**Madiha Aimon Tappal**

[madihaaimon@gmail.com](mailto:madihaaimon@gmail.com)

**Data Engineering Batch – 1**

**Day – 22 Assignment**

**Azure Databricks**

**What is Azure Data Factory?**

Azure Data Factory is a cloud-based data integration service that allows you to create data-driven workflows in the cloud for orchestrating and automating data movement and data transformation.

ADF does not store any data itself. It allows you to create data-driven workflows to orchestrate the movement of data between supported data stores and then process the data using compute services in other regions or in an on-premise environment. It also allows you to monitor and manage workflows using both programmatic and UI mechanisms.

**Azure Data Factory use cases**

ADF can be used for:

* Supporting data migrations
* Getting data from a client’s server or online data to an Azure Data Lake
* Carrying out various data integration processes
* Integrating data from different ERP systems and loading it into Azure Synapse for reporting

**How does Azure Data Factory work?**

The Data Factory service allows you to create data pipelines that move and transform data and then run the pipelines on a specified schedule (hourly, daily, weekly, etc.). This means the data that is consumed and produced by workflows is time-sliced data, and we can specify the pipeline mode as scheduled (once a day) or one time.

Azure Data Factory pipelines (data-driven workflows) typically perform three steps.

**Step 1: Connect and Collect**

Connect to all the required sources of data and processing such as SaaS services, file shares, FTP, and web services. Then,  move the data as needed to a centralized location for subsequent processing by using the Copy Activity in a data pipeline to move data from both on-premise and cloud source data stores to a centralization data store in the cloud for further analysis.

**Step 2: Transform and Enrich**

Once data is present in a centralized data store in the cloud, it is transformed using compute services such as HDInsight Hadoop, Spark, Azure Data Lake Analytics, and Machine Learning.

**Step 3: Publish**

Deliver transformed data from the cloud to on-premise sources like SQL Server or keep it in your cloud storage sources for consumption by BI and analytics tools and other applications.

**Data migration activities with Azure Data Factory**

By using Microsoft Azure Data Factory, data migration occurs between two cloud data stores and between an on-premise data store and a cloud data store.

*Copy Activity* in Azure Data Factory copies data from a source data store to a sink data store. Azure supports various data stores such as source or sink data stores like Azure Blob storage, Azure Cosmos DB (DocumentDB API), Azure Data Lake Store, Oracle, Cassandra, etc. For more information about Azure Data Factory supported data stores for data movement activities, refer to Azure documentation for data movement activities.

Azure Data Factory supports transformation activities such as Hive, MapReduce, Spark, etc that can be added to pipelines either individually or chained with other activities. For more information about ADF-supported data stores for data transformation activities, refer to the following Azure Data Factory documentation: Transform data in Azure Data Factory.

If you want to move data to/from a data store that Copy Activity doesn’t support, you should use a .NET custom activity in Azure Data Factory with your own logic for copying/moving data. To learn more about creating and using a custom activity, check the Azure documentation and see “Use custom activities in an Azure Data Factory pipeline”.

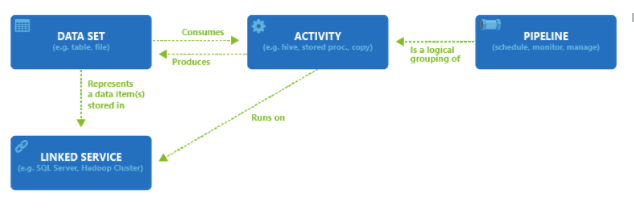
**Azure Data Factory key components**

Azure Data Factory has four key components that work together to define input and output data, processing events, and the schedule and resources required to execute the desired data flow:

* **Datasets represent data structures within the data stores.** An input dataset represents the input for an activity in the pipeline. An output dataset represents the output for the activity. For example, an Azure Blob dataset specifies the blob container and folder in the Azure Blob Storage from which the pipeline should read the data. Or, an Azure SQL Table dataset specifies the table to which the output data is written by the activity.
* **A pipeline is a group of activities.** They are used to group activities into a unit that together performs a task. A data factory may have one or more pipelines. For example, a pipeline could contain a group of activities that ingests data from an Azure blob and then runs a Hive query on an HDInsight cluster to partition the data.
* **Activities define the actions to perform on your data.** Currently, Azure Data Factory supports two types of activities: data movement and data transformation.
* **Linked services define the information needed for Azure Data Factory to connect to external resources.** For example, an Azure Storage linked service specifies a connection string to connect to the Azure Storage account.

**How the Azure Data Factory components work together**

The following schema shows us the relationships between the **Dataset**, **Activity**, **Pipeline**, and **Linked Services** components:



**Azure Data Factory access zones**

Currently, you can create data factories in the West US, East US, and North Europe regions. However, a data factory can access data stores and compute services in other Azure regions to move data between data stores or process data using compute services.  
For example, let’s say that your compute environments such as Azure HDInsight cluster and Azure Machine Learning are running out of the West Europe region. You can create and use an Azure Data Factory instance in North Europe and use it to schedule jobs on your compute environments in West Europe. It takes a few milliseconds for Data Factory to trigger the job on your compute environment but the time for running the job on your computing environment does not change.

You can use one of the following tools or APIs to create data pipelines in Azure Data Factory:

* Azure portal
* Visual Studio
* PowerShell
* .NET API
* REST API
* Azure Resource Manager template

**Data Migration in action**

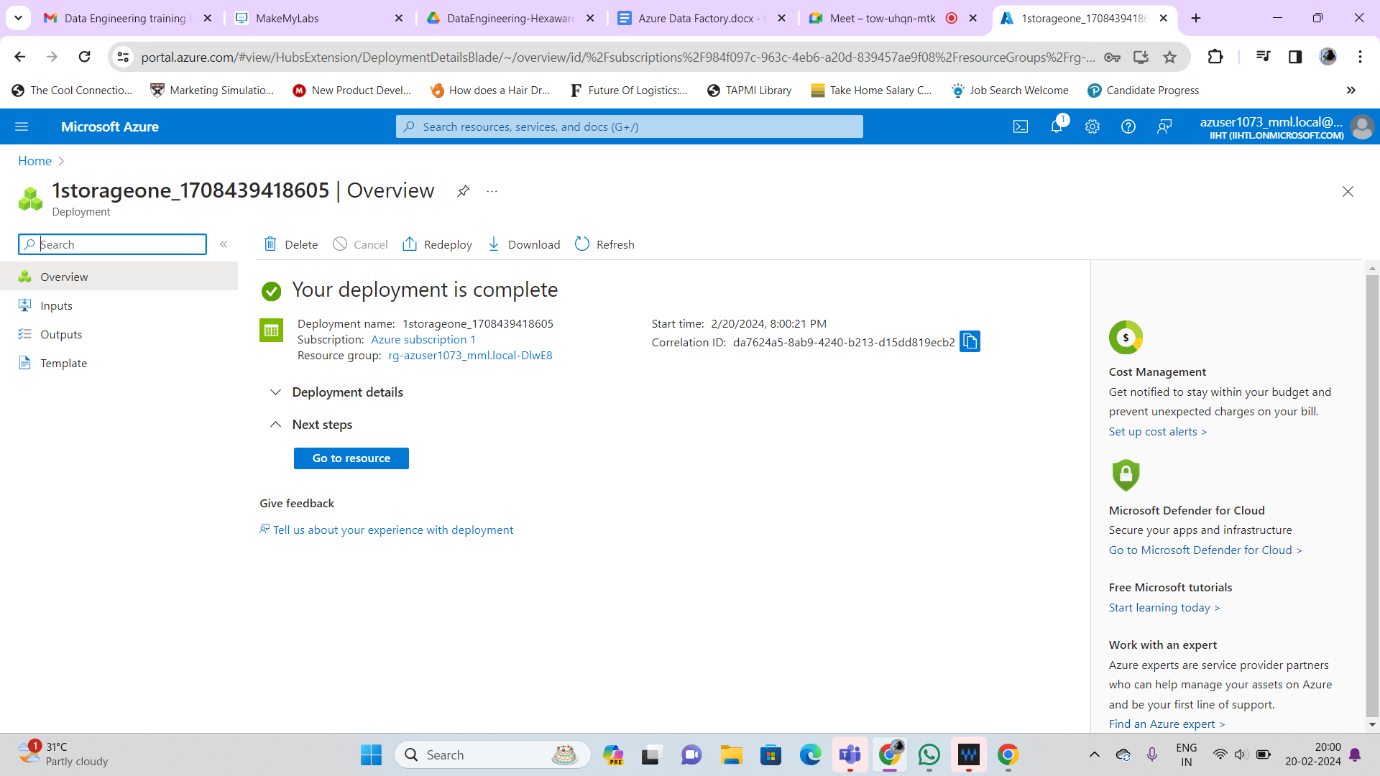
To get started with Data Factory, you should create a Data Factory on Azure, then create the four key components with Azure Portal, Virtual Studio, or PowerShell etc. Since the four components are in editable JSON format, you can also deploy them in a whole ARM template on the fly.

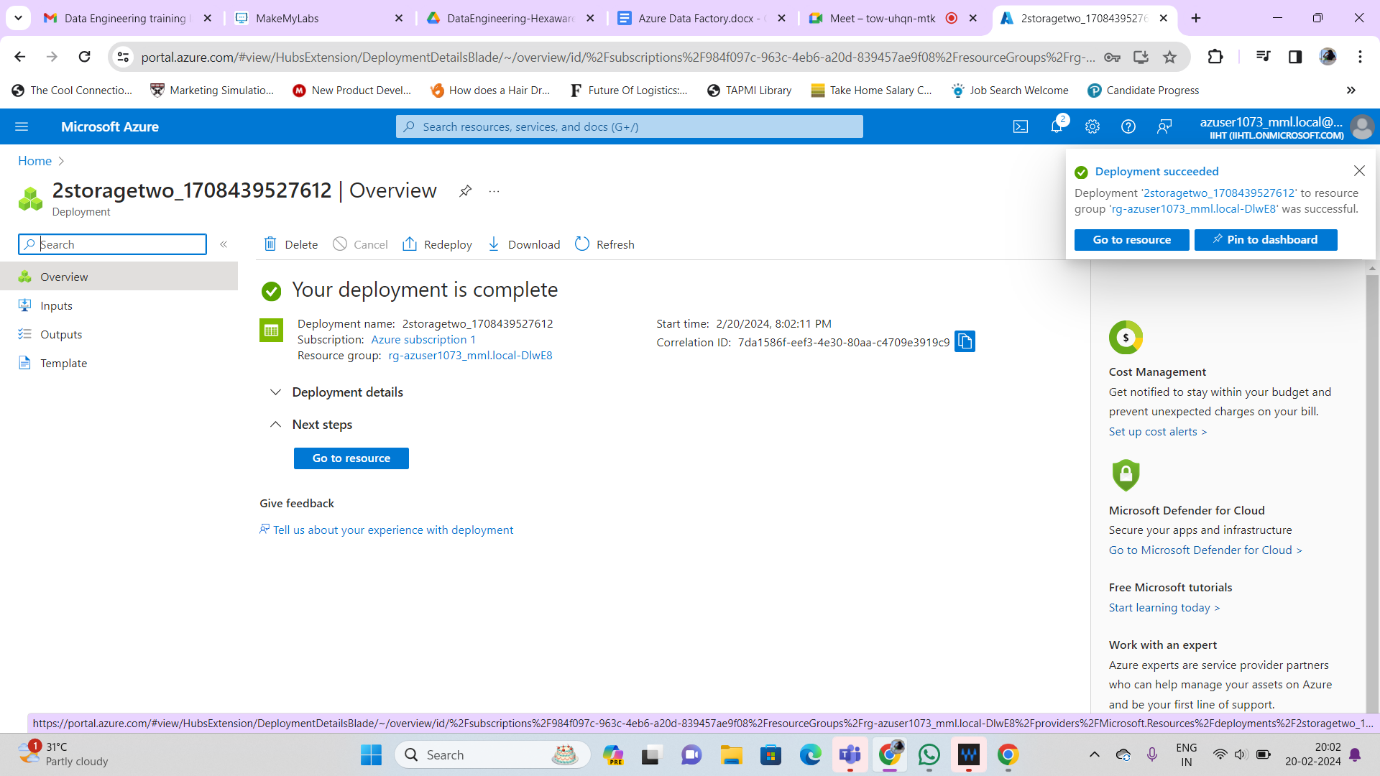
**DataCopy Wizard on Azure**

To start migrating the data on Blob storage to Azure SQL, the most simple way is to use Data Copy Wizard, which is currently in preview. It allows you to quickly create a data pipeline that copies data from a supported source data store to a supported destination data store. For more information on creating your migration related components with Data Copy Wizard, refer to the Microsoft tutorial: Create a pipeline with Copy Activity using Data Factory Copy Wizard.

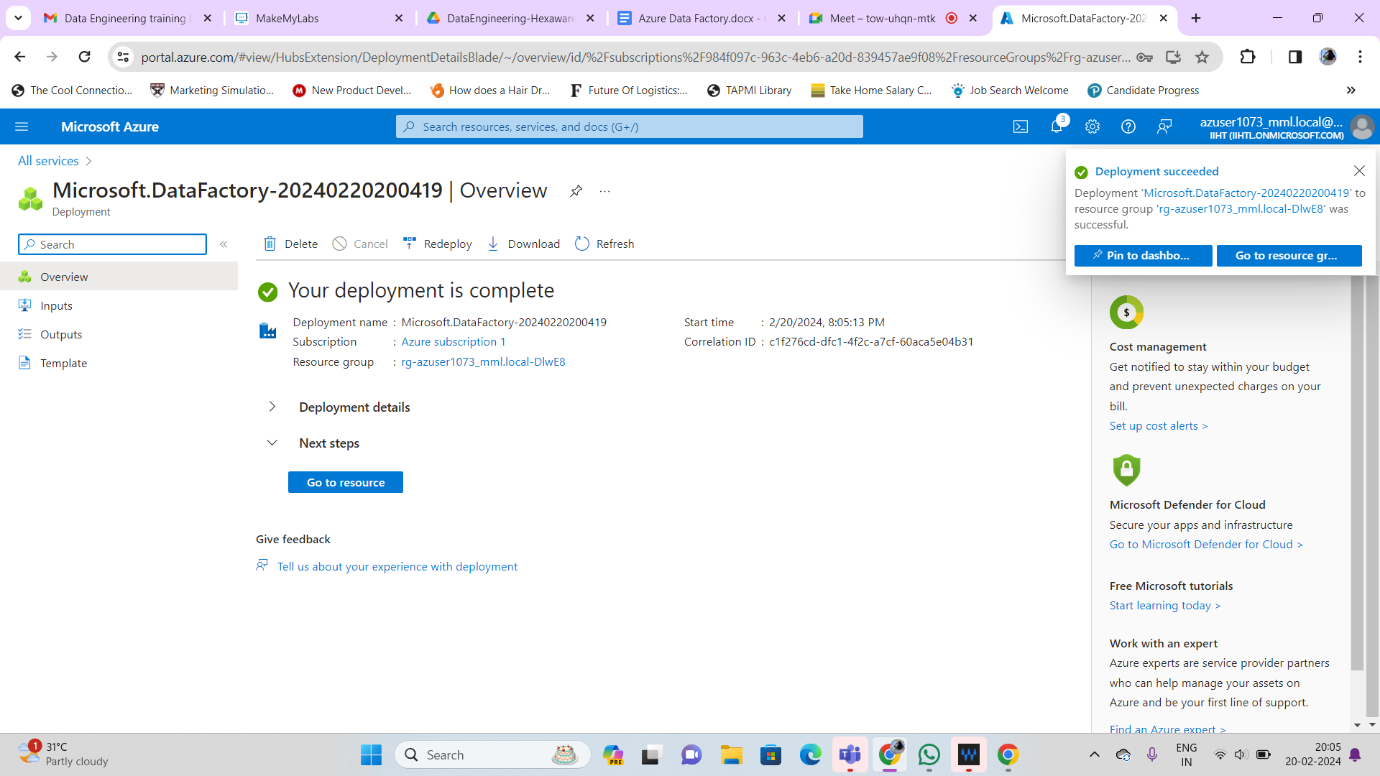
**Hands On Practice:**

**Creating Storage accounts :**

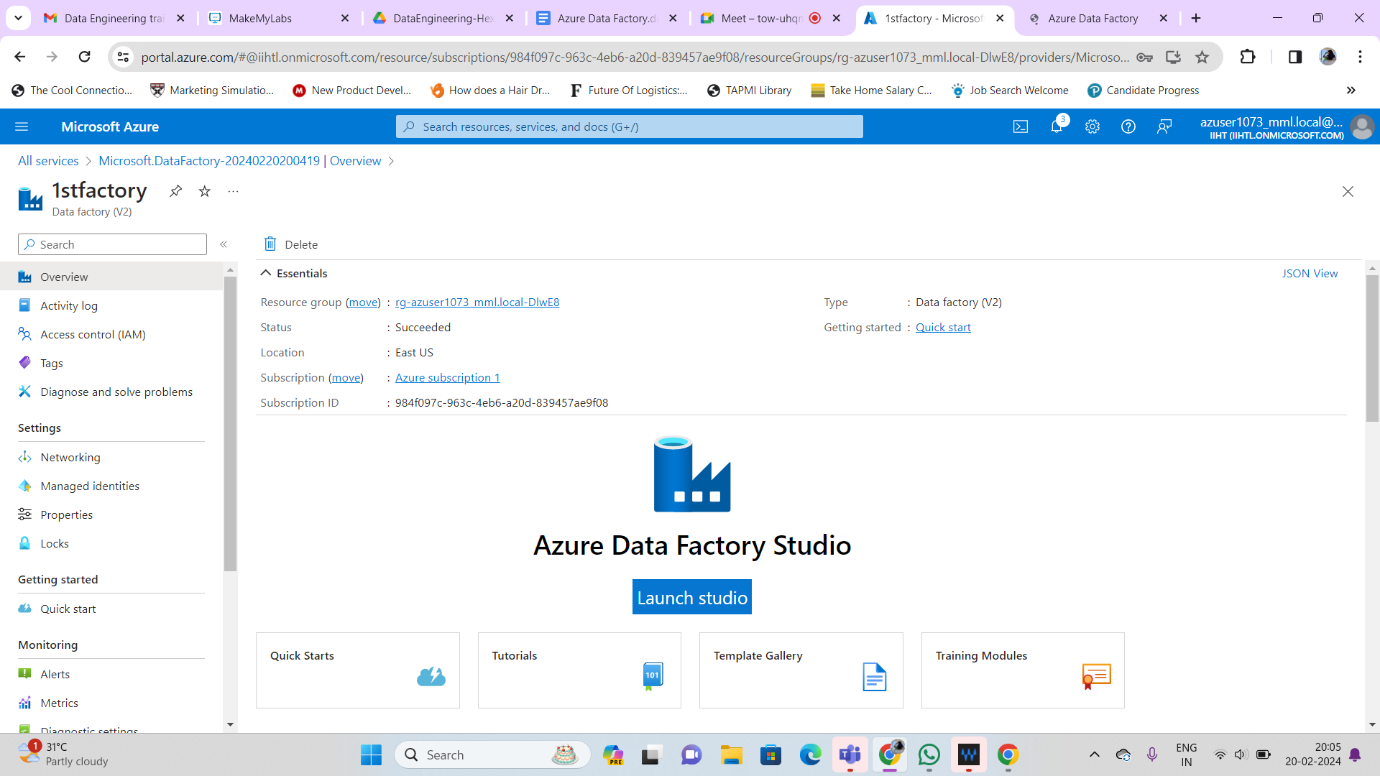
****

****

**Creating account in datafactory :**

****

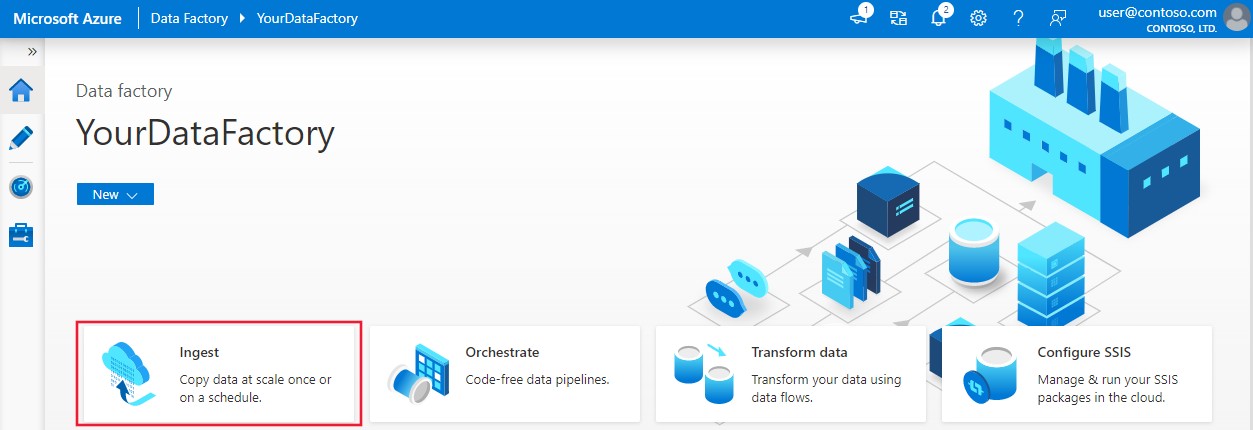
**Launch studio :**

****

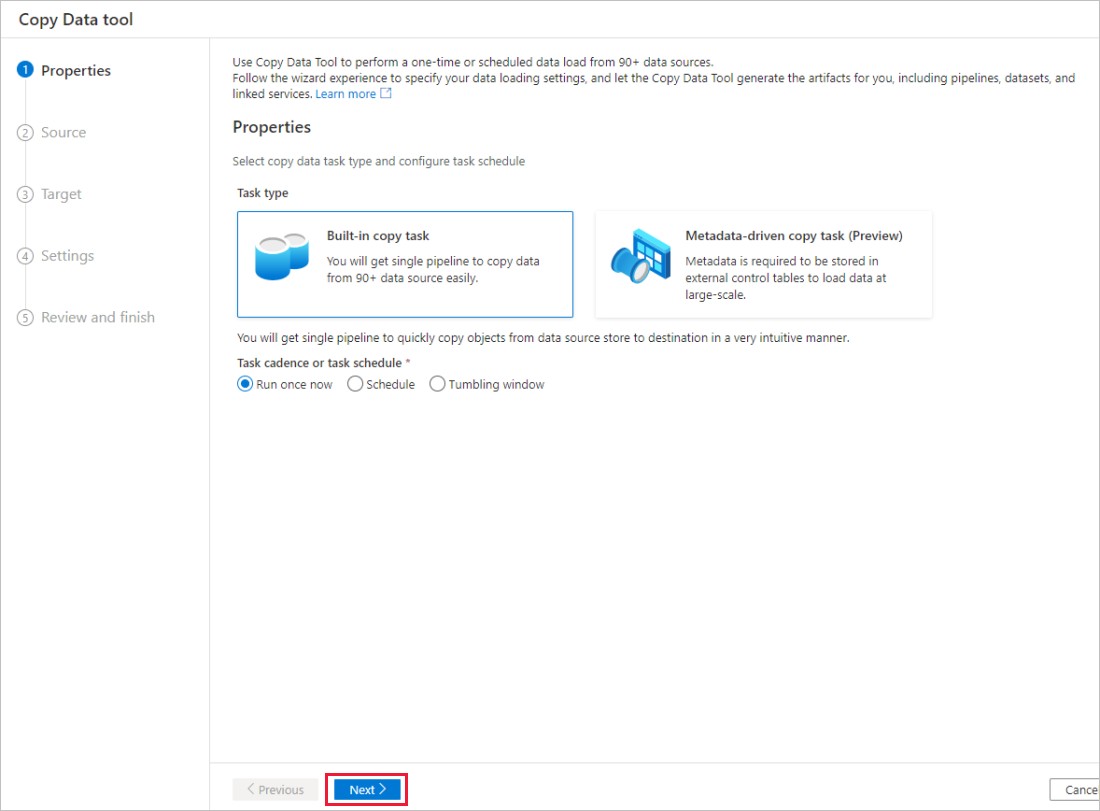
**Create a data factory**

You can use your existing data factory or create a new one as described in Quickstart: Create a data factory by using the Azure portal. Use the copy data tool to copy data The steps below will walk you through how to easily copy data with the copy data tool in Azure Data Factory.

Step 1: Start the copy data Tool 1. On the home page of Azure Data Factory, select the Ingest tile to start the Copy Data tool.

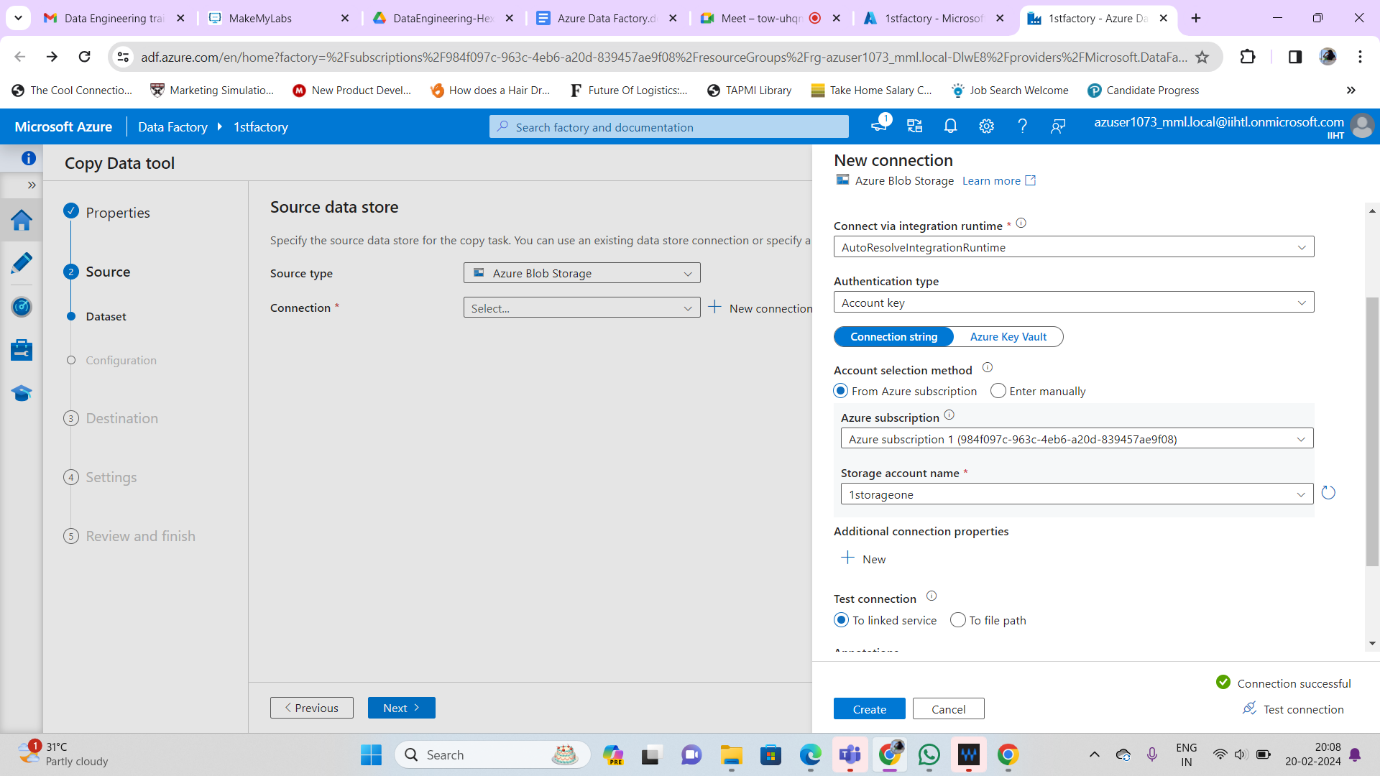


On the **Properties** page of the Copy Data tool, choose **Built-in copy task** under **Task type**, then select **Next**.



# Step 2: Complete source conﬁguration

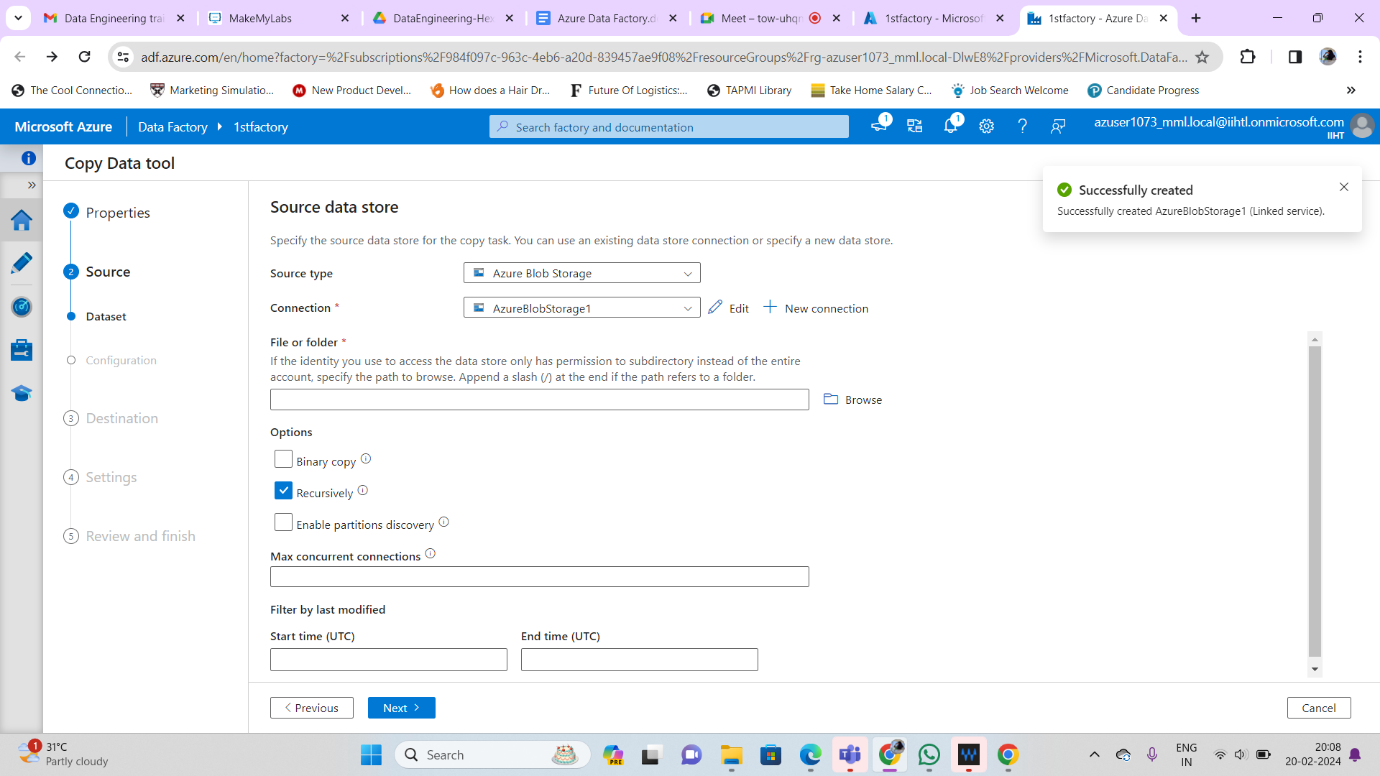
1. Click **+ Create new connection** to add a connection.
2. Select the linked service type that you want to create for the source connection. In this tutorial, we use **Azure Blob Storage**. Select it from the gallery, and then select **Continue**.

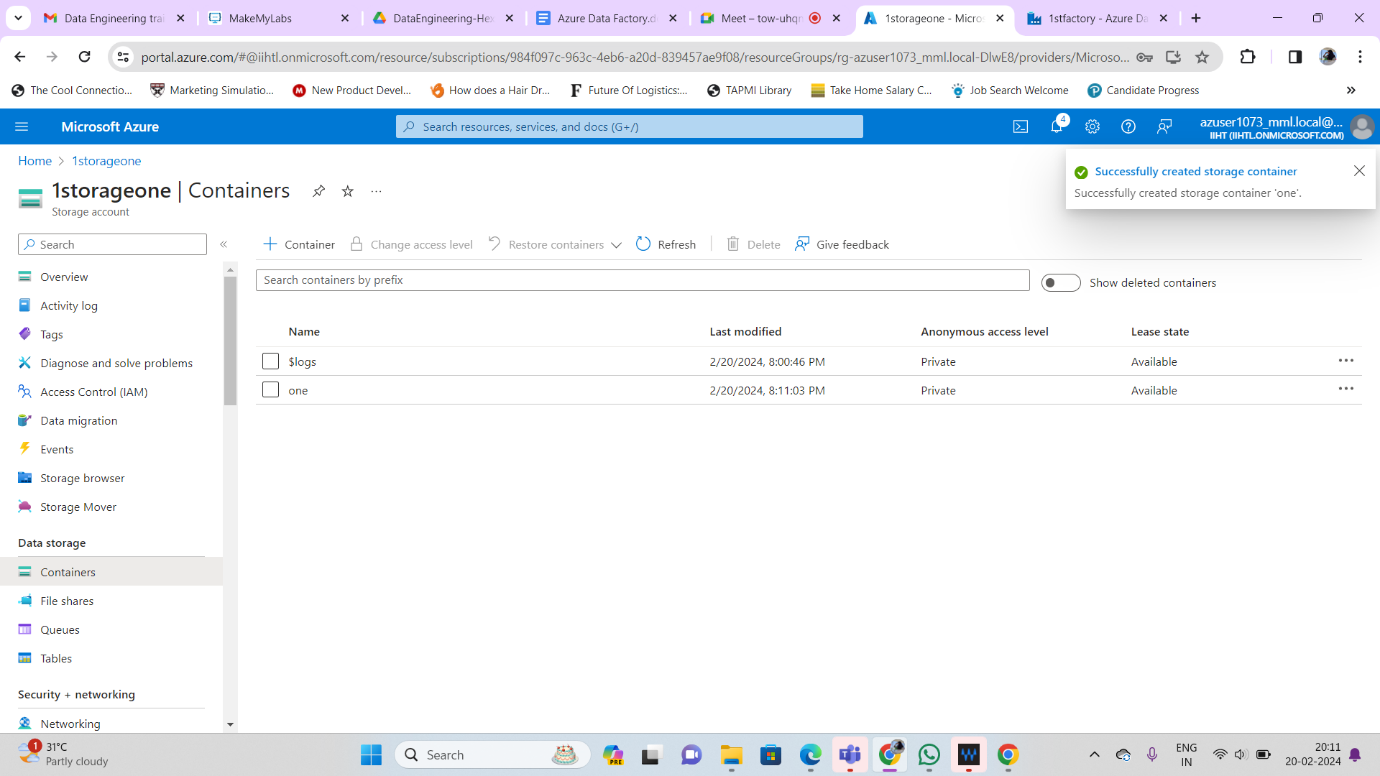
****

Select the newly created connection in the **Connection** block.

In the **File or folder** section, select **Browse** to navigate to the **adftutorial/input** folder, select the **emp.txt** ﬁle, and then click **OK**.

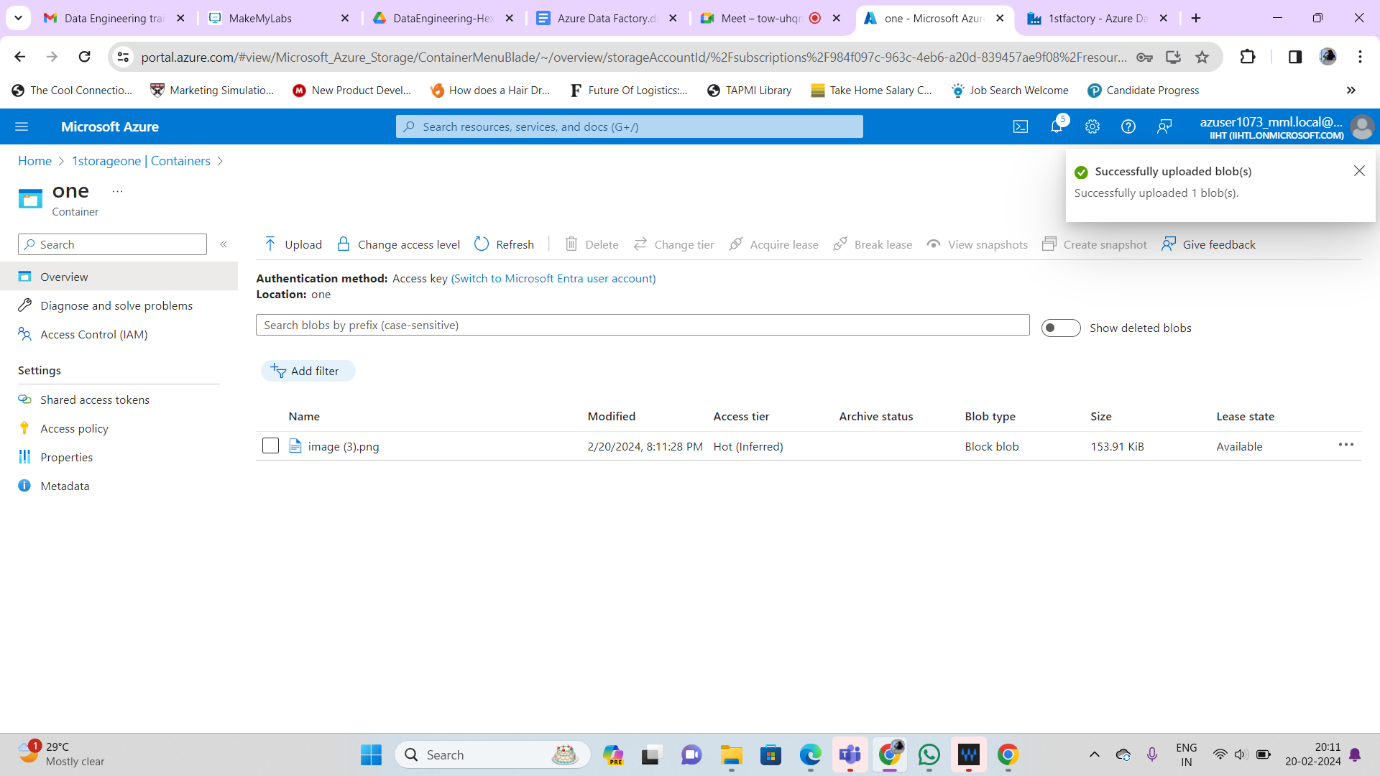
Select the **Binary copy** checkbox to copy ﬁle as-is, and then select **Next**.

****

****

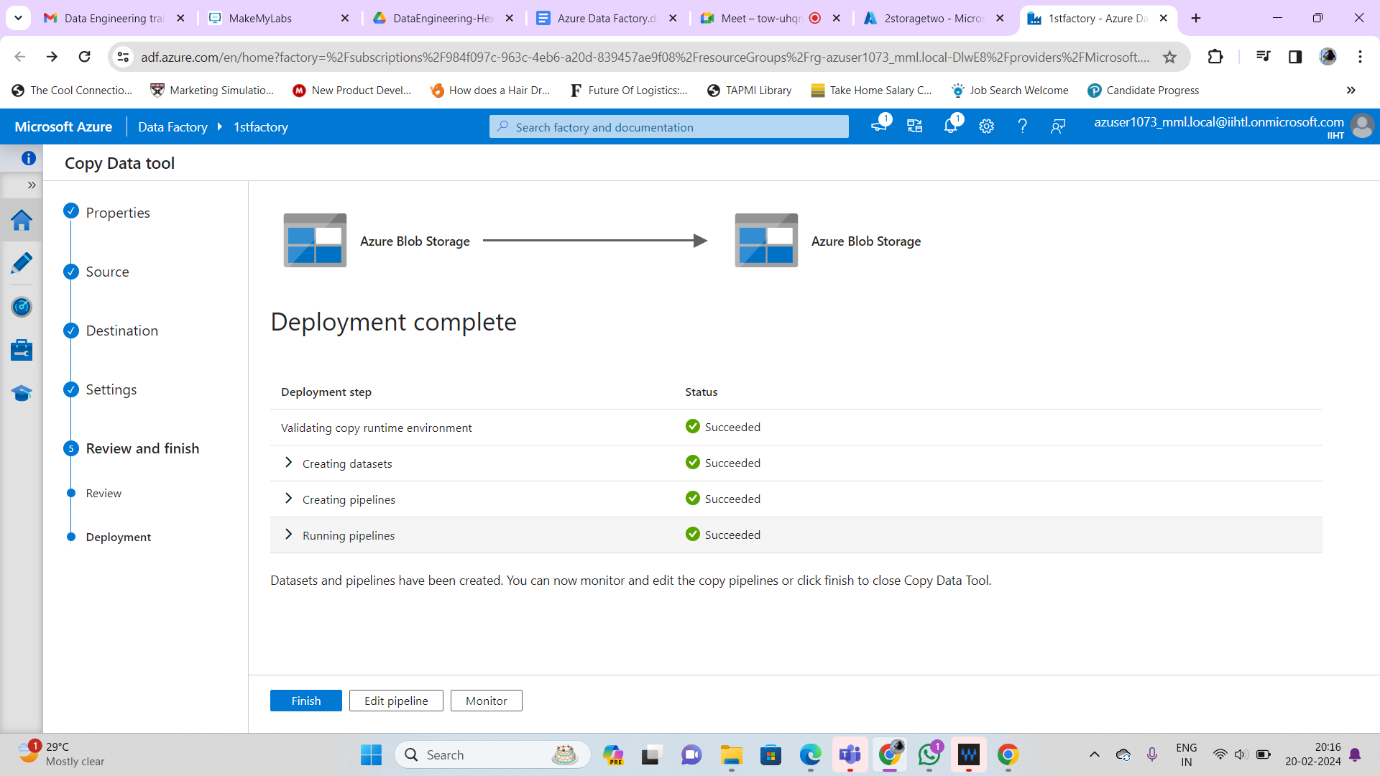
# Step 3: Complete destination conﬁguration

1. Select the **AzureBlobStorage** connection that you created in the **Connection** block.
2. In the **Folder path** section, enter **adftutorial/output** for the folder path.

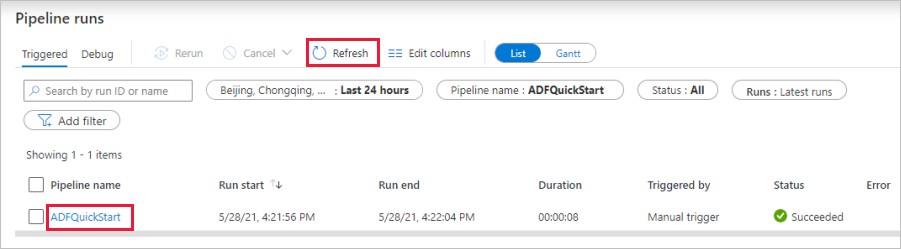
****

# Step 4: Review all settings and deployment

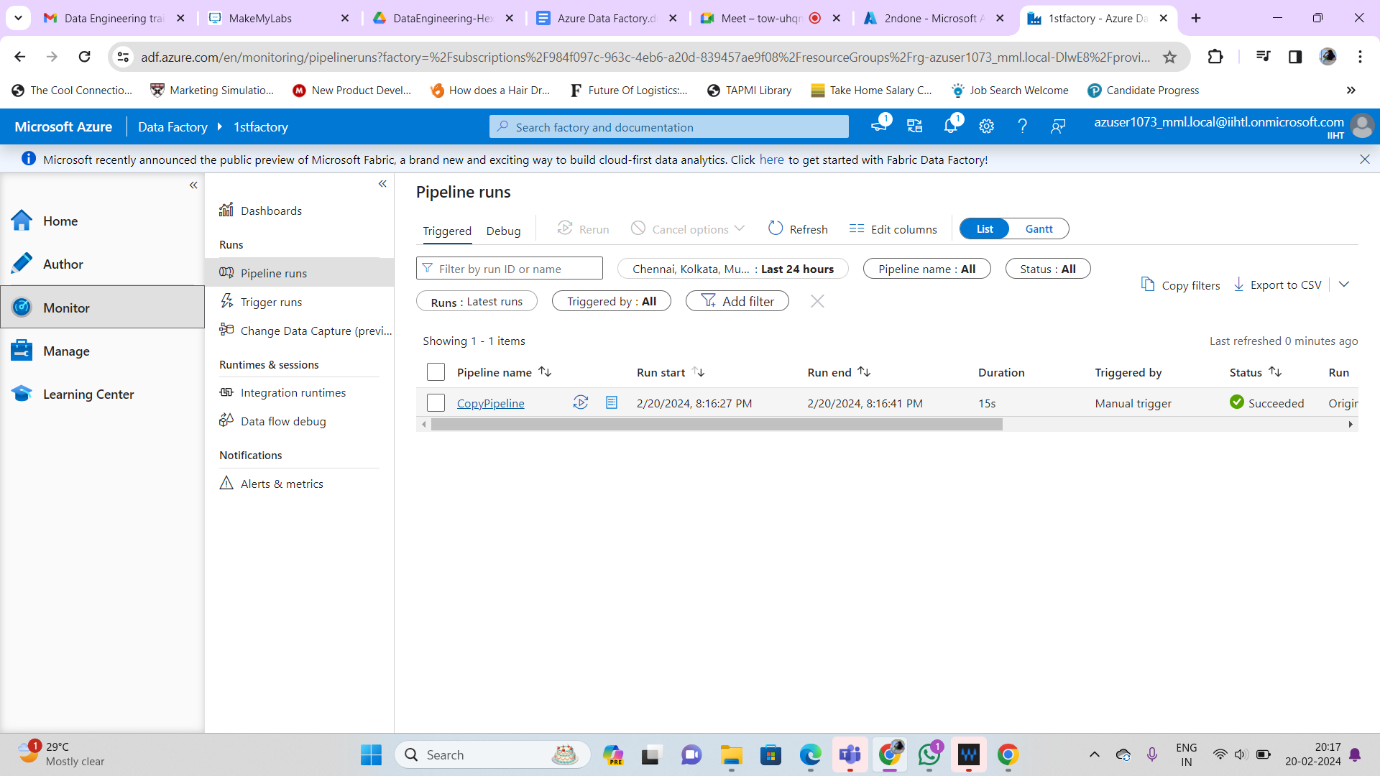
1. On the **Settings** page, specify a name for the pipeline and its description, then select **Next** to use other default conﬁgurations.
2. On the **Summary** page, review all settings, and select **Next**.
3. On the **Deployment complete** page, select **Monitor** to monitor the pipeline that you created.

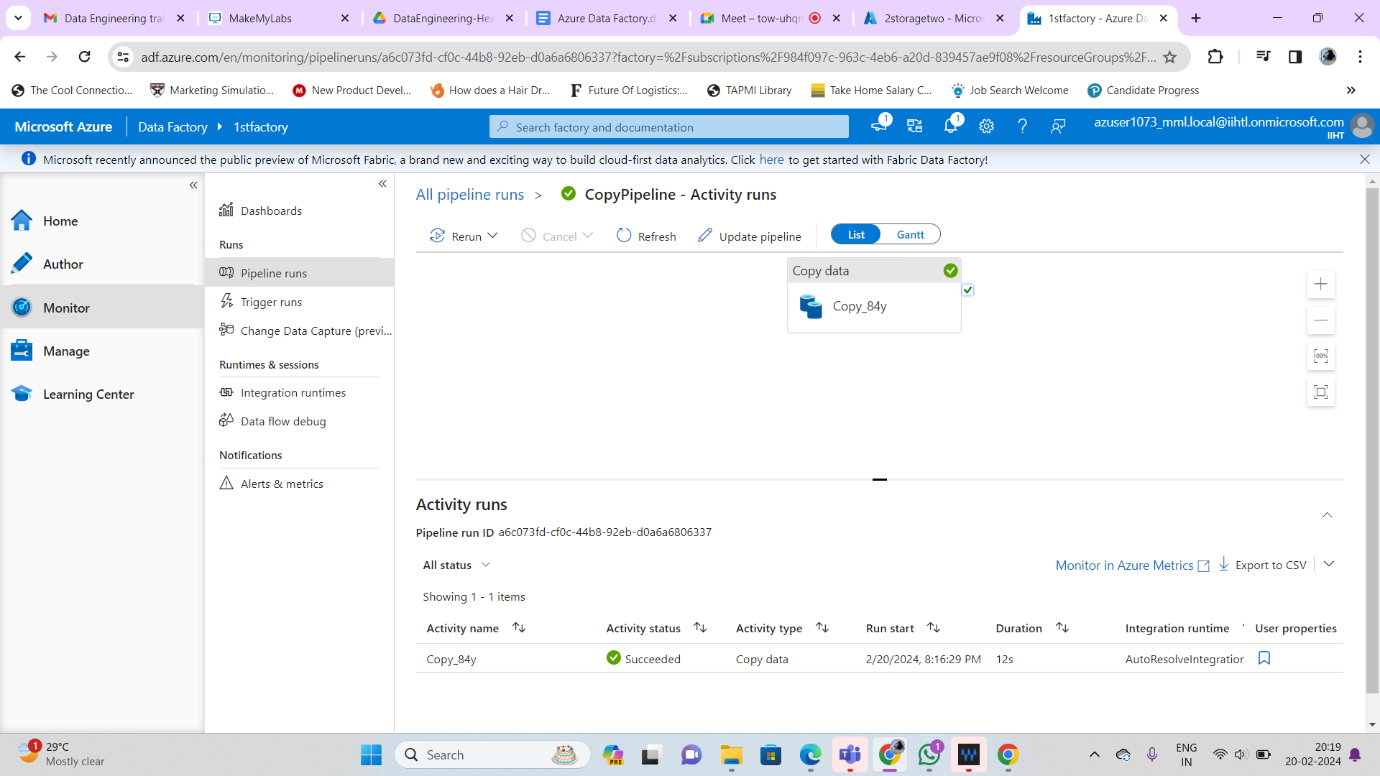
****

# Step 5: Monitor the running results

1. The application switches to the **Monitor** tab. You see the status of the pipeline on this tab. Select **Refresh** to refresh the list. Click the link under **Pipeline name** to view activity run details or rerun the pipeline.
2. On the Activity runs page, select the **Details** link (eyeglasses icon) under the **Activity name** column for more details about copy operation. For details about the properties, see Copy Activity overview.

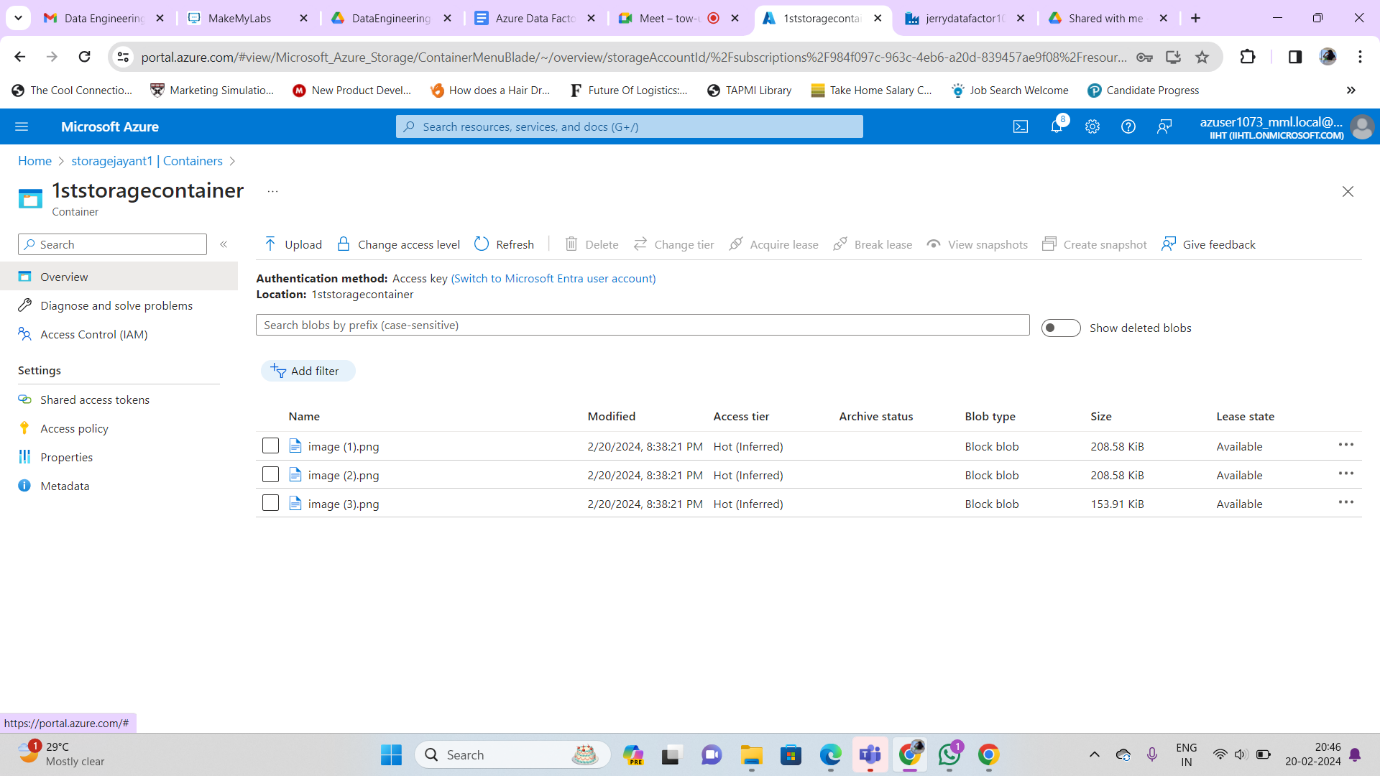
**Check Pipeline status**

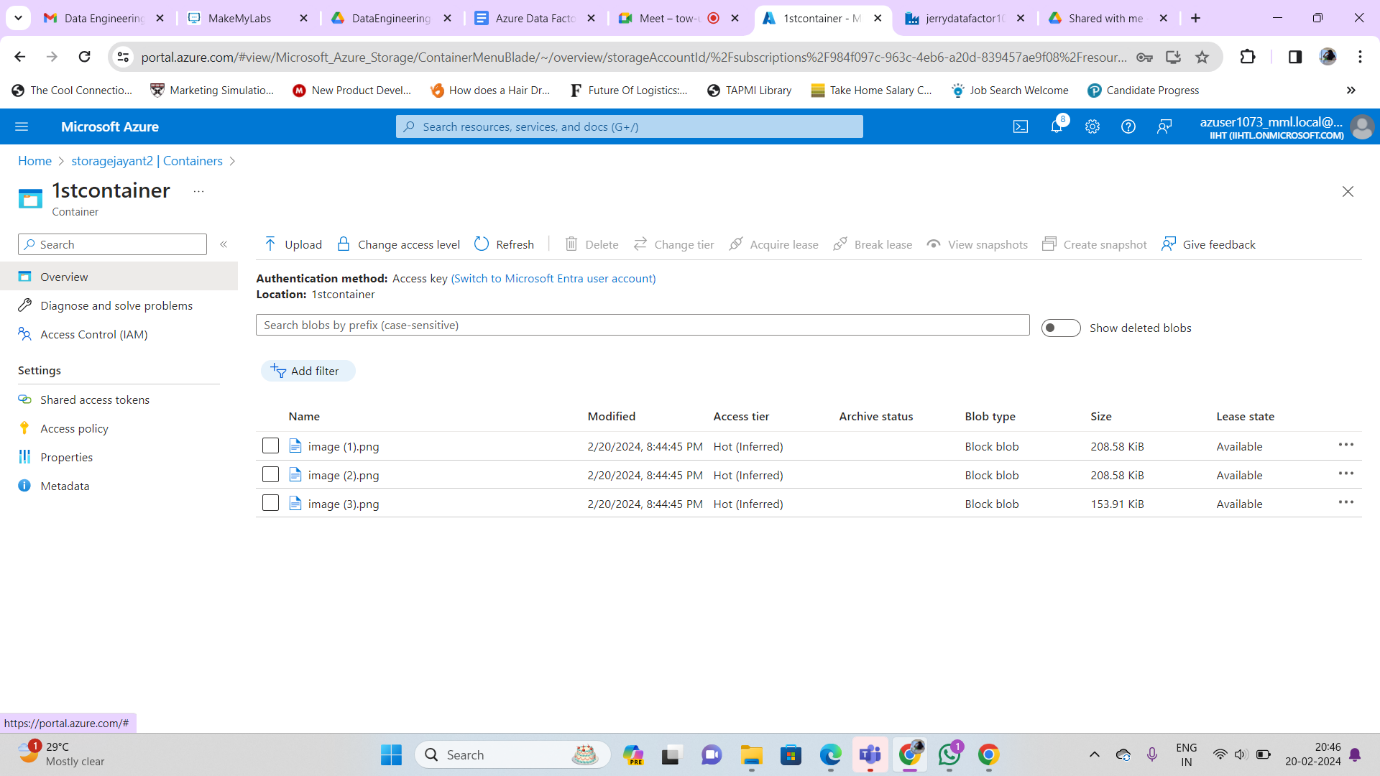
****

****

**Check data in both container**

The files in the 1st storage container has been copied to 2nd storage container.

****

****

**Notes Written:**

