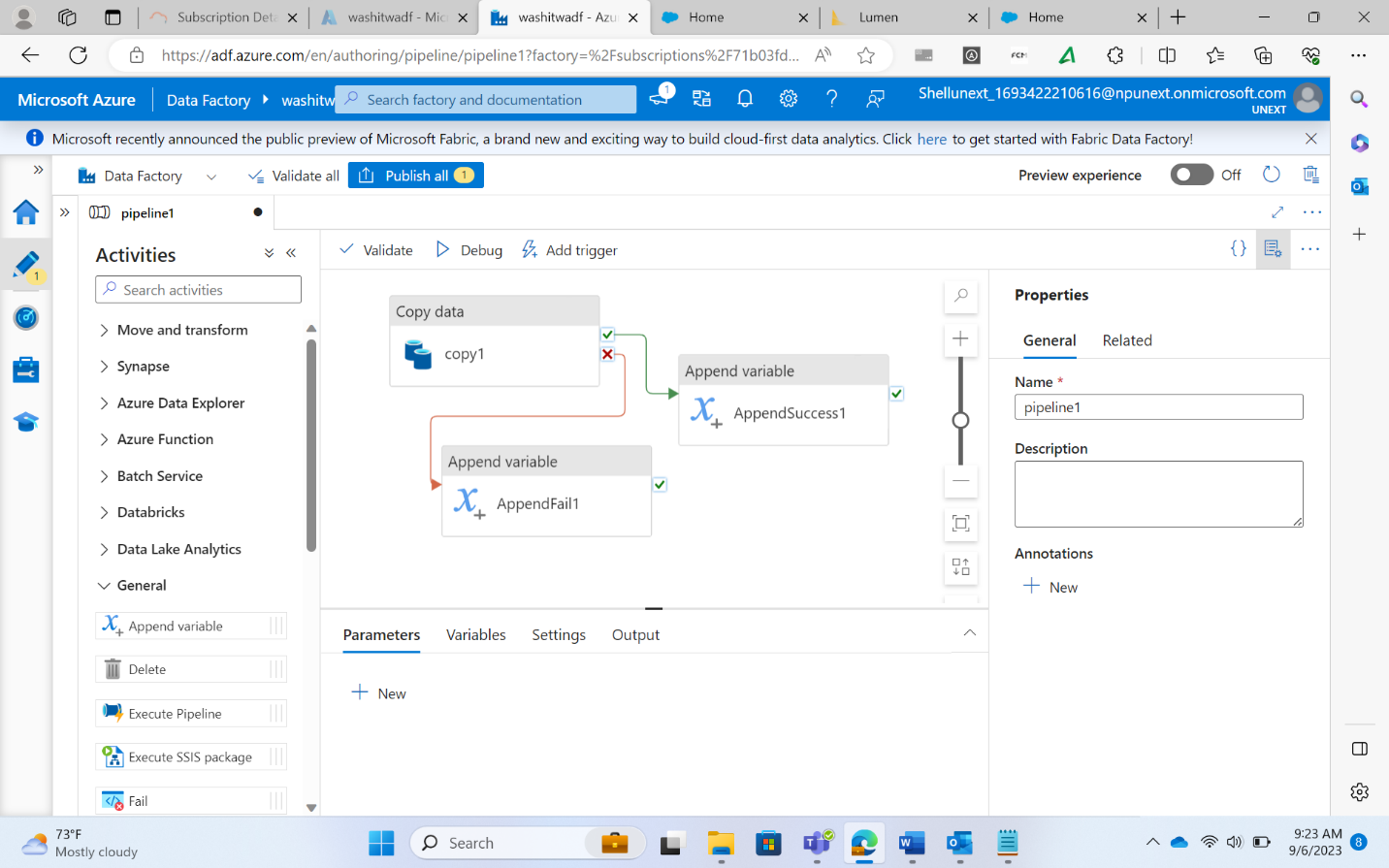
PIPELINE AND ACTIVITIES

Activities

* Learn online
  + [Append Variable Activity - Azure Data Factory & Azure Synapse | Microsoft Learn](https://learn.microsoft.com/en-us/azure/data-factory/control-flow-append-variable-activity)
* In one activity no more than 40 activities can be present
* If more than 40 activities are required, then create a new pipeline containing the additional activities
* Then use a pipeline activity to connect one pipeline to another and execute it
* Here use the Master pipeline to invoke the other pipeline
* This process is called **Pipeline Chaining**
* Set Variable
  + To set a variable with some values
* Append
  + To append values to a variable
  + The variable must be array type



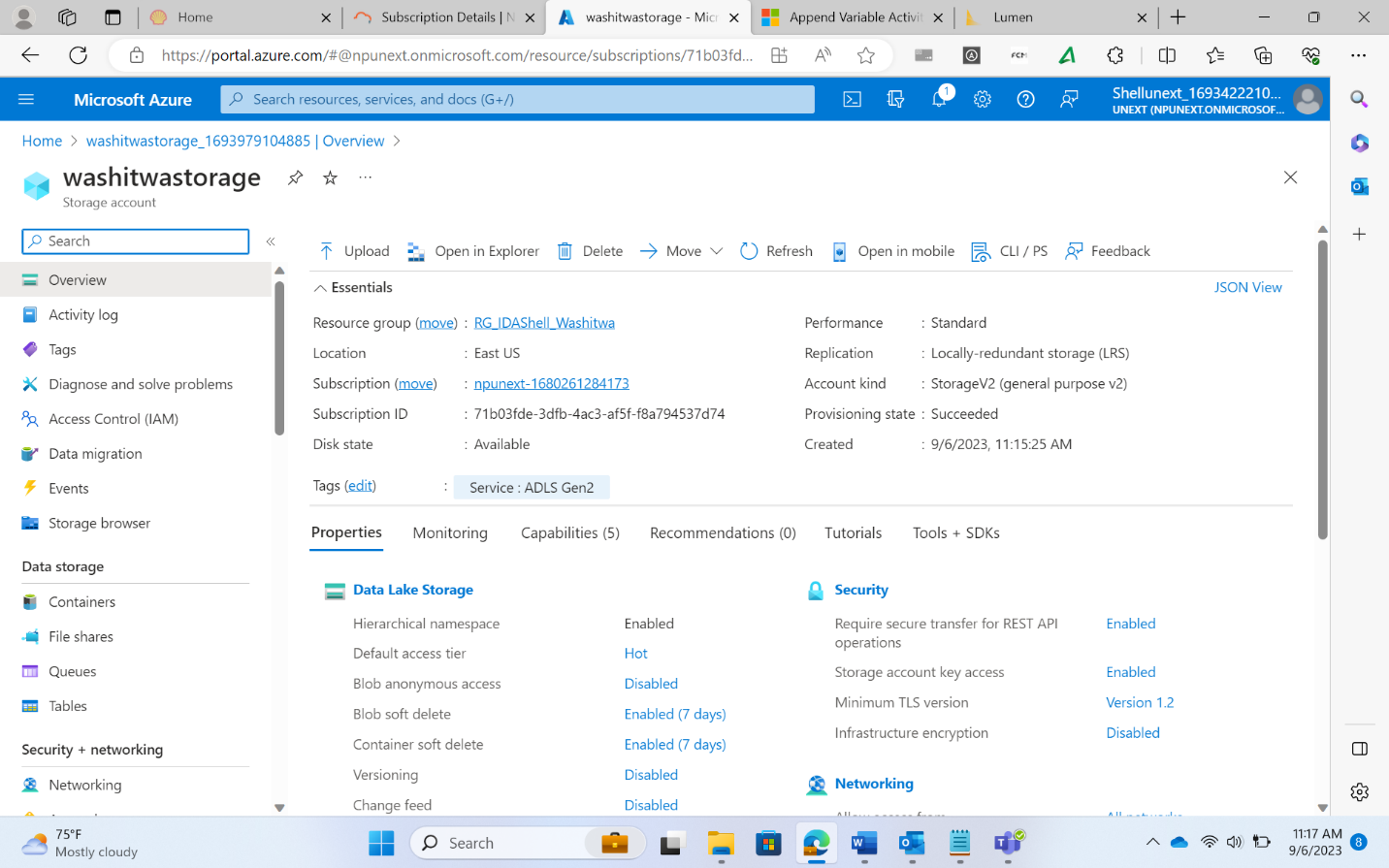
* Delete
  + To delete the values
* Execute SSIS package
  + To execute SSIS package
  + Only for this activity, Azure SSIS Runtime is required
* Get Metadata
  + To check the metadata of a file/folder
  + It gives all the details of the file/folder like, size, name, etc.
* Filter
  + To filter some data
  + Example: filter file sizes greater than some value
* Lookup
  + To get some data from a table
  + We can use normal SQL queries for this
* Set Variable
  + Supports all data types unlike Append Variable which only supports arrays
* Web Hook
  + Allows altering the behavior of a web page
* Wait
  + To make some activity wait for a specified time before executing

Pipeline Expression Builder

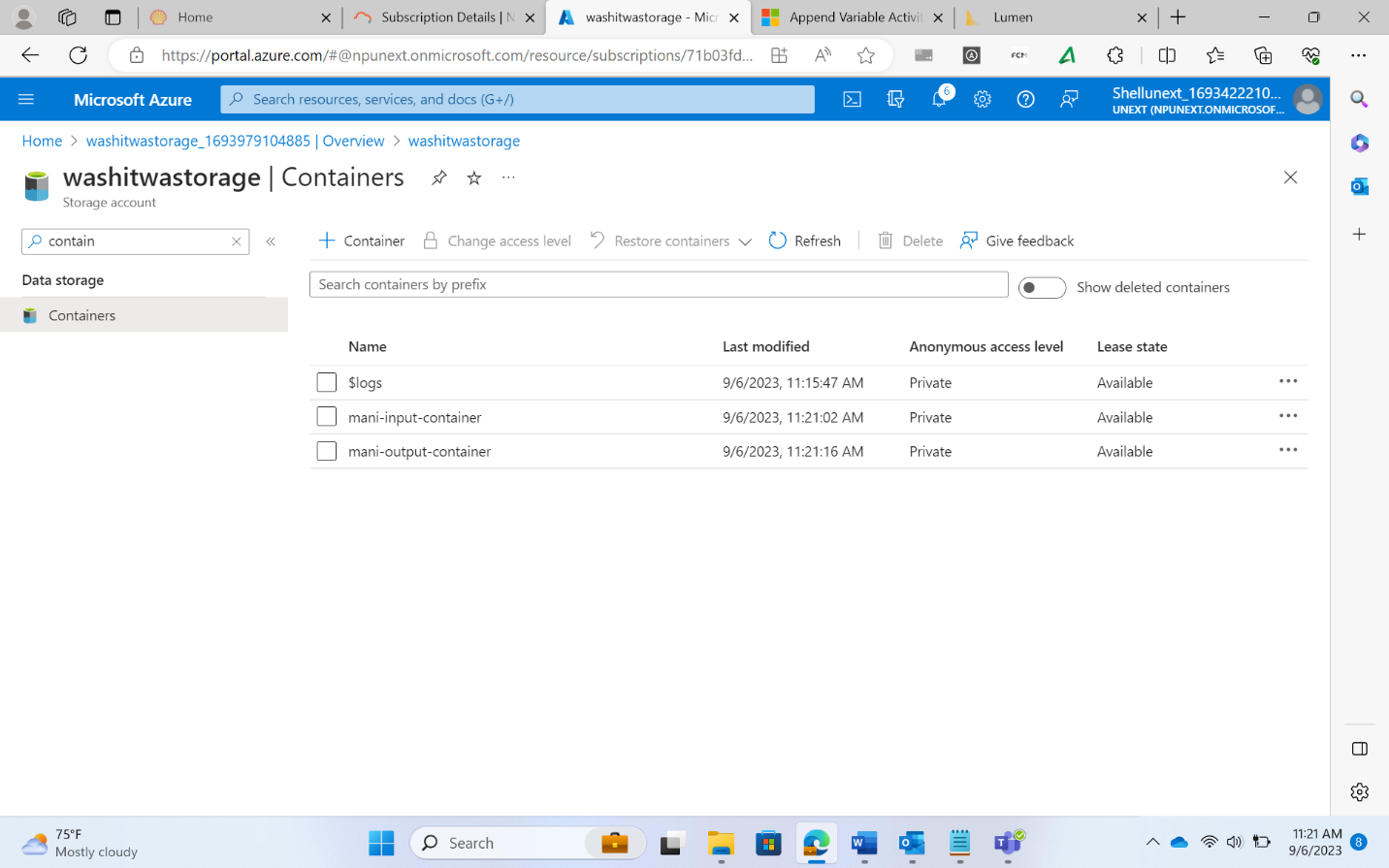
* To build expressions
* These expressions work as commands that return specific outputs upon execution like any programming language

Pipeline Creation from scratch

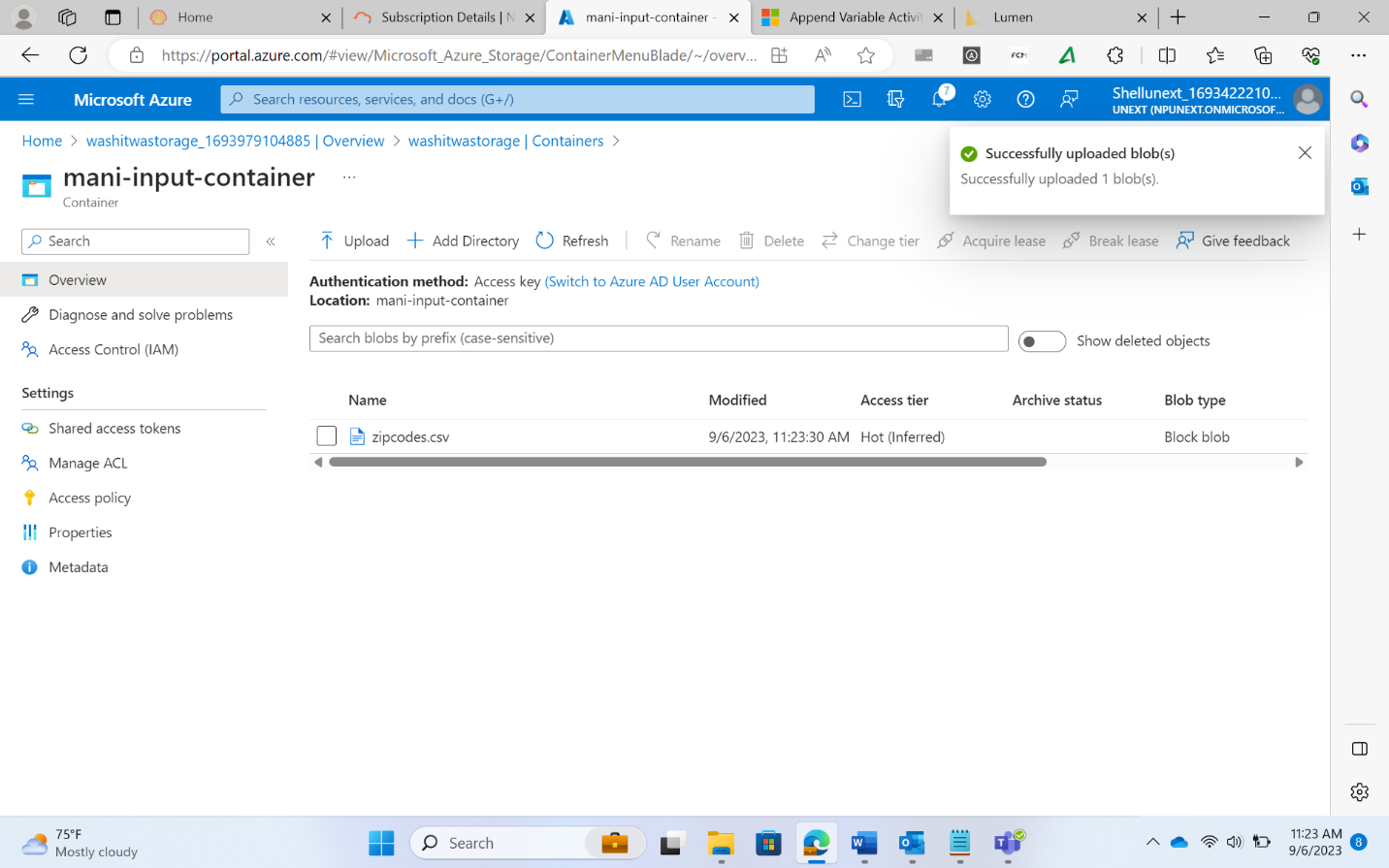
* Creation of Storage Account, type: ADLS Gen2



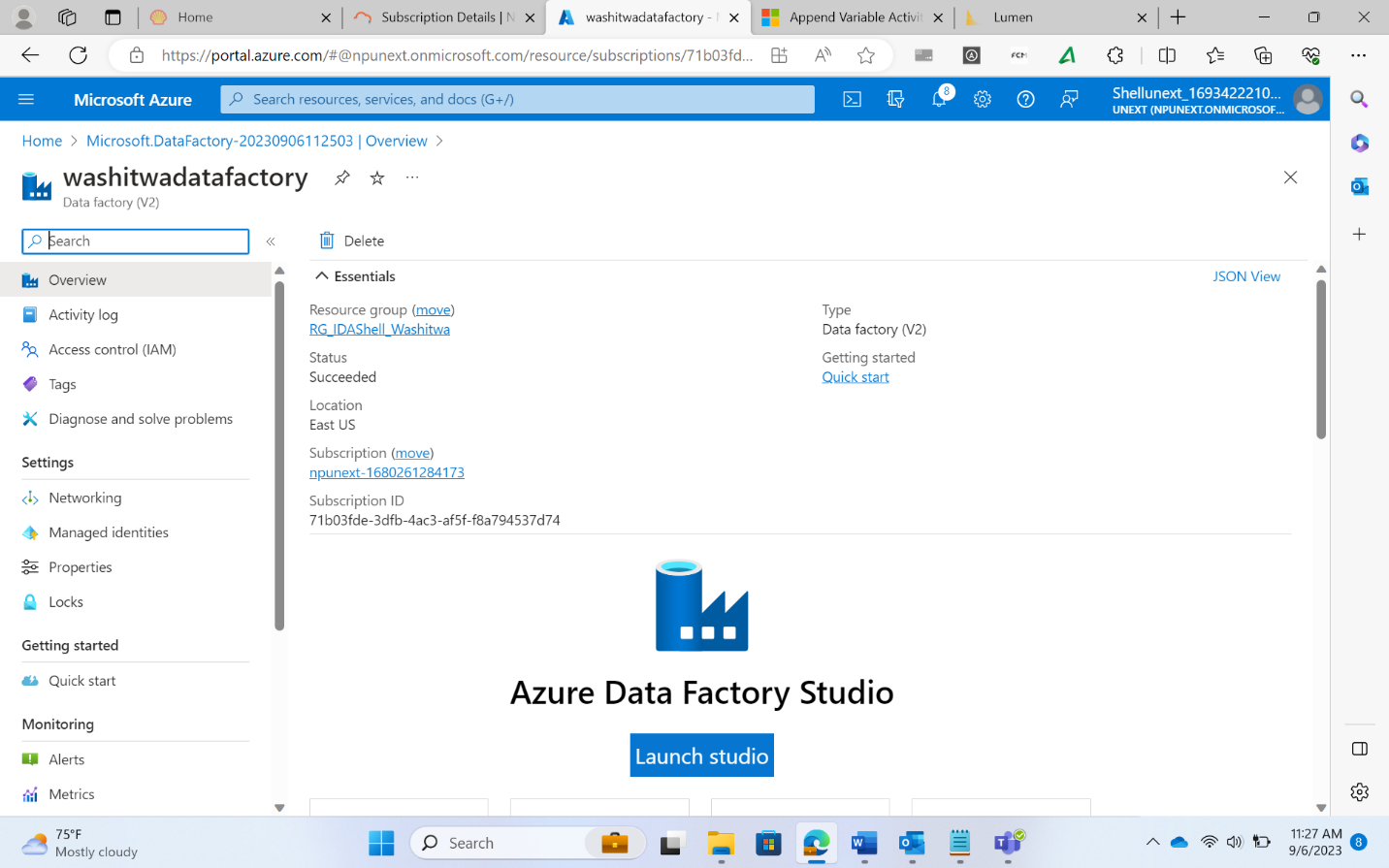
* Inside the storage account, create input and output containers



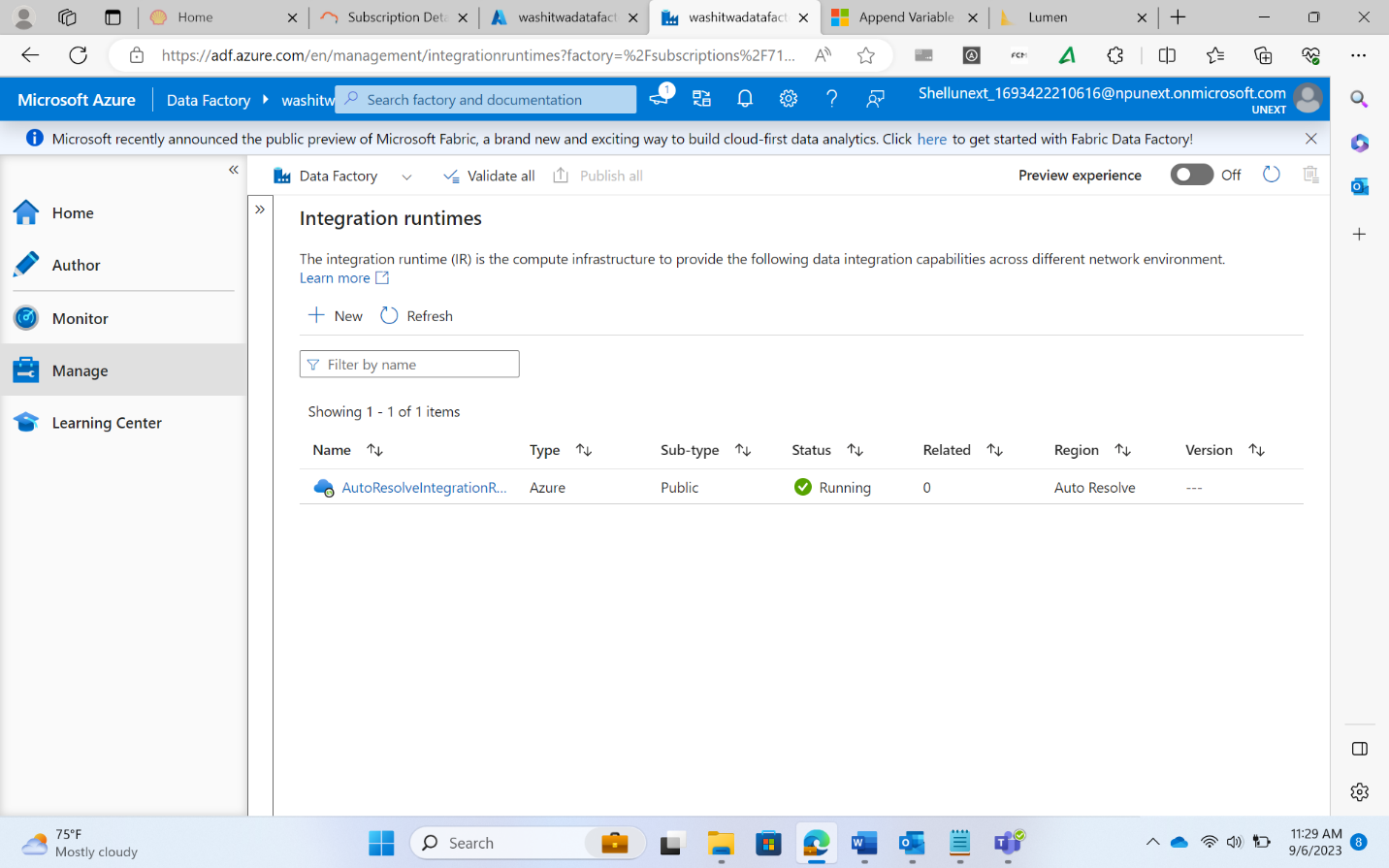
* Inside the input container, upload the csv file



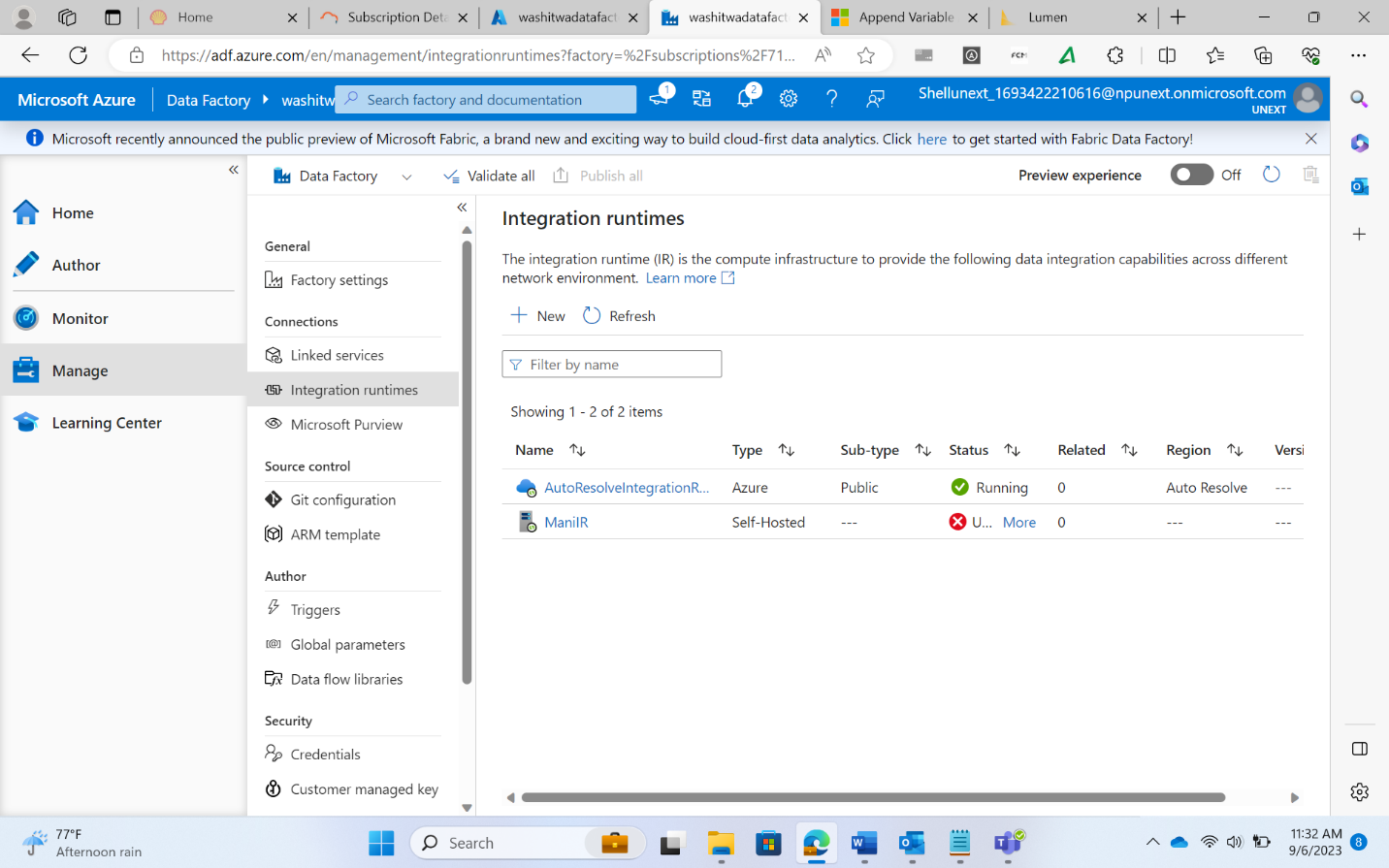
* Create a Data factory



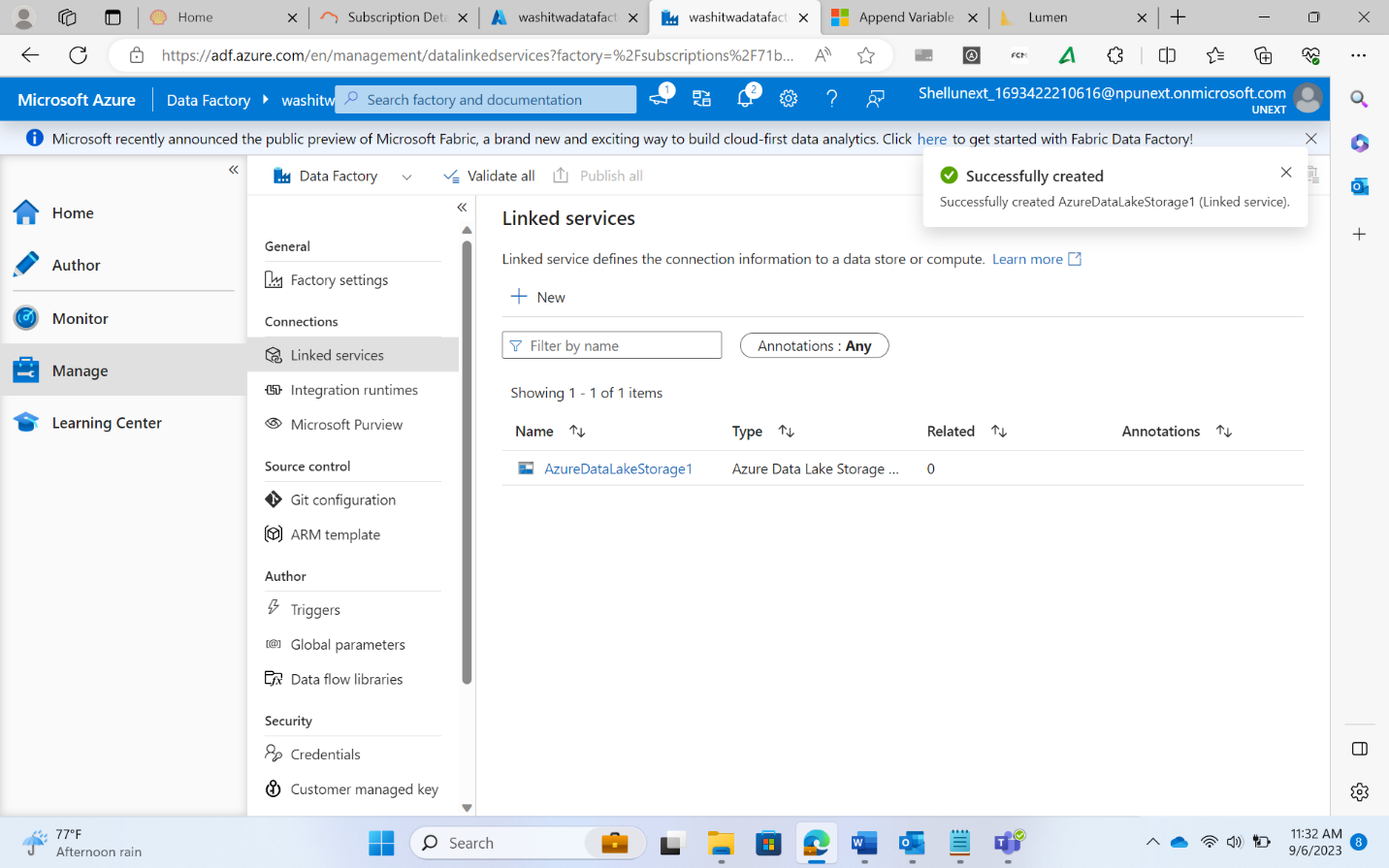
* Launch the Data Factory IDE



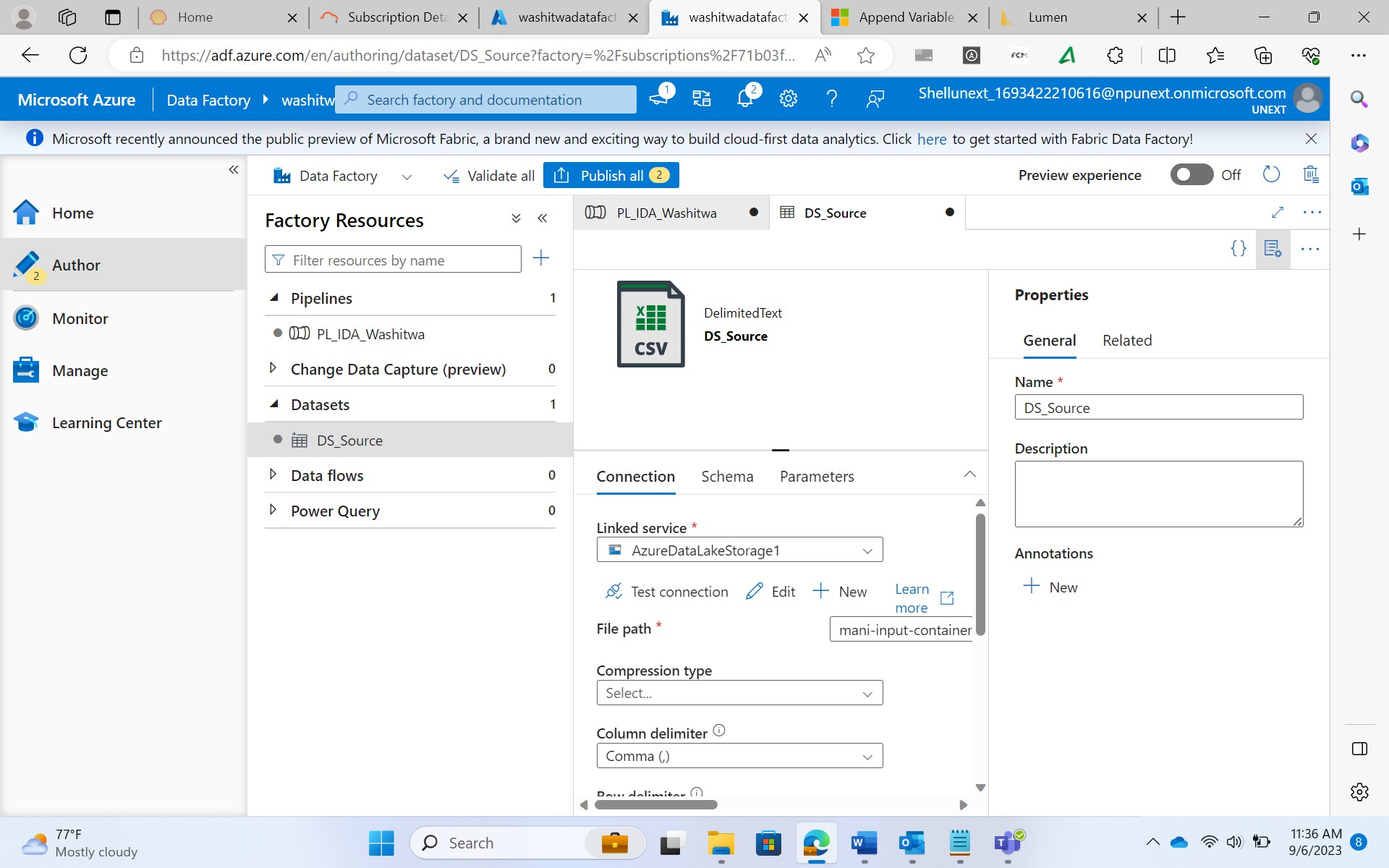
* Create an Integration Runtime



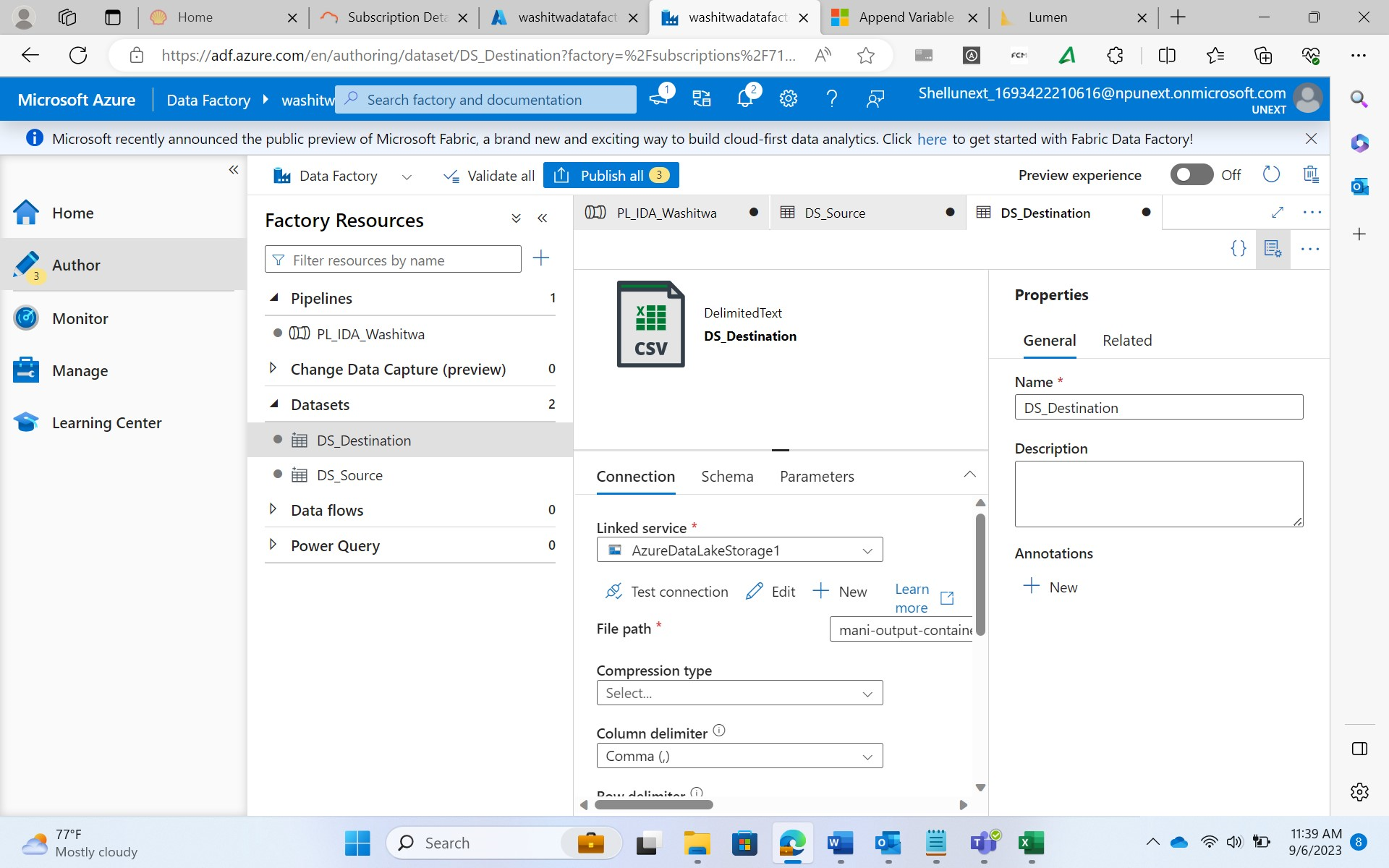
* Create a Linked Service with ADLS Gen2



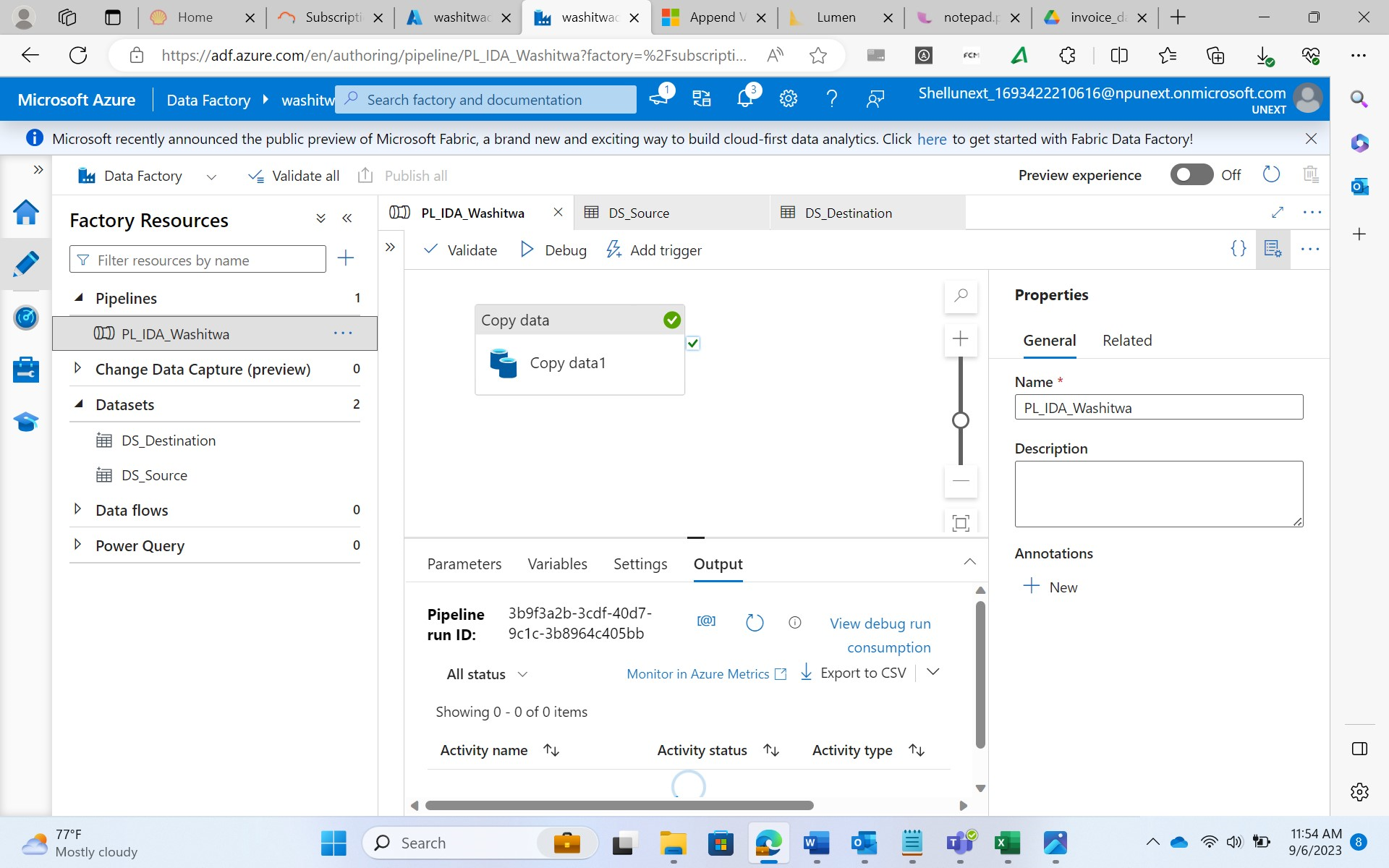
* Create 2 datasets for source and destination folders



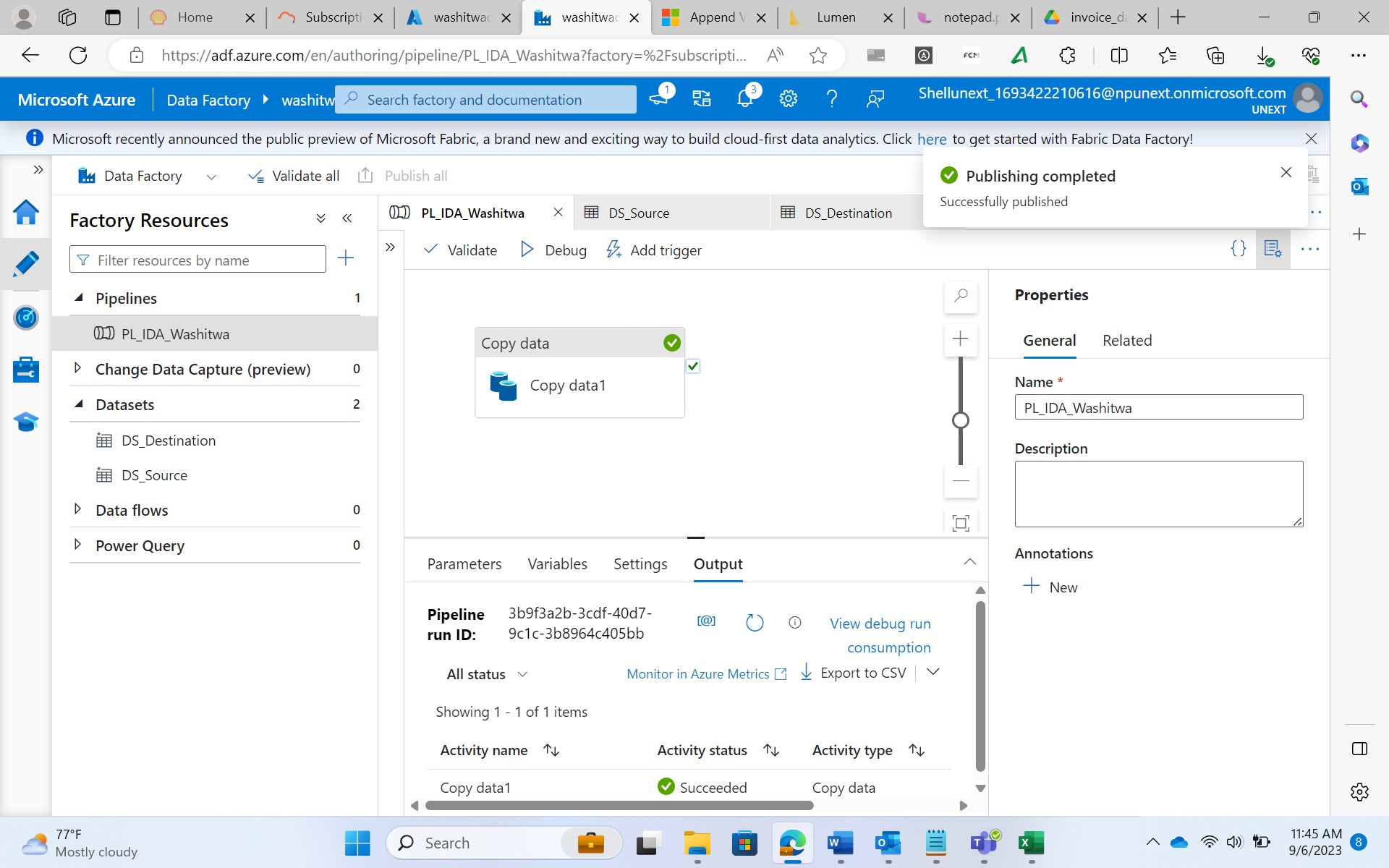
* + Here parameters can be added for dynamic execution
  + For the container name, sub folder name, and the file name, parameters can be added to perform activities dynamically



* Create a Pipeline



* + Here the same number of parameters need to be created as the number of total parameters for the datasets collectively
  + Mapping of the parameter names should also be done
* Debug and Publish results

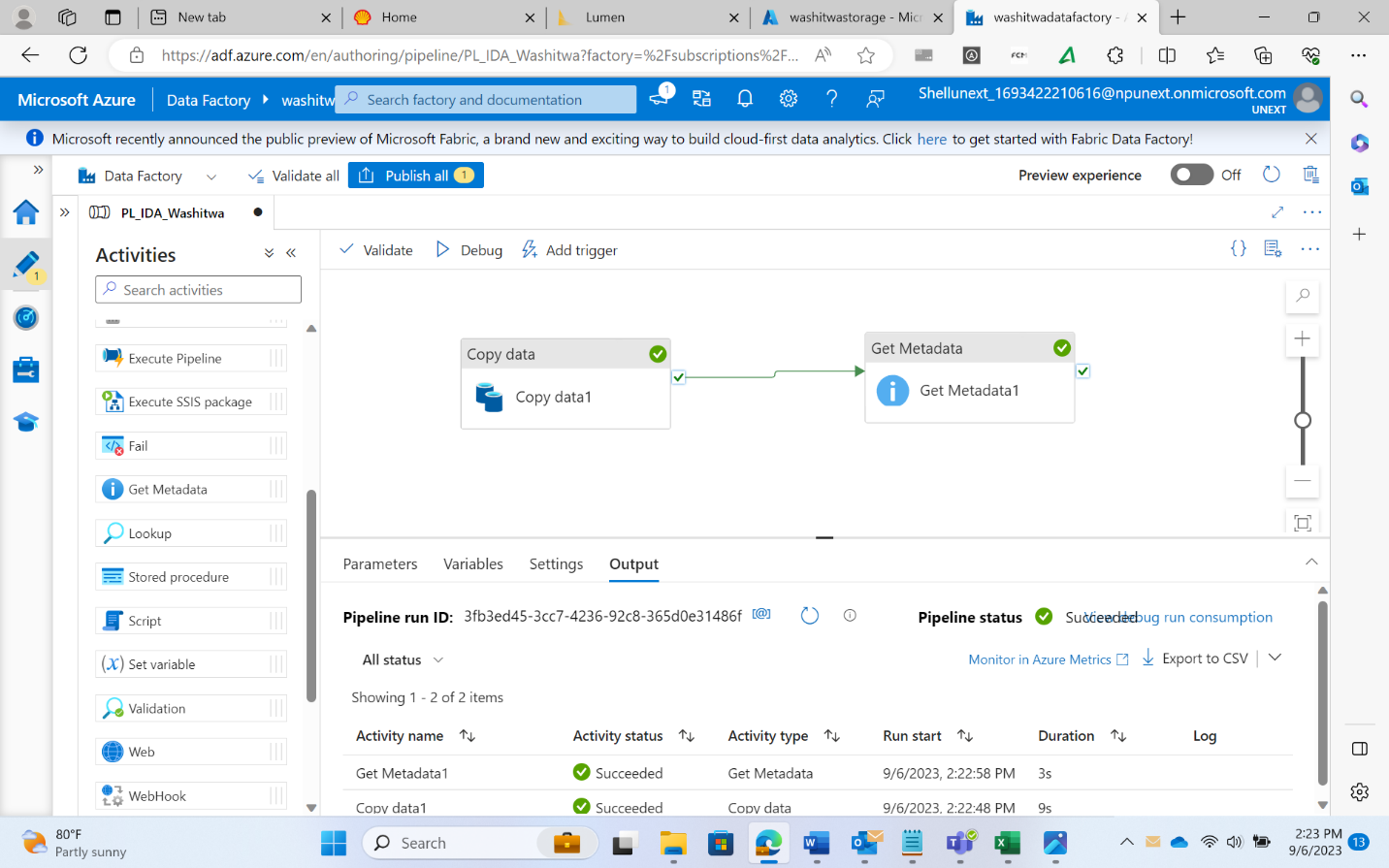


* + Upon clicking the debug option, the container name, sub folder name, file name need to be added for the source and destination
  + Once the execution is successful, the results need to be published

Triggers in ADF

* A way to automate the pipeline executions.
* There exists a many to many relationship between the triggers and the pipelines
* Triggers determine the schedule for pipeline executions
* 3 types of triggers
  + Tumbling Window Triggers
    - Triggers work at periodic time intervals from a specified start time
    - They can be used to run past and future dated activities
    - Execution happens for every window which is a non-overlapping period of time
  + Scheduled Triggers
    - Runs pipelines on a clock schedule
    - It only runs future dated activities
  + Event-Based Triggers
    - These triggers cause the pipeline to be executed in response to an event
    - These events can be the arrival of a file (file being uploaded to a container), deletion of a file from the container, etc.

Get Metadata Activity



{

"size": 3035,

"itemName": "zipcodes.csv",

"itemType": "File",

"lastModified": "2023-09-06T05:53:30Z",

"effectiveIntegrationRuntime": "AutoResolveIntegrationRuntime (East US)",

"executionDuration": 0,

"durationInQueue": {

"integrationRuntimeQueue": 0

},

"billingReference": {

"activityType": "PipelineActivity",

"billableDuration": [

{

"meterType": "AzureIR",

"duration": 0.016666666666666666,

"unit": "Hours"

}

]

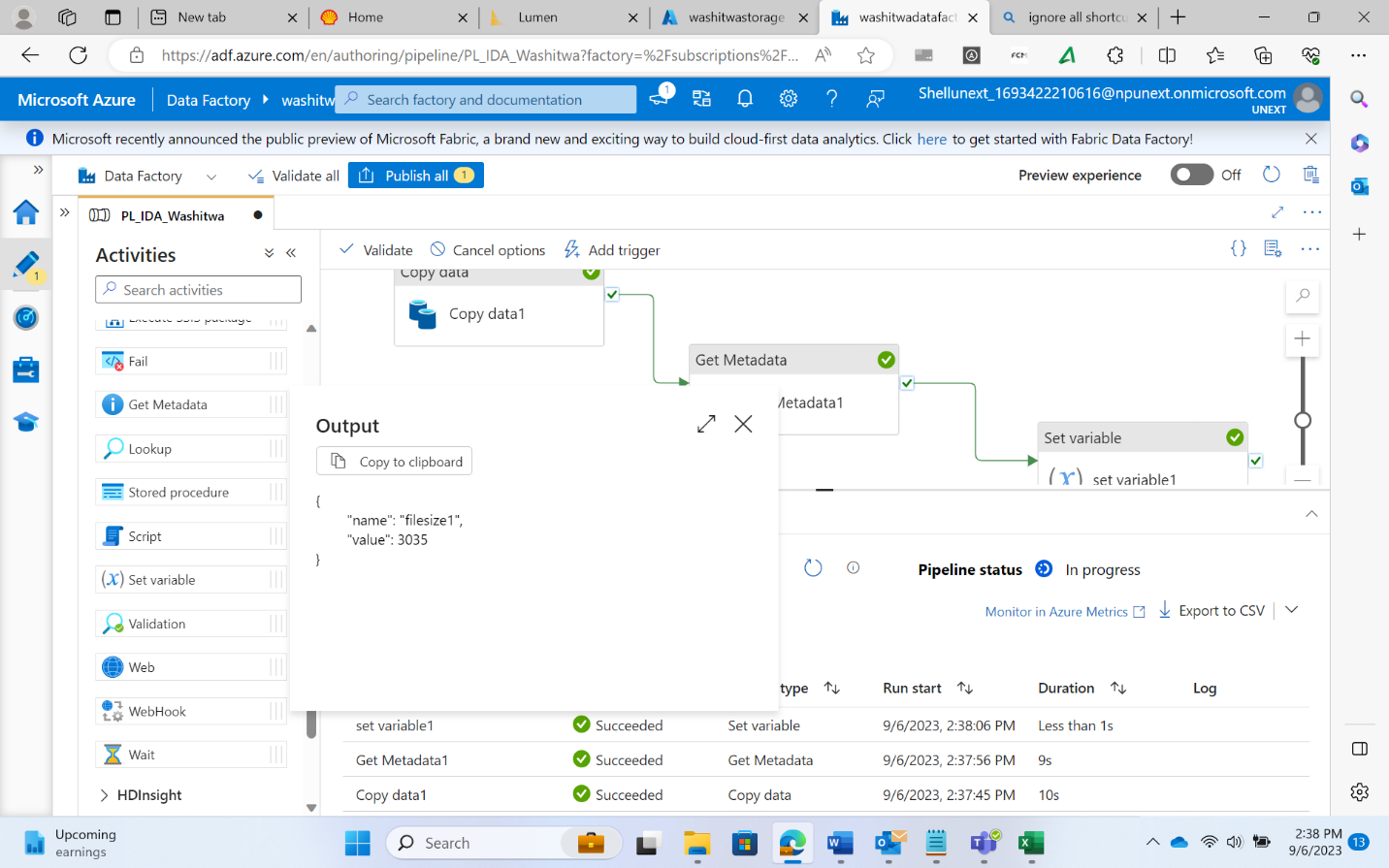
}

}

* The output for the size, file type, last modified etc. are present in the output as seen above

Set Variable Activity

* A variable is created in the pipeline and using the Set Variable activity, the value of the file size is attached to the variable
* The output returned is as shown



{

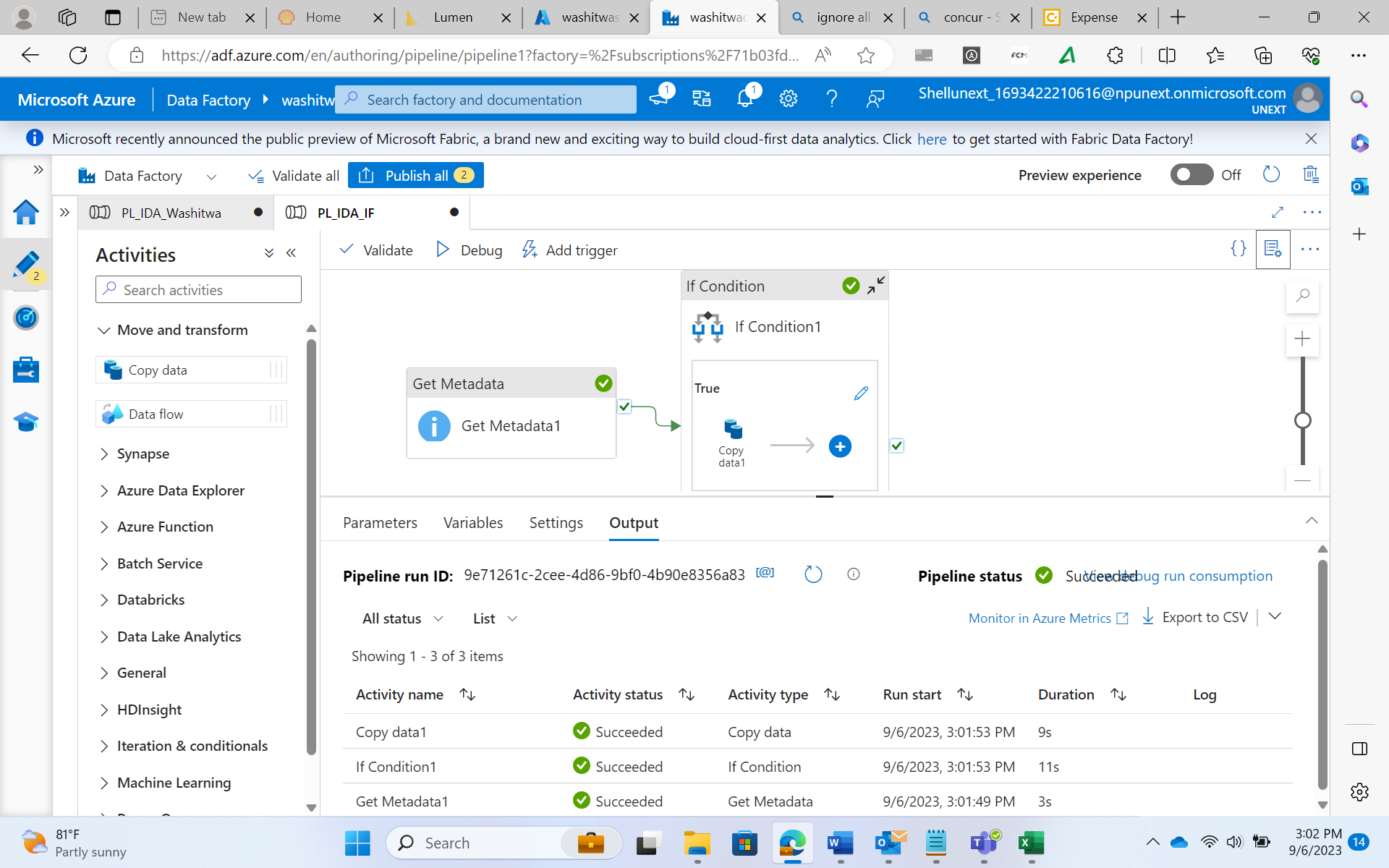
"name": "filesize1",

"value": 3035

}

Task

* Requirement
  + Create a data pipeline to get the size of a file and check if the file is greater than 10 KB
  + If the file is indeed greater than 10 KB, copy the file from the source to the destination
* Solution
  + Use of Get Metadata activity to get the file size
  + Then upon success use the If Condition under Iteration & Conditionals
  + In the If Condition, add the copy activity
  + Give the expression (condition check) by using logical operators and activity outputs
  + Finally debug and publish



For Each Activity

* The drawback of If Condition activity is that operations for only one file can be added for the if condition expression
* To deal with this, there exists the For Each Activity that can carry out operations for multiple files