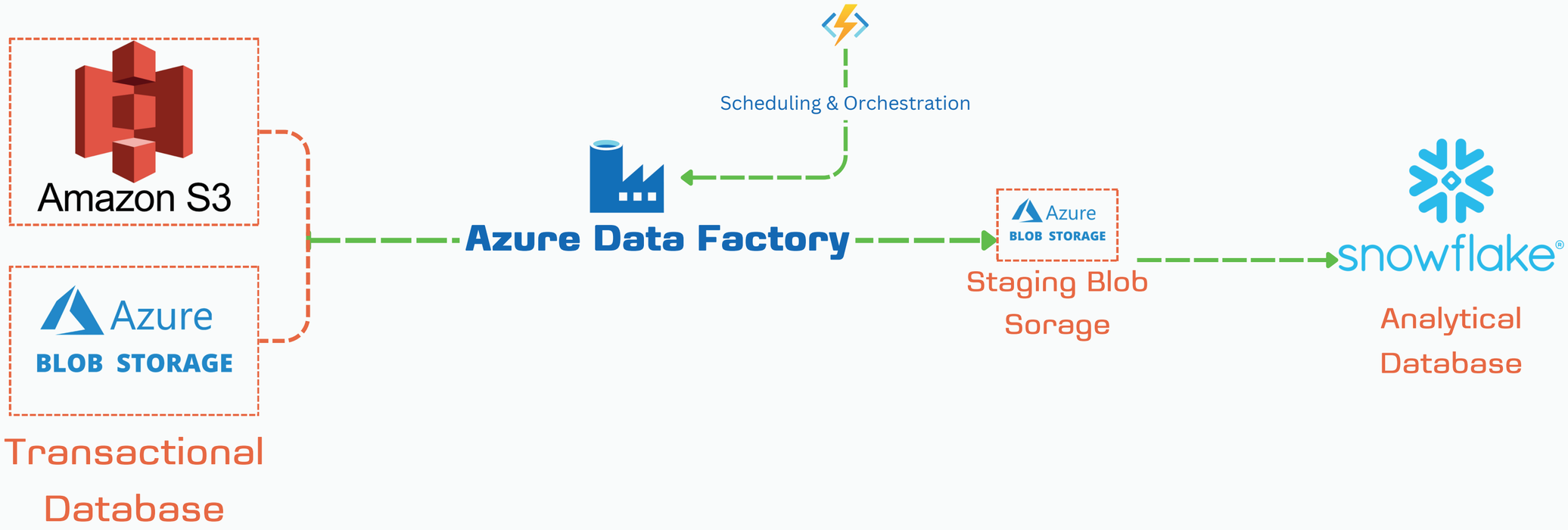
Multi-Cloud Data Integration



/prayagv
/prayag-verma

Prayag Verma
Data Engineer

Probelm Statement

Companies use different cloud platforms to store data, which makes it hard to bring everything together, therefore,

In this project, we'll take "customers" and "orders" data from 'AWS S3' and 'Azure Blob Storage' and move it into Snowflake Warehouse using Azure Data Factory (ADF) and Trigger. We will integrate data from both clouds into one place for the business analysis.

Step 1

Prerequisites

We need the following to proceed with the plan:

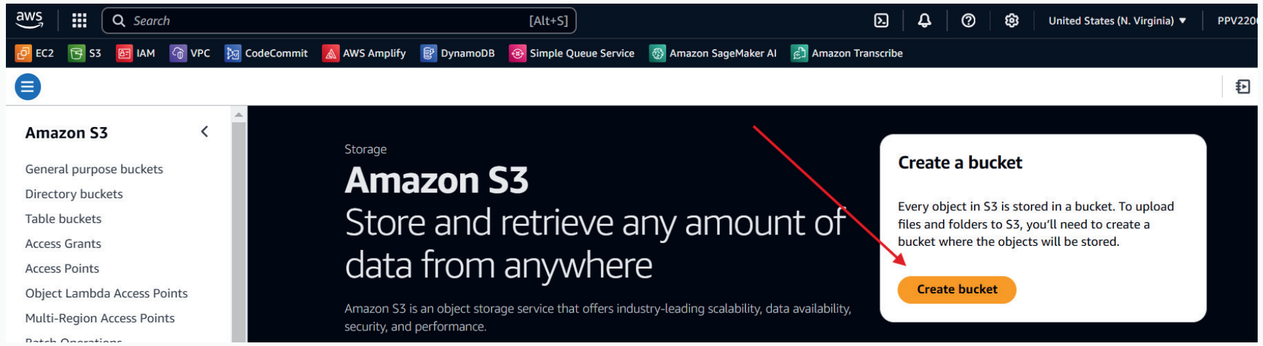
1. **AWS S3 Bucket (For customer data - customers.csv)**
   Bucket Name: ‘customers-data-s3-bucket’
   File name: ‘customers.csv’
1. Azure Blob Storage (For order data - orders.csv)
   Containers Name: ‘orders’
   File Name: ‘orders.csv’
1. Snowflake Account
   Create tables: ‘customers’ and ‘orders’
1. Azure Data Factory (ADF)
2. ADF Trigger (For scheduling and orchestration)

Step 2 Create S3 Bucket and Upload File

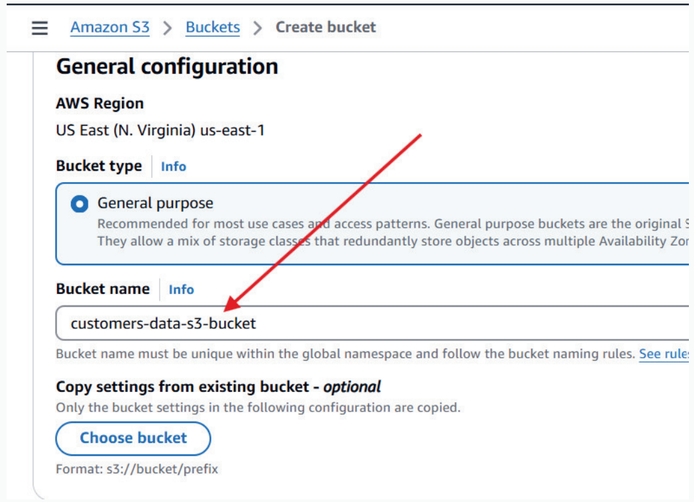
Step 2a: Create an S3 bucket and upload the customers.csv file.

1. **Create S3 Bucket:** Go to **AWS dashboard**, **search for 'S3'** and select it> Click on '**Create bucket**', enter a unique name (e.g., customers-data-s3-bucket), **enable** 'Block all public access for this bucket' option, and click on '**create bucket**' button.
2. **Upload File:** Click on the bucket just created, then click '**Upload**', then click on '**add files**' and select '**customers.csv** 'file, and finely click on '**upload**'. **Note:** Ensure the file is uploded in the bucket.

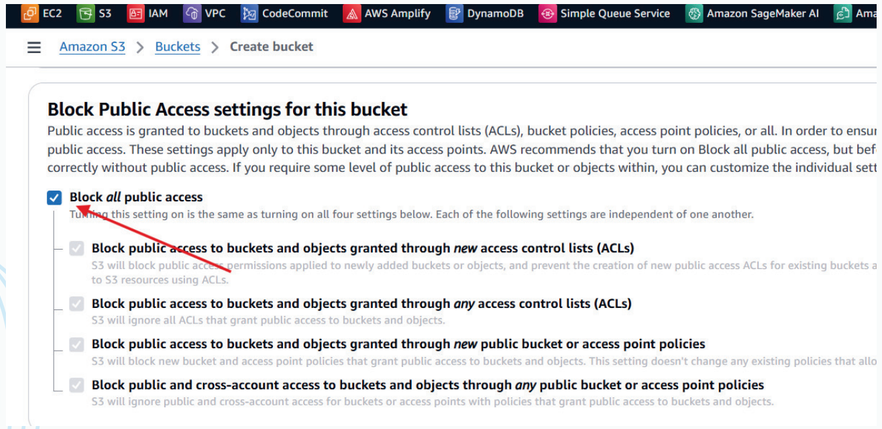
1



2



3



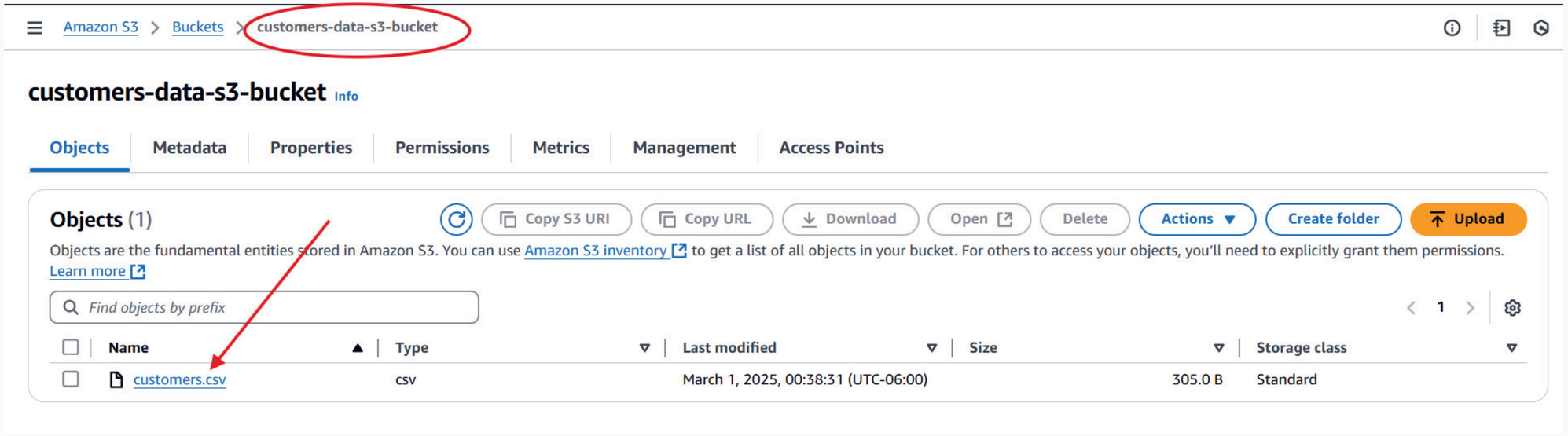
4

Step 2

Create S3 Bucket and Upload File

Step 2b: Upload ‘customers.csv’ file to S3 bucket

1. **Upload File:** Click on the bucket just created, then click ‘**Upload**’, then click on ‘**add files**’ and select ‘**customers.csv** ‘file, and finely click on ’**upload**’. **Note:** Ensure the file is uploded in the bucket.

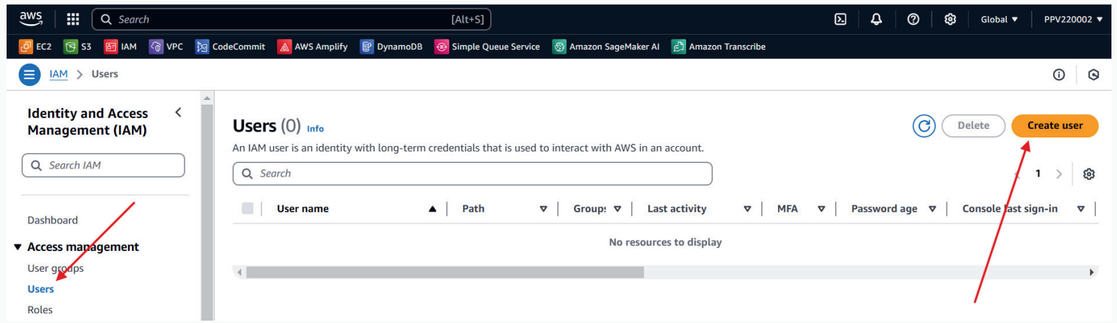


Step 2 Create S3 Bucket and Upload File

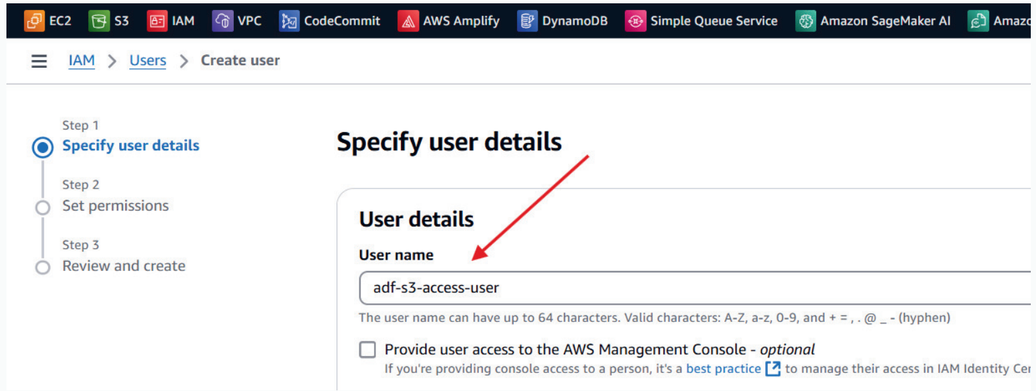
Step 2c: To access this bucket from ADF(3rd Party), Create an IAM user credentials and assign a permission to it

1. **Create IAM User:** Go to **AWS dashboard**, search for '**IAM**' and click on in> '**Users**' > then '**Creat user**', enter a name (e.g., adf-s3-access-user), go to '**next**' step, Select '**Attach policies directly**' then search and select '**AmazonS3ReadOnlyAccess policy**', and create the user.

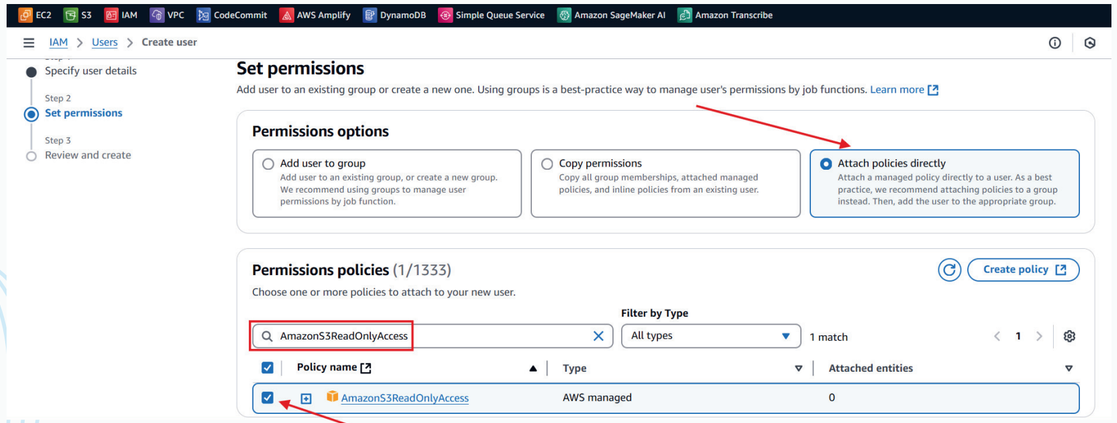
1



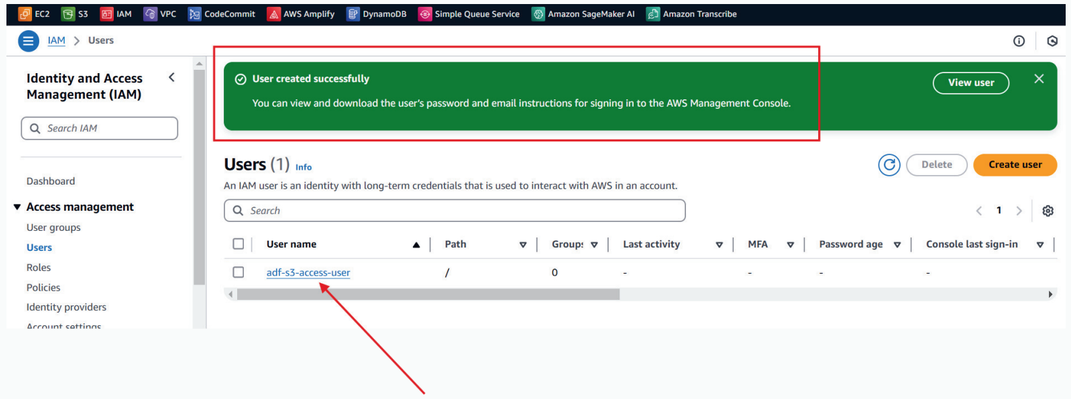
2



3



4

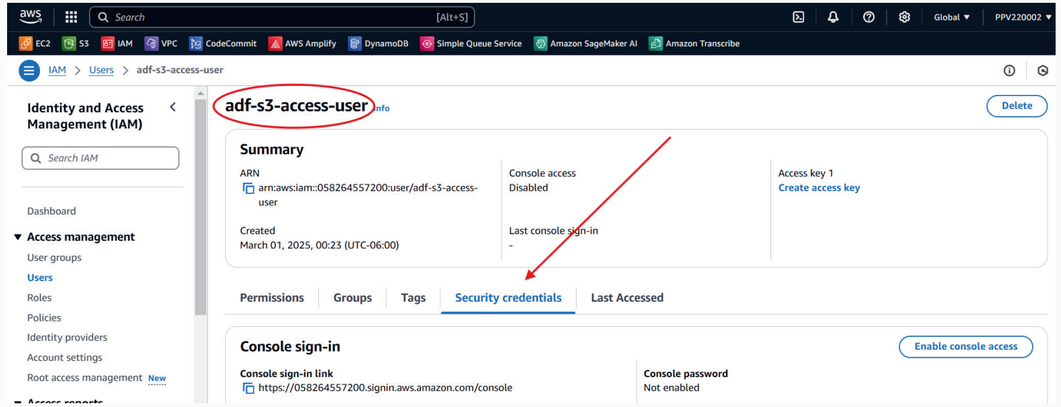


Step 2 Create S3 Bucket and Upload File

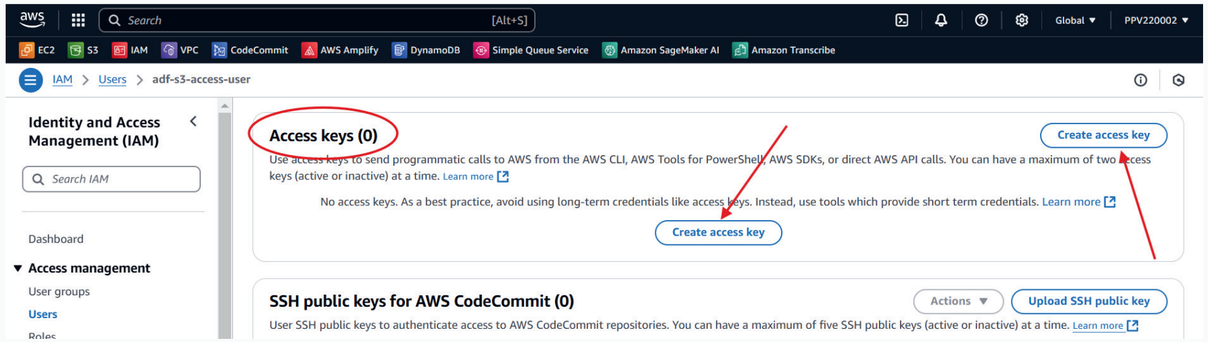
Step 2d: Next, Generate Access Keys (Programmatic Access)

1. **Generate Access Keys:** Go to '**IAM**' dashboard, and select '**adf-s3-access-user**' the one you just created, **Next**, Go to the '**Security credentials**' **tab** and scroll to '**Access Keys**' section, then click '**Create access key**' and select '**Third-party service**', and save the '**Access Key**' and '**Secret Access Key**' or **download** the **csv** file for your better.

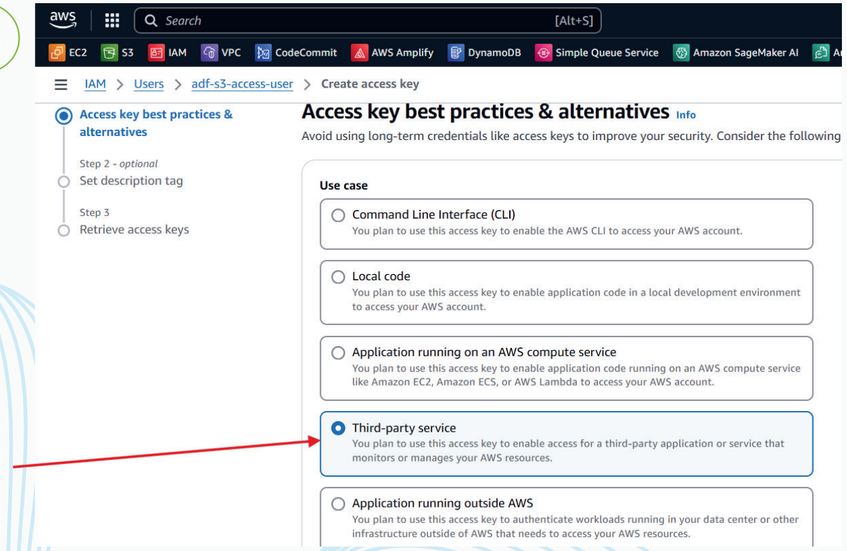
1



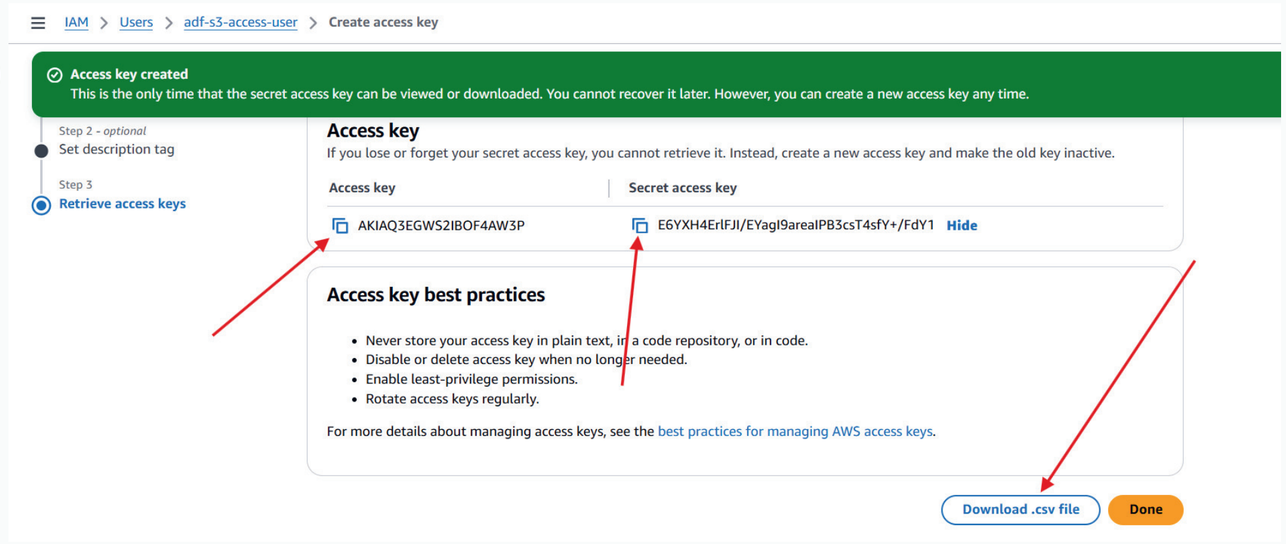
2



3



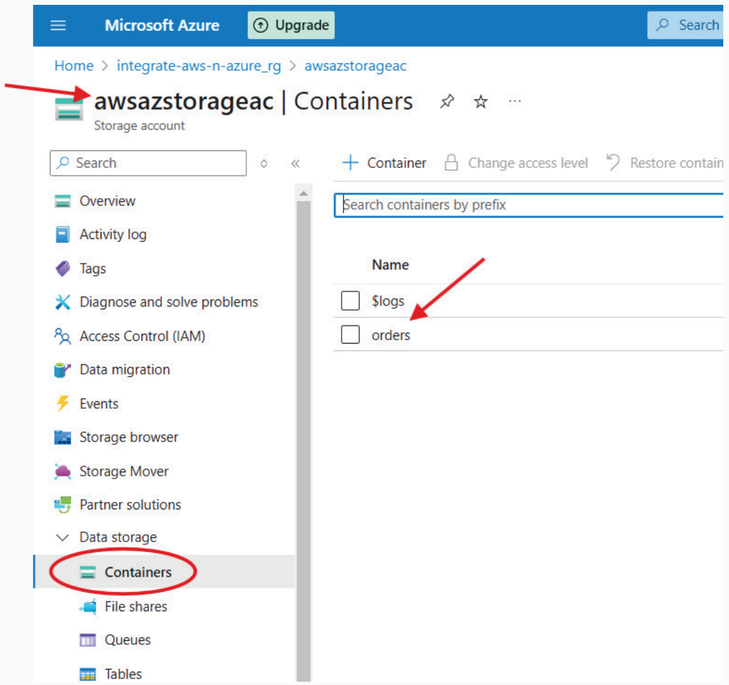
4



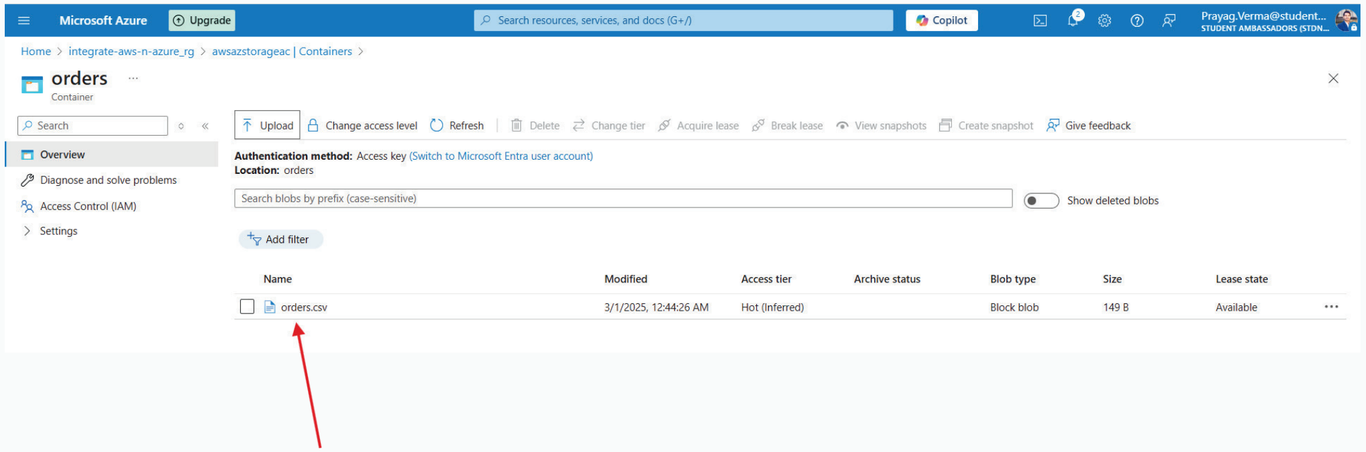
Step 3 Create Azure Container & Upload File

Note: Create a storage account with name '**awsazstorageac**', and also create a container with name '**orders**' and then upload the '**orders.csv**' file to the same containers!

1



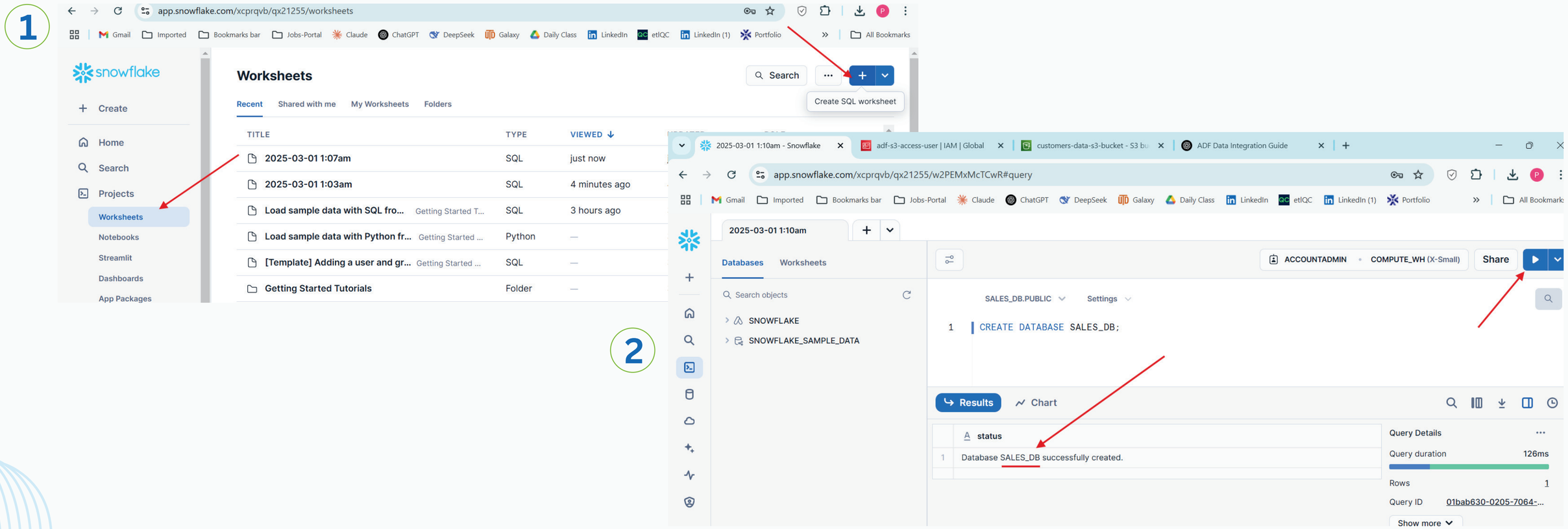
2



Step 4 — Set Up Snowflake (Analytical DB)

Step 4a: Create a 'Database'

Log in to snowflake - from homepage, click on '**Project**' then '**Worksheets**' Next, create a db using '**CREATE DATABASE SALES_DB;**' and execute by clicking '**Play**' icon visible at the **top right corner** position.



NOTE: You can also verify by executing 'SHOW DATABASES;'

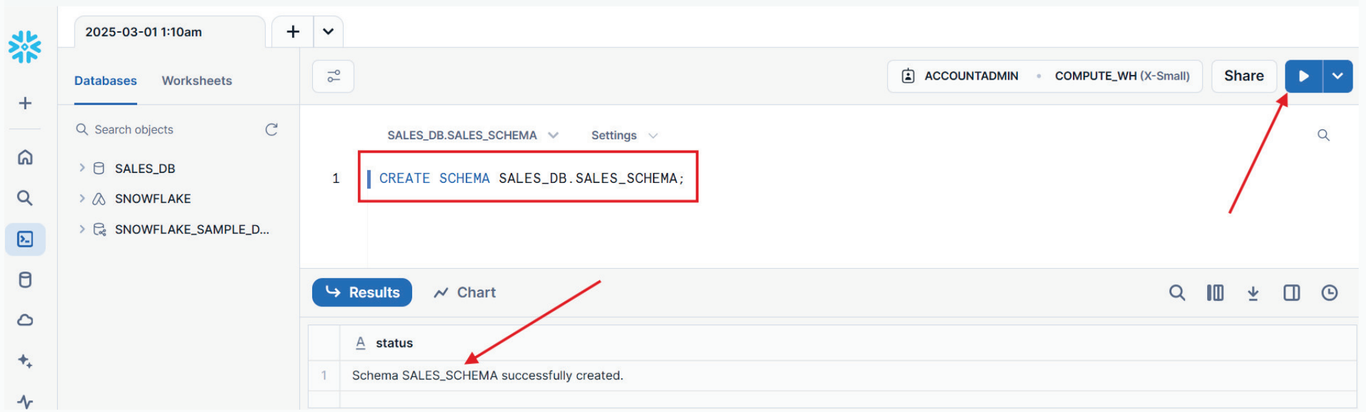
Step 4

Set Up Snowflake (Analytical DB)

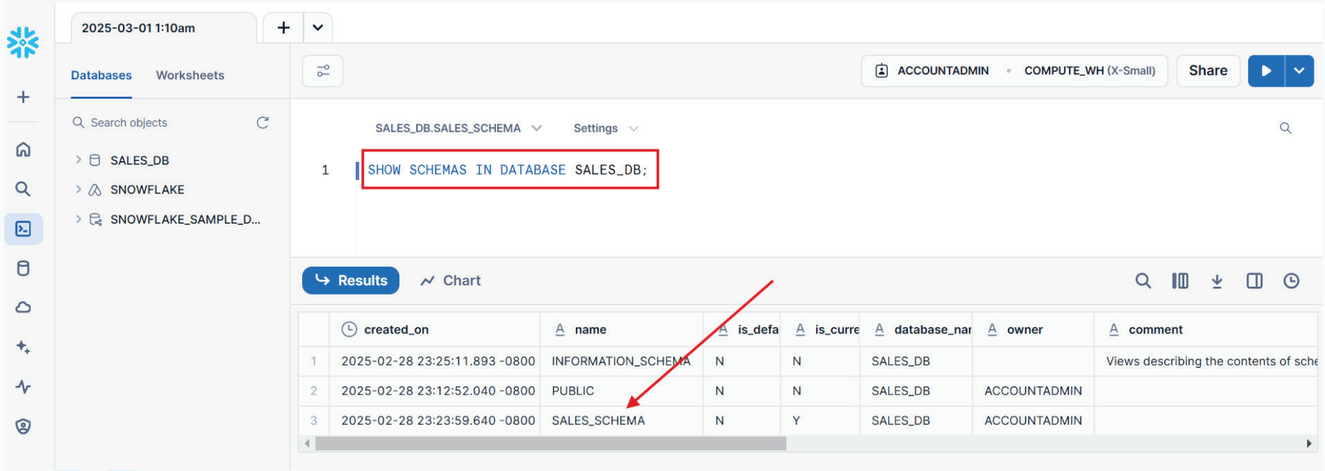
Step 4b: Create a ‘Schema’:

A schema is a **logical** container for the **table**. Run the ‘CREATE SCHEMA **SALES_DB.SALES_SCHEMA;**’ SQL queiry to create a schema.

1



2 **NOTE: Verify the schema was created**



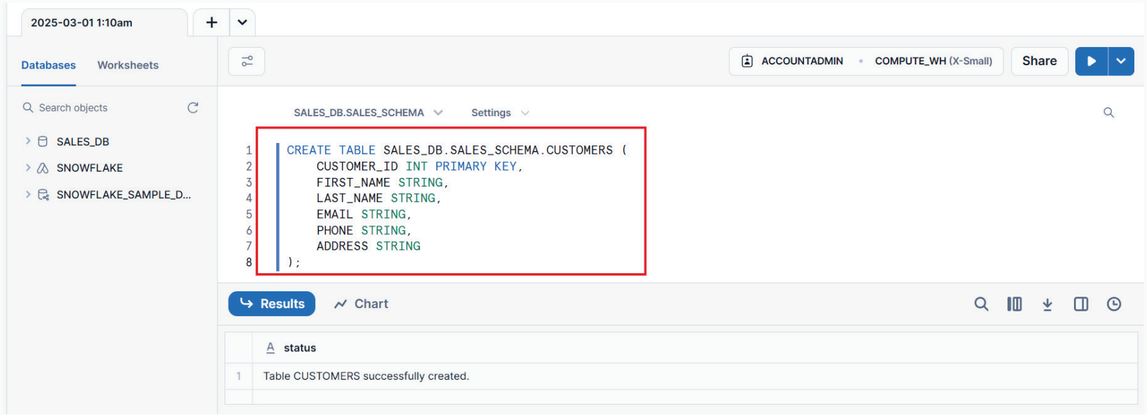
Step 4

Set Up Snowflake (Analytical DB)

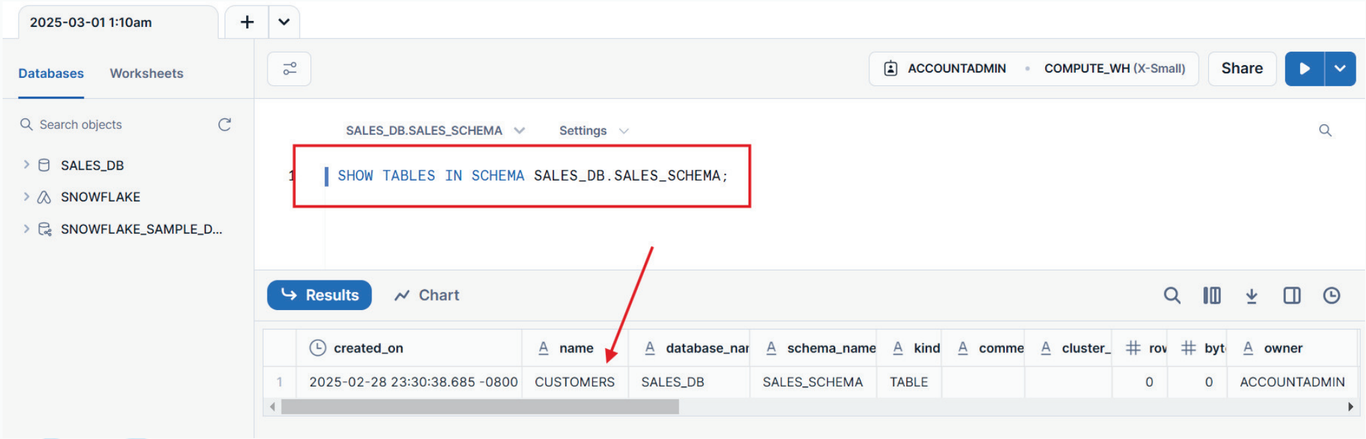
Step 4c: Create the ‘customers’ Table:

Run the following SQL command to create the customers table as shown in the screenshots

1



2 **NOTE: Verify the table ‘customers’ was created**



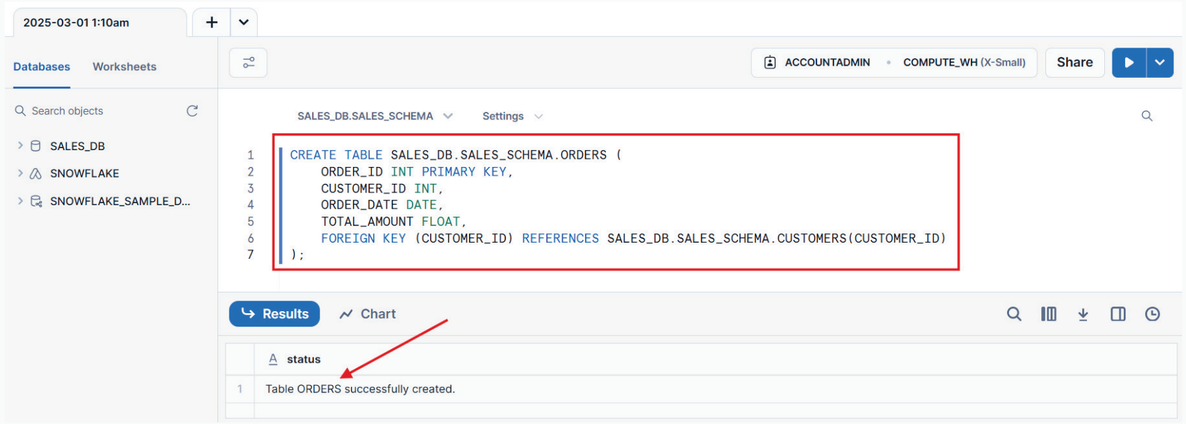
Step 4

Set Up Snowflake (Analytical DB)

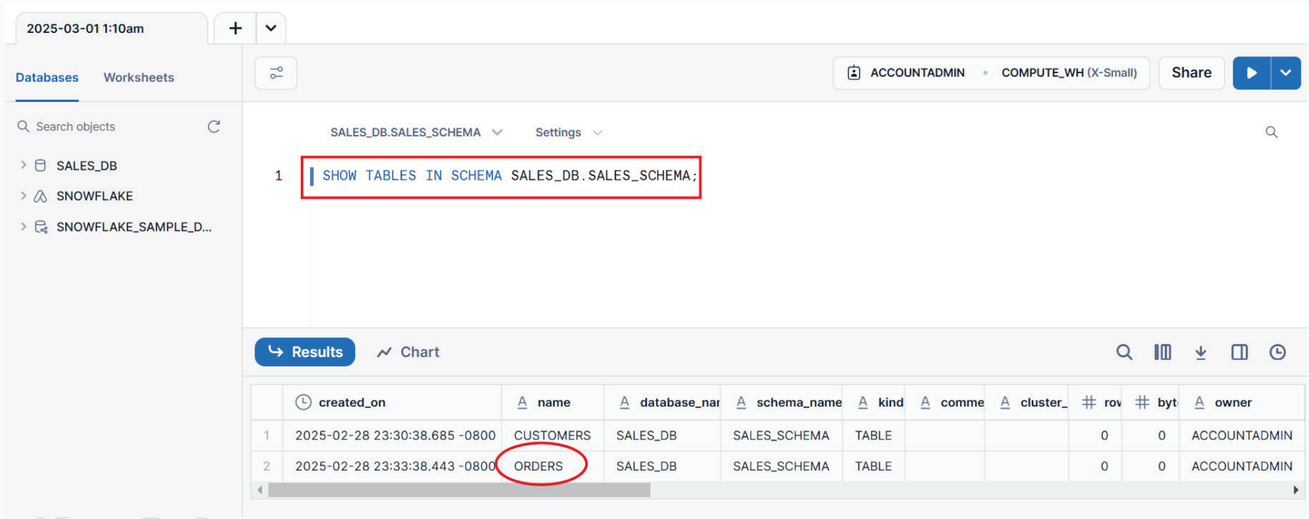
Step 4d: Create the ‘orders’ Table:

Run the following SQL command to create the customers table as shown in the screenshots

1



2 **NOTE: Verify the table ‘orders’ was created**

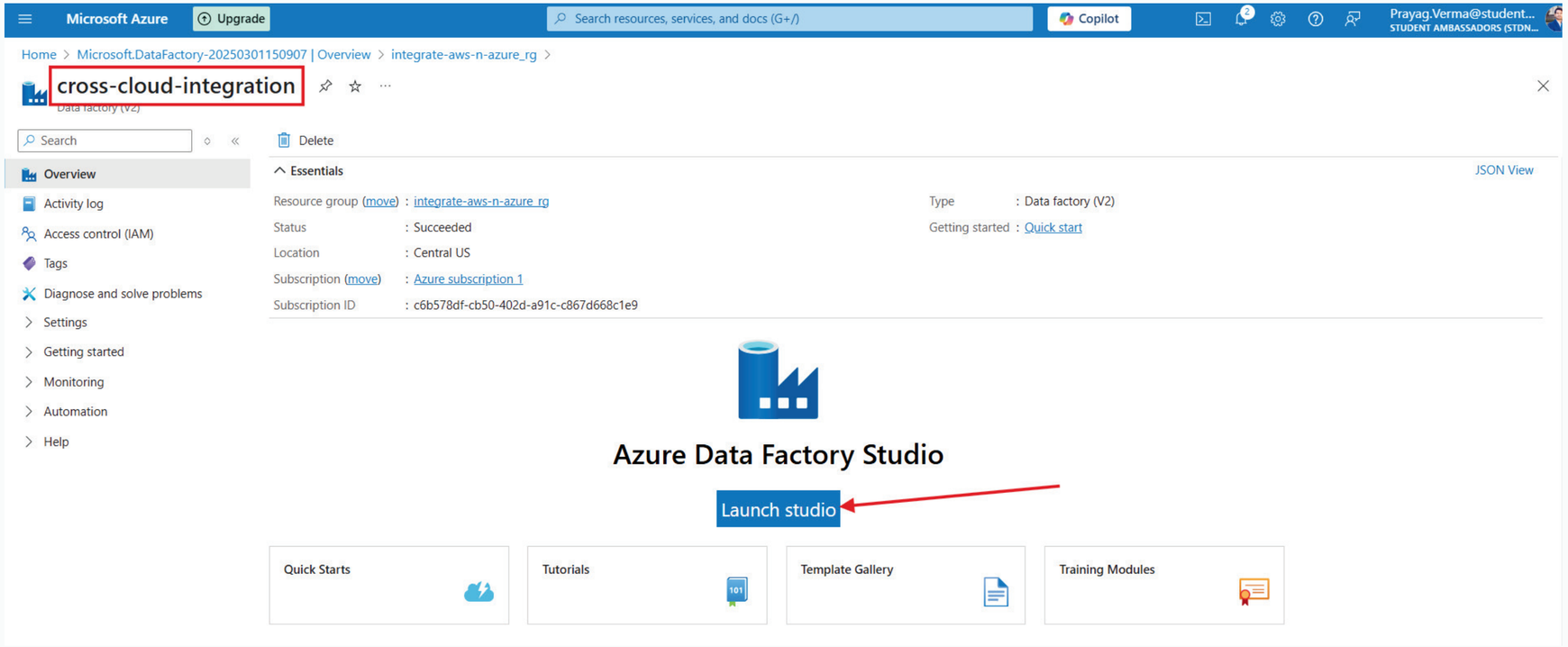


Step 5

Create Linked Services In ADF

Step 5a: Navigate to the **Azure dashboard**, find **Data Factory**, and **create** a new data factory named '**cross-cloud-integration.**' Launch it and follow the steps provided in the **next slide**.

1

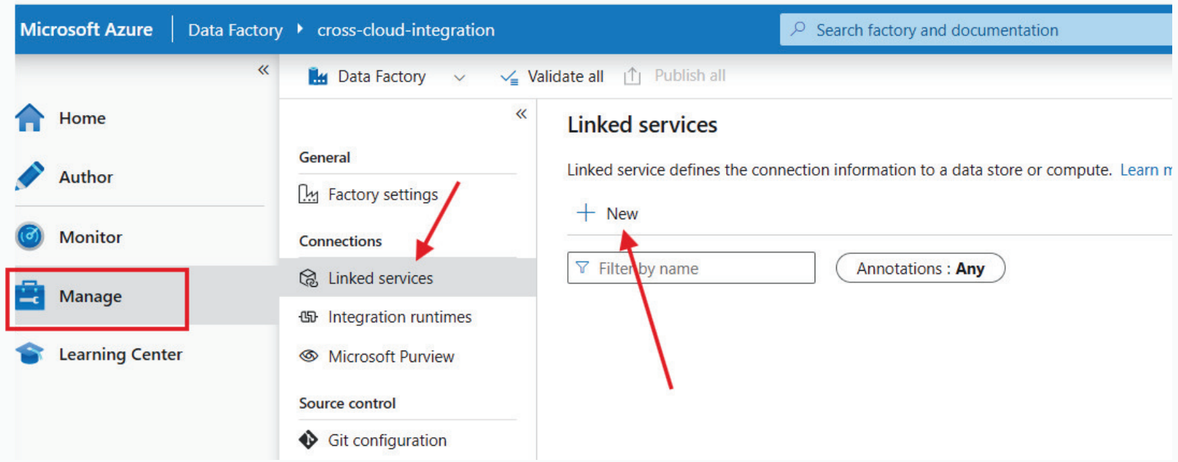


Step 5

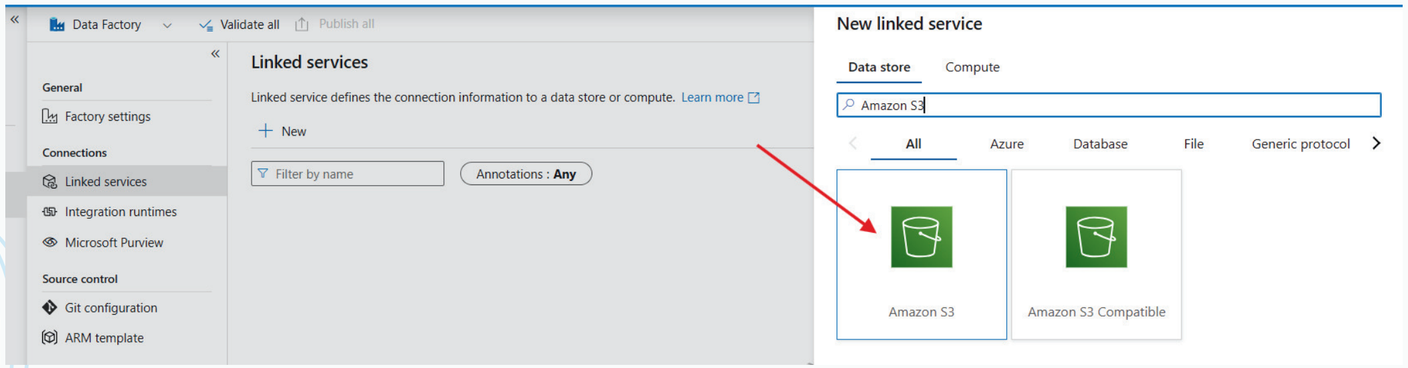
Create Linked Services In ADF

Step 5b: For AWS S3 - Go to Azure Data Factory Studio → **Manage** → **Linked Services** → **+ New** → Search for **Amazon S3** and select it. Then, name it as '**AmazonS3LinkedService**', enter details Access Key ID, Secret Access Key, Bucket name, Region, and then, Click '**Test connection**' then **Create.**

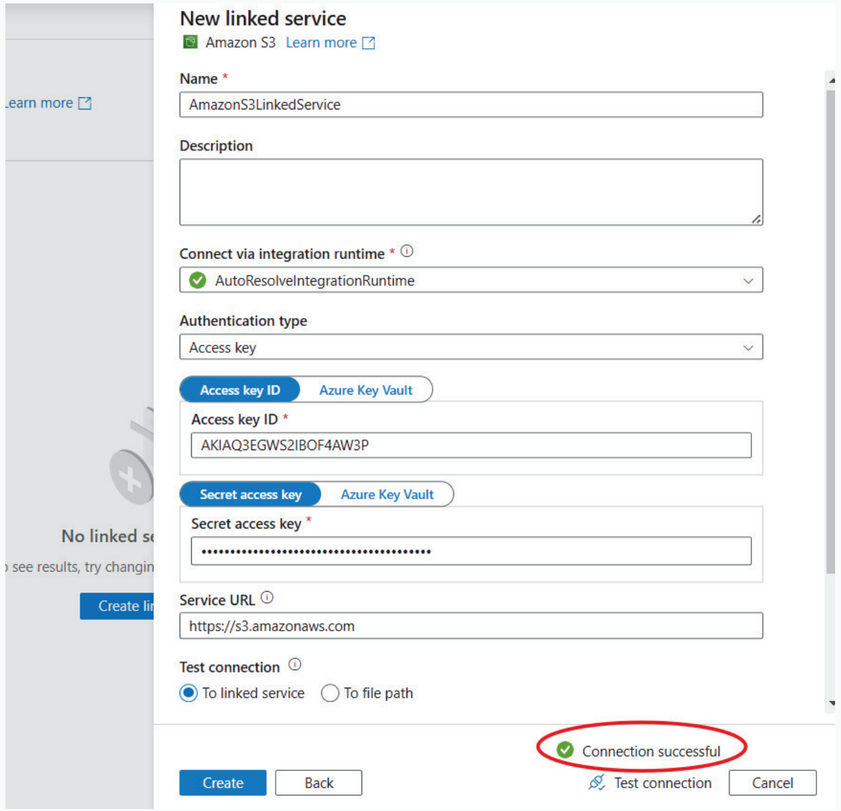
1



2



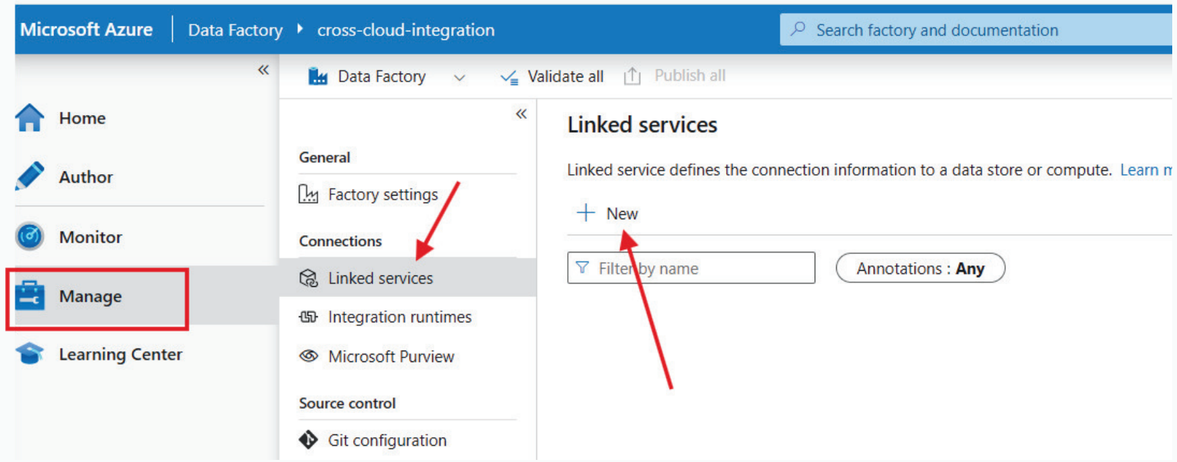
3



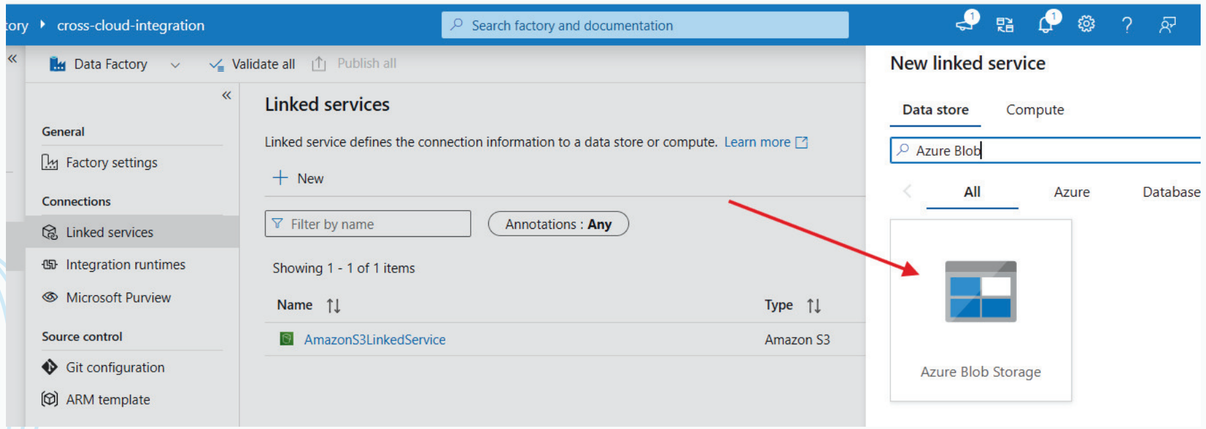
Step 5 Create Linked Services In ADF

Step 5c: For Azure Blob Storage - Go to Azure Data Factory Studio → **Manage** → **Linked Services** → **+ New** → Search for **Azure Blob Storage** and select it. Then, name it as '**AzureBlobLinkedService**', select Authentication type 'Account Key', Account name, and Account key, and then, Click '**Test connection**' then **Create.**

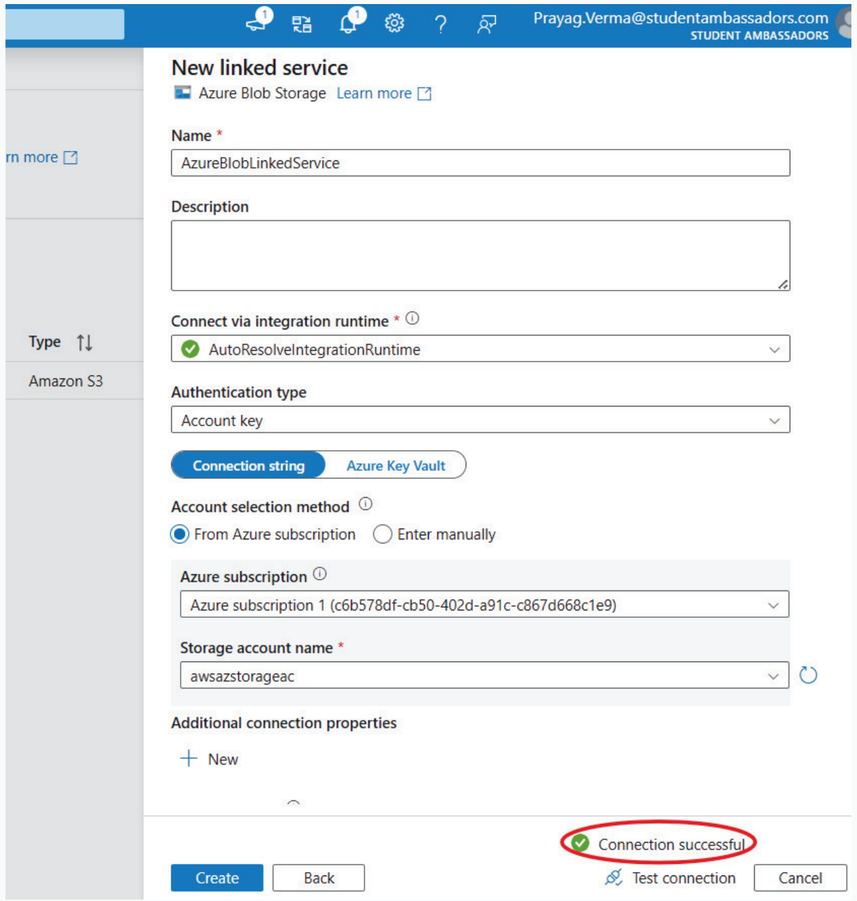
1



2



3

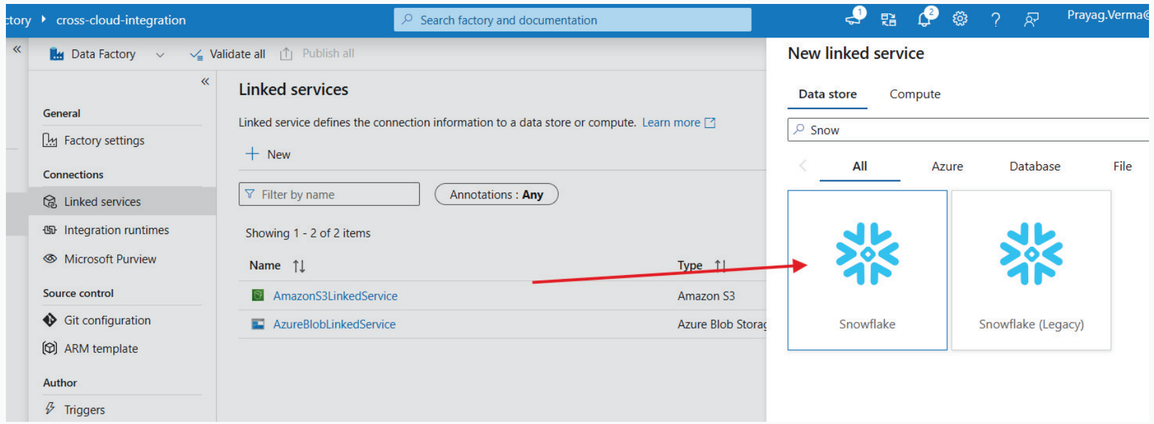


Step 5

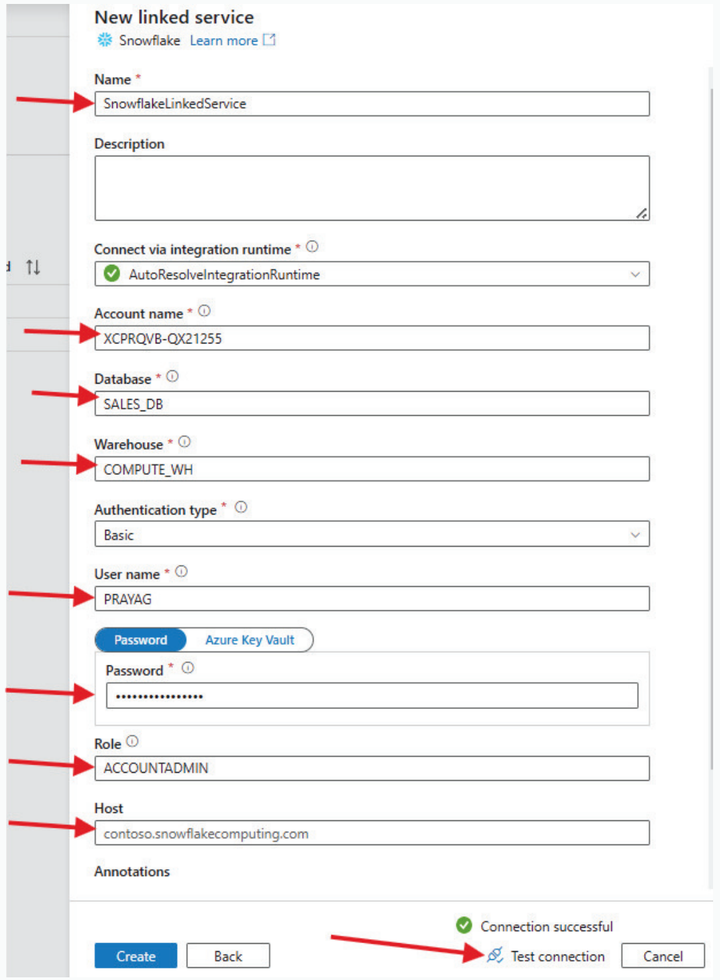
Create Linked Services In ADF

Step 5d: For Snowflake- Go to Azure Data Factory Studio → **Manage** → **Linked Services** → **+ New** → Search for **Snowflake** and select it. Then, name it as '**SnowflakeLinkedService**', select name, Account, Database, Schema, Warehouse, Authentication type, and User name, then, Click '**Test connection**' then **Create.**

1



2



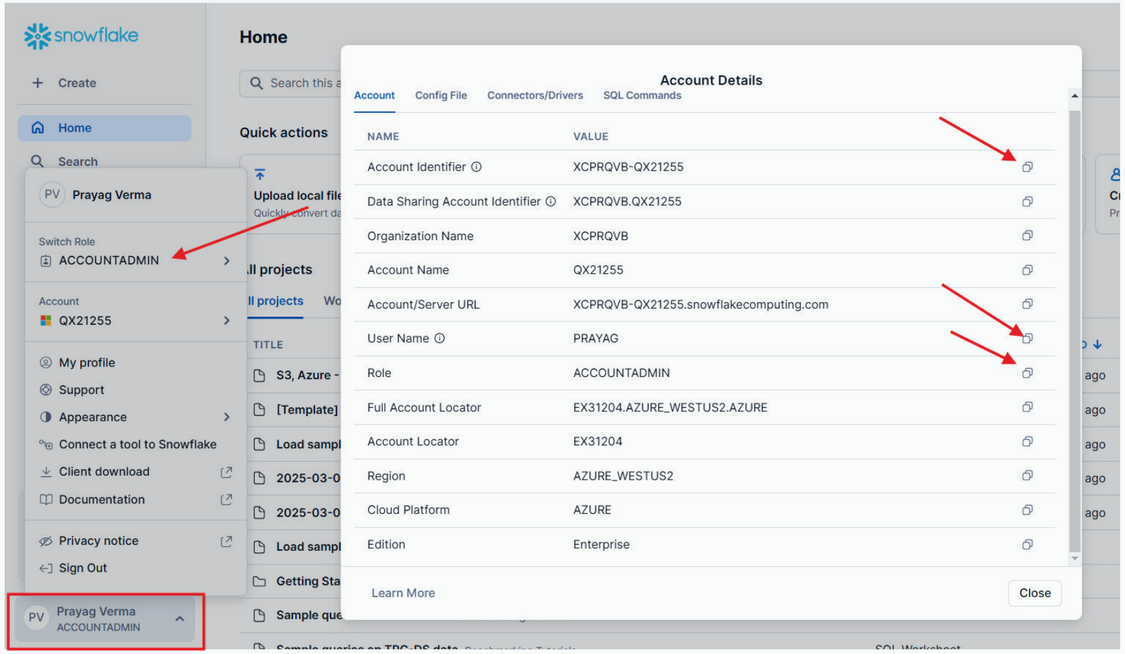
Note: I you don't know how to find these informations, refer to the next slide **Step 5e**!

Step 5 — Create Linked Services In ADF

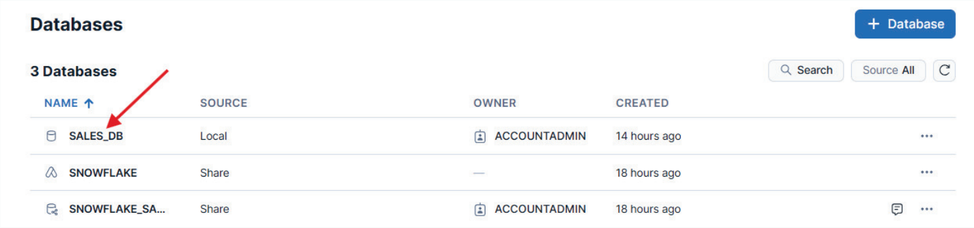
Step 5e: For Snowflake - Skip this slide if you've already set up & tested the **Sknowflake Linked Services**!

Note:

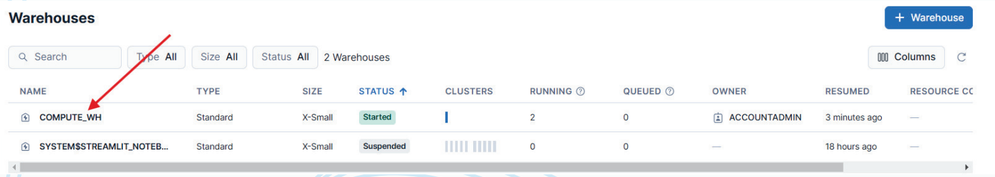
1. To find **Account Name**, Go to **Snowflake Dashboard → Profile → Account → View Account Details,** then Copy '**Account Identifier**' and past in **Account Name**.



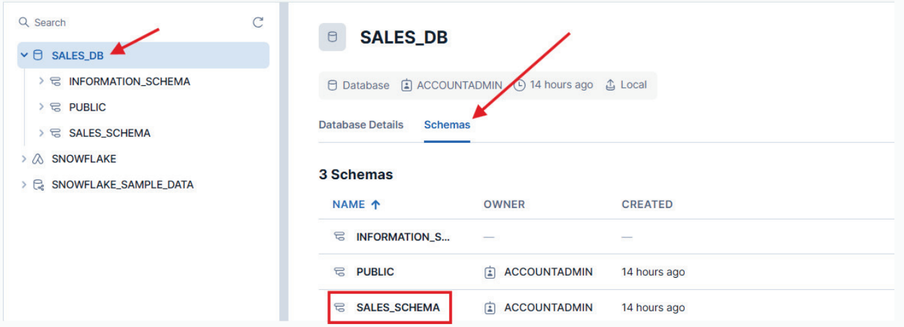
2. To find **Database Name**, Go to **Snowflake Dashboard → Admin → Data**, then all DBs is displyed.



3. To find **Warehouse**, Go to **Snowflake Dashboard → Admin → Warehouses**, then all warehouses is displyed.



4. To find **Database Schema**, Go to **Snowflake Dashboard → Admin → Data →** select **<your_database>** → Click on '**Schemas**' Tab**.**

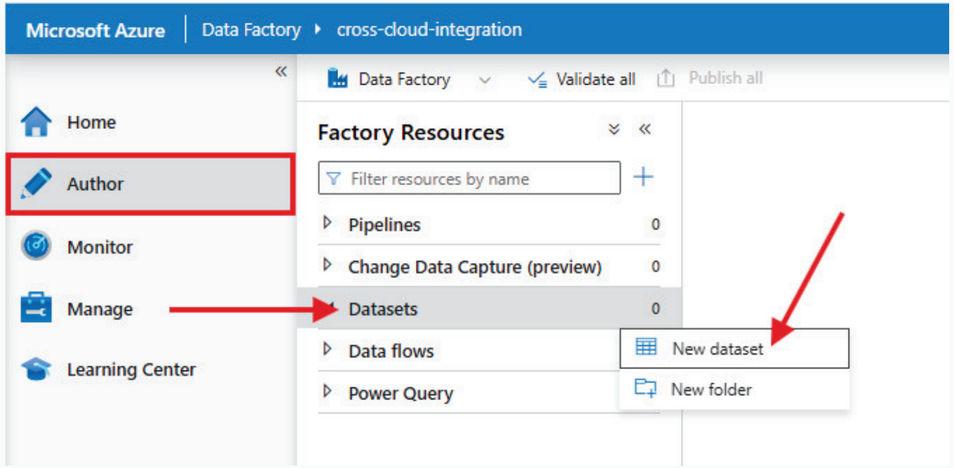


5. **User Name** and **Password** are the one you used to login to **Snowflake**

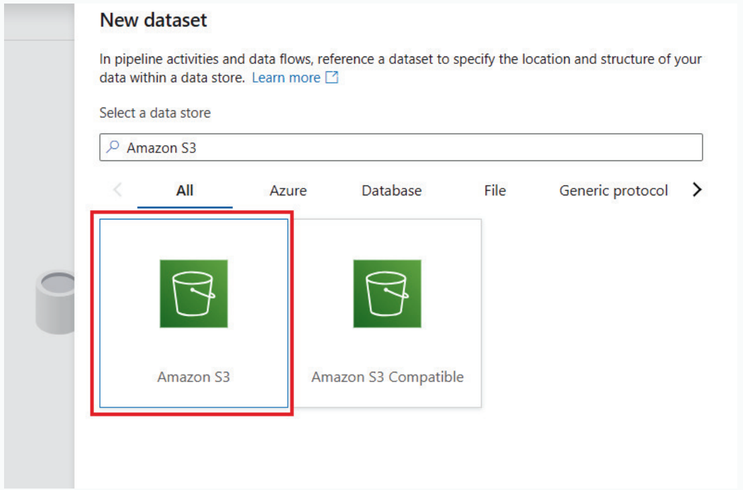
Step 6 Create AWS S3 Datasets In ADF

Step 6a: Go to **Data Factory dashboard** → **Author** → **Datasets** → **New dataset** → Search & select **Amazon S3** → **Select 'Delimeited Text' CSV** → and fill in all the informations as shown in **screenshot 4** and **Click OK**.

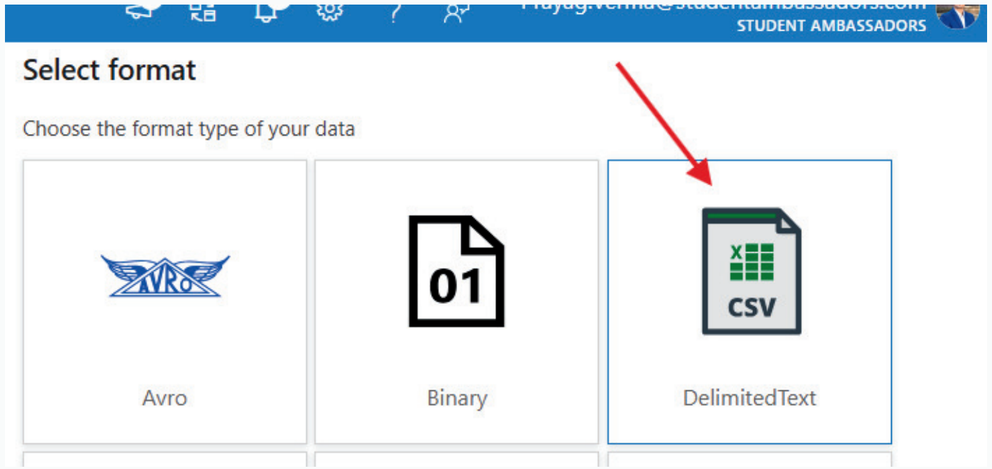
1



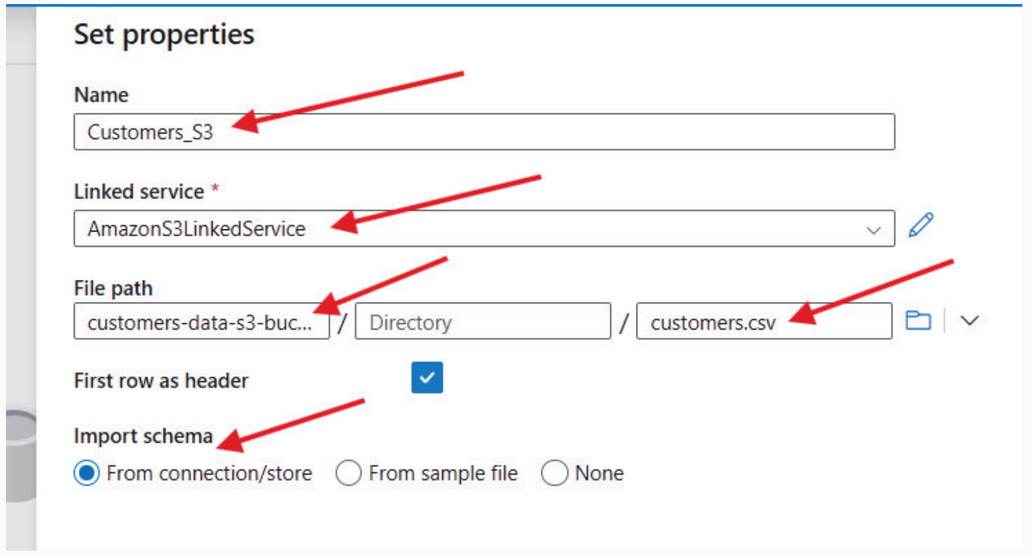
2



3



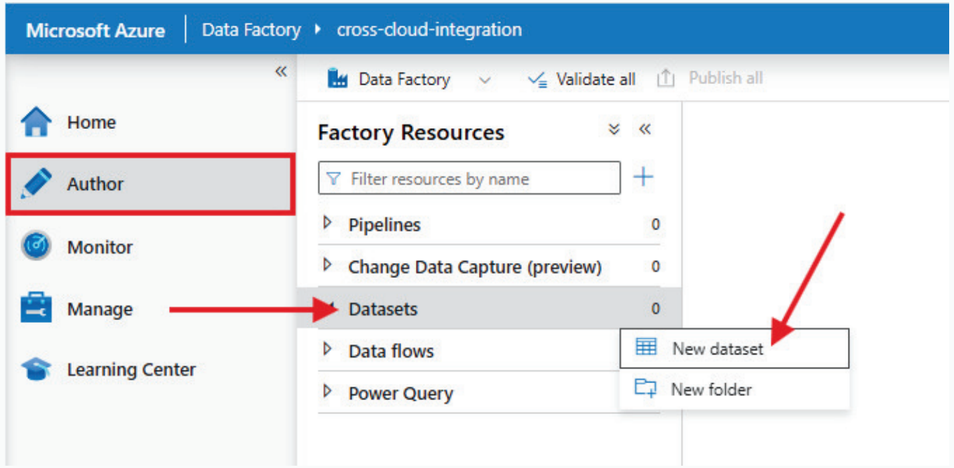
4



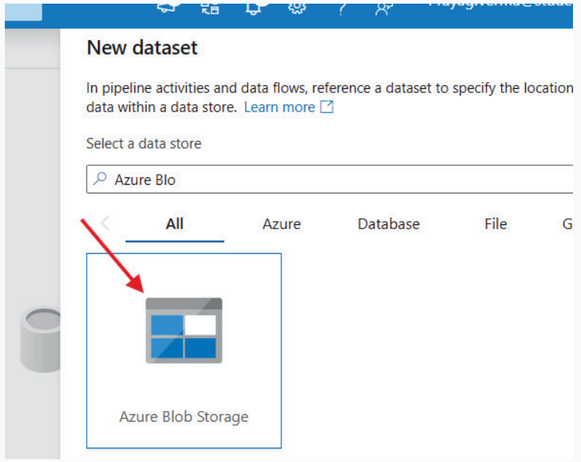
Step 6 — Create Azure Blob Datasets In ADF

Step 6b: Go to **Data Factory dashboard** → **Author** → **Datasets** → **New dataset** → Search & select **Amazon S3** → **Select 'Delimeited Text' CSV** → and fill in all the informations as shown in **screenshot 4** and **Click OK**.

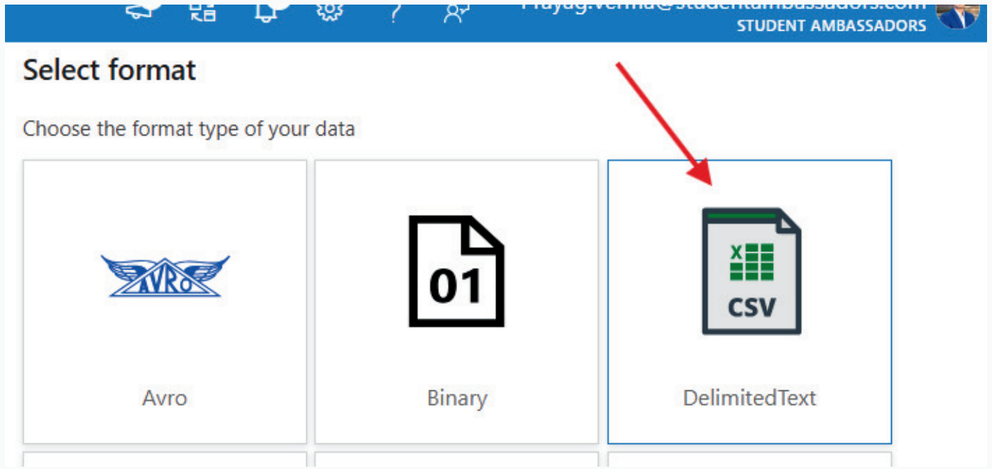
1



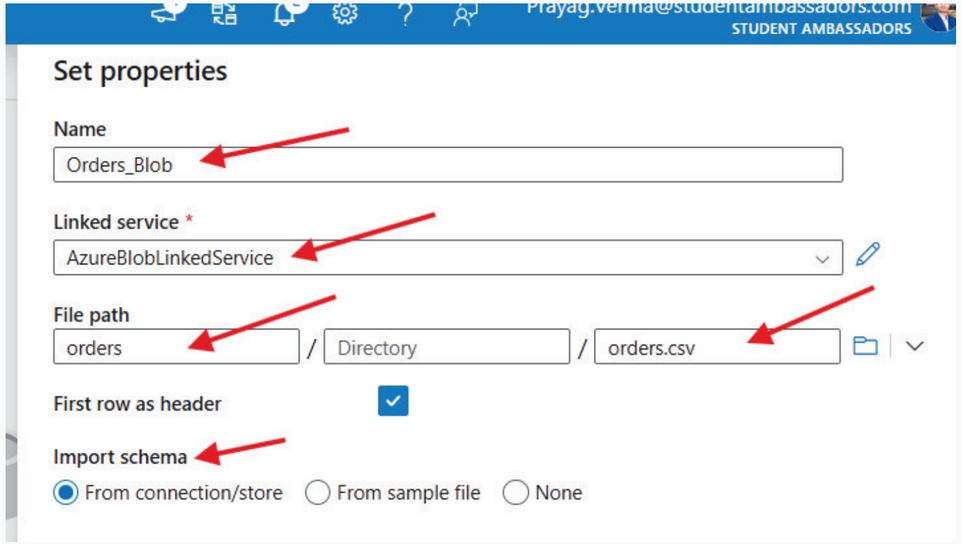
2



3



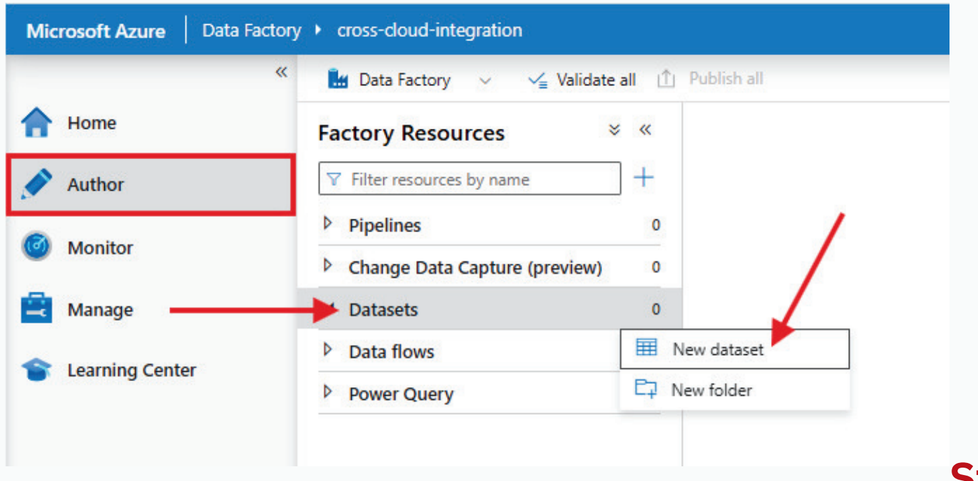
4



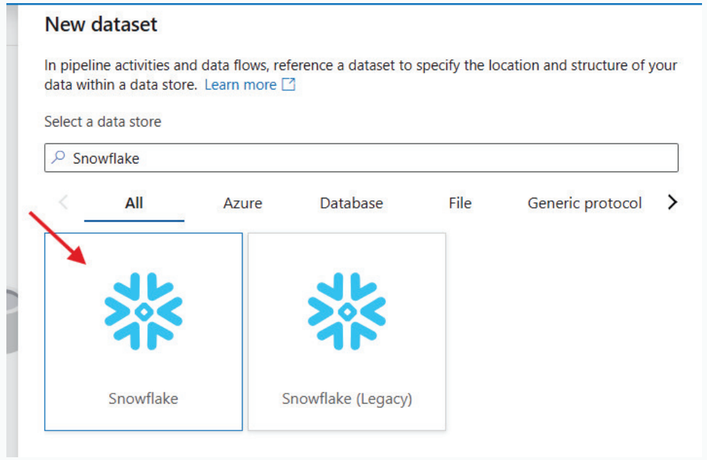
Step 6 — Create Snowflake Datasets In ADF

Step 6c: Go to **Data Factory dashboard** → **Author** → **Datasets** → **New dataset** → Search & select **Snowflake** → enter Name '**Customers_Snowflake**' → Select '**SnowflakeLinkedService**' → Select **Table name** '**Customers**' then **Click OK**.

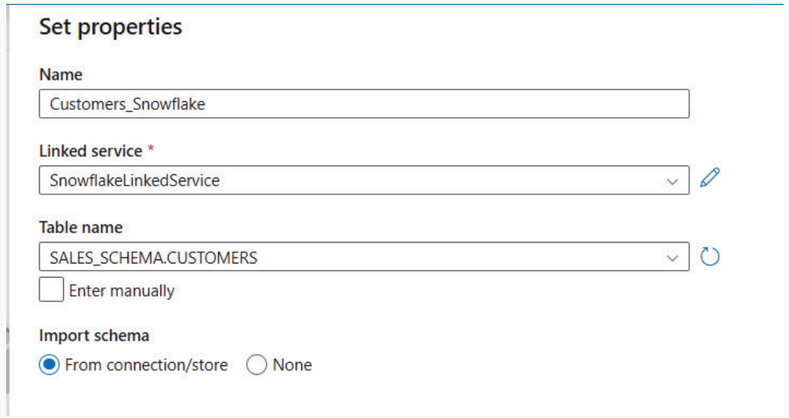
1



2

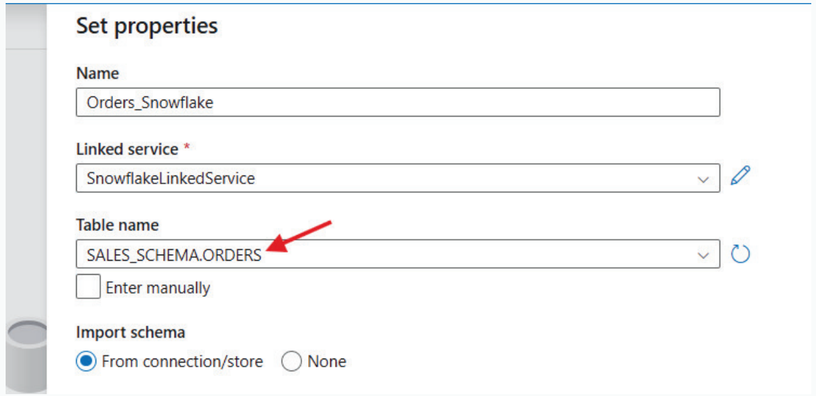


3



Step 6d: Repeate the same steps and create another **Datasets** with name '**Orders_Snowflake**' and select **Orders** table in Table Name

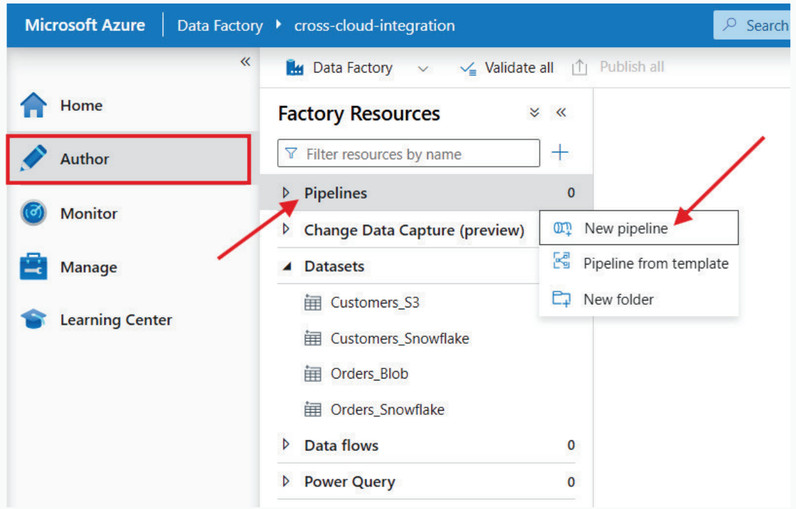
4



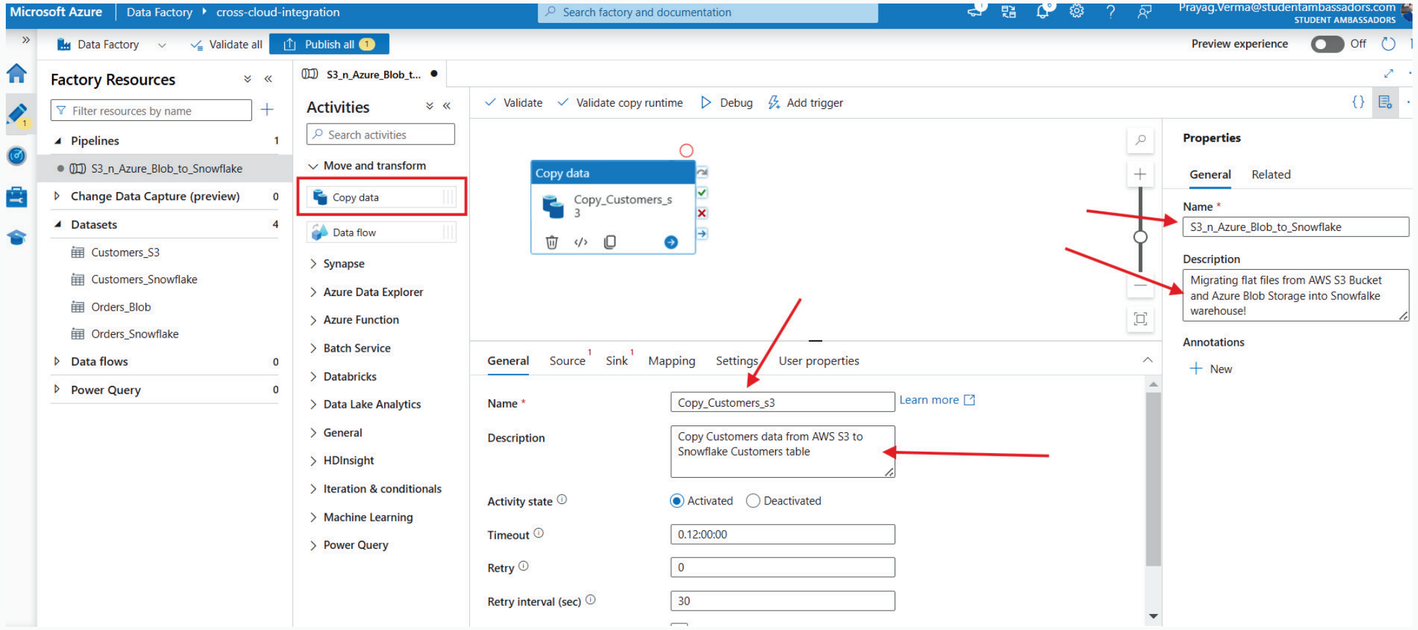
Step 7 Create Azure Pipeline

Step 7c: Go to **Data Factory** dashboard → **Author** → **Pipelines** → **New pipeline** → Name it **'S3_n_Azure_Blob_to_Snowflake'** → Add **'Copy data'** activity as shown below.

1



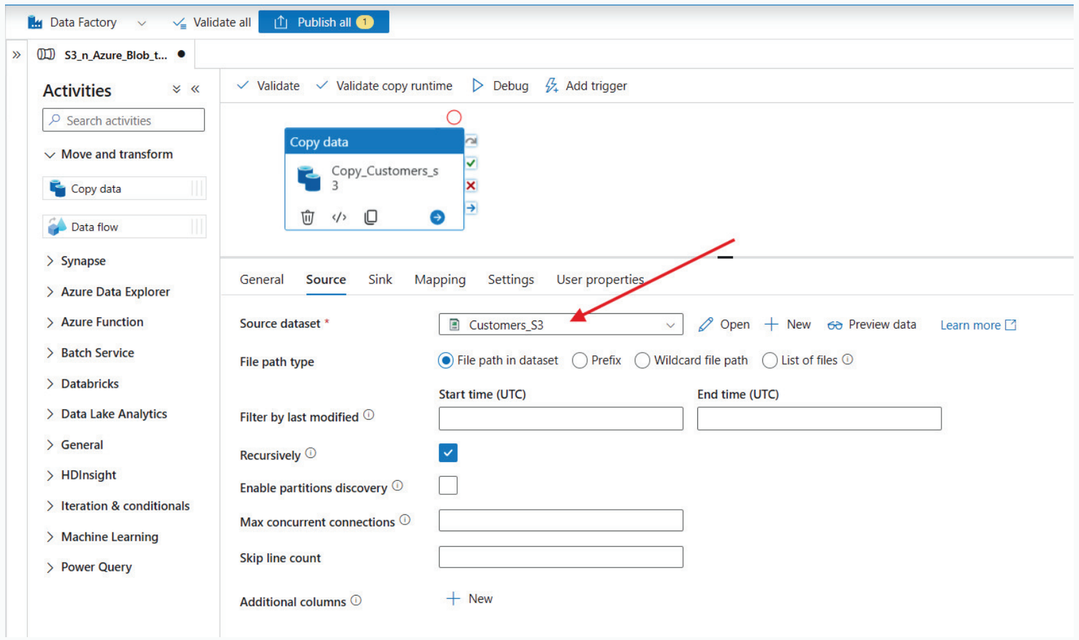
2



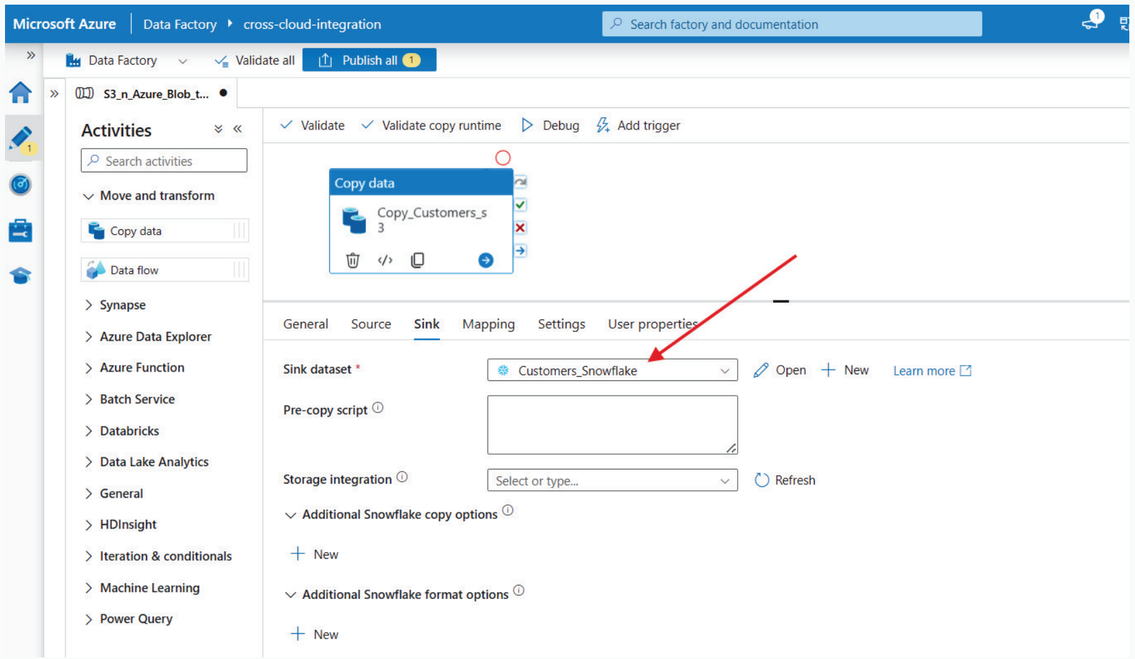
Step 7 Create Azure Pipeline

Step 7c: Under **Source tab,** Select **Customer_S3**

1



2 Under **Sink tab,** Select **Customer_Snowflake**

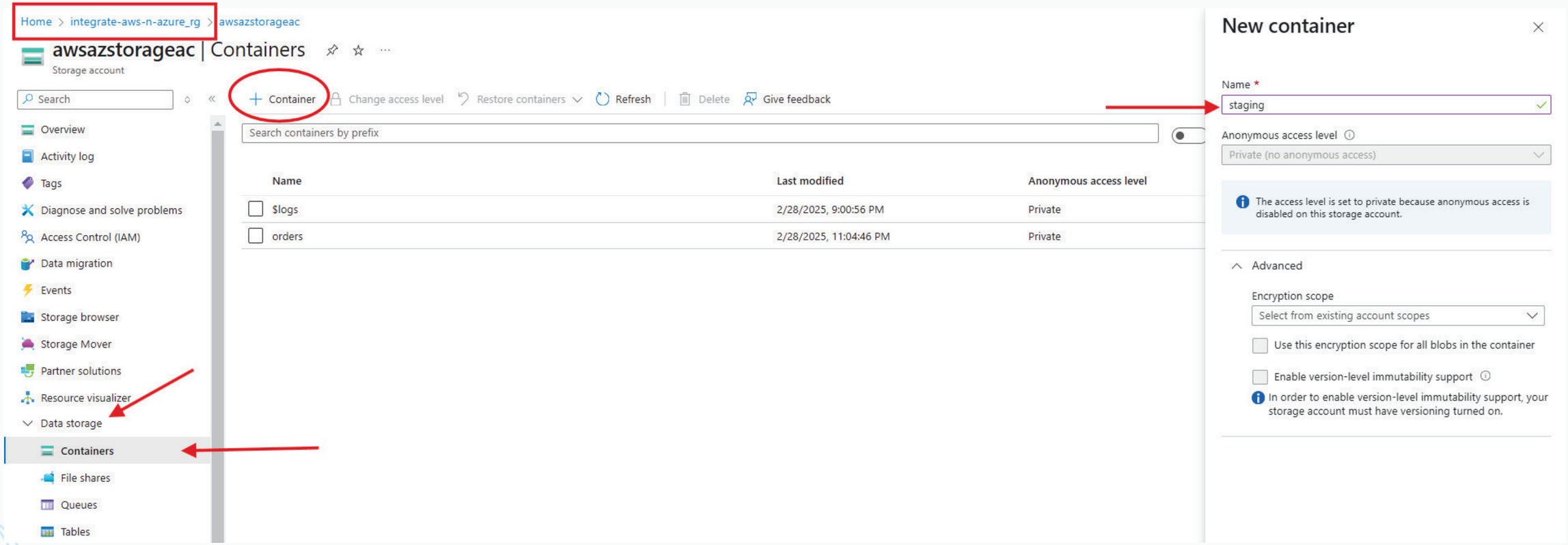


Note:

1. Under "**Mapping**" tab, Schemas should be imported if not import it and make sure it's mapped properly.
2. Here, data moves from AWS S3 **Customers.csv** → Snowflake's **Customers** table.
3. Now, do not close the current browser tab, and open a new tab in the same browser and see the next **slide**.

Step 7 Create Azure Pipeline

Step 7c: Now, go to **Azure dashboard** → **resource group** → select your container → **Data storage** → **Containers** → **+ Containers**, name it as **'staging',** once container is created go to the next slide:



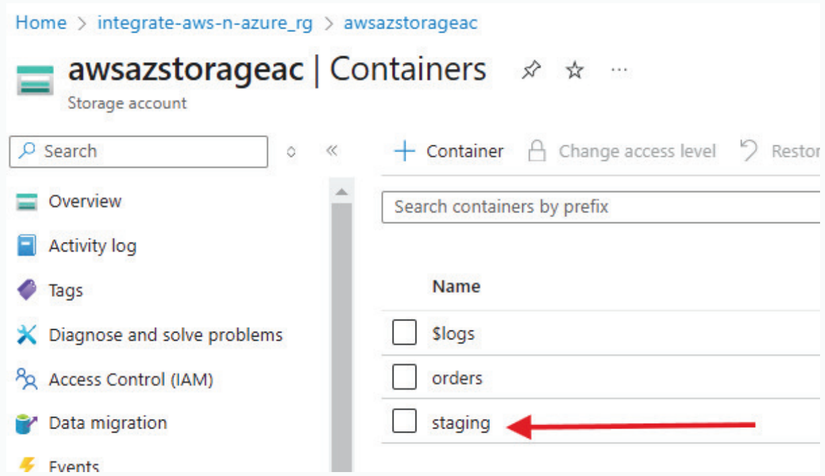
Step 7 — Create Azure Pipeline

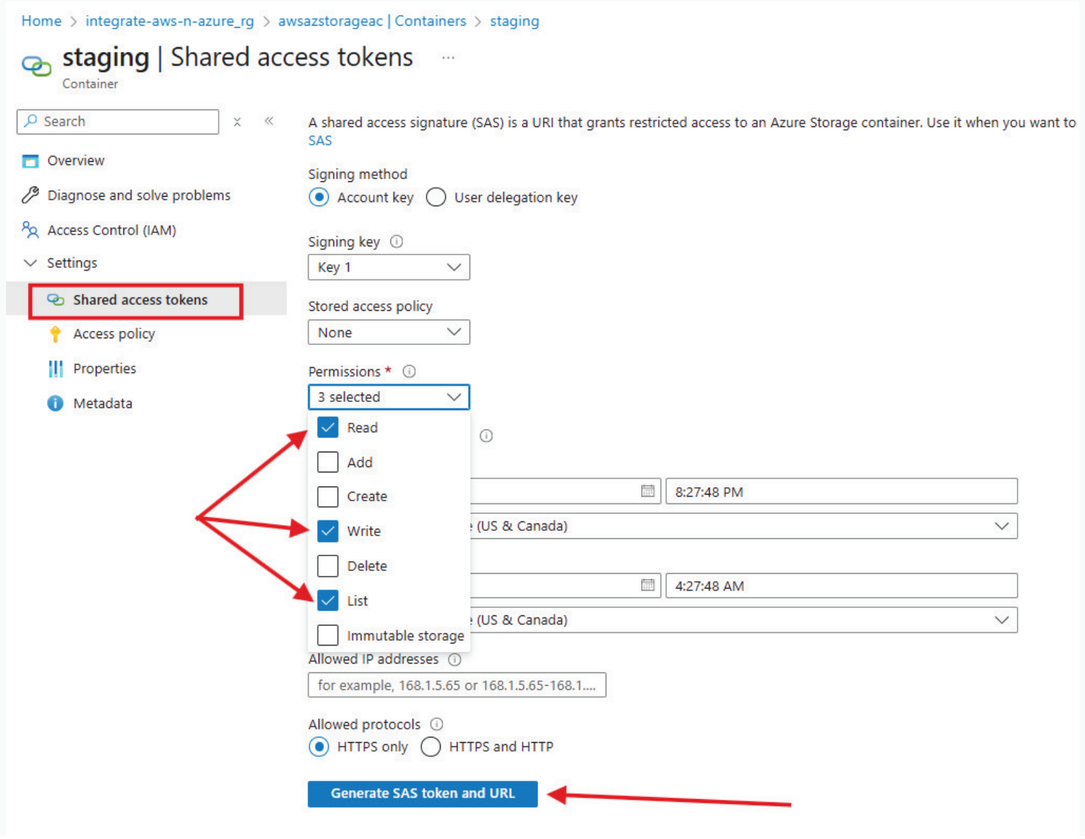
Step 7c: Now click & open the container ‘staging’ the one you just created, then click on ➜ **Shared access tokens** ➜ select **read**, **Write**, and **List** permissions, then select ➜ **Start and expiry date/time** ➜ now Click on ‘**Generate SAS token and URL**’.

Note:

1. Copy ‘**Blob SAS token**, **Blob SAS URL**’ and **store** them somewhere as you will not be able to access it later.

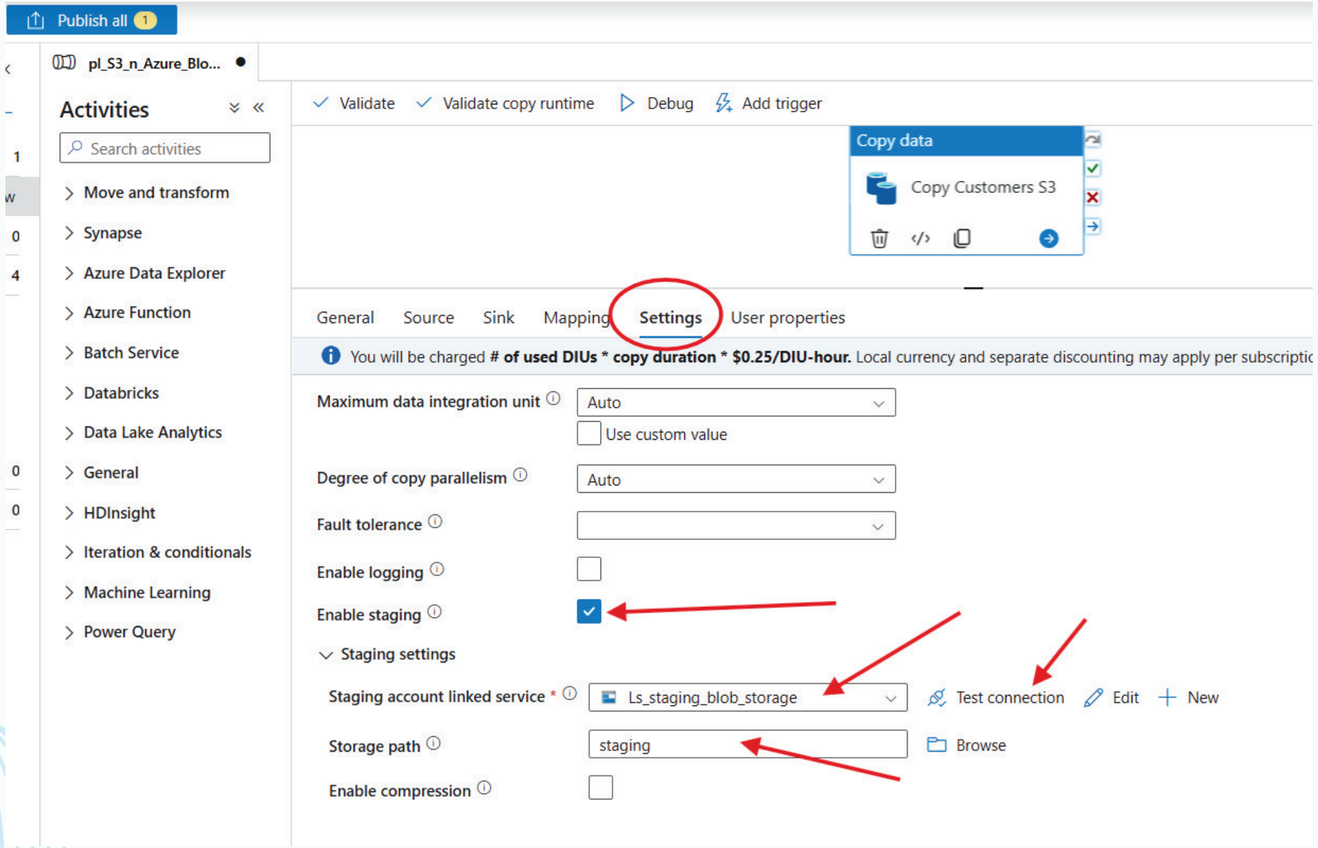
1

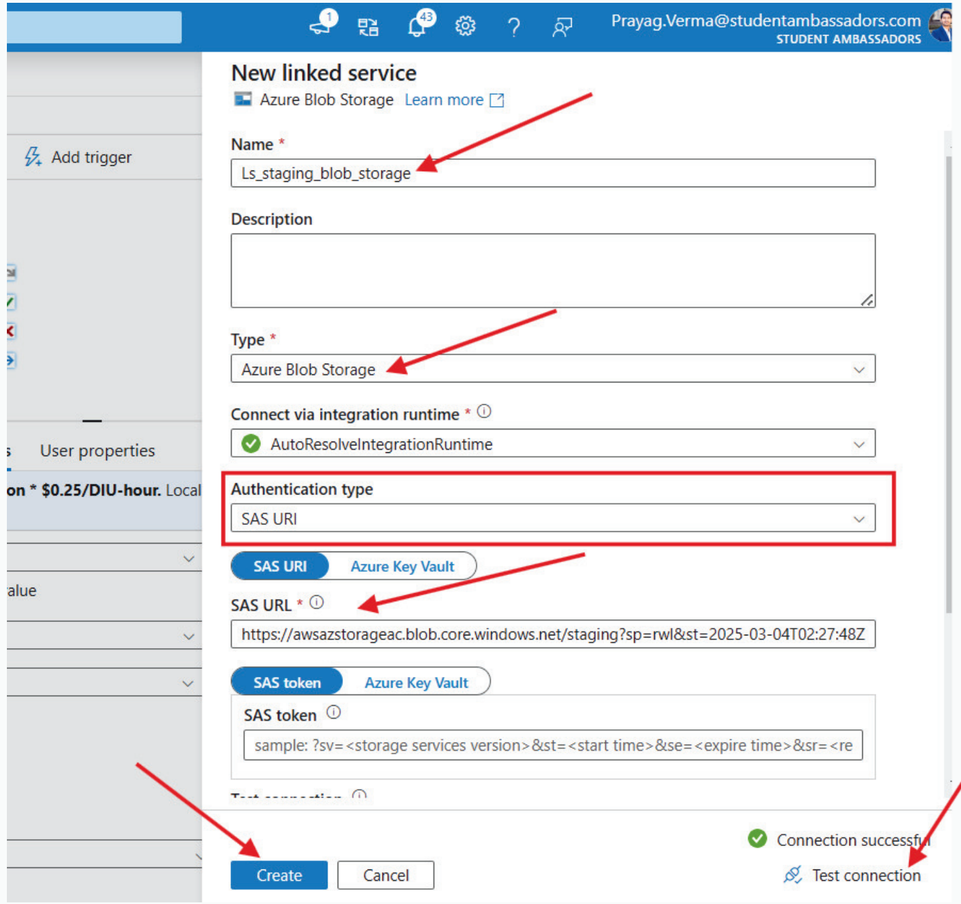




Step 7 — Create Azure Pipeline

Step 7c: Now, Return to the Azure Data Factory tab. Under Settings→ ensure Enable staging is selected → enter '**Storage path**' as '**staging**' → Click '**+ New**', name it **Ls_staging_blob_storage**, and choose **SAS URI** for authentication type. Paste the **Blob SAS URL** (saved earlier refer to **page no 24**) into the SAS URL field → Click on **Test connection** if success click on '**Create**' else **re confirm** the **SAS URL** properly.



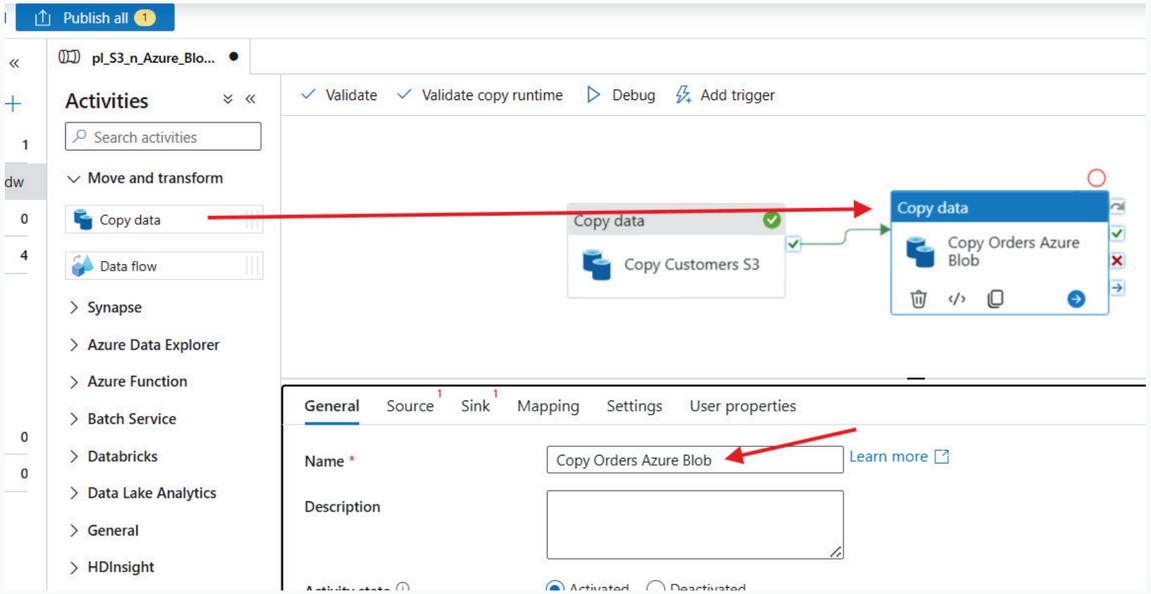


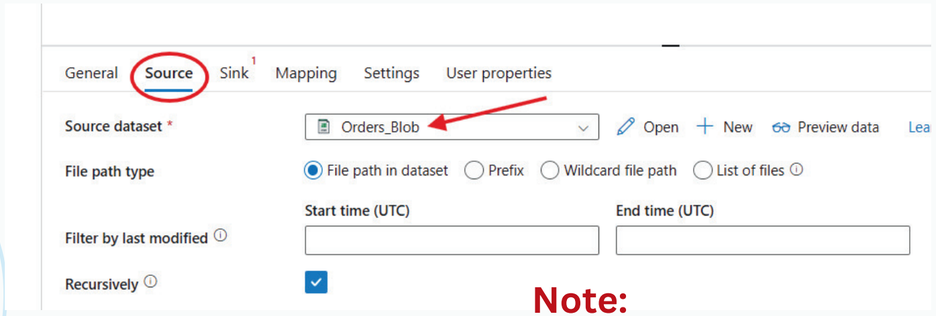
Now, Click on '**Publish all**' and save the changes.

Step 7 — Create Azure Pipeline

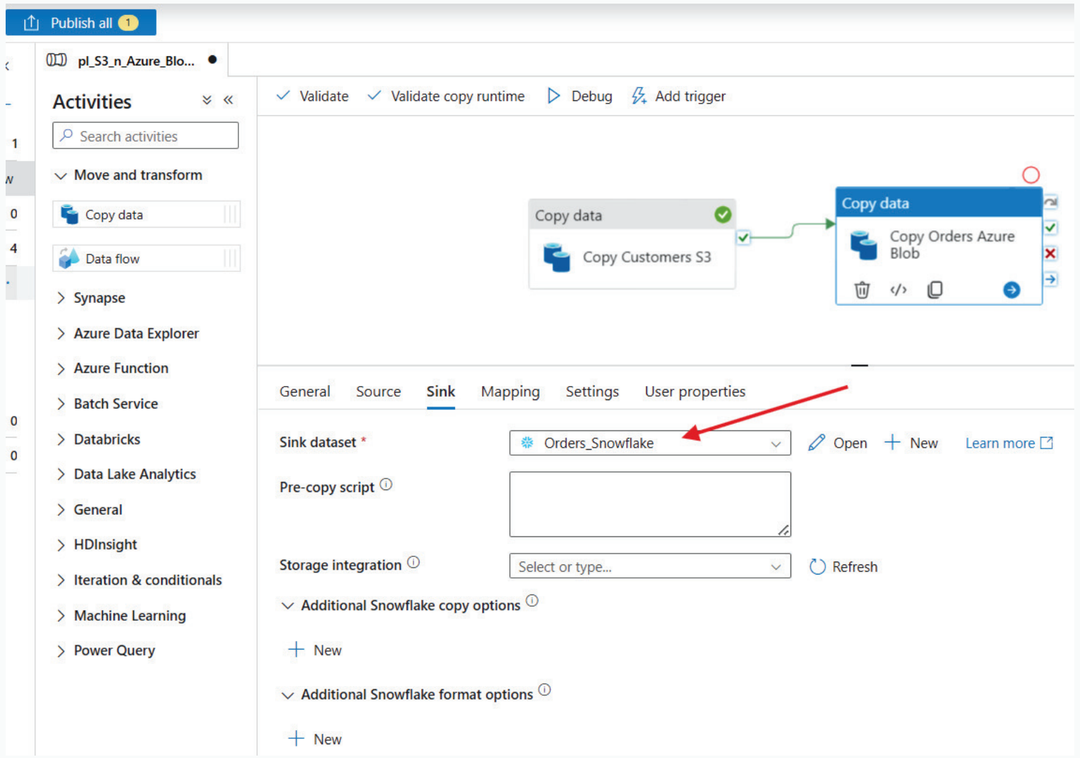
Step 7c: Similarly add another '**Copy Data**' activity for '**Orders**'

1 Under **Source tab,** Select **Orders_Blob**





2 Under **Sink tab,** Select **Orders_Snowflake**

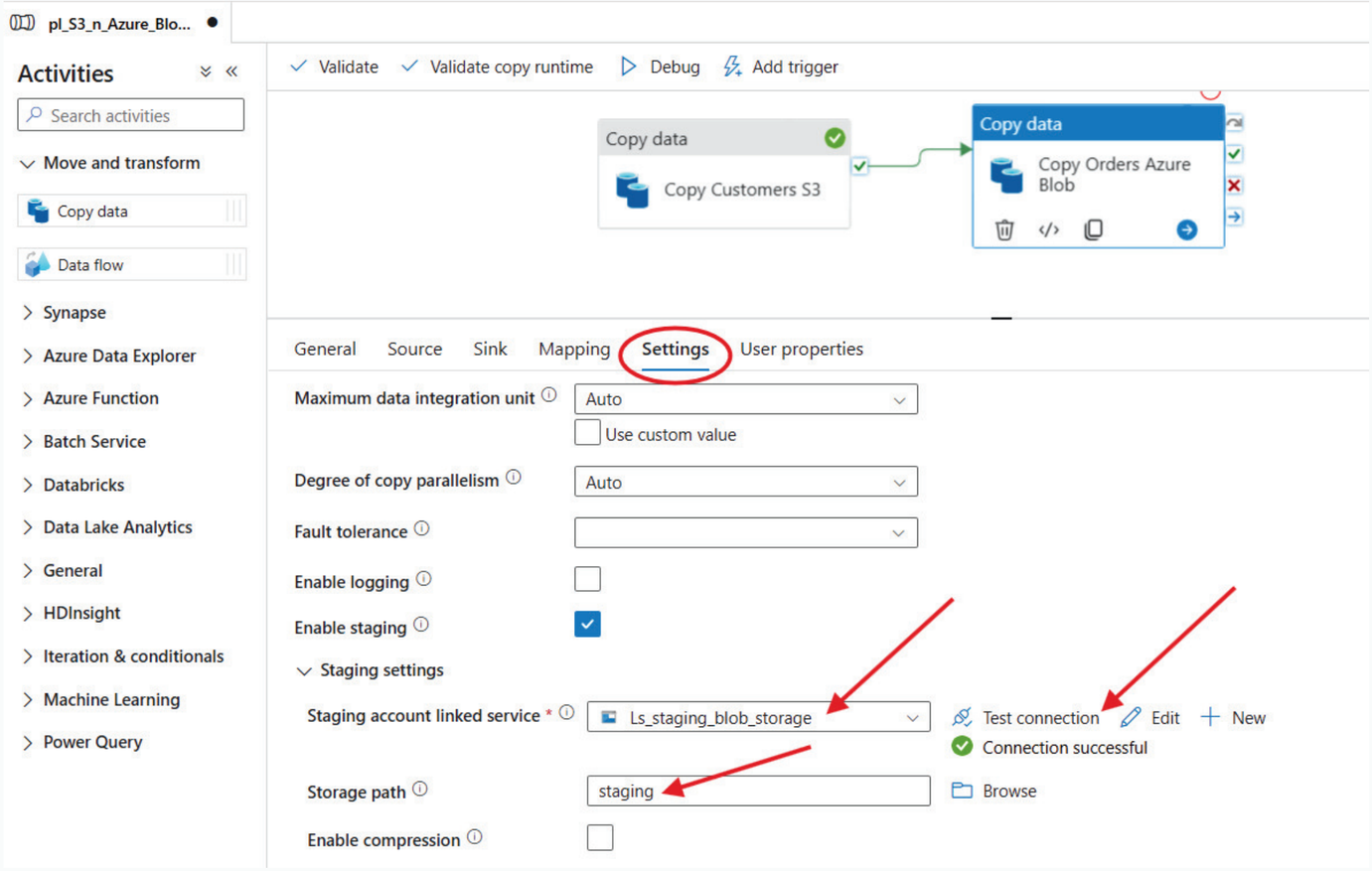


Note:

1. Under "**Mapping**" tab, Schemas should be imported if not, import and verify mapping properly.
2. Here, data moves from Azure Blob Container Storage **Orders.csv** → Snowflake's **Orders** table.

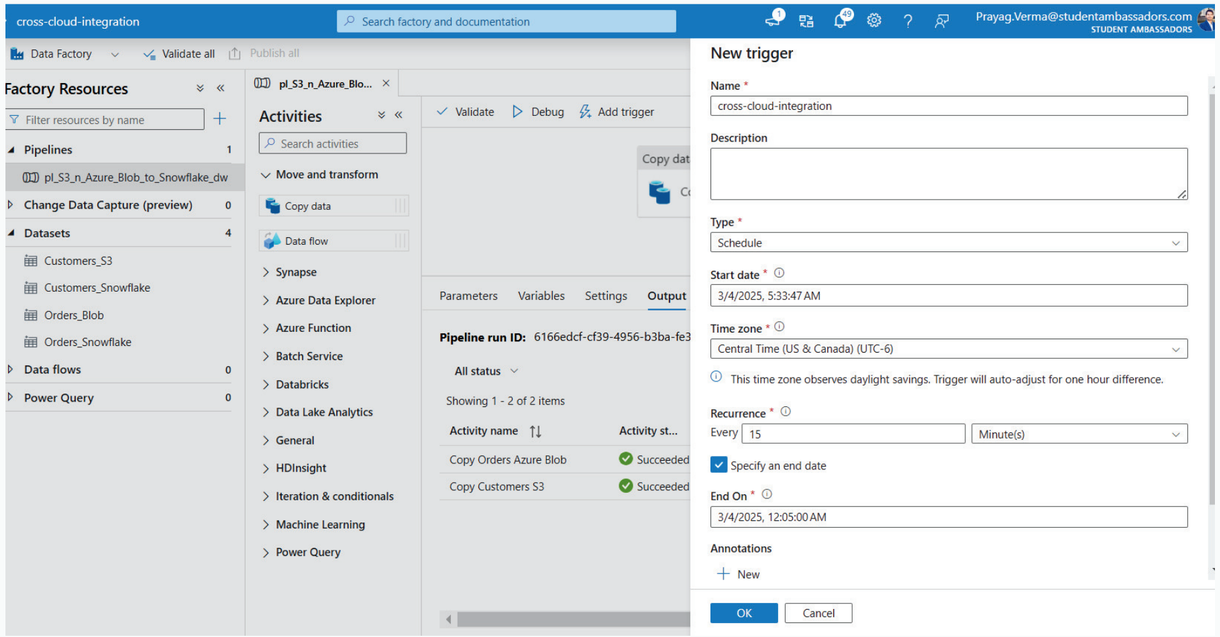
Step 7 Create Azure Pipeline

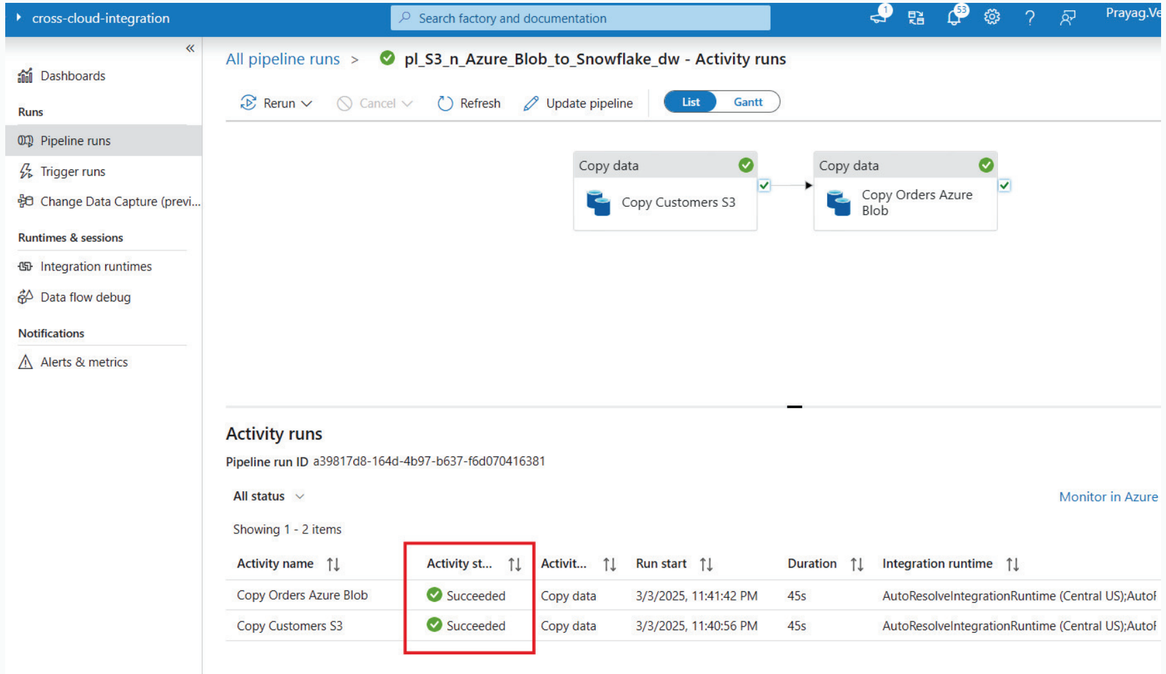
Step 7c: Under **Settings** → ensure **Enable staging** is **selected** → enter '**Storage path**' as '**staging**' → Select '**Ls_staging_blob_storage**' → **Test connection** if success click on '**Publish all**'.

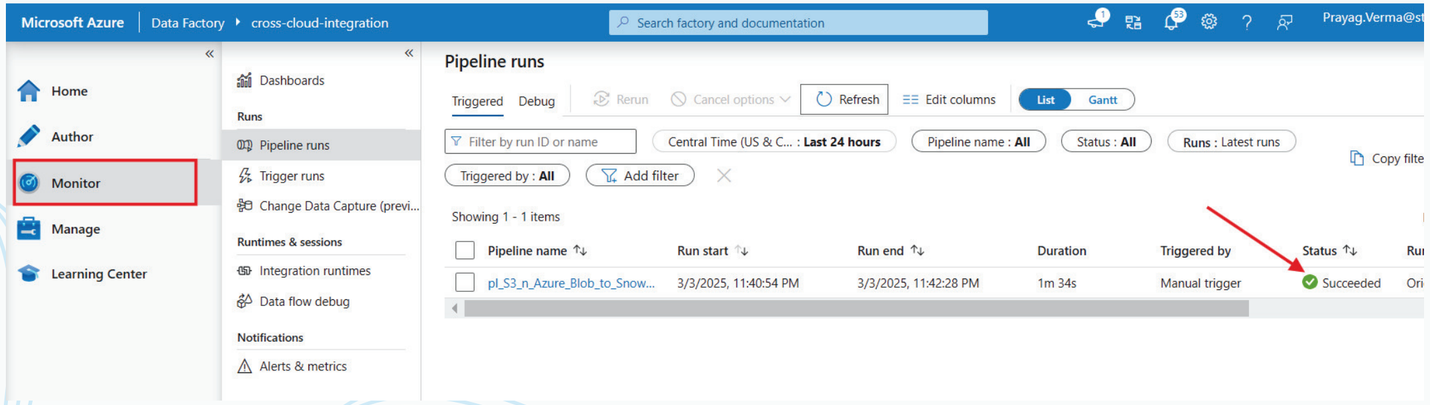


Step 8 Create ADF Trigger

Step 8a: ADF Studio → Author → Select **Pipeline** → Add Trigger → Configure Schedule → Publish & Test.



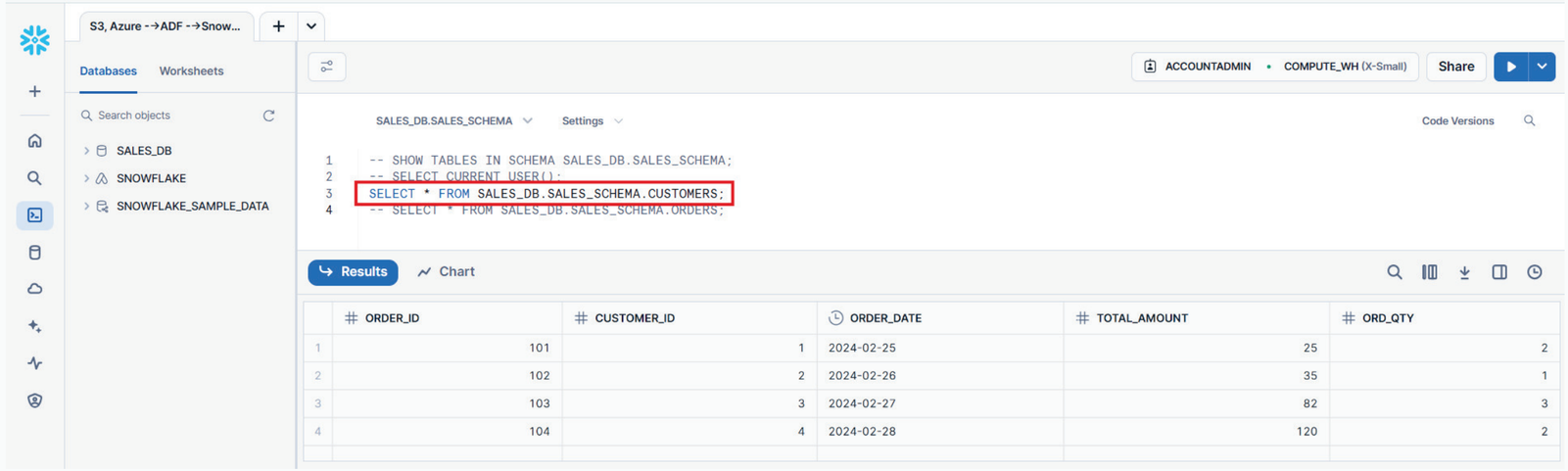




Step 9

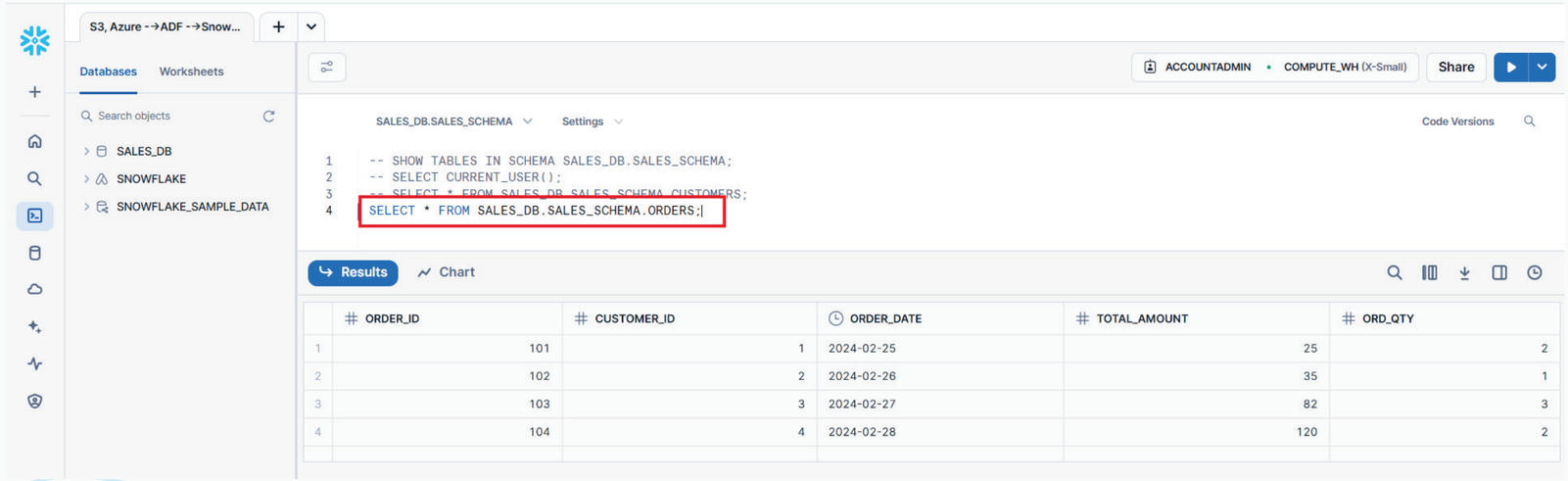
Data Load Review

SELECT * FROM SALES_DB.SALES_SCHEMA.CUSTOMERS;



	ORDER_ID	CUSTOMER_ID	ORDER_DATE	TOTAL_AMOUNT	ORD_QTY
1	101	1	2024-02-25	25	2
2	102	2	2024-02-26	35	1
3	103	3	2024-02-27	82	3
4	104	4	2024-02-28	120	2

SELECT * FROM SALES_DB.SALES_SCHEMA.ORDERS;



	ORDER_ID	CUSTOMER_ID	ORDER_DATE	TOTAL_AMOUNT	ORD_QTY
1	101	1	2024-02-25	25	2
2	102	2	2024-02-26	35	1
3	103	3	2024-02-27	82	3
4	104	4	2024-02-28	120	2

- THANK YOU -

Prayag Verma | Data Engineer

/prayagv /prayag-verma

Follow Me To Learn More!