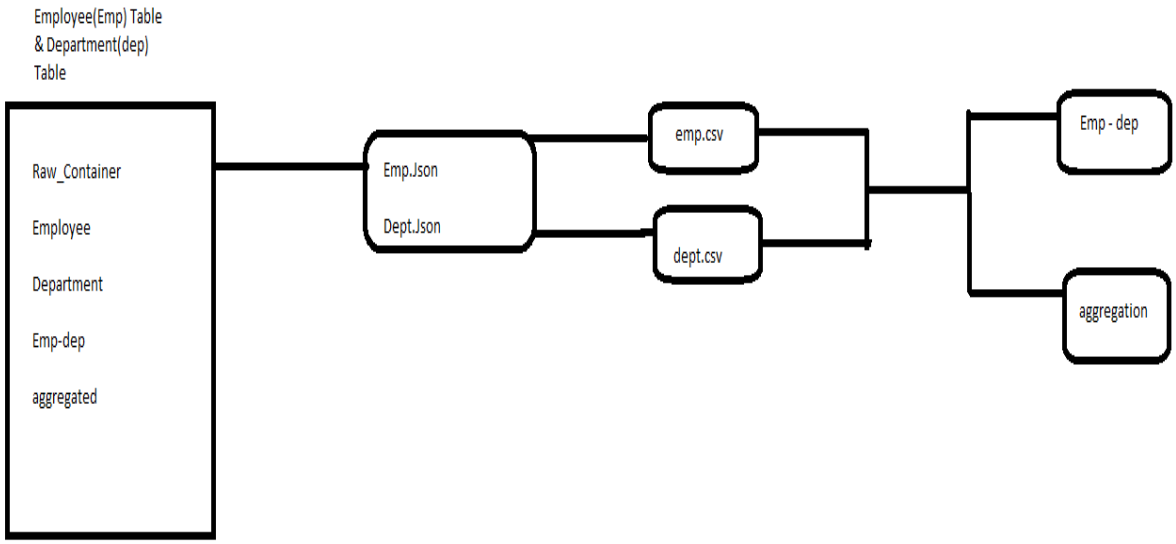
End-to-End ETL with Azure Data Factory and Data Flow

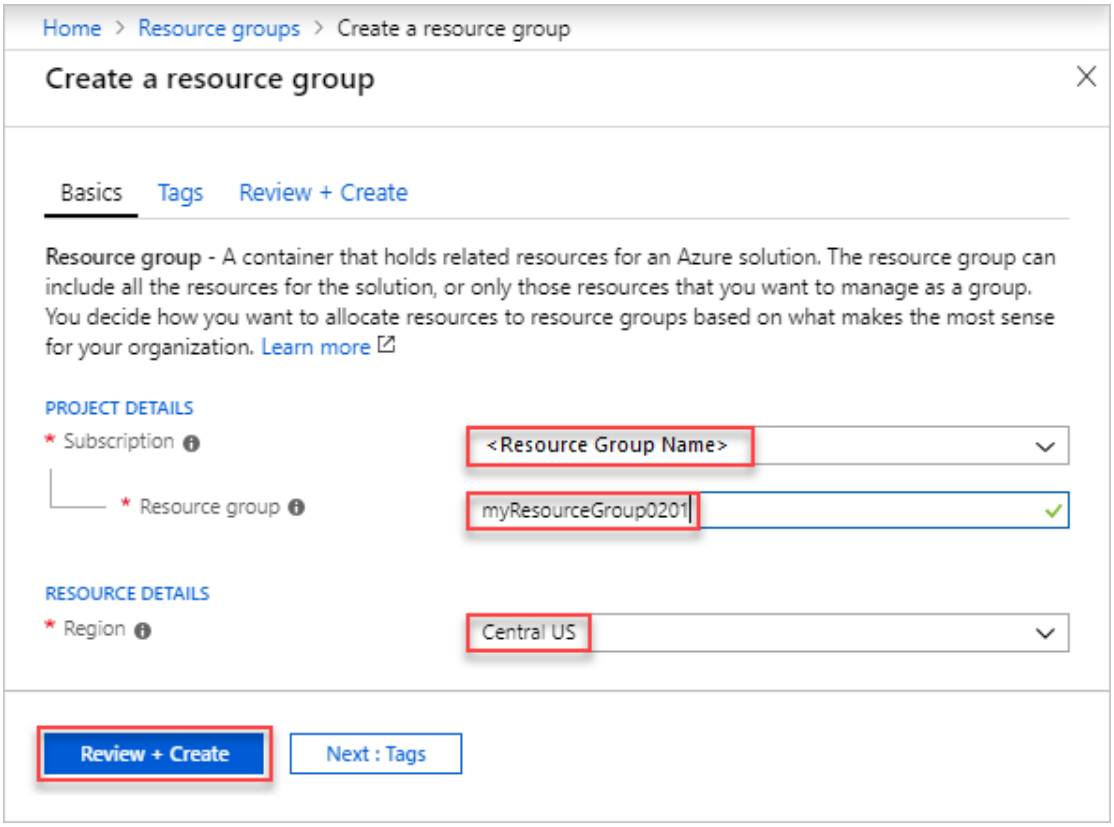
➢ Project OverView



✧ This project showcases a complete Extract, Transform, Load (ETL) pipeline using Azure Data Factory and Azure Data Flow. The pipeline extracts data from two JSON files, converts them to CSV format, and stores them in Azure Blob Storage.

✧ It then utilizes Azure Data Flow to perform data transformations, specifically joining employee and department data, and generating aggregated results.

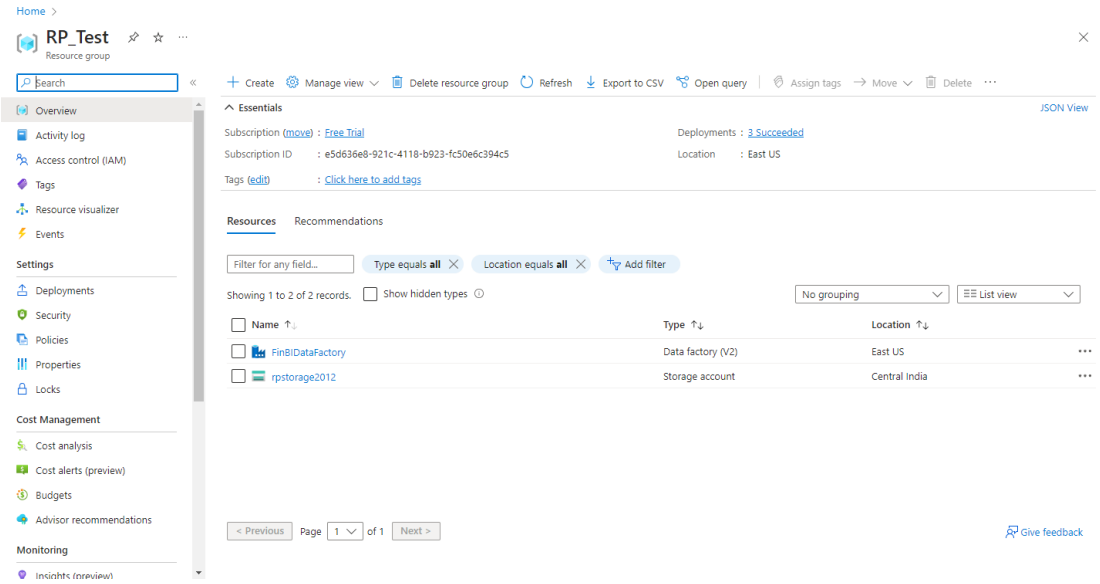
1] Create Resource Groups



- A resource group is a logical container or grouping of related resources within a cloud platform such as Microsoft Azure . It provides a way to manage and organize resources collectively, making it easier to manage, monitor, and control access to those resources.

- A resource group acts as a management unit that holds related resources for a specific application, project, or environment. It allows you to manage and apply policies, permissions, and tags to a set of resources collectively rather than managing them individually.

2] Create Azure Data Factory & Storage Account



➢ Azure Data Factory

Azure Data Factory (ADF) is a cloud-based data integration service provided by Microsoft Azure. It allows you to create, schedule, and manage data pipelines that can ingest, transform, and move data between various on-premises and cloud data sources.

➢ Azure Data Factory Perform -

✓ Data Transformation

✓ Data Movement and Copy

✓ Data Orchestration and Scheduling

✓ Monitoring and Management

✓ Integration with Ecosystem

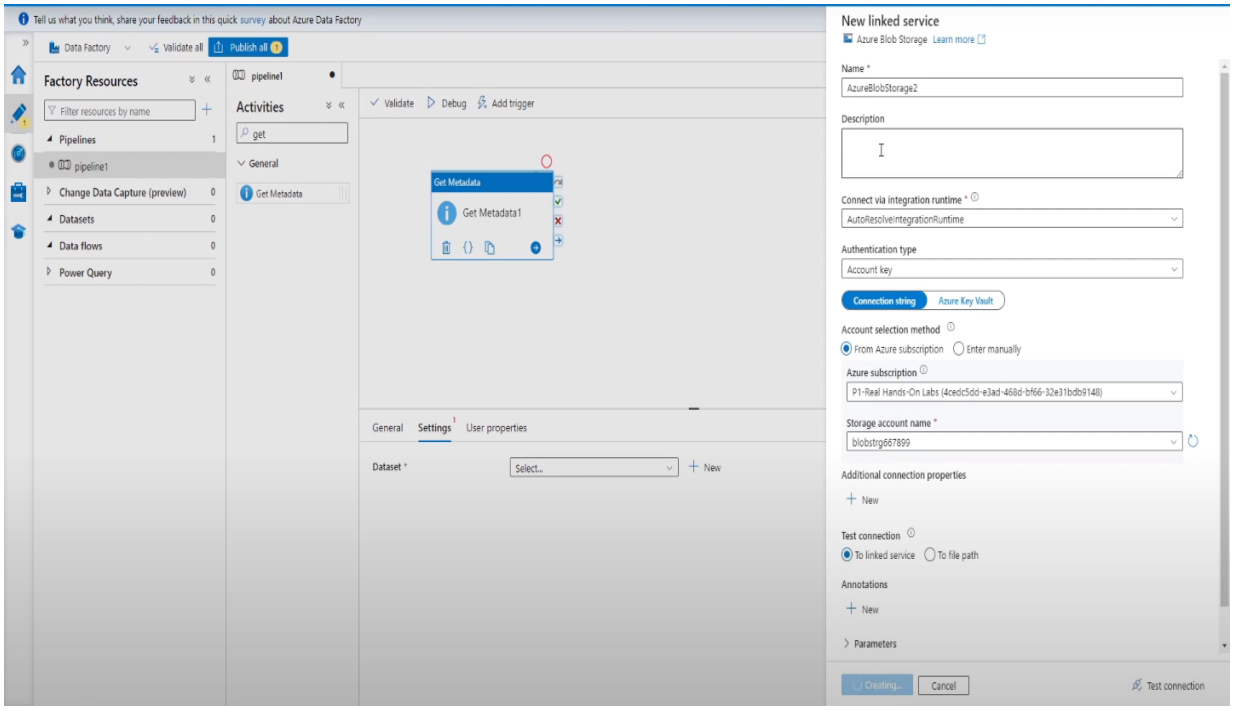
➢ Storage Account

In Azure Data Factory (ADF), a Storage Account is a key component used for storing data during data integration and movement processes. A Storage Account provides a salable and secure storage solution within the Azure ecosystem.

✧ Here's how a Storage Account is utilized in ADF:

- ✓ Source and Destination:
- ✓ Connector:
- ✓ Linked Service:
- ✓ Data Movement:
- ✓ Data Integration:
- ✓ Data Lake Integration:
- ✓ Data Partitioning:

3] Create Linked service



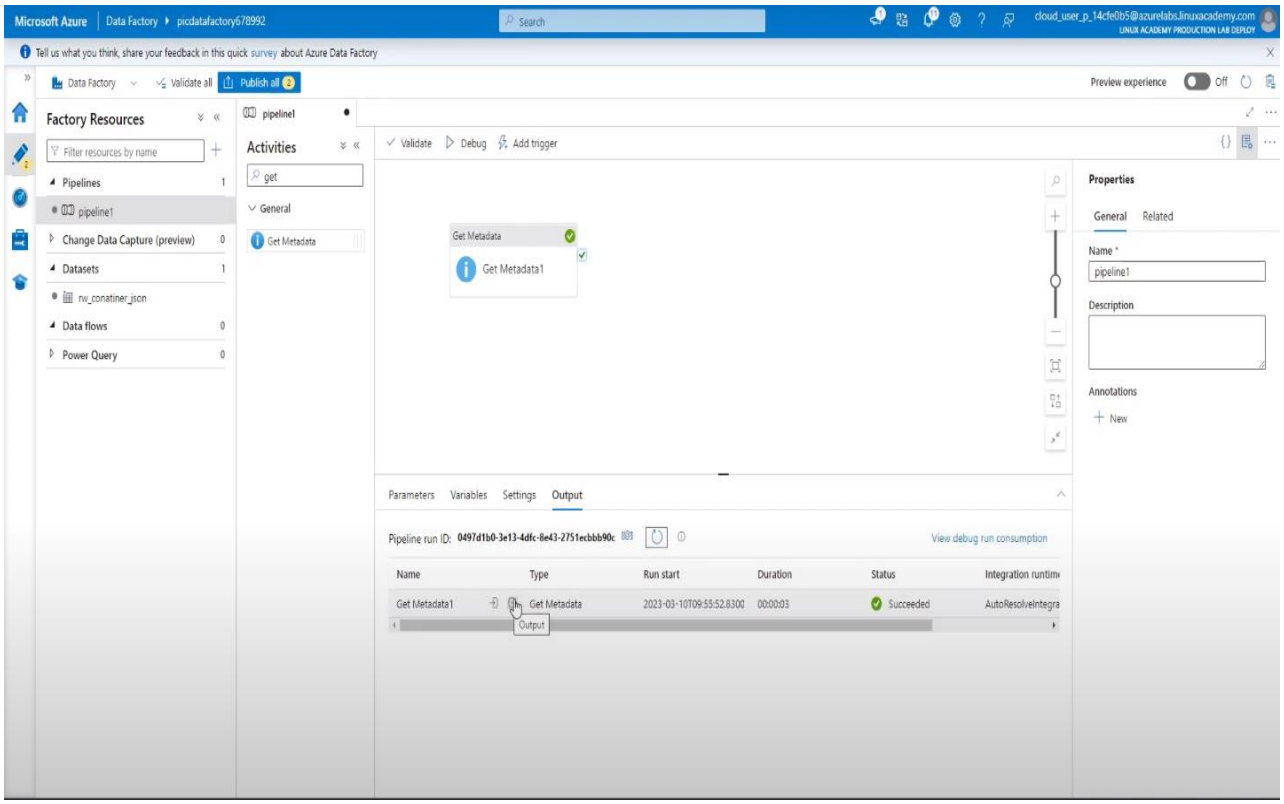
- ✓ In Azure Data Factory (ADF), a linked service is a configuration entity that defines the connection information and credentials required to connect to an external data store or service.

- ✓ It acts as a bridge between ADF and the data source, enabling data movement and transformation activities within ADF pipelines.

- ✓ Specifically, when using the GetMetadata activity in an ADF pipeline, a linked service is used to establish the connection and retrieve metadata about the specified data source.

- ✓ The GetMetadata activity is used to retrieve information about files, folders, tables, or other objects in the data source without actually moving or transforming the data.

3] Run the Get metadata activity



✓ Create a pipeline in Azure Data Factory.

✓ Within the pipeline, add a Get Metadata activity.

✓ Configure the Get Metadata activity as follows:

- Name: Provide a meaningful name for the activity.

- Linked Service: Select or create a linked service that represents the connection to your data source.

- Dataset: Select or create a dataset that represents the specific object or path for which you want to retrieve metadata.

- Field List: Specify the fields or properties you want to retrieve metadata for. You can choose to retrieve all fields or select specific ones based on your requirements.

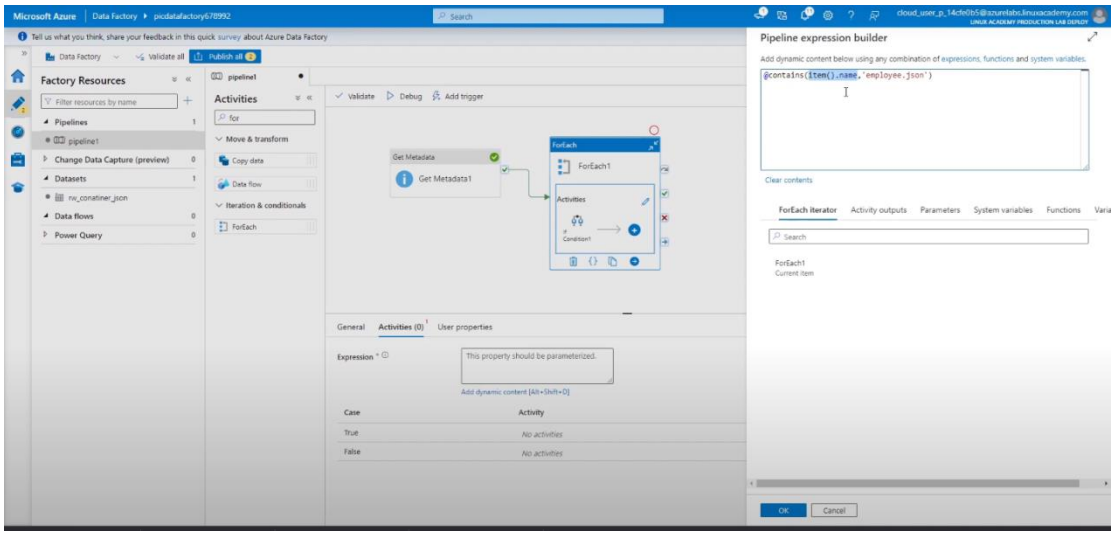
- **Save and publish the pipeline.**

- ✓ Trigger the pipeline manually or schedule it to run at specific intervals.

- ✓ Monitor the pipeline run to view the status and progress of the Get Metadata activity.

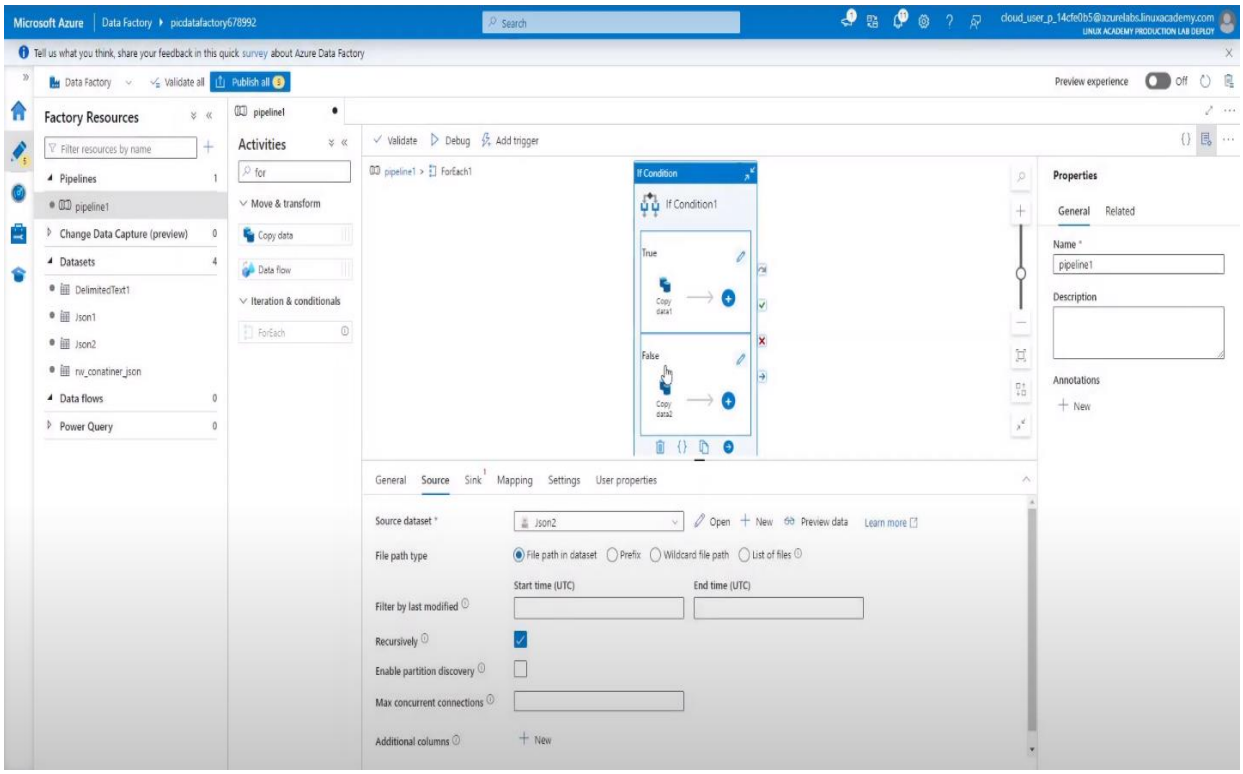
- ✓ Once the pipeline run completes, you can access the retrieved metadata from the Get Metadata activity's output. This metadata can be used in subsequent activities or stored for further analysis.

4] Create For Each Activity



- ✓ Create Linked Services: Set up the linked services for the source and destination data stores.
- ✓ Define Dataset: Define the dataset that represents the source data for the Foreach loop.
- ✓ Define Foreach Activity: Create a Foreach activity in your pipeline and configure it to iterate over the source dataset.
- ✓ Define If Condition: Within the Foreach activity, add an If Condition activity to evaluate a specific condition for each item in the loop.
- ✓ Define Copy Data Activities: Depending on the result of the If Condition, add a Copy Data activity within the "If true" branch to copy the data to the desired destination. Similarly, add another Copy Data activity within the "If false" branch to handle the alternative destination.
- ✓ Complete the Pipeline: Connect the activities and define the desired control flow within the pipeline.

5] Set The For-each Activity If Condition



1. Create a Pipeline: Start by creating a new pipeline in your Azure Data Factory instance. Give it a meaningful name.

2. Add a Foreach Activity: Within the pipeline, add a Foreach activity. Configure the settings of the Foreach activity as follows:

- Set the `Items` property to specify the items you want to iterate over. This can be an array, a dynamic expression, or a dataset that provides the list of items.

- Configure any additional settings, such as maximum concurrency or batch count, based on your requirements.

3. Add an If Condition Activity: Inside the Foreach activity, add an If Condition activity. Configure the If Condition activity as follows:

- Specify the condition you want to evaluate. For example, you can compare a property or value from the current item in the loop using expressions.

- Connect the output of the Foreach activity to the If Condition activity.

4. Add Activities for True and False Branches:

- Add a Copy Data activity within the "If true" branch of the If Condition activity.
- Configure the Copy Data activity to specify the source and destination datasets, mappings, and any required transformations or settings for data movement.
- Connect the "If true" branch of the If Condition activity to the Copy Data activity.

5. Add Activities for False Branch:

- Add another Copy Data activity within the "If false" branch of the If Condition activity.

- Configure the Copy Data activity for the alternative destination, specifying the source and destination datasets, mappings, and any required transformations or settings.

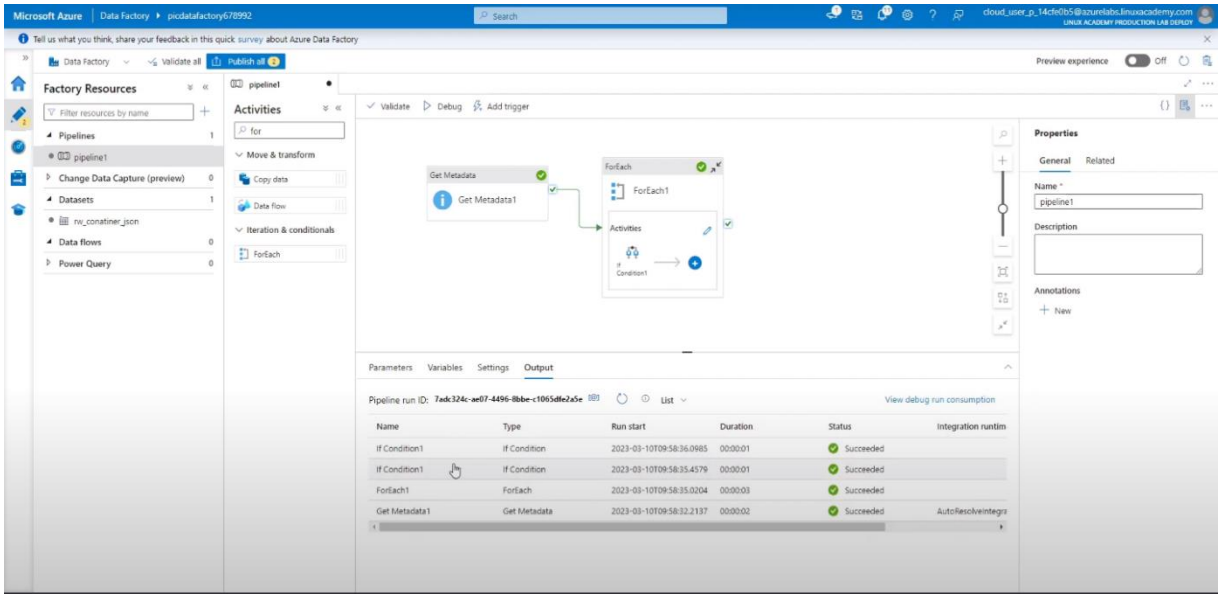
- Connect the "If false" branch of the If Condition activity to the second Copy Data activity.

6. Complete the Pipeline: Connect any remaining activities or define additional control flow logic as needed.

7. Publish and Trigger the Pipeline: Once you have completed the pipeline, publish it to save the changes. You can then trigger the pipeline manually or set up a schedule or trigger based on your requirements.

8. Monitor and Validate: Monitor the pipeline runs in Azure Data Factory to ensure that the Foreach activity is executing as expected. Check the output and logs of the Copy Data activities to validate the data movement and conditional branching logic.

6] Run the Activity



✓ Create a Pipeline: Start by creating a new pipeline in your Azure Data Factory instance. Give it a meaningful name that reflects its purpose.

✓ Add a Foreach Activity: Within the pipeline, add a Foreach activity. Configure the settings of the Foreach activity as follows:

Items: Specify the items that you want to iterate over. This could be an array of values, a dynamic expression, or a dataset that provides the list of items.

Settings: Configure any additional settings for the Foreach activity, such as the maximum concurrency or the batch count for parallel execution.

✓ Add Activities within the Foreach Loop: Inside the Foreach activity, you can add different activities to perform actions on each item in the loop. For this example, we will add an If Condition activity.

✓ Add If Condition Activity: Configure the If Condition activity within the Foreach loop. Set the condition based on your requirements. For example, you can compare a specific property or value from the current item in the loop.

- ✓ Add Activities for True and False Branches: Based on the evaluation result of the If Condition, add the desired activities within the "If true" and "If false" branches. In this case, add Copy Data activities for each branch.

- ✓ Configure Copy Data Activities: Configure the Copy Data activities within the respective branches of the If Condition. Specify the source and destination datasets, mappings, and any transformations or settings required for the data movement.

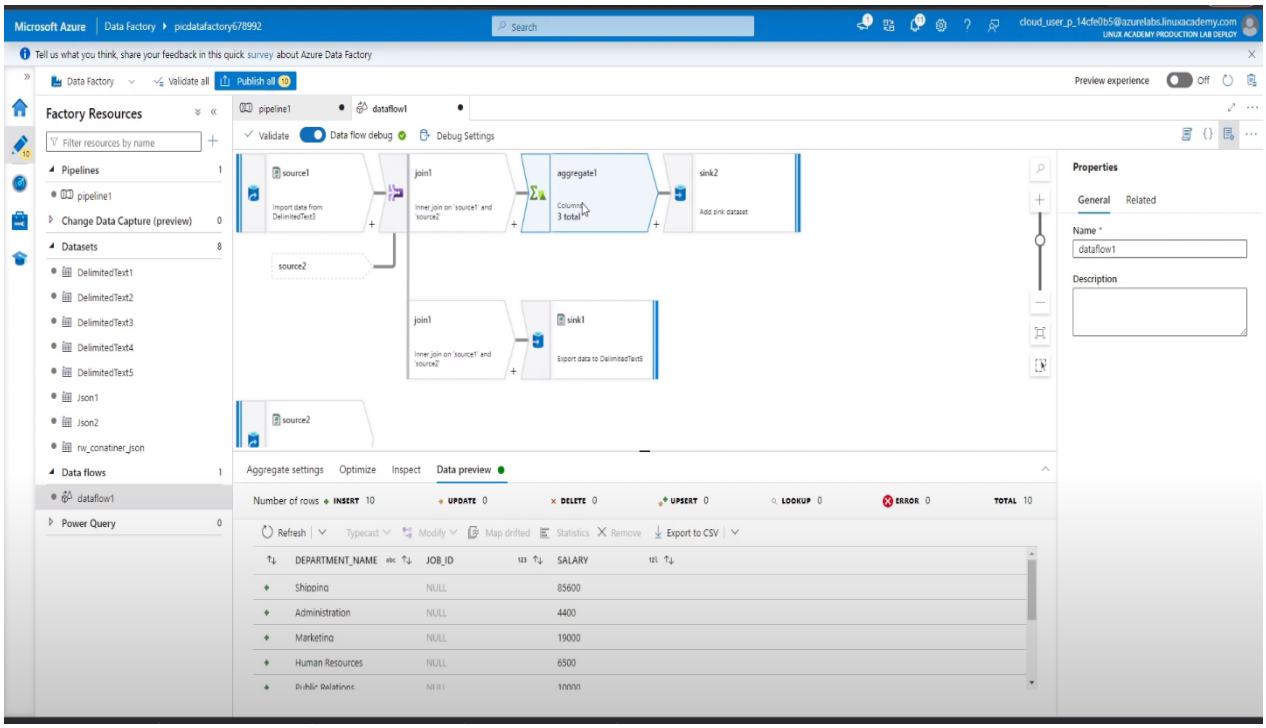
- ✓ Complete the Pipeline: Connect the activities in the pipeline to define the desired control flow. You can add additional activities before or after the Foreach activity as needed.

- ✓ Publish and Trigger the Pipeline: Once you have completed the pipeline, publish it to save the changes. You can then trigger the pipeline manually or set up a schedule or trigger based on your requirements.

- ✓ Monitor and Validate: Monitor the pipeline runs in Azure Data Factory to ensure that the Foreach activity is executing as expected. Check the output and logs of the Copy Data activities to validate the data movement and the conditional branching logic.

- ✓ By following these steps, you can run the Foreach activity within your pipeline in Azure Data Factory, iterate over a list of items, and perform different actions based on a conditional branching condition.

7] Create A Data Flow



To join two files in a data flow in Azure Data Factory, you can utilize the Join transformation along with the appropriate join conditions. Here's how you can perform two separate joins in a data flow, one for employee and department data, and another for aggregation:

1. Create Source Datasets: Begin by creating two source datasets, one for the employee data and another for the department data. Configure the datasets to connect to the appropriate file sources.

2. Create a Data Flow: Create a new data flow within your Azure Data Factory instance.

3. Add Source transformations: Add the source transformations for the employee data and department data. Connect each source transformation to the respective source dataset.

4. Apply Join for Employee and Department: Add a Join transformation to the data flow canvas. Configure the Join transformation to join the employee data and department data based on the appropriate join conditions, such as a common key column.

5. Define Join Mapping: Specify the mapping for the join operation, mapping the columns from both sources that you want to include in the joined output.

6. Add Aggregation Transformation: Add an Aggregation transformation to the data flow canvas. Connect it to the output of the Join transformation.

7. Configure Aggregation: Configure the Aggregation transformation to perform the desired aggregations based on your requirements, such as calculating the average, sum, count, or any other aggregation functions.

8. Connect Output Destination: Connect the output of the Aggregation transformation to the desired destination, such as a sink dataset or file.

9. Configure Output Mapping: Define the output mapping, specifying which columns from the Aggregation transformation should be written to the output destination.

10. Validate and Publish: Validate the data flow to ensure it is configured correctly. Once validated, publish the data flow to make it available for use in your pipelines.

Thank You