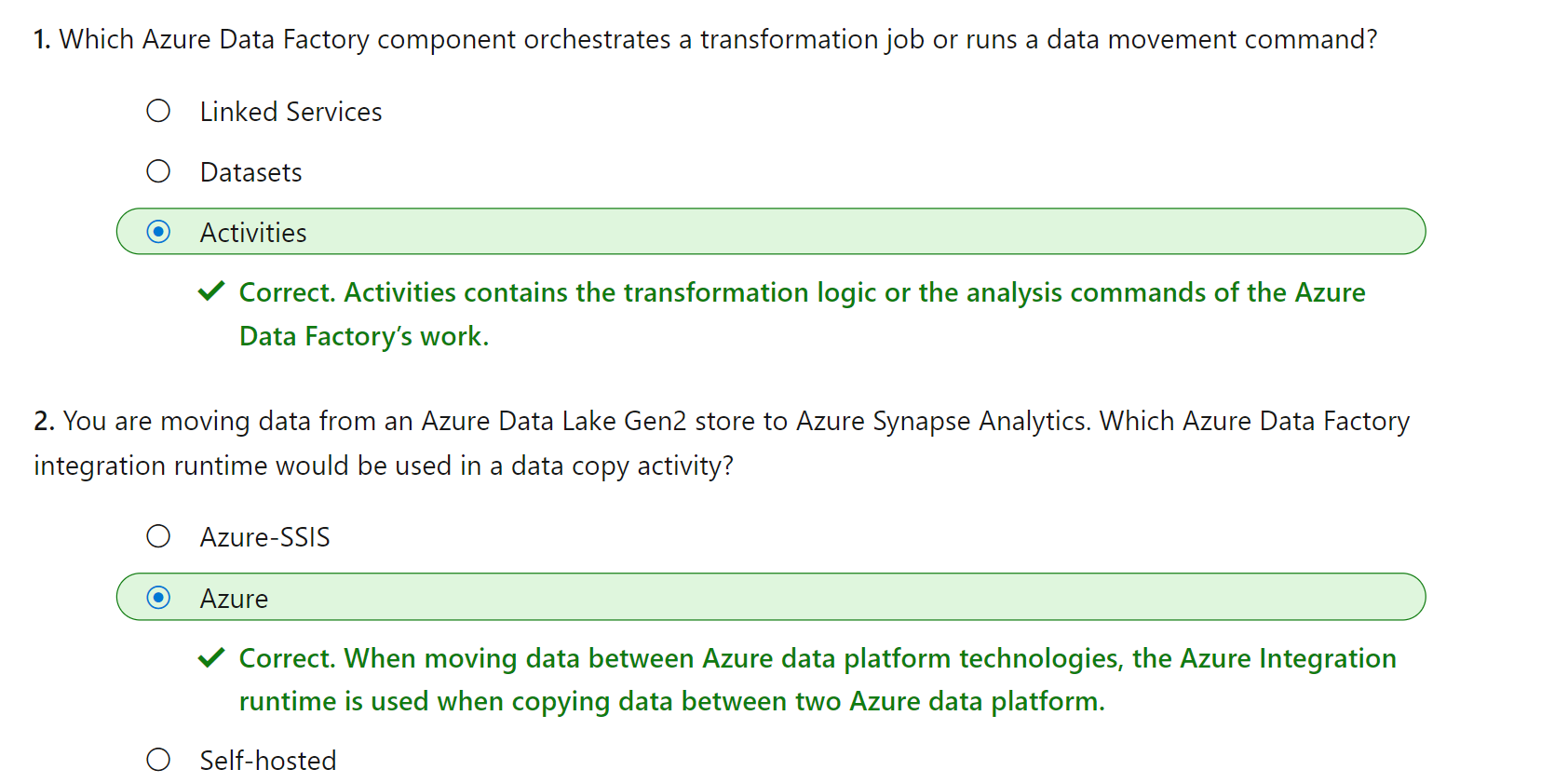
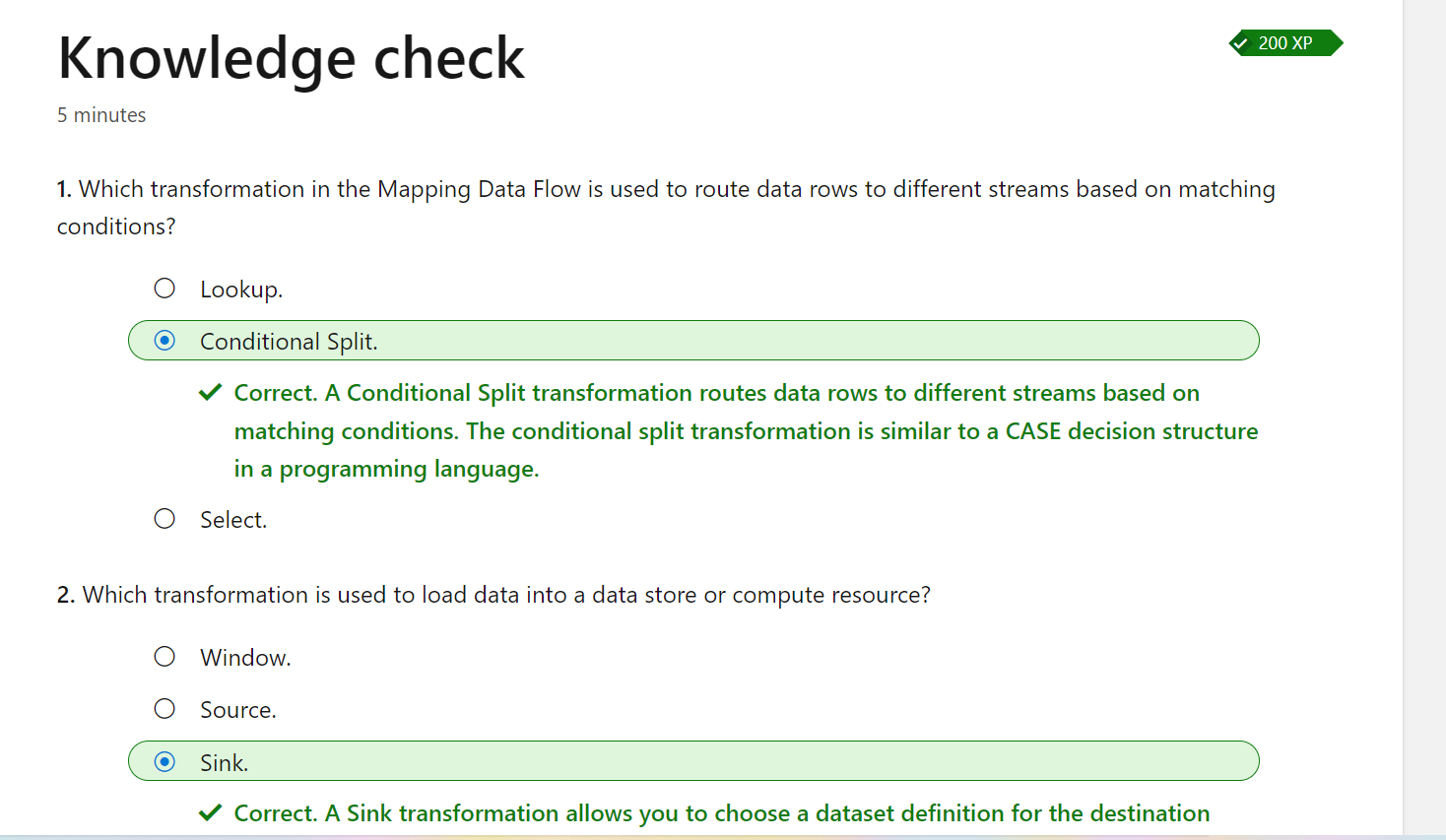
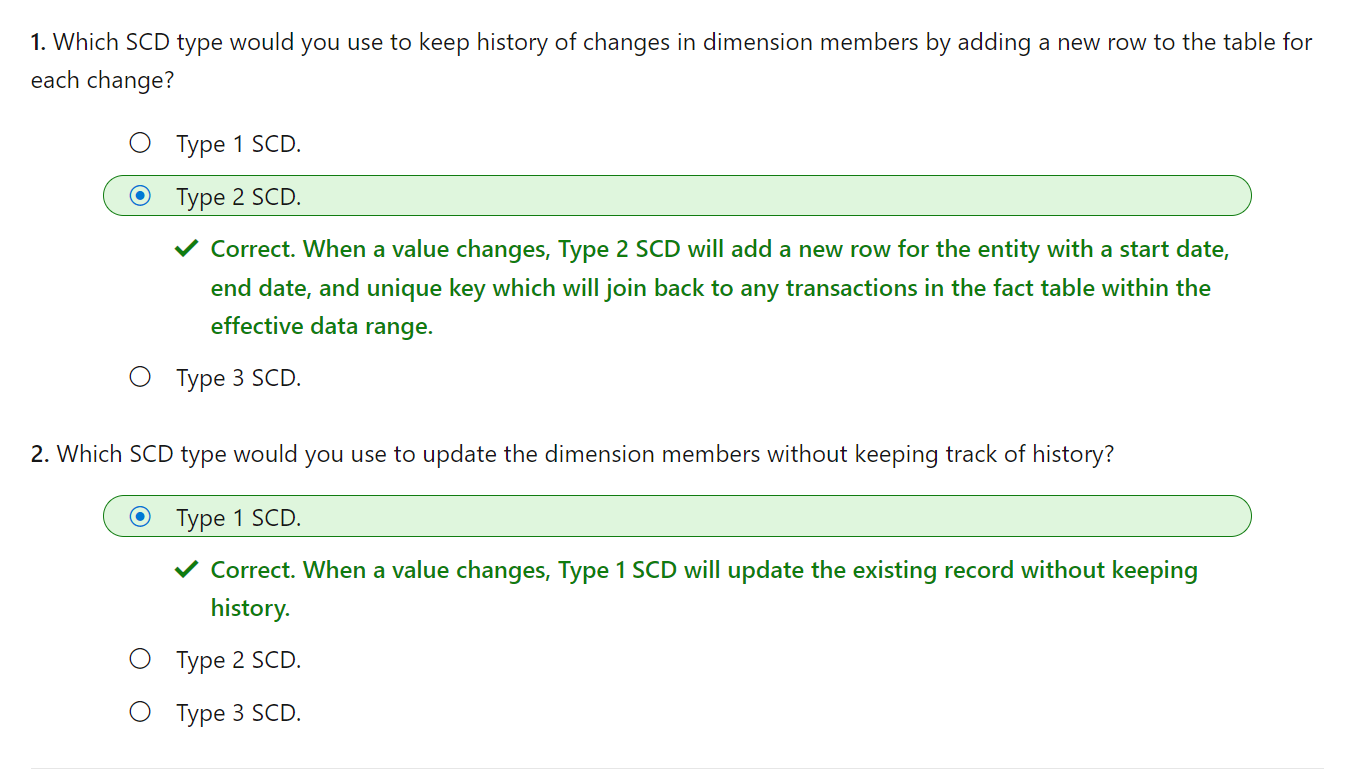
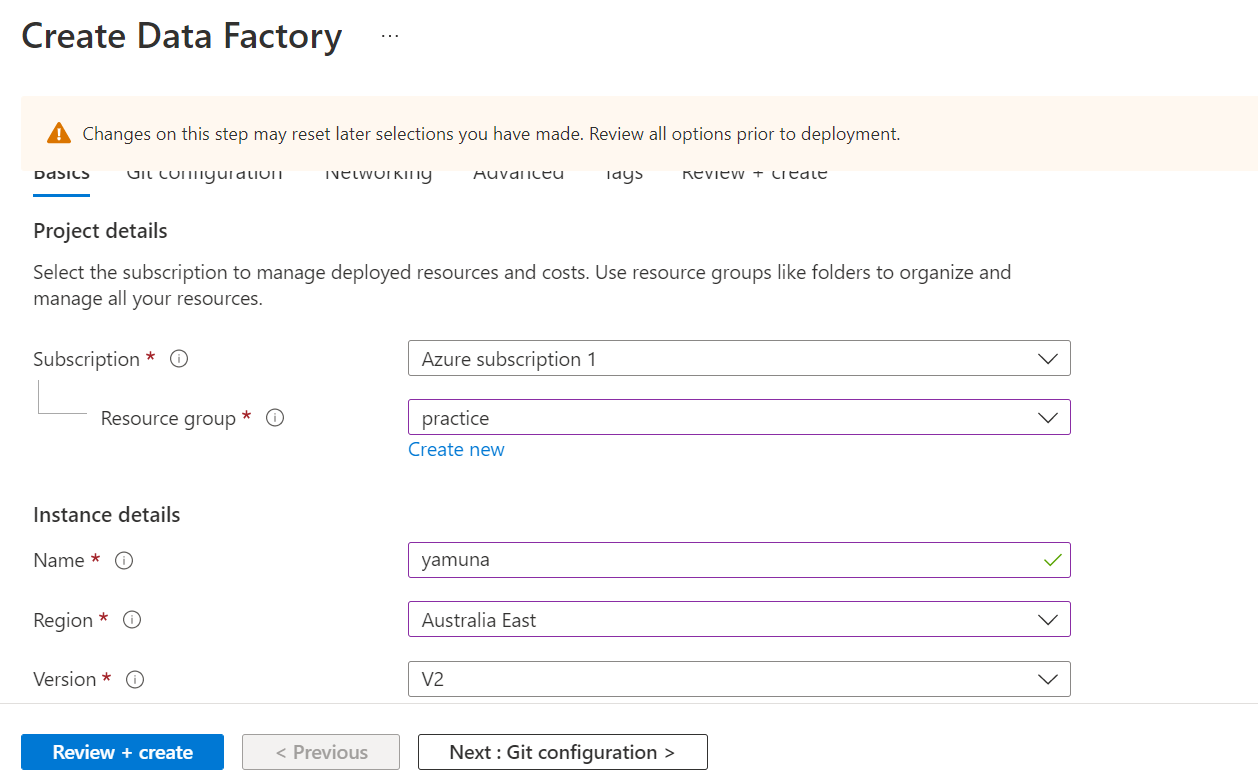
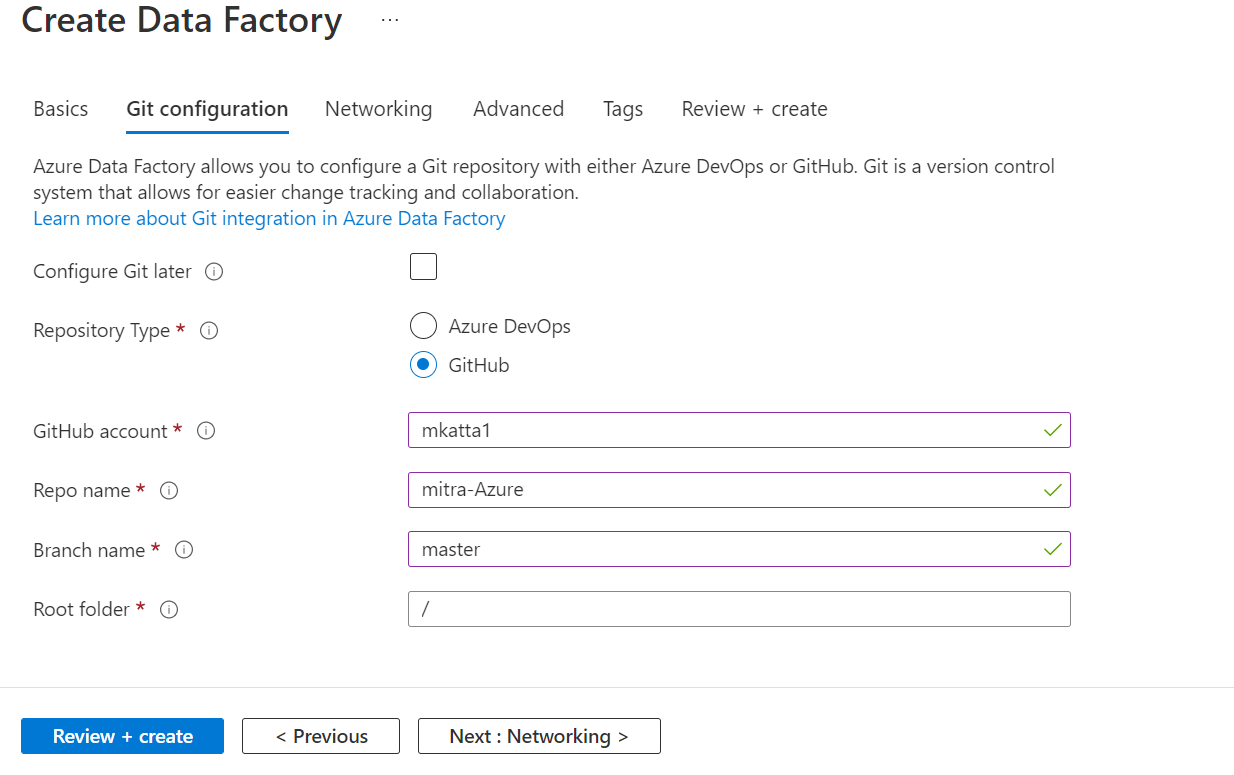
****

****

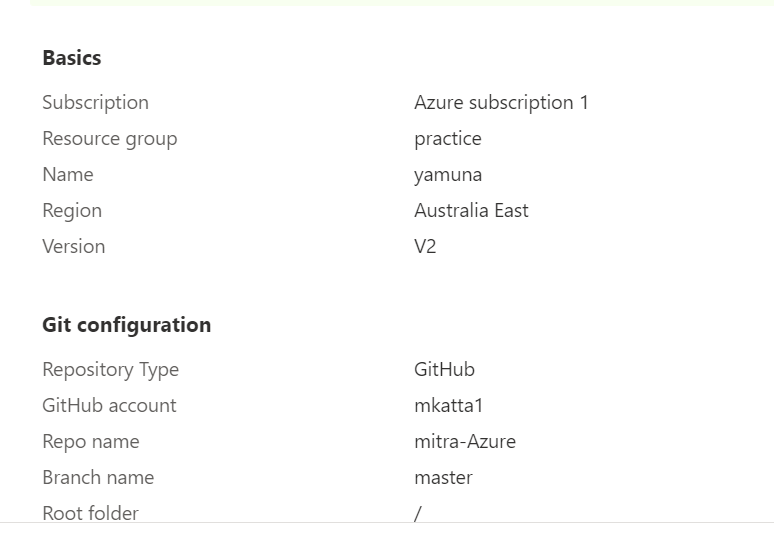
****

**Self-hosted integration runtime inbound connectivity to Azure Data Factory service**

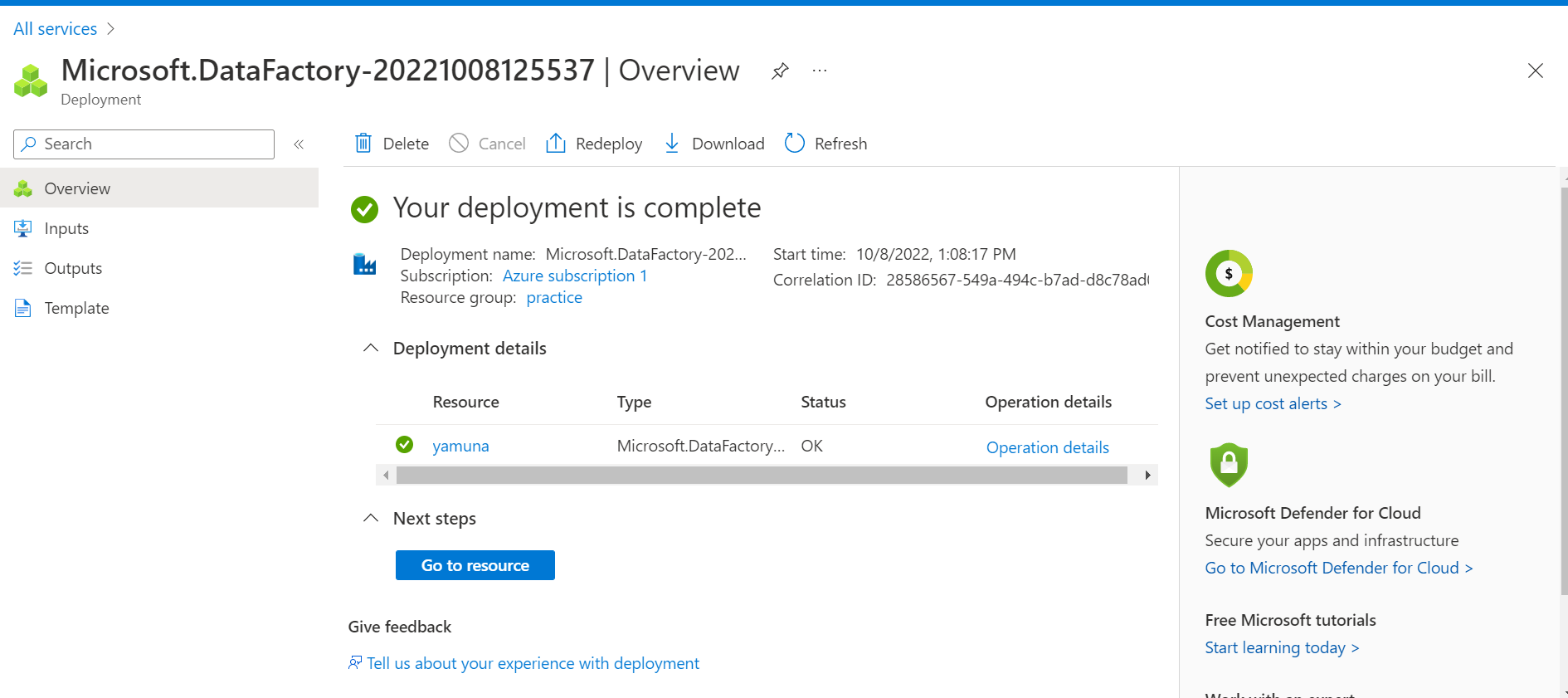
****



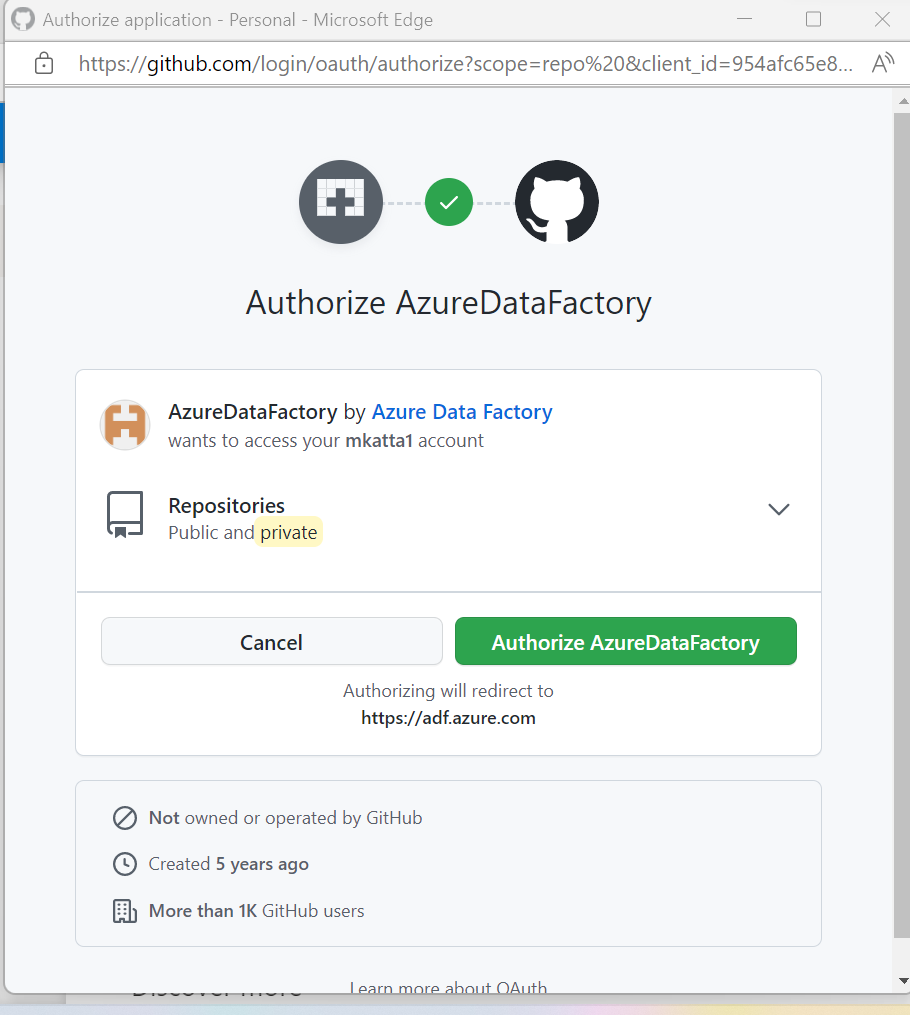
By default, data is encrypted with Microsoft-managed keys. For additional control over encryption keys, you can supply customer-managed keys to use for encryption of blob and file data. Customer-managed keys must be stored in an Azure Key Vault. You can either create your own keys and store them in a key vault, or you can use the Azure Key Vault APIs to generate keys. The storage account and the key vault must be in the same region, but they can be in different

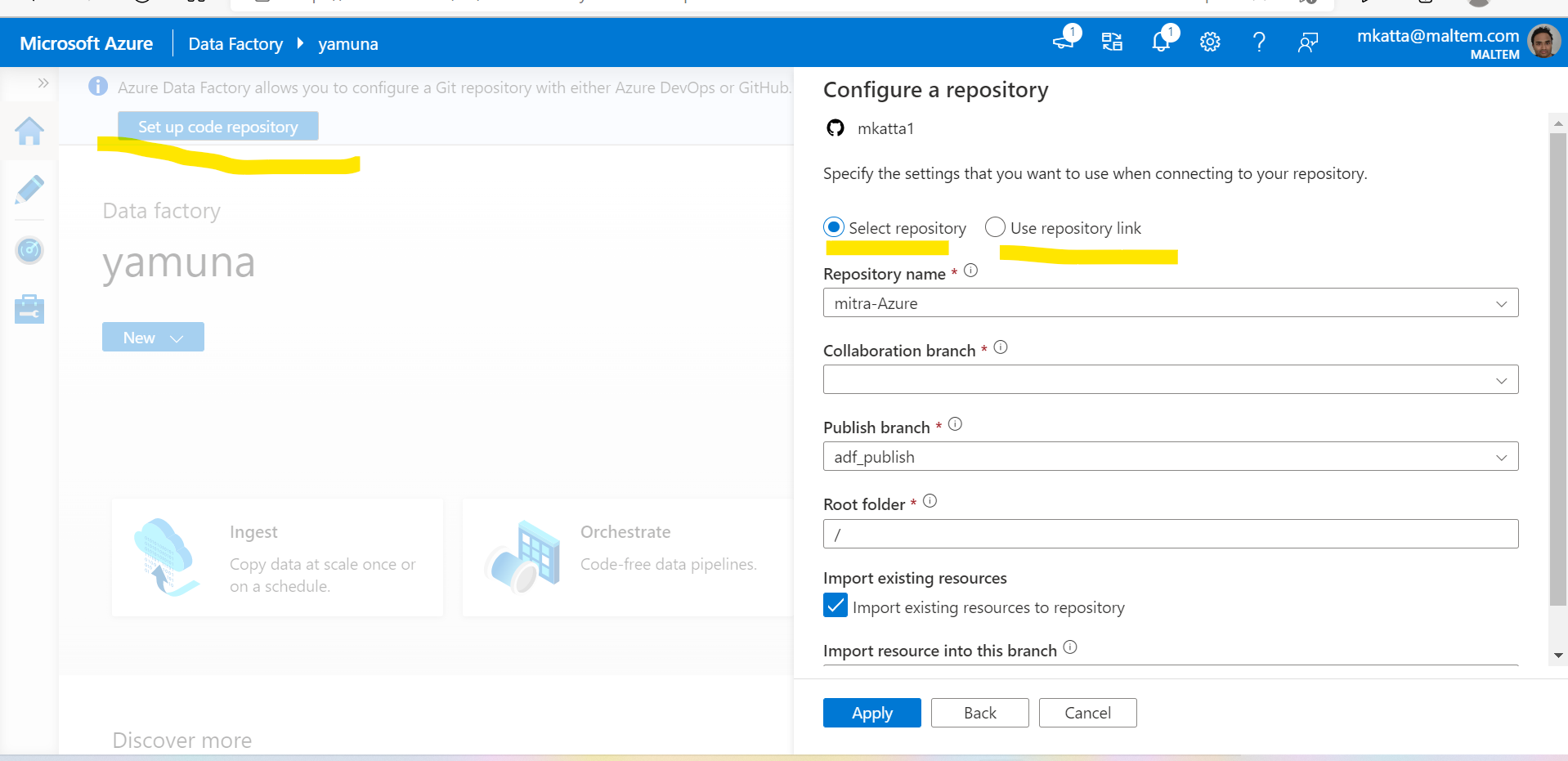


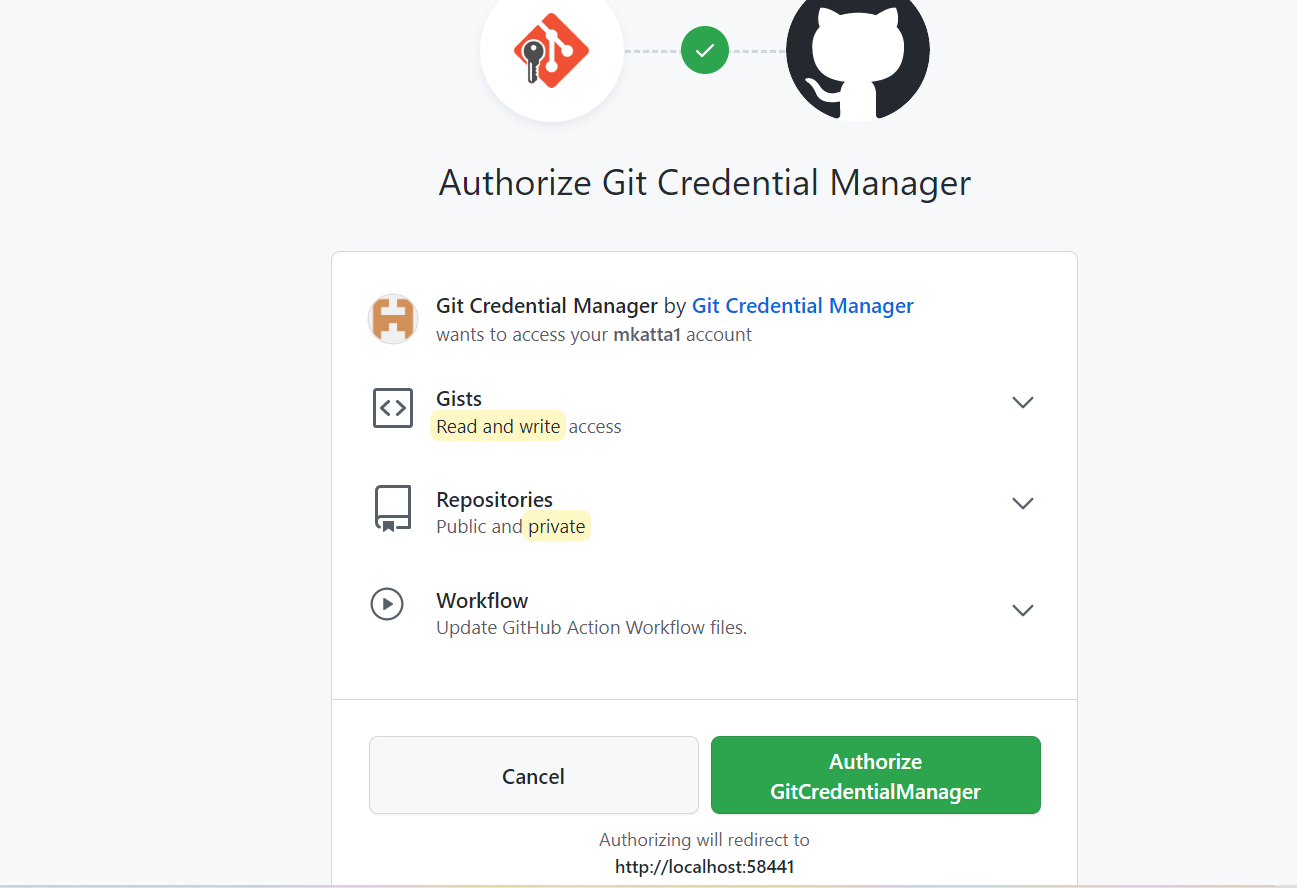
Deployed ADF:

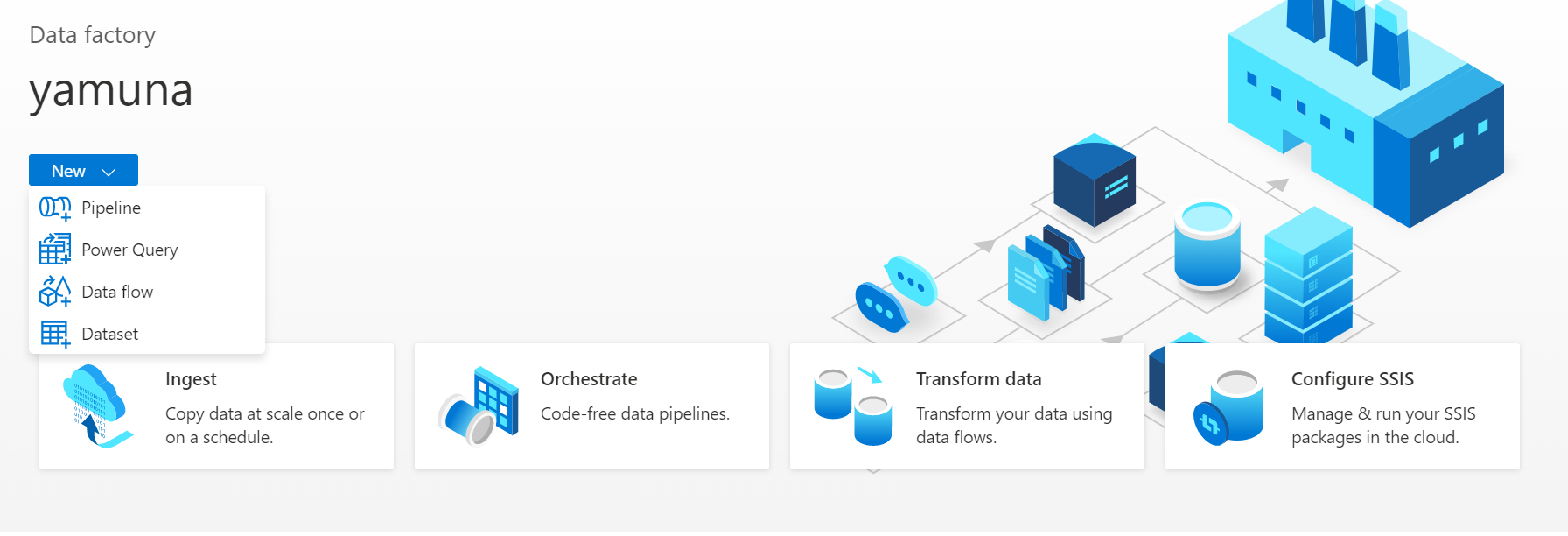


**Login into GitHub and authorize ADF:**



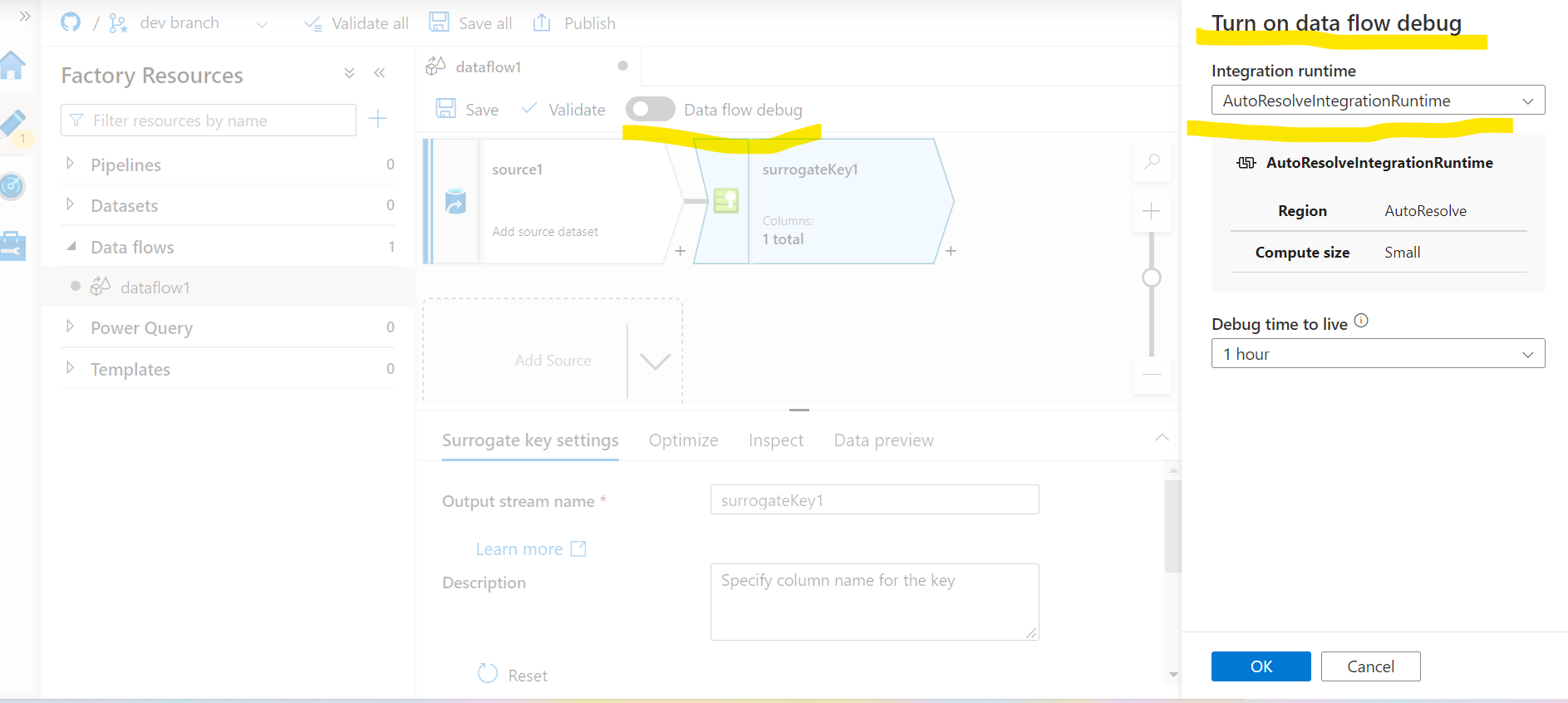


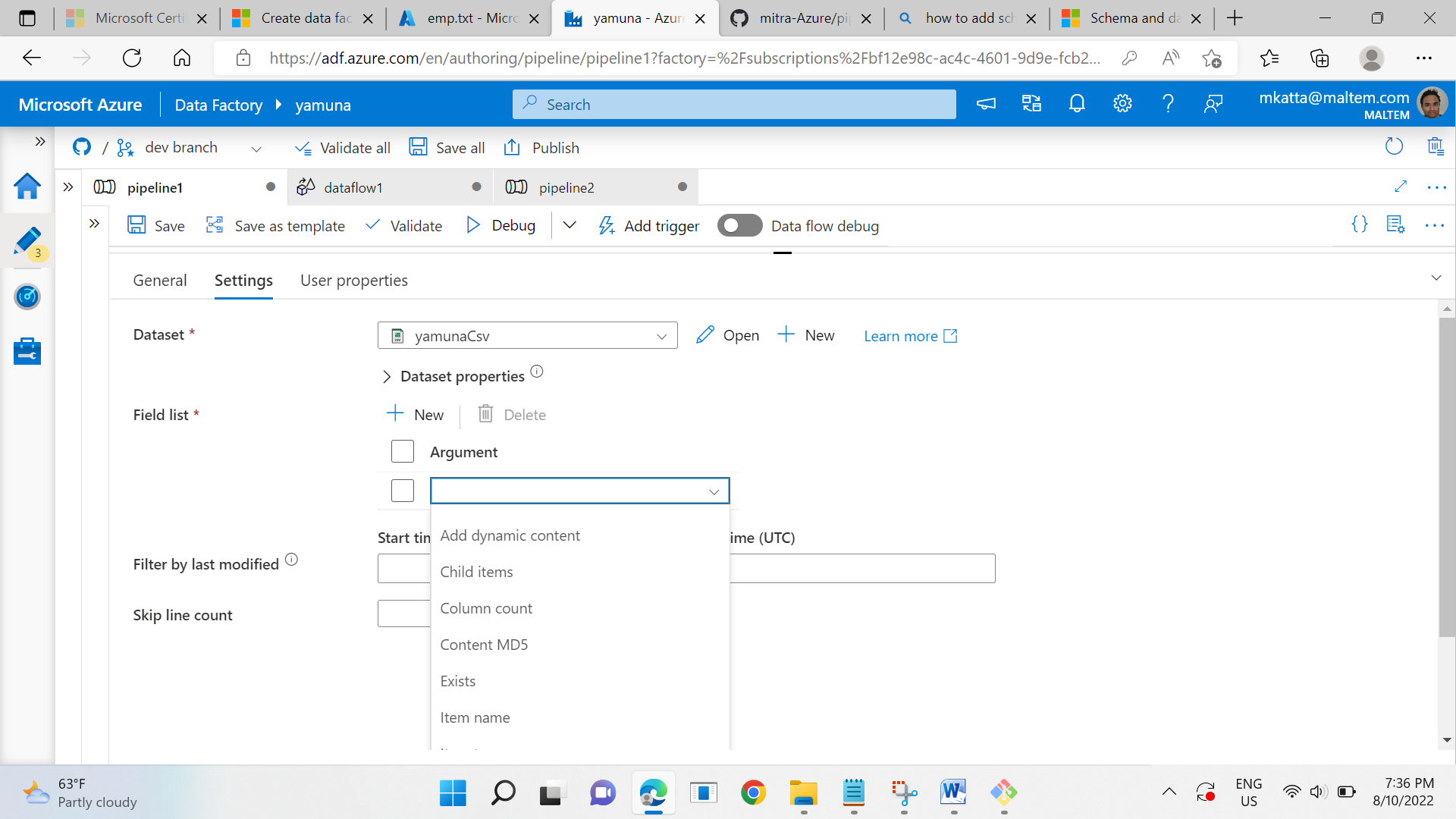




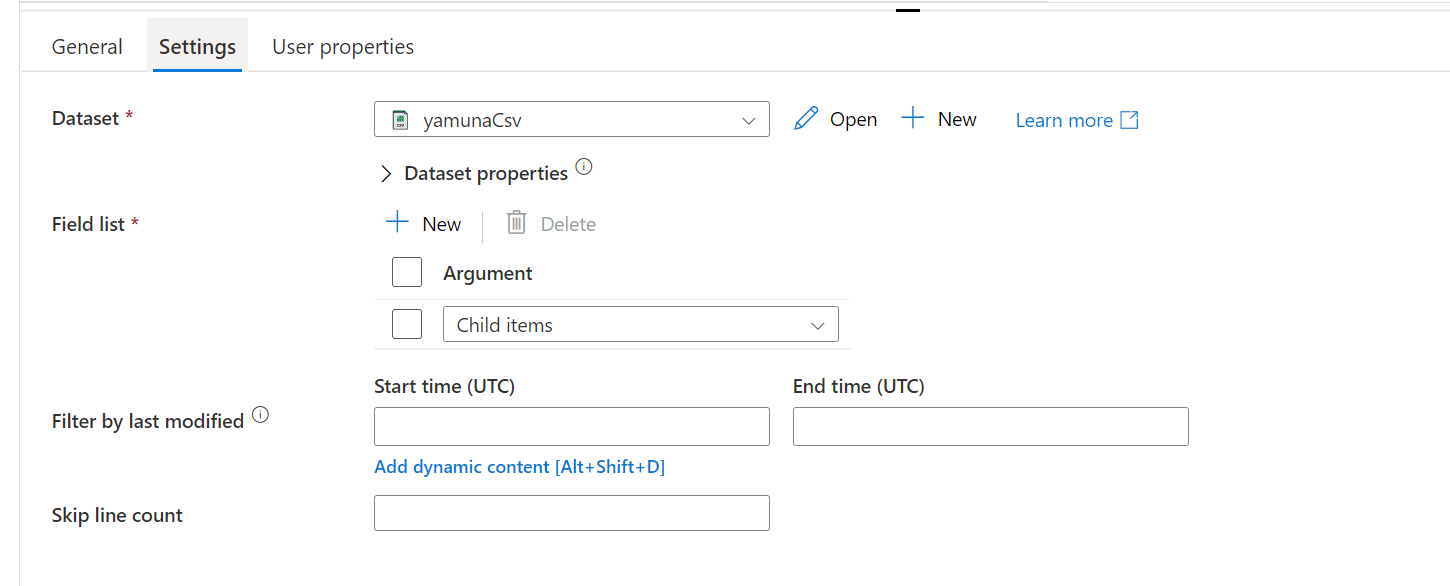
To preview data we need to select the data flow debug mode.

In auto resolve runtime it will select the region automatically





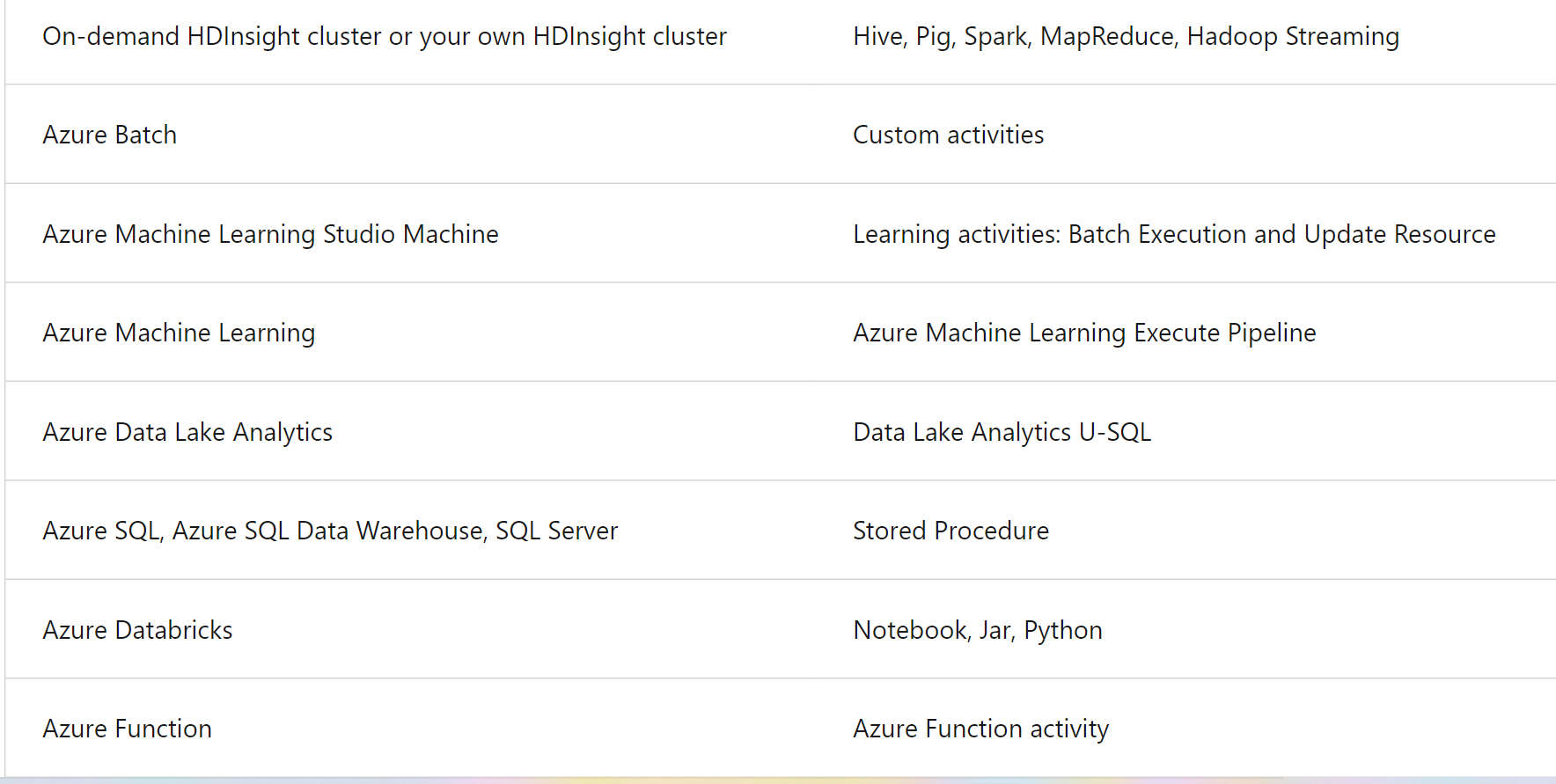
Item Type, Last Modified, size, structure



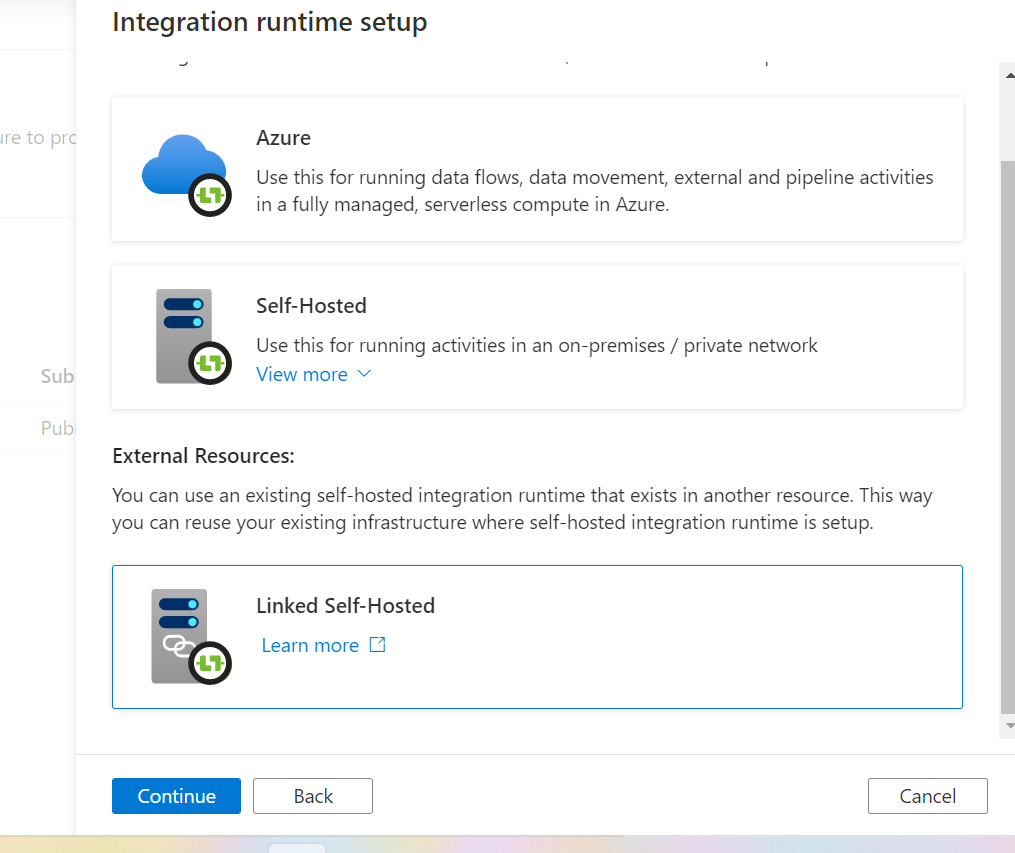
| **Control Activity** | **Description** |
| --- | --- |
| Execute Pipeline Activity | Execute Pipeline activity allows a Data Factory pipeline to invoke another pipeline. |
| ForEachActivity | ForEach Activity defines a repeating control flow in your pipeline. This activity is used to iterate over a collection and executes specified activities in a loop. The loop implementation of this activity is similar to Foreach looping structure in programming languages. |
| WebActivity | Web Activity can be used to call a custom REST endpoint from a Data Factory pipeline. You can pass datasets and linked services to be consumed and accessed by the activity. |
| Lookup Activity | Lookup Activity can be used to read or look up a record/ table name/ value from any external source. This output can further be referenced by succeeding activities. |
| Get Metadata Activity | GetMetadata activity can be used to retrieve metadata of any data in Azure Data Factory. |
| Until Activity | Implements Do-Until loop that is similar to Do-Until looping structure in programming languages. It executes a set of activities in a loop until the condition associated with the activity evaluates to true. You can specify a timeout value for the until activity in Data Factory. |
| If Condition Activity | The If Condition can be used to branch based on condition that evaluates to true or false. The If Condition activity provides the same functionality that an if statement provides in programming languages. It evaluates a set of activities when the condition evaluates to true and another set of activities when the condition evaluates to false. |
| Wait Activity | When you use a Wait activity in a pipeline, the pipeline waits for the specified period of time before continuing with execution of subsequent activities. |

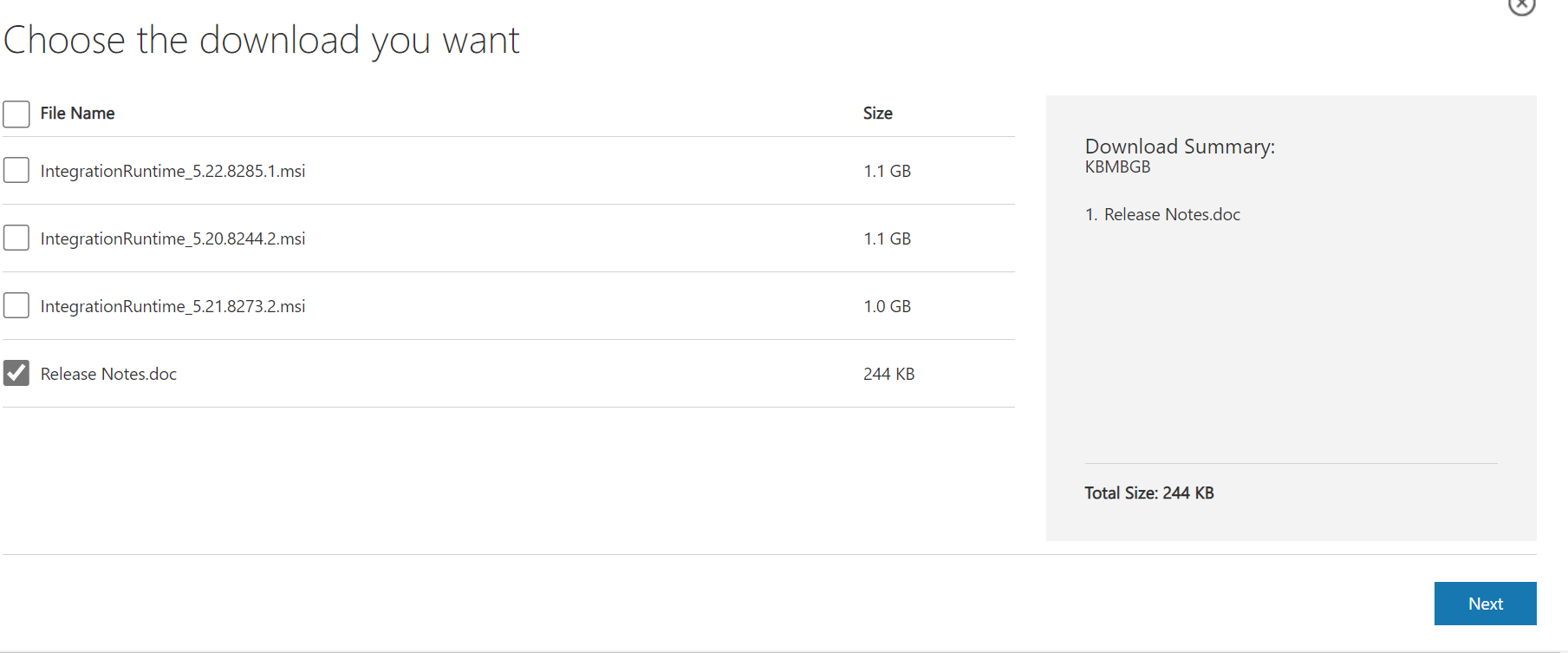
The following table describes properties in the above JSON:

| **Property** | **Description** | **Required** |
| --- | --- | --- |
| name | Name of the activity. | Yes |
| description | Text describing what the activity or is used for. | No |
| type | Defines the type of the activity. | Yes |
| linkedServiceName | Name of the linked service used by the activity. | Yes for HDInsight, Machine Learning Batch Scoring Activity and Stored Procedure Activity |
| typeProperties | Properties in the typeProperties section depend on each type of activity. | No |
| policy | Policies that affect the run-time behavior of the activity. This property includes timeout and retry behavior. | No |
| dependsOn | This property is used to define activity dependencies, and how subsequent activities depend on previous activities. | No |



Azure Data Factory provides the ability to lift and shift existing SSIS workload, by creating an Azure-SSIS Integration Runtime to natively execute SSIS packages, and will enable you to deploy and manage your existing SSIS packages with little to no change using familiar **tools such as SQL Server Data Tools (SSDT) and SQL Server Management Studio (SSMS)**, just like using SSIS on-premises.



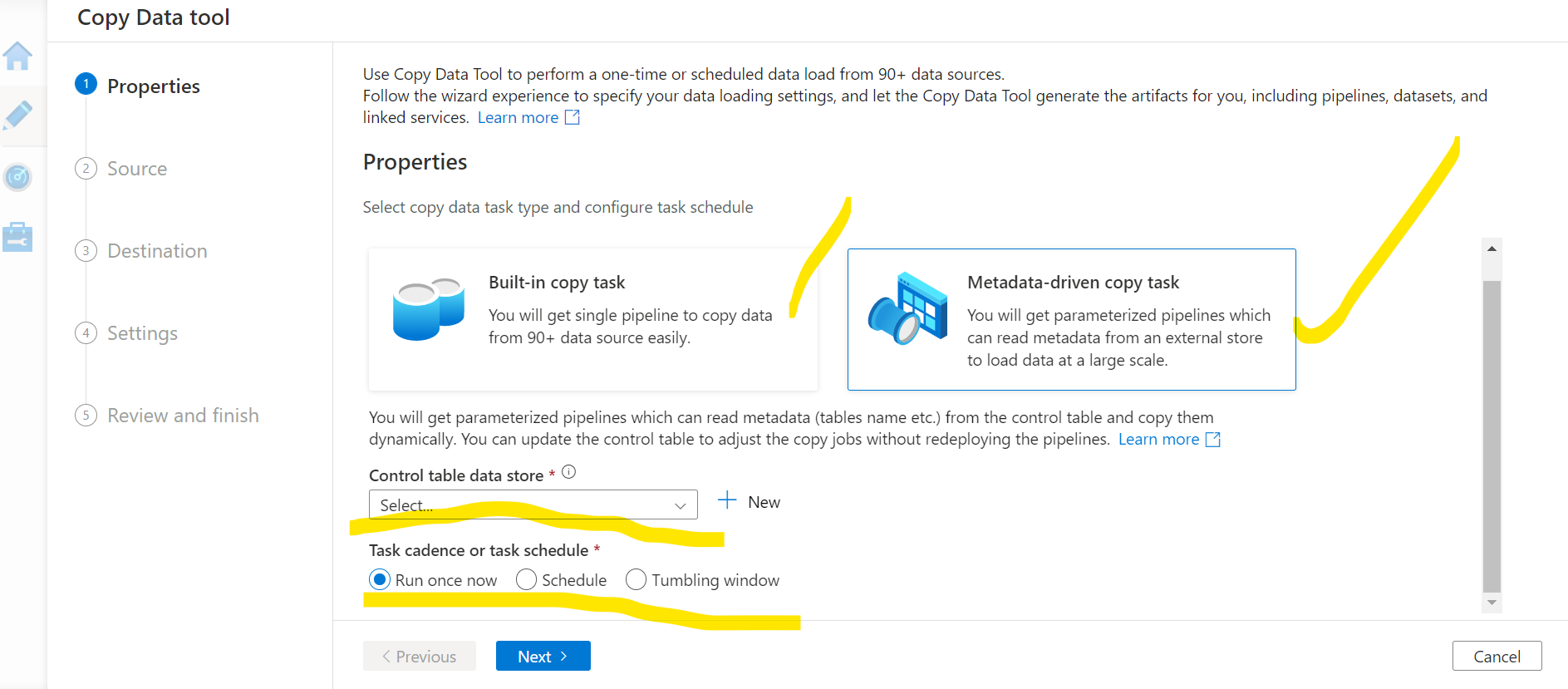


READ THIS CLEARLY:

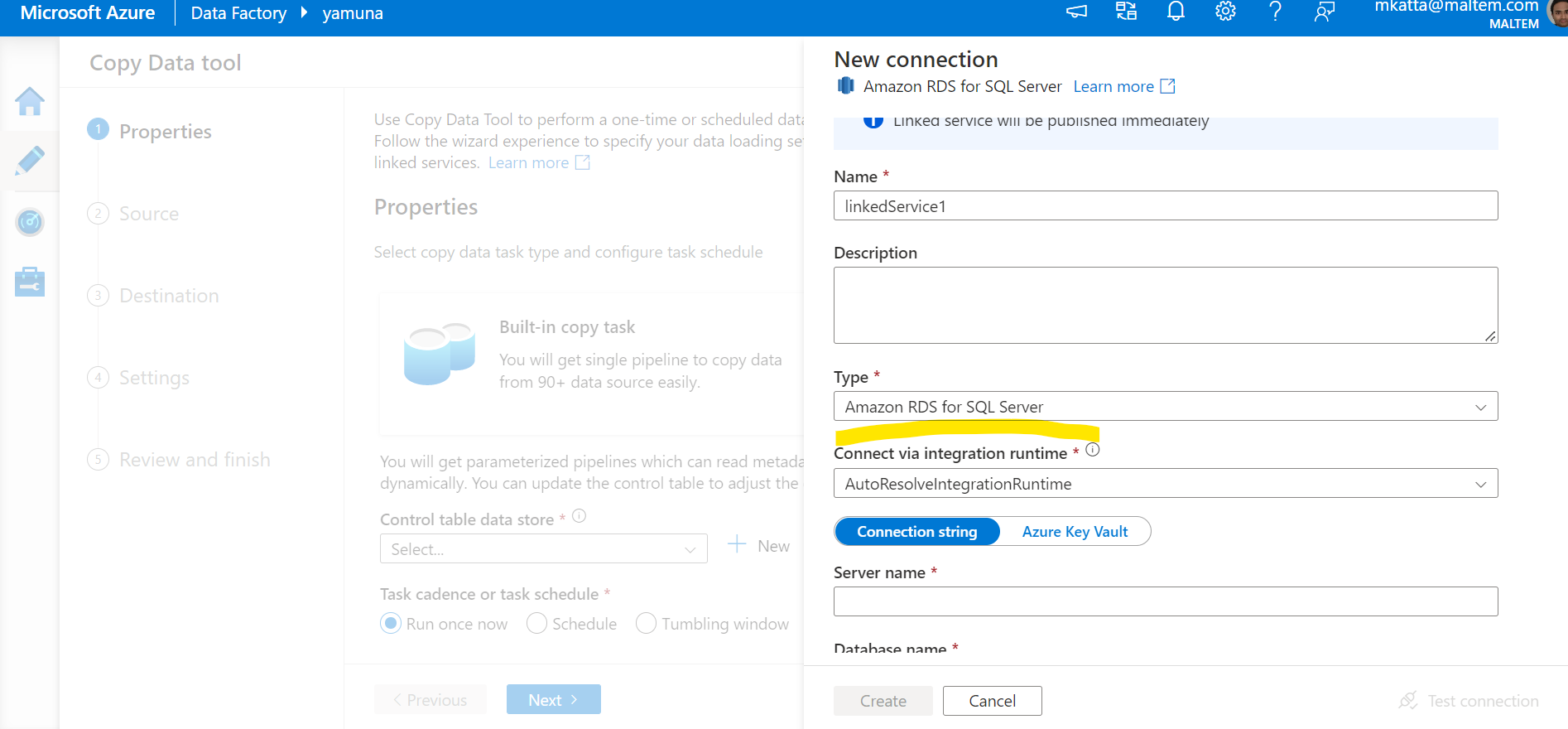
[Understand data ingestion security considerations - Training | Microsoft Learn](https://learn.microsoft.com/en-us/training/modules/petabyte-scale-ingestion-azure-data-factory/7-understand-data-ingestion-security-considerations)

**COPY DATA TOOL IN ADF:**

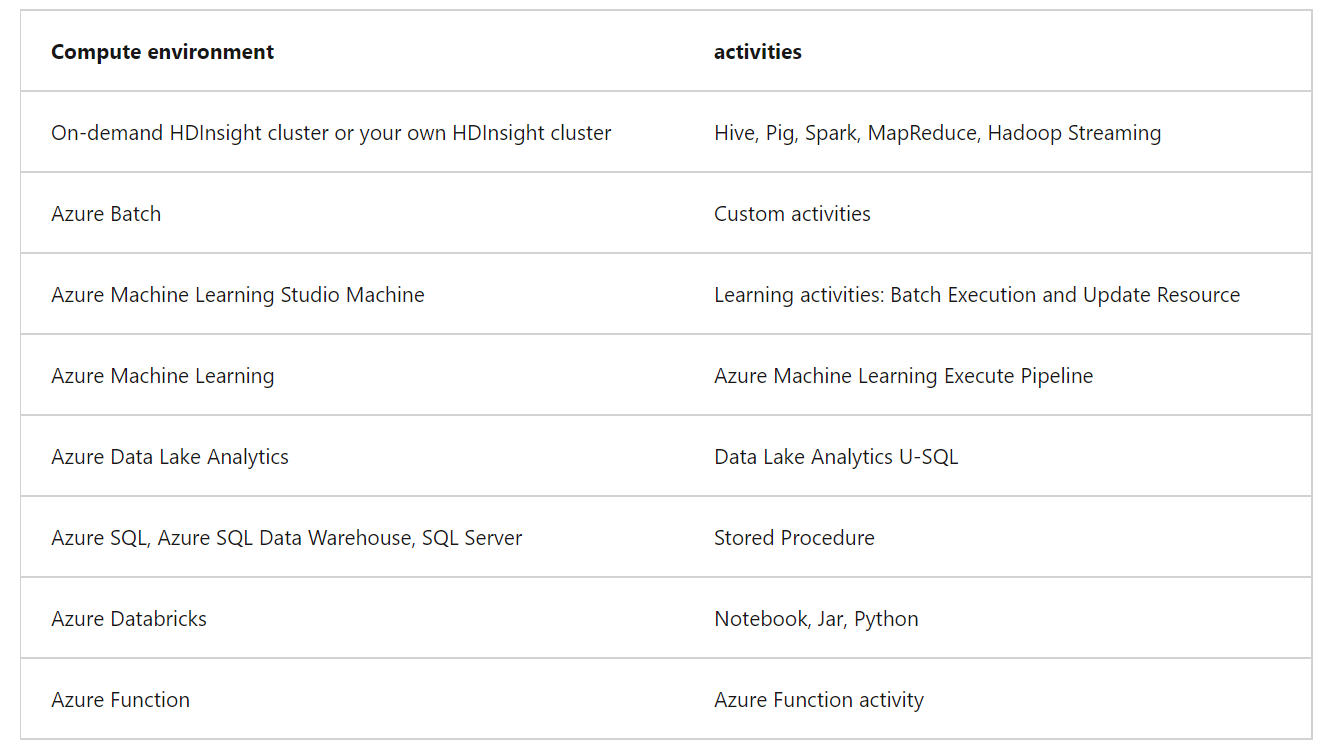
Copy data tool to perform one time or scheduled data load from 90+ data sources.



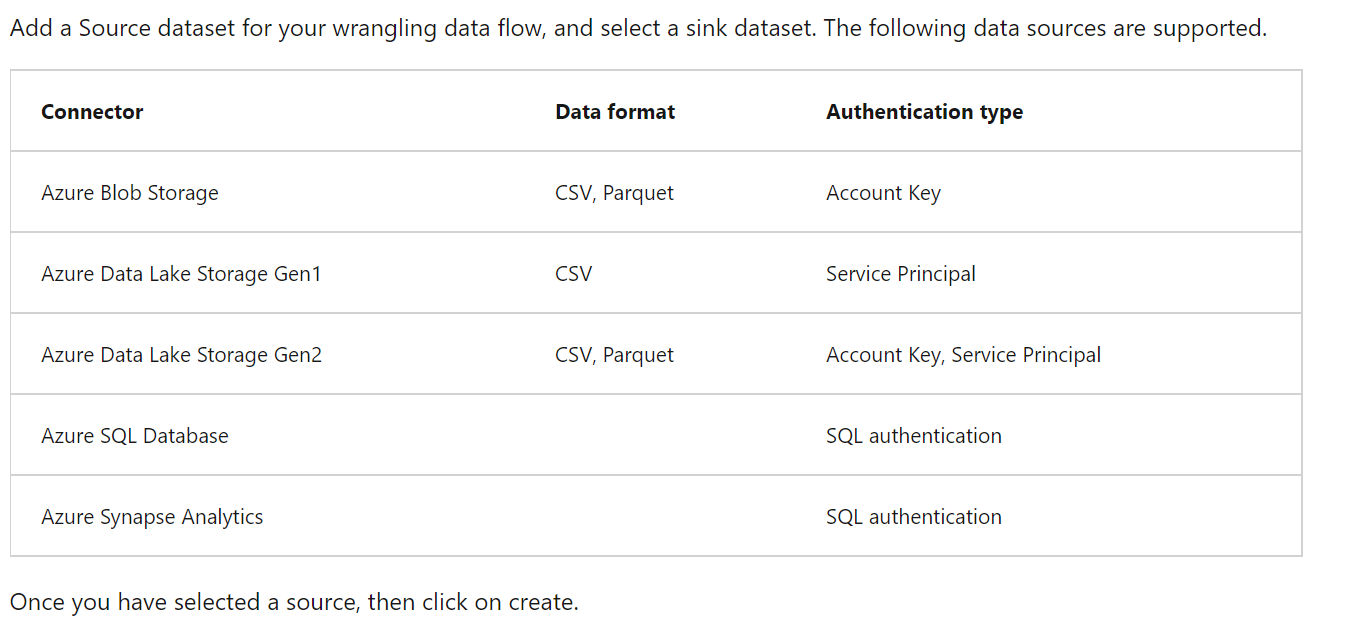
We can select instead of Amazon RDS for SQL Server also like Azure Synapse, Azure SQL database, Azure SQL Databse Managed Instance, and SQL SERVER.



**Transforming data using compute resoures:**



**Clicking Debug will provision the Spark clusters required to interact with the Mapping Data Flow transformations. On turning Debug on, you will be prompted to select the Integration Runtime that you require to use in the environment. If you select AutoResolveIntegrationRuntime, a cluster with eight cores that will be available with a time to live value of 60 minutes.**

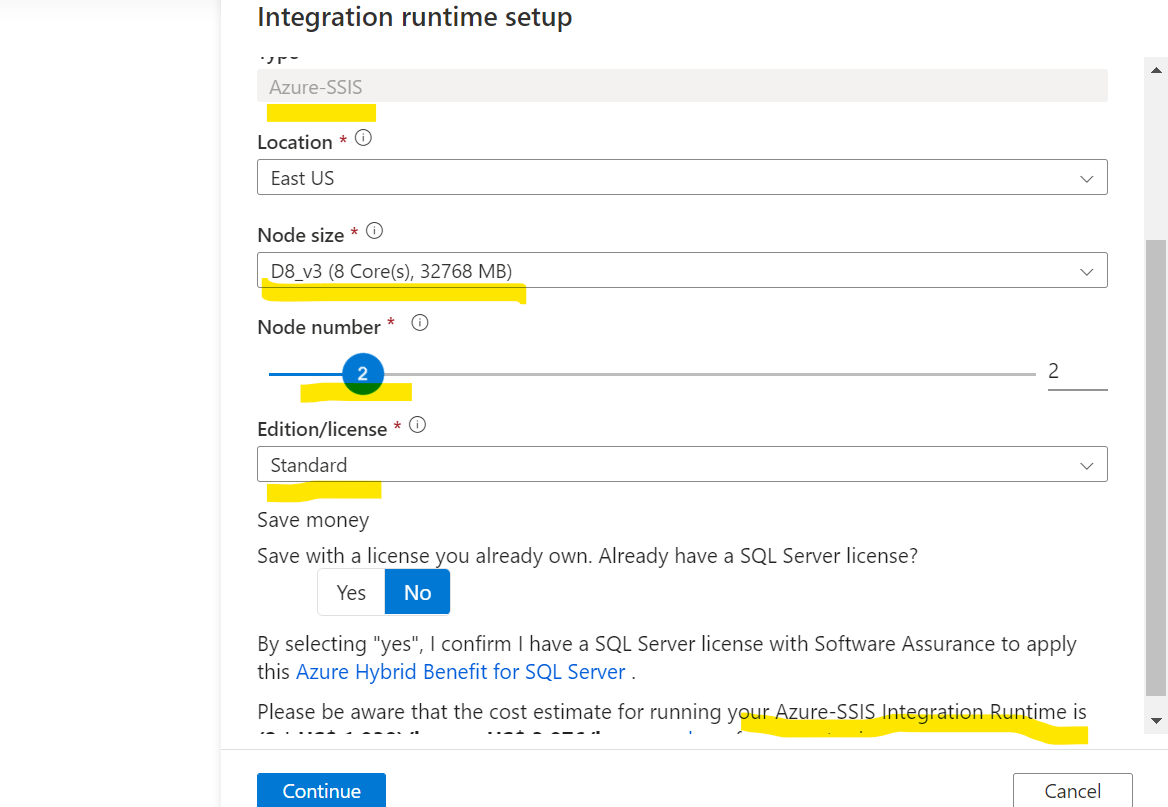


**Exercise - Use Azure Data Factory wrangling data**

This is related to power query which is not in place:

[Exercise - Use Azure Data Factory wrangling data - Training | Microsoft Learn](https://learn.microsoft.com/en-us/training/modules/code-free-transformation-scale/6-use-wrangling-data)

**AZURE SSIS INTEGRATION RUNTIME NOTES:**

****

# Exercise - Design and implement a Type 1 slowly changing dimension with mapping data flows

# CHECK THIS TOPIC PERFECTLY IN ADF

## Map dataflow debug

Sometimes, you don't want to debug the whole pipeline but test a part of the pipeline. A Debug run allows you to do just that. You can test the pipeline end to end or set a breakpoint. By doing so in debug mode, you can interactively see the results of each step while you build and debug your pipeline.

After the debug mode is on, you will actually build the data flow with an active Spark Cluster. The Spark cluster will close once the debug is off. You do have a choice in what compute you're going to use. When you use an existing debug cluster, it will reduce the start-up time. However, for complex or parallel workloads you might want to spin up your own just-in-time cluster.

**Global parameters are constants across a Data Factory that can be consumed by a pipeline in any expression.**

