**Azure Synapse Coding Challenge**

**Name:** Tejaswini Gokanakonda

**Roll No:**DE142

**Date:** 19-12-2024

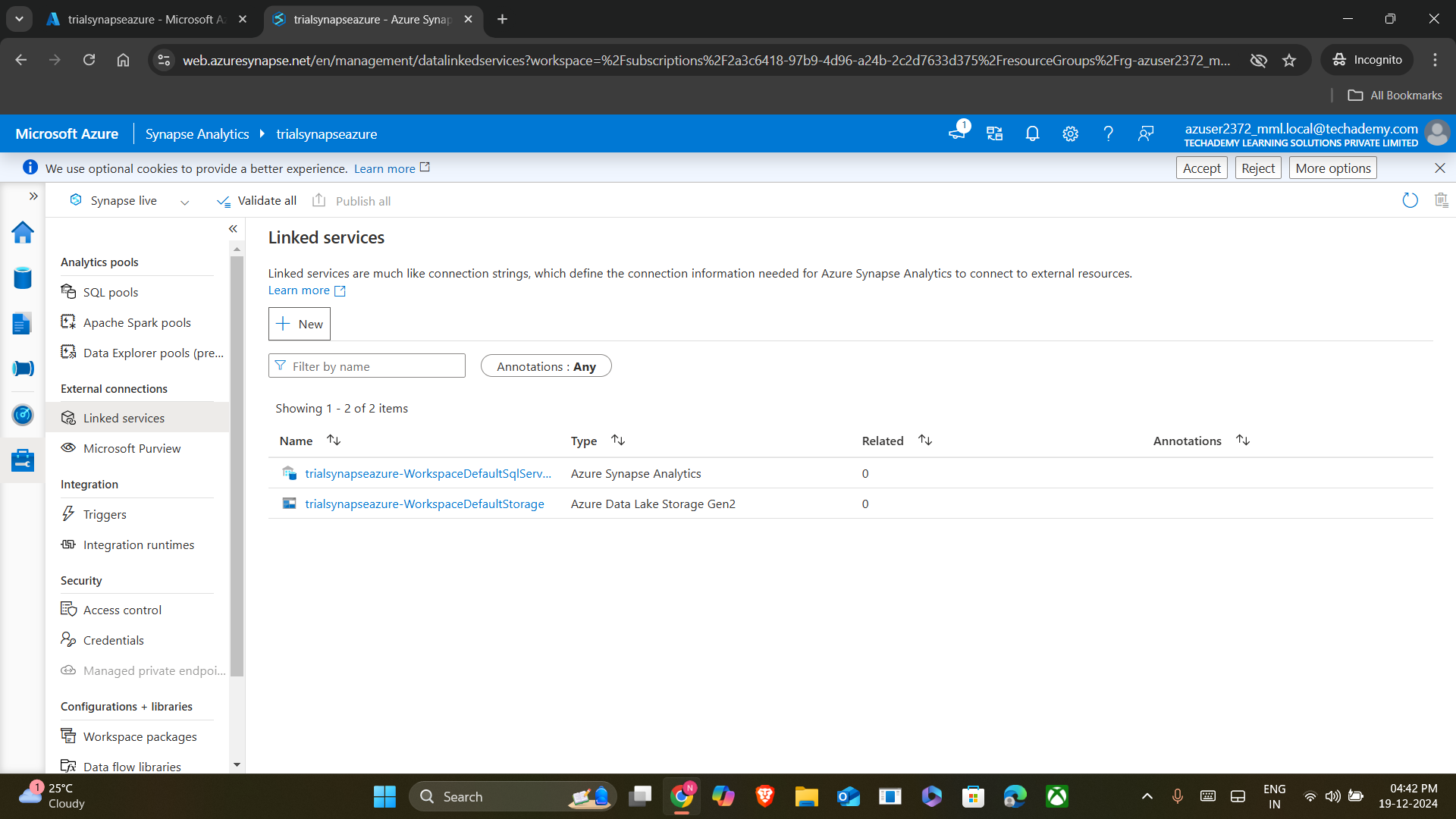
**Build an ETL pipeline with azure synapse with dataflow running on it.  
->Steps to Build an ETL Pipeline with Azure Synapse**

* **Step 1: Set Up Azure Synapse Workspace**

1. **Create Synapse Workspace**:
   * Navigate to the Azure portal.
   * Search for "Azure Synapse Analytics" and create a new workspace.
   * Configure networking, storage account, and firewall rules.
2. **Integrate a Data Lake**:
   * Connect Azure Data Lake Storage Gen2 as your primary data store.
   * Ensure permissions are set correctly for Synapse access.

* **Step 2: Set Up the Linked Service**

1. **Navigate to Manage > Linked Services**:
   * In the Synapse Studio, go to the **Manage** tab.
   * Click on **Linked Services**.



1. **Create a Linked Service for Azure Databricks:**
   * Select **+ New** and choose **Azure Databricks** as the type.
   * Provide the required details:
     + **Cluster details**: Workspace URL, token, and cluster name.
     + **Authentication**: Use a personal access token or Azure Active Directory.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* **Step 3: Create a Pipeline**

1. **Go to Integrate > Pipelines**:
   * In Synapse Studio, go to the **Integrate** tab.
   * Click **+ New pipeline**.
2. **Add a Databricks Notebook Activity**:
   * Drag and drop the **Databricks Notebook** activity from the activity pane into the pipeline canvas.

A screenshot of a computer

Description automatically generated

* **Step 4: Configure the Databricks Notebook Activity**

1. **Link the Notebook**:
   * Click on the **Databricks Notebook** activity and go to the **Settings** tab.
   * Select the Databricks **Linked Service** you created earlier.

A screenshot of a computer

Description automatically generated

1. **Choose the Notebook**:
   * In the **Notebook Path**, browse and select the notebook you want to run.
2. **Provide Notebook Parameters (Optional)**:
   * If your notebook has parameters, pass them in the **Base parameters** section.

A screenshot of a computer

Description automatically generated

* **Step 5: Run the Pipeline**

1. **Validate the Pipeline**:
   * Click on **Validate All** to ensure there are no configuration errors.
2. **Debug or Trigger the Pipeline**:
   * For a quick test, click **Debug** to run the pipeline without publishing it.
   * To run it officially, publish the pipeline and trigger it manually or using a schedule.

A screenshot of a computer

Description automatically generated

* **Step 6: Monitor the Pipeline**

1. **Navigate to Monitor**:
   * Go to the **Monitor** tab in Synapse Studio.
2. **Check Pipeline Runs**:
   * Look for the pipeline you ran in the **Pipeline Runs** section.
   * Check the status of the run (Success, Failed, In Progress).
   * If needed, click on the activity to view the logs and details.

A screenshot of a computer

Description automatically generated

* **Step 7: Analyze Notebook Output**
* Since the notebook doesn’t connect to a sink:
  + Any transformations or output logs are generated and stored within the notebook execution.
  + You can analyze the results either by inspecting the notebook itself or by saving the processed data to a supported storage format (if the notebook writes data).

### **Summary of Steps:**

1. **Set Up Linked Services**
   * Create a secure connection between Azure Synapse and Azure Databricks using Linked Services.
2. **Design and Build the Pipeline**
   * Use the **Integrate** tab in Synapse Studio to create a pipeline and add a Databricks Notebook activity for data transformation tasks.
3. **Configure and Validate**
   * Link the Databricks Notebook activity to your Linked Service, configure parameters, and validate the pipeline to ensure it’s error-free.
4. **Execute the Pipeline**
   * Test the pipeline using the **Debug** option or run it officially by publishing and triggering it.
5. **Monitor and Analyze**
   * Track pipeline runs and activity logs in the **Monitor** tab to ensure smooth execution.
   * Analyze output logs or save processed data to a storage format for further use.