

# Data-Engineering-Project-Using-Azure Databricks

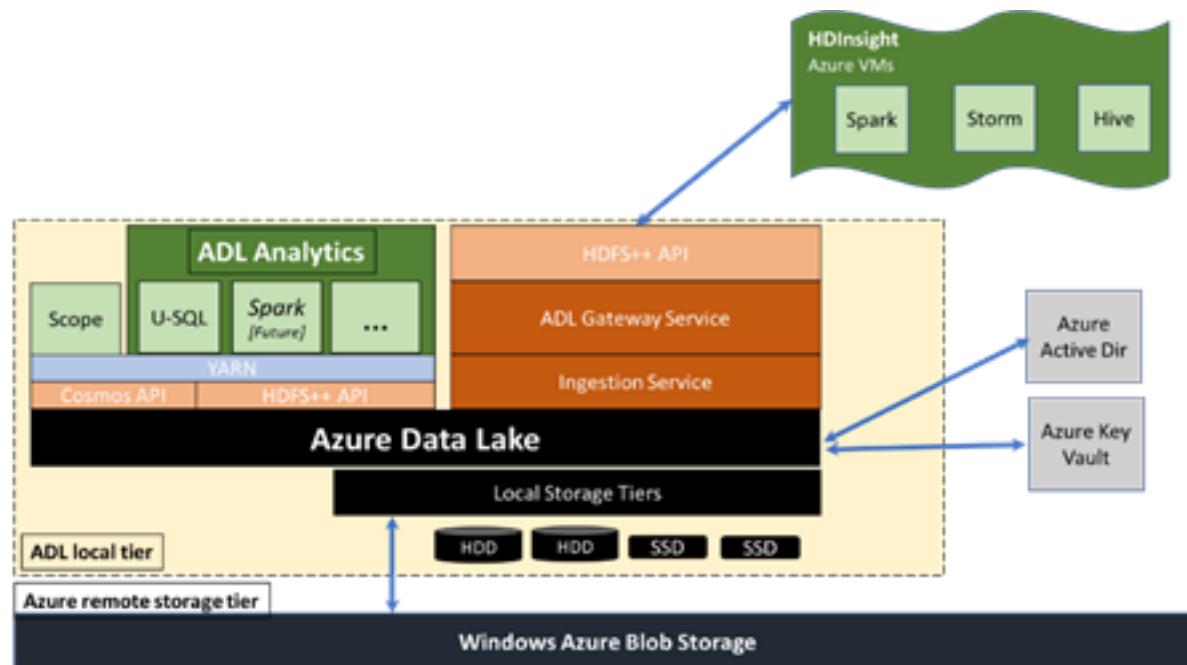
## Integrate Azure Data Factory with Azure Data Lake Storage

### Project Overview:

### Objective:

This project aims to establish a data pipeline using Azure Data Factory (ADF) to transfer data between various folders within an Azure Data Lake Storage (ADLS) account.

### Architecture diagram:



## Components:

### 1. Azure Data Factory (ADF)

- **Data Factory Workspace:** A cloud-based environment within Azure to develop and manage data pipelines.
- **Linked Services:** Connections established within ADF to access external data sources and sinks, including the source and destination ADLS folders in this case.
- **Data Flows :** A visual interface within ADF to design data transformation logic, including filtering, cleansing, and enriching data (if required).
- **Pipelines:** Workflows defined in ADF that orchestrate data movement between various sources and destinations. They consist of activities that specify the data operations to be performed.
- **Monitoring & Management Tools:** Features within ADF to track pipeline execution status, identify errors, and manage pipeline schedules.

### 2. Azure Data Lake Storage (ADLS)

- **Data Lake Account:** A managed storage service in Azure for storing large datasets in a highly scalable and secure manner.
- **Folders:** Organizational units within the ADLS account to categorize and manage data. The project will define a source folder containing the data to be moved and a target folder for the transferred data.

## Data Flow:

The data flow in this project involves transferring data between folders within an Azure Data Lake Storage (ADLS) account using Azure Data Factory (ADF). Here's a breakdown of the steps:

### **1. Source:**

- ADF initiates the process by accessing the **source folder** within your ADLS account.
- It identifies the data files based on specific criteria like file format, naming convention, or last modified date (configurable within the pipeline).

### **2. Data Processing :**

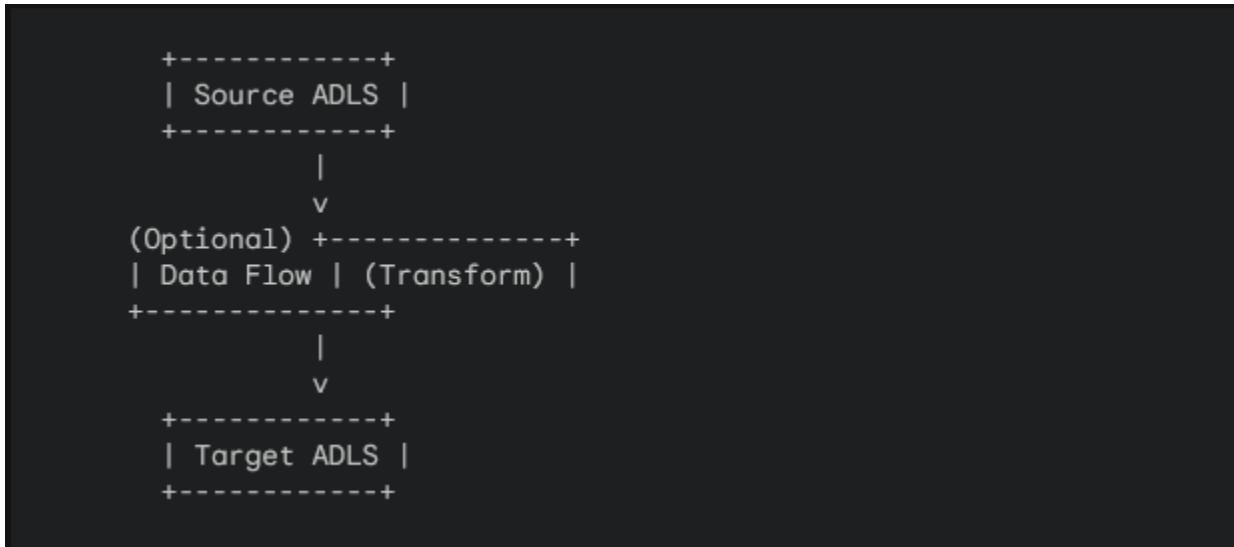
- This step is **optional** and depends on your specific needs.
- If data transformation is required, ADF utilizes its **data flow** capabilities to perform the necessary operations.
- This data flow can involve:
  - **Filtering:** Selecting specific data based on defined criteria.
  - **Cleansing:** Removing inconsistencies or errors from the data.
  - **Enriching:** Adding additional data points from other sources.
- The transformed data becomes the input for the next step.

### **3. Destination:**

- The processed data (or original data if no transformation is applied) is then transferred to the designated **target folder** within ADLS.
- ADF ensures the data is written to the target location in the desired format.

## 4. Visualization:

Here's a simplified visualization of the data flow:



## 5. Additional Points:

- The data flow can be configured to handle various file formats commonly used in data lakes, such as CSV, Parquet, and Avro.
- Error handling mechanisms can be implemented within the pipeline to address any issues during data transfer, such as file corruption or network errors.
- The entire process can be scheduled to run at specific intervals using ADF's scheduling capabilities, ensuring data is transferred and potentially transformed regularly.

## How it works:

This project outlines three pipelines using Azure Data Factory (ADF) to move data between various locations:

### **1. ADLS to ADLS (adls-adls):**

This pipeline transfers data from one folder to another within the same Azure Data Lake Storage (ADLS) account.

#### **Components:**

- **Source Linked Service:** Connection to the source ADLS folder.
- **Sink Linked Service:** Connection to the destination ADLS folder.
- **Copy Activity:** The activity responsible for copying data between the folders.

#### **Steps:**

- **Define Linked Services:** Configure linked services for both the source and destination ADLS folders, specifying the account name, access key, and other relevant details.
- **Create Pipeline:** Design a new pipeline in ADF.
- **Add Copy Activity:** Drag and drop a "Copy Data" activity onto the pipeline canvas.
- **Configure Copy Activity:**
  - **Source:** Select the source linked service and specify the source folder path.
  - **Destination:** Select the sink linked service and specify the destination folder path.
  - **File format:** Choose the appropriate file format based on your data (e.g., CSV, Parquet).
- **Schedule and Run:** Set up a schedule (optional) or run the pipeline manually.

## **2. ADLS to Azure Blob Storage (adls-blob):**

This pipeline transfers data from an ADLS folder to an Azure Blob Storage container.

### **Components:**

- **Source Linked Service:** Connection to the source ADLS folder.
- **Sink Linked Service:** Connection to the destination Azure Blob Storage container.
- **Copy Activity:** The activity responsible for copying data between ADLS and Blob storage.

### **Steps:**

- **Define Linked Services:** Configure linked services for both the source ADLS folder and the destination Blob storage container.
- **Create Pipeline:** Design a new pipeline in ADF.
- **Add Copy Activity:** Drag and drop a "Copy Data" activity onto the pipeline canvas.
- **Configure Copy Activity:**
  - **Source:** Select the source linked service and specify the source folder path.
  - **Destination:** Select the sink linked service and specify the destination Blob storage container and the desired blob name (optional).
  - **File format:** Choose the appropriate format based on your data.
- **Schedule and Run:** Set up a schedule (optional) or run the pipeline manually.

### **3. Azure Blob Storage to ADLS (blob-adls):**

This pipeline transfers data from an Azure Blob Storage container to an ADLS folder.

#### **Components:**

- **Source Linked Service:** Connection to the source Azure Blob Storage container.
- **Sink Linked Service:** Connection to the destination ADLS folder.
- **Copy Activity:** The activity responsible for copying data between Blob storage and ADLS.

#### **Steps:**

- **Define Linked Services:** Configure linked services for both the source Blob storage container and the destination ADLS folder.
- **Create Pipeline:** Design a new pipeline in ADF.
- **Add Copy Activity:** Drag and drop a "Copy Data" activity onto the pipeline canvas.
- **Configure Copy Activity:**
  - **Source:** Select the source linked service and specify the Blob storage container name and the blob name (optional).
  - **Destination:** Select the sink linked service and specify the destination ADLS folder path.
  - **File format:** Choose the appropriate format based on your data.
- **Schedule and Run:** Set up a schedule (optional) or run the pipeline manually.

## **Overall Working:**

These pipelines utilize the "Copy Data" activity in ADF to transfer data between specified locations. You can schedule the pipelines to run periodically or trigger them based on specific events. By combining these individual pipelines, you can achieve complex data movement workflows within your data lake environment.

## **Tools Used:**

1. **Azure Data Factory (ADF):** Cloud-based service for orchestrating data movement and transformation.
2. **Azure Data Lake Storage (ADLS):** Managed storage service for storing large datasets in a scalable and secure manner.

## **Execution Overview:**

This project aims to establish three data pipelines using Azure Data Factory (ADF) to automate data movement between various locations in your Azure cloud environment:

- 1. ADLS to ADLS (adls-adls):** Transfers data between folders within the same Azure Data Lake Storage (ADLS) account.
- 2. ADLS to Blob Storage (adls-blob):** Transfers data from an ADLS folder to an Azure Blob Storage container.
- 3. Blob Storage to ADLS (blob-adls):** Transfers data from an Azure Blob Storage container to an ADLS folder.

## Execution Flow:

1. **Define Linked Services:** Establish connections between ADF and your data sources and sinks. This involves creating separate linked services for each source and destination in each pipeline (ADLS accounts, Blob storage containers).
2. **Develop Pipelines:** Design individual pipelines in ADF for each data transfer scenario (adls-adls, adls-blob, blob-adls).
3. **Configure Copy Activities:** Within each pipeline, utilize the "Copy Data" activity to specify the source and destination locations, file format, and other relevant settings.
4. **Schedule and Run Pipelines:** Choose the execution model for each pipeline (e.g., manual, scheduled). Schedule the pipelines to run periodically or trigger them manually based on your needs.
5. **Monitor Execution:** Track the execution status and history of each pipeline using the ADF monitoring tools. This allows you to identify any errors or issues and ensure successful data movement.

## Benefits:

- **Automation:** Eliminates manual data transfer tasks, ensuring consistency and reliability.
- **Scalability:** Handles large datasets efficiently, adapting to your data processing needs.
- **Scheduling:** Runs pipelines at specific intervals or based on events, keeping data up-to-date.
- **Monitoring:** Provides centralized oversight of data movement processes for troubleshooting and management.

## Azure Resources Used for this Project:

This project utilizes several Azure resources to facilitate data transfer between different storage locations:

### **1. Azure Data Factory (ADF):**

- **ADF Workspace:** A cloud-based environment within Azure to develop, manage, and orchestrate data pipelines.

This serves as the central hub for designing and executing the data movement workflows.

### **2. Azure Data Lake Storage (ADLS):**

- **ADLS Account:** A managed storage service in Azure for storing large datasets in a highly scalable and secure manner.
- This project utilizes ADLS accounts in two ways:
  - **Source:** Stores the data to be transferred in one or more folders within the ADLS account.
  - **Destination:** Holds the transferred data in a designated folder within the ADLS account.

### **3. Azure Blob Storage (Optional, used in adls-blob and blob-adls pipelines):**

- **Blob Storage Container:** A container within an Azure Blob Storage account that serves as a repository for object data.
- This project utilizes Blob storage containers in two ways:
  - **Source:** Stores the data to be transferred in the adls-blob pipeline.
  - **Destination:** Holds the transferred data in the blob-adls pipeline.

## **4. Additional Resources (Optional):**

- **Azure Key Vault:** A managed service that securely stores and manages cryptographic keys and secrets.

This can be used to store access keys for ADLS and Blob storage accounts, improving security by avoiding hardcoding them in the pipelines

## **Project Requirements:**

This project outlines the requirements for developing three data pipelines using Azure Data Factory (ADF) to automate data movement between various storage locations:

### **1. Functional Requirements:**

- **Data Transfer:** The pipelines must successfully transfer data between the specified source and destination locations for each scenario (ADLS-to-ADLS, ADLS-to-Blob, Blob-to-ADLS).
- **File Format Handling:** The pipelines must be able to handle the chosen file formats for the data being transferred (e.g., CSV, Parquet).
- **Scheduling:** The pipelines should be configurable to run at specific intervals or be triggered manually based on your needs.
- **Monitoring:** The project should enable monitoring of pipeline execution status and history to track progress and identify any errors.

### **2. Non-Functional Requirements:**

- **Scalability:** The pipelines should be able to handle large datasets efficiently and scale to meet future data processing needs.

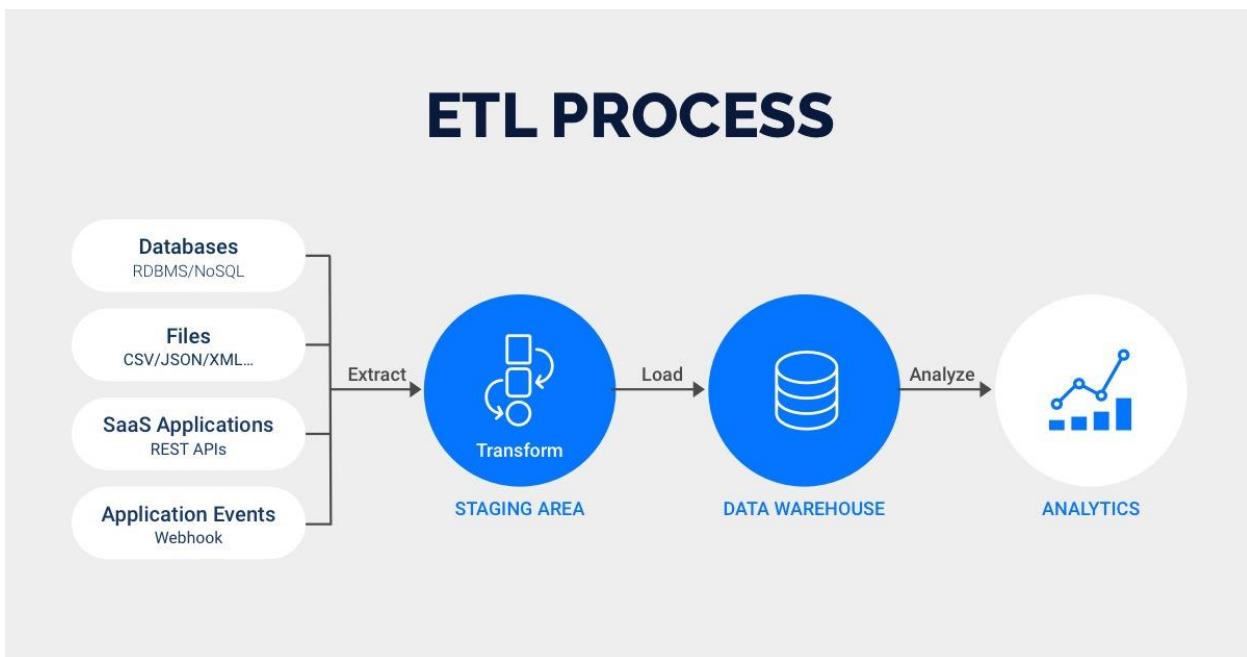
- **Security:** The project should implement security best practices, such as using Azure Key Vault to store access keys securely and avoiding hardcoding them in the pipelines.
- **Performance:** The pipelines should be designed and optimized for efficient data transfer, minimizing processing time and resource utilization.
- **Logging:** The pipelines should log relevant information during execution, such as success/failure status, data volume transferred, and any encountered errors.
- **Documentation:** The project should be well-documented, including details on pipeline configurations, data formats, and troubleshooting steps.

### **3. Additional Requirements:**

- Familiarity with Azure Data Factory and its capabilities.
- Access to the necessary Azure resources, including an ADF workspace, ADLS accounts (potentially), and Blob storage accounts (optional).
- Understanding of the data formats involved in the transfer process.

## ETL pipeline

An ETL pipeline, which stands for Extract, Transform, Load, is a series of processes used to move data from one or more sources to a target system, typically a data warehouse or data lake. It's essentially a mechanism for preparing raw data for analysis and further use.



**Here's a breakdown of the three stages involved:**

- 1. Extract:** In this stage, data is retrieved from its various sources. These sources can be diverse, ranging from databases and applications to flat files and social media feeds.
- 2. Transform:** Once extracted, the data is cleaned, filtered, and manipulated to fit the specific needs of the target system. This might involve correcting errors, formatting inconsistencies, combining data sets, or deriving new information.
- 3. Load:** Finally, the transformed data is loaded into the designated target system, where it's readily available for analysis and

reporting. This could be a data warehouse, data lake, or any other system designed to store and manage large amounts of data.

## Benefits of using ETL pipelines:

1. **Improved data quality:** By cleaning and transforming data, ETL pipelines ensure consistency, accuracy, and completeness, leading to more reliable insights.
2. **Centralized data management:** ETL pipelines consolidate data from various sources into a single location, simplifying data access and analysis.
3. **Increased efficiency:** Automating data extraction and transformation saves time and reduces manual effort compared to handling data manipulation tasks individually.
4. **Enhanced decision-making:** By providing clean and readily available data, ETL pipelines empower data-driven decision making within organizations.

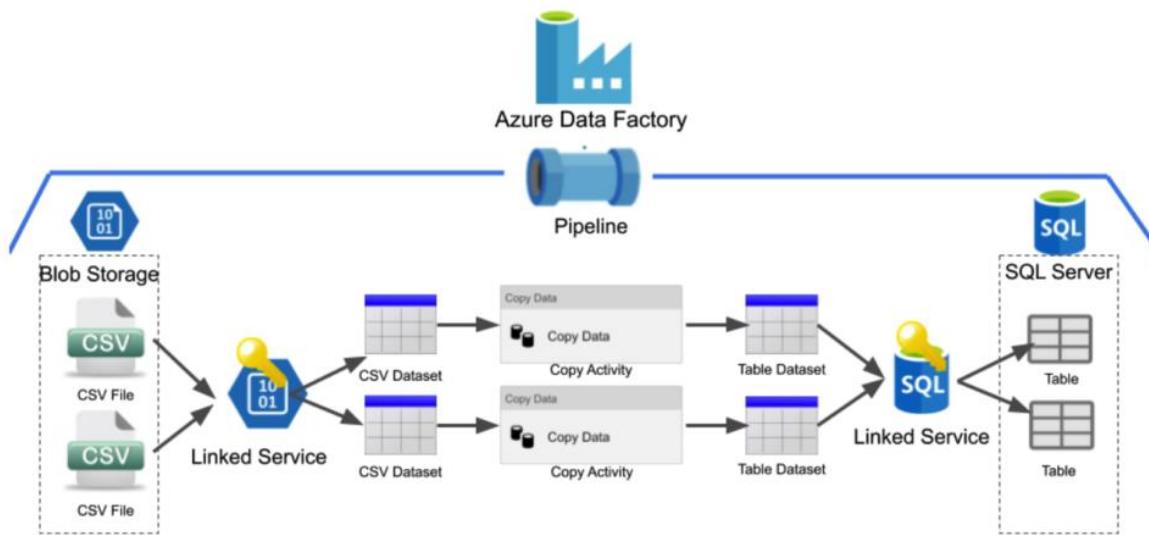
## Use cases for ETL pipelines:

1. **Data migration:** Moving data from legacy systems to new platforms.
2. **Data warehousing:** Preparing data for analysis and reporting.
3. **Data cleansing:** Addressing inconsistencies and errors in data.
4. **Data integration:** Combining data from multiple sources for holistic views.

The project described utilizes an EL (Extract, Load) process rather than a full-fledged ETL pipeline. While data transformation isn't explicitly mentioned, it might be implicitly handled during the copy operation or could be integrated using additional functionalities within the Azure Data Factory (ADF) pipeline.

## Azure Data factory pipeline

Data Factory Pipeline to implement data copy between folders in Azure Data Lake Storage (ADLS). While it doesn't encompass the full ETL (Extract, Transform, Load) process, it leverages a key component of Data Factory Pipelines:



### **The Copy Activity:**

This activity acts as the engine for moving data between various data stores.

### **In this specific scenario, the Copy Activity is configured to:**

1. Extract data from a source folder within an ADLS account (using a linked service).
2. Load the data into a designated destination folder within the same or potentially another ADLS account (using another linked service).
3. While data transformation isn't explicitly included in this instance, the Copy Activity might handle basic transformations like file format conversion during the copy process.

## **Key characteristics of Data Factory Pipelines:**

1. **Visual interface:** ADF provides a user-friendly interface for building and orchestrating data pipelines.
2. **Drag-and-drop functionality:** You can easily create pipelines by dragging and dropping various activities, including the Copy Activity.
3. **Scheduling:** Pipelines can be scheduled to run at specific intervals or triggered by events.
4. **Monitoring and logging:** ADF offers comprehensive monitoring and logging capabilities to track pipeline execution and troubleshoot any issues.

Therefore, while the overall process represents an EL (Extract, Load) approach, it's crucial to understand that a Data Factory Pipeline with the Copy Activity is the fundamental mechanism enabling the data movement between folders within ADLS.

## **Tasks performed:**

### **1 . Create a Pipeline to copy the data task from**

#### **Azure Data Lake Storage Gen2 to Azure Data Lake Storage Gen2**

##### **Steps:**

Login in to Azure Portal and create a Storage Account

Storage Account Name - projectadls

The screenshot shows the 'Create a storage account' wizard in the Microsoft Azure portal. The current step is 'Basics'. The page title is 'Create a storage account'. The top navigation bar includes 'Microsoft Azure', a search bar, and user information 'azuser1051\_mml.local@...'. Below the title, the breadcrumb path is 'Home > Storage accounts > Create a storage account'. The main content area has tabs for 'Basics', 'Advanced', 'Networking', 'Data protection', 'Encryption', 'Tags', and 'Review'. A note states: 'Azure Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, scalable, and redundant. Azure Storage includes Azure Blobs (objects), Azure Data Lake Storage Gen2, Azure Files, Azure Queues, and Azure Tables. The cost of your storage account depends on the usage and the options you choose below.' A link 'Learn more about Azure storage accounts' is provided. The 'Project details' section asks to select a subscription and resource group. The 'Subscription' dropdown is set to 'Azure subscription 1' and the 'Resource group' dropdown is set to 'rg-azuser1051\_mml.local-MUqzl'. Below these fields are 'Create new' and 'Next : Advanced >' buttons. The bottom of the screen shows a taskbar with various icons and system status.

## Region - East US

The screenshot shows the 'Create a storage account' wizard in the Microsoft Azure portal. The current step is 'Instance details'. The page title is 'Create a storage account'. The top navigation bar includes 'Microsoft Azure', a search bar, and user information 'azuser1051\_mml.local@...'. Below the title, the breadcrumb path is 'Home > Storage accounts > Create a storage account'. The main content area has tabs for 'Basics', 'Advanced', 'Networking', 'Data protection', 'Encryption', 'Tags', and 'Review'. The 'Instance details' section contains fields for 'Storage account name' (set to 'projectadls'), 'Region' (set to '(US) East US 2'), and 'Performance'. Under 'Performance', the 'Standard' radio button is selected with the note: 'Recommended for most scenarios (general-purpose v2 account)'. The 'Premium' radio button is also present. Under 'Redundancy', the 'Geo-redundant storage (GRS)' dropdown is selected, and the checkbox 'Make read access to data available in the event of regional unavailability.' is checked. Below these fields are 'Review', '< Previous', and 'Next : Advanced >' buttons. The bottom of the screen shows a taskbar with various icons and system status.

# Enable hierarchical namespace

The screenshot shows the Microsoft Azure portal interface for creating a storage account. The user is on the 'Review' tab of the 'Create a storage account' wizard. In the 'Basics' section, the 'Enable hierarchical namespace' setting is highlighted as 'Enabled'. Other settings include 'Subscription: Azure subscription 1', 'Resource Group: rg-azuser1051\_mml.local-MUqzl', 'Location: eastus2', 'Storage account name: projectadls', 'Deployment model: Resource manager', 'Performance: Standard', and 'Replication: Read-access geo-redundant storage (RA-GRS)'. The 'Advanced' section shows 'Enable network file system v2' as 'Disabled'. At the bottom, there are 'Create' and 'Next >' buttons, along with a 'Download a template for automation' link. The status bar at the bottom right shows the date as 2/27/2024 and the time as 11:40 AM.

## Storage Account Created

The screenshot shows the Microsoft Azure portal interface displaying deployment details for 'projectadls\_1709014208381'. The deployment status is shown as 'Deployment succeeded'. Key information includes the deployment name, start time (2/27/2024, 11:40:10 AM), subscription, resource group, and correlation ID. On the right side, there are links for 'Go to resource' and 'Pin to dashboard'. A sidebar on the right provides links to 'Cost Management', 'Microsoft Defender for Cloud', 'Free Microsoft tutorials', and 'Work with an expert'. The status bar at the bottom right shows the date as 2/27/2024 and the time as 11:40 AM.

## Create container – projectcontainer1

The screenshot shows the Microsoft Azure portal interface. The left sidebar is for a storage account named 'projectadls'. The main area is titled 'Containers' and shows a list of existing containers: 'Logs' and 'Slogs'. A modal window titled 'New container' is open on the right, prompting for a container name. The input field contains 'projectcontainer1'. Below it, the 'Anonymous access level' dropdown is set to 'Private (no anonymous access)'. A note states: 'The access level is set to private because anonymous access is disabled on this storage account.' At the bottom of the modal are 'Create' and 'Give feedback' buttons.

## Create container-projectcontainer2

The screenshot shows the Microsoft Azure portal interface. The left sidebar is for a storage account named 'projectadls'. The main area is titled 'Containers' and shows a list of three containers: 'Logs', 'projectcontainer1', and 'projectcontainer2'. Each container has a timestamp of '2/27/2024, 11:40:34 AM' and an 'Anonymous access level' of 'Private'. The 'Lease state' column shows 'Available' for all three. The bottom of the screen shows the Windows taskbar with various pinned icons.

Name	Last modified	Anonymous access level	Lease state
Logs	2/27/2024, 11:40:34 AM	Private	Available
projectcontainer1	2/27/2024, 11:41:19 AM	Private	Available
projectcontainer2	2/27/2024, 11:41:37 AM	Private	Available

# Upload sample data in projectcontainer1

The screenshot shows the Microsoft Azure Storage Container blade for the container 'projectcontainer1'. At the top, there's a success message: 'Successfully uploaded blob(s)' and 'Successfully uploaded 1 blob(s)'. Below this, there are standard blob management actions: Upload, Add Directory, Refresh, Rename, Delete, Change tier, Acquire lease, Break lease, and Give feedback. The 'Overview' tab is selected. The main content area displays blob details with columns: Name, Modified, Access tier, Archive status, Blob type, Size, and Lease state. Two blobs are listed: 'annual-enterprise-su...' and 'environmental-prote...'. A 'Search blobs by prefix (case-sensitive)' input field and a 'Show deleted objects' toggle switch are also present. On the left, a sidebar lists container settings like Shared access tokens, Manage ACL, Access policy, Properties, and Metadata.

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
annual-enterprise-su...	2/27/2024, 11:47:13 ...	Hot (Inferred)		Block blob	1.42 MiB	Available
environmental-prote...	2/27/2024, 11:47:29 ...	Hot (Inferred)		Block blob	67.3 KiB	Available



## Sample data

## Annual-Enterprise-Survey-2021

1	year	industry	industry_r	rme_size	variable	value	unit			
2	2011	A	Agricultura_0	Activity un	46134	COUNT				
3	2011	A	Agricultura_0	Rolling me	0	COUNT				
4	2011	A	Agricultura_0	Salaries ari	279	DOLLARS(millions)				
5	2011	A	Agricultura_0	Sales, gov	8187	DOLLARS(millions)				
6	2011	A	Agricultura_0	Total inco	8866	DOLLARS(millions)				
7	2011	A	Agricultura_0	Total expen	7618	DOLLARS(millions)				
8	2011	A	Agricultura_0	Operating	770	DOLLARS(millions)				
9	2011	A	Agricultura_0	Total asse	55700	DOLLARS(millions)				
10	2011	A	Agricultura_0	Fixed tang	32155	DOLLARS(millions)				
11	2011	A	Agricultur b_1-5	Activity un	21777	COUNT				
12	2011	A	Agricultur b_1-5	Rolling me	38136	COUNT				
13	2011	A	Agricultur b_1-5	Salaries ari	1435	DOLLARS(millions)				
14	2011	A	Agricultur b_1-5	Sales, gov	13359	DOLLARS(millions)				
15	2011	A	Agricultur b_1-5	Total inco	13771	DOLLARS(millions)				
16	2011	A	Agricultur b_1-5	Total expen	12316	DOLLARS(millions)				
17	2011	A	Agricultur b_1-5	Operating	1247	DOLLARS(millions)				
18	2011	A	Agricultur b_1-5	Total asse	52666	DOLLARS(millions)				
19	2011	A	Agricultur b_1-5	Fixed tang	31235	DOLLARS(millions)				
20	2011	A	Agricultur c_6-9	Activity un	1965	COUNT				
21	2011	A	Agricultur c_6-9	Rolling me	13848	COUNT				

## Environmental-Protection-Expend

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	year	sector	class	cfn_tle1	cfn_tle2	units	magnitude	source	data_valu	flag			
2	2009	Central gc Total		Environment	Final cons	Proportion	Actual	Environment		1.2 F			
3	2010	Central gc Total		Environment	Final cons	Proportion	Actual	Environment		1.2 F			
4	2011	Central gc Total		Environment	Final cons	Proportion	Actual	Environment		1.1 F			
5	2012	Central gc Total		Environment	Final cons	Proportion	Actual	Environment		1.1 F			
6	2013	Central gc Total		Environment	Final cons	Proportion	Actual	Environment		1.2 F			
7	2014	Central gc Total		Environment	Final cons	Proportion	Actual	Environment		1.1 F			
8	2015	Central gc Total		Environment	Final cons	Proportion	Actual	Environment		1.2 F			
9	2016	Central gc Total		Environment	Final cons	Proportion	Actual	Environment		1.1 F			
10	2017	Central gc Total		Environment	Final cons	Proportion	Actual	Environment		1.2 F			
11	2018	Central gc Total		Environment	Final cons	Proportion	Actual	Environment		1.2 F			
12	2019	Central gc Total		Environment	Final cons	Proportion	Actual	Environment		1.1 F			
13	2020	Central gc Total		Environment	Final cons	Proportion	Actual	Environment		1.2 P			
14	2021	Central gc Total		Environment	Final cons	Proportion	Actual	Environment		1.2 P			
15	2009	Central gc Total		Environment	Final cons	Dollars	Millions	Environment		408 F			
16	2010	Central gc Total		Environment	Final cons	Dollars	Millions	Environment		420 F			
17	2011	Central gc Total		Environment	Final cons	Dollars	Millions	Environment		399 F			
18	2012	Central gc Total		Environment	Final cons	Dollars	Millions	Environment		411 F			
19	2013	Central gc Total		Environment	Final cons	Dollars	Millions	Environment		438 F			
20	2014	Central gc Total		Environment	Final cons	Dollars	Millions	Environment		440 F			
21	2015	Central gc Total		Environment	Final cons	Dollars	Millions	Environment		465 F			

## Create Azure data factory

The screenshot shows the Microsoft Azure portal with the URL [portal.azure.com/#create/Microsoft.DataFactory](https://portal.azure.com/#create/Microsoft.DataFactory). The page title is "Create Data Factory". The top navigation bar includes "Microsoft Azure", "Search resources, services, and docs (G+)", and user information. The main content area has tabs for "Basics", "Git configuration", "Networking", "Advanced", "Tags", and "Review + create". The "Review + create" tab is currently selected. Below the tabs, there is a "View automation template" link. A "TERMS" section contains a legal agreement about creating a data factory. The "Basics" section shows the following details:

Subscription	Azure subscription 1
Resource group	rg-azuser1051_mml.local-MUqzI
Name	hex-deg-1062

At the bottom, there are "Previous", "Next", and "Create" buttons, along with a "Give feedback" link. The status bar at the bottom right shows the date and time: 11:43 AM 2/27/2024.

# Azure data factory created

The screenshot shows the Microsoft Azure portal with a deployment status message. The message indicates that the deployment 'Microsoft.DataFactory-20240227114204' has succeeded. It provides details such as the deployment name, subscription, resource group, start time (2/27/2024, 11:43:09 AM), and correlation ID. There are links to pin the deployment to the dashboard and go to the resource group.

**Deployment succeeded**  
Deployment 'Microsoft.DataFactory-20240227114204' to resource group 'rg-azuser1051\_mml.local-MUqzl' was successful.

[Pin to dashboard](#) [Go to resource group](#)

**Your deployment is complete**

Deployment name : Microsoft.DataFactory-20240227114... Start time : 2/27/2024, 11:43:09 AM  
Subscription : Azure subscription 1 Correlation ID : a2595813-52d3-46e9-a05e-b904817...  
Resource group : rg-azuser1051\_mml.local-MUqzl

**Deployment details**

**Next steps**

[Go to resource](#)

**Give feedback**  
Tell us about your experience with deployment

**Cost management**  
Get notified to stay within your budget and prevent unexpected charges on your bill.  
[Set up cost alerts >](#)

**Microsoft Defender for Cloud**  
Secure your apps and infrastructure  
[Go to Microsoft Defender for Cloud >](#)

**Free Microsoft tutorials**  
[Start learning today >](#)

**Work with an expert**  
Azure experts are service provider partners

25°C Haze

The screenshot shows the Azure Data Factory web interface. The page title is 'hex-deg-1062 - Microsoft Azure' and the sub-page title is 'Azure Data Factory'. The main content area displays a large blue box with the text 'Azure Data Factory' and 'Loading' below it.



## Select copy data task type and configure task schedule

### Built in copy task

The screenshot shows the 'Copy Data tool' wizard in Microsoft Azure Data Factory. The left sidebar lists steps: Properties, Source, Destination, Settings, and Review and finish. The main area is titled 'Properties' and describes using the Copy Data Tool to perform one-time or scheduled data loads from 90+ data sources. It highlights the wizard experience for specifying loading settings and generating artifacts like pipelines, datasets, and linked services. Two task types are shown:

- Built-in copy task**: You will get single pipeline to copy data from 90+ data source easily.
- Metadata-driven copy task**: You will get parameterized pipelines which can read metadata from an external store to load data at a large scale.

Below these, a note says: "You will get single pipeline to quickly copy objects from data source store to destination in a very intuitive manner." The 'Task cadence or task schedule' dropdown is set to "Run once now".

### Specify the source data store for copy task

### Azure Data Lake Storage Gen2

The screenshot shows the 'Copy Data tool' wizard in Microsoft Azure Data Factory, currently on the 'Source' step. The left sidebar shows steps: Properties, Source, Dataset, Configuration, Destination, Settings, and Review and finish. The main area is titled 'Source data store' and asks to specify the source data store for the copy task, using an existing connection or creating a new one. It shows the 'Source type' as 'Azure Data Lake Storage Gen2' and the 'Connection' as 'AzureDataLakeStorage1'. Under 'Options', 'Binary copy' is checked. Other options include 'Compression type' (None), 'Recursively' (checked), and 'Delete files after completion' (unchecked). The 'Max concurrent connections' field is empty.

Create new connection and also specify storage account name

Copy Data tool

Source data store

Specify the source data store for the copy task. You can use an existing data store or create a new one.

Source type: Azure Data Lake Storage Gen2

Connection: Select...

New connection

Azure Data Lake Storage Gen2

Authentication type: Account key

Account selection method: From Azure subscription

Azure subscription: Azure subscription 1 (984f097c-963c-4eb6-a20d-839457ae9f08)

Storage account name: projectadls

Test connection: To linked service (selected) To file path

Annotations: + New

Parameters: > Parameters

Create Cancel

Specify the destination data store for copy task

## Azure Data Lake Storage Gen2

Copy Data tool

Destination data store

Specify the destination data store for the copy task. You can use an existing data store connection or specify a new data store.

Destination type: Azure Data Lake Storage Gen2

Connection: AzureDataLakeStorage2

Compression type: None

Copy behavior: Select...

Max concurrent connections:

Block size (MB):

Metadata: + New

< Previous Next >

Cancel

Create new connection and also specify storage account name

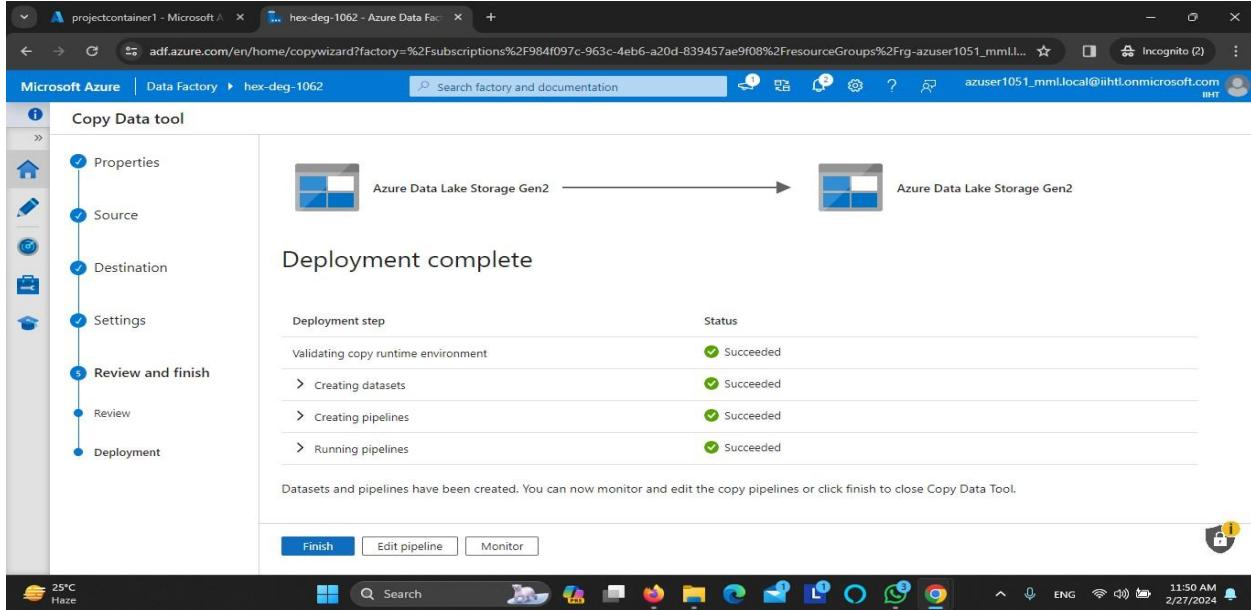
The screenshot shows the 'Copy Data tool' interface in the Azure Data Factory. On the left, a navigation pane lists steps: Properties, Source, Destination, Dataset, Configuration, Settings, and Review and finish. The 'Destination' step is selected. The main panel is titled 'Destination data store' and asks to specify the destination data store for the copy task. It shows 'Destination type' as 'Azure Data Lake Storage Gen2' and 'Connection' as 'Select...'. To the right, the 'New connection' configuration pane is open, showing 'Azure Data Lake Storage Gen2' selected under 'Destination type'. The 'Authentication type' is set to 'Account key'. Under 'Account selection method', 'From Azure subscription' is selected. A dropdown menu shows 'Azure subscription 1 (984f097c-963c-4eb6-a20d-839457ae9f08)'. The 'Storage account name' is set to 'projectadls'. A 'Test connection' button is present, with a success message 'Connection successful' and a 'Test connection' link. At the bottom of the configuration pane are 'Create' and 'Cancel' buttons.

Enter copy data task name

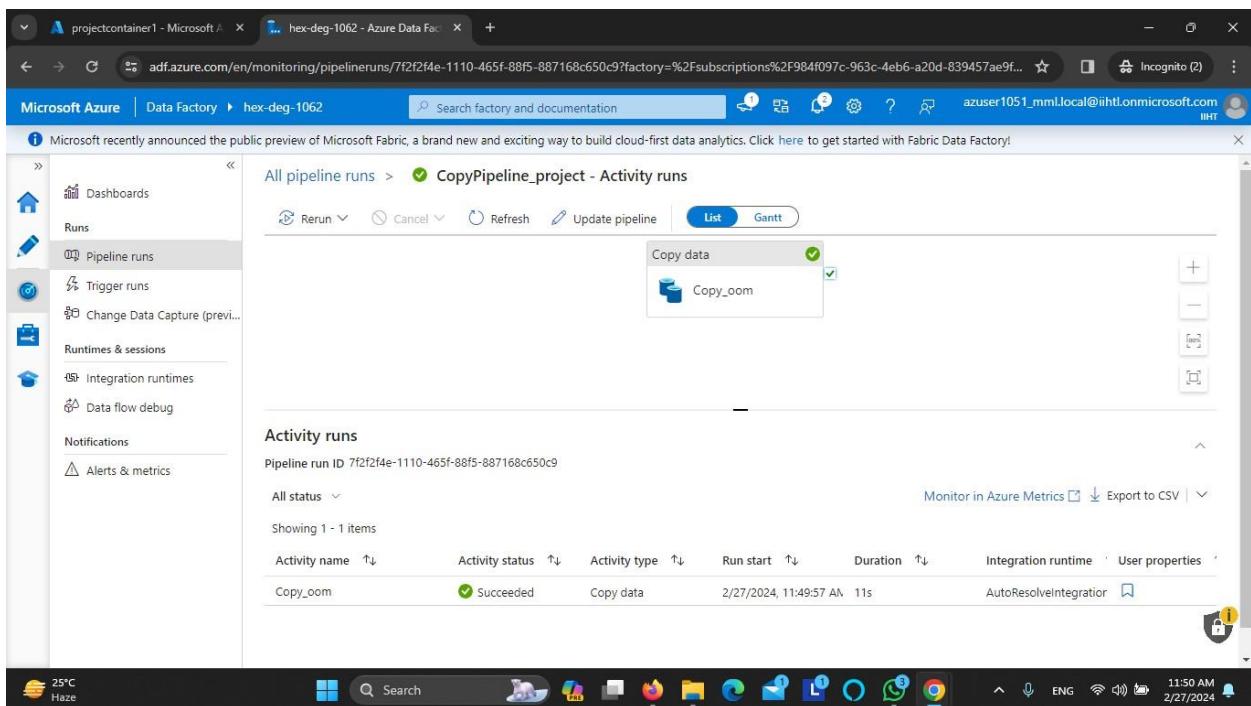
CopyPipeline\_project

The screenshot shows the 'Copy Data tool' interface in the Azure Data Factory. The navigation pane shows the 'Settings' step selected. The main panel is titled 'Settings' and prompts to enter a name and description for the copy data task. The 'Task name' field is filled with 'CopyPipeline\_project'. Below it, 'Task description' is empty. Under 'Data consistency verification', there is a checkbox which is unchecked. Under 'Fault tolerance', a dropdown menu shows 'Skip missing files'. Under 'Enable logging' and 'Enable staging', there are checkboxes which are unchecked. An 'Advanced' section is collapsed. At the bottom of the settings pane are 'Previous' and 'Next' buttons. The status bar at the bottom of the screen shows '25°C Haze' and the date '2/27/2024'.

# Pipeline created to copy the data task from Azure Data Lake Storage Gen2 to Azure Data Lake Storage Gen2



Activity succeeded and pipeline ingested



## Data copied in projectcontainer2

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
annual-enterprise-su...	2/27/2024, 11:50:07 ...	Hot (Inferred)		Block blob	1.42 MiB	Available
environmental-prote...	2/27/2024, 11:50:07 ...	Hot (Inferred)		Block blob	67.3 KIB	Available

## 2 . Create a Pipeline to copy the data task from Azure Data Lake Storage Gen2 to Azure Blob Storage

Steps:

Login in to Azure Portal create a Storage Account

# Storage Account Name - projectadls1

Storage account name: projectadls1

Region: (US) East US 2

Performance: Standard

Redundancy: Geo-redundant storage (GRS)

Make read access to data available in the event of regional unavailability.

## Enable hierarchical namespace

Hierarchical namespace: Enabled

Default access tier: Hot

Blob anonymous access: Disabled

Blob soft delete: Enabled (7 days)

Container soft delete: Enabled (7 days)

Versioning: Disabled

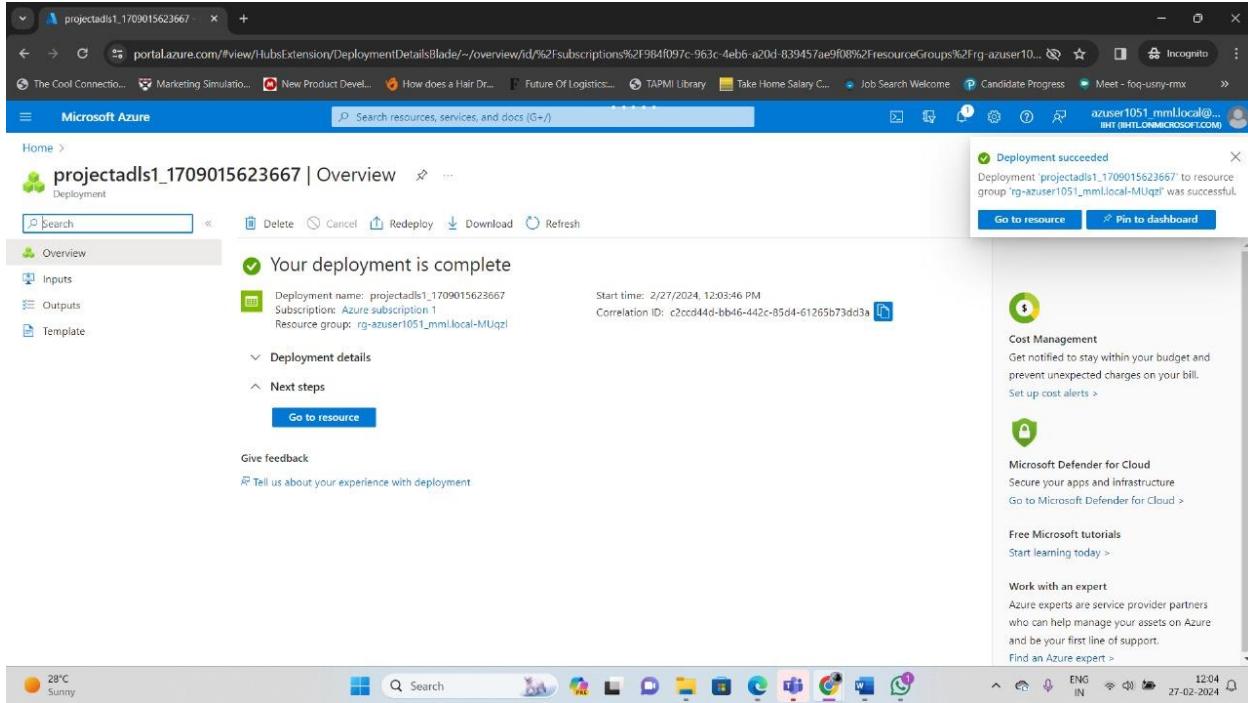
Require secure transfer for REST API operations: Enabled

Storage account key access: Enabled

Minimum TLS version: Version 1.2

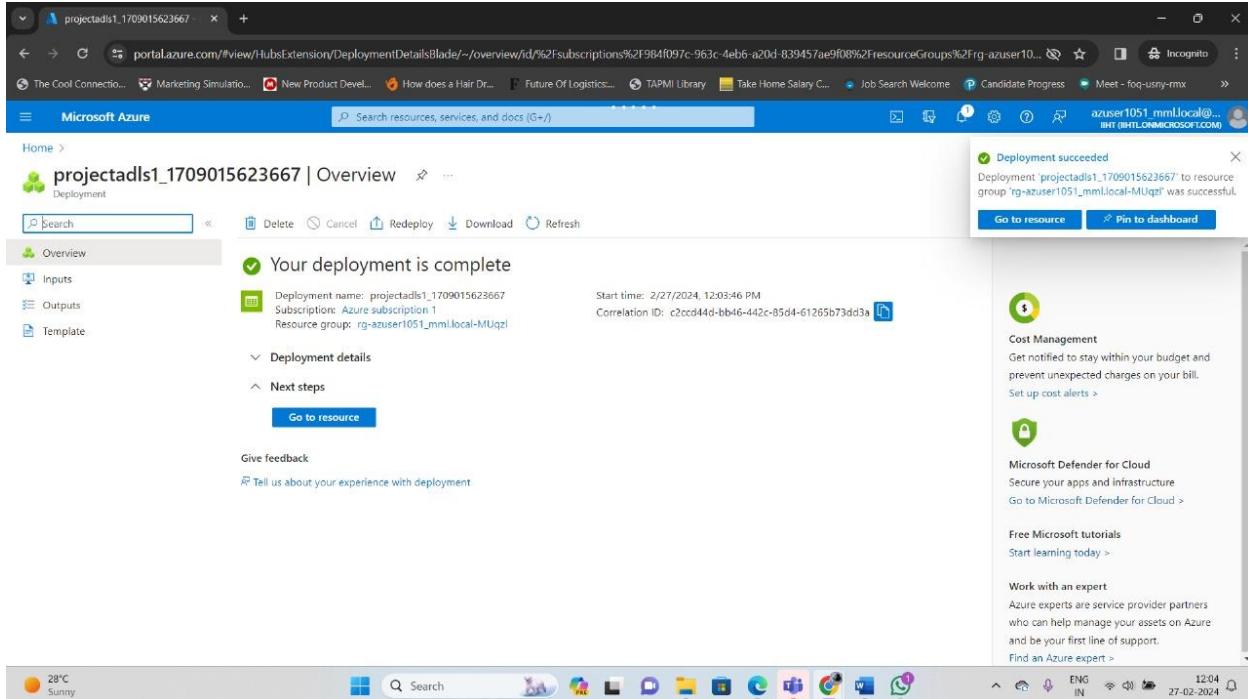
Infrastructure encryption: Disabled

# Storage account created



## Create Storage account

Storage account name - projectadls2



## Storage account created

The screenshot shows the Microsoft Azure portal interface. The main title bar reads "projectadls2\_1709015779382 | Overview". On the left, there's a navigation sidebar with "Overview", "Inputs", "Outputs", and "Template" options. The main content area displays a green checkmark icon and the message "Your deployment is complete". Below this, it shows deployment details: "Deployment name: projectadls2\_1709015779382", "Subscription: Azure subscription 1", and "Resource group: rg-azuser1051\_mml.local-MUqzI". To the right, there's a success message: "Deployment succeeded" and "Deployment 'projectadls2\_1709015779382' to resource group 'rg-azuser1051\_mml.local-MUqzI' was successful". There are "Go to resource" and "Pin to dashboard" buttons. On the far right, there are sections for "Cost Management", "Microsoft Defender for Cloud", "Free Microsoft tutorials", and "Work with an expert". The bottom status bar shows the date and time as "27-02-2024 12:06".

## Create container container-1 in projectadls1

The screenshot shows the Microsoft Azure portal interface. The main title bar reads "projectadls1 - Microsoft Azure". The left sidebar has "Containers" selected under "Data storage". The main content area shows a table of existing containers: one entry for "\$logs" and another for "container-1". A search bar at the top says "Search containers by prefix". The bottom status bar shows the date and time as "27-02-2024 12:05".

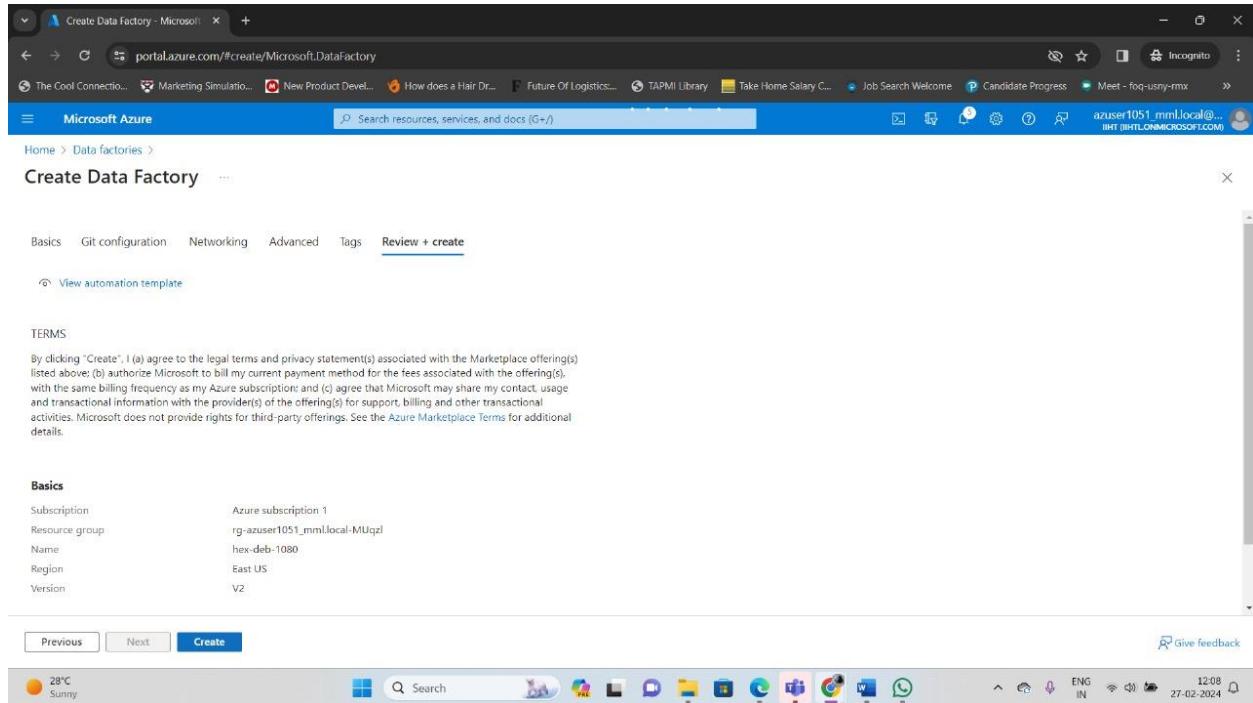
## Upload sample data

The screenshot shows the Microsoft Azure Storage Container blade for 'container-1' in the 'projectadls1' storage account. The left sidebar shows navigation options like Overview, Diagnose and solve problems, Access Control (IAM), Settings, Shared access tokens, Manage ACL, Access policy, Properties, and Metadata. The main area displays blob details with two entries: 'image (2).png' and 'image (3).png'. A success message at the top right indicates 'Successfully uploaded blob(s)' and 'Successfully uploaded 2 blob(s)'. The taskbar at the bottom shows various pinned icons.

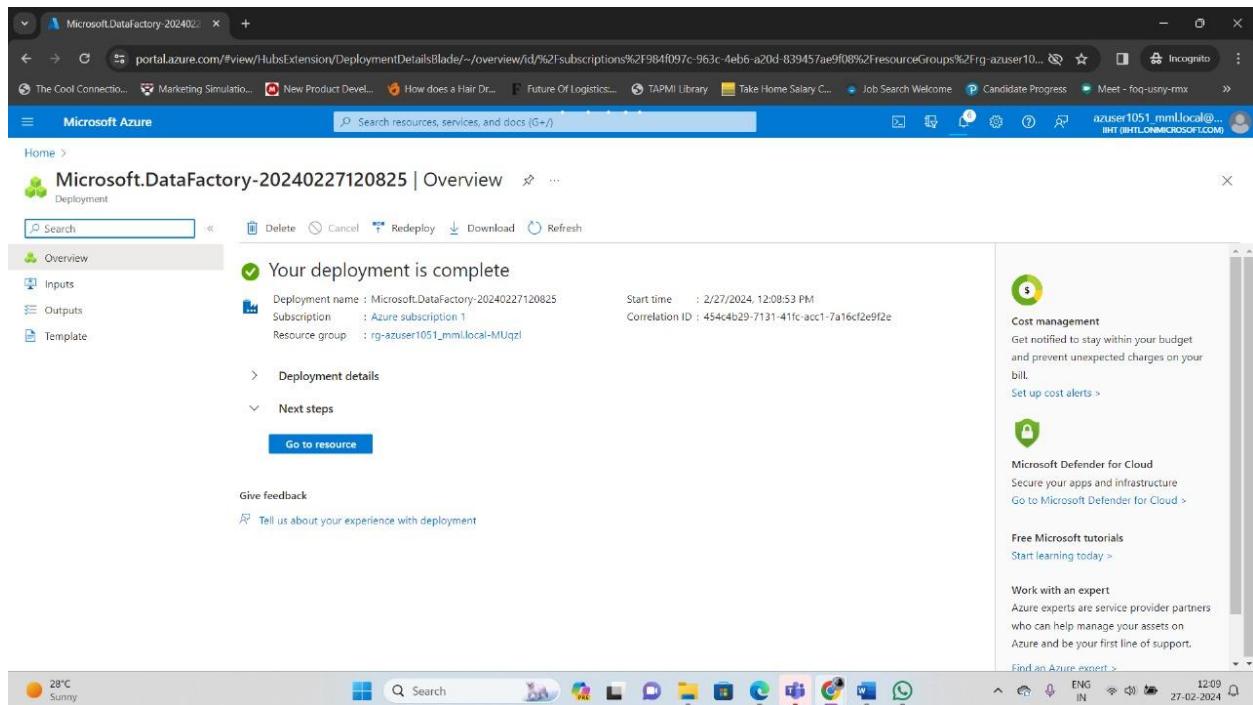
## Create container-2 in projectadls2

The screenshot shows the Microsoft Azure Storage Container blade for 'projectadls2' in the 'projectadls2' storage account. The left sidebar shows navigation options like Overview, Activity log, Tags, Diagnose and solve problems, Access Control (IAM), Data migration, Events, Storage browser, Storage Mover, Data storage (Containers, File shares, Queues, Tables), and Security + networking (Networking). The main area displays container details with two entries: '\$logs' and 'container-2'. A success message at the top right indicates 'Successfully created storage container' and 'Successfully created storage container 'container-2''. The taskbar at the bottom shows various pinned icons.

# Create azure data factory

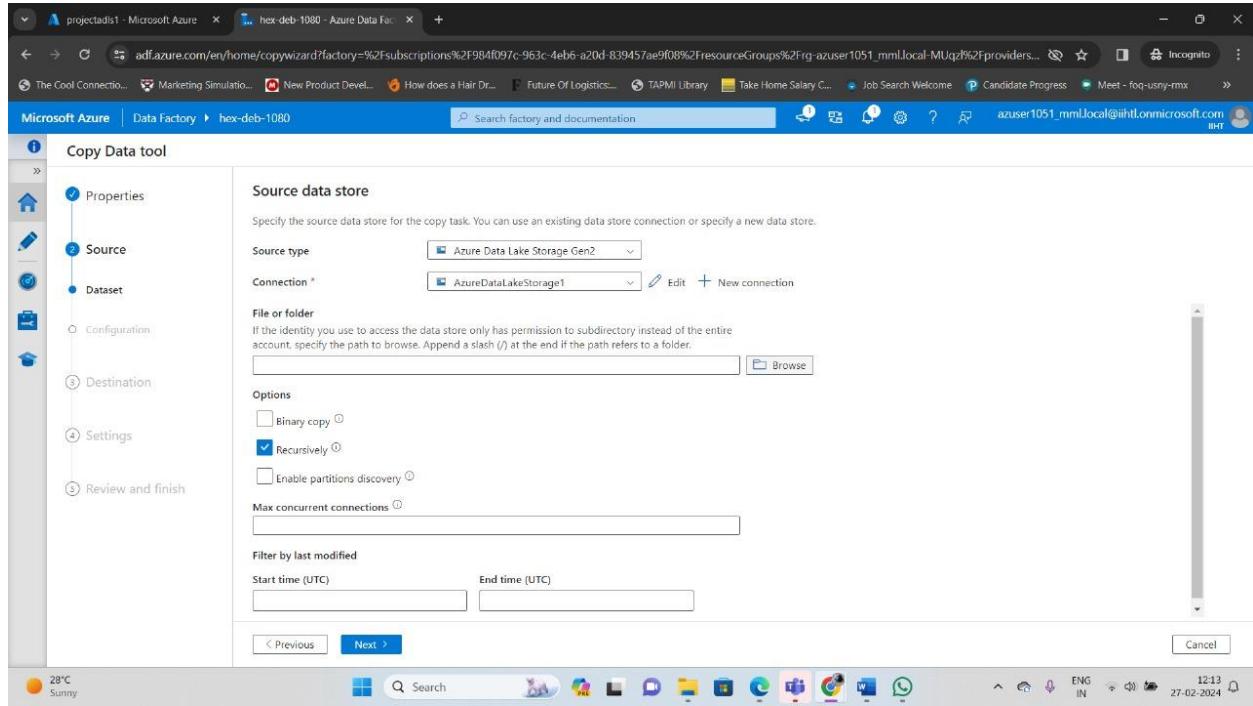


## Azure data factory created

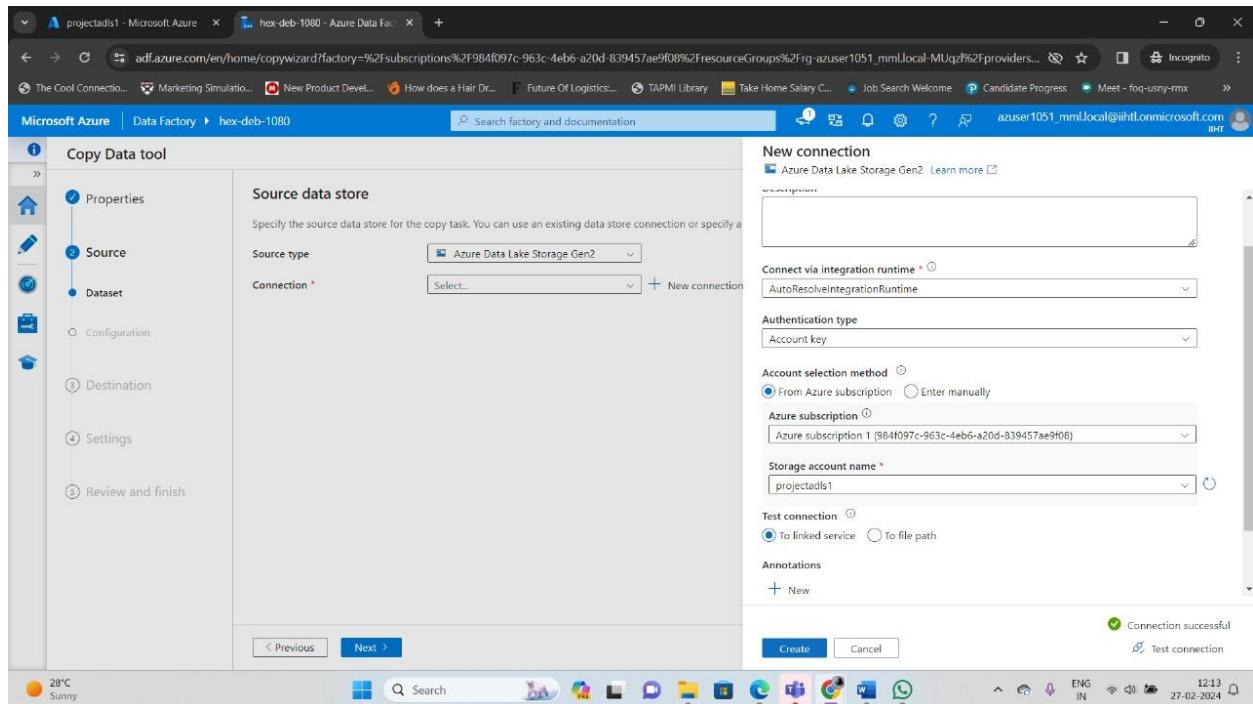


## Specify the source data store for copy task

### Azure Data Lake Storage Gen2

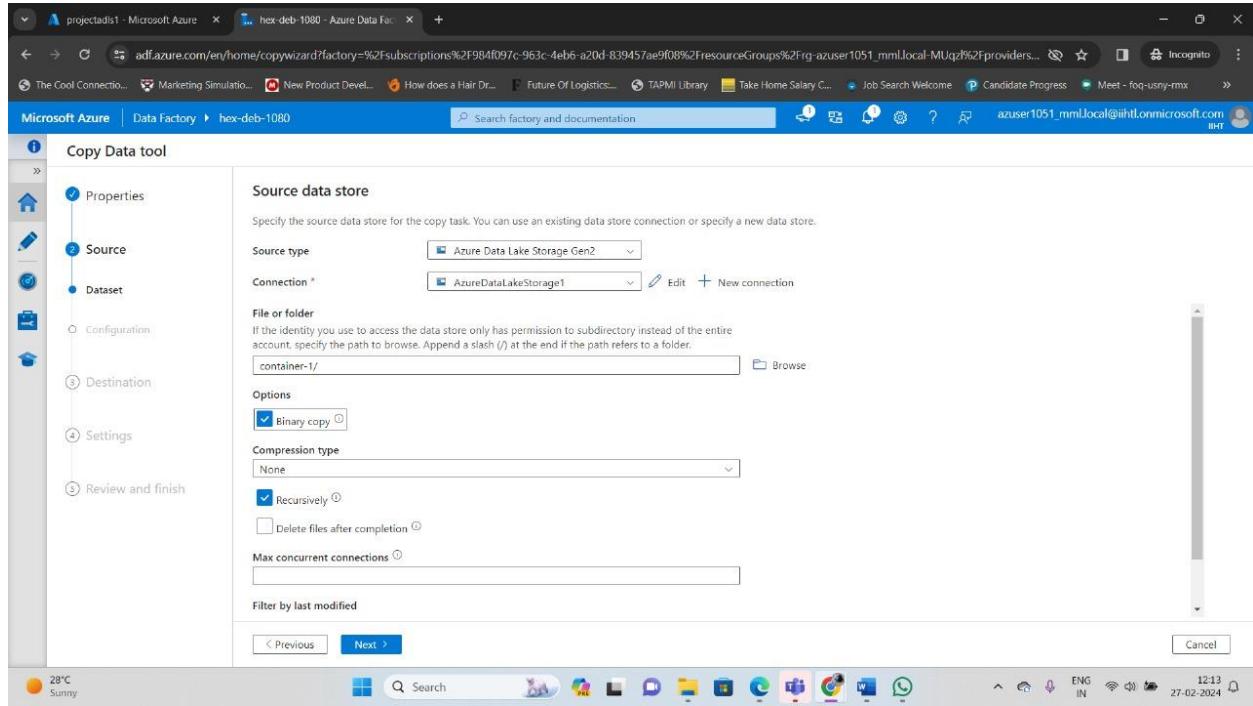


### Create connection and specify storage account name

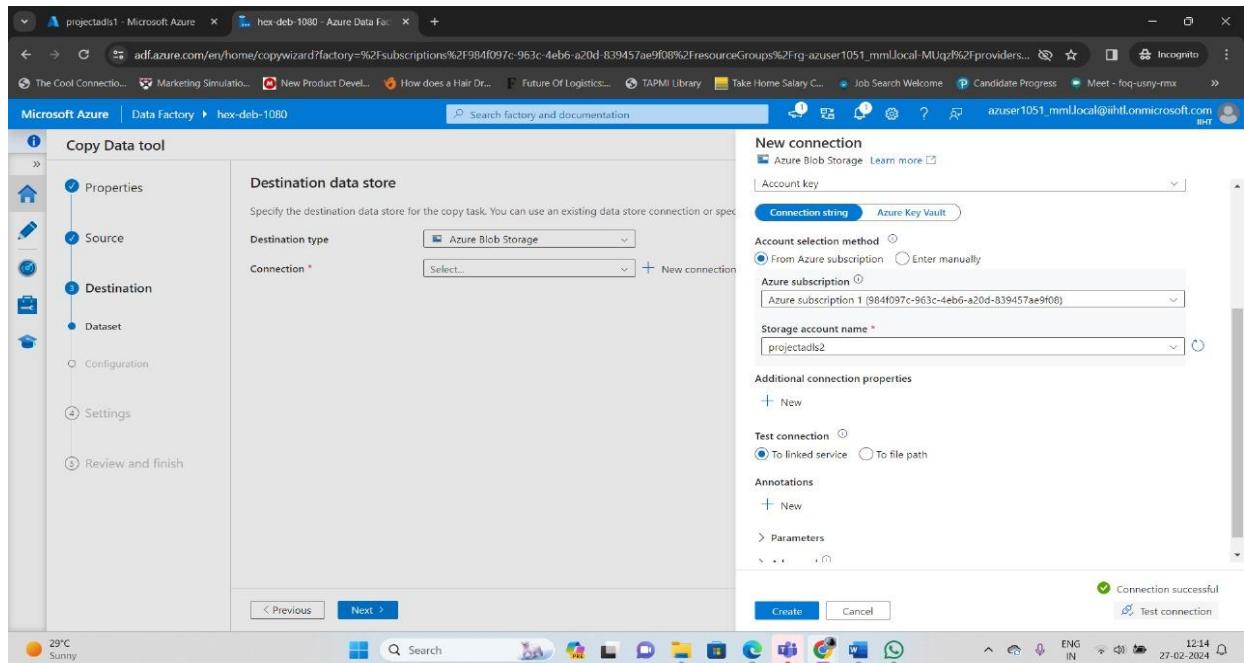


## Specify the destination data store for the copy task

### AzureDataLakeStorage1



## Create connection and specify storage account name



Enter data task name

## CopyPipeline\_project

The screenshot shows the 'Copy Data tool' settings page in Microsoft Azure Data Factory. The task name is set to 'CopyPipeline\_project'. The 'Source' section is selected in the left sidebar. The right pane displays settings for the copy task, including 'Task name' (CopyPipeline\_project), 'Task description' (empty), 'Data consistency verification' (unchecked), 'Fault tolerance' (set to 'Skip missing files'), 'Enable logging' (unchecked), 'Enable staging' (unchecked), and an 'Advanced' section. Navigation buttons at the bottom include '< Previous', 'Next >', and 'Cancel'.

Running pipeline to copy data from

## Azure data lake storage gen2 to Azure blob storage

The screenshot shows the 'Copy Data tool' summary page in Microsoft Azure Data Factory. It indicates that a pipeline is running to copy data from Azure Data Lake Storage Gen2 to Azure Blob Storage. The flow diagram shows 'Azure Data Lake Storage Gen2' connected to 'Azure Blob Storage'. The 'Properties' section details the task name ('CopyPipeline\_project'), source connection ('AzureDataLakeStorage1'), source dataset ('SourceDataset\_a3p'), destination connection ('AzureBlobStorage1'), and destination dataset ('DestinationDataset\_a3p'). Navigation buttons at the bottom include '< Previous', 'Next >', and 'Cancel'.

# Activity succeeded and pipeline ingested

The screenshot shows the Microsoft Azure Data Factory interface. The left sidebar has a 'Pipeline runs' section selected. The main area displays a 'CopyPipeline\_project - Activity runs' section with a table titled 'Activity runs'. The table shows one item: 'Copy\_a3p' with status 'Succeeded', activity type 'Copy data', run start '2/27/2024, 12:16:09 PM', duration '58s', integration runtime 'AutoResolveIntegration', user properties 'None', and destination 'container-2://'. A message at the top states: 'Pipeline was modified after this run. The current pipeline configuration is shown.'

Data copied in container-2

The screenshot shows the Microsoft Azure Storage Container interface for 'container-2'. The left sidebar has an 'Overview' section selected. The main area shows a table of blobs. There are two entries: 'image (2).png' and 'image (3).png'. Both were modified on '2/27/2024, 12:17:06 ...' and have an 'Access tier' of 'Hot (Inferred)'. They are both 'Block blob' types with sizes of 208.58 KiB and 153.91 KiB, both marked as 'Available'.

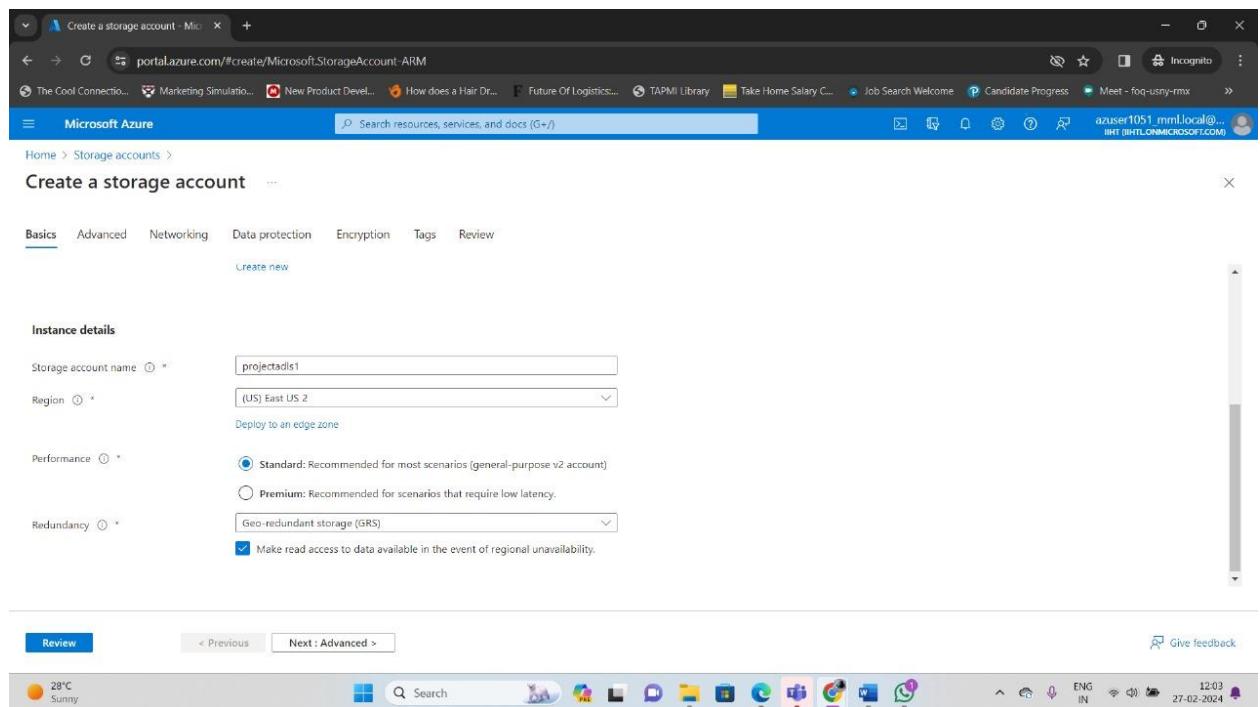
Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
image (2).png	2/27/2024, 12:17:06 ...	Hot (Inferred)		Block blob	208.58 KiB	Available
image (3).png	2/27/2024, 12:17:06 ...	Hot (Inferred)		Block blob	153.91 KiB	Available

## **2 . Create a Pipeline to copy the data task from Azure Blob Storage to Azure Data Lake Storage Gen2**

Steps:

Login in to Azure Portal create a Storage Account

Storage Account Name - projectadls1



# Enable hierarchical namespace

The screenshot shows the Microsoft Azure portal with the URL [https://portal.azure.com/#@ihill.onmicrosoft.com/resource/subscriptions/984f097c-963c-4eb6-a20d-839457ae9f08/resourcegroups/rq\\_azuser1051\\_mmilocal-MUqzl/providers/Microsoft...](https://portal.azure.com/#@ihill.onmicrosoft.com/resource/subscriptions/984f097c-963c-4eb6-a20d-839457ae9f08/resourcegroups/rq_azuser1051_mmilocal-MUqzl/providers/Microsoft...). The page displays the overview of a storage account named 'projectadls1'. In the 'Data storage' section, under 'Data Lake Storage', the 'Hierarchical namespace' setting is listed as 'Enabled'. Other settings shown include 'Default access tier' (Hot), 'Blob anonymous access' (Disabled), 'Blob soft delete' (Enabled (7 days)), 'Container soft delete' (Enabled (7 days)), and 'Versioning' (Disabled). The 'Security' section shows 'Require secure transfer for REST API operations' and 'Storage account key access' both set to 'Enabled'. The 'Networking' section shows 'Minimum TLS version' as 'Version 1.2' and 'Infrastructure encryption' as 'Disabled'.

## Storage account created

The screenshot shows the Microsoft Azure portal with the URL [https://portal.azure.com/#view/HubsExtension/DeploymentDetailsBlade/-/overview/id/%2Fsubscriptions%2F984f097c-963c-4eb6-a20d-839457ae9f08%2FresourceGroups%2Frq\\_azuser1051\\_mmilocal-MUqzl/deploymentId/1709015623667](https://portal.azure.com/#view/HubsExtension/DeploymentDetailsBlade/-/overview/id/%2Fsubscriptions%2F984f097c-963c-4eb6-a20d-839457ae9f08%2FresourceGroups%2Frq_azuser1051_mmilocal-MUqzl/deploymentId/1709015623667). The page displays deployment details for 'projectadls1\_1709015623667'. The status is 'Deployment succeeded'. Deployment information includes: Deployment name: projectadls1\_1709015623667, Subscription: Azure subscription 1, Resource group: rg-azuser1051\_mmilocal-MUqzl. The deployment started at 12:03:46 PM on 2/27/2024. A 'Go to resource' button is present. On the right side, there are promotional cards for 'Cost Management', 'Microsoft Defender for Cloud', 'Free Microsoft tutorials', and 'Work with an expert'.

# Create Storage account

Storage account name - projectadls2

The screenshot shows the Microsoft Azure portal with the URL [https://portal.azure.com/#view/HubsExtension/DeploymentDetailsBlade/~/overview/id/%2Fsubscriptions%2F984f097c-963c-4eb6-a20d-839457ae9f08%2FresourceGroups%2Frsg-azuser1051\\_mml.local-MUqzl](https://portal.azure.com/#view/HubsExtension/DeploymentDetailsBlade/~/overview/id/%2Fsubscriptions%2F984f097c-963c-4eb6-a20d-839457ae9f08%2FresourceGroups%2Frsg-azuser1051_mml.local-MUqzl). The page displays a deployment summary for 'projectadls1\_1709015623667'. Key details include:

- Deployment name:** projectadls1\_1709015623667
- Subscription:** Azure subscription 1
- Resource group:** rg-azuser1051\_mml.local-MUqzl
- Start time:** 2/27/2024, 12:03:46 PM
- Correlation ID:** c2cd44d-bb46-442c-85d4-61265b73dd3a

The deployment status is marked as "Deployment succeeded". A message states: "Deployment 'projectadls1\_1709015623667' to resource group 'rg-azuser1051\_mml.local-MUqzl' was successful." There are links to "Go to resource" and "Pin to dashboard".

The system tray at the bottom shows the date as 27-02-2024 and the time as 12:04.

Storage account created

The screenshot shows the Microsoft Azure portal with the URL [https://portal.azure.com/#view/HubsExtension/DeploymentDetailsBlade/~/overview/id/%2Fsubscriptions%2F984f097c-963c-4eb6-a20d-839457ae9f08%2FresourceGroups%2Frsg-azuser1051\\_mml.local-MUqzl](https://portal.azure.com/#view/HubsExtension/DeploymentDetailsBlade/~/overview/id/%2Fsubscriptions%2F984f097c-963c-4eb6-a20d-839457ae9f08%2FresourceGroups%2Frsg-azuser1051_mml.local-MUqzl). The page displays a deployment summary for 'projectadls2\_1709015779382'. Key details include:

- Deployment name:** projectadls2\_1709015779382
- Subscription:** Azure subscription 1
- Resource group:** rg-azuser1051\_mml.local-MUqzl
- Start time:** 2/27/2024, 12:06:21 PM
- Correlation ID:** 73d2645d-63ab-4060-bee0-80215679a4b6

The deployment status is marked as "Deployment succeeded". A message states: "Deployment 'projectadls2\_1709015779382' to resource group 'rg-azuser1051\_mml.local-MUqzl' was successful." There are links to "Go to resource" and "Pin to dashboard".

The system tray at the bottom shows the date as 27-02-2024 and the time as 12:06.

## Create container container-1 in projectadls1

The screenshot shows the Microsoft Azure portal interface for the storage account 'projectadls1'. The left sidebar navigation includes 'Overview', 'Activity log', 'Tags', 'Diagnose and solve problems', 'Access Control (IAM)', 'Data migration', 'Events', 'Storage browser', 'Data storage' (with 'Containers' selected), 'Security + networking', 'Networking', and 'Access keys'. The main content area displays a table of containers:

Name	Last modified	Anonymous access level	Lease state
\$logs	2/27/2024, 12:04:12 PM	Private	Available
container-1	2/27/2024, 12:04:56 PM	Private	Available

A search bar at the top says 'Search resources, services, and docs (G+)'. A message bar at the bottom right says 'azuser1051.mml.local@... (HTTP) (HTML) (MICROSOFT.COM)'.

## Create container-2 in projectadls2

The screenshot shows the Microsoft Azure portal interface for the storage account 'projectadls2'. The left sidebar navigation is identical to the previous screenshot. The main content area displays a table of containers:

Name	Last modified	Anonymous access level	Lease state
\$logs	2/27/2024, 12:06:46 PM	Private	Available
container-2	2/27/2024, 12:07:22 PM	Private	Available

A success message box in the top right corner says 'Successfully created storage container' and 'Successfully created storage container 'container-2''. A search bar at the top says 'Search resources, services, and docs (G+)'. A message bar at the bottom right says 'azuser1051.mml.local@... (HTTP) (HTML) (MICROSOFT.COM)'.

# Upload sample data

The screenshot shows the Microsoft Azure Storage Container Overview page for 'container-2'. The container has one blob named 'Adaptability (2).pptx'. The blob details are as follows:

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
Adaptability (2).pptx	2/27/2024, 12:21:33 ...	Hot (Inferred)		Block blob	12.44 MiB	Available



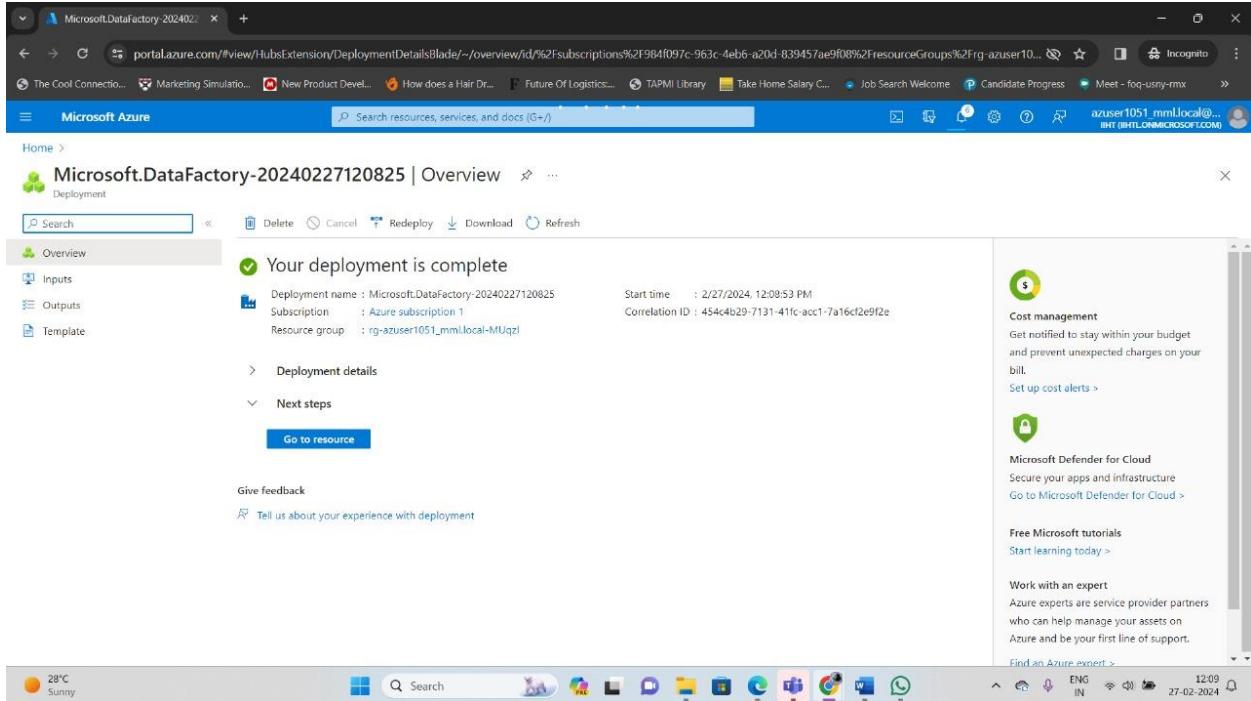
# Create azure data factory

The screenshot shows the Microsoft Azure Create Data Factory page in the 'Create Data Factory' wizard, currently on the 'Review + create' step. The 'Basics' tab is selected. The form fields are as follows:

Subscription	Azure subscription 1
Resource group	rg-azuser1051-mm.local-MUqz1
Name	hex-deb-1080
Region	East US
Version	V2

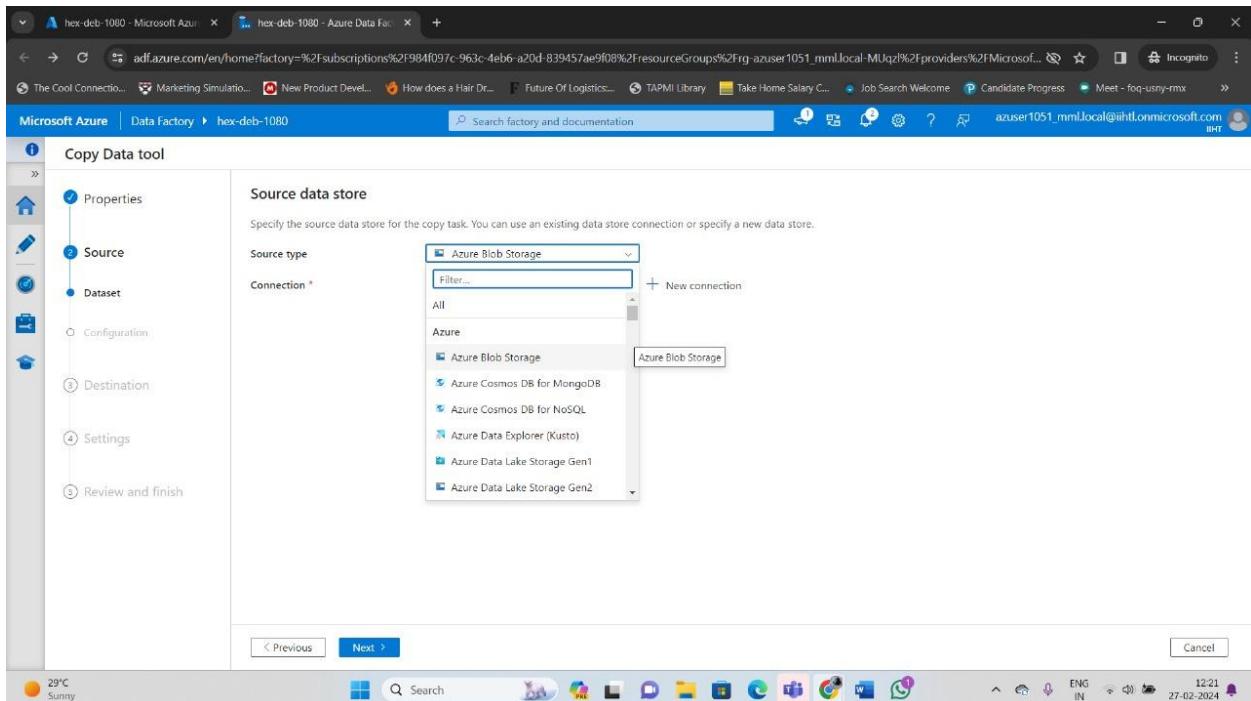
At the bottom, there are 'Previous', 'Next', and 'Create' buttons, along with a 'Give feedback' link.

# Azure data factory created

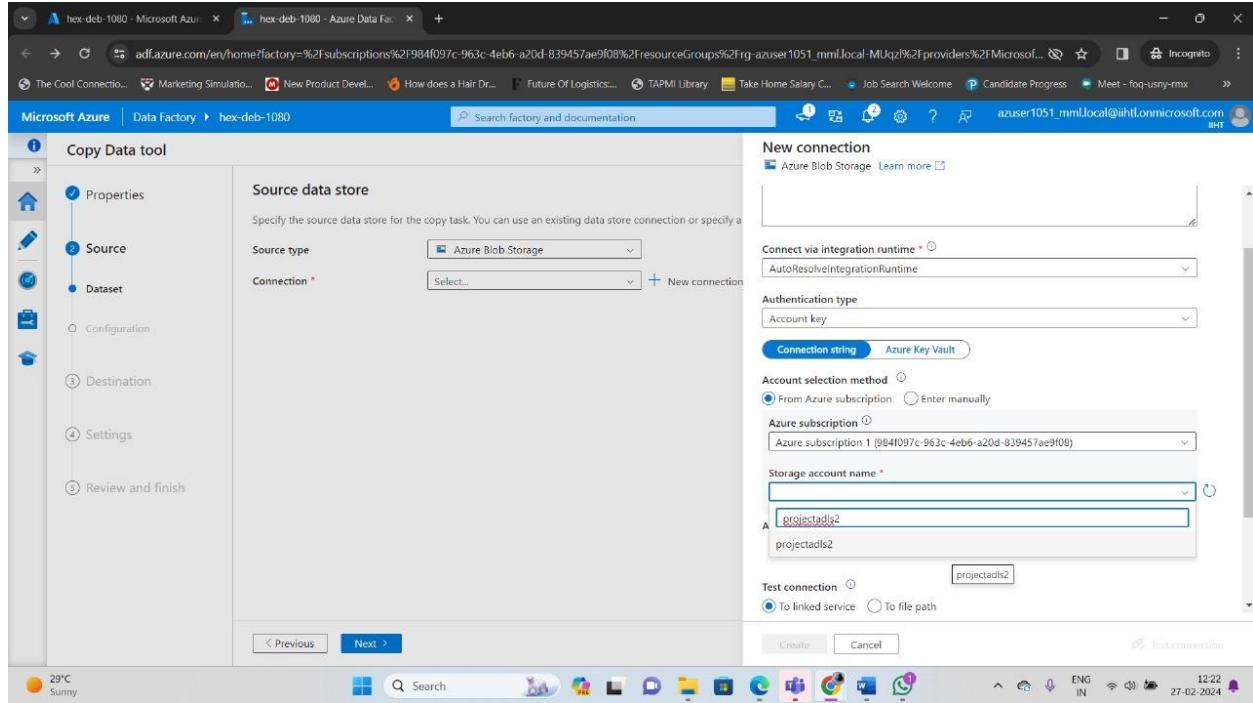


Specify the source data store for the copy task

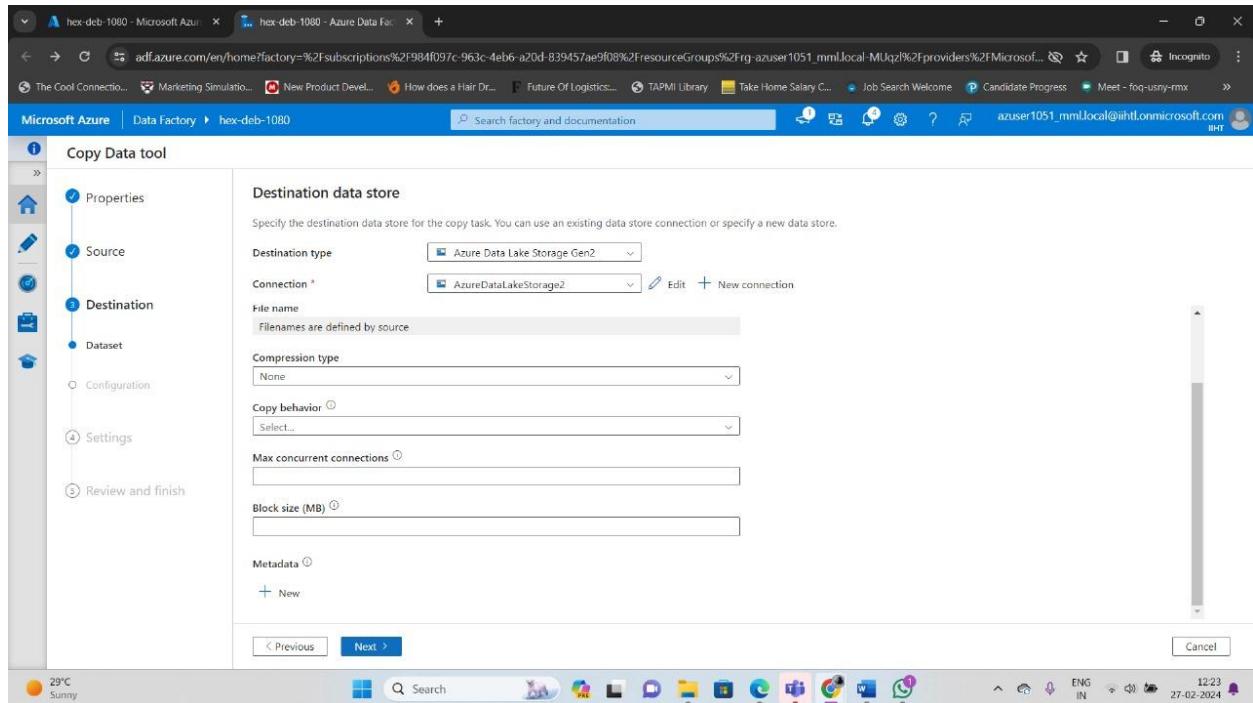
## Azure blob storage



## Create new connection and specify storage account name



## Specify the destination data store for the copy task



## Create connection and specify storage account name

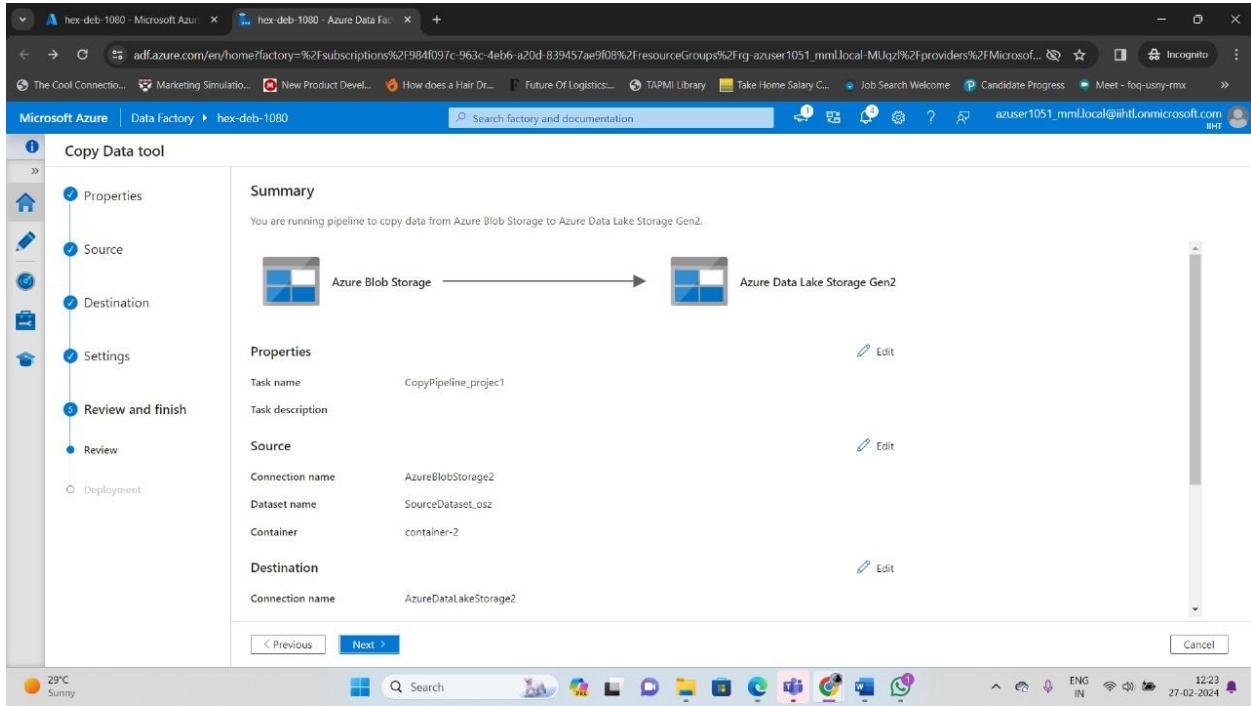
The screenshot shows the 'Destination data store' configuration step of the Copy Data tool. On the left, a sidebar lists steps: Properties, Source, Destination (selected), Dataset, Configuration, Settings, and Review and finish. The main area is titled 'Destination data store' with the sub-instruction 'Specify the destination data store for the copy task. You can use an existing data store connection or specify the connection settings here.' A dropdown 'Destination type' is set to 'Azure Data Lake Storage Gen2'. Below it, a 'Connection' dropdown is set to 'Select...' with a 'New connection' button. To the right, a 'New connection' panel is open, showing fields for 'Description' (empty), 'Connect via integration runtime' (set to 'AutoResolveIntegrationRuntime'), 'Authentication type' (set to 'Account key'), 'Account selection method' (radio button selected for 'From Azure subscription'), 'Azure subscription' (set to 'Azure subscription 1 (984f097c-963c-4eb6-a20d-839457ae9f08)'), and 'Storage account name' (set to 'projectadls1'). At the bottom of the panel are 'Test connection' and 'Create' buttons.

Enter the name of the copy data task

CopyPipeline\_project1

The screenshot shows the 'Settings' configuration step of the Copy Data tool. The sidebar shows steps: Properties, Source, Destination, Settings (selected), and Review and finish. The main area is titled 'Settings' with the sub-instruction 'Enter name and description for the copy data task, more options for data movement.' It includes fields for 'Task name' (set to 'CopyPipeline\_project1'), 'Task description' (empty), 'Data consistency verification' (unchecked), 'Fault tolerance' (set to 'Skip missing files'), 'Enable logging' (unchecked), 'Enable staging' (unchecked), and an 'Advanced' section. At the bottom are 'Previous' and 'Next >' buttons, and a 'Cancel' button on the right.

# Running pipeline to copy data from Azure Blob Storage to Azure Data Lake Storage Gen2

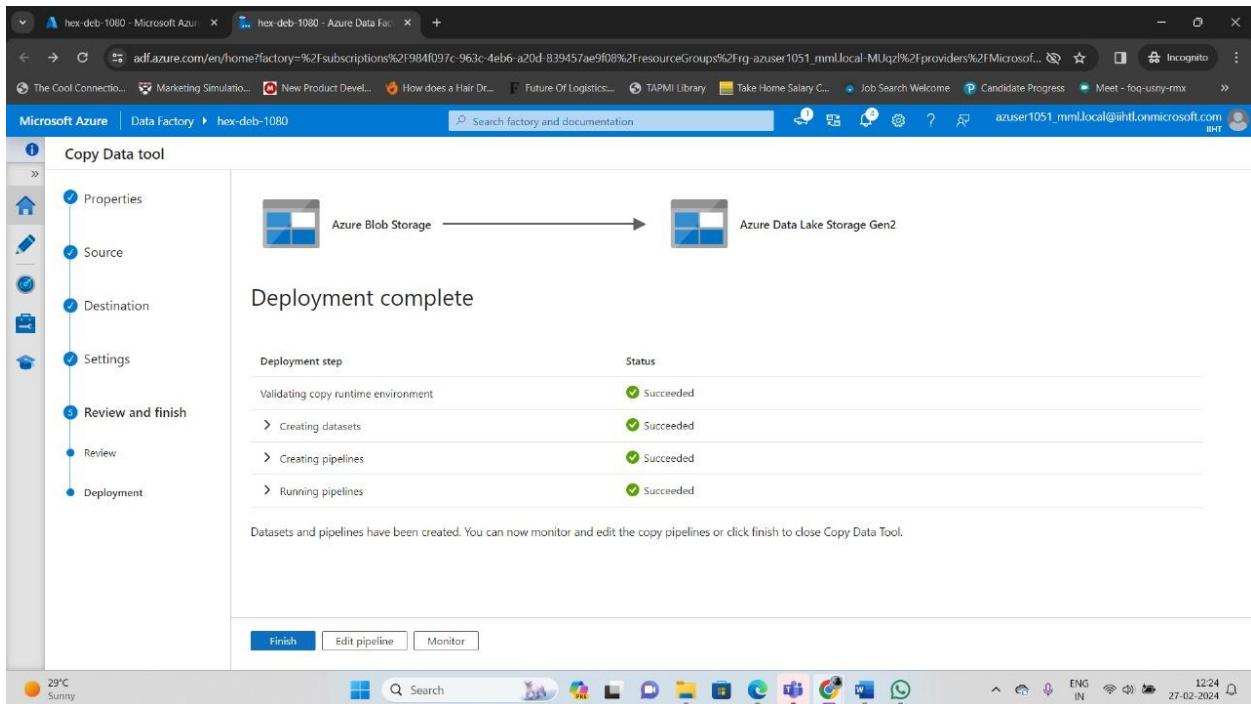


The screenshot shows the Microsoft Azure Data Factory Copy Data tool interface. On the left, a vertical navigation bar lists steps: Properties, Source, Destination, Settings, Review and finish, Review, and Deployment. The 'Properties' step is currently selected. The main panel displays a summary: "You are running pipeline to copy data from Azure Blob Storage to Azure Data Lake Storage Gen2." Below this is a diagram showing an arrow pointing from "Azure Blob Storage" to "Azure Data Lake Storage Gen2". The "Properties" section contains the following details:

Task name	CopyPipeline_project1	<a href="#">Edit</a>
Task description		<a href="#">Edit</a>
Source		
Connection name	AzureBlobStorage2	<a href="#">Edit</a>
Dataset name	SourceDataset_0sz	<a href="#">Edit</a>
Container	container-2	<a href="#">Edit</a>
Destination		
Connection name	AzureDataLakeStorage2	<a href="#">Edit</a>

At the bottom of the main panel are buttons for "Next >" and "Cancel". The status bar at the bottom of the browser window shows the date and time as 27-02-2024.

Activity succeeded and pipeline ingested



The screenshot shows the Microsoft Azure Data Factory Copy Data tool interface after deployment. The vertical navigation bar on the left is identical to the previous screenshot. The main panel now displays the message "Deployment complete". Below this, a table shows the deployment steps and their status:

Deployment step	Status
Validating copy runtime environment	<span style="color: green;">✓ Succeeded</span>
> Creating datasets	<span style="color: green;">✓ Succeeded</span>
> Creating pipelines	<span style="color: green;">✓ Succeeded</span>
> Running pipelines	<span style="color: green;">✓ Succeeded</span>

Below the table, a note states: "Datasets and pipelines have been created. You can now monitor and edit the copy pipelines or click finish to close Copy Data Tool." At the bottom are buttons for "Finish", "Edit pipeline", and "Monitor". The status bar at the bottom of the browser window shows the date and time as 27-02-2024.

## Data copied in container-1

The screenshot shows the Microsoft Azure Storage Container blade for 'container-1'. The left sidebar has sections for Overview, Diagnose and solve problems, Access Control (IAM), Settings (Shared access tokens, Manage ACL, Access policy, Properties, Metadata), and a 'Search' bar. The main area displays blob details with a table:

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
Adaptability (2).pptx	2/27/2024, 12:25:06 ...	Hot (Inferred)		Block blob	12.44 MiB	Available

At the bottom, there's a taskbar with icons for various Windows applications like File Explorer, Edge, and Mail.

## Reference links:

<https://learn.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-introduction>

<https://learn.microsoft.com/en-us/azure/data-factory/load-azure-data-lake-storage-gen2>

