Data Transformation and Loading Using Azure Cloud Services

Kishore Kumar Loganathan

05-06-2023

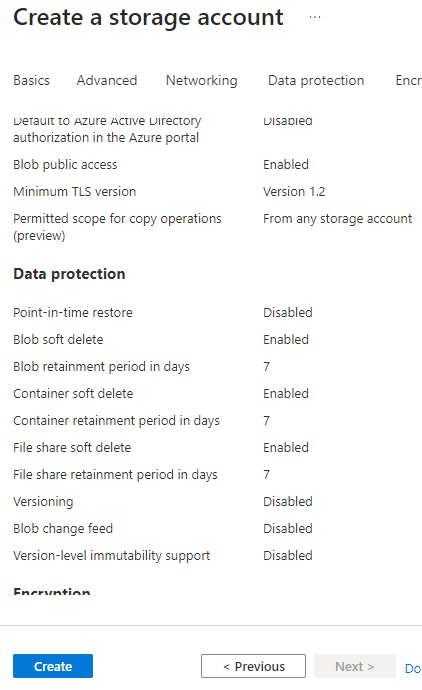
Table of Contents:

1. Azure Environment Setup
   1. Creation of Azure Data Lake
   2. Creation of Azure Synapse Analytics
   3. Creation of Azure Cosmos DB
2. Data Injection
   1. Detailed Steps involved
   2. Challenges Faced
3. Data Transformation
   1. Detailed Steps involved
   2. Challenges Faced
4. Data Loading
   1. Detailed Steps involved
   2. Challenges Faced

**Azure Environment Setup**

**1. Creation of Azure Data Lake:**

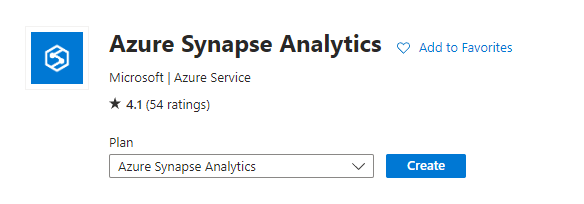
* Create an Azure account if you don't have one.
* Navigate to the Azure portal and create a new Azure Data Lake Storage account using the create a resource option.
* Create a new Resource group. Moving forward all the resources will under this same Resource group that was created.
* Configure the storage account settings, such as storage type, redundancy options, and access control.
* Check the Hierarchical Namespace option to create Data Lake Gen 2 storage account.



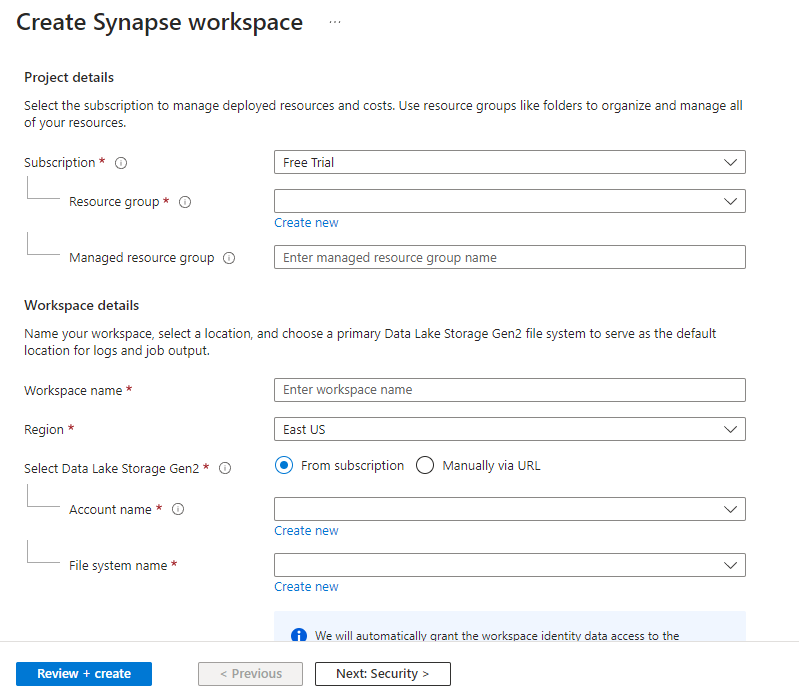
* Click create to start deploying the storage account
* Now wait for it to deploy and you could find the storage account under Resources in the Azure services portal.

1. **Creation of Azure Synapse Analytics:**

* Create the Azure Synapse analytics resource by clicking the “Create a Resource Button” and searching for Azure synapse Analytics.



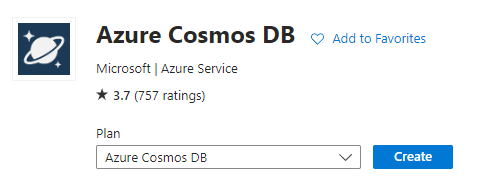
* Now Create synapse Workspace interface opens.



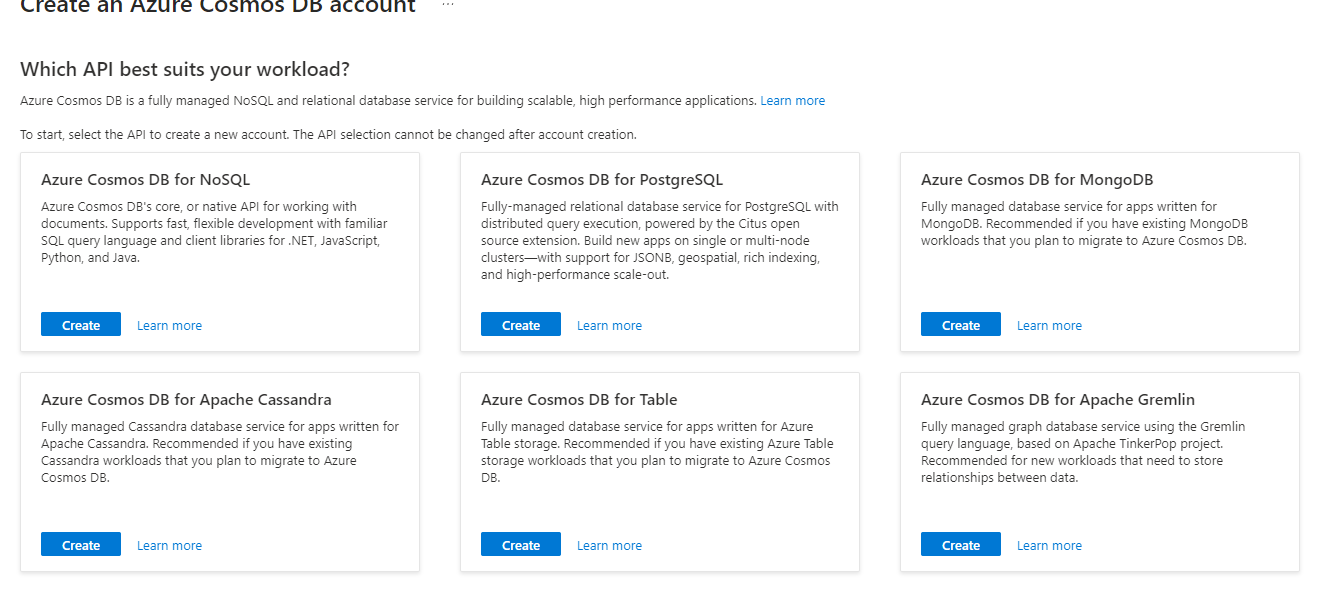
* Fill all the required sections and Click “Review + Create”.
* You can access the Synapse Studio from the Azure portal under your Resources.

**3. Creation of Azure Cosmos DB:**

* Create the Azure Cosmos DB resource by clicking the “Create a Resource Button” and searching for Cosmos DB.



* Once you click create you will be directed to the following page where you can select the SQL API you want your Cosmos DB to work on.

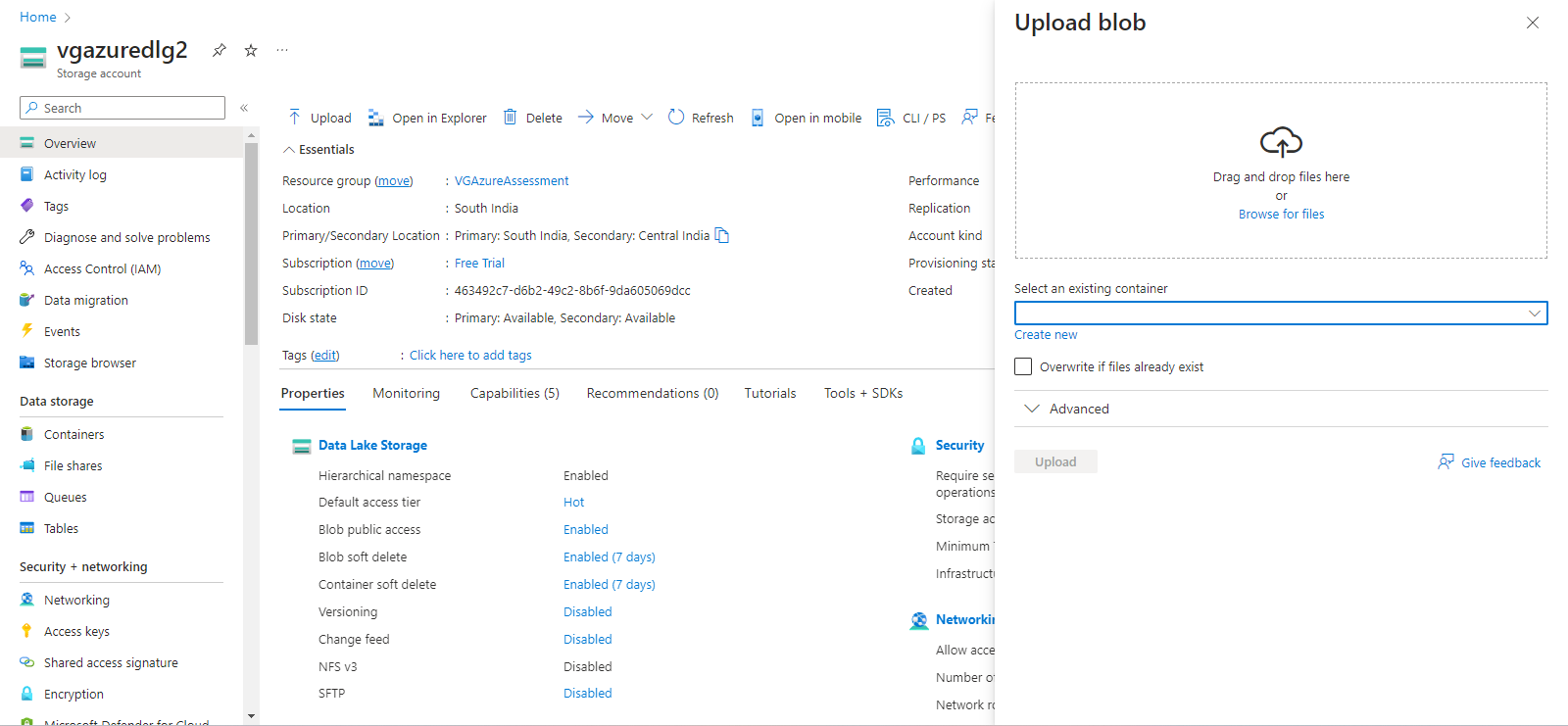


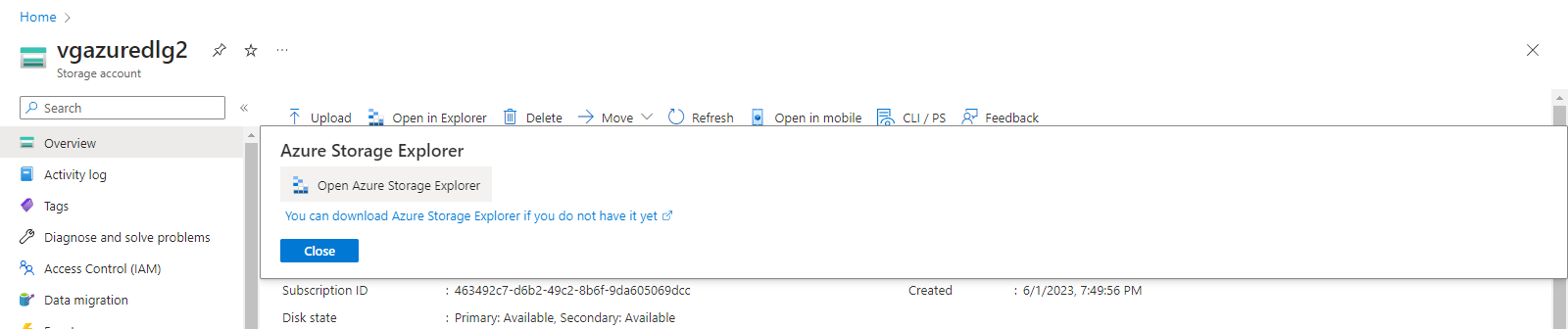
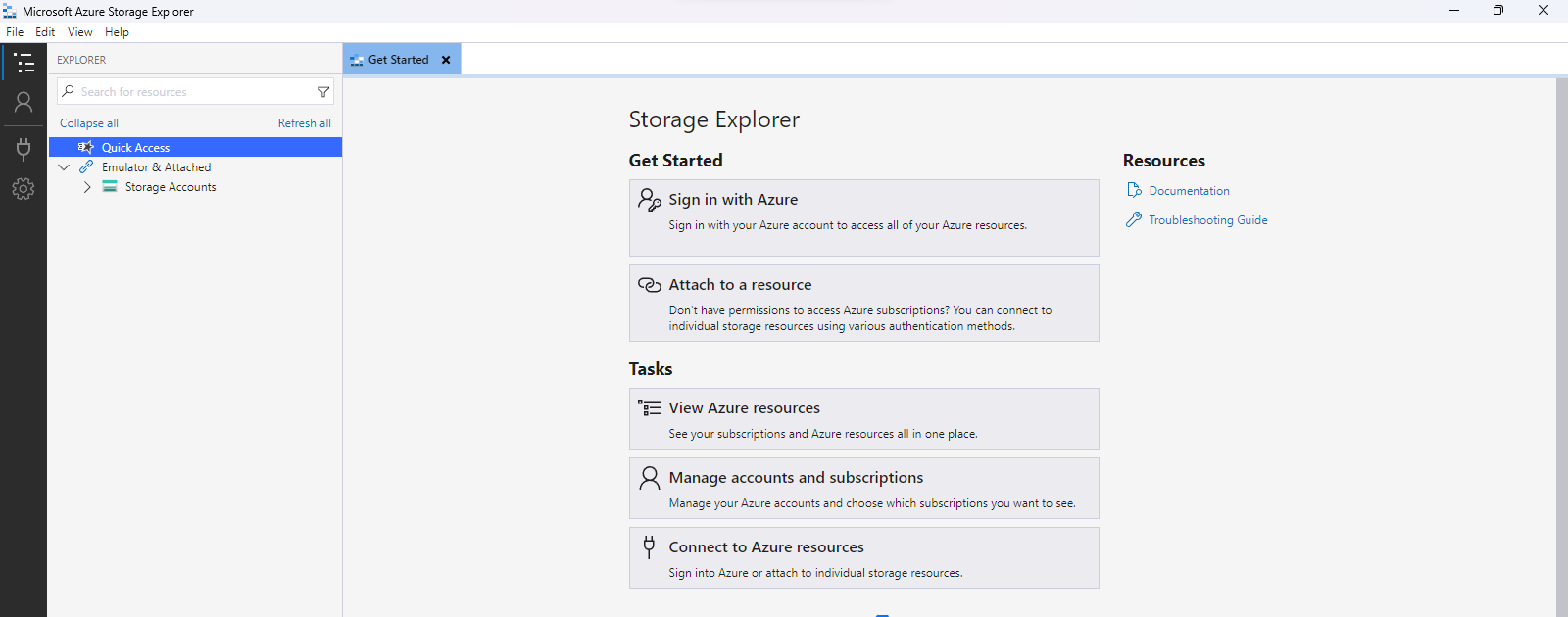
* We can select Azure cosmos DB for NoSQL for our use case scenario.
* You will further be directed to a page where you have to fill the mandatory information to create your Cosmos DB.
* The Cosmos DB resource will be deployed with the given name once you click create.

Data Ingestion

**1. Detailed Steps involved:**

* First Step of the Process is to ingest data into our Data Lake Gen 2 resource.
* Click on the created storage account resource to open access the Storage Account.
* To upload the file from your local directory click on upload and create a container for your file and click upload.
* Now you can find the uploaded file in the containers under Data storage Tab in the left.

****

* Another way to upload the data is through the Azure Storage Explorer.
* 
* One can download the storage explorer and use it locally and it also makes you job of accessing the containers In your Data Lake easier.
* Using the Sign in using Azure option you can link your storage explorer with your Data Lake.
* 

Challenged Faced:

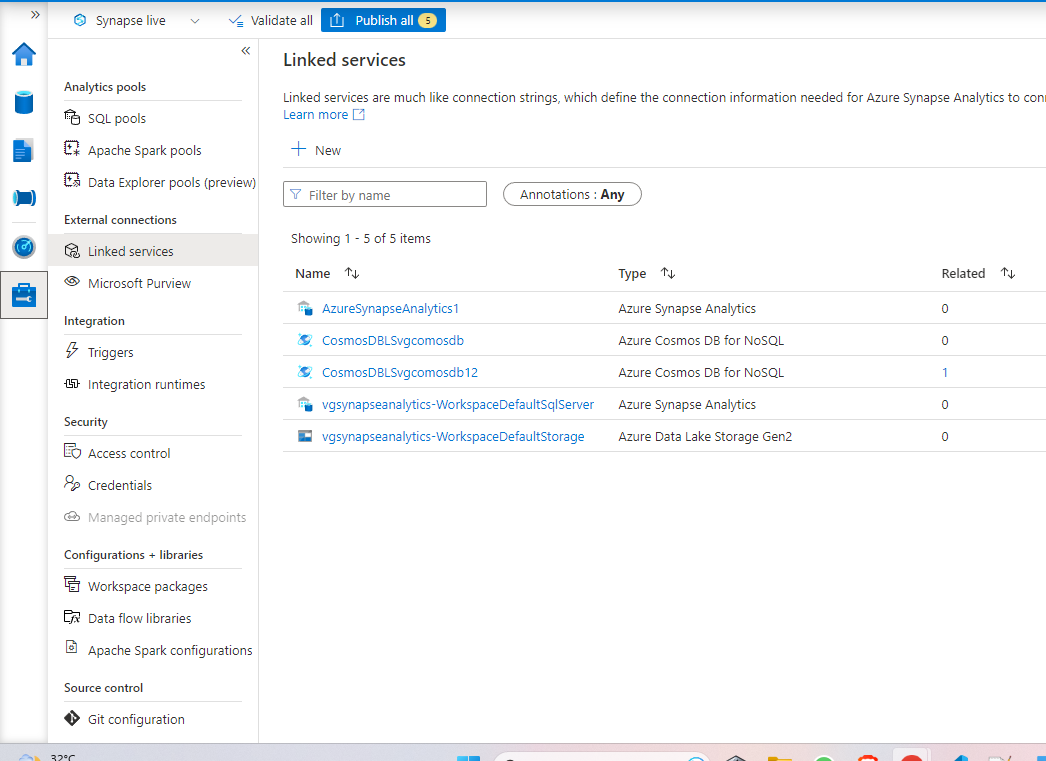
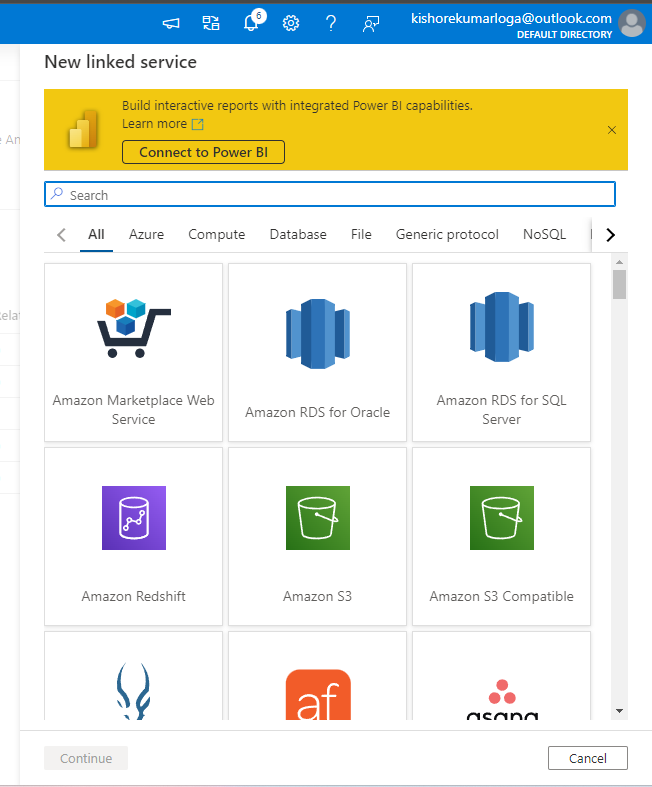
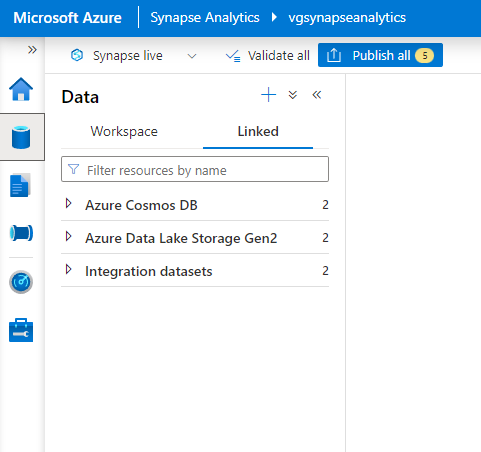
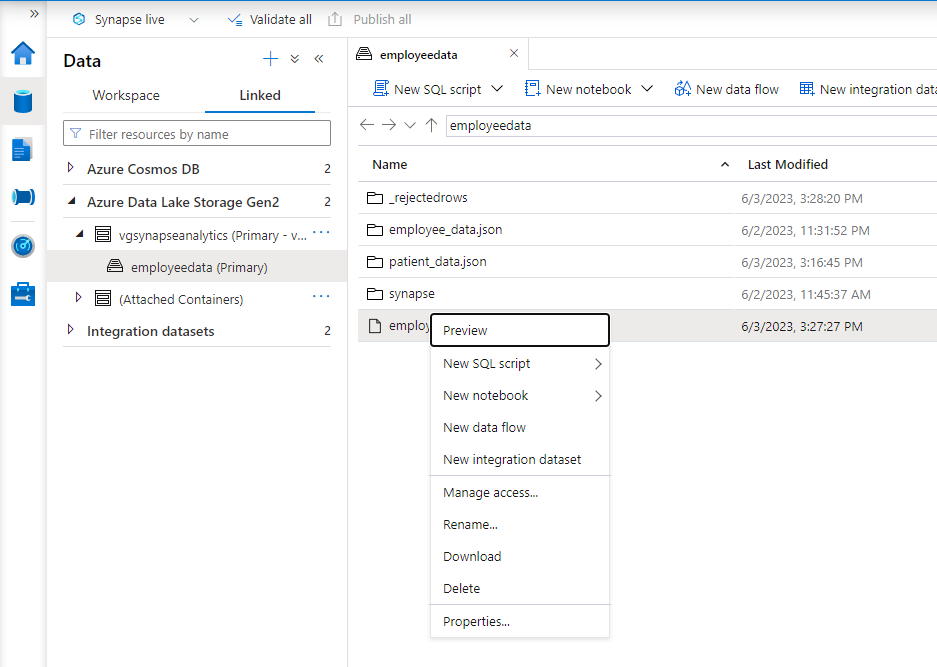
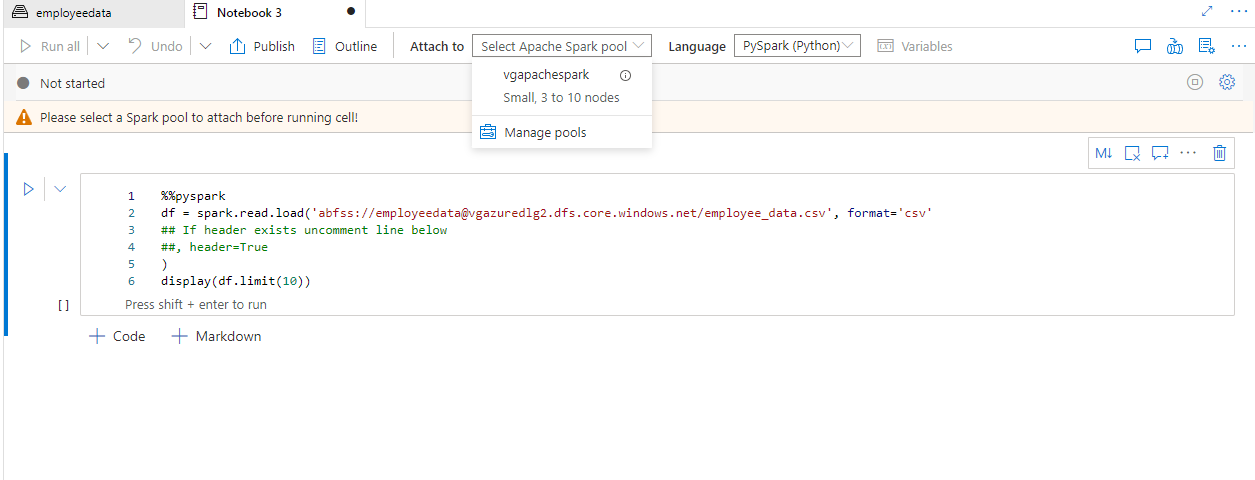
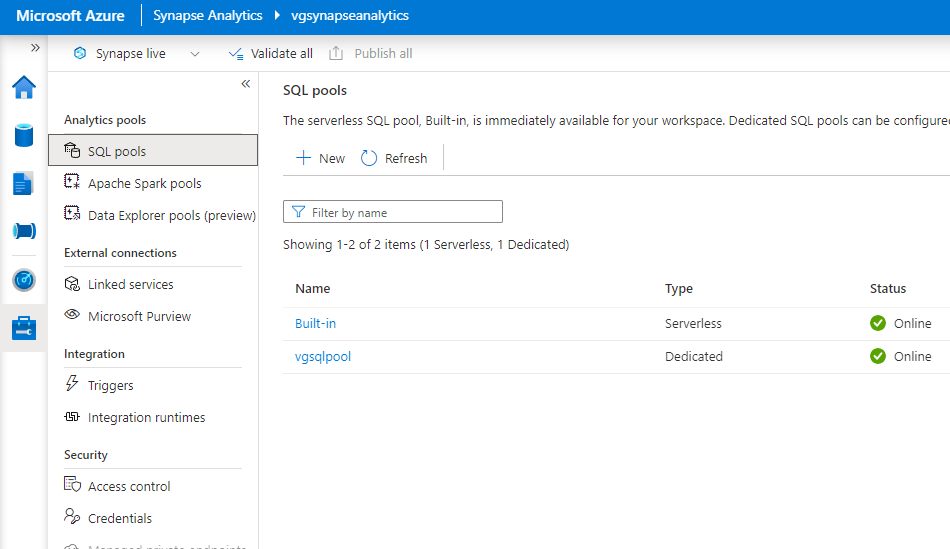
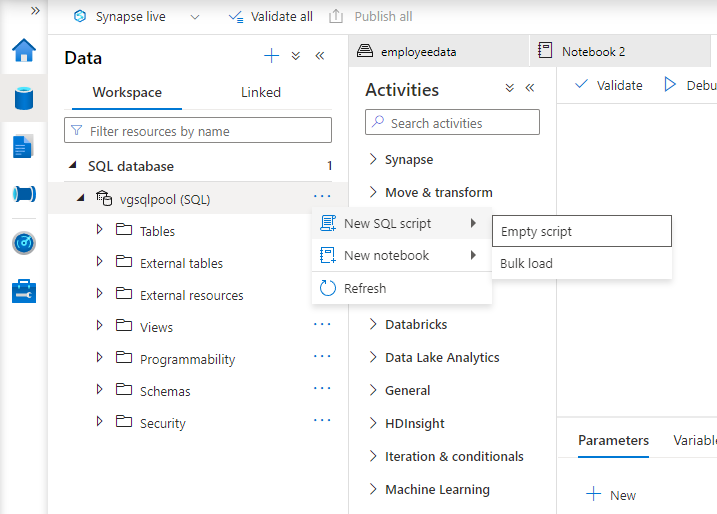
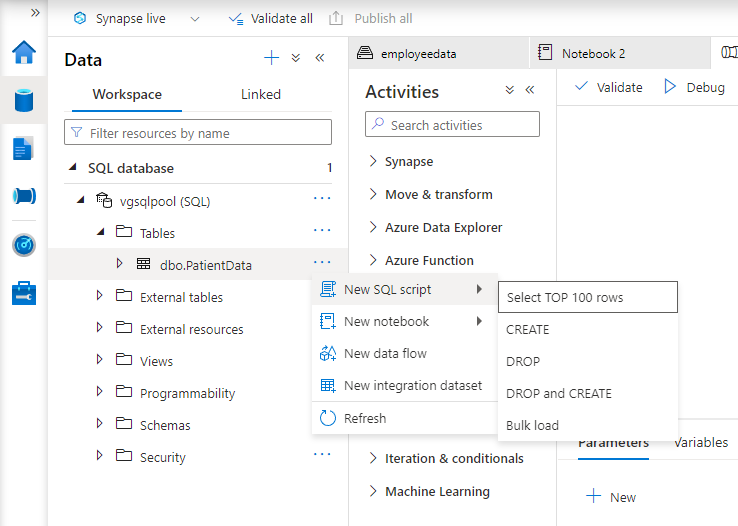
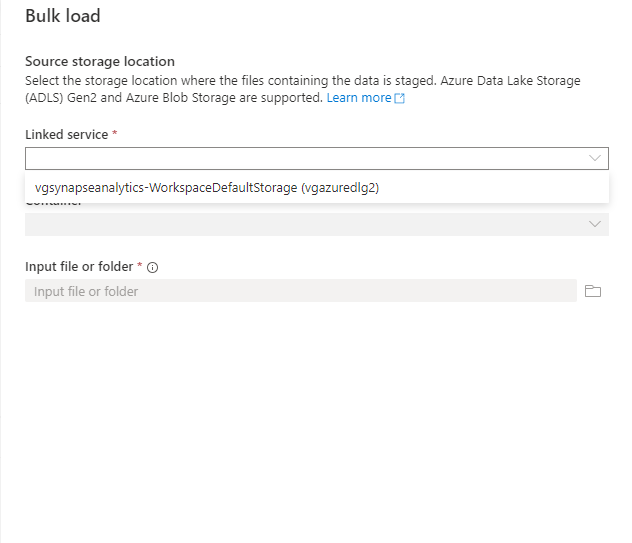
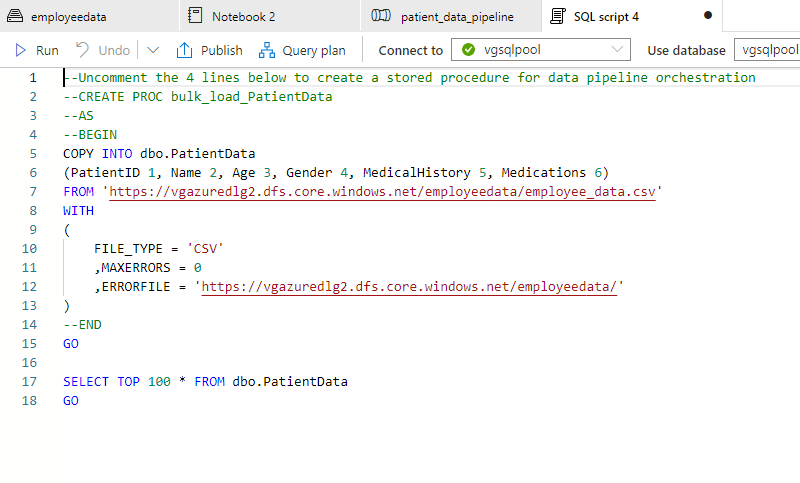
1. Figuring out where Data Lake Gen 2 was since its not a separate service but a service within storage account.

2. Linking Data Lake to Synapse Analytics.

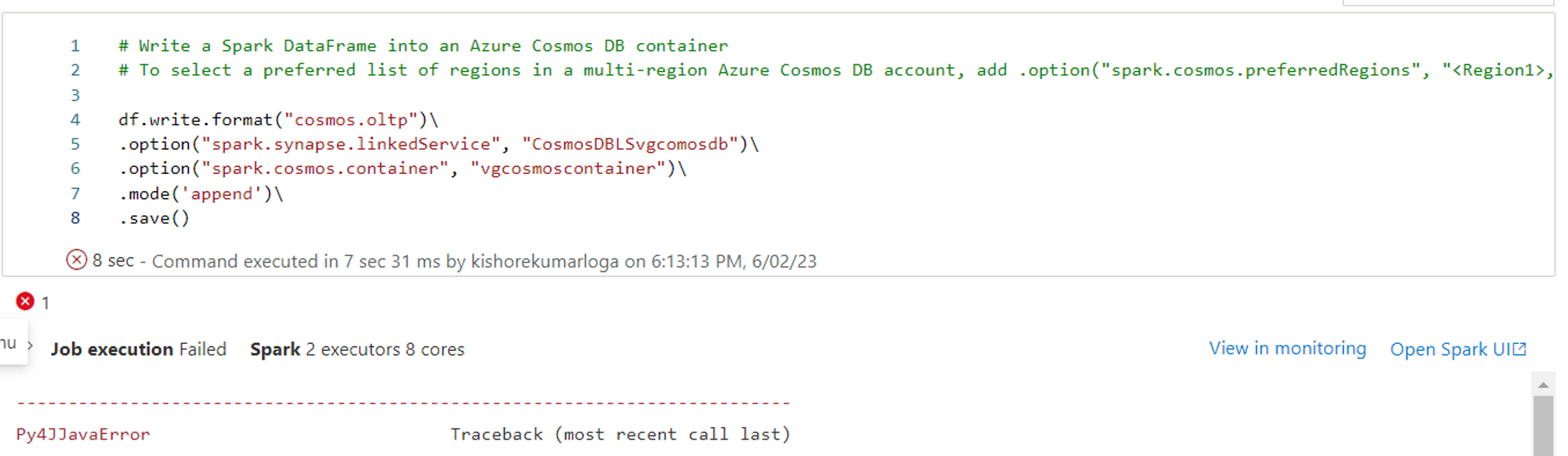
3. Linking the containers in Data Lake to Azure Storage Exlporer and directly accessing the containers.

Data Transformation

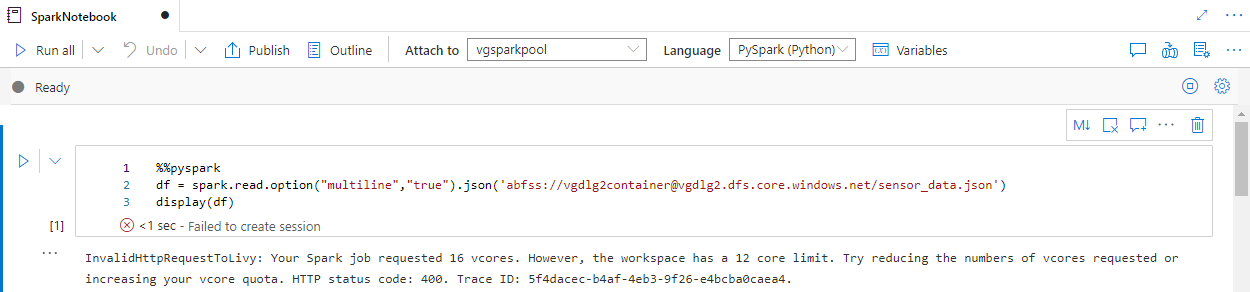
**1. Detailed Steps involved:**

* Once you have ingested the data into Data Lake, the next step is to transform the data using Azure Synapse Analytics.
* Accessing data from the Data Lake:
  + To access the files in our data lake it is necessary to form a connection.
  + To form a connection we first need to create a linked service.
  + Select the Manage tab in the right most column in our Synapse studio.
  + 
  + Now click on New to create a new linked service.
  + 
  + Now search for Azure Data Lake Gen 2 and select it. Once you click continue you will be asked for your subscription and your storage account name. Click create once you are finished wait for some time for it to deploy.
  + Once its done you can find your files uploaded under Linked in the Data tab.
  + 
* Transforming Data using Spark Notebooks:
  + Now select your container in the Azure Data Lake Storage Gen2 to view its content.
  + 
  + Right Click on the file you want to work on and click New Notebook and Load to Data Frame.
  + A new notebook will be opened for you and create a new apache spark pool by clicking Manage pools option or select the Spark pool you wish to work on if already exists.
  + 
  + We can also change the Language by clicking the language drop down box.
  + Now the necessary changes are made to the unstructured dataset.
  + Finally the file is saved to the data lake in the CSV format.
* Creating a dedicated SQL pool and ingesting the file into the table.
  + Go to the Manage Tab and click SQL pools to view all the pools that exists.
  + Now click New to create a new dedicated SQL pool.
  + 
  + Once it gets created you can access it in the Data tab under Workspace as SQL Database.
  + Now right click on the SQL pool created and click Empty script.
  + 
  + In the SQL script now create a new table along with the appropriate columns and of similar data types to the csv file saved in the data lake.
  + Once the query is successfully executed you can check if the table is created by refreshing the SQL pool and check it under the Tables drop down list.
  + Now right on the table and navigate to New SQL script and you will find the Bulk Load option, select it.
  + 
  + Now select the linked service, the container name and select the input file which you had previously saved from the Spark notebook.
  + 
  + After selecting the file and after the initial detection is successful you can open the SQL script consisting of query that was already written.
  + 
  + Make the necessary changes to the query and execute the query to load the transformed data of our spark notebook into the table that was created in the SQL pool.
  + Once the data is loaded you can select the table and run a SELECT all command to check if all rows have been loaded.

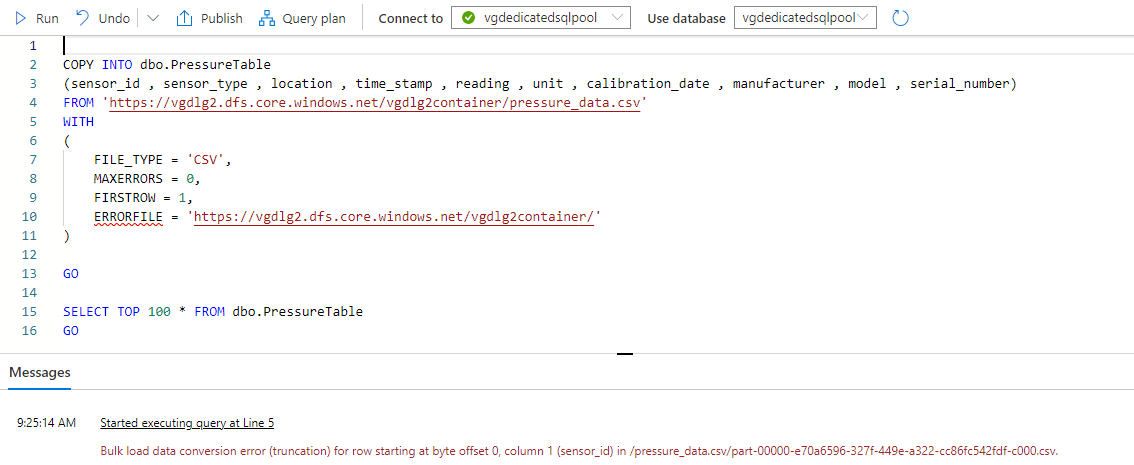
Challenges Faced:

1. 

Py4JJavaError: An error occurred while calling o3938.save. : org.apache.spark.SparkException: Writing job aborted.

2. 

InvalidHttpRequestToLivy: Your Spark job requested 16 vcores. However, the workspace has a 12 core limit. Try reducing the numbers of vcores requested or increasing your vcore quota. HTTP status code: 400. Trace ID: 5f4dacec-b4af-4eb3-9f26-e4bcba0caea4.

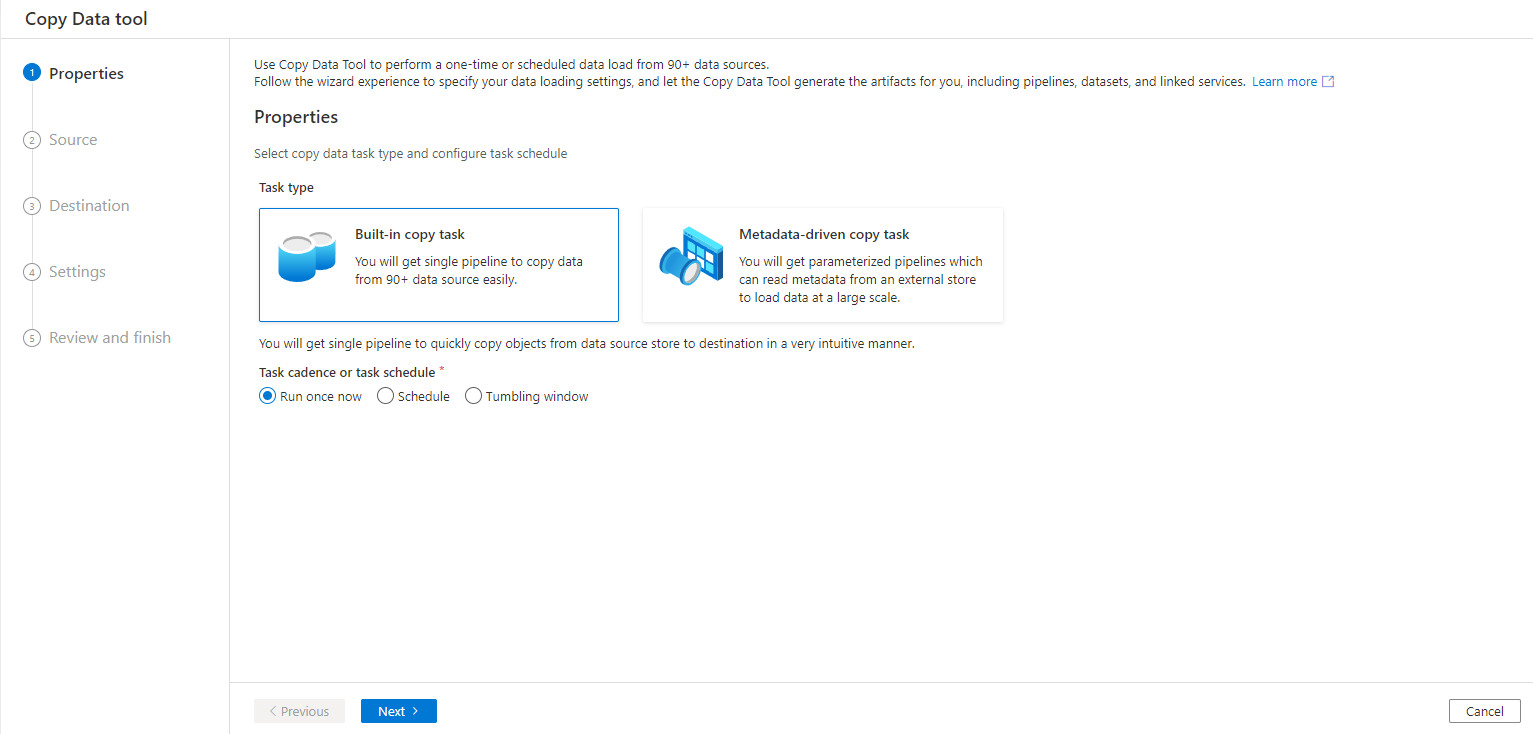
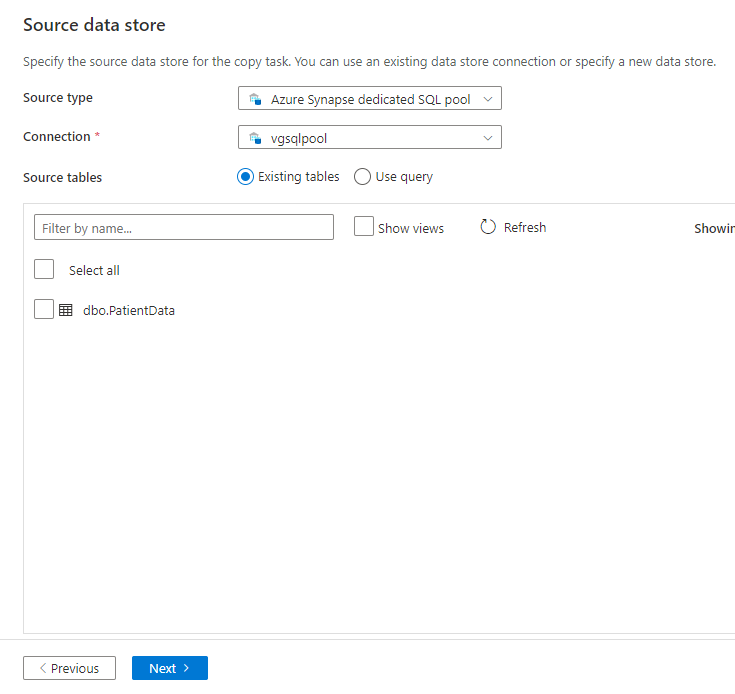
3. 

Bulk load data conversion error (truncation) for row starting at byte offset 0

4. There aren’t proper documentation available online for Synapse Analytics.

Data Loading

**1. Detailed Steps involved:**

* Once the data is loaded into the table in the SQL pool, the next step is to load the table into Cosmos DB.
* For this process there are two main steps:
  + Creating a Database and a Container in Cosmos DB
  + Creating a pipeline that copies the data from the table in Data Lake to the container in Cosmos DB.
* To create a database in your Cosmos DB, go to the cosmos db resource and open it.
* Now select Add container and enter the database and container name for the container you are creating. It is essential to select **Analytical Store On.**
* Once the container is deployed we can go back to Synapse Analytics to create the Pipeline.
* To create a pipeline go to the Integrate tab in the Synapse studio and select **Copy Data Tool** in add new resource option.
* 
* Select Run once now radio button and click next. Select the desired Source data.
* Select the source type, Connection and the Tables to proceed.
* 
* Click next and then select the destination as your cosmos db container. Here you have to map the columns of your source table to the columns in the destination container.
* Once its done you can execute the pipeline.
* Now you can go back to your cosmos db and run a SELECT query in your container and view the data that was transferred.

Challenges Faced:

1. Figuring out the existence of Azure Synapse Link which makes it convenient to access data from Cosmos DB in Synapse Analytics.

2. Roadblock in uploading the data into Cosmos DB and later found the issue was due to a option called Analytical Store that needs to be turned on while creating a container.