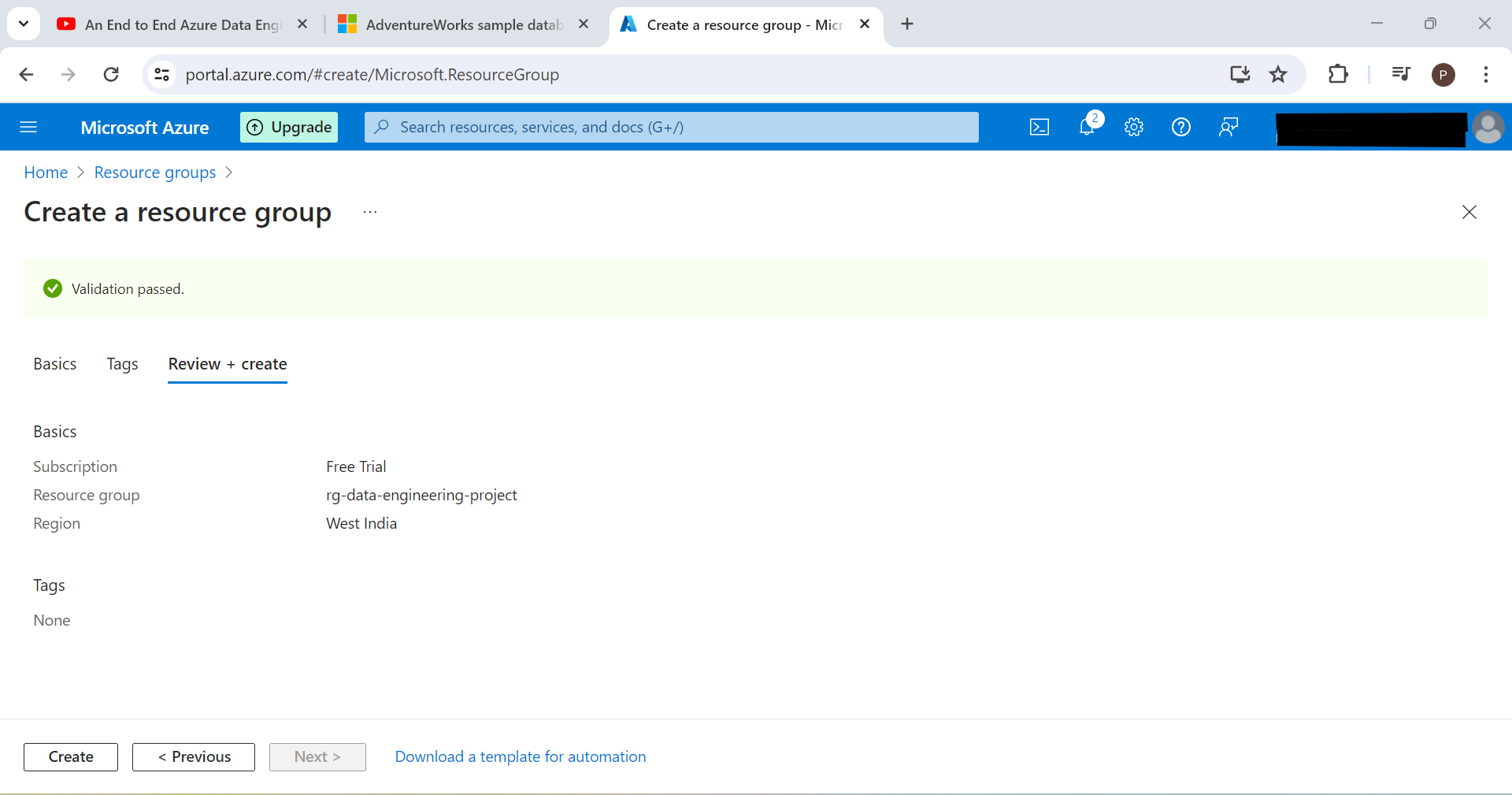
# **End to End Azure Data Engineering Real Time Project**

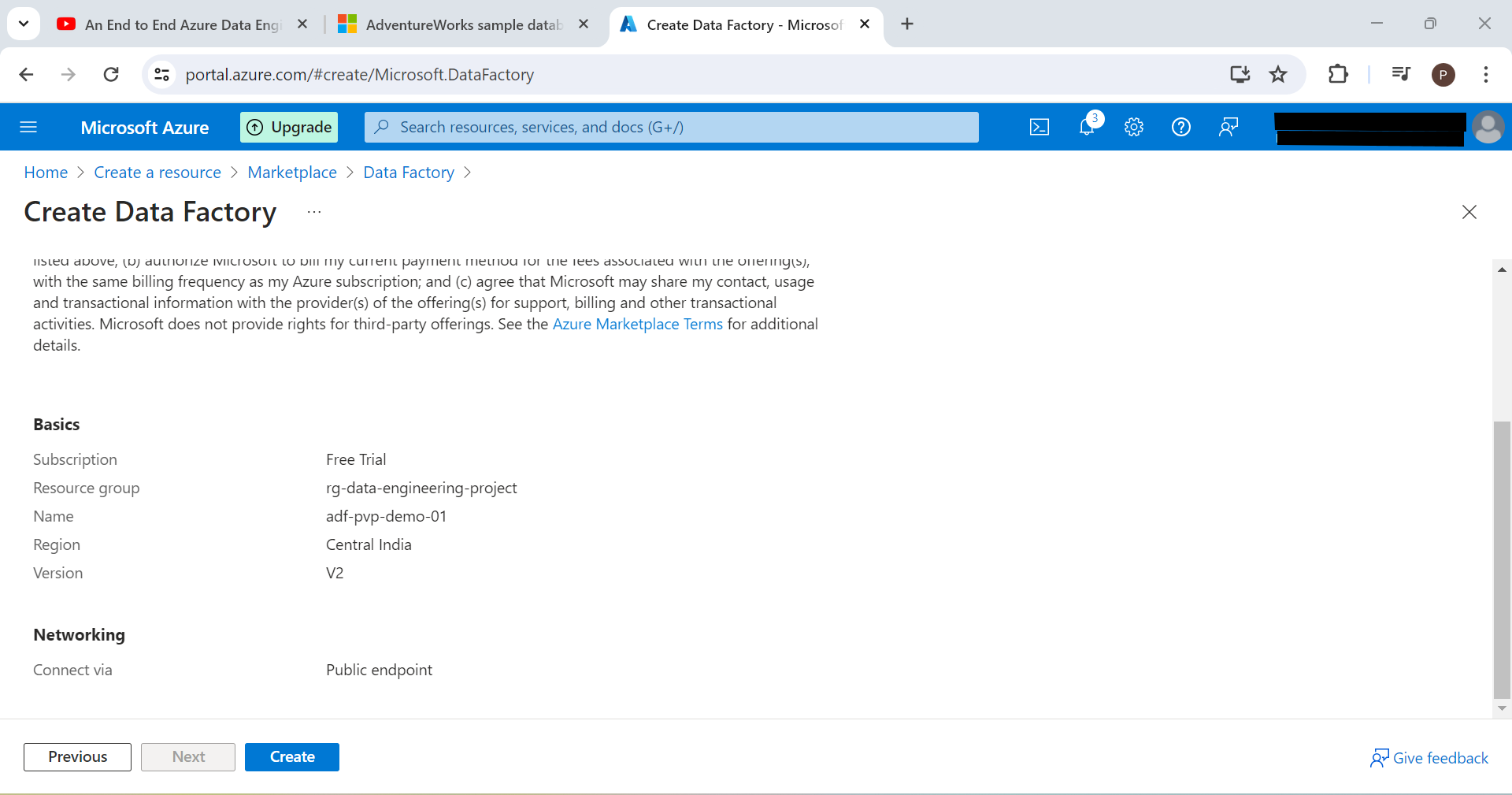
1. Installed latest SQL Server Developer Edition
2. Installed latest SSMS
3. Get source database AdventureWorksLT2017.bak from

<https://learn.microsoft.com/en-us/sql/samples/adventureworks-install-configure?view=sql-server-ver16&tabs=ssms>

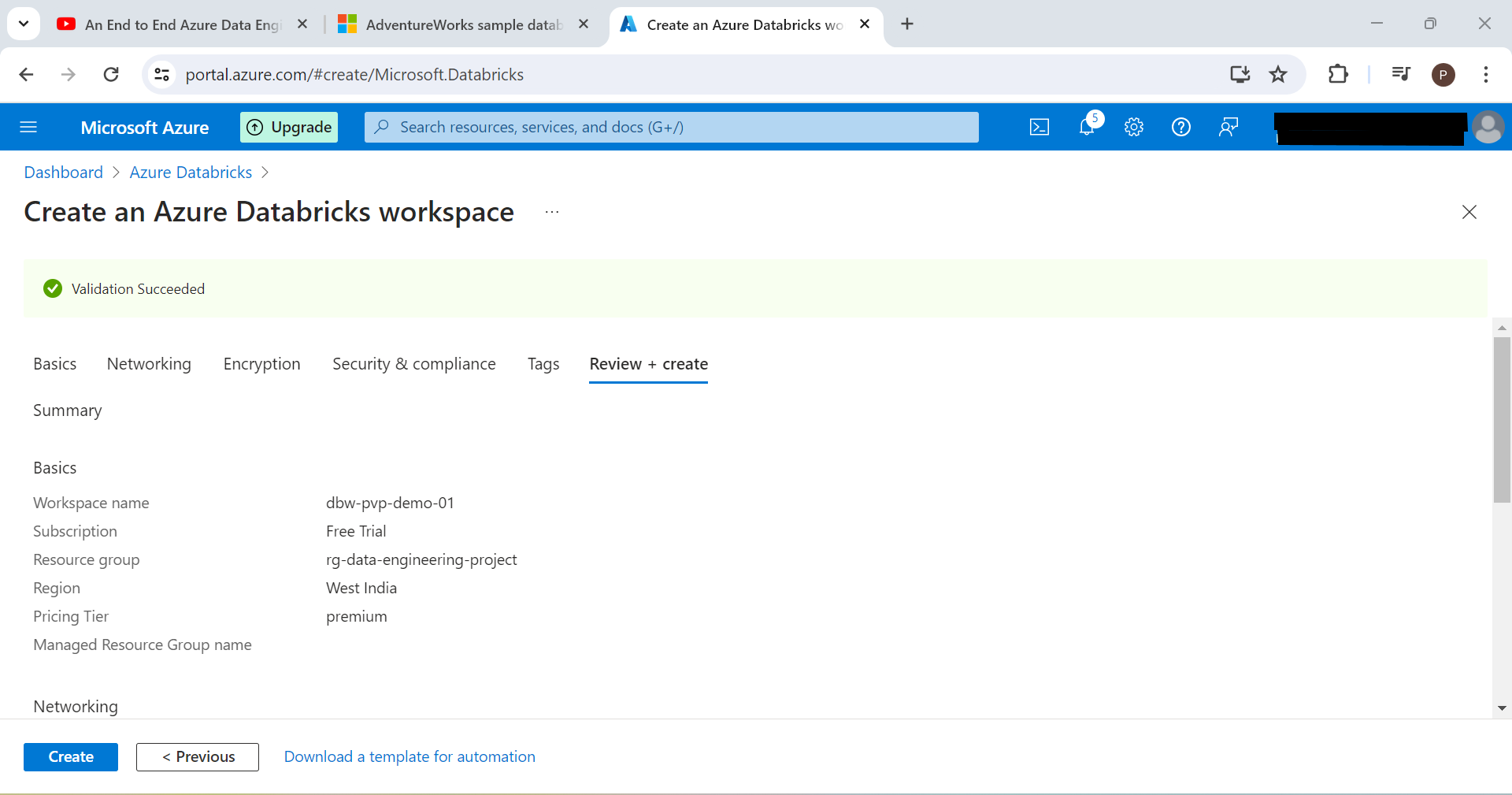
1. Login into <https://portal.azure.com>
2. Create resource group



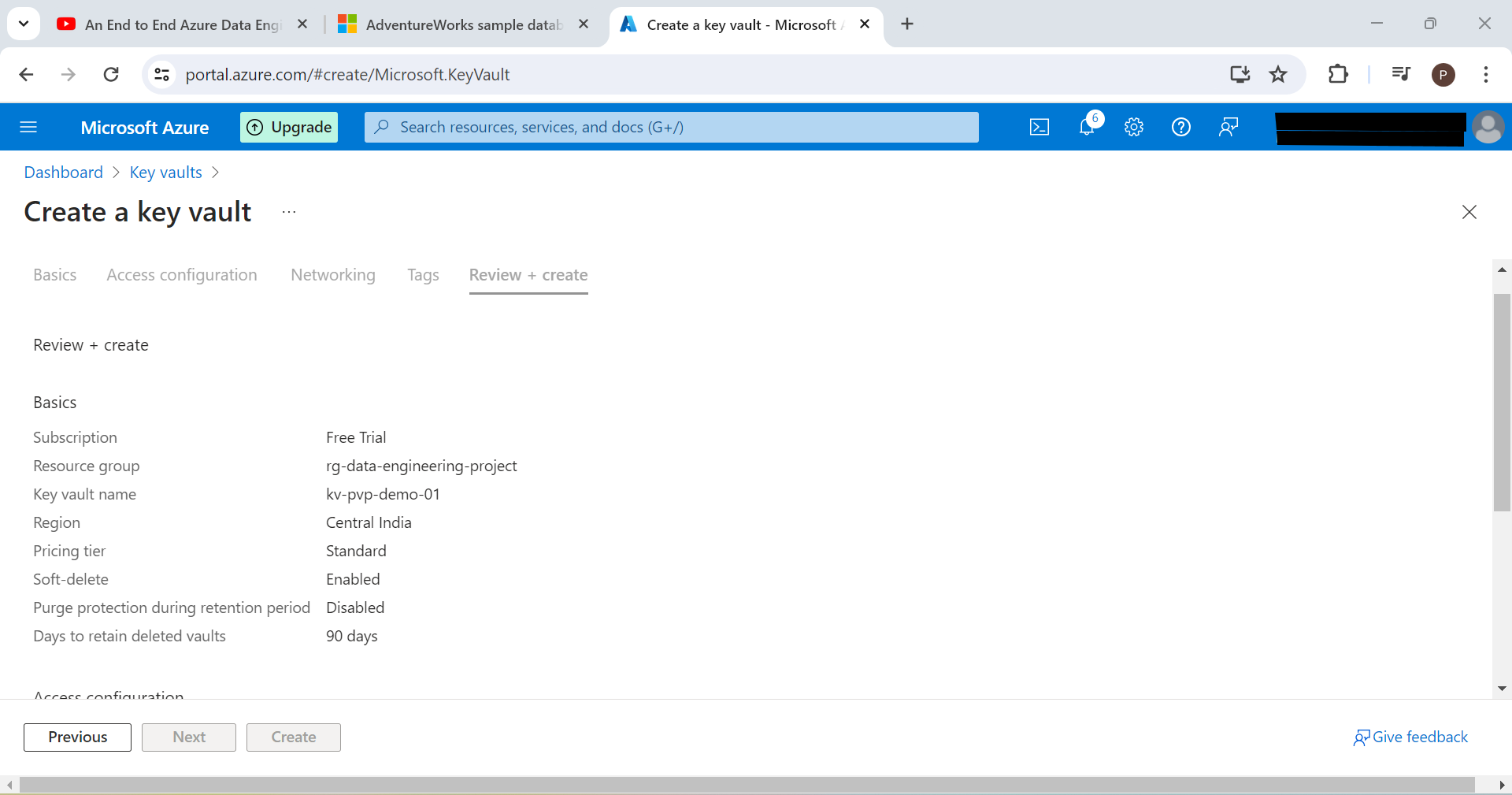
1. Create Azure Data Factory



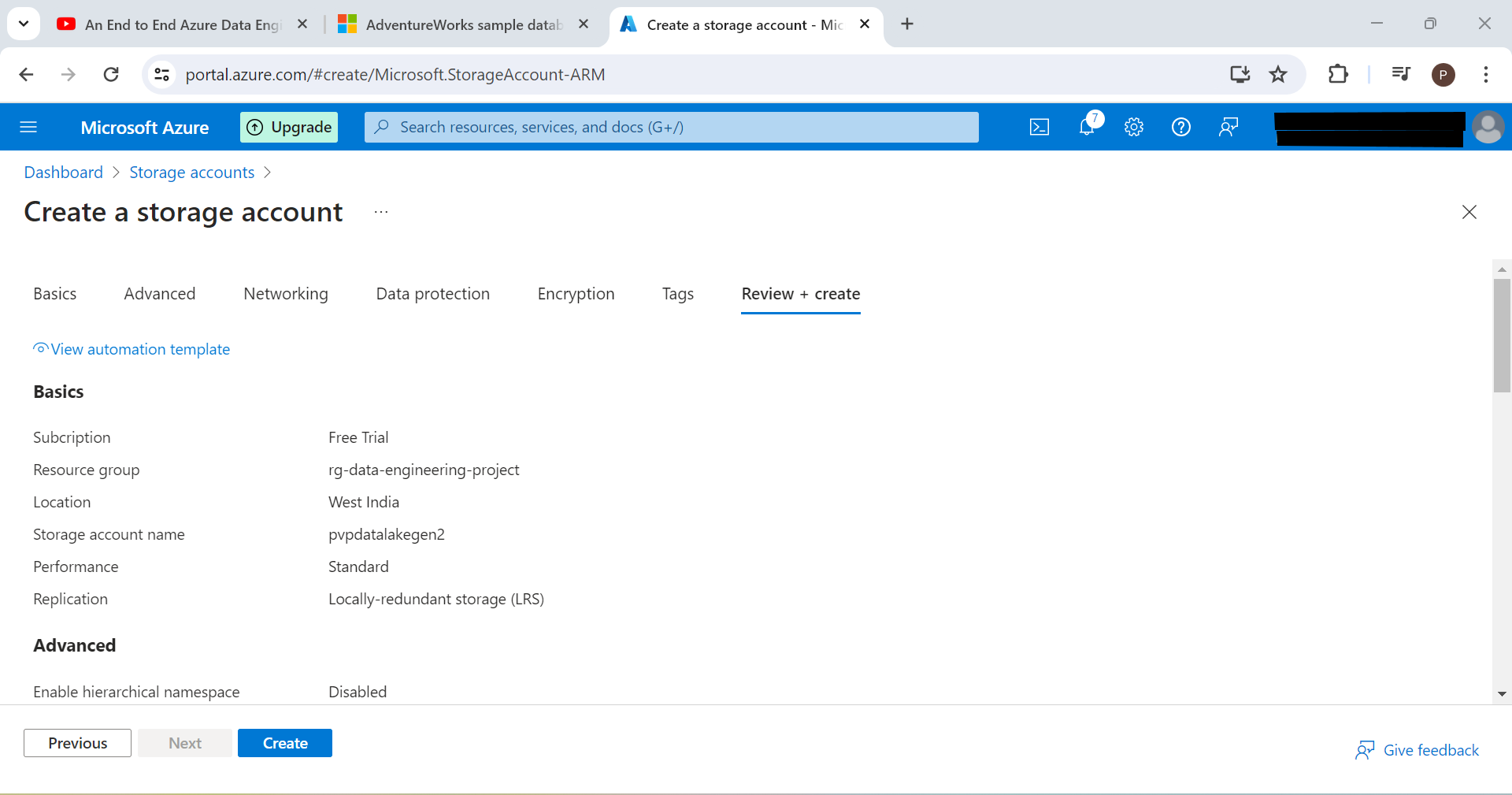
1. Create Azure Databricks Workspace



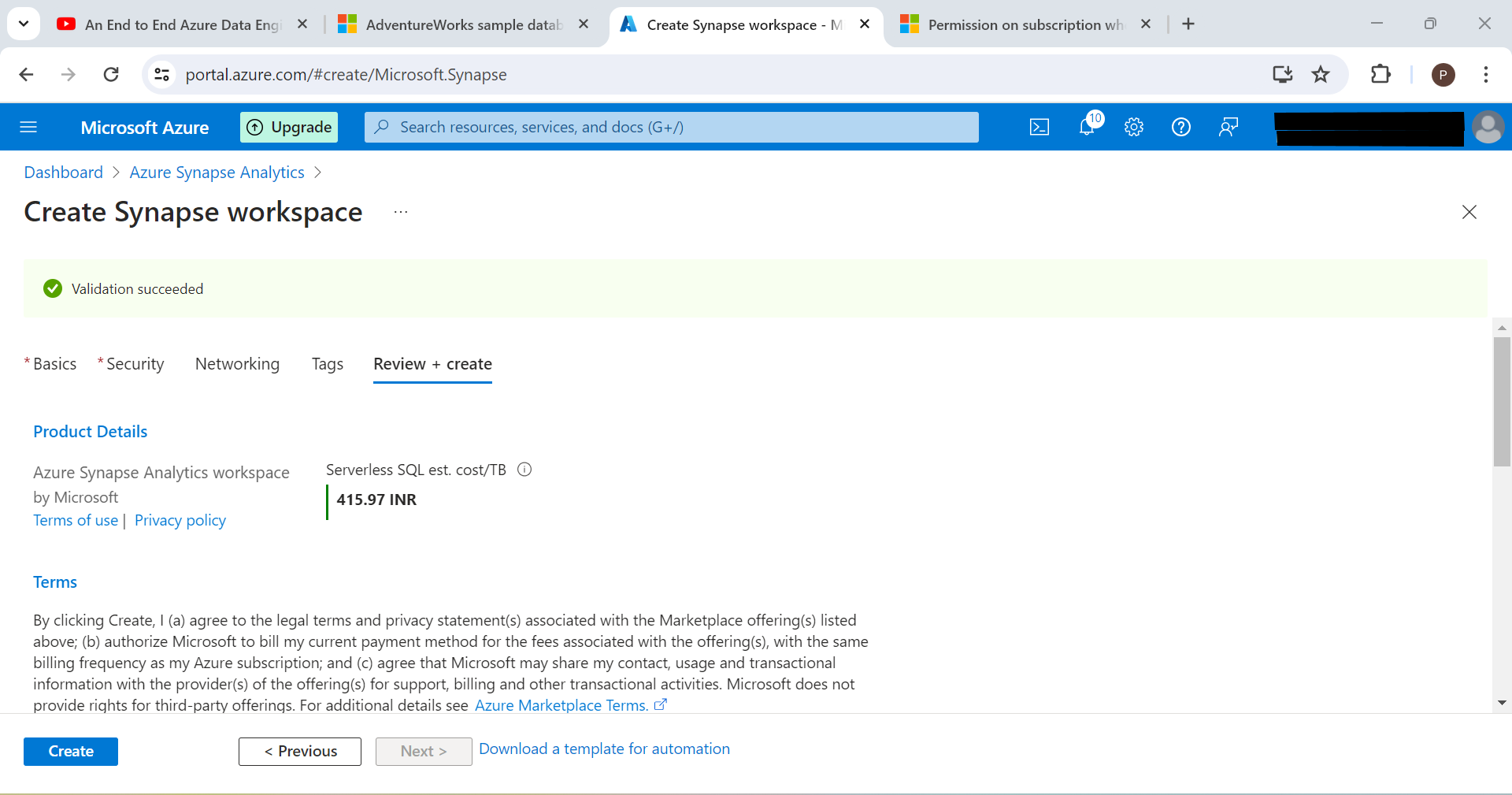
1. Create Azure key vault



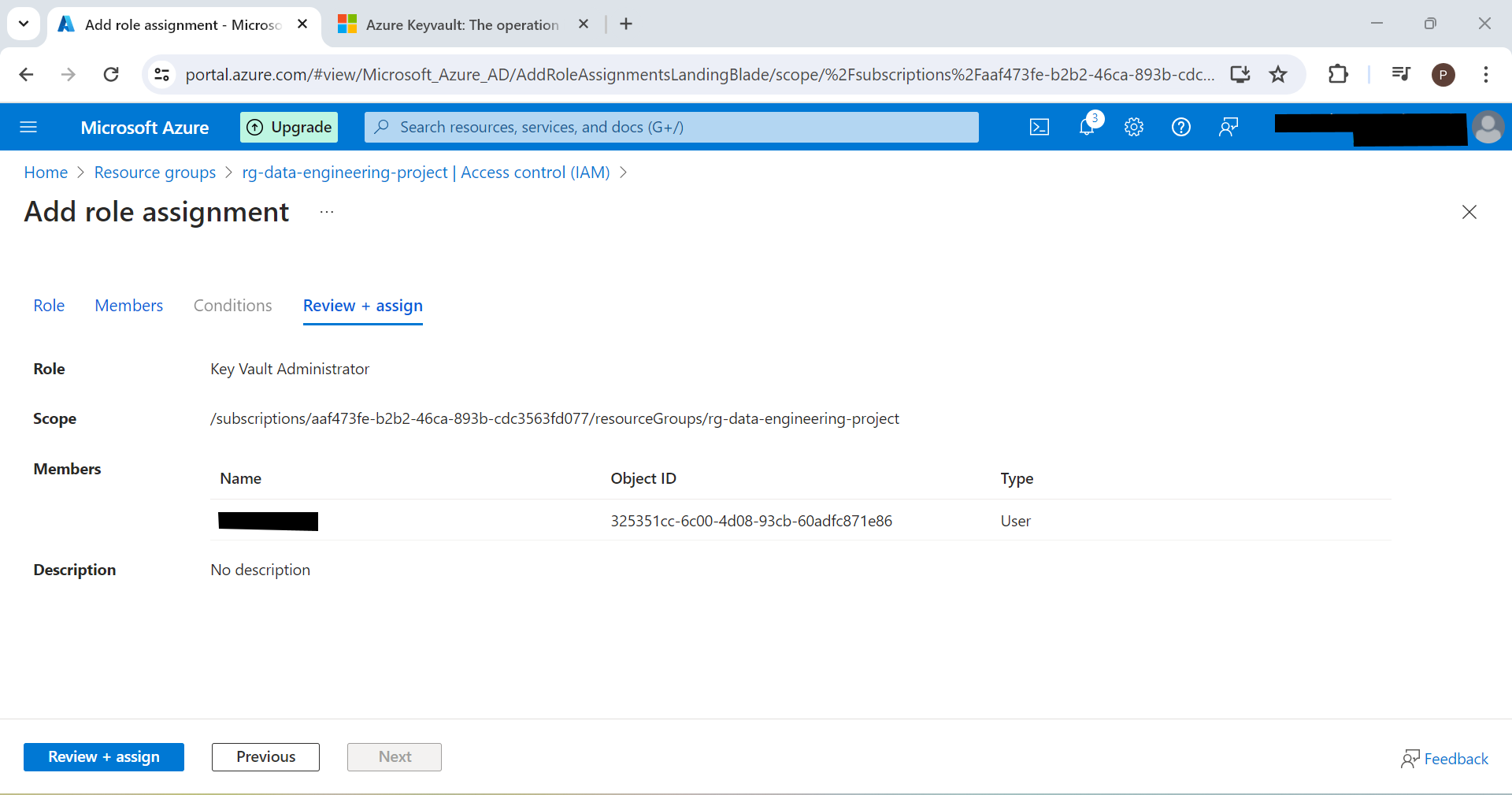
1. Create storage account



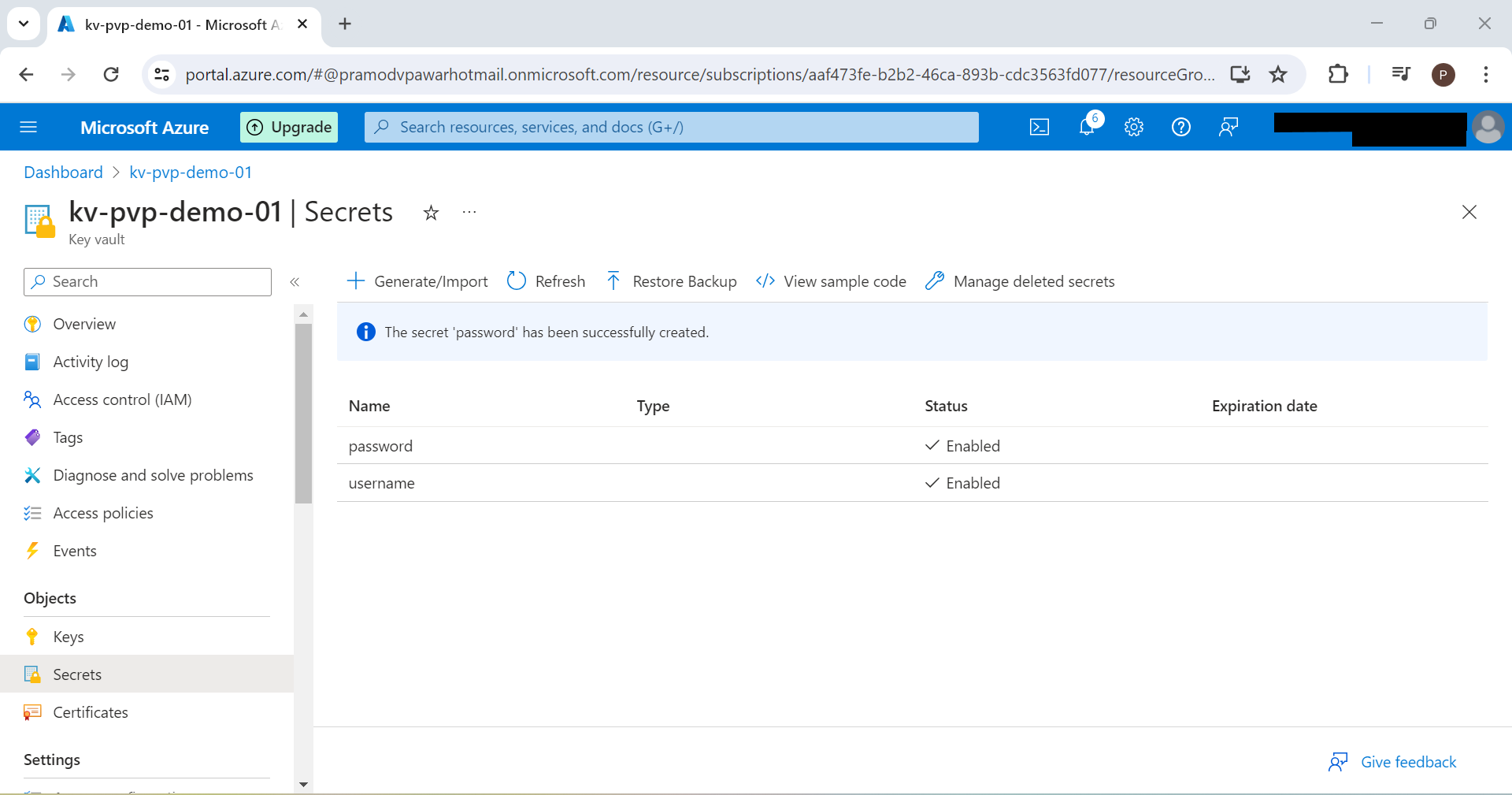
1. Create Synapse Analytics Workspace



1. Create new login user in local SQL Server database. This will be used in Azure Data Factory to connect to this database to extract data.
   1. use [AdventureWorksLT2017]
   2. CREATE LOGIN <user> WITH PASSWORD = <password>
   3. CREATE USER <user> FOR LOGIN <user>
   4. Assign db\_datareader role to new user: Goto [AdventureWorksLT2017] 🡪 Security 🡪 Users 🡪 Right Click New User 🡪 Owner Schema 🡪 db\_datareader
   5. Sometimes new user may not be able to access database. We need to modify authentication type as both “Windows and SQL Server” and restart SQL Server services from Services in control panel.
2. Add role based access control (in order to create Azure key vault secrets)

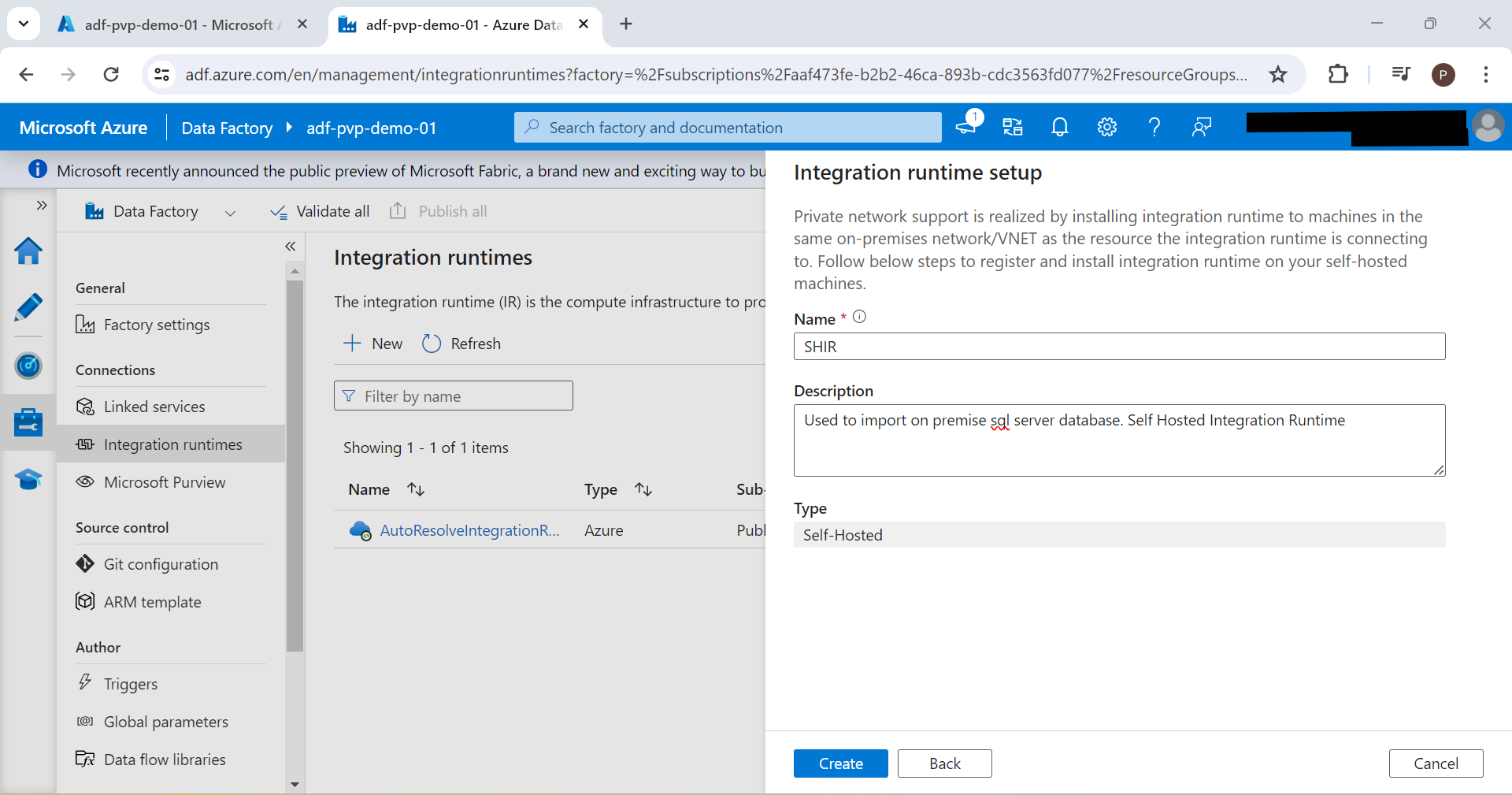


1. Goto Azure portal and open key vault kv-pvp-demo-01. Click Secrets and create two secrets for username and password

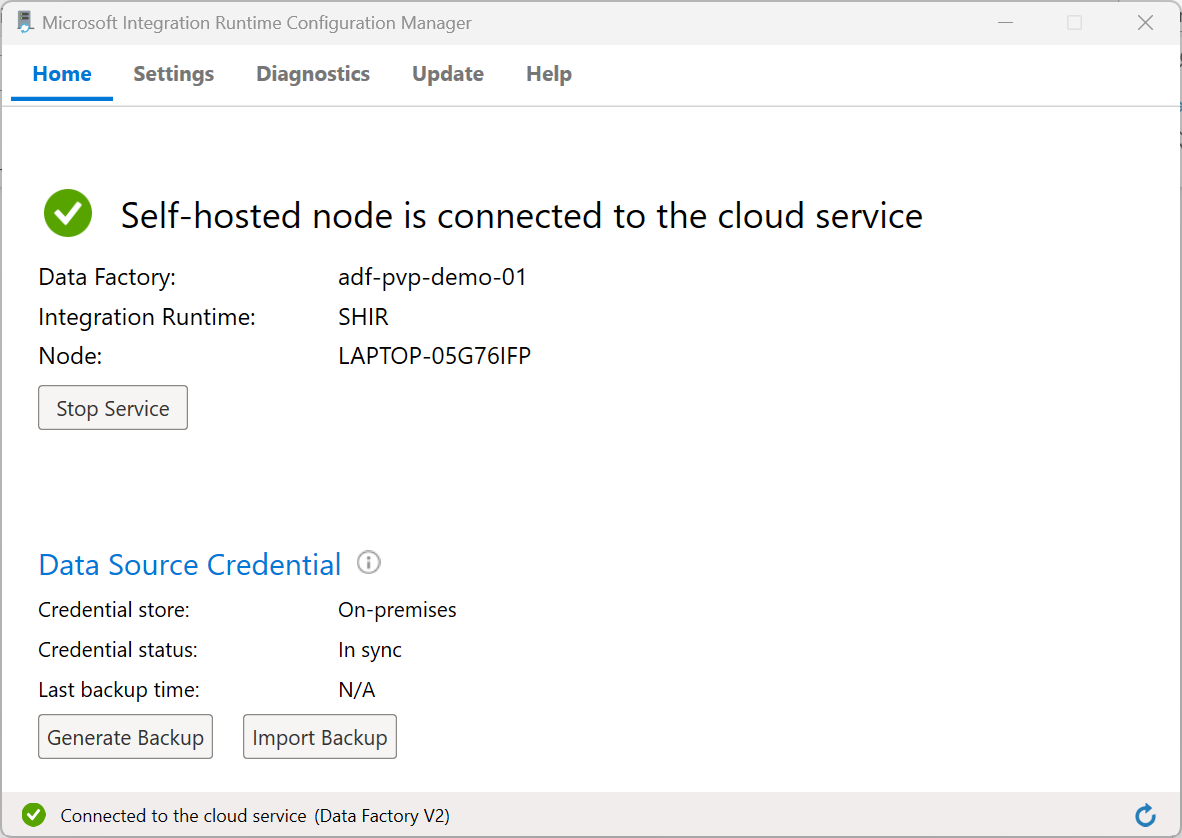


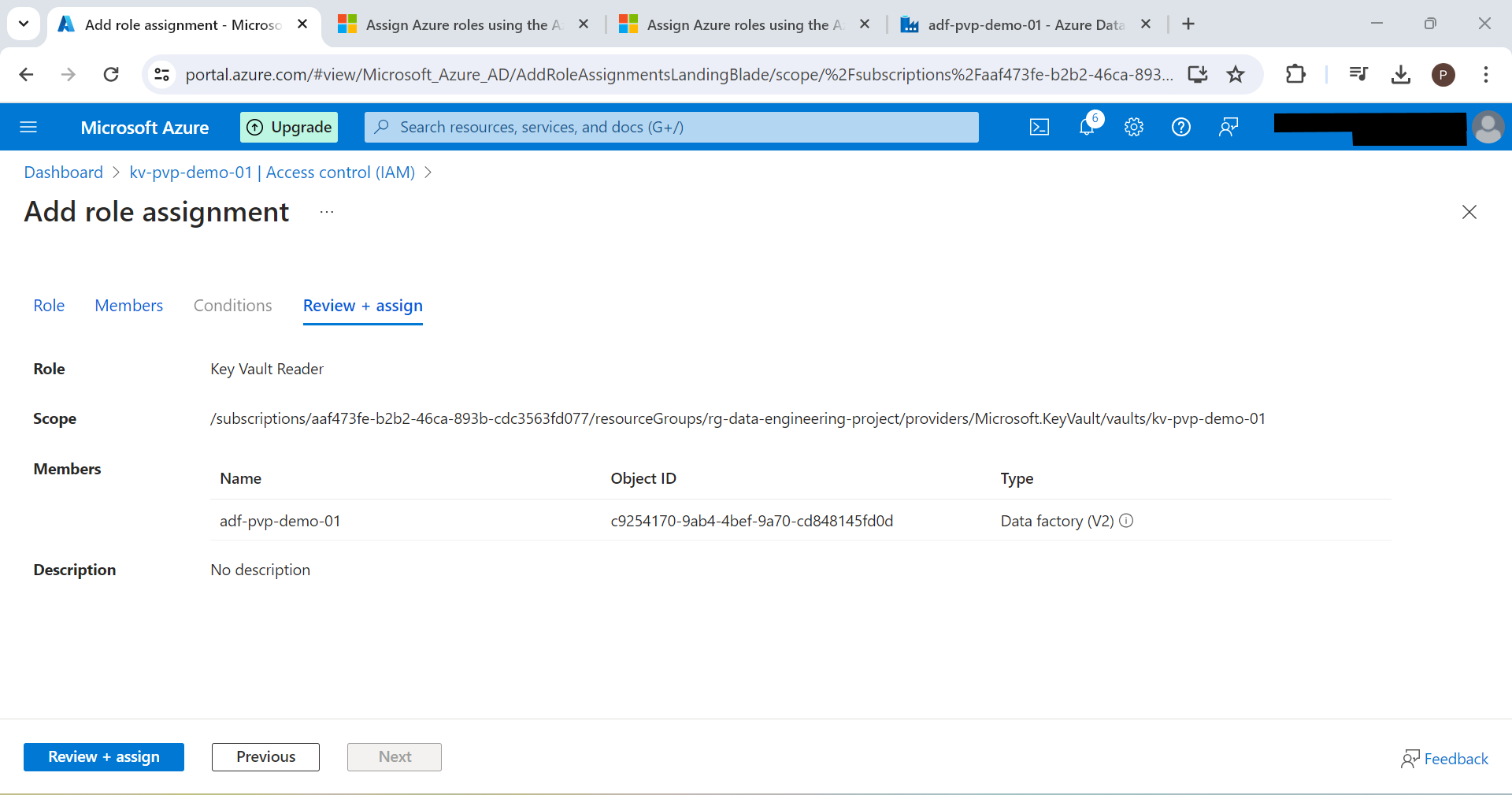
## Data Ingestion using ADF

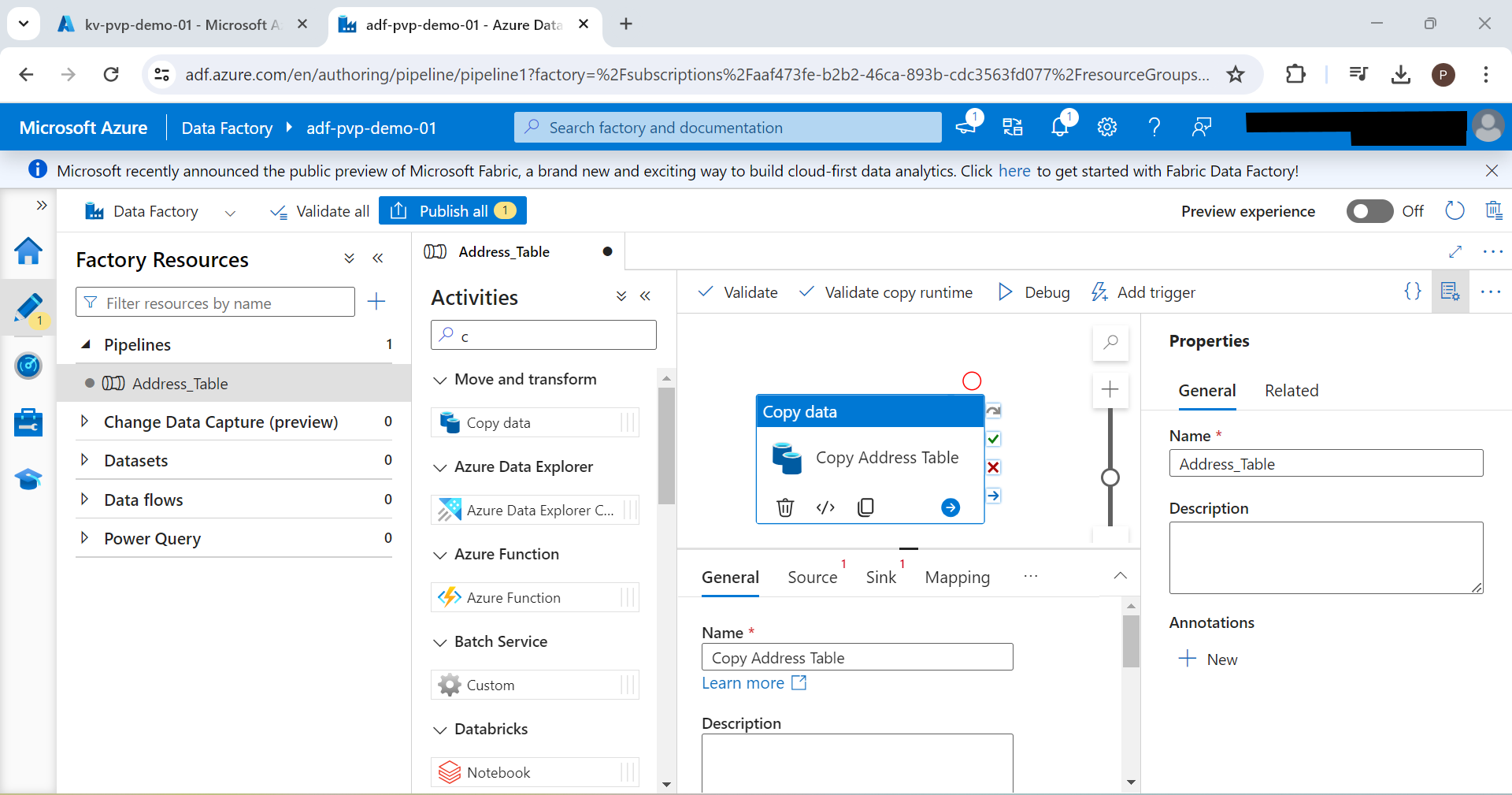
1. Goto Azure Data Factory adf-pvp-demo-01 and click on Launch Studio
2. Since there is no connection between on premises database and azure data factory; we will need to install self hosted integration runtime on our local machine. For this, goto Manage 🡪 Integration Runtime 🡪 New 🡪 Azure Self Hosted 🡪 Self Hosted 🡪 Continue 🡪 Give Name as SHIR and Description as “Used to import on premise sql server database. Self Hosted Integration Runtime”



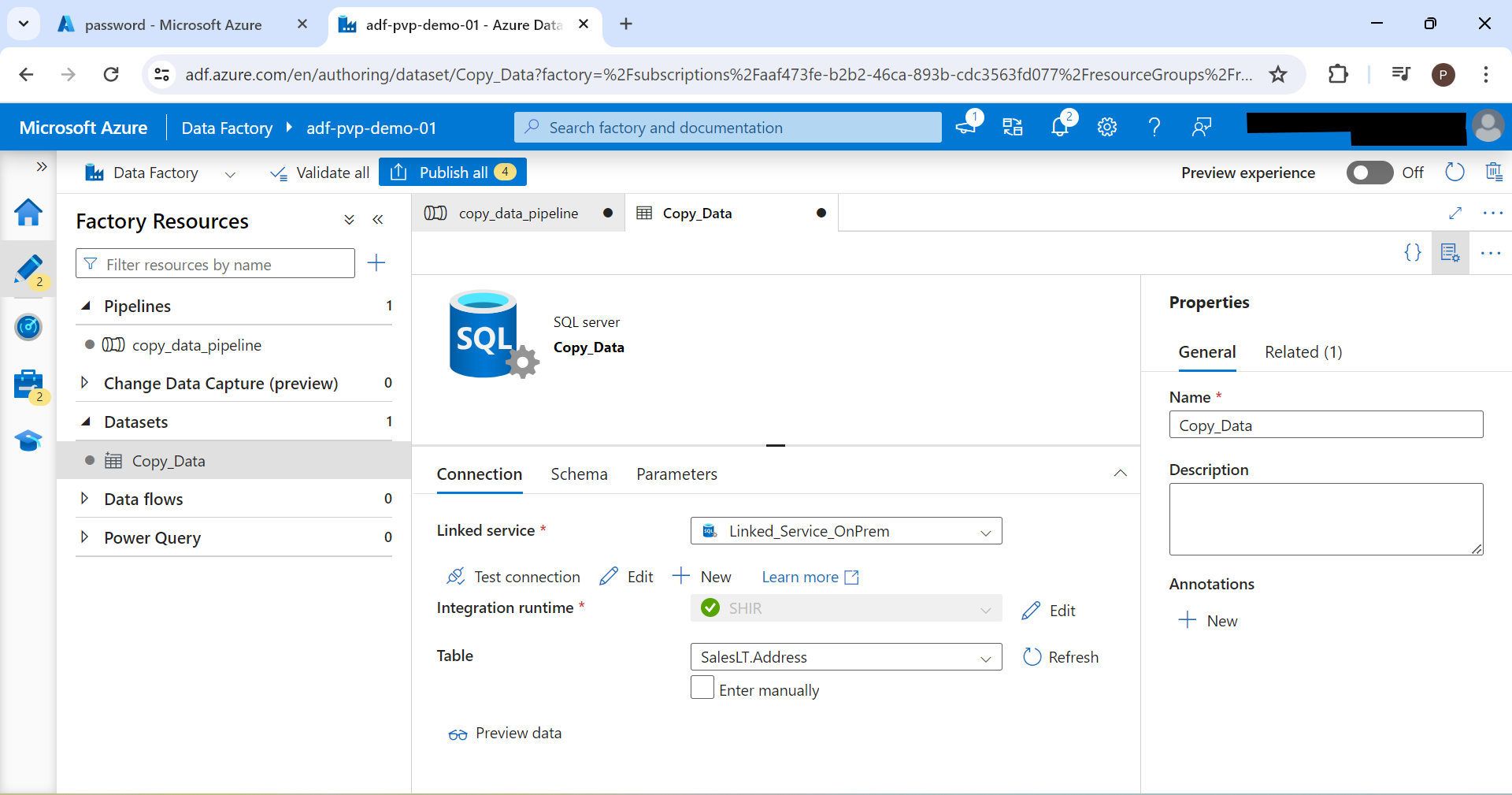
1. Once created; choose option 1 to install express setup (of integration runtime) on local machine. We can use manual setup with authentication keys when we are going to install integration runtime on another machine.
2. To check if Microsoft Integration Runtime is installed and connected to azure data factory; open Microsoft Integration Runtime from local machine



1. We will try to load sample table from on premise database to azure data factory. In order to establish connection between local database and azure data factory; we will use azure key vault. Before using Azure key vault; we need to grant privileges of it to Azure data factory.
   1. Goto Azure key vault 🡪 Access Control (IAM) 🡪 Grant access to this resource 🡪 Add role assignment 🡪 Search Role as Key Vault Reader 🡪 Search member as Managed Identity and search adf (adf-pvp-demo-01)
   2. 
   3. You will get message as “adf-pvp-demo-01 was added as Key Vault Reader for kv-pvp-demo-01.”
2. Go back to Azure Data Factory. Click New 🡪 Pipeline 🡪 Name as Address\_Pipeline. Search activity as Copy Data and drag it. Give its name as Copy Address Name



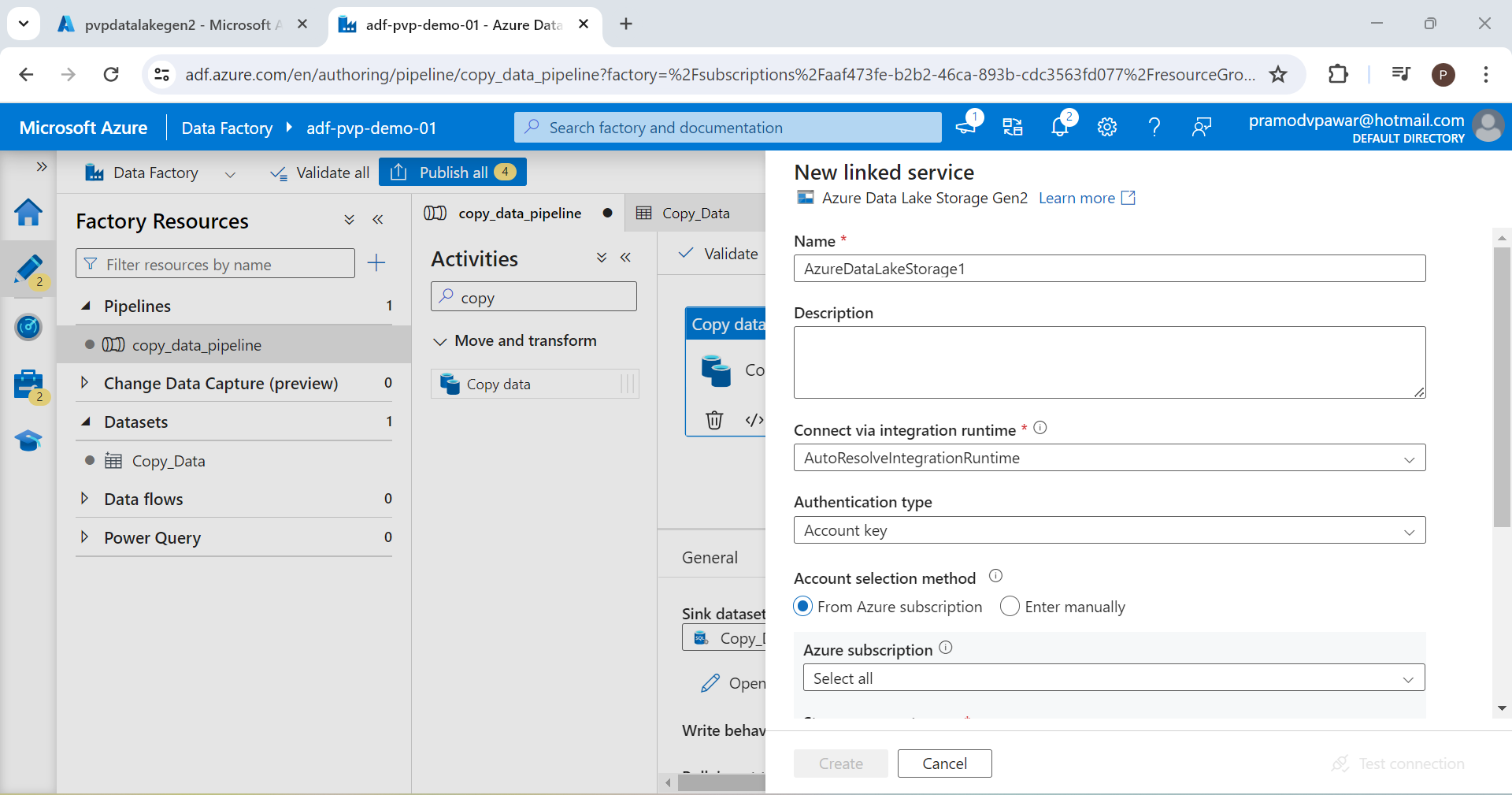
1. Setup source in data pipeline (using self hosted integration runtime – SHIR). Connect to local database



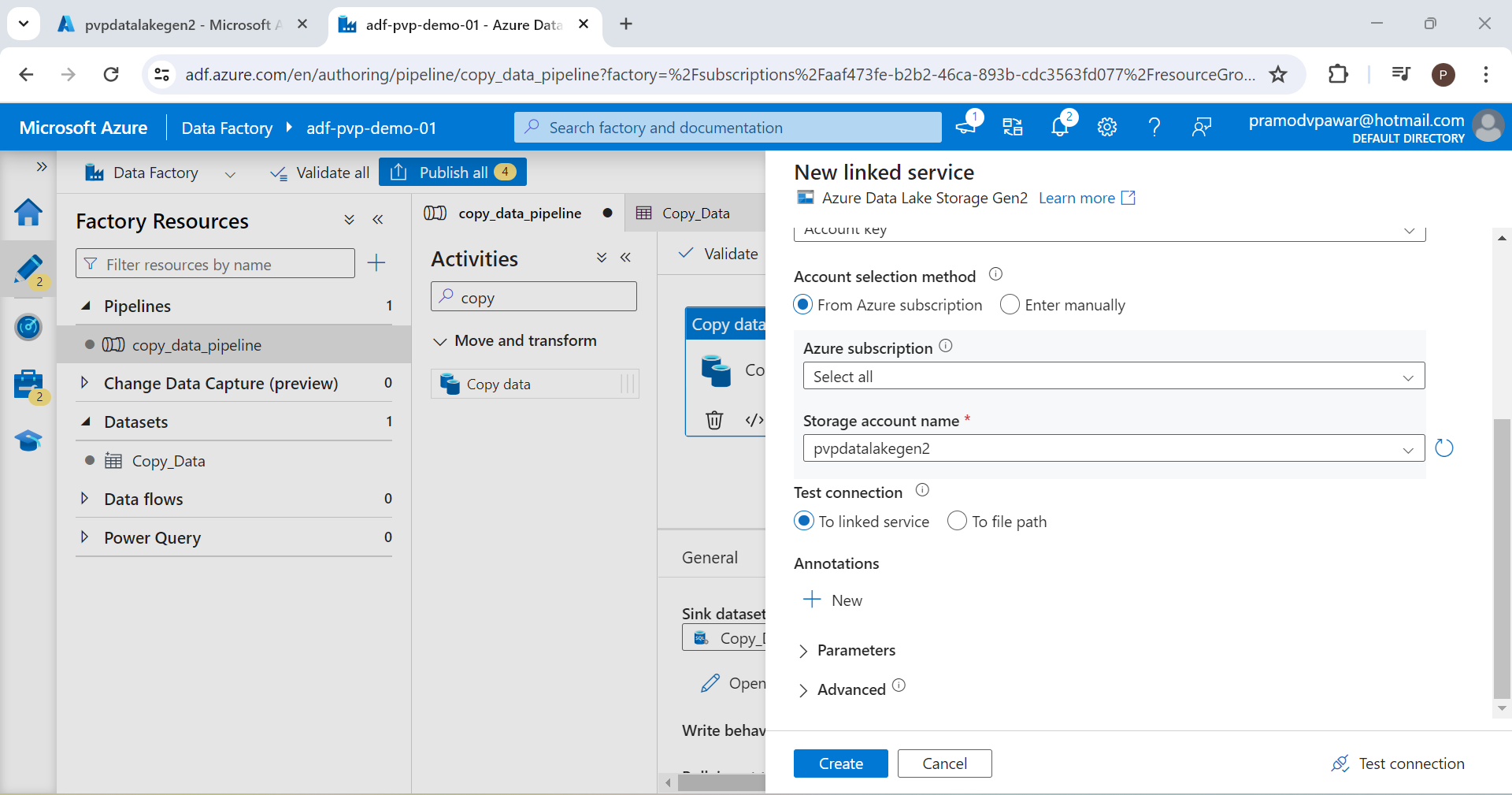
1. Setup Sink connection (to load data in Azure Data Lake Gen 2 – AutoResolveIntegrationRuntime). Before we setup sink connection; lets open storage account pvpdatalakegen2 that we created previously. We will go to containers and create new container as bronze



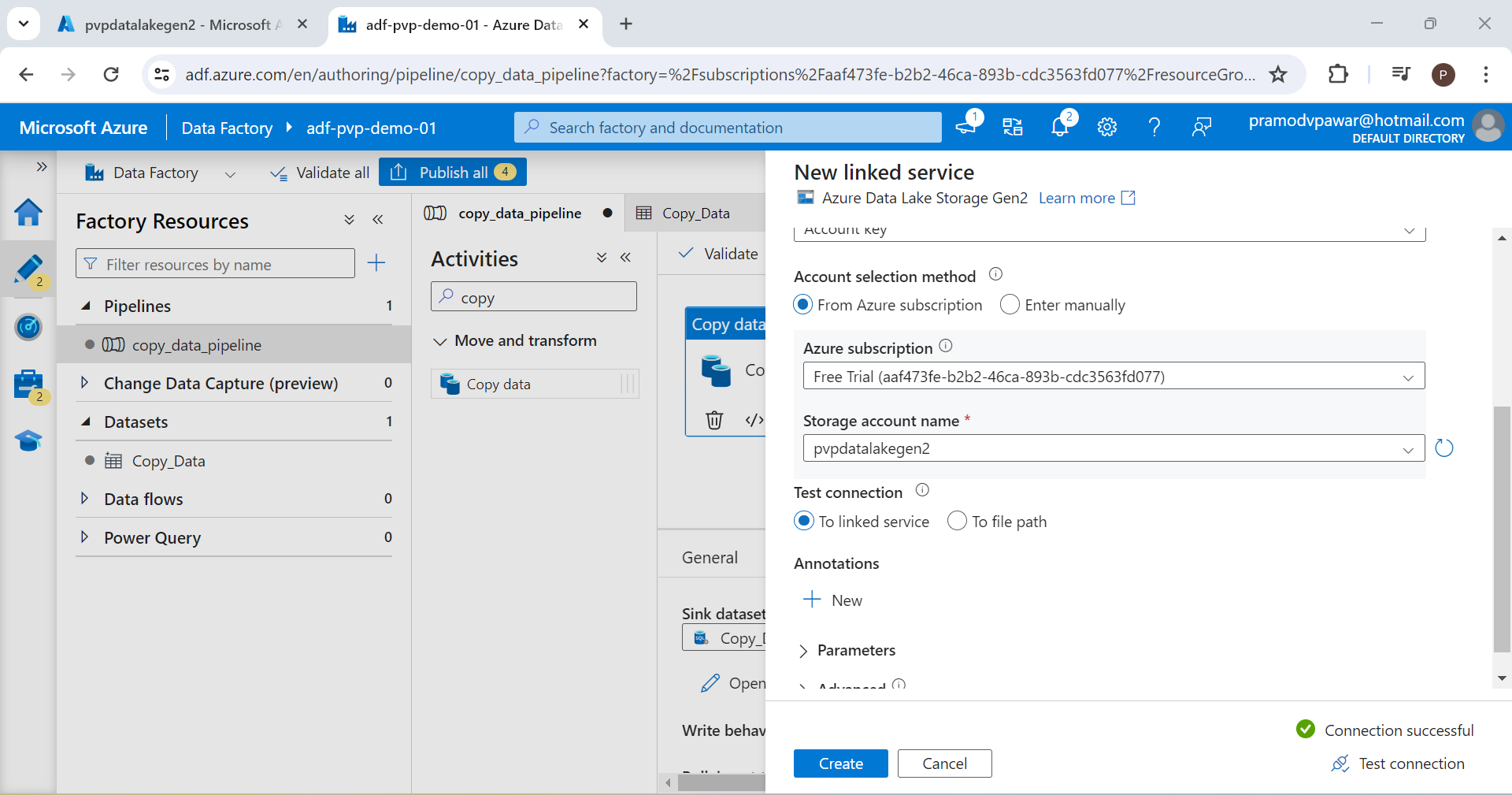
1. In order to sink dataset; click sink 🡪 click new 🡪 Select Azure Data Lake Storage Gen2 🡪 Parquet 🡪 Click on new linked service 🡪





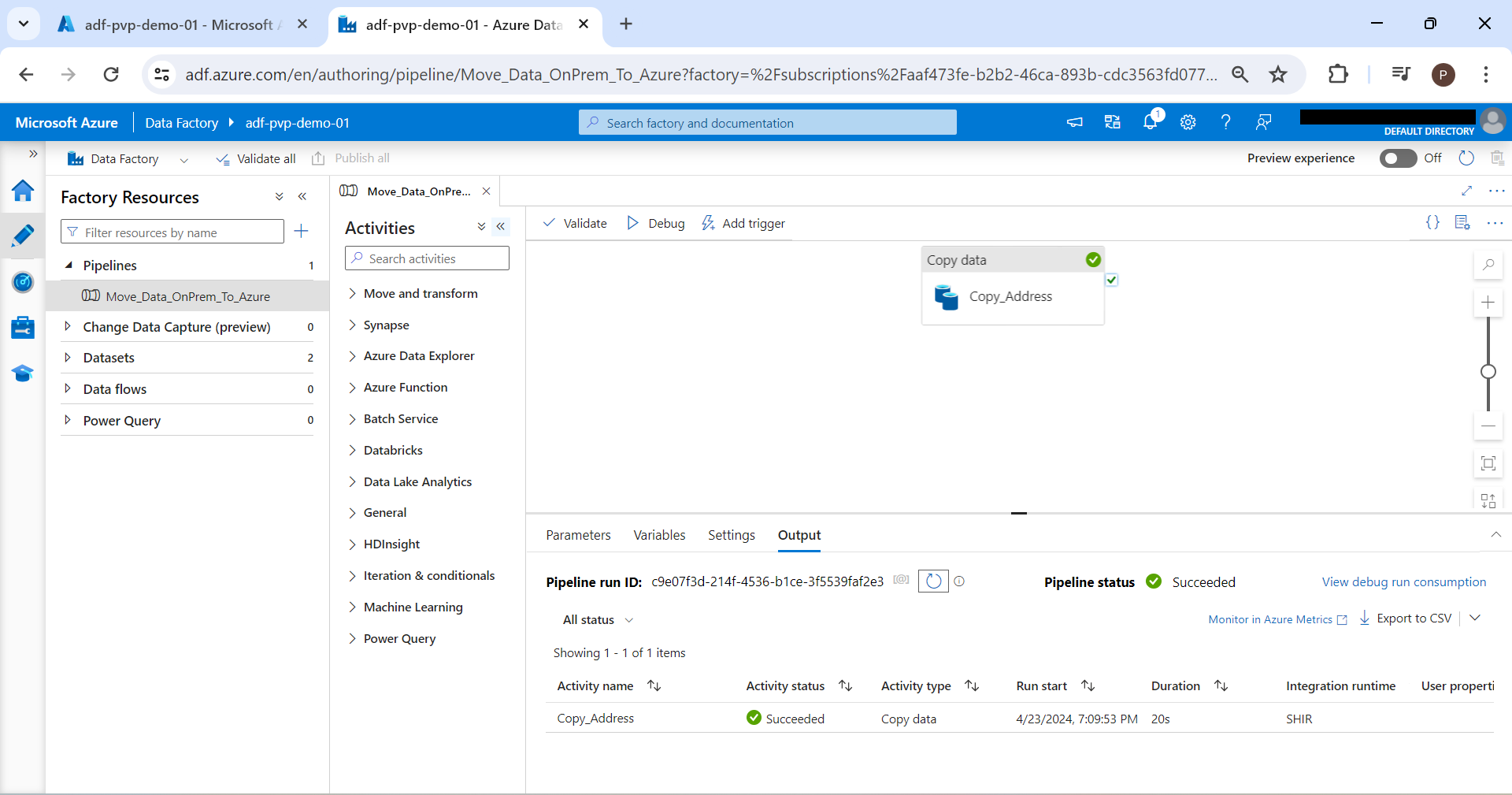




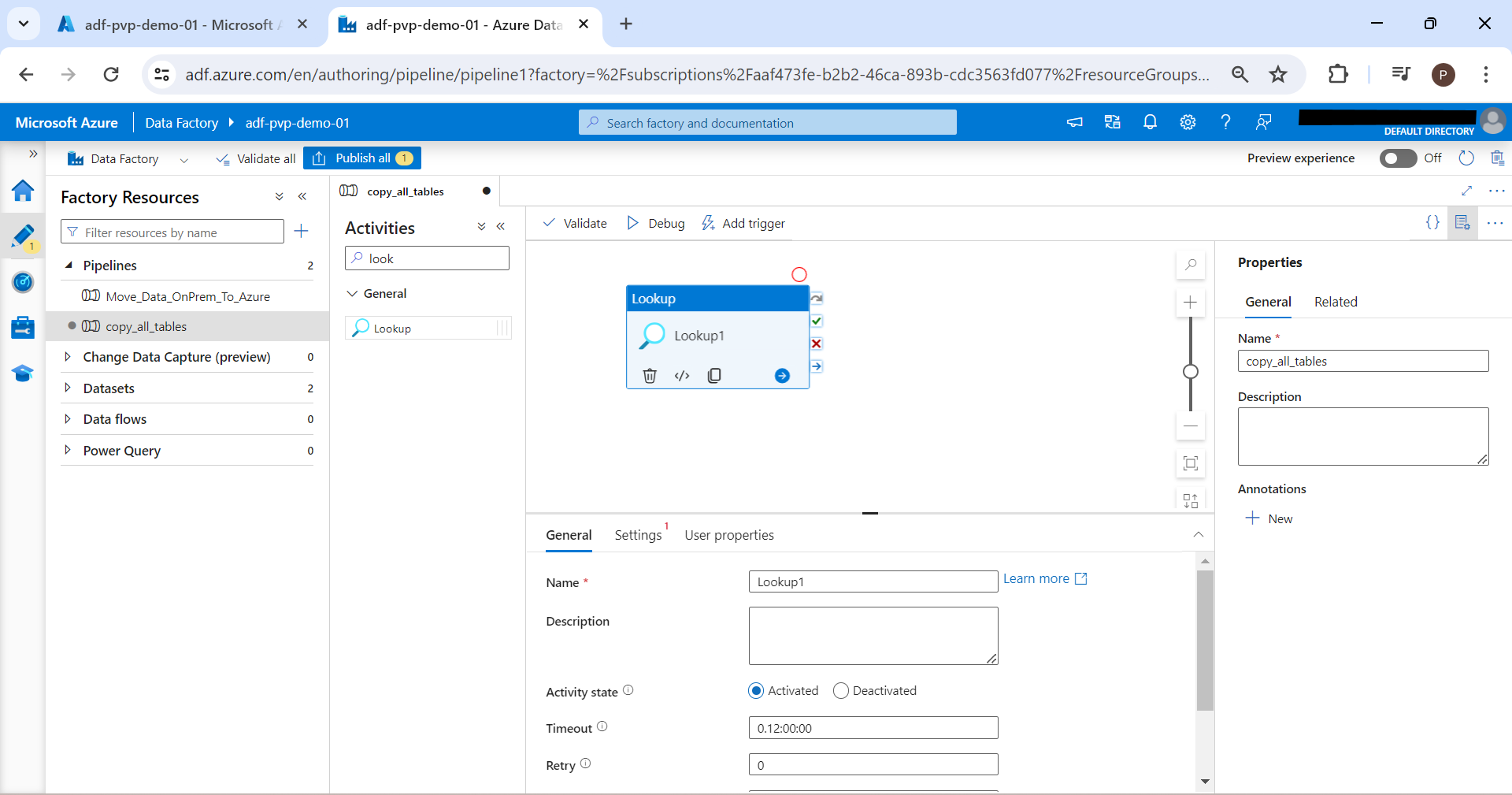




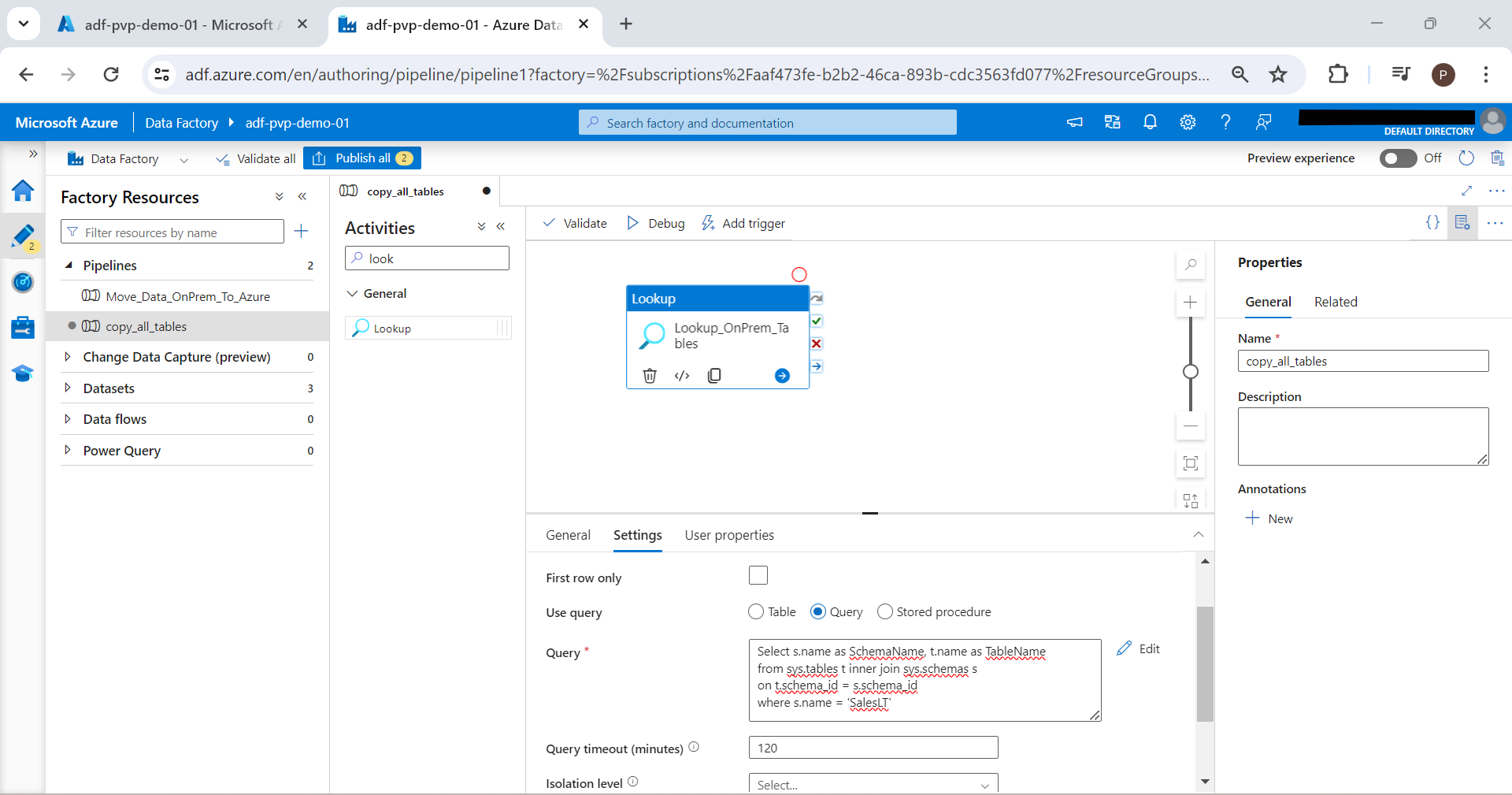
1. We can now run the pipeline using Debug option



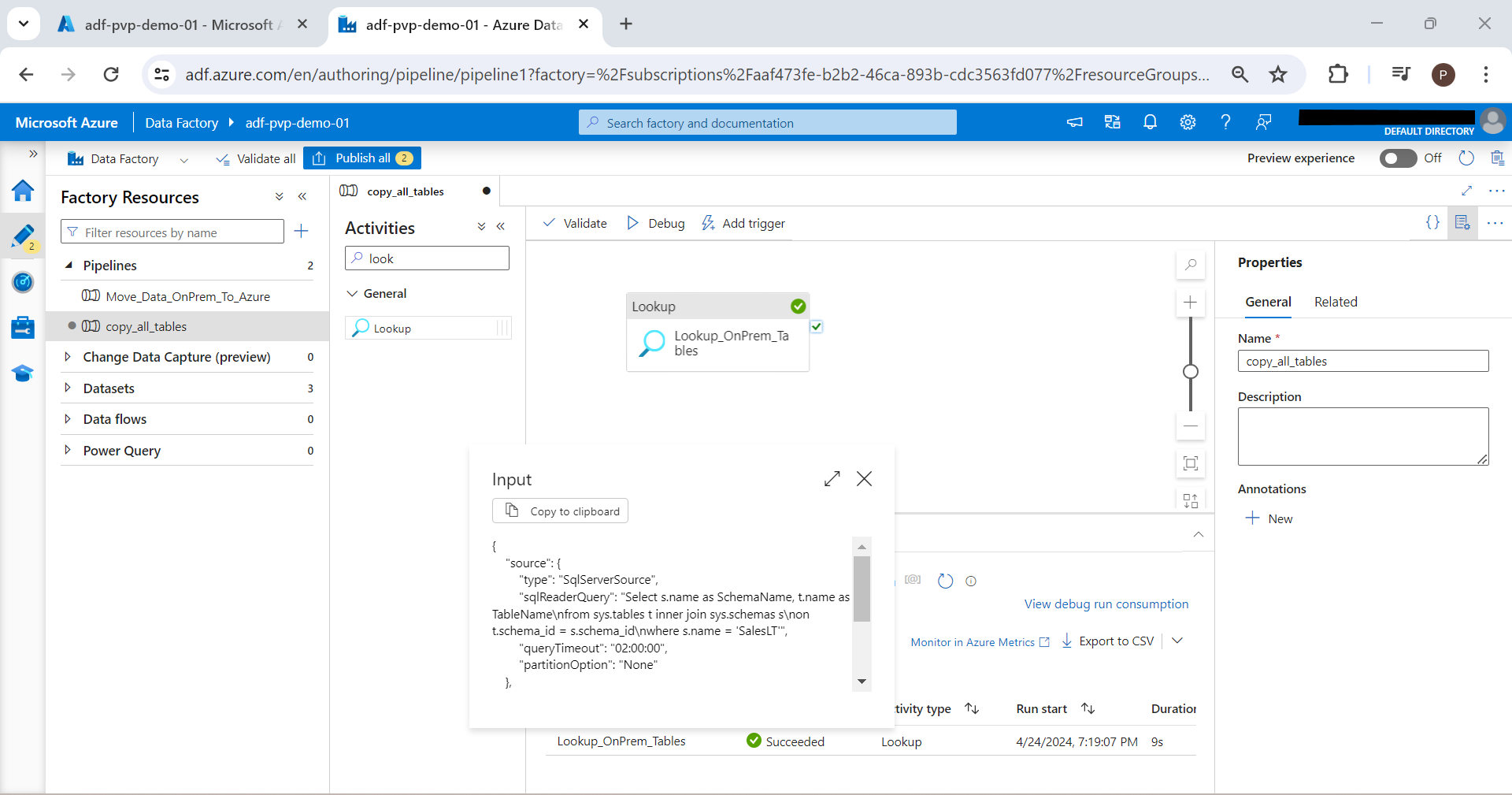
1. Now that we have created pipeline to copy 1 table from source to destination; lets try copy all tables. For this, we will create new pipeline and add Lookup transformation in it

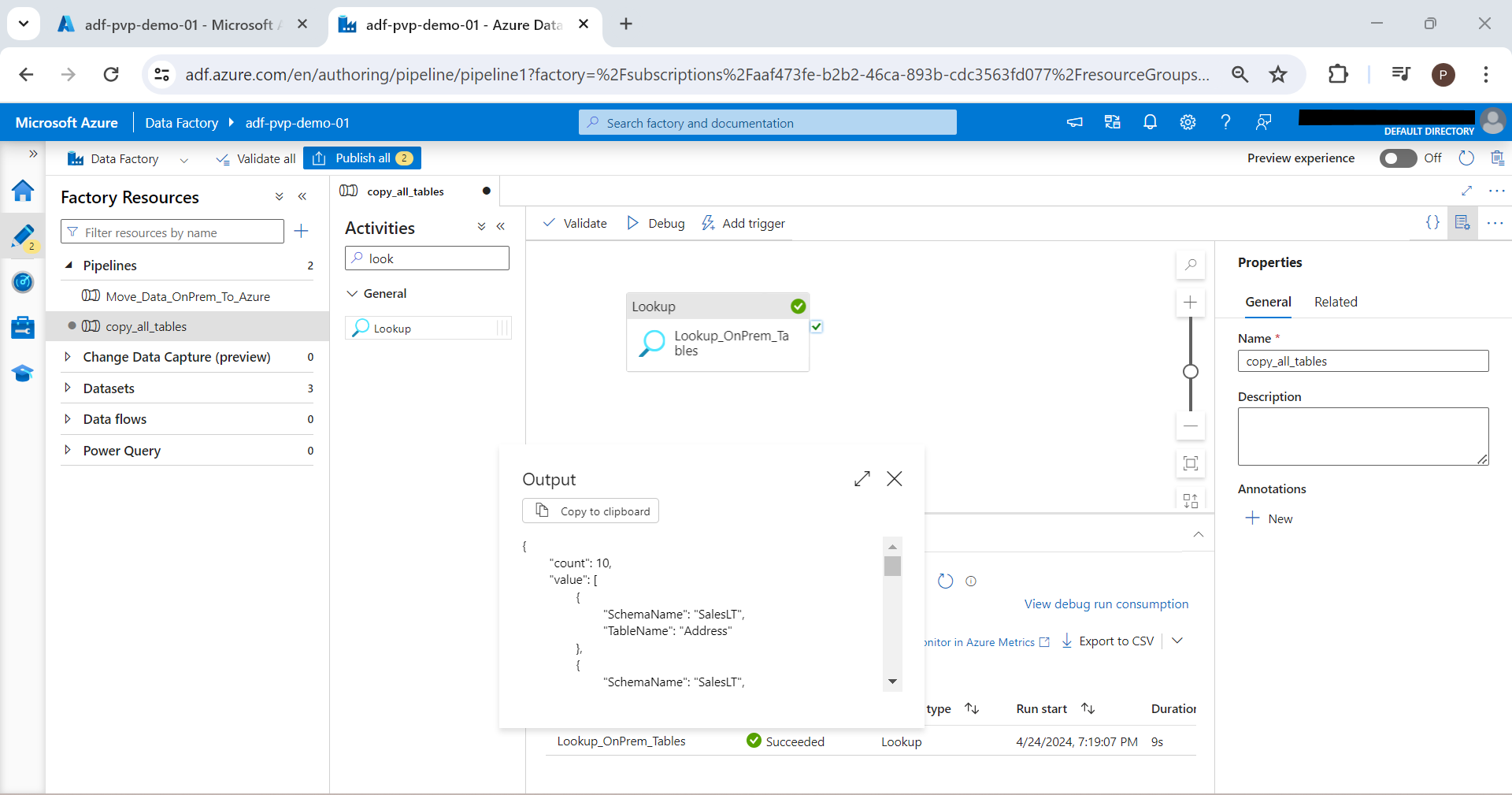


1. Click Settings 🡪 New Dataset 🡪 Select SQL Server 🡪 Continue 🡪 Select Linked Service which we have already created while copying 1 table 🡪 Do not select table and click ok 🡪 Now select Query option and type the query as given below



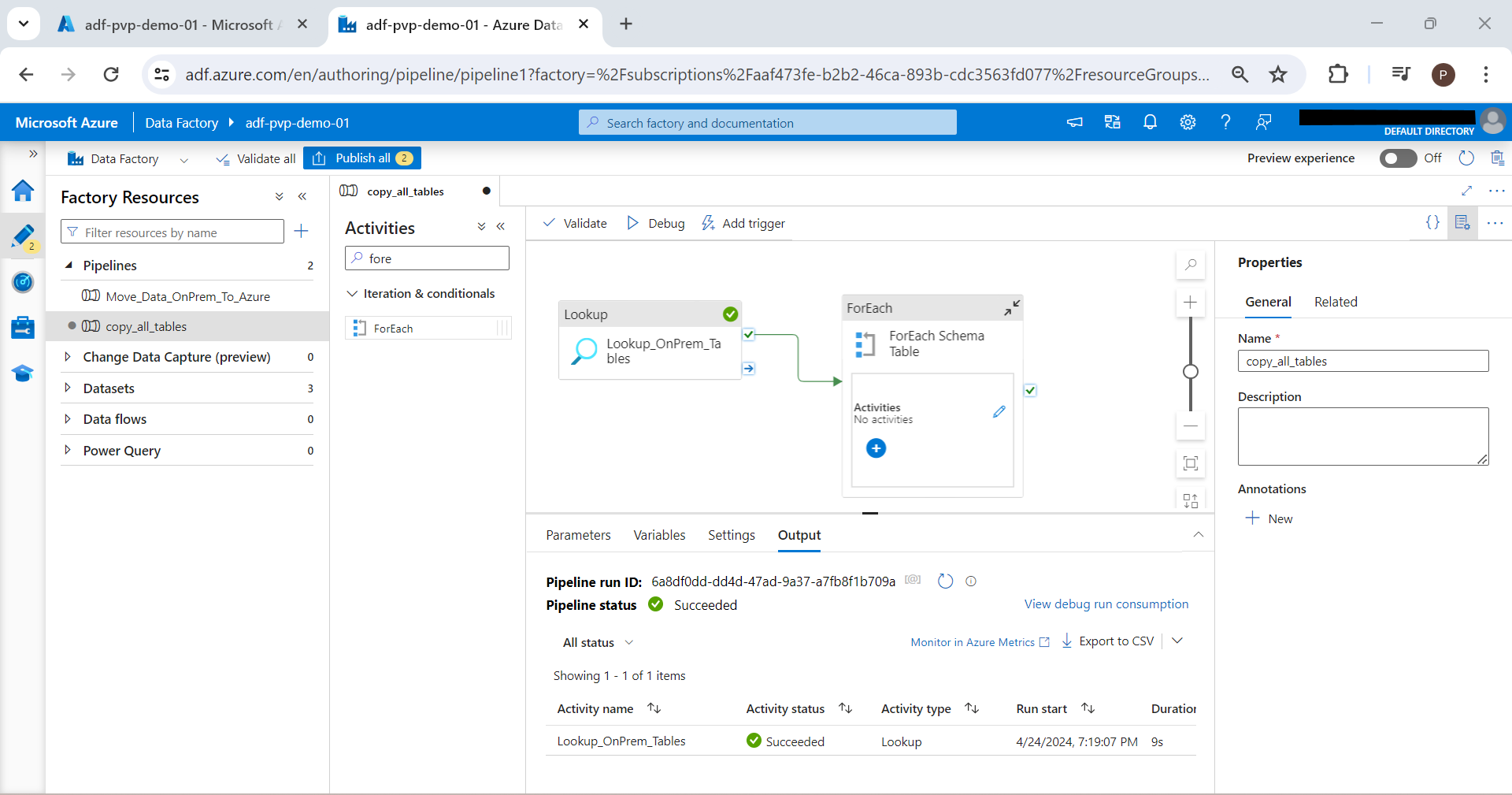
1. We can click Debug and see input and output by clicking activity name





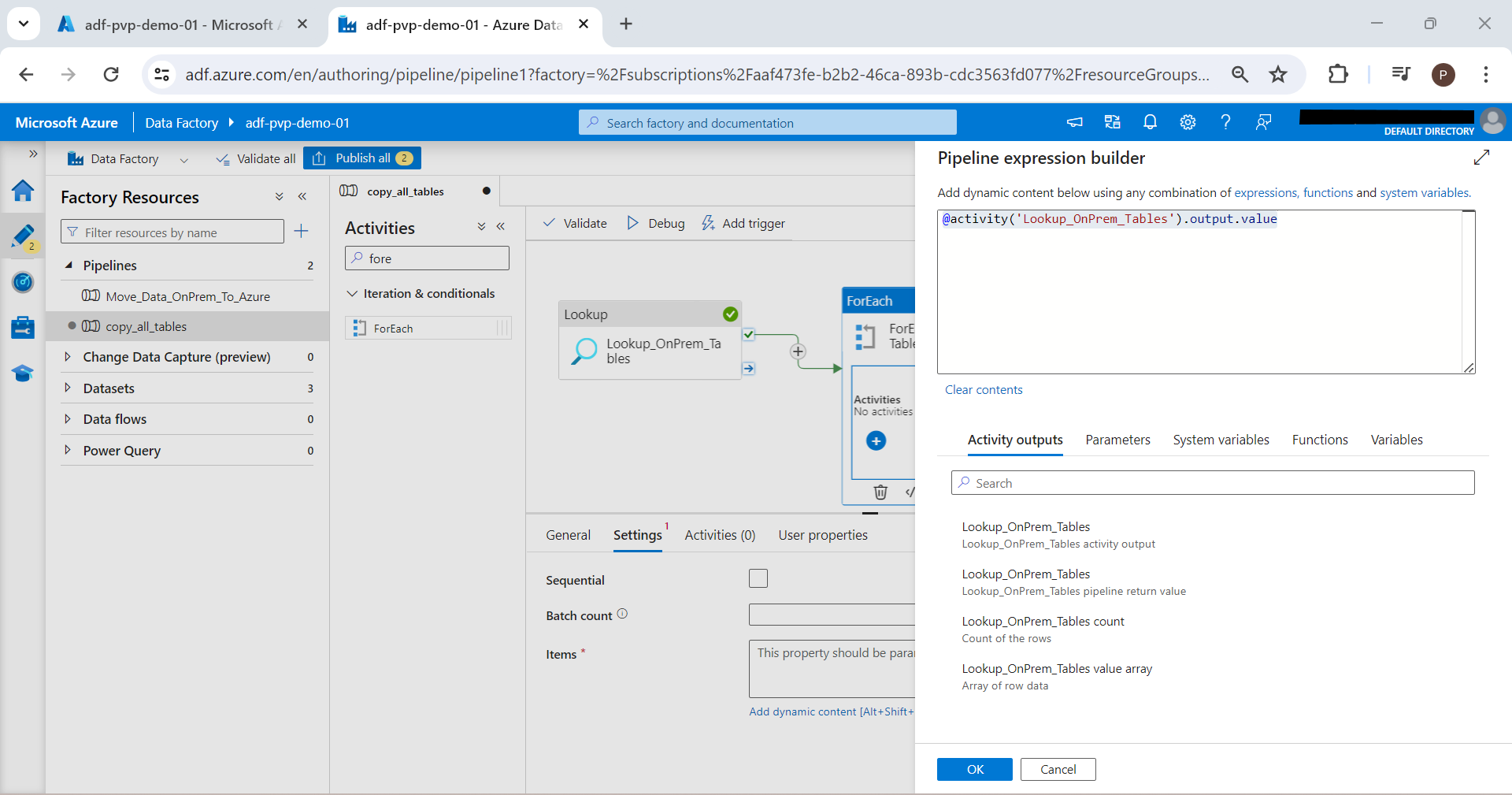
We can see that output structure is generated as JSON. We will use this for next step

1. Now search for activity ForEach and add it. Drag “On Successful completion arrow and attach it to ForEach

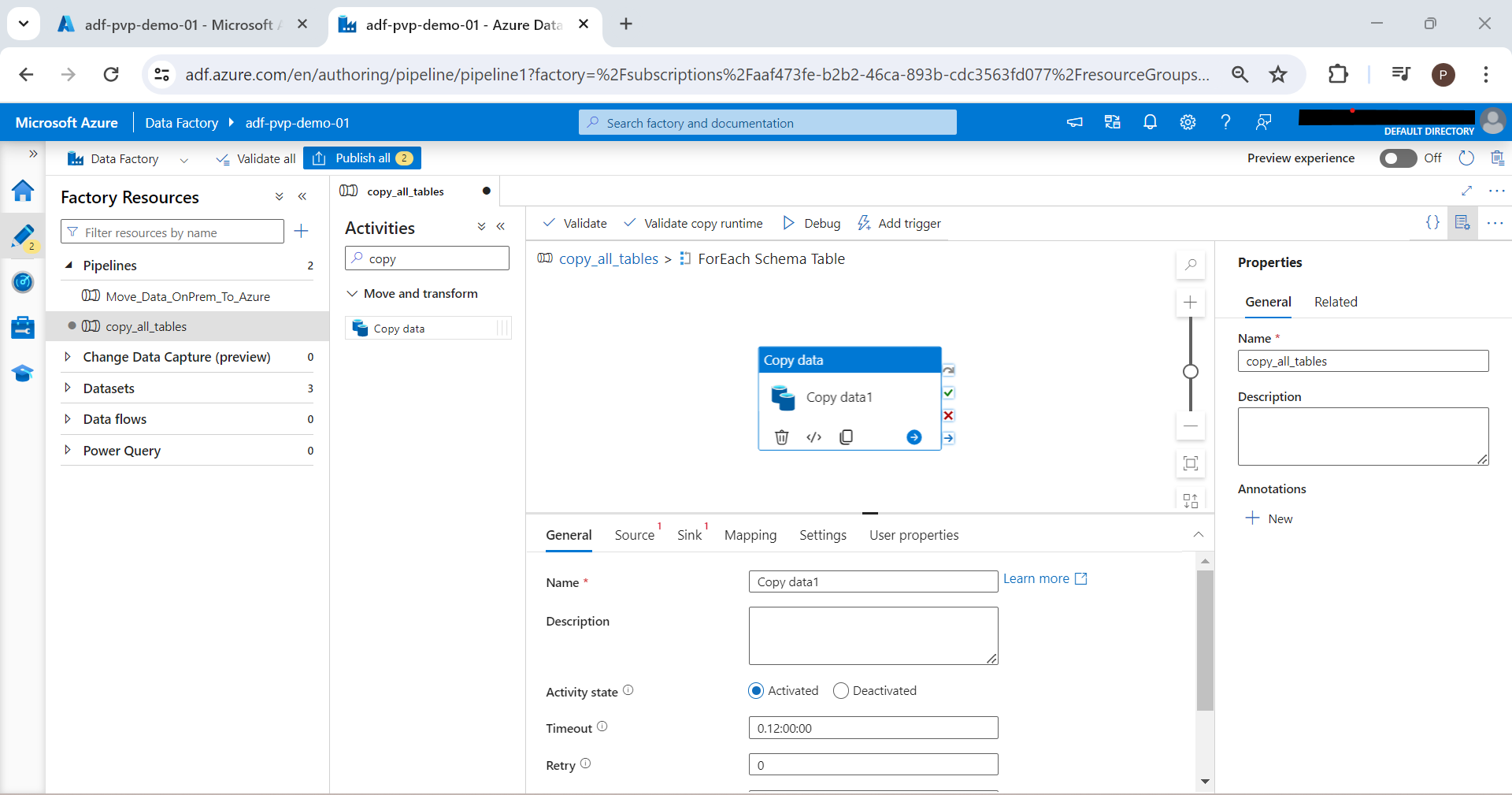


1. We have to configure Settings 🡪 Item 🡪 Add Dynamic content 🡪 Click Lookup\_OnPrem\_Table as shown 🡪 Click OK

@activity('Lookup\_OnPrem\_Tables').output.value



1. Go to Activities and click pencil icon which will take us to code window. Now inside this activity; we will drag copy data activity

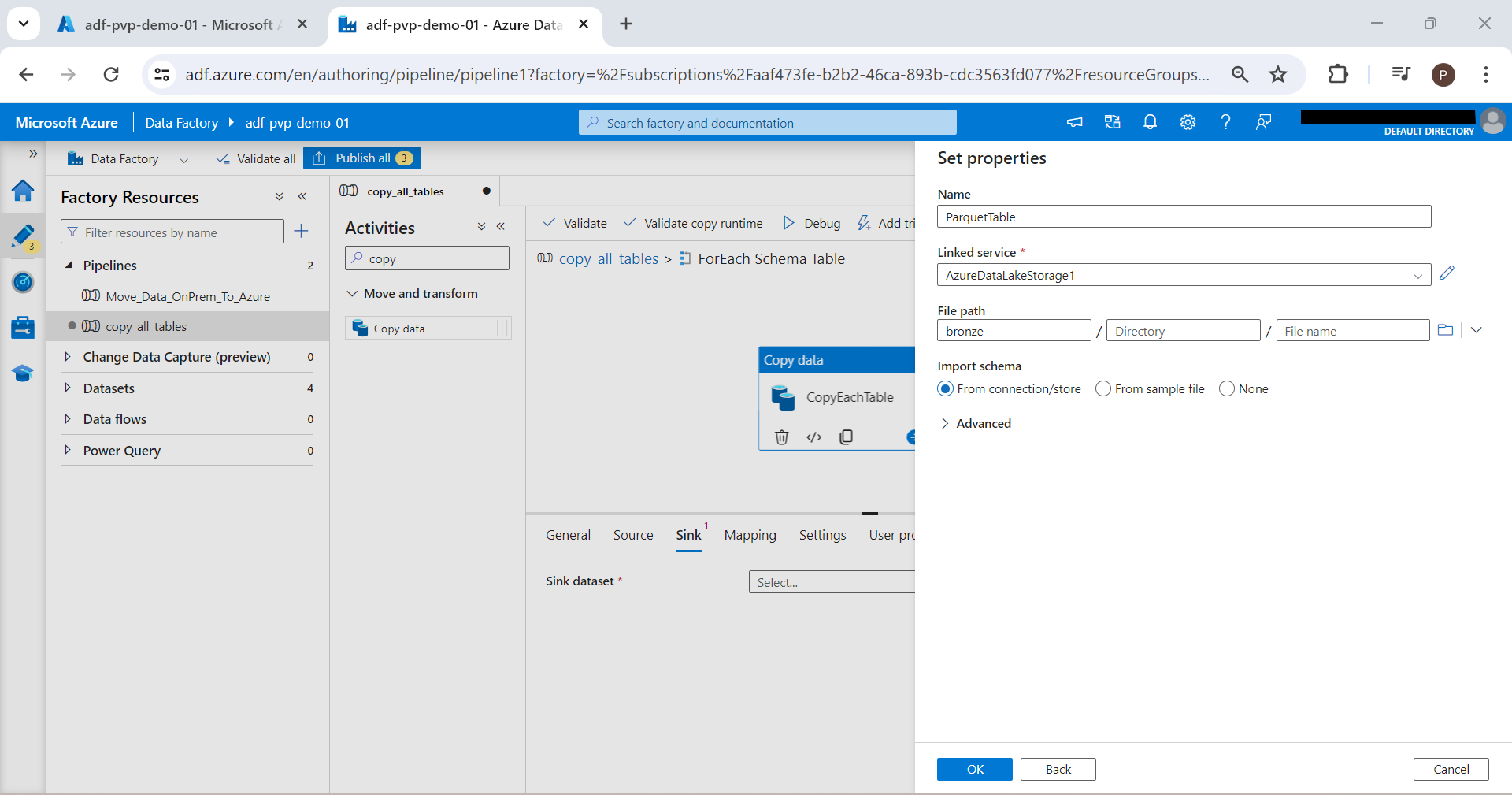


1. Configure source again. Select previously used linked service. Click query and click add dynamic content.

@{concat('SELECT \* FROM ', item().SchemaName, '.', item().TableName)}

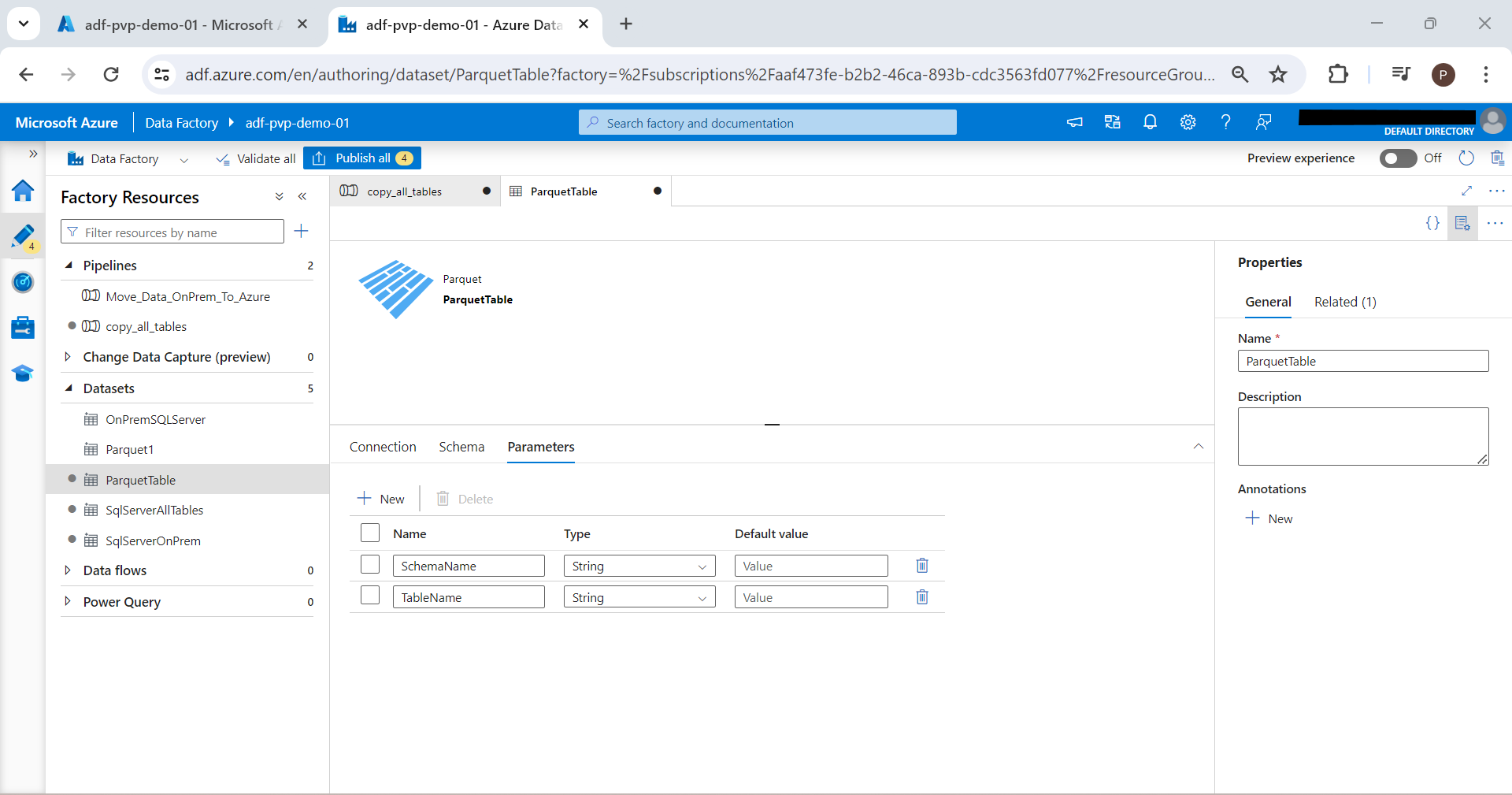


1. Now we have to configure destination (i.e. Sink). Click on Sink and Select New Dataset 🡪 Azure Data Lake Storage Gen 2🡪 Parquet 🡪 Set Name 🡪 Select Existing Linked Service of AzureDataLakeStorage1 🡪 Select File Path as Bronze

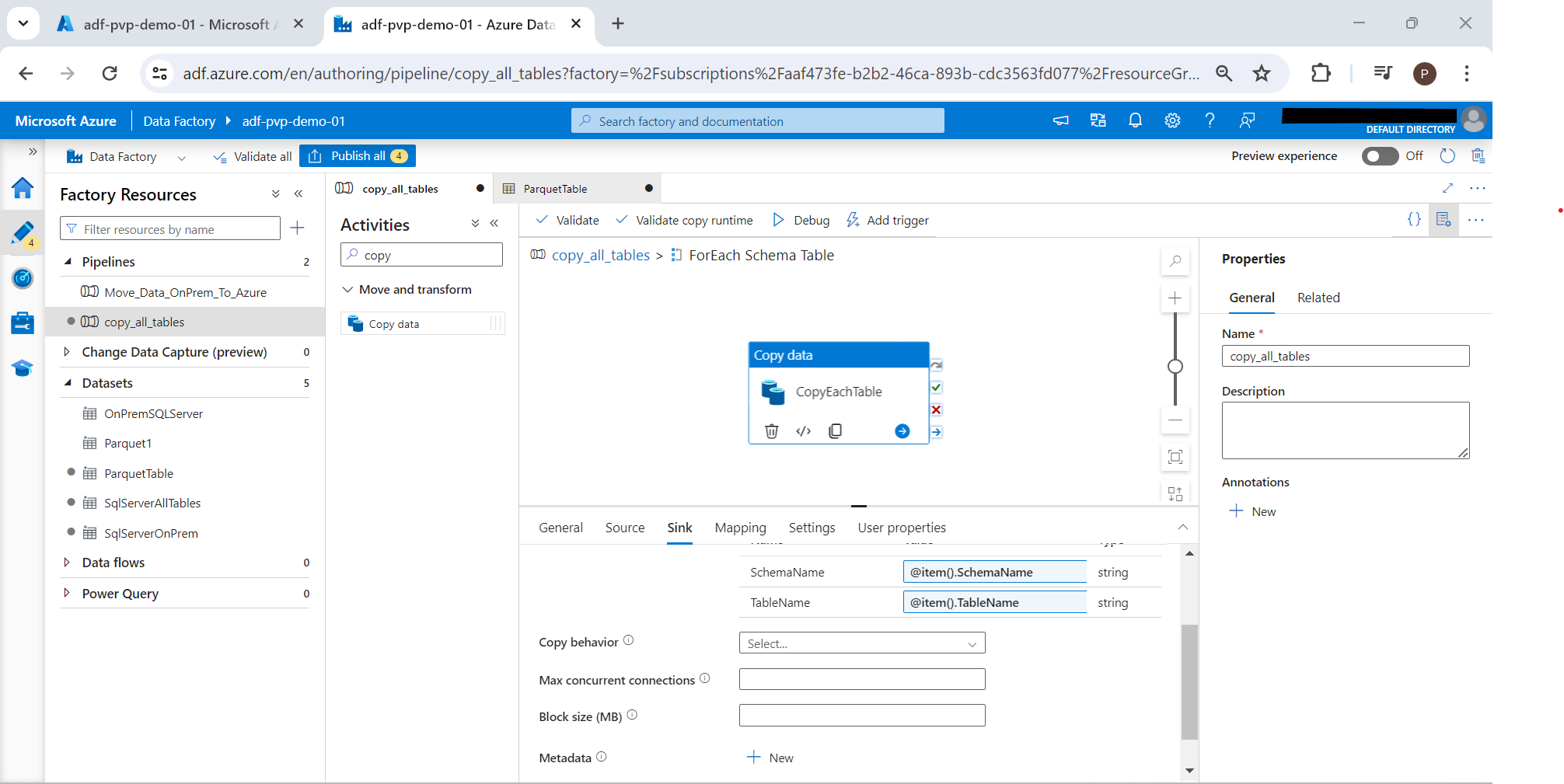


We have to define folder structure as bronze/schema/tablename/tablename.parquet

Open Sink Dataset ; goto Parameters ; create 2 new parameters



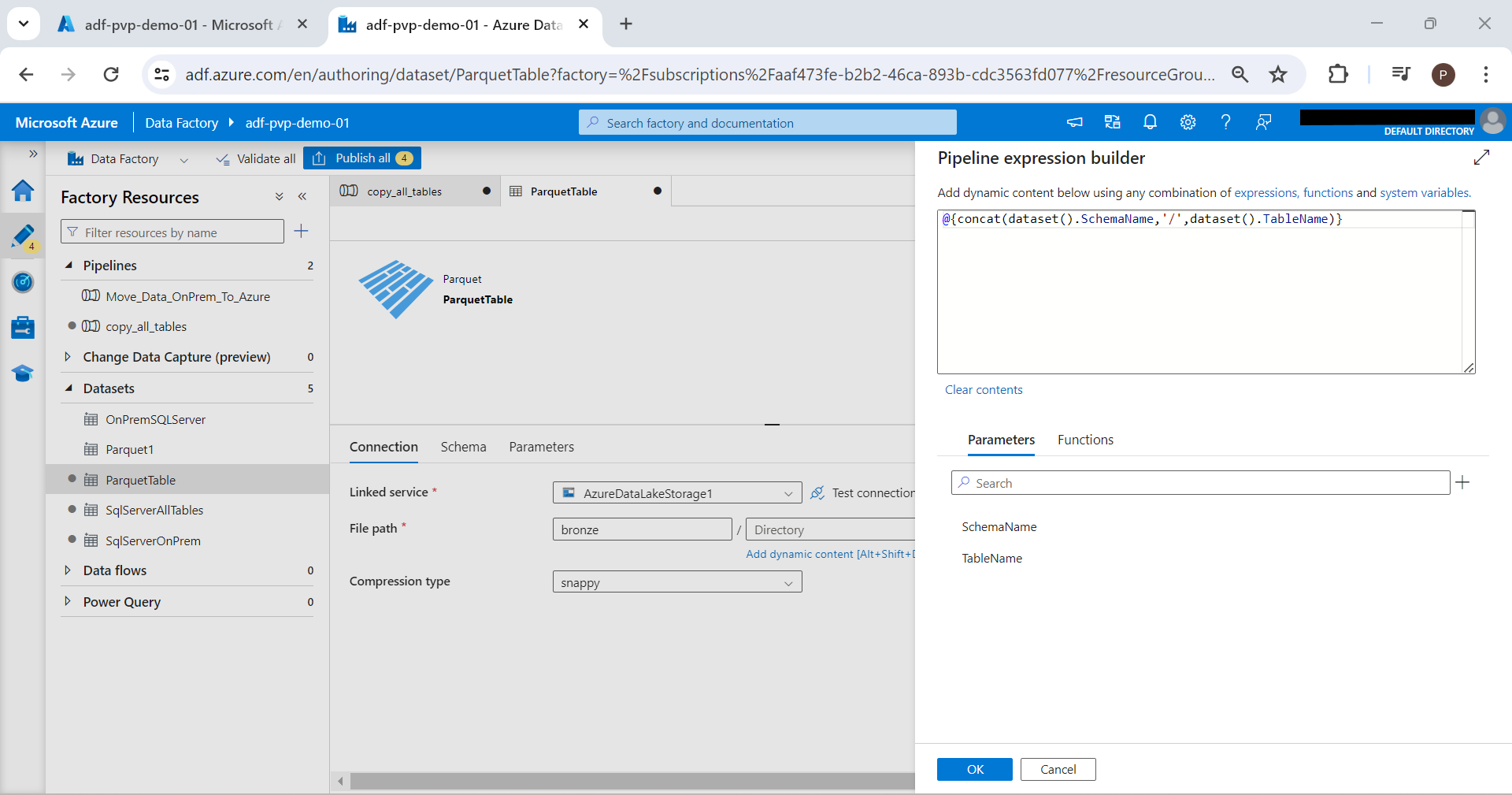
Go back to sink again and we will see 2 new parameters added. We need to add values to the new parameter as dynamic content



Now we can use these dynamic parameters to create folder structure

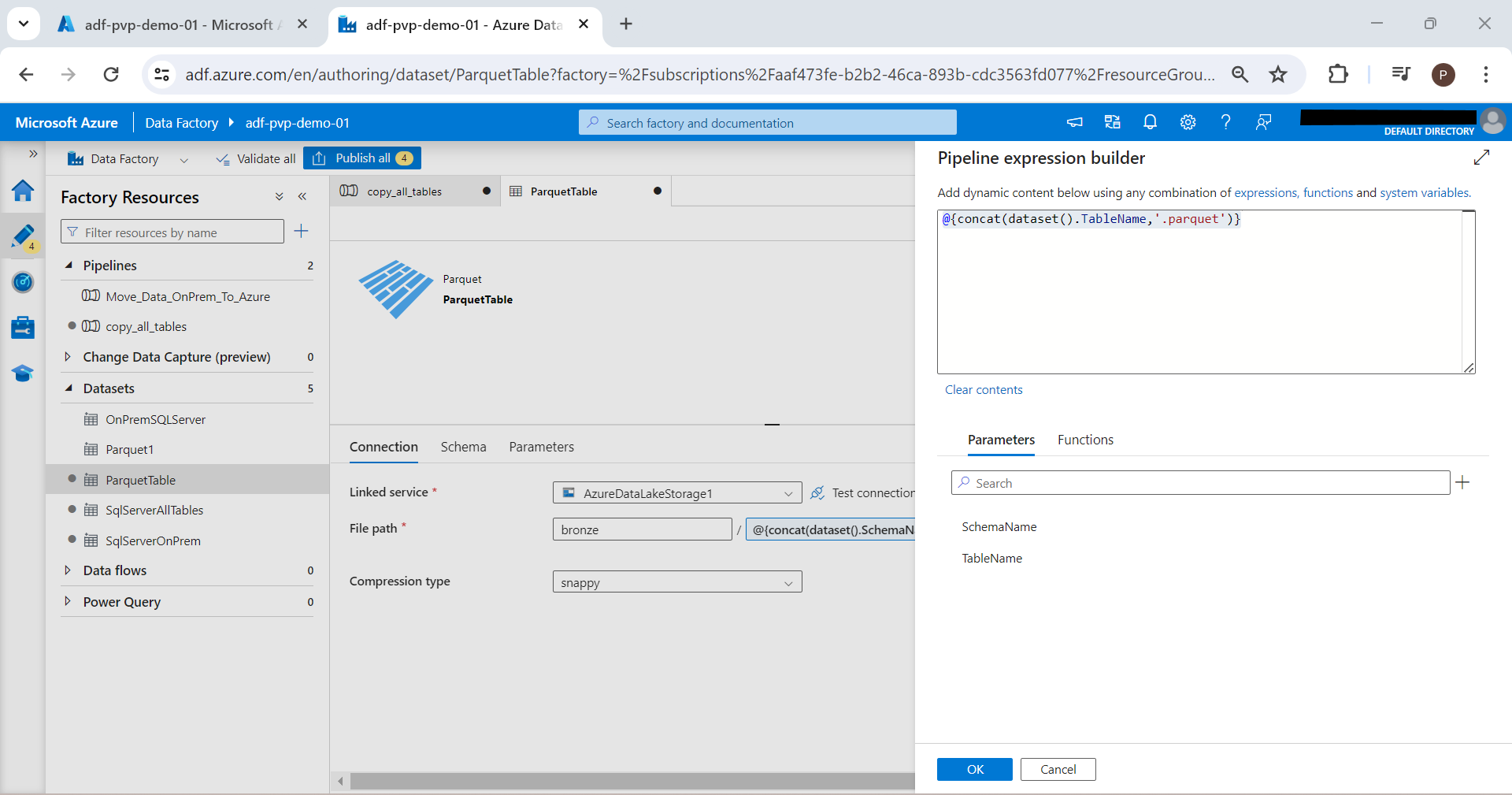
Open the sink again and go to connection and add dynamic content as follows (in Directory):

@{concat(dataset().SchemaName,'/',dataset().TableName)}

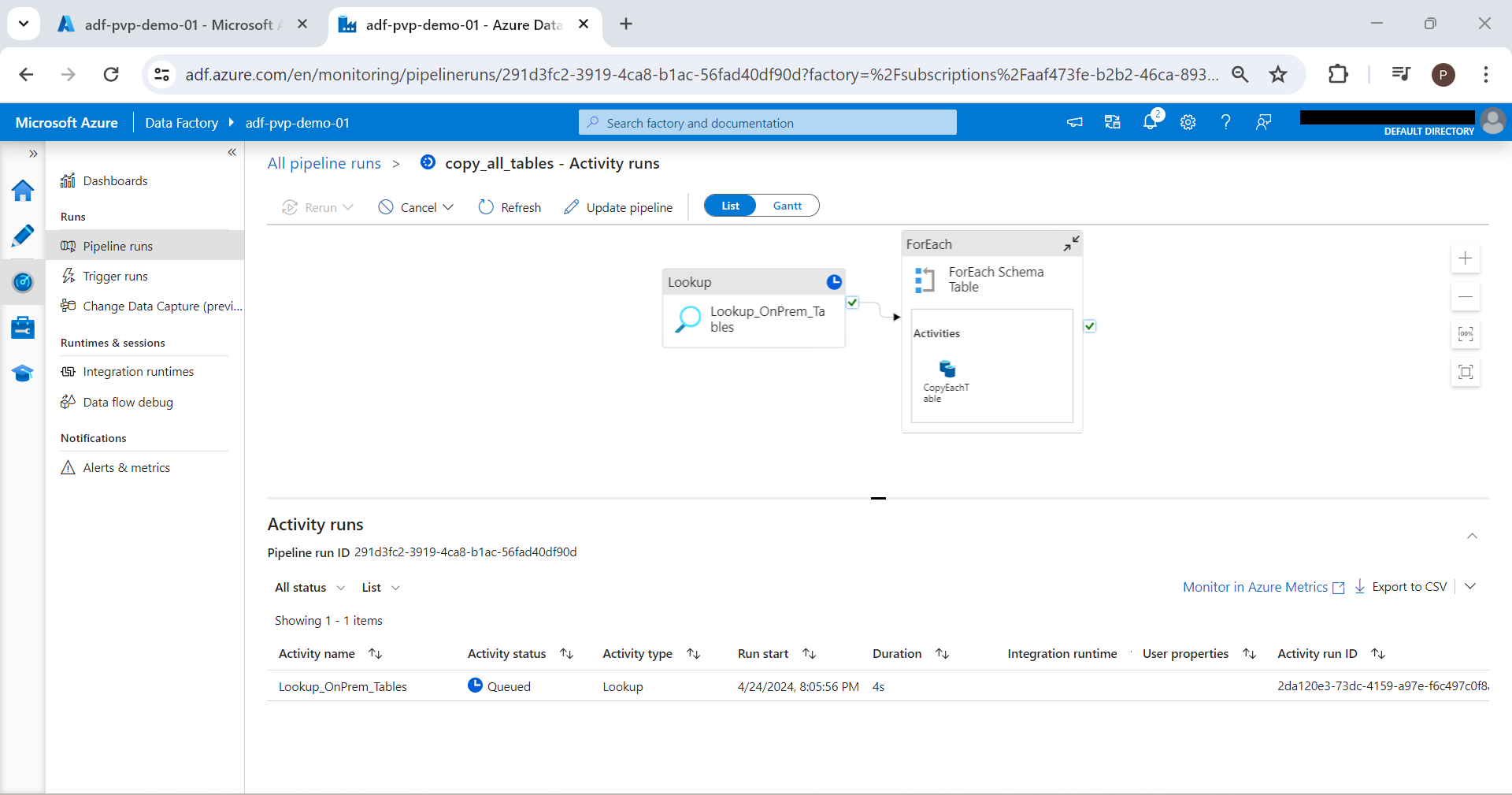


Similarly we have to configure File name as Tablename.parquet

@{concat(dataset().TableName,'.parquet')}



1. We have configured all things now. Its time to publish and trigger the pipeline. Click Publish All and after publishing; click Add Trigger and Trigger Now



1. We can click Monitor to see the progress

