**2.Unity Catalog 3 – level namespace:**

Unity Catalog provides centralized access control, auditing, lineage, and data discovery capabilities across Azure Databricks workspaces.

Metastore is the top level container of the objects in the azure storage account.

**Catalog:**

Catalog is the first layer in the 3-level namespace, it provides access privilege to the user.

**Schema:**

Schema is the second layer of the 3-level namespace, we generally use the default schema. Default schema will be present in the main catalog.

**Tables:**

Tables are the third layer of the 3-level namespace, it contains rows of data. Tables can be managed or external.

**Managed tables:**

Default tables that are created in unity catalog.. They are stored in root storage location that is connected to the metastore. No open source tools should be used to change the schema or data in the table.

They always use delta table format.

**External Tables:**

External Tables are tables in which whole data lifecycles and file layout are not managed by unity catalog and will be stored in the tabular data. Externally managed data file format could be DELTA, CSV, JSON, AVRO, PARAQUET, ORC, TEXT.

**Steps for creating unity catalog objects are:**

Step1 : Attach your workspace to a Unity Catalog metastore

Step 2 : Add users and assign the workspace admin role

Step 3 : Create clusters or SQL warehouses that users can use to run queries and create objects

Step 4 : Grant privileges to users

Step 5 : Create new catalogs and schemas

**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.

It can be used for:

1. Supporting data migrations.
2. Getting data from a client server or online data to azure data lake.
3. Carrying out various data integration processes.
4. Integrating data from different ERP systems and loading it onto azure synapse for reporting.

Azure data factory works in 3 stages:

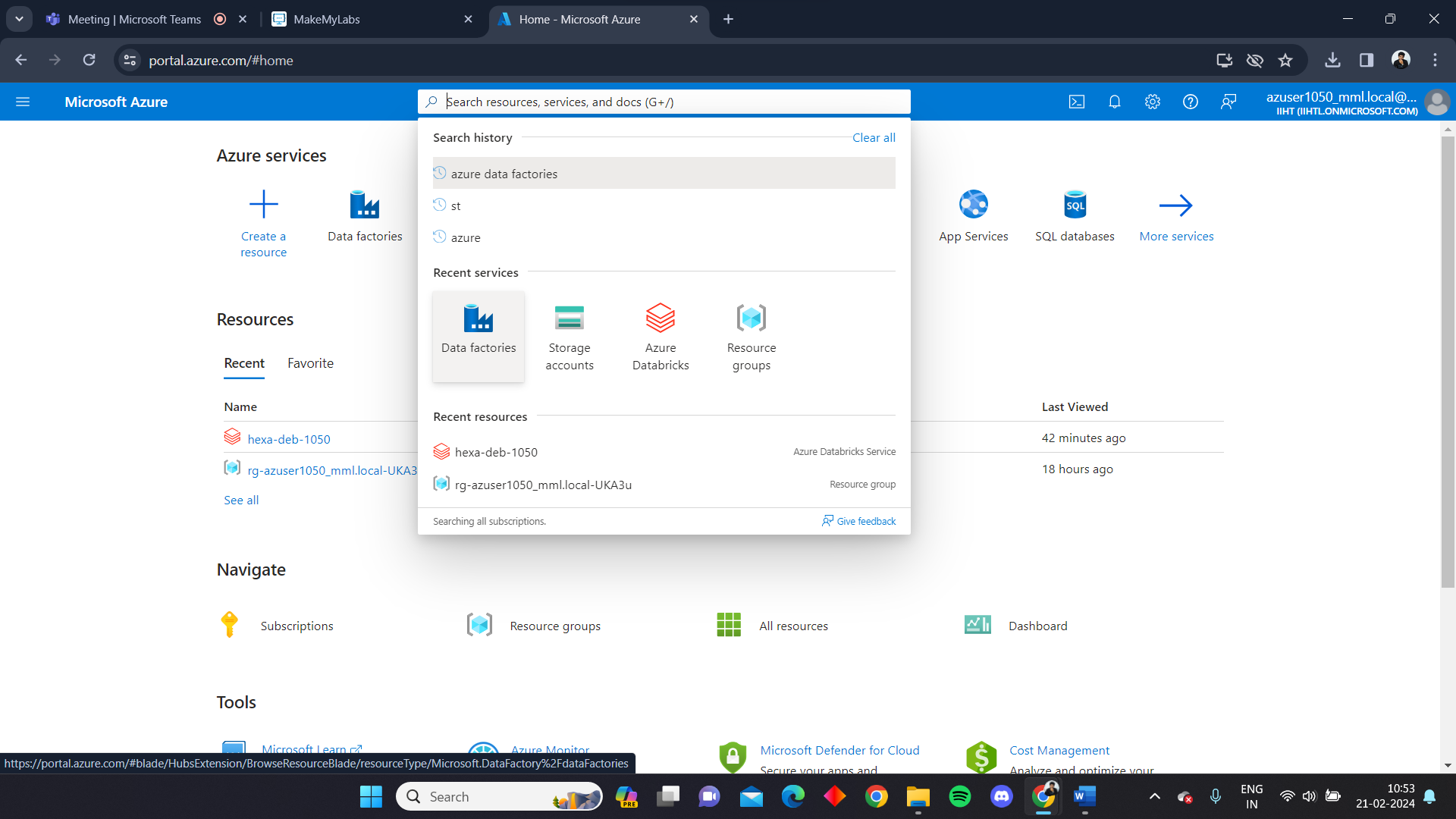
1. Connect and collect : Connect to the server and collect the data
2. Transform and Enrich : Transform the data based on the business requirement.
3. Publish : Publish the data on to the server based on the client requirement.

**ADF CopyActivity :**

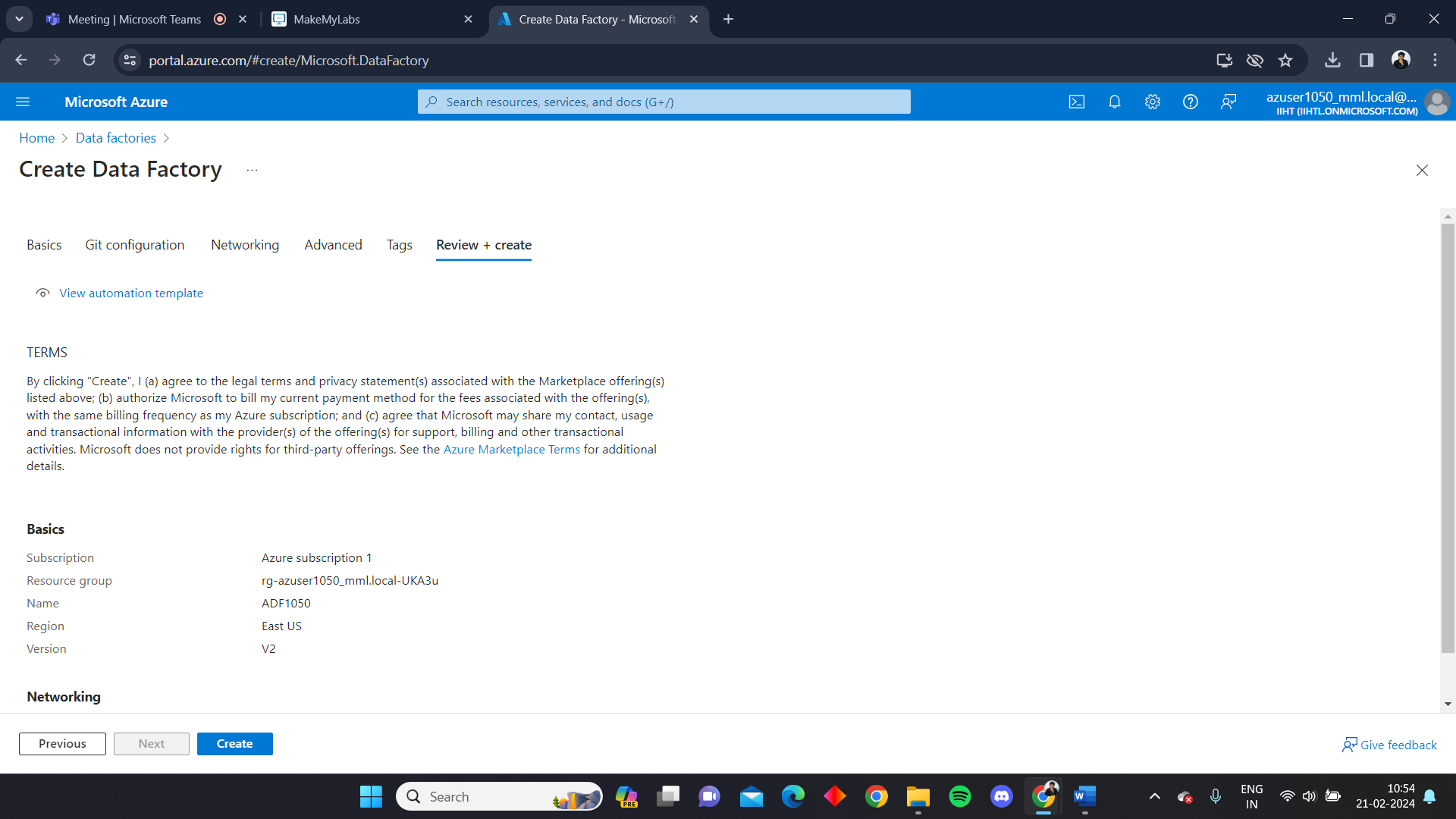
Through the ADF we an perform copy activity which is copying the files or data from source container to the sink container using pipelines.

This is explained through steps below :

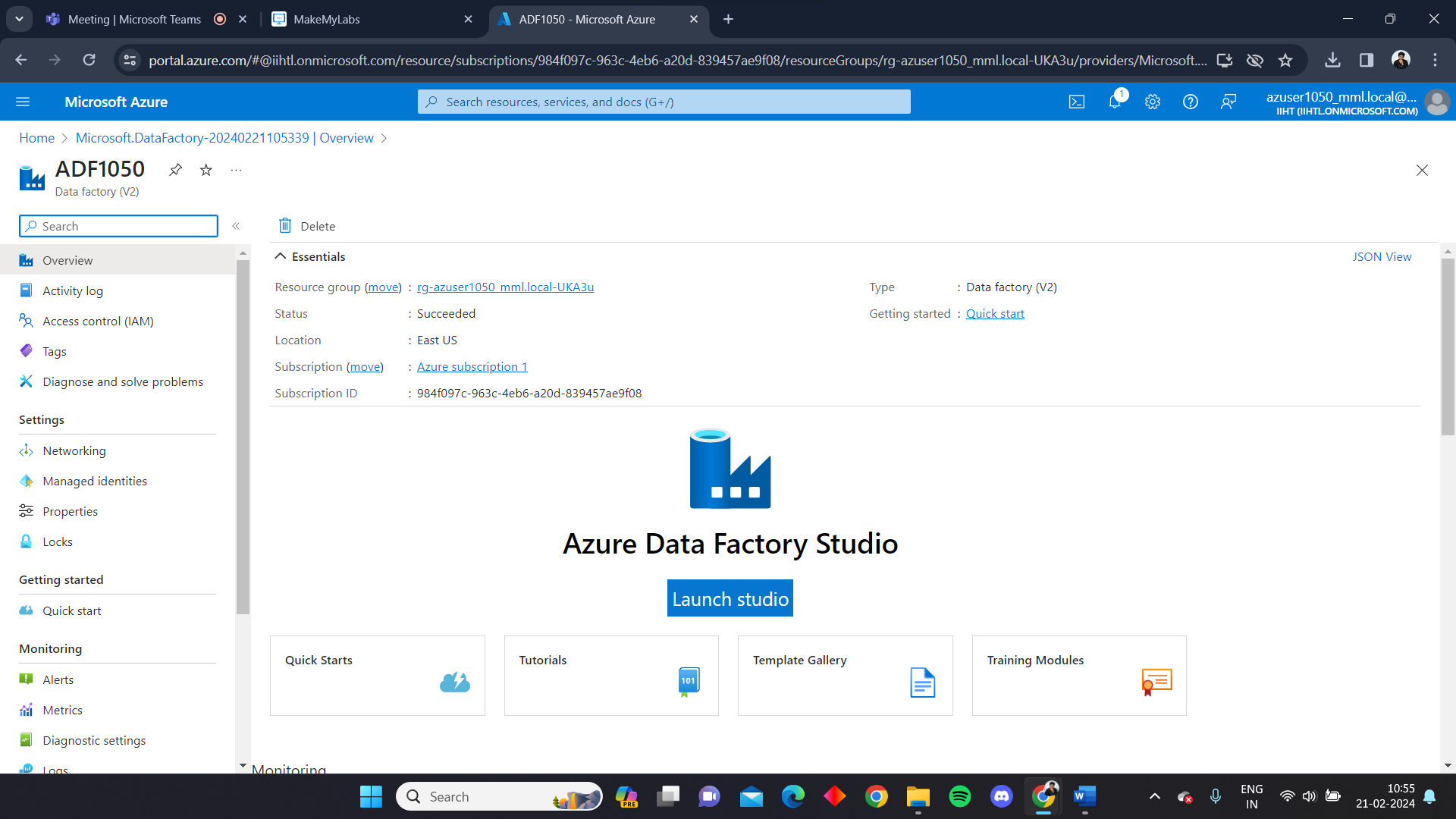
First create a data factory



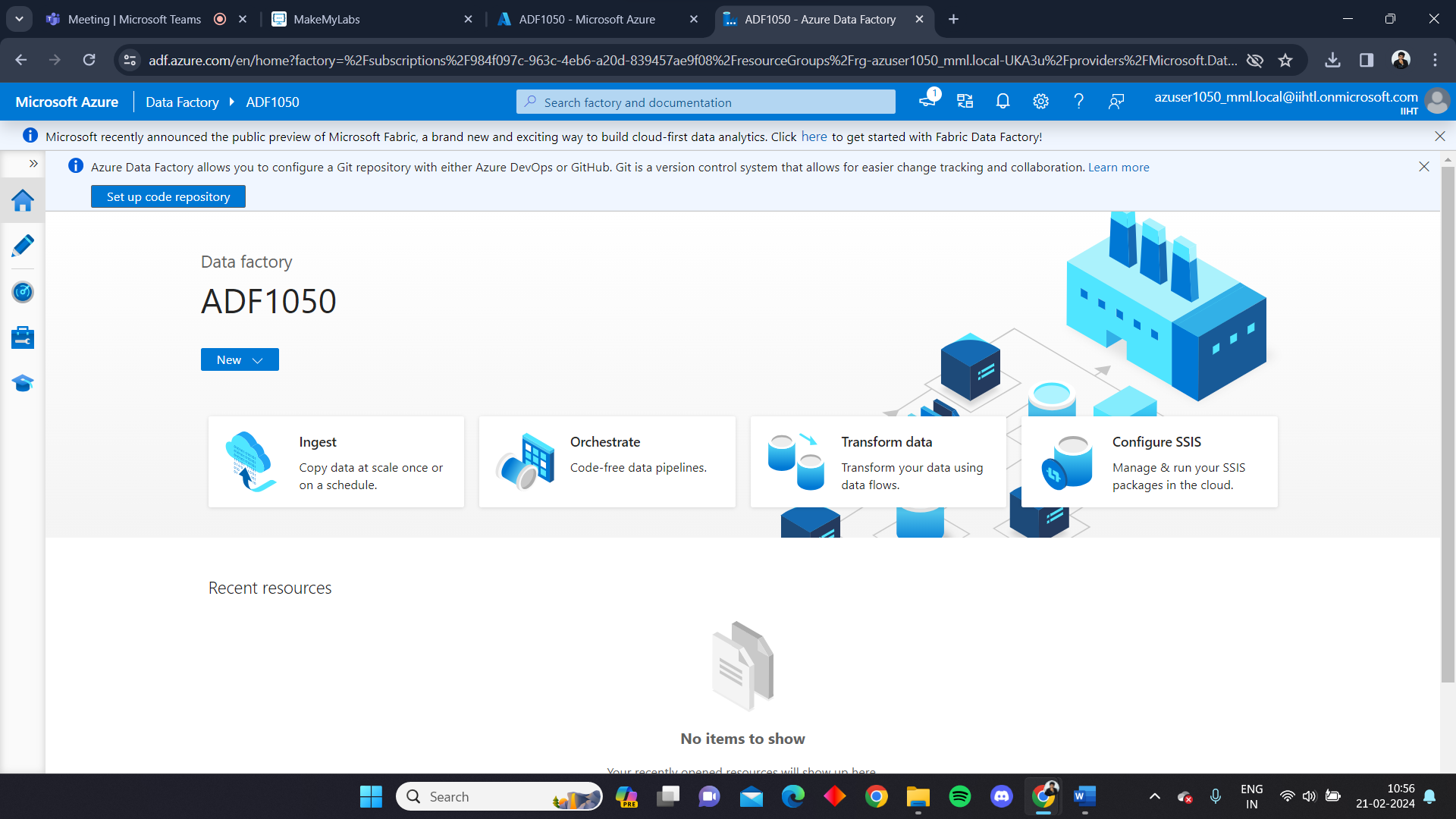
Create the data factory with following features



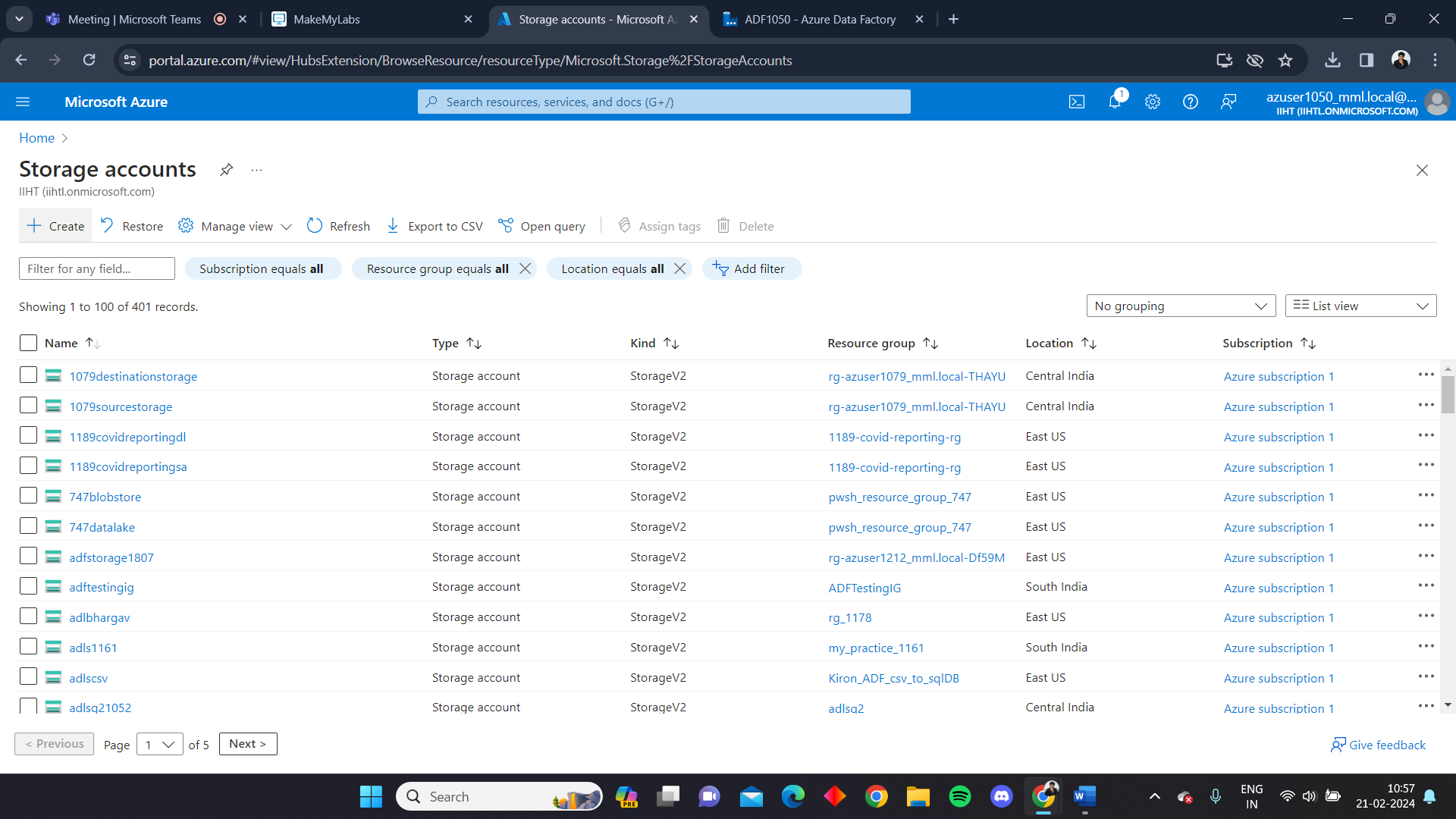
Launch the ADF studio

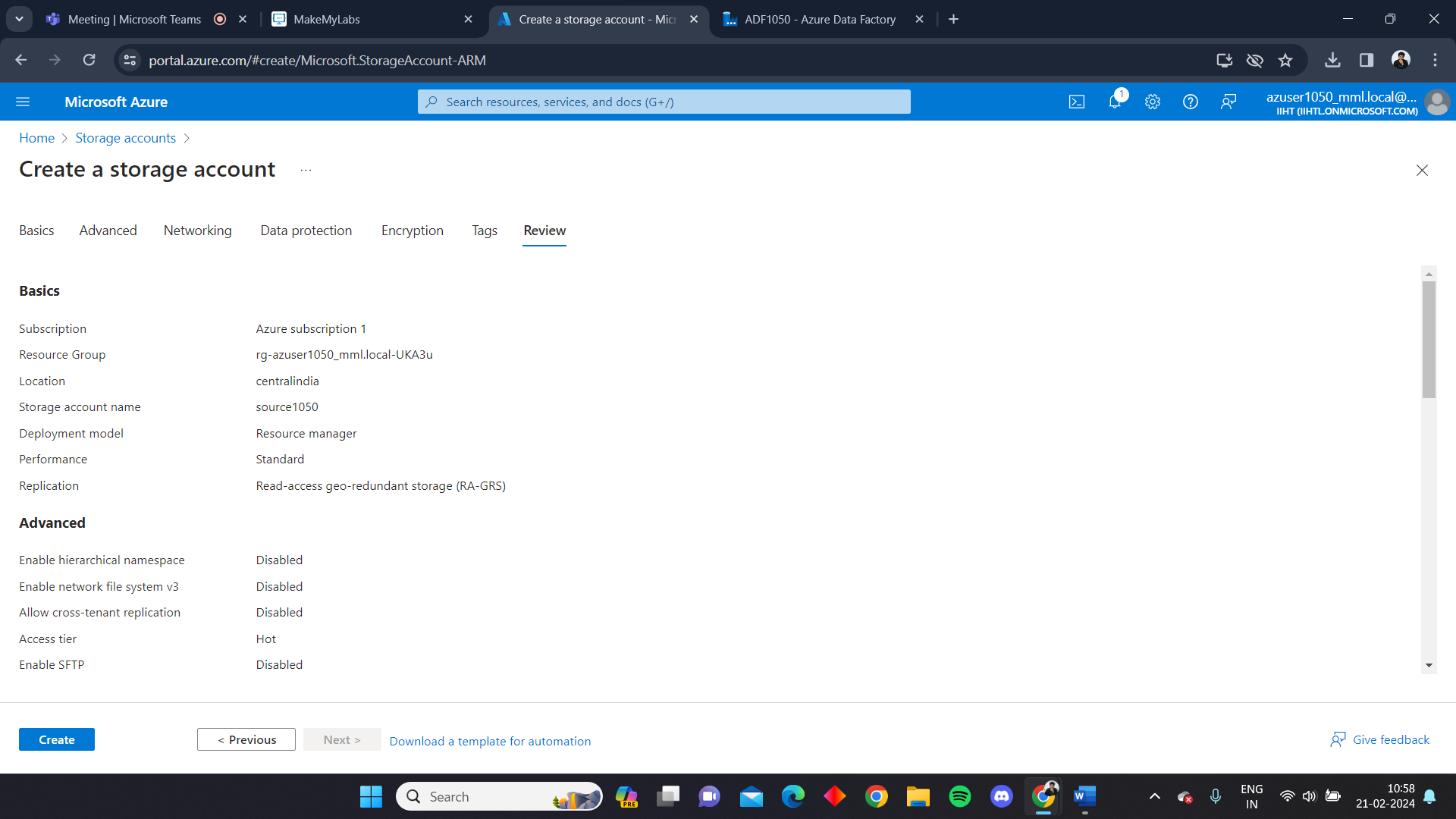


ADF studio looks like this

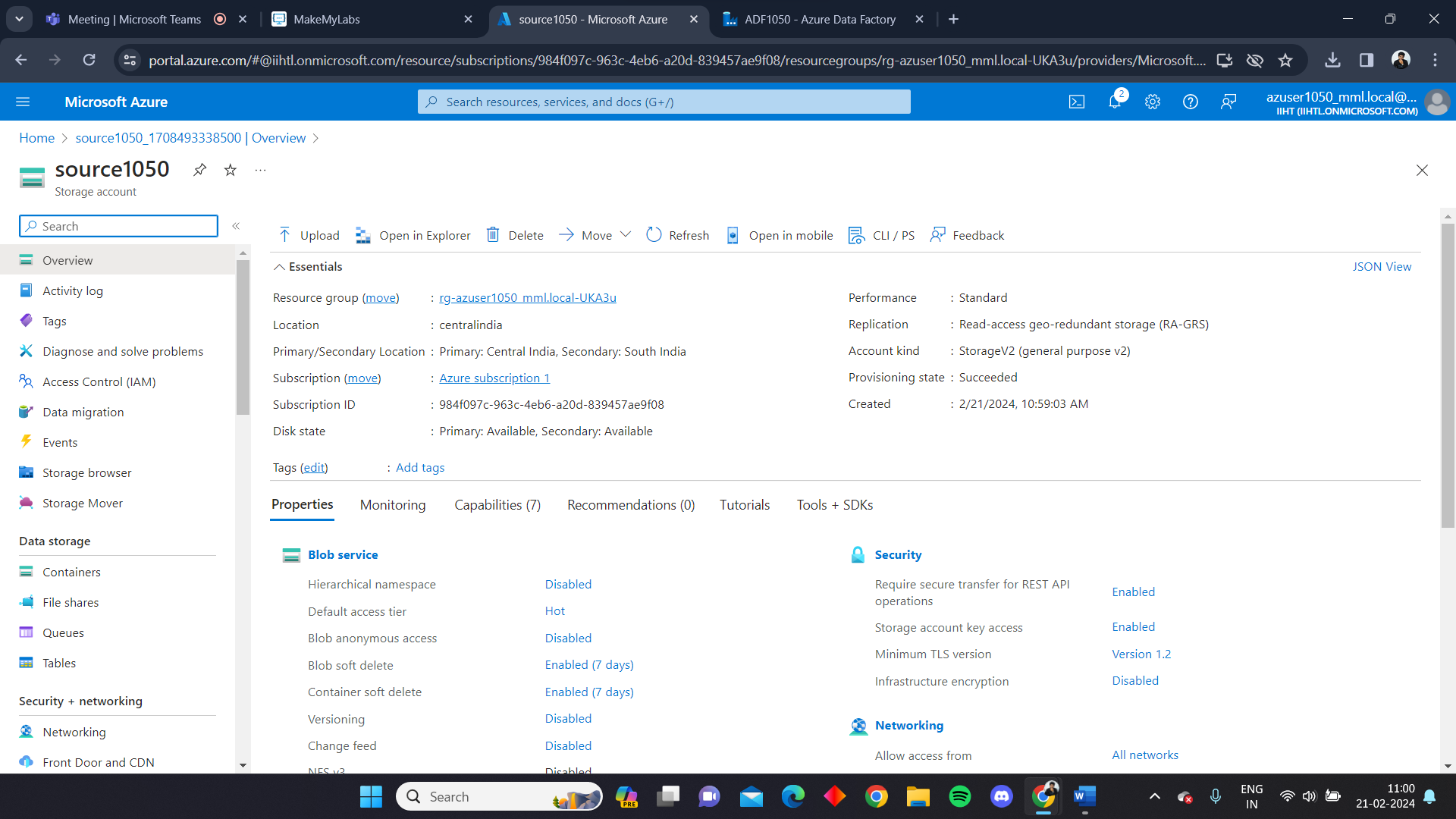


Now create a source storage account

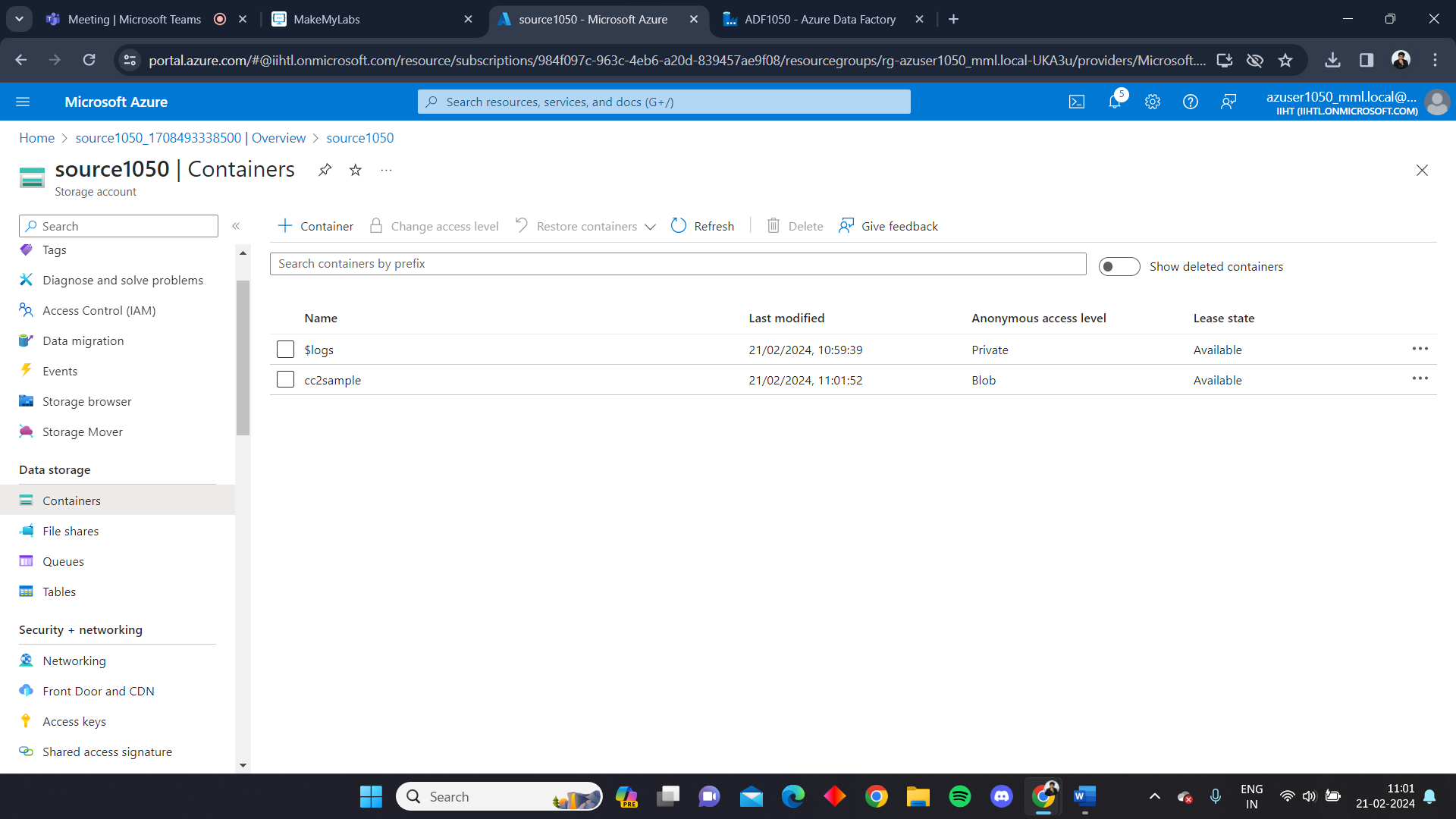




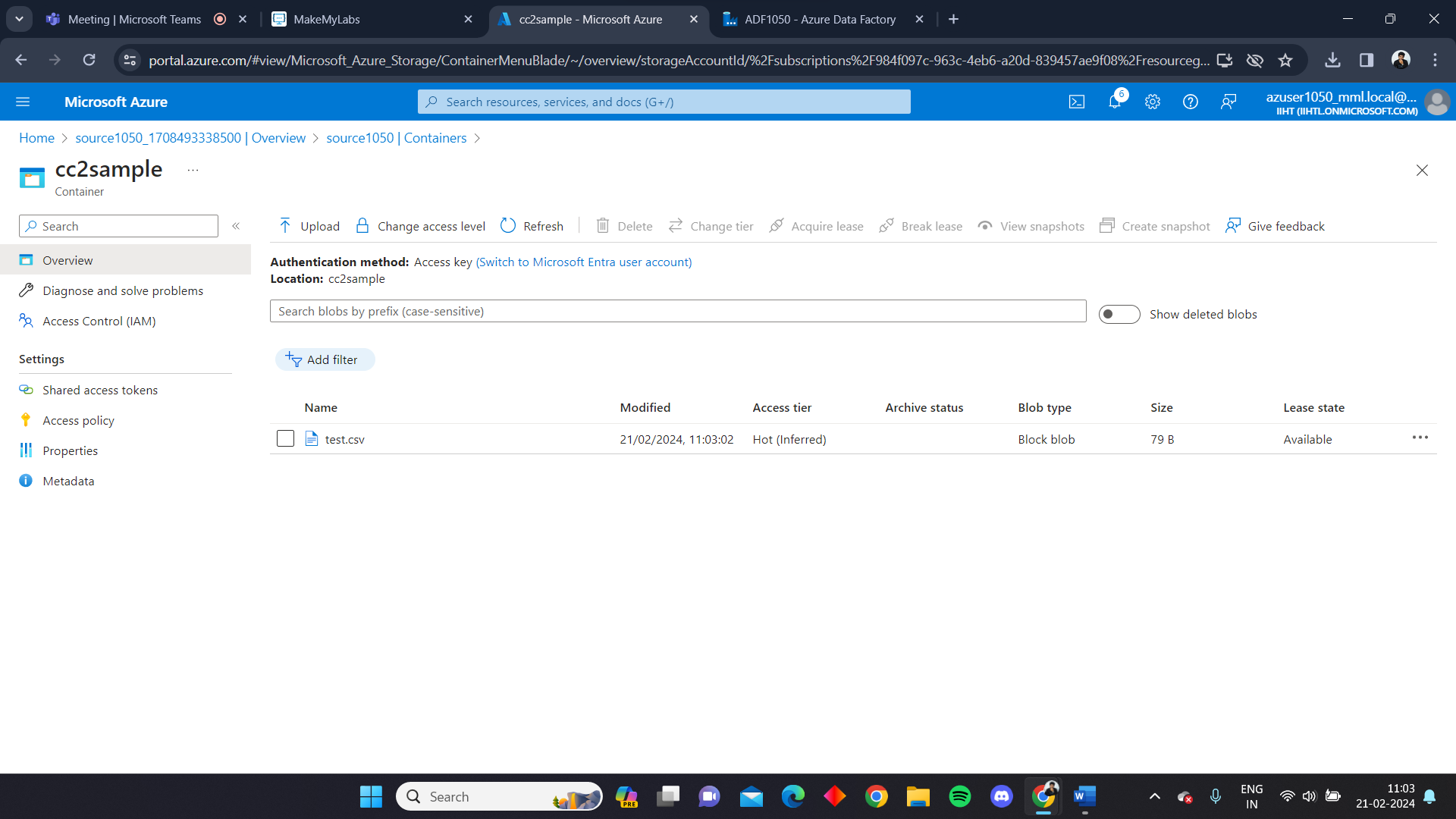
The source1050 is the source account



Create a container in the source account



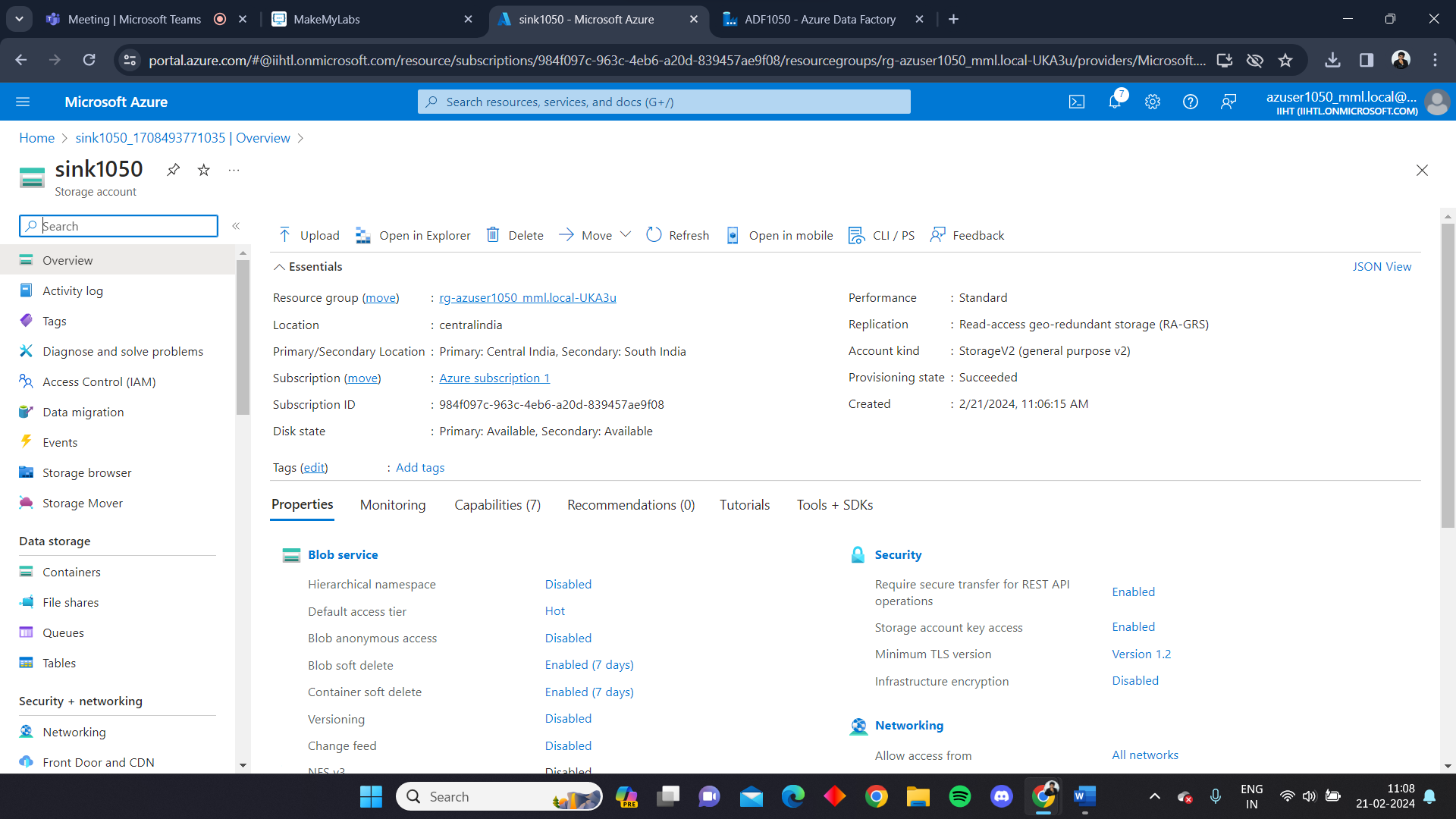
Add some data in the contaiuner



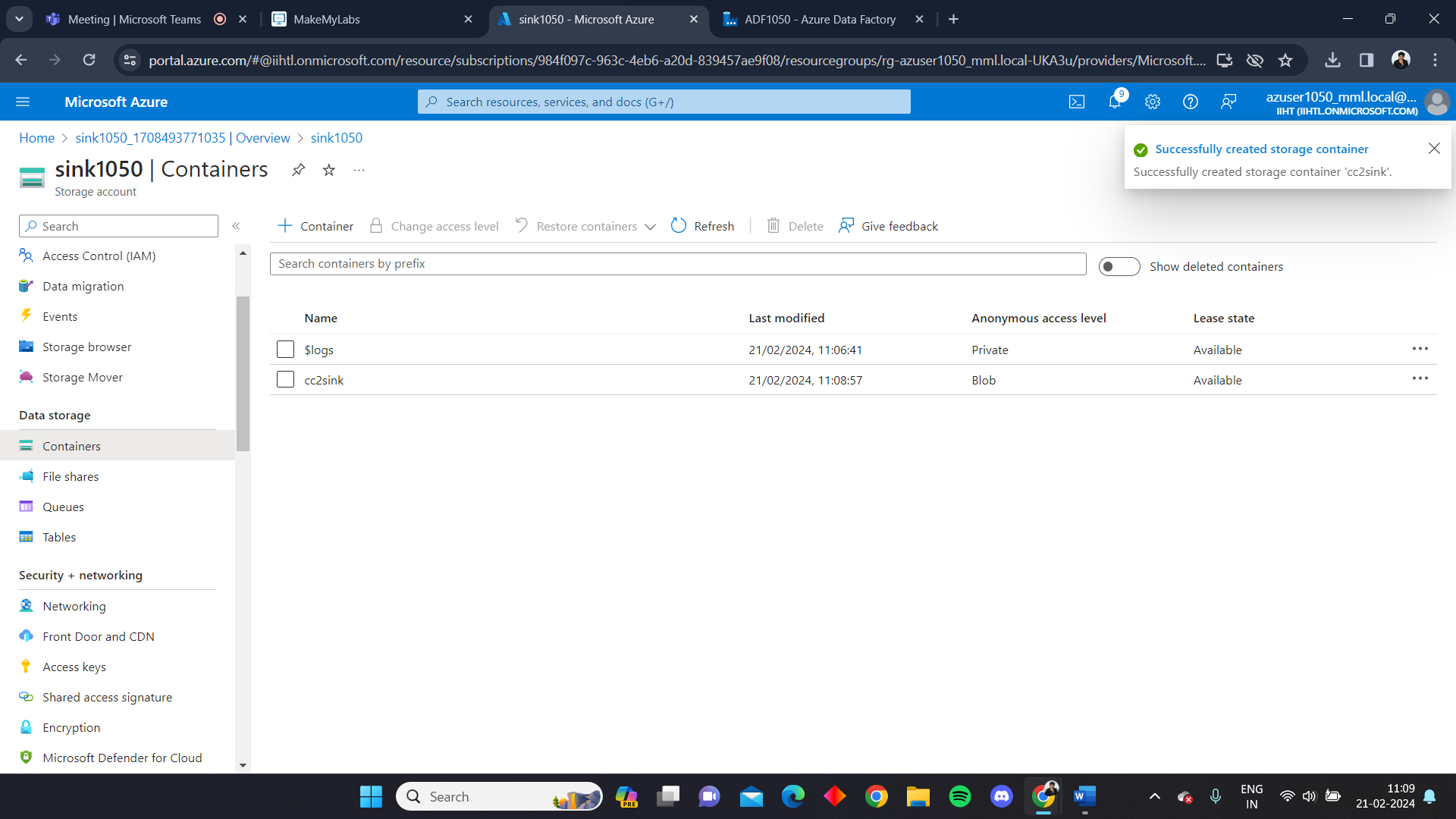
Now create a sink storage account



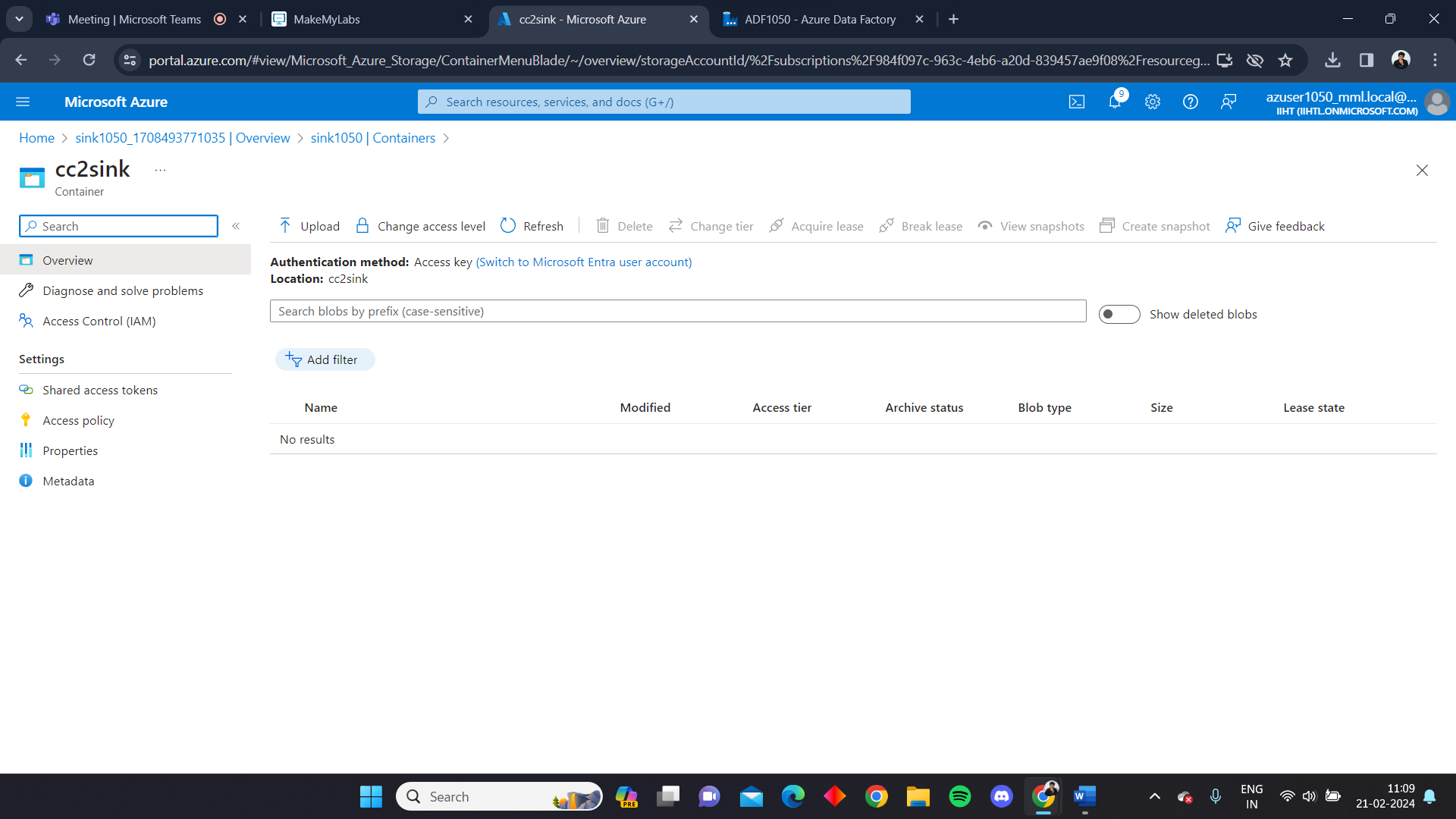
Sink1050 is the sink account



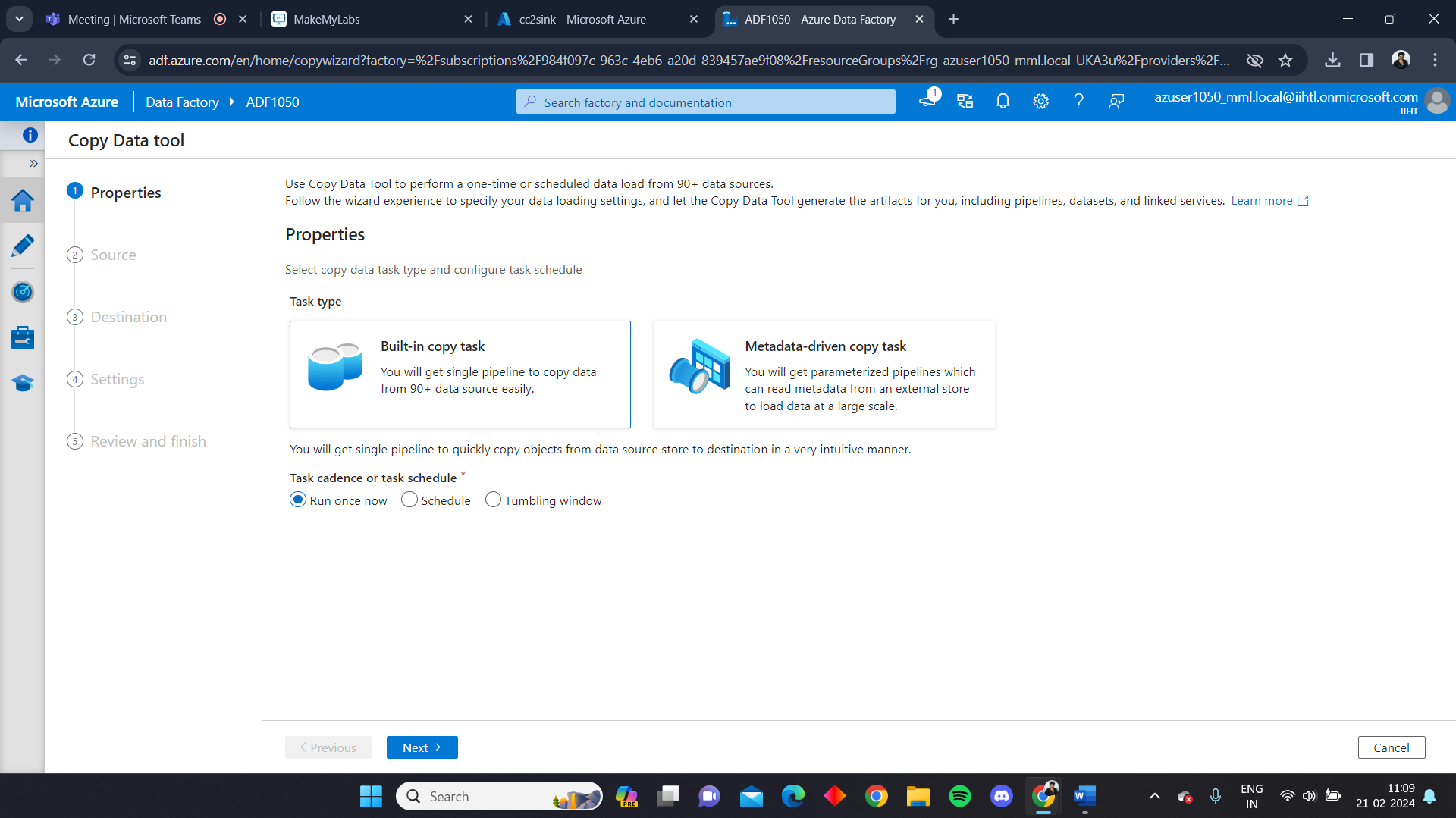
Create a container in the sink account



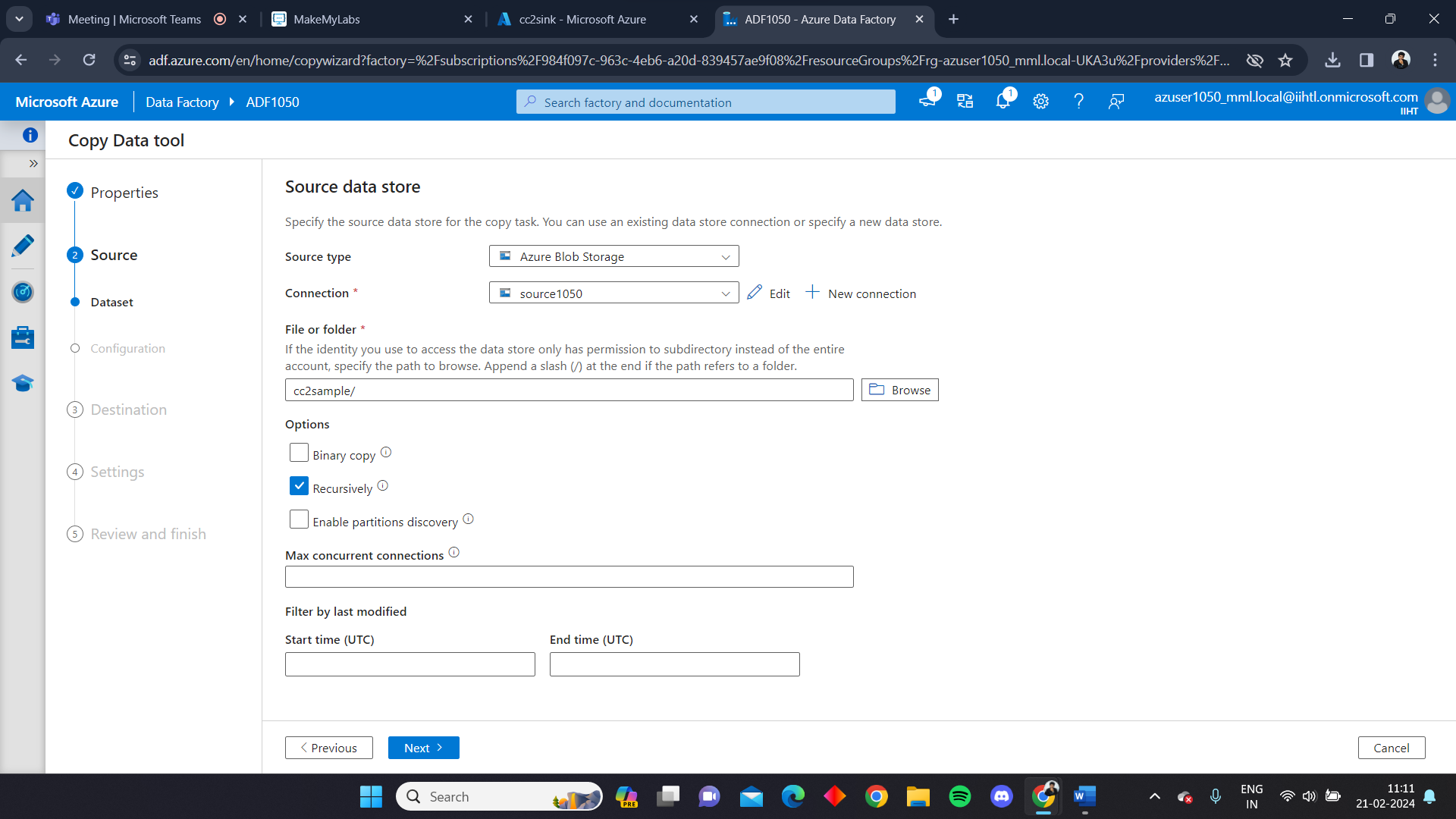
We can see this container is empty



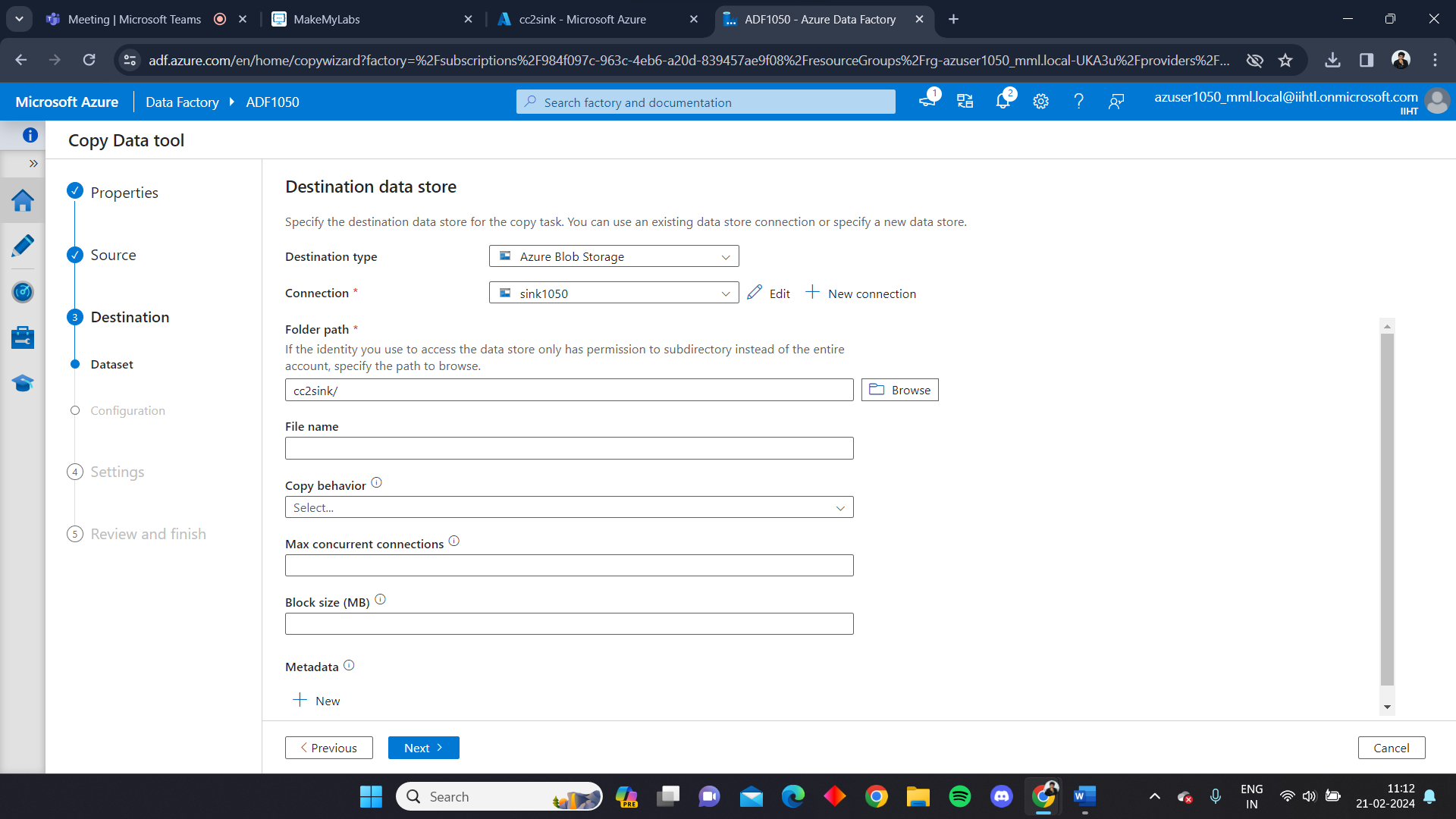
After launching the ADF select int ingest process



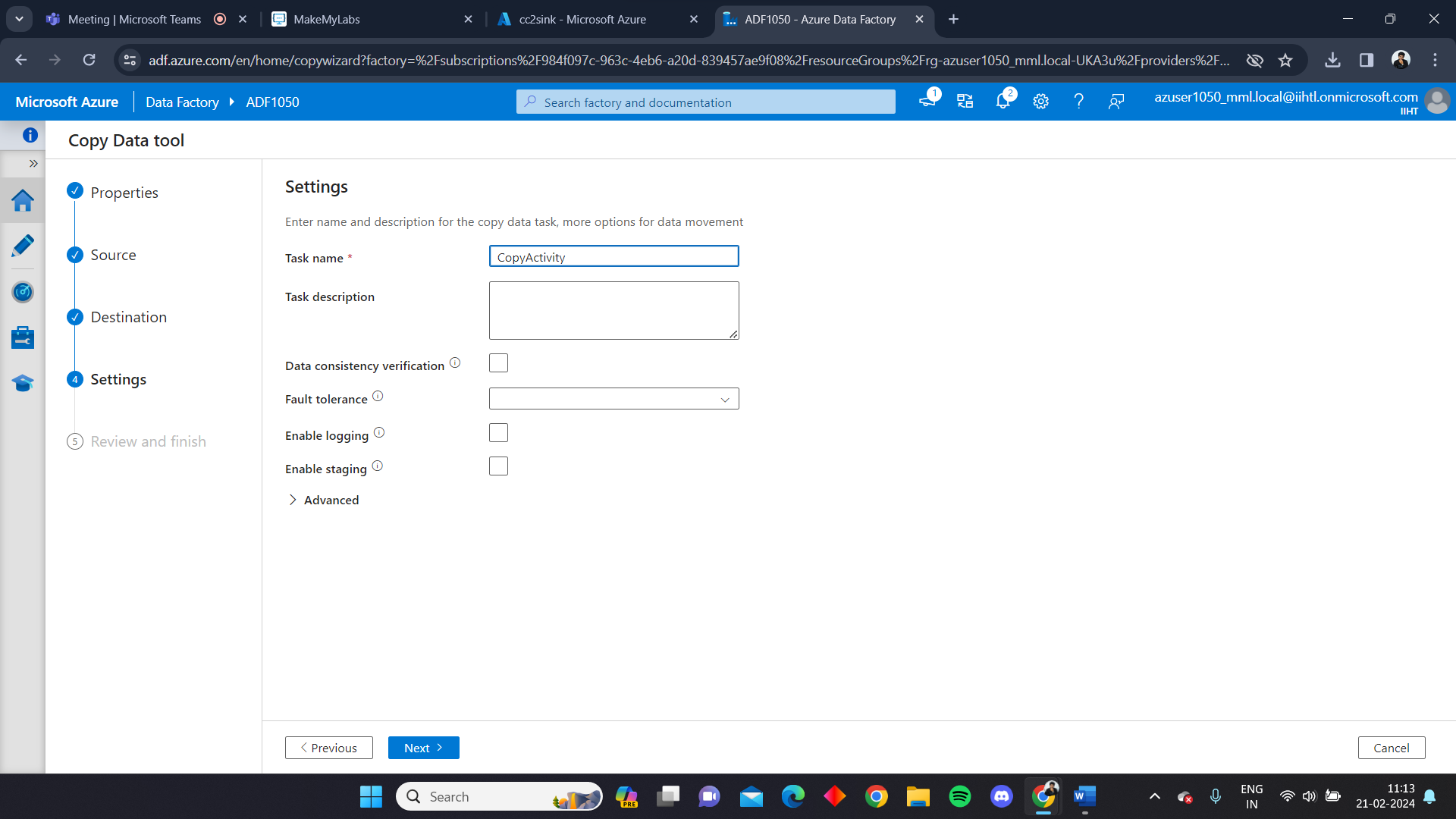
Create a connection in the ADF using the source account



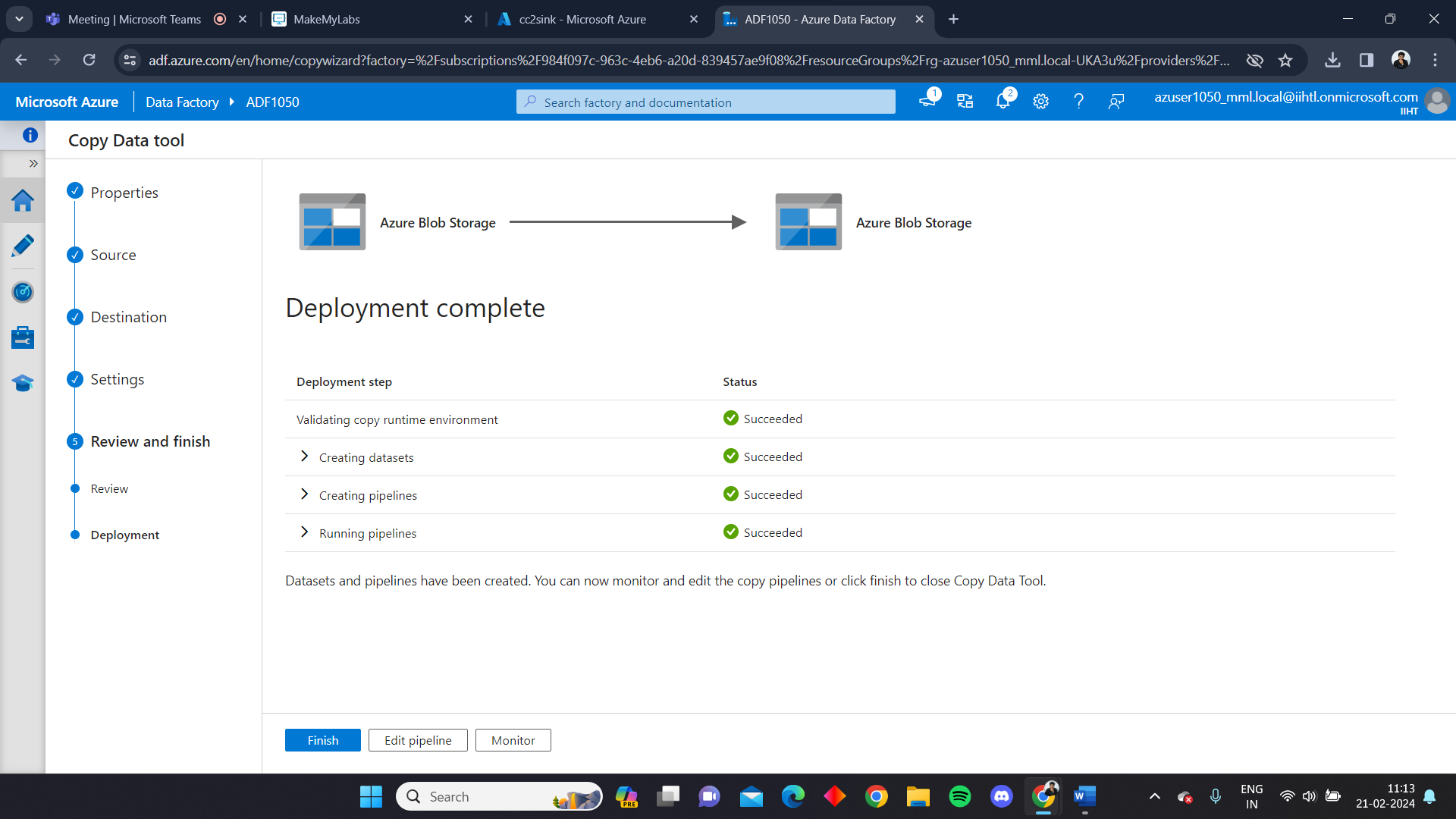
Likewise, create connection with the sink account



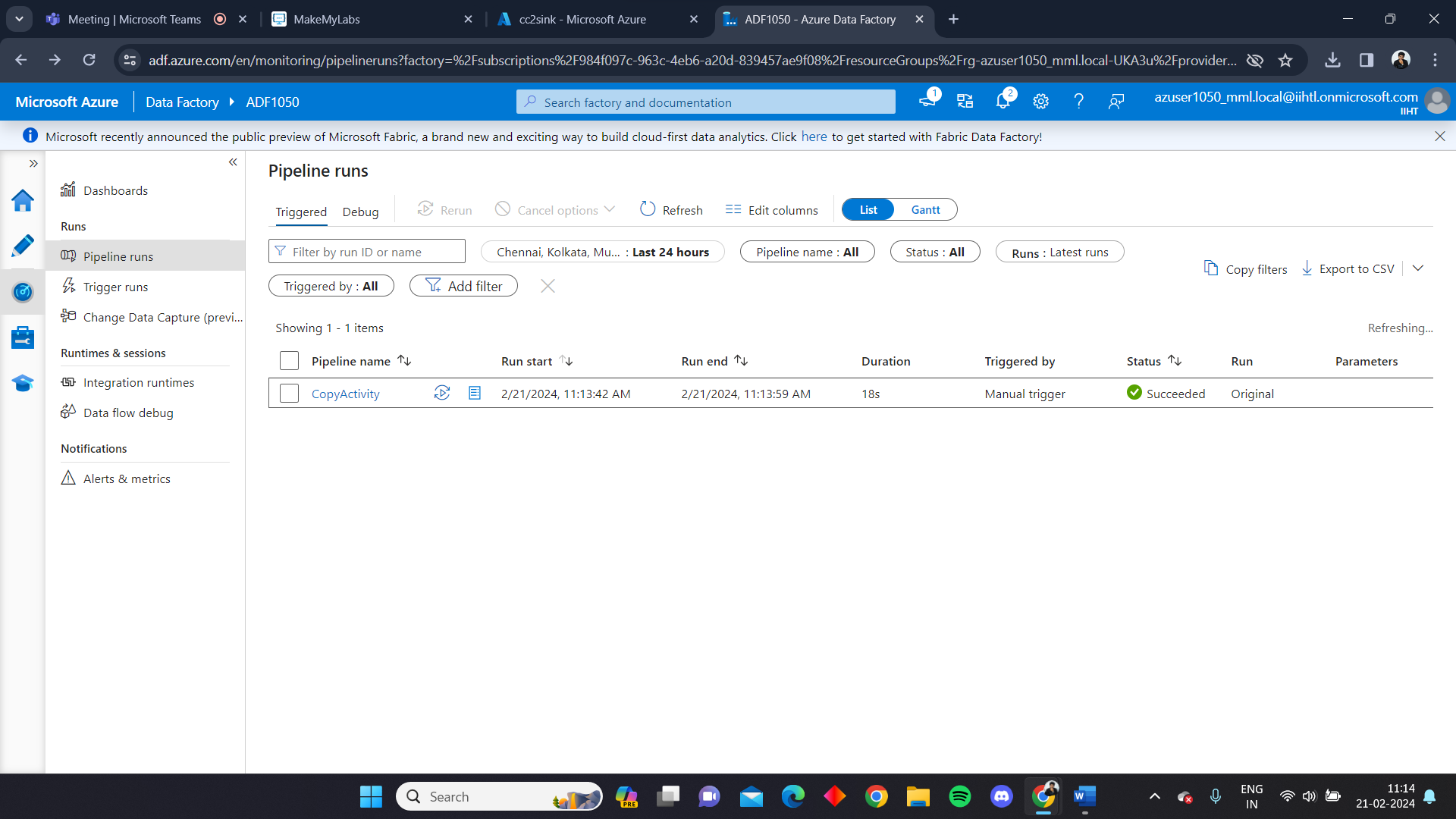
Now give the task name as CopyActivity to perform the copy activity



We can see the deployment will be completed



We can monitor the pipeline through ADF



We can see that the activity is complete



Here in the sink storage account we can see that the file present in the source storage account is copied successfully through the pipeline created using ADF. Hence copy activity is performed.

