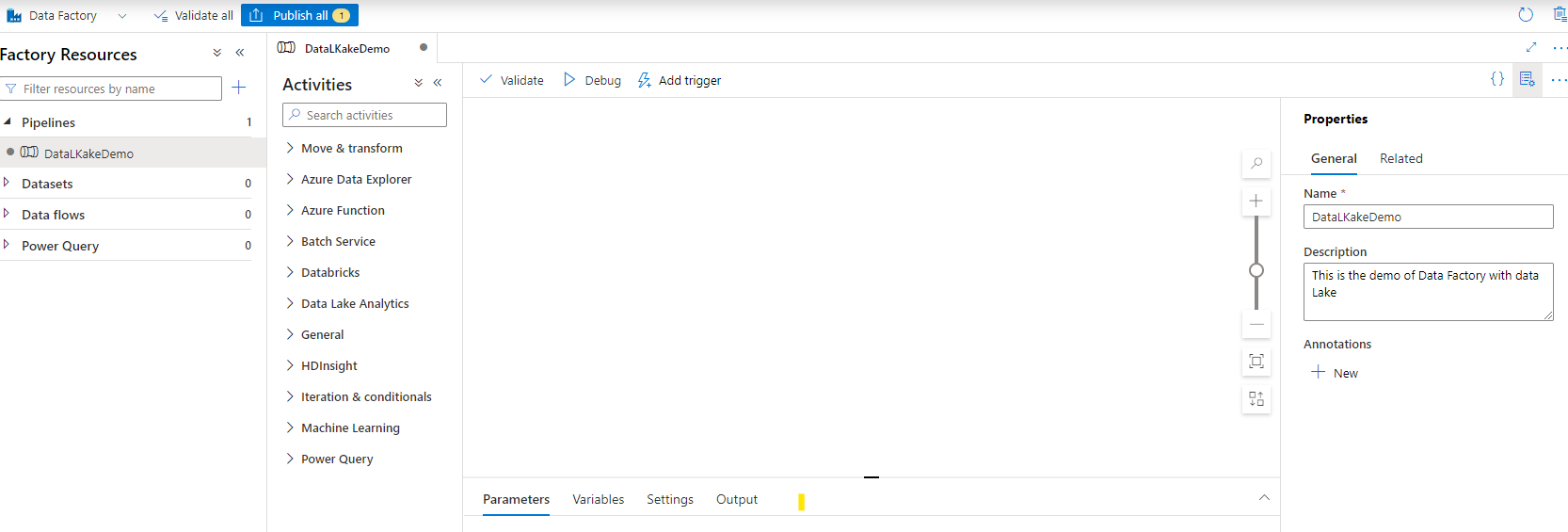
****

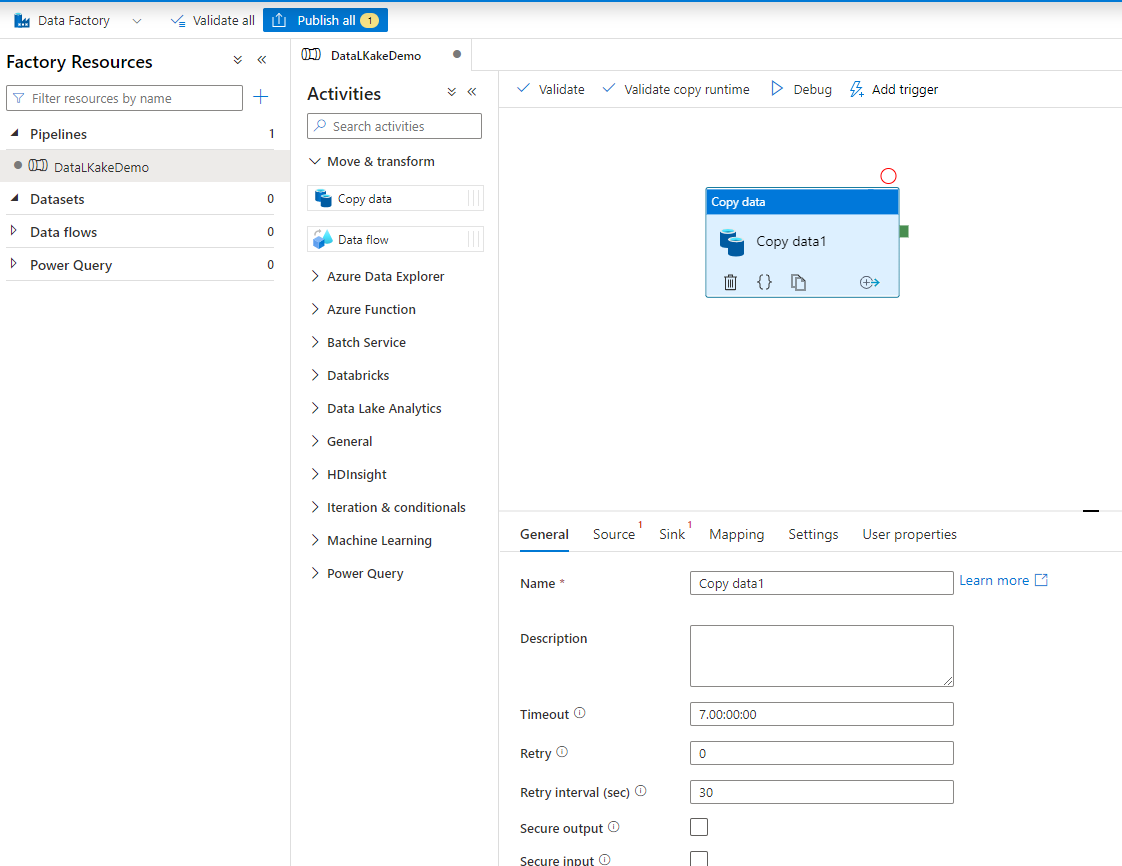
****

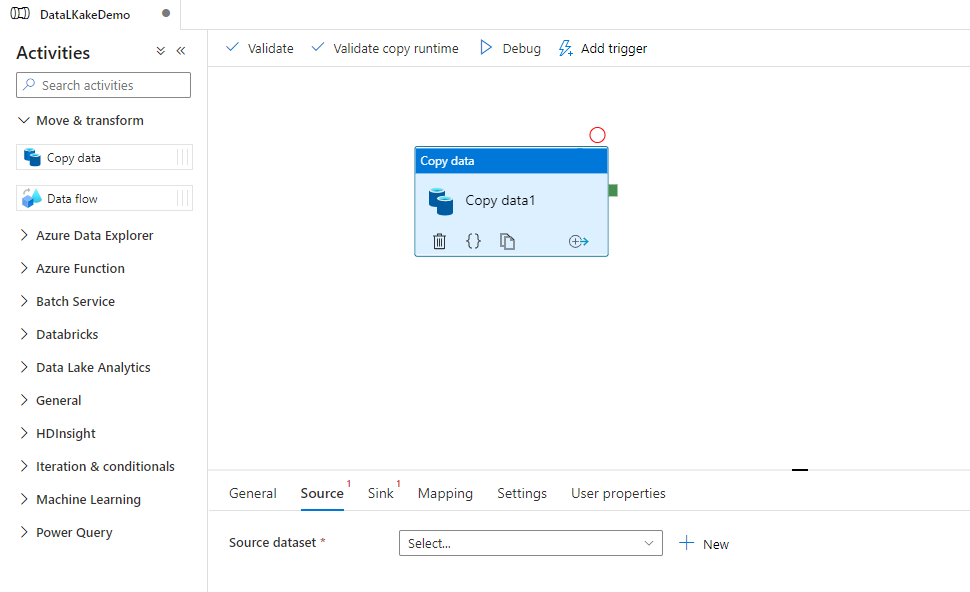
****

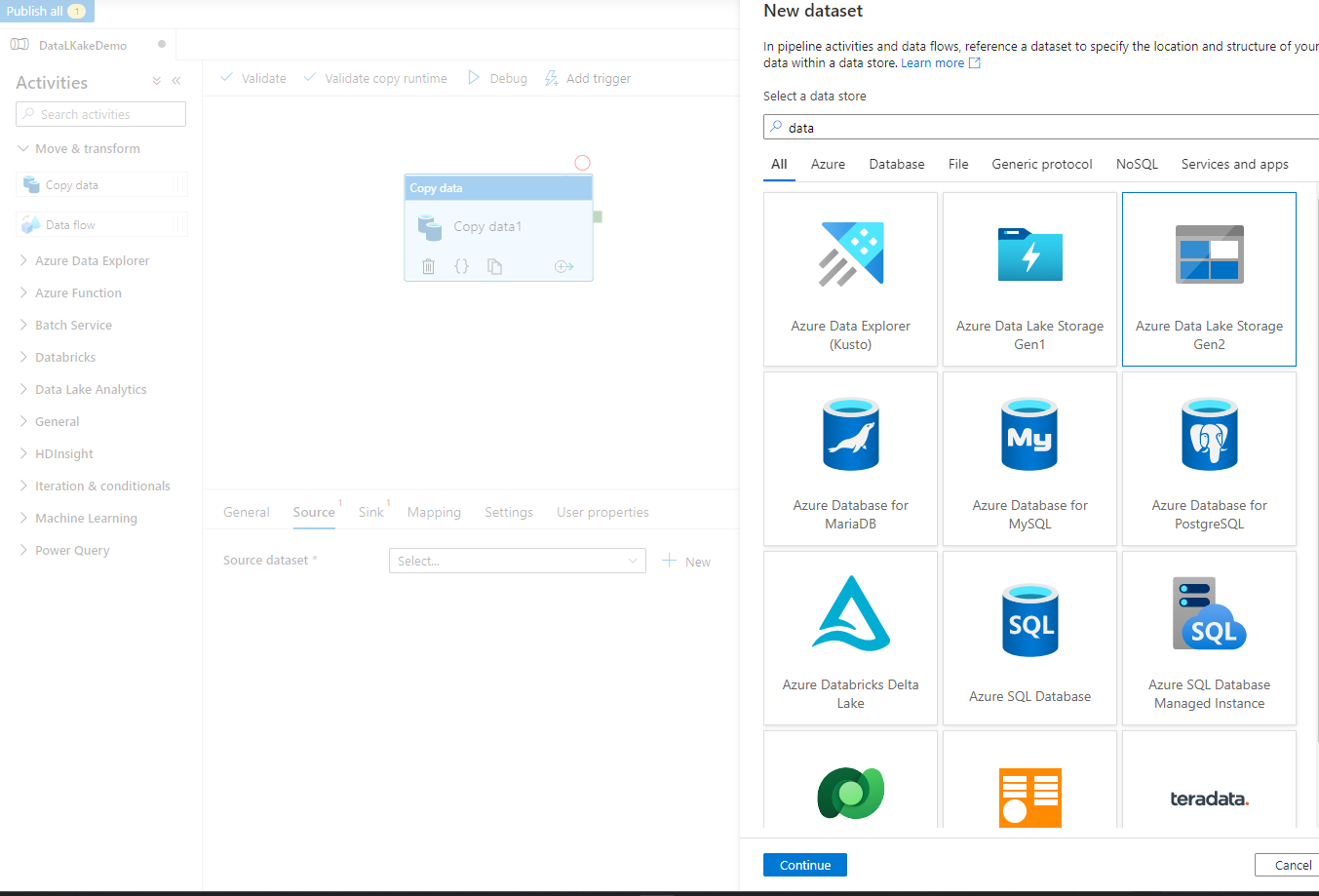
# Exercise 1 – Ingest Data from GitHub

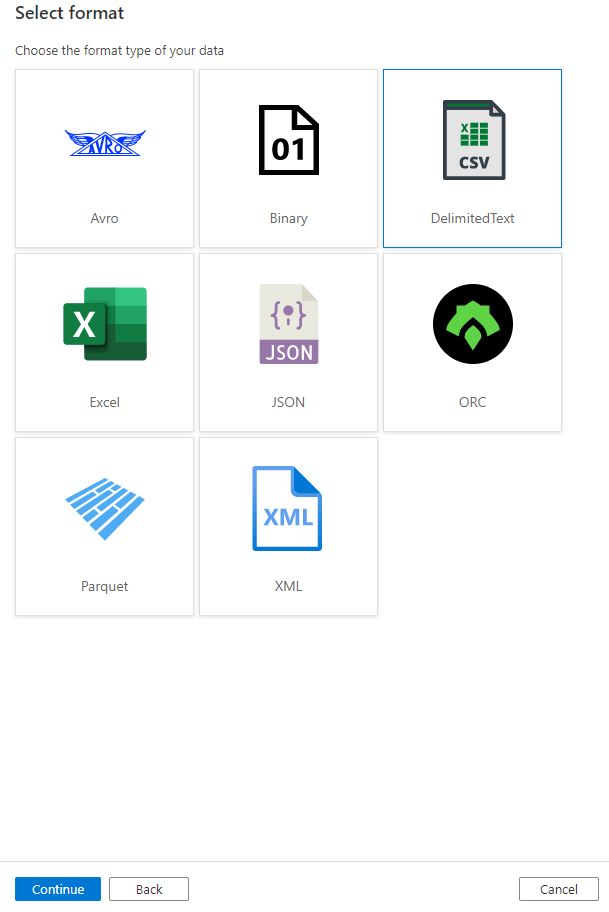
****

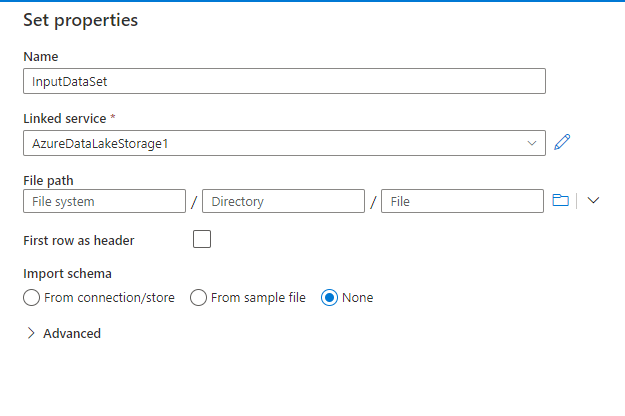
****

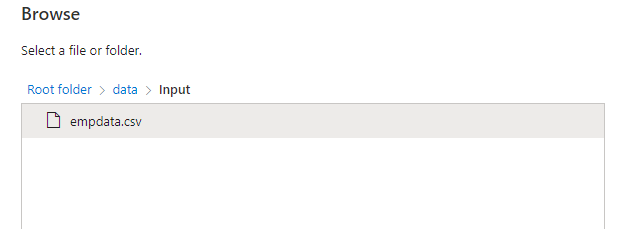
****

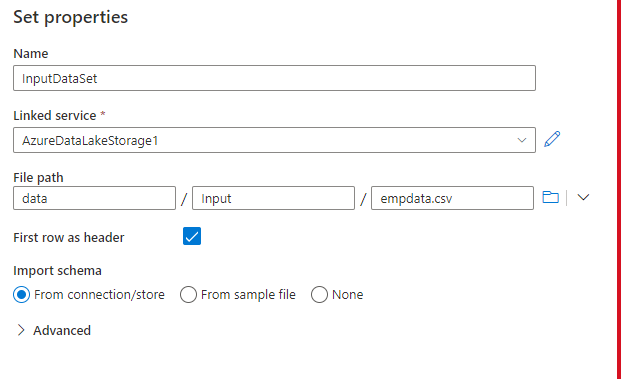
****

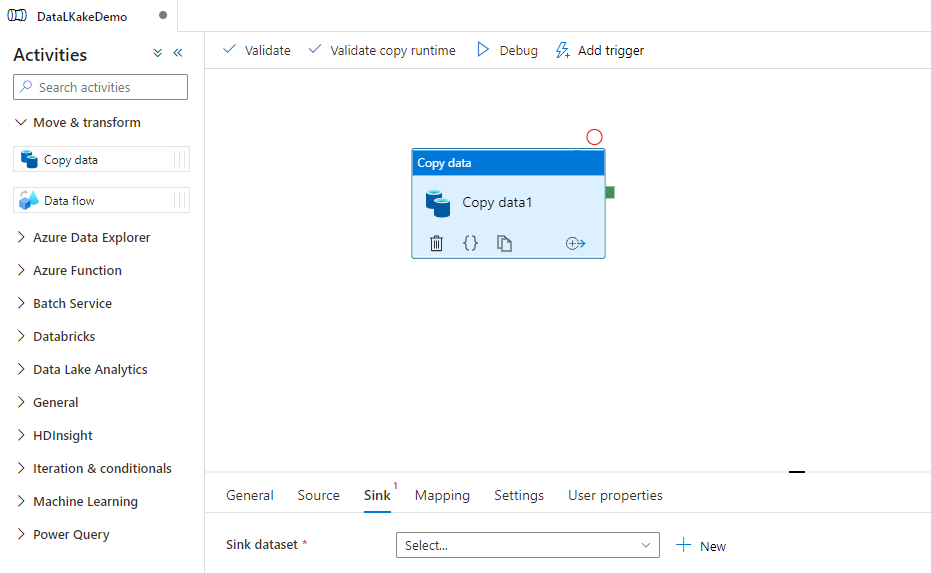
****

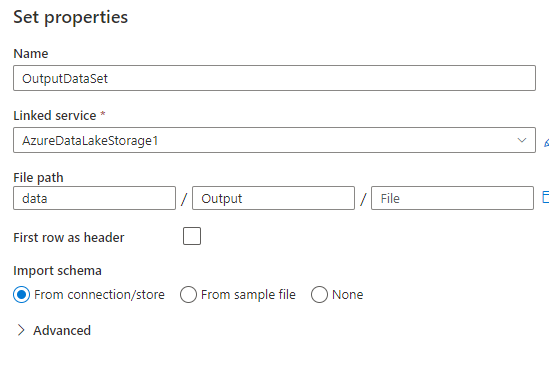
****

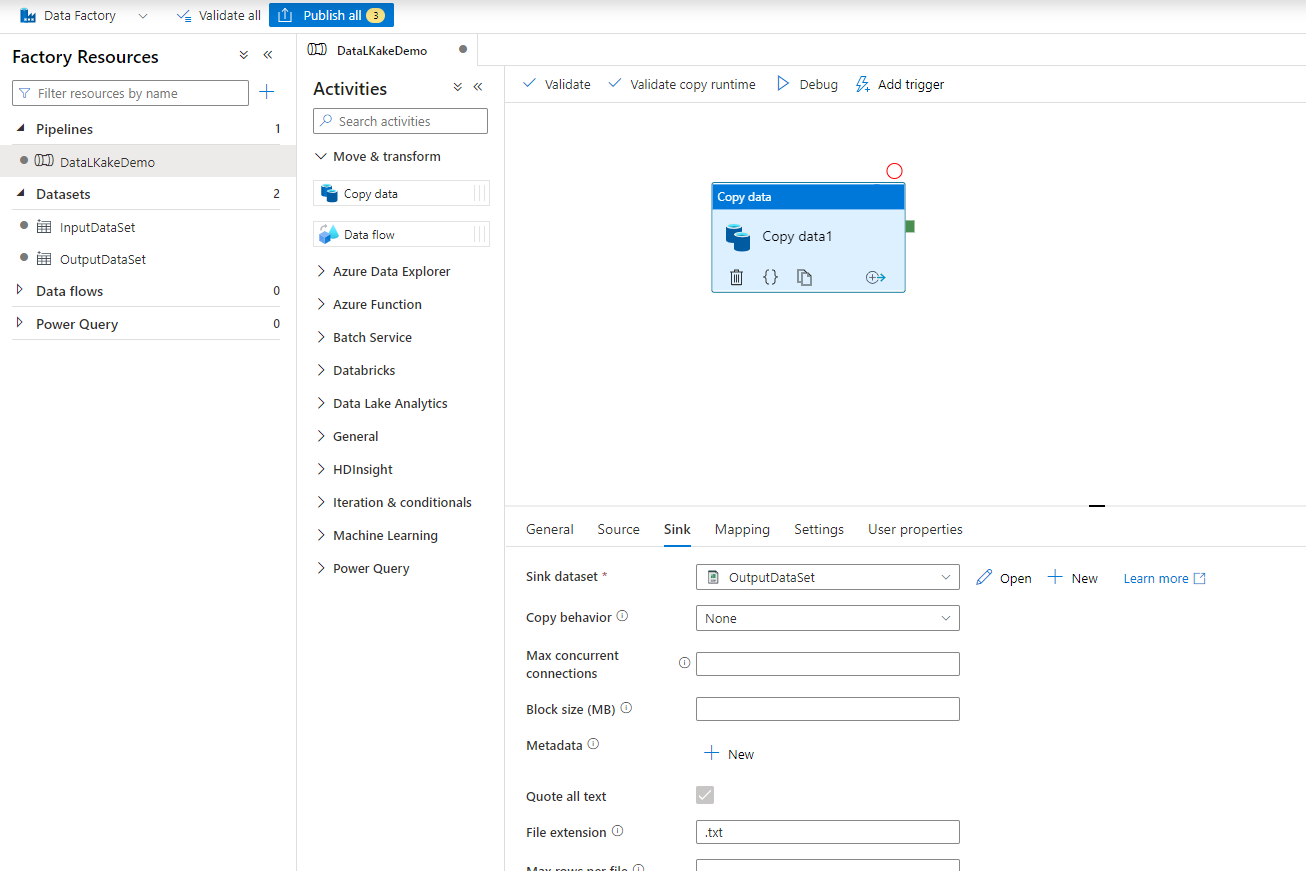
****

****

****

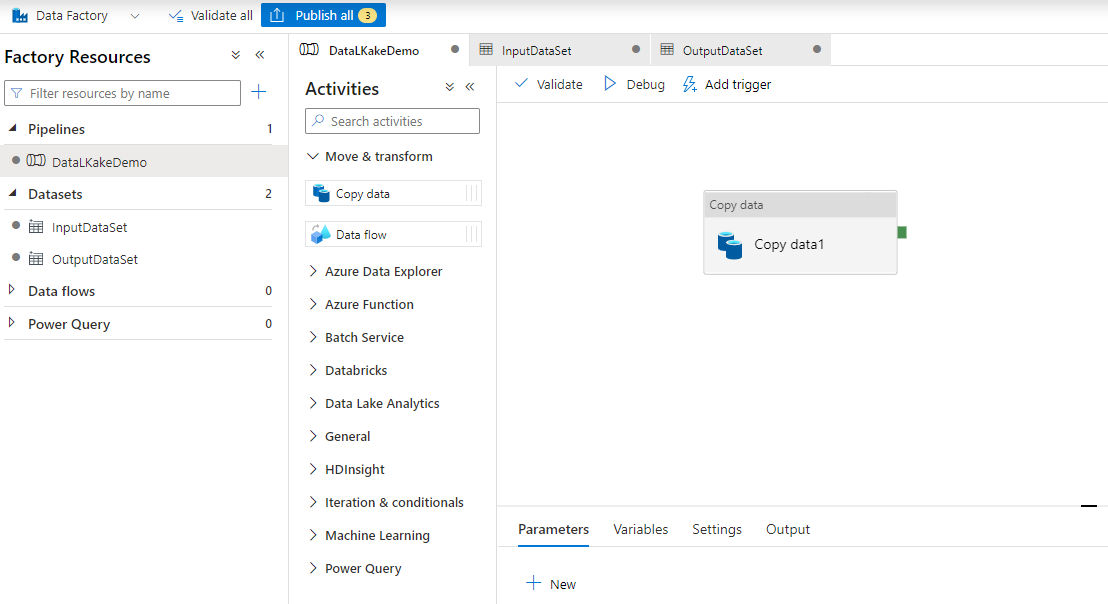
****

****

****

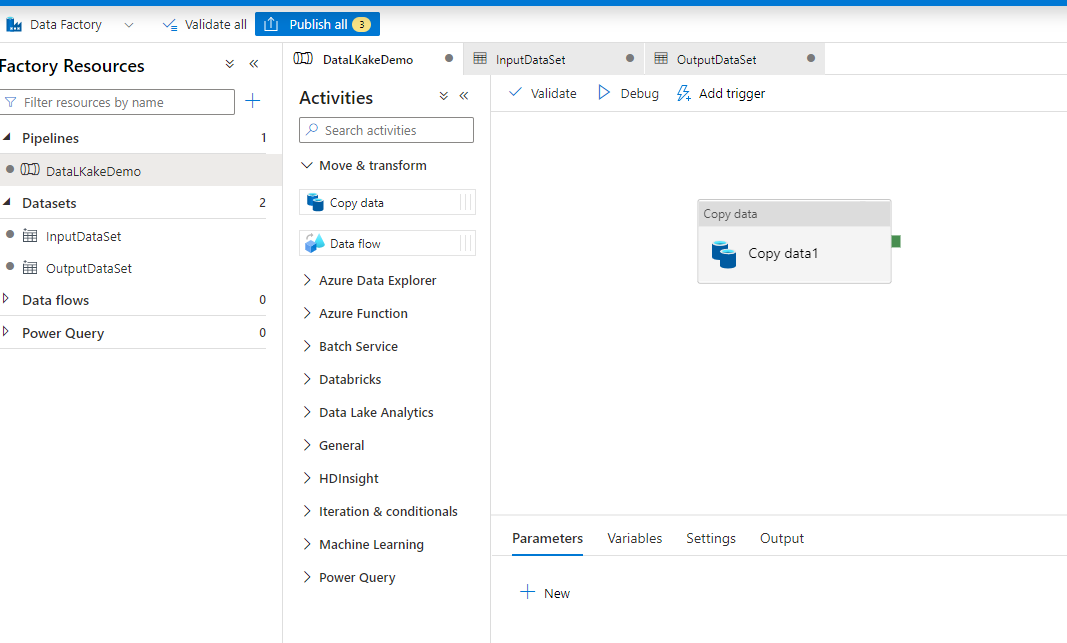
# Exercise 1 – Ingest Data from GitHub

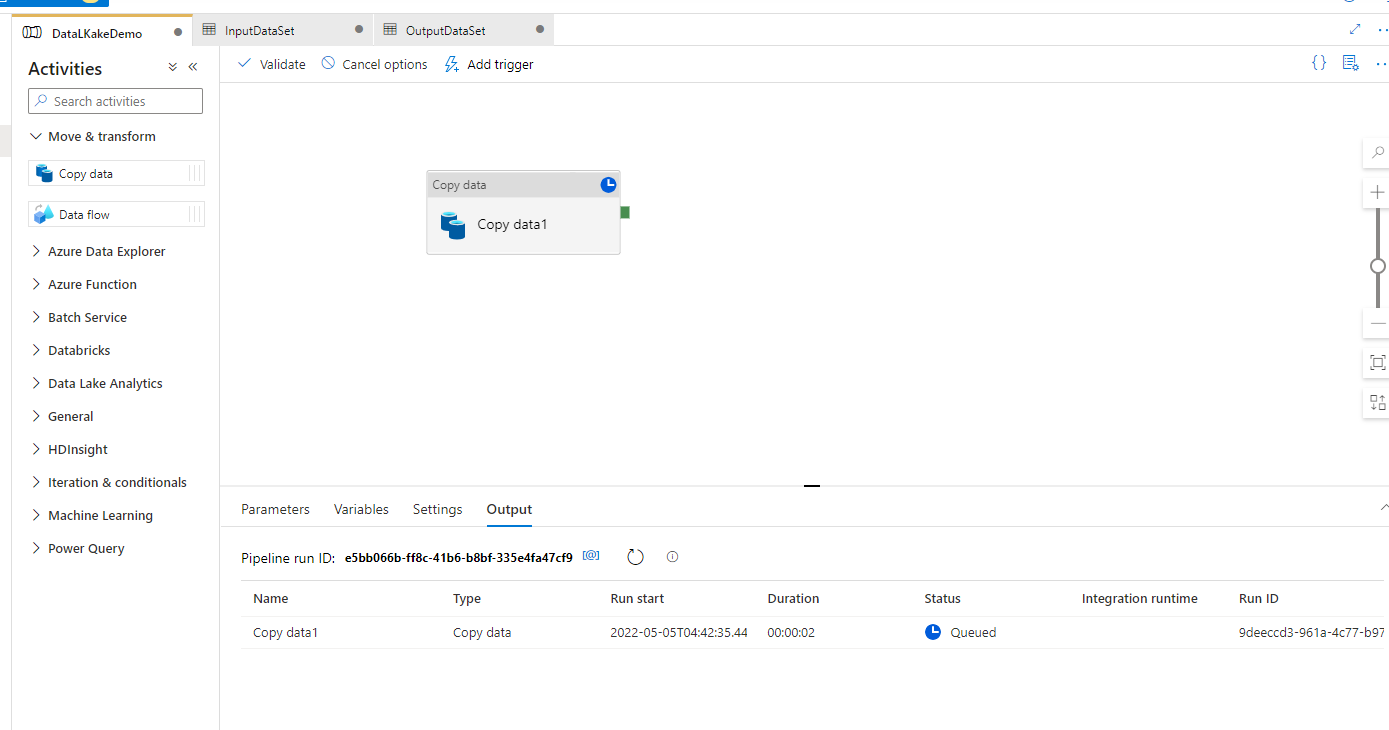
**Run Validate**

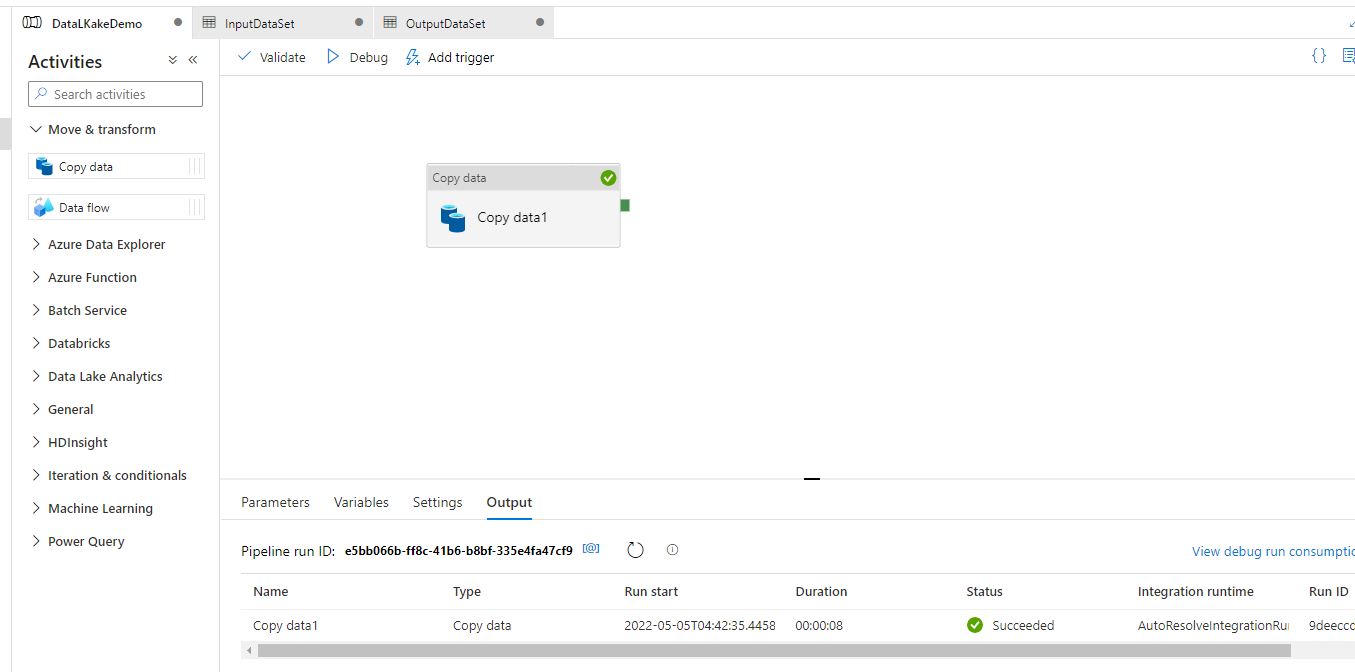
****

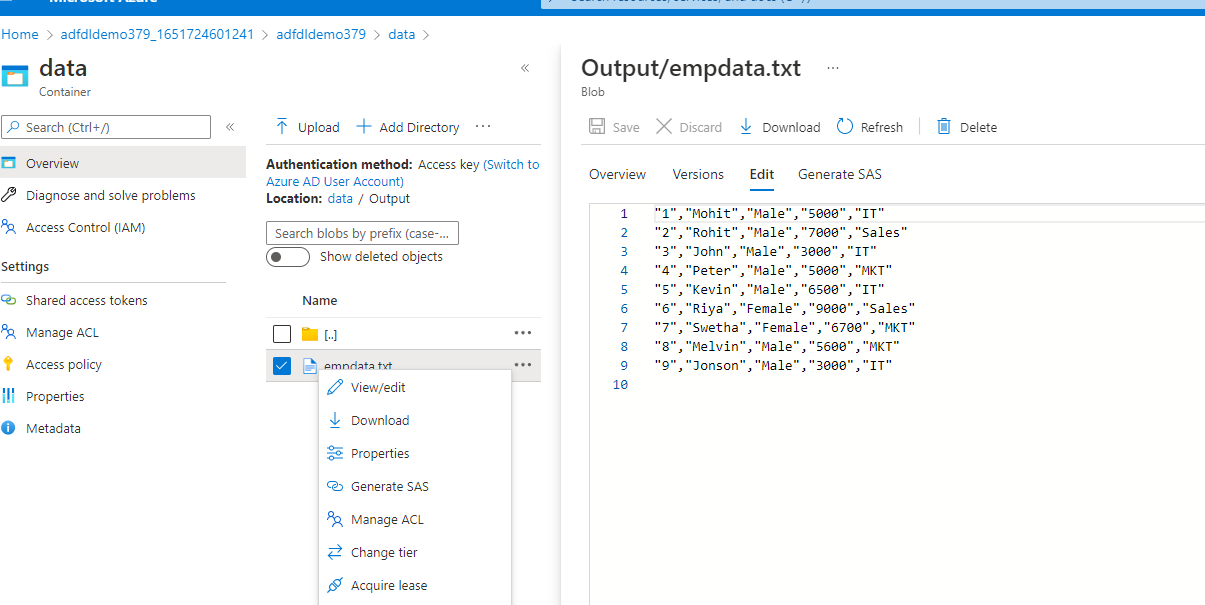
****

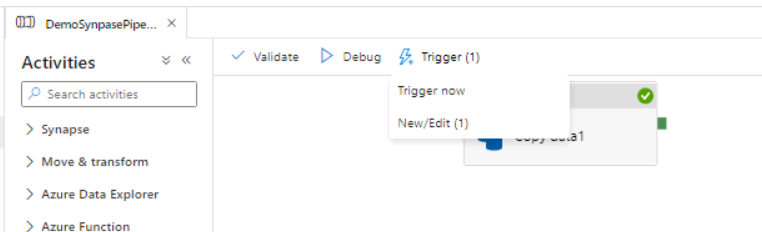
**Click debug**

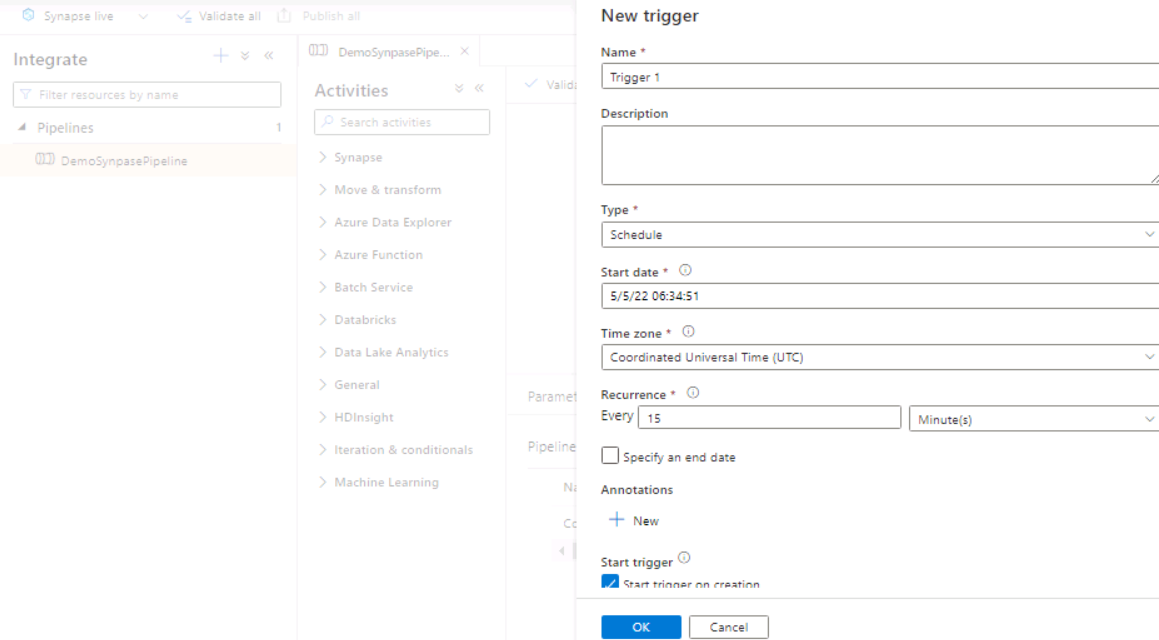
****

****

****

****

****

**c**