**AZURE ASSESSMENT**

**Que1)** Exploratory data analysis (EDA) in Databricks & Visualizing data in Databricks

Exploratory Data Analysis (EDA) in Databricks is a powerful way to understand and analyse your data before diving into modelling or further processing. Databricks provides a unified analytics platform that allows for easy data exploration, manipulation, and visualisation using Apache Spark.

Here's a step-by-step guide on how to perform EDA and visualise data in Databricks:

**Load Data**:

Start by loading your dataset into Databricks. You can load data from various sources like CSV files, databases, or cloud storage systems like AWS S3 or Azure Blob Storage.

**Understanding the Data**:

Use built-in functions to understand the structure, schema, and basic statistics of your dataset.

**Data Cleaning**:

Perform any necessary data cleaning steps such as handling missing values, removing duplicates, or correcting data types.

**Data Visualization**:

Databricks provides various visualisation libraries like Matplotlib, Seaborn, Plotly, and built-in visualisation capabilities.

**Advanced Analysis**:

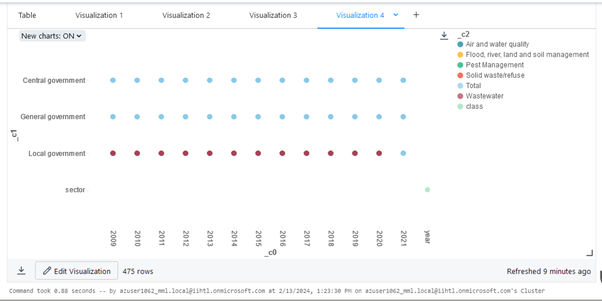
Perform more advanced analysis like correlation, distribution analysis, or time series analysis.

**Dashboard Creation**:

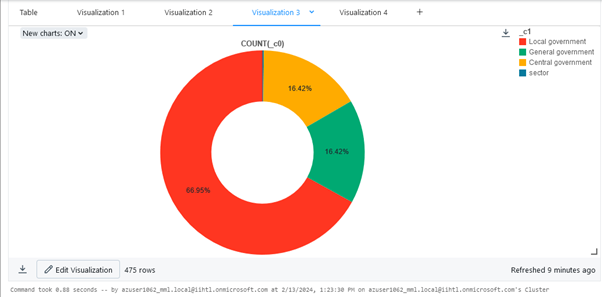
Databricks allows you to create interactive dashboards using tools like Databricks Visualization or third-party libraries like Plotly Dash.

**Exploratory Data Analysis (EDA) in Databricks is a powerful way to understand and analyse your data which can be done using visualisation using Apache Spark.**

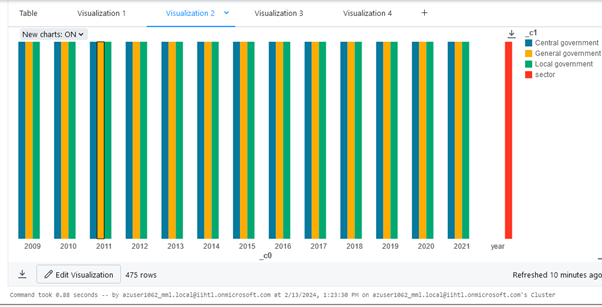
**SCATTER**

****

**PIE**

****

**HISTOGRAM**

****

**Que 2)** Explain Overview of 3 level namespace and creating Unity Catalog objects.

The 3-level namespace in Unity Catalog is a hierarchical system for organising and managing data assets. It consists of three main levels:

**Catalogue:**

Think of this as the top-level container, similar to a database in other systems.

Used to group related data assets based on project, department, or any other relevant criteria.

Users need "USE CATALOGUE" permission to see and access a catalogue.

**Schema:**

Acts like a subfolder within a catalogue, further organising data within a specific category.

Often used to represent different business domains or functional areas.

Users need "USE SCHEMA" permission to access objects within a schema.

**Objects:**

This is where your actual data resides, including:

Tables: Stores rows and columns of data.

Views: Virtual tables based on underlying data and calculations.

Volumes: Hold non-tabular data like images or documents.

Models: Registered machine learning models.

User-defined Functions (UDFs): Custom SQL functions.

Objects inherit access permissions from their parent schema and catalogue.

**Benefits of the 3-Level Namespace:**

Improved organisation: Makes it easier to find and manage data assets.

Reduced naming conflicts: Ensures unique names for objects across different contexts.

Enhanced security: Granular access control based on the level of hierarchy.

**Que3**) Execute & explain, Azure datafactory and its copy activity.

Azure Data Factory (ADF) is a cloud-based data integration service that allows you to create, schedule, and manage data pipelines for ingesting, transforming, and loading data across various data stores and services. A key component of Azure Data Factory is the Copy Activity, which is used to move data between different data stores. Let's execute and explain the process of using Azure Data Factory and its Copy Activity:

**Create an Azure Data Factory**:

Navigate to the Azure portal and create a new Azure Data Factory instance.

Provide the necessary details such as name, subscription, resource group, and region.

**Create Linked Services**:

Linked Services are connections to external data sources or destinations. You need to create linked services for the source and destination data stores you want to move data between.

For example, you might create linked services for Azure Blob Storage, Azure SQL Database, or Azure Data Lake Storage Gen2.

**Create Datasets**:

Datasets represent the structure and location of your data within the linked services.

Define datasets for both the source and destination data stores. These datasets specify the data format, file paths, tables, etc.

**Create Pipelines**:

Pipelines are the workflow definitions that orchestrate data movement and transformation activities.

Create a new pipeline and add a Copy Activity to it.

**Configure Copy Activity**:

Configure the Copy Activity by specifying the source dataset, destination dataset, and any additional settings such as data transformation options, file format conversions, etc.

You'll map the columns from the source dataset to the columns in the destination dataset if necessary.

**Trigger Execution**:

Once the pipeline and Copy Activity are configured, you can trigger the execution manually or schedule it to run at specific intervals.

Monitor the pipeline execution to track the progress and ensure that data is being copied successfully.

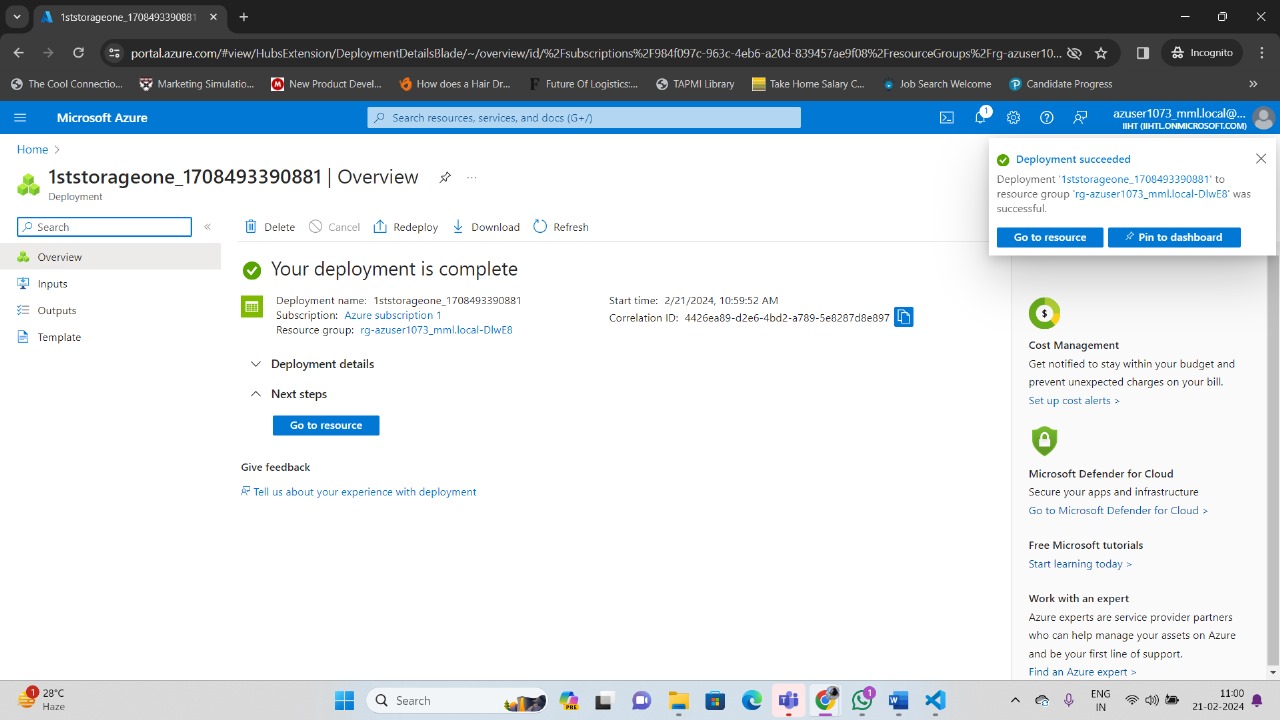
**Monitor and Manage**:

Azure Data Factory provides monitoring and management capabilities to track the performance and health of your data pipelines.

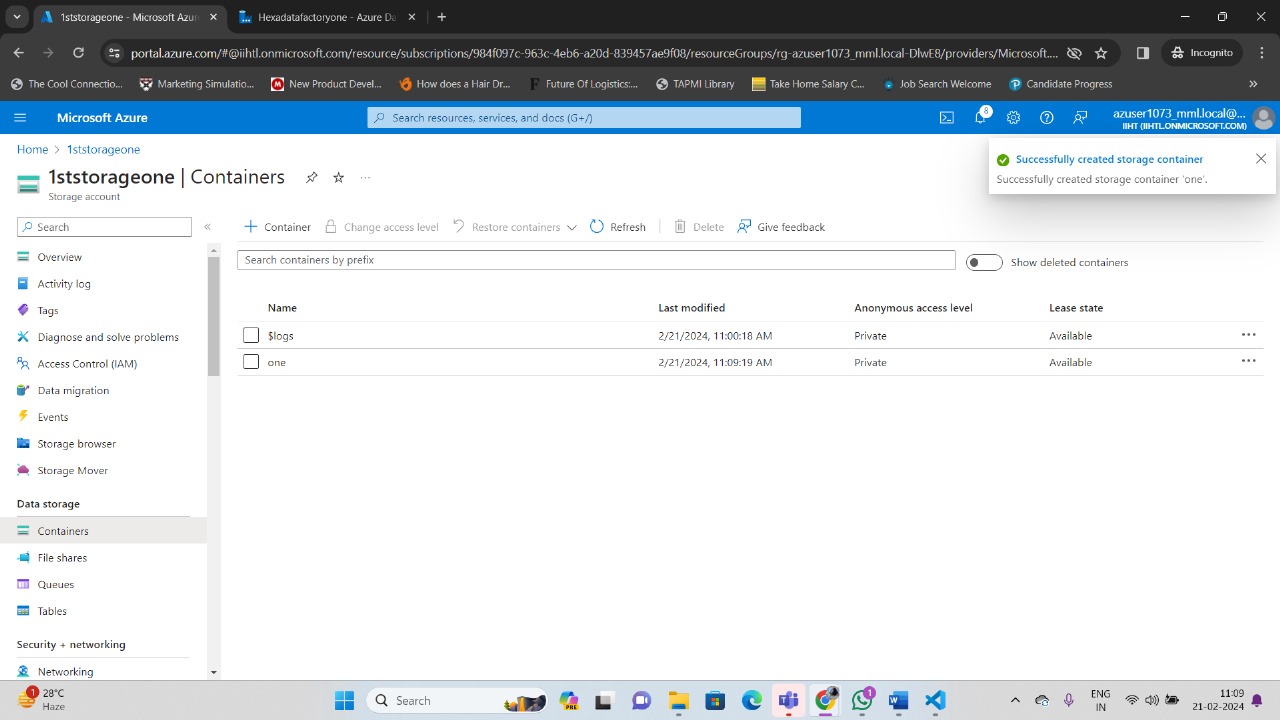
You can view execution logs, monitor data movement activities, and troubleshoot any issues that arise during execution.

By following these steps, you can use Azure Data Factory and its Copy Activity to move data between different data stores efficiently and reliably. This process allows you to integrate data from various sources, transform it as needed, and load it into your target data stores for further analysis or consumption.

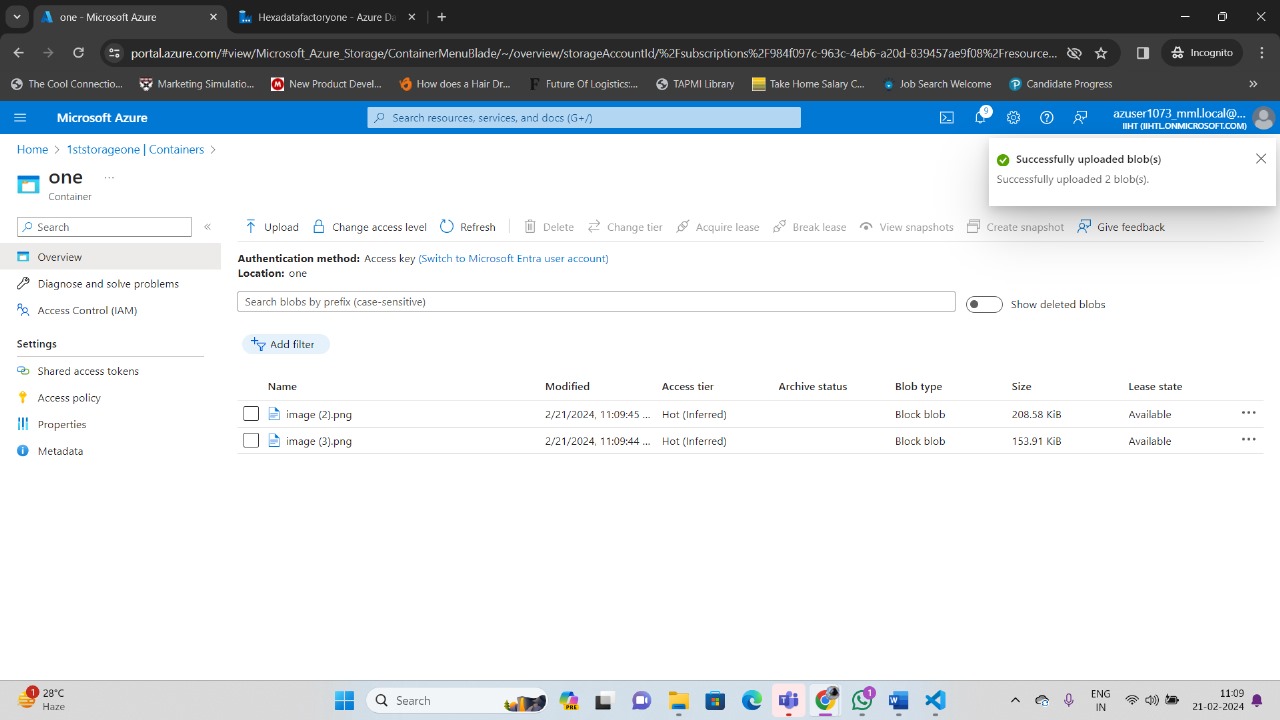
**First Storage Account**



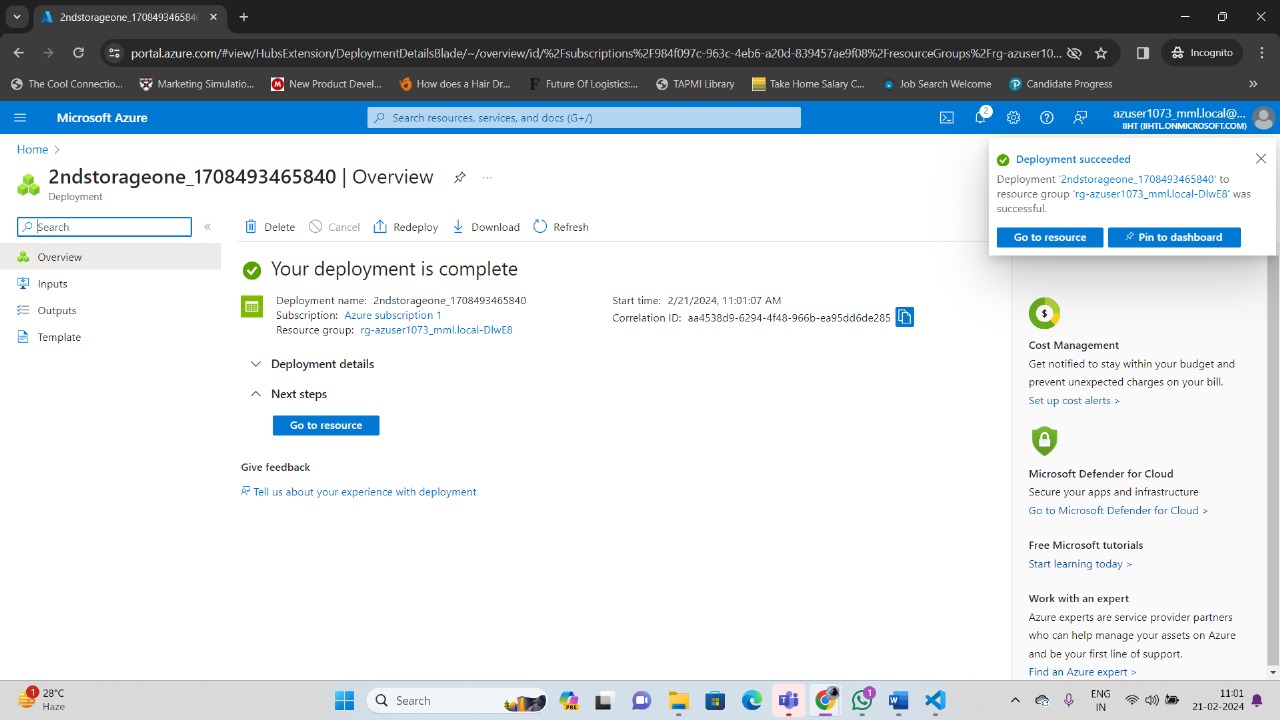
**Storage Container**



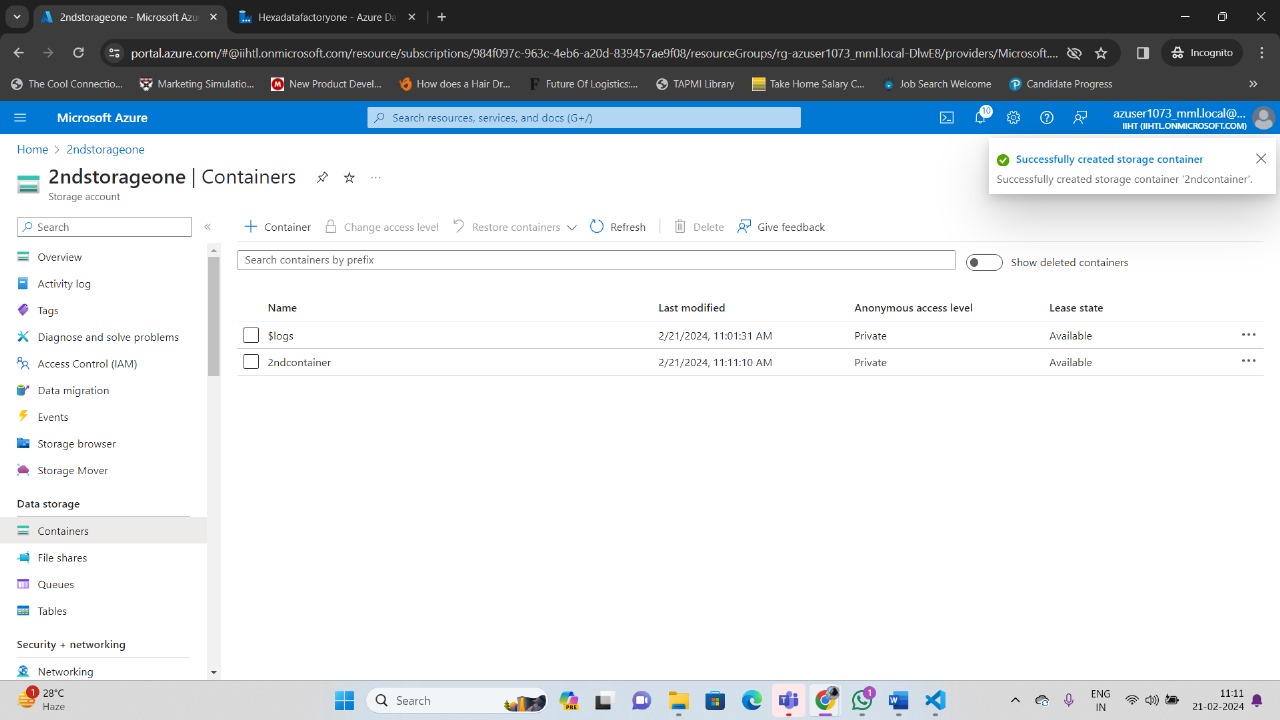
**Data in First Storage Account**



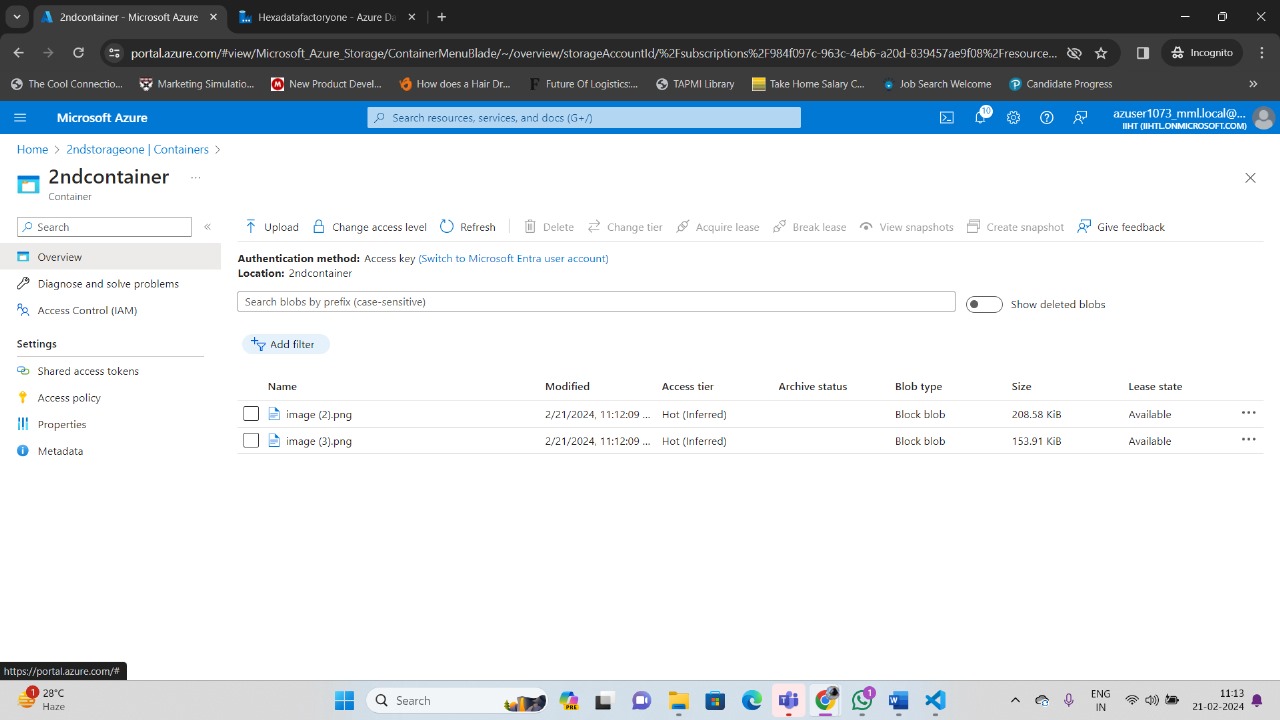
**Second Storage Account**



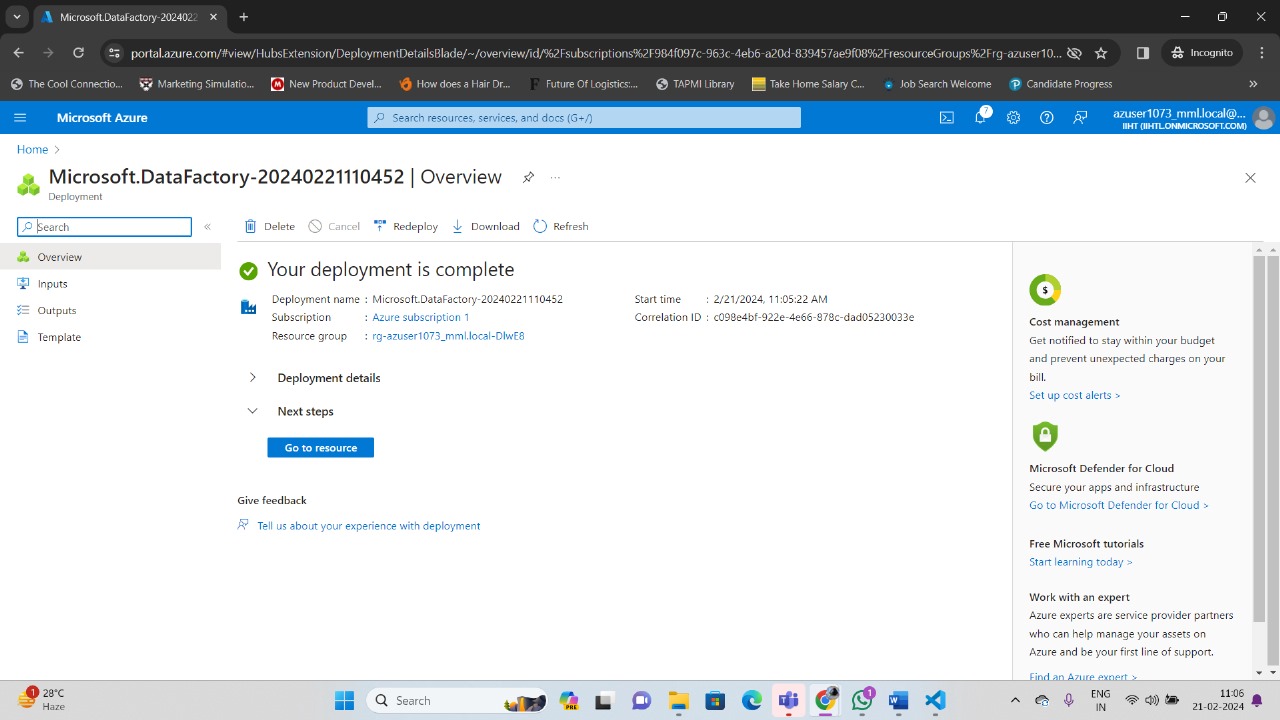
**Container**

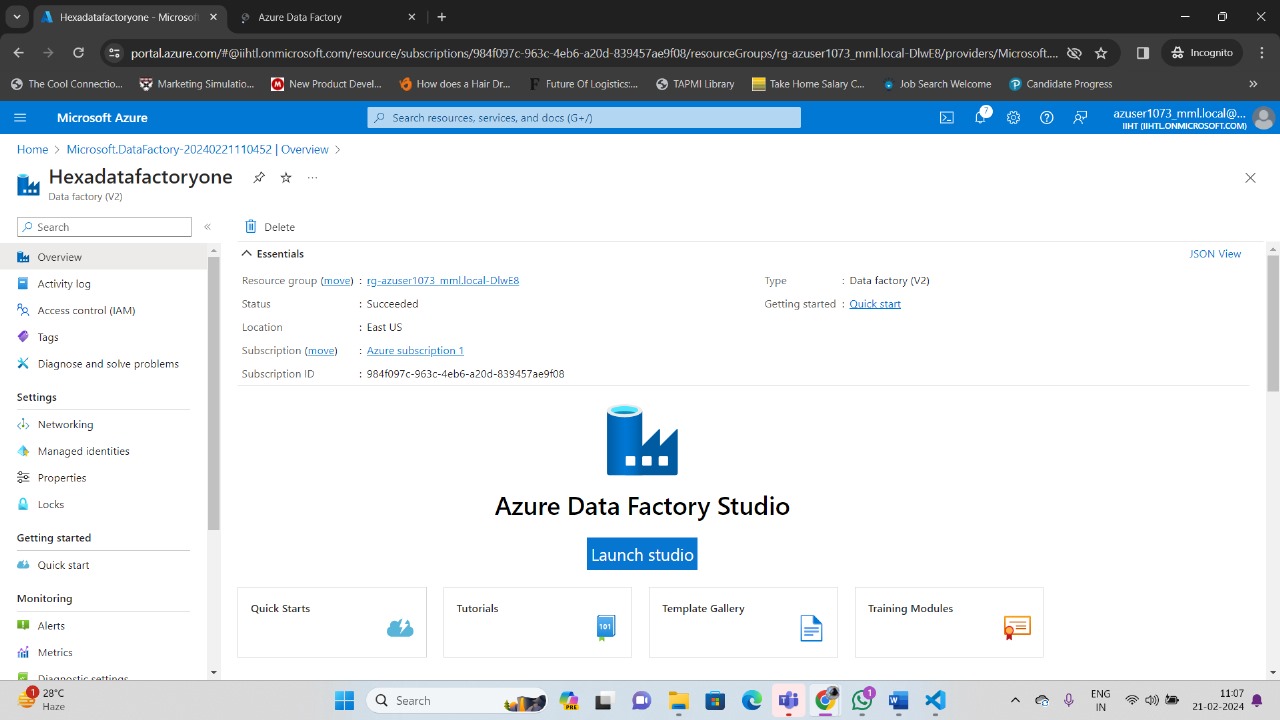


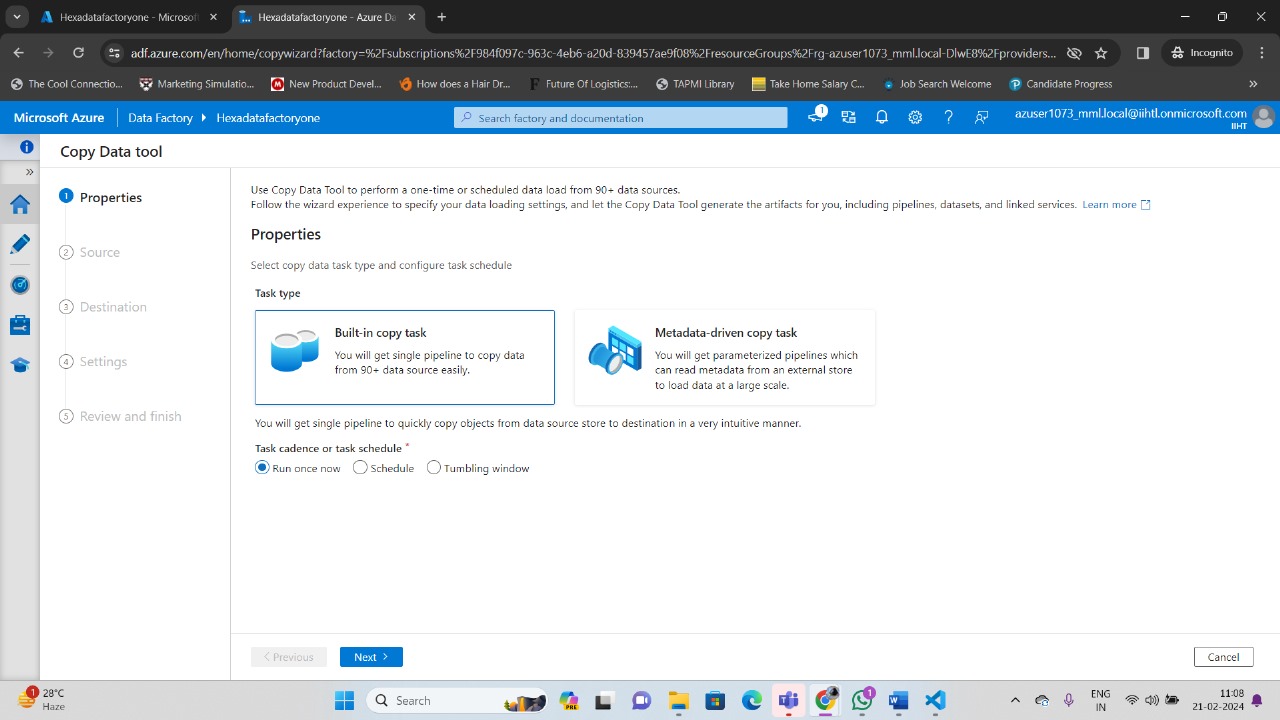
**Data in Second Storage Account Copied from First Storage**

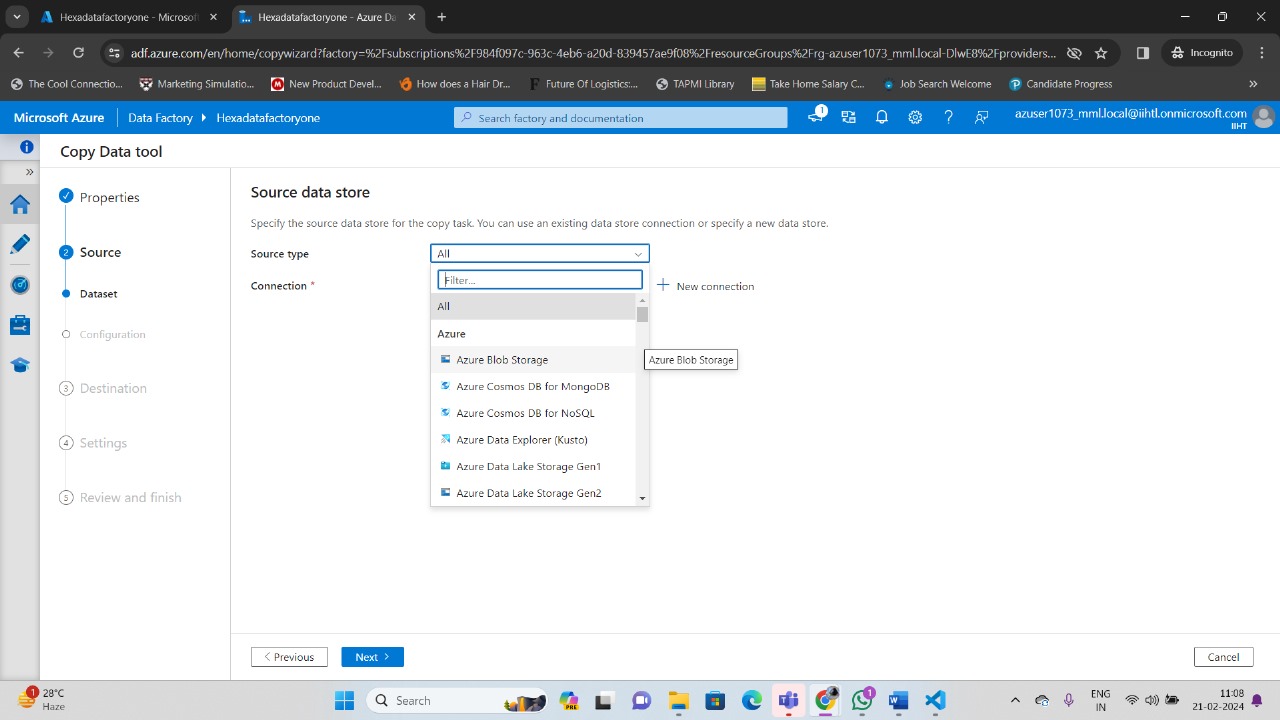


**Azure Data Factory**

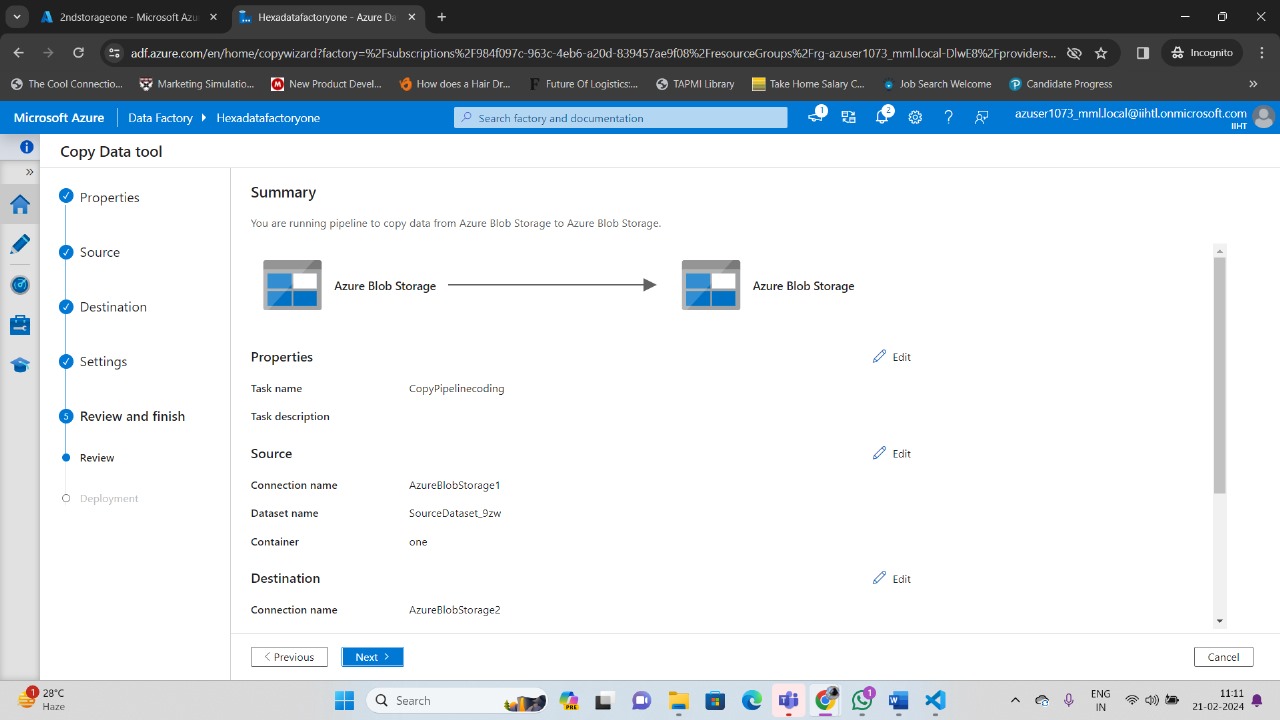


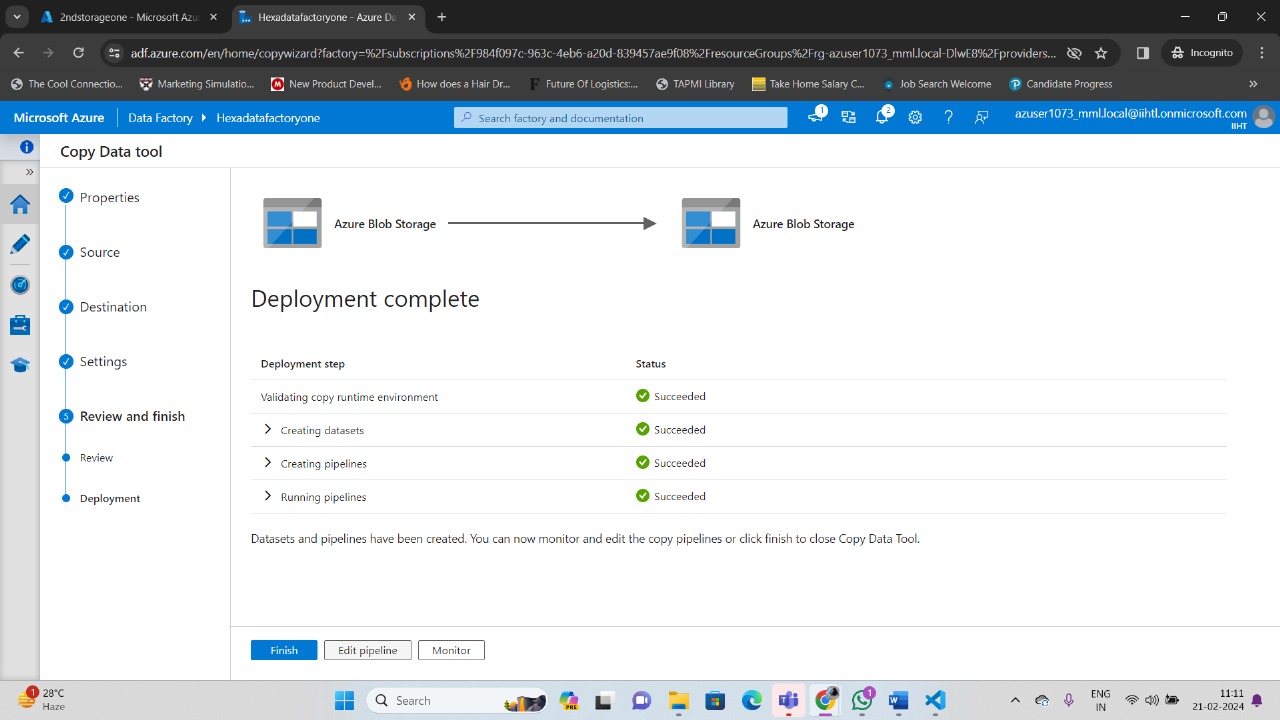


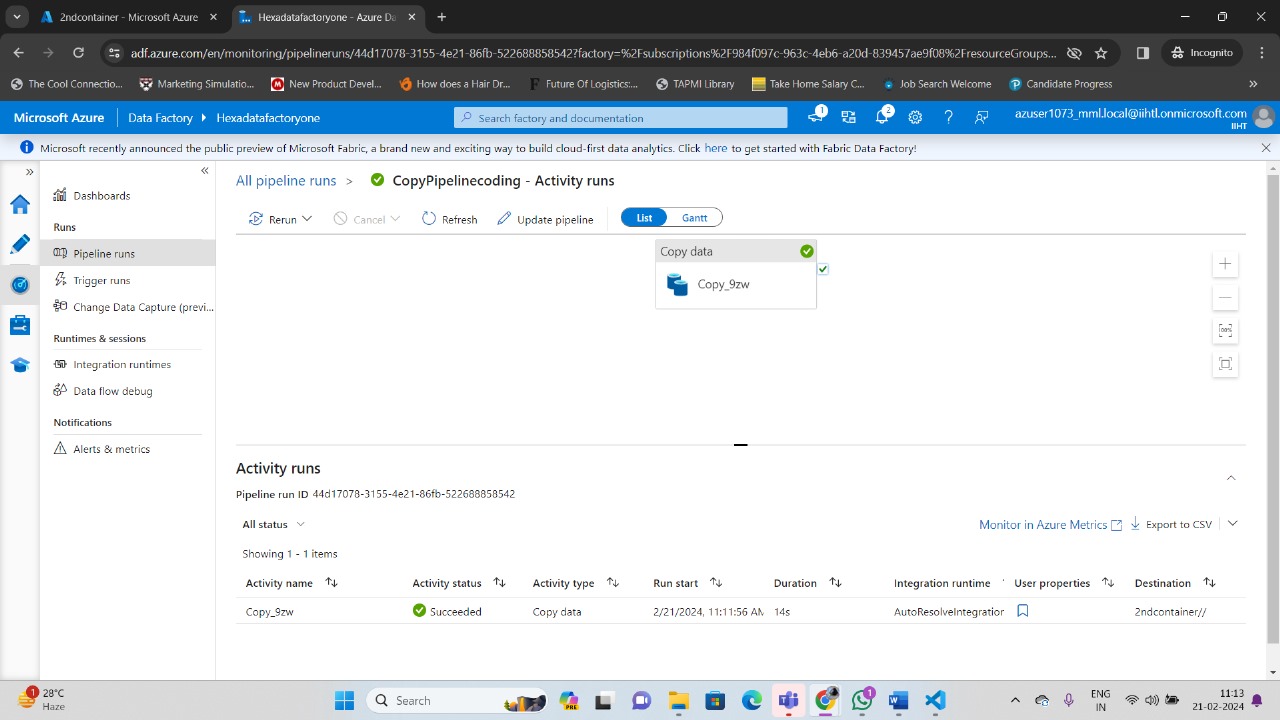




**Copy Data Tools**



****

****