**A**

**PROJECT REPORT**

**ON**

*“Configuration of Azure Data Factory and create pipelines to take data from Azure Blob and insert in into Azure SQL”*

**Bachelor of Technology**

****

**Poornima University, Jaipur**

**Session: 2018-2022**

**Submitted By:**

**Aarish Ahmed Siddique**

**Abhishek Singh**

**Anshita Goyal**

**Arushi Sukhwal**

**Teli Udayram**

**Submitted To:**

**Celebal Technologies**

**Prerequisites**

* **Azure subscription**

If you don't have an Azure subscription, create a free account before you begin.

* **Azure roles**

To create Data Factory instances, the user account that you use to sign in to Azure must be a member of the *contributor* or *owner* role, or an *administrator* of the Azure subscription.

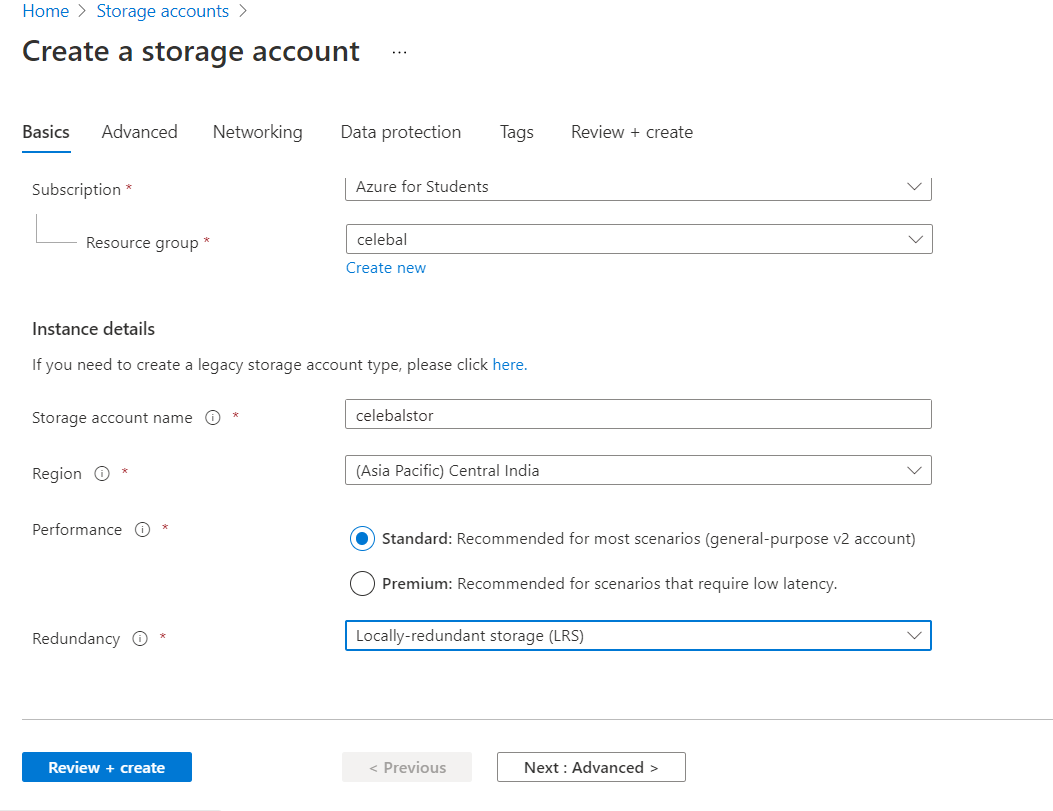
**Steps:**

#### **Create a storage account**

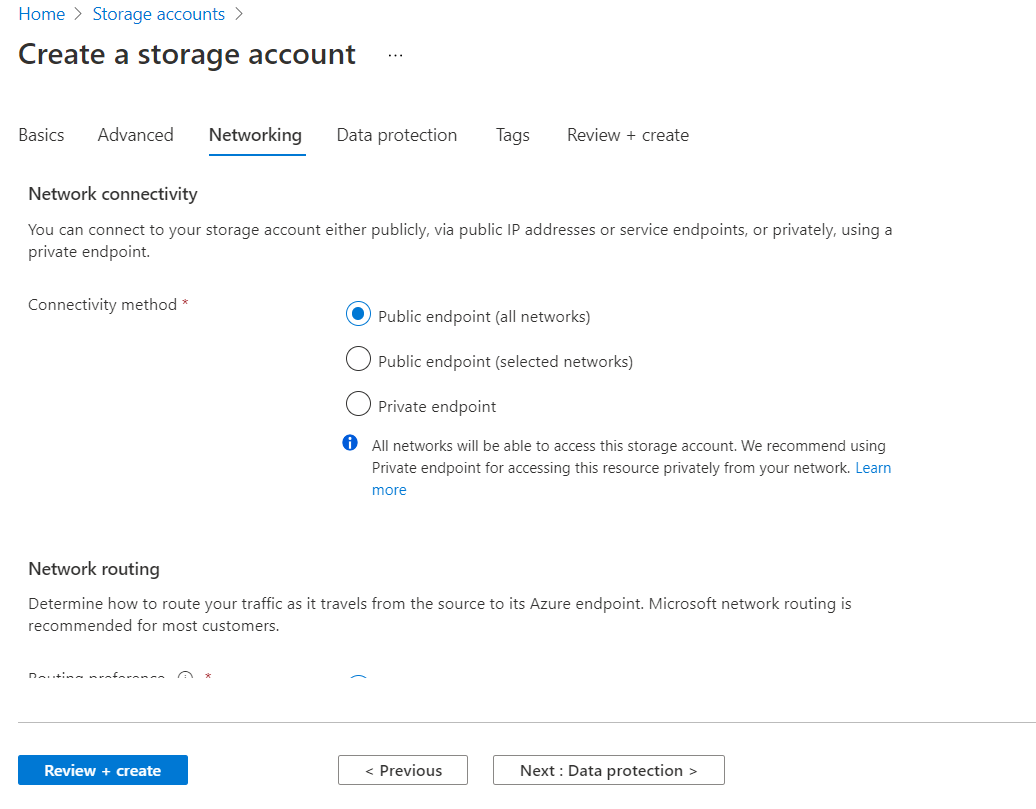
To create an Azure storage account with the Azure portal, follow these steps:

1. In the search bar, select **Storage accounts**.
2. On the **Storage accounts** page, select **Create**.

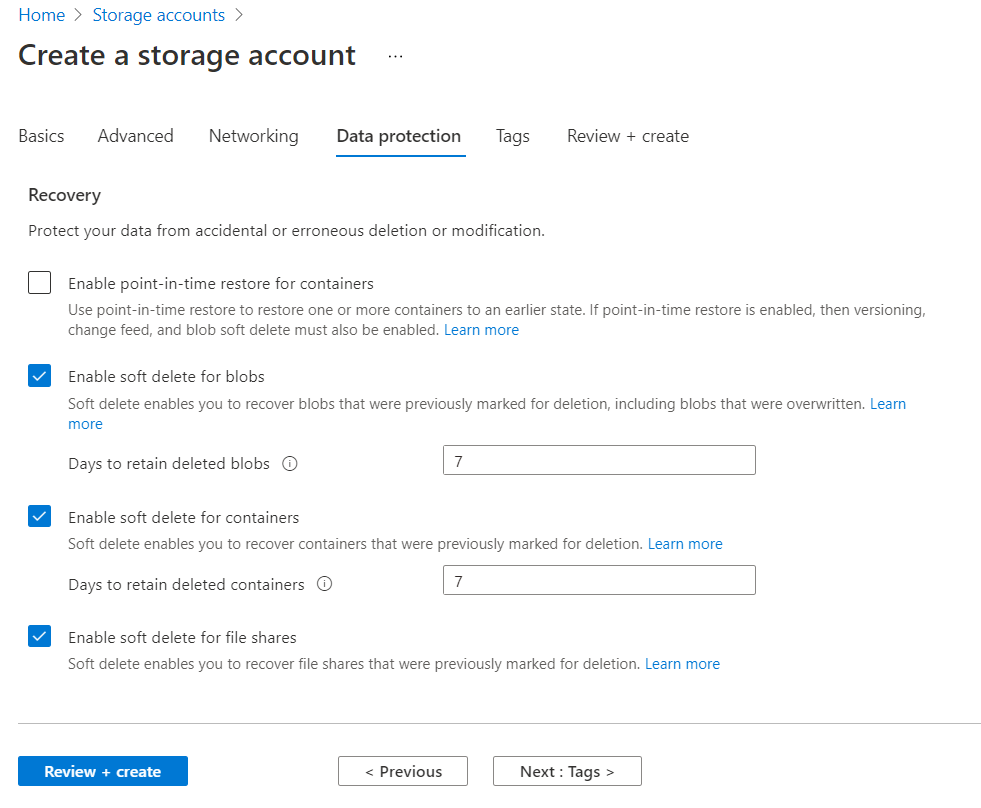
On the **Basics** tab, provide the essential information for your storage account. After you complete the **Basics** tab, you can choose to further customize your new storage account by setting options on the other tabs, or you can select **Review + create** to accept the default options and proceed to validate and create the account.



On the **Networking** tab, you can configure network connectivity and routing preference settings for your new storage account. These options can also be configured after the storage account is created.

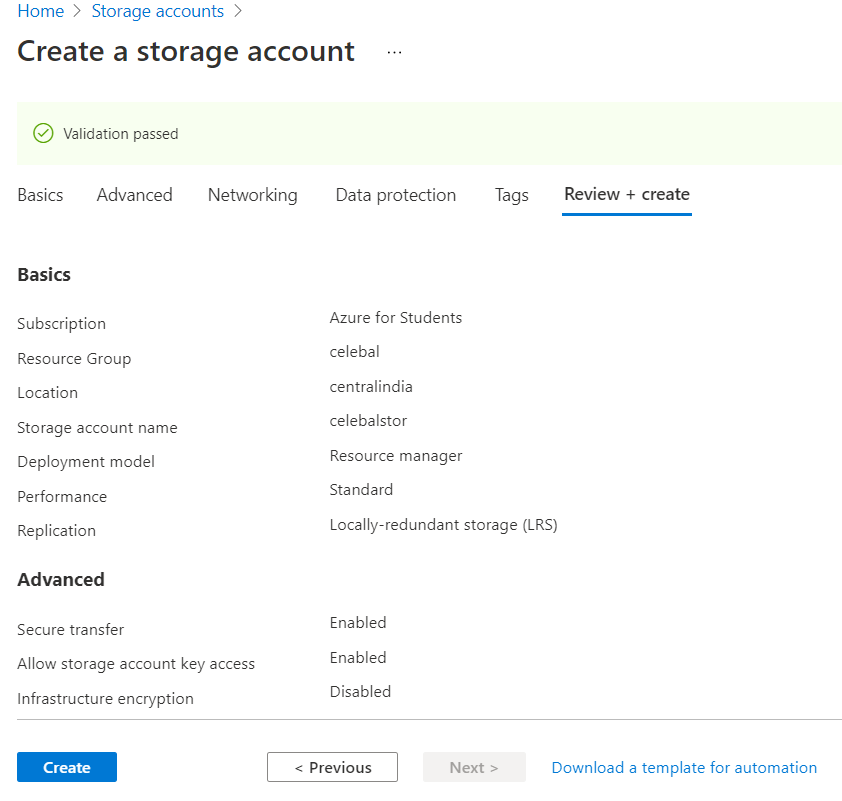


On the **Data protection** tab, you can configure data protection options for blob data in your new storage account. These options can also be configured after the storage account is created.



### Review + create tab

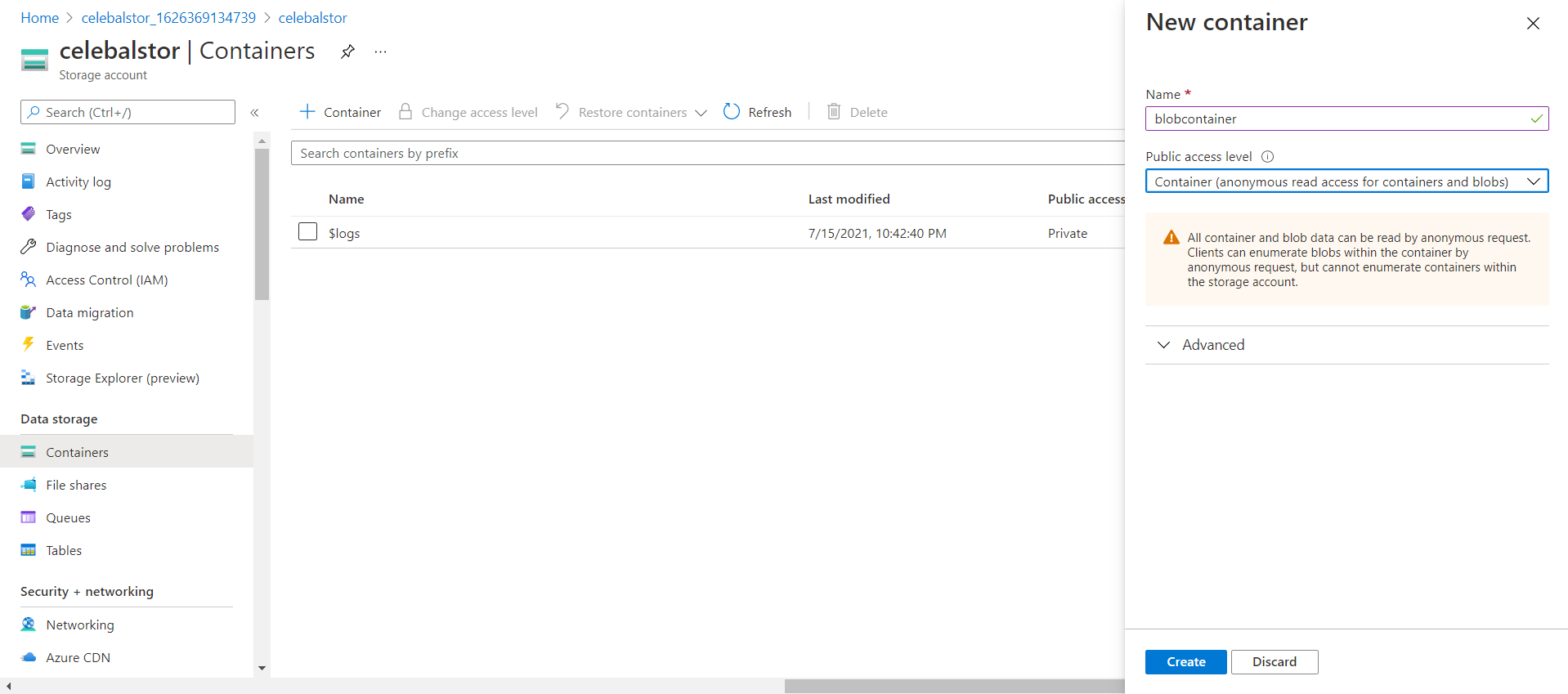
When you navigate to the **Review + create** tab, Azure runs validation on the storage account settings that you have chosen. If validation passes, you can proceed to create the storage account.



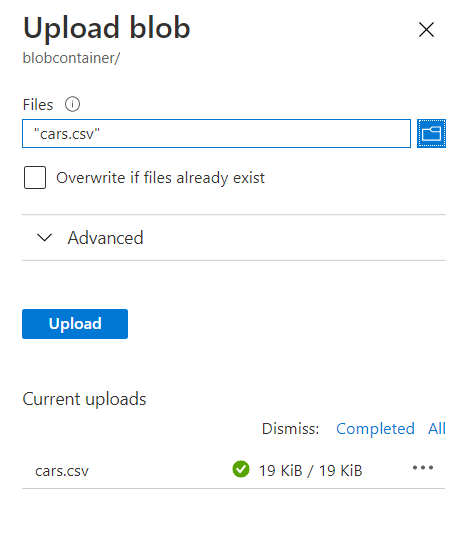
#### **Create a Blob Storage**

Create a blob container named **blobcontainer** in Azure Blob storage.

1. From the storage account page, select **Overview** > **Containers**.
2. In the **New container** dialog box, enter **blobcontainer** for the name, give access level as **Container** and then select **OK**. The **Containers** page is updated to include **blobcontainer** in the list of containers.
3. Select your container from the list.



1. On the **blobcontainer** container page's toolbar, select **Upload**.
2. In the **Files** box, browse and select the **cars.csv** file.



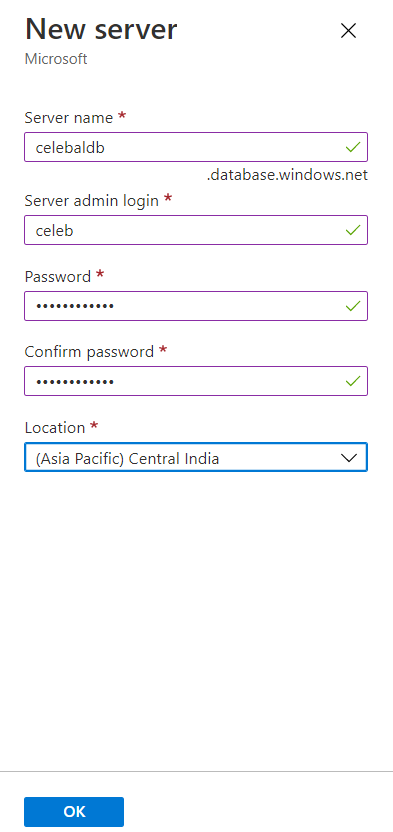
1. Select the **Upload** button. You should see the **cars.csv** file and the status of the upload in the list.

#### **Create a SQL Database**

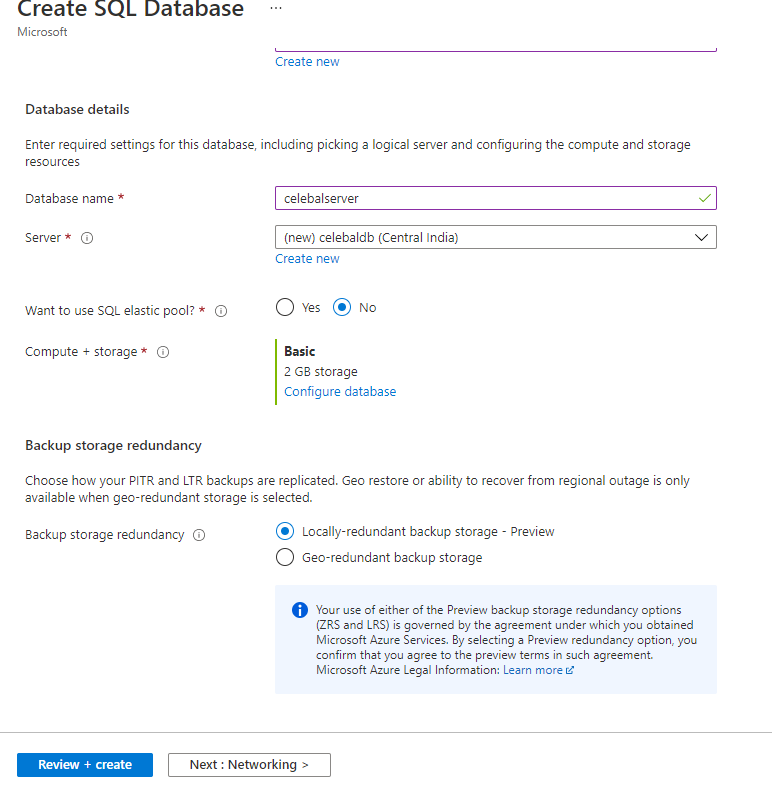
To create an Azure SQL Database with the Azure portal, follow these steps:

1. In the search bar, select **SQL Database**.
2. On the **SQL Database** page, select **Create**.
3. On the **Basics** tab of the **Create SQL Database** form, under **Project details**, select the desired Azure **Subscription**.
4. Select resource group *celebal.*
5. For **Database name** enter *celebaldb*.
6. For **Server**, select **Create new**, and fill out the **New server** form with the following values:
   1. **Server name**: Enter *celebaldb*, and add some characters for uniqueness. We can't provide an exact server name to use because server names must be globally unique for all servers in Azure, not just unique within a subscription.
   2. **Server admin login**: Enter *celebuser*.
   3. **Password**: Enter a password that meets requirements, and enter it again in the **Confirm password** field.
   4. **Location**: Select a location from the dropdown list.

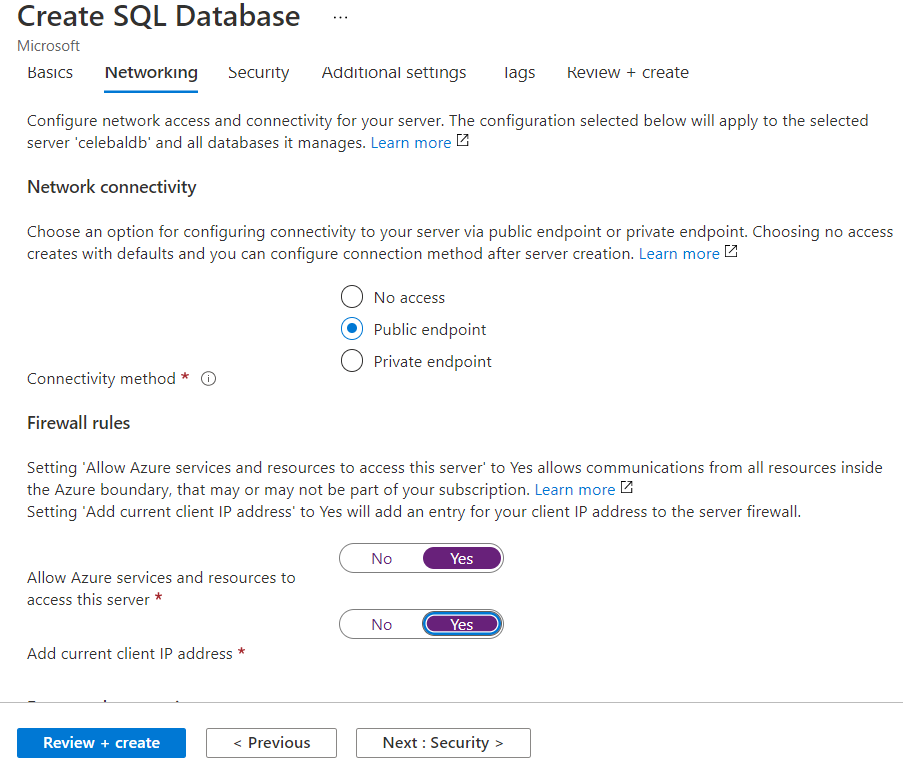
Select **OK**.



1. Leave **Want to use SQL elastic pool** set to **No**.
2. Under **Compute + storage**, select **Configure database**.
3. select **Basic** service tier as for less workloads, and then select **Apply**.



1. On the **Networking** tab, for **Connectivity method**, select **Public endpoint**.
2. For **Firewall rules**, set **Add current client IP address** to **Yes**. And **Allow Azure services and resources to access this server** set to **YES**.

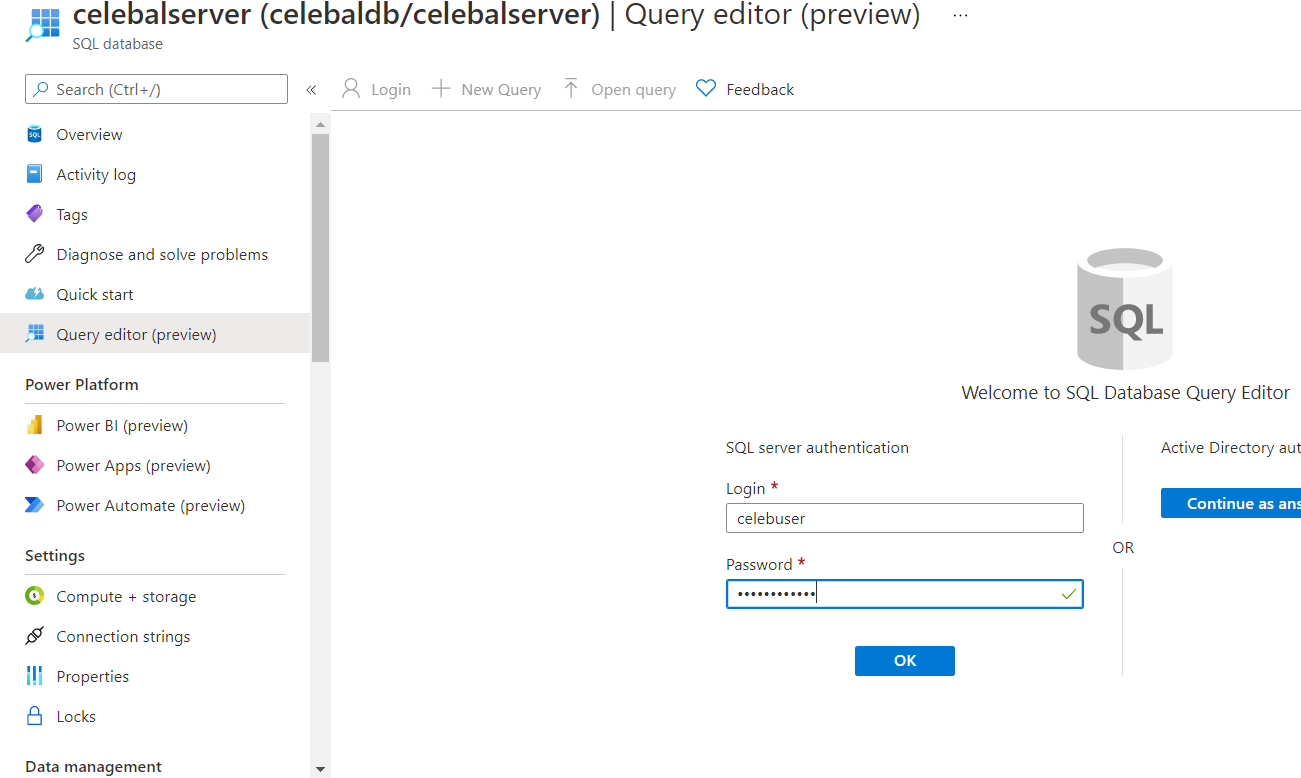


1. Select **Next: Additional settings** at the bottom of the page.
2. Select **Review + create** at the bottom of the page:

## Query the database

Once your database is created, you can use the **Query editor (preview)** in the Azure portal to connect to the database and query data.

1. In the portal, search for and select **SQL databases**, and then select your database from the list.
2. On the page for your database, select **Query editor (preview)** in the left menu.
3. Enter your server admin login information, and select **OK**.



1. Enter the following query:

CREATE TABLE Cars (

Make nvarchar(100),

Model nvarchar(200),

Type nvarchar(100),

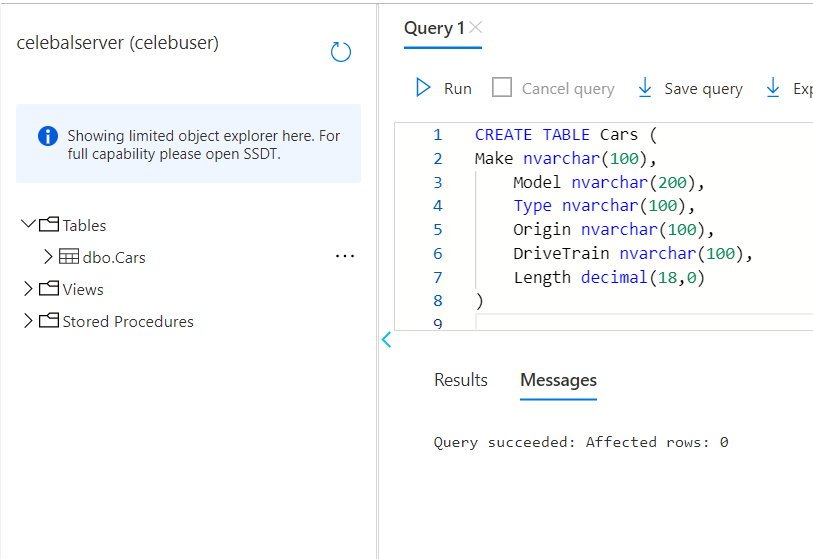
Origin nvarchar(100),

DriveTrain nvarchar(100),

Length decimal(18,0)

)

Select **Run**, and then review the query results in the **Results** pane.

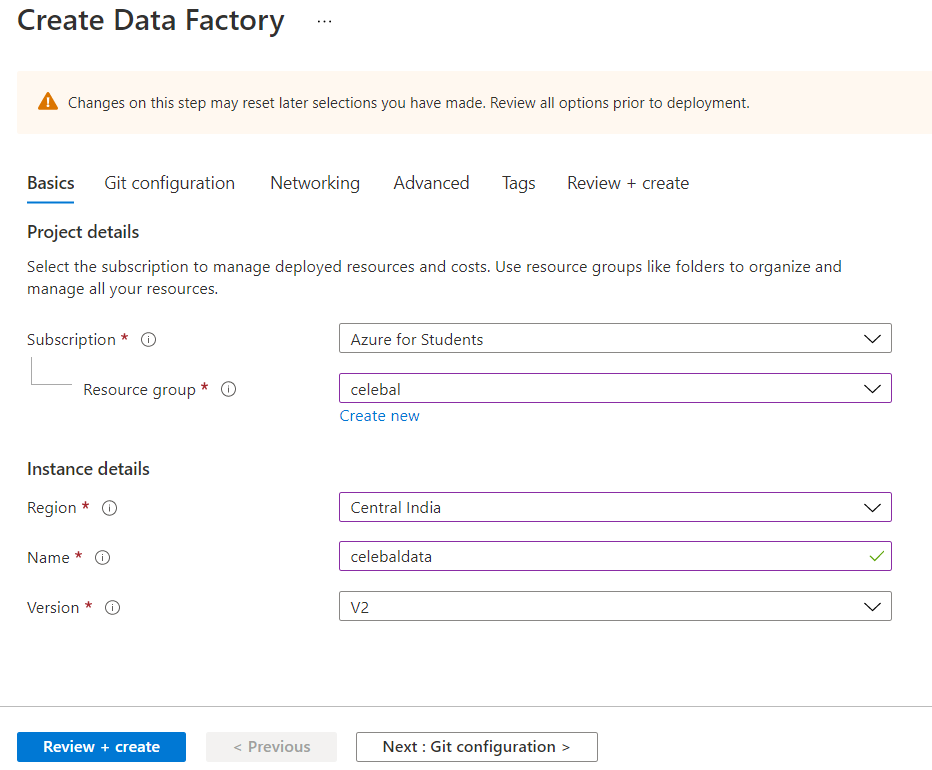


## Create a data factory

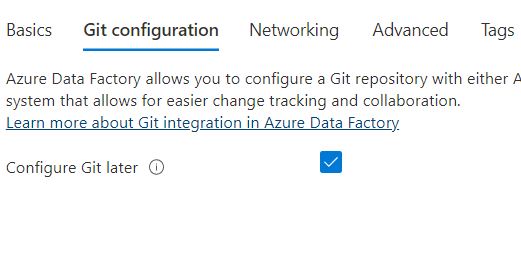
1. Launch **Microsoft Edge** or **Google Chrome** web browser. Currently, Data Factory UI is supported only in Microsoft Edge and Google Chrome web browsers.
2. Select Data Factories from search bar.
3. Select **Create Data Factory.**
4. On the **Create Data Factory** page, under **Basics** tab, select your Azure **Subscription** in which you want to create the data factory.
5. For **Resource Group**, select *celebal*.
6. For **Region**, select the location for the data factory.

The list shows only locations that Data Factory supports, and where your Azure Data Factory meta data will be stored. The associated data stores (like Azure Storage and Azure SQL Database) and computes (like Azure HDInsight) that Data Factory uses can run in other regions.

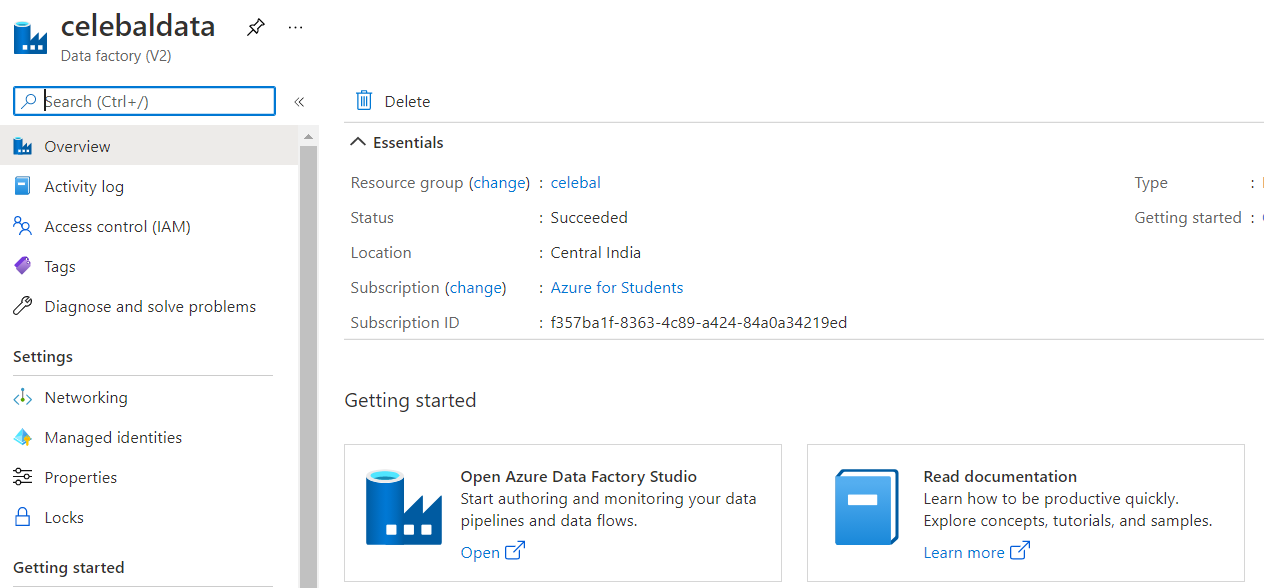
1. For **Name**, enter **celebaldata**. The name of the Azure data factory must be globally unique. If you see the following error, change the name of the data factory
2. For **Version**, select **V2**.



1. Select **Next: Git configuration**, and then select **Configure Git later** check box.



1. Select **Review + create**, and select **Create** after the validation is passed. After the creation is complete, select **Go to resource** to navigate to the **Data Factory** page.
2. Select **Open** on the **Open Azure Data Factory Studio** tile to start the Azure Data Factory user interface (UI) application on a separate browser tab.

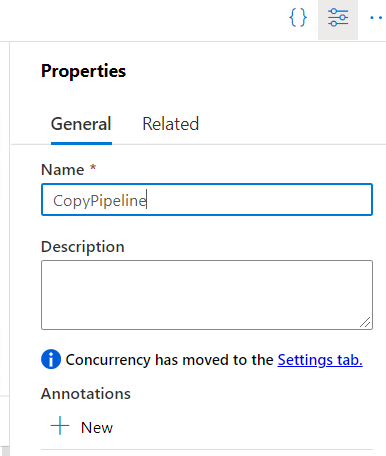


## Create a pipeline

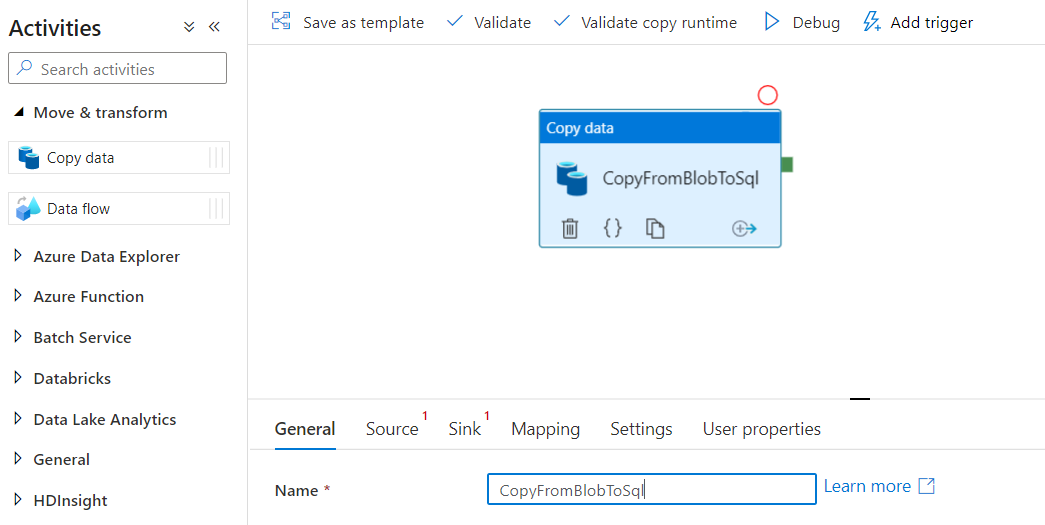
## On the home page, select ****Orchestrate****.

## 

* 1. In the General panel under **Properties**, specify **CopyPipeline** for **Name**. Then collapse the panel by clicking the Properties icon in the top-right corner.



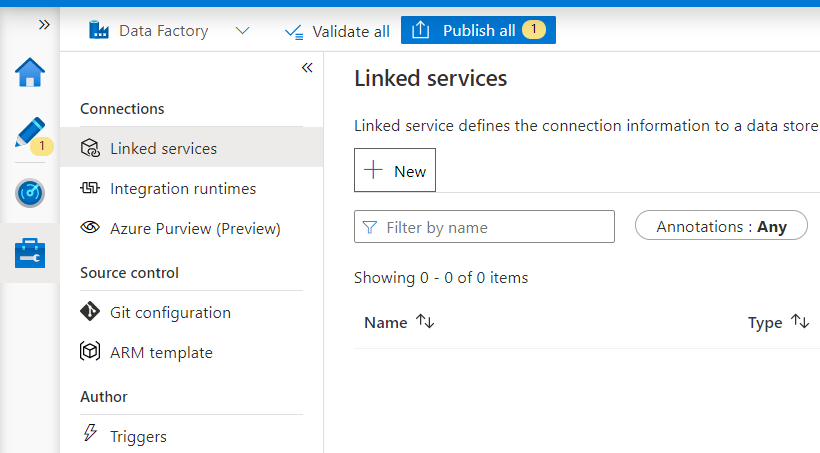
* 1. In the **Activities** tool box, expand the **Move and Transform** category, and drag and drop the **Copy Data** activity from the tool box to the pipeline designer surface. Specify **CopyFromBlobToSql** for **Name**.



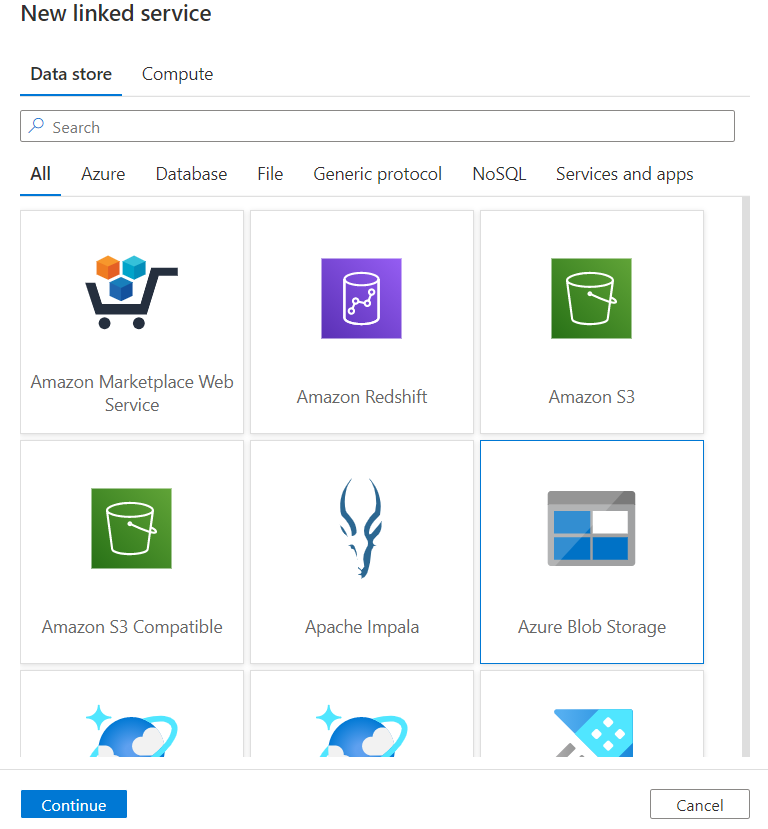
## Create a Linked service for Blob Storage

In this procedure, you create a linked service to link your Azure Storage account to the data factory. The linked service has the connection information that the Data Factory service uses at runtime to connect to it.

1. On the Azure Data Factory UI page, open **Manage** tab from the left pane.



1. On the Linked services page, select **+New** to create a new linked service
2. On the **New Linked Service** page, select **Azure Blob Storage**, and then select **Continue**.



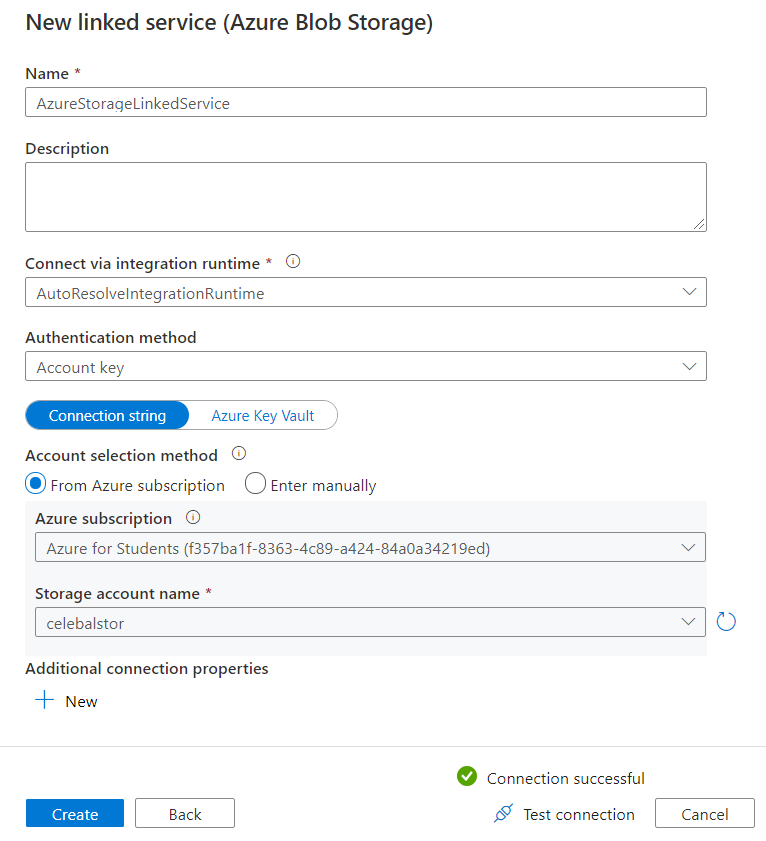
1. On the New Linked Service (Azure Blob Storage) page, complete the following steps:

a. For **Name**, enter **AzureStorageLinkedService**.

b. For **Storage account name**, select the name of your Azure Storage account.

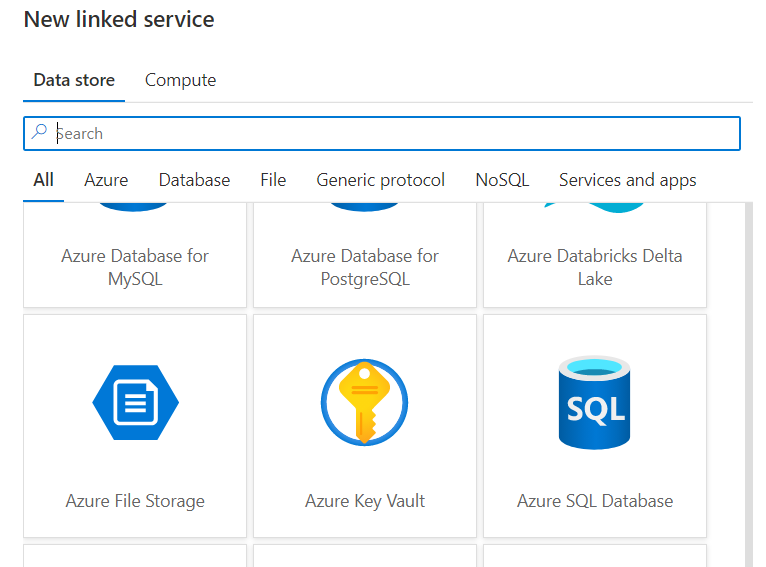
c. Select **Test connection** to confirm that the Data Factory service can connect to the storage account.

d. Select **Create** to save the linked service.



Repeat the steps to create New Linked service for Azure SQL Database:

1. On the Azure Data Factory UI page, open **Manage** tab from the left pane.
2. On the Linked services page, select **+New** to create a new linked service
3. On the **New Linked Service** page, select **Azure SQL Database**, and then select **Continue**.



1. On the New Linked Service (Azure SQL Database) page, complete the following steps:

a. For **Name**, enter **AzureSqlDatabaseLinkedService**.

b. For **Server name**, select **celebaldb**.

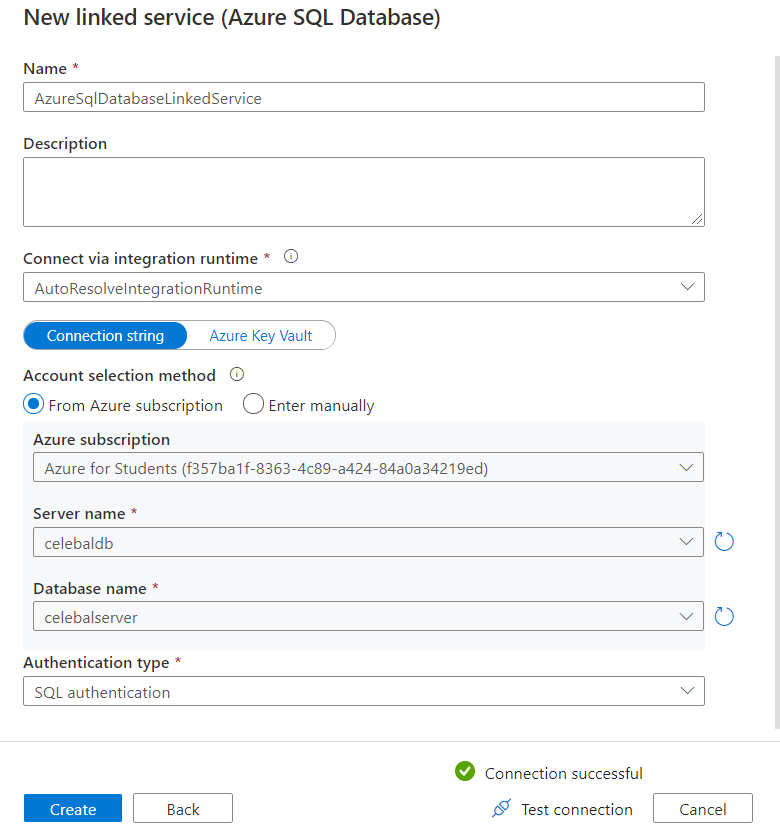
c. For **Server name**, select **celebalserver**.

d. For **Authentication** type, select **SQL** **authentication**.

e. Give Username and password.

f. Select **Test connection** to confirm that the Data Factory service can connect to the storage account.

g. Select **Create** to save the linked service.



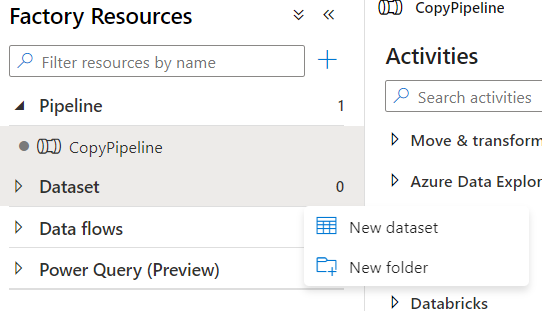
## Create Datasets

In this procedure, you create two datasets: **InputDataset** and **OutputDataset**.

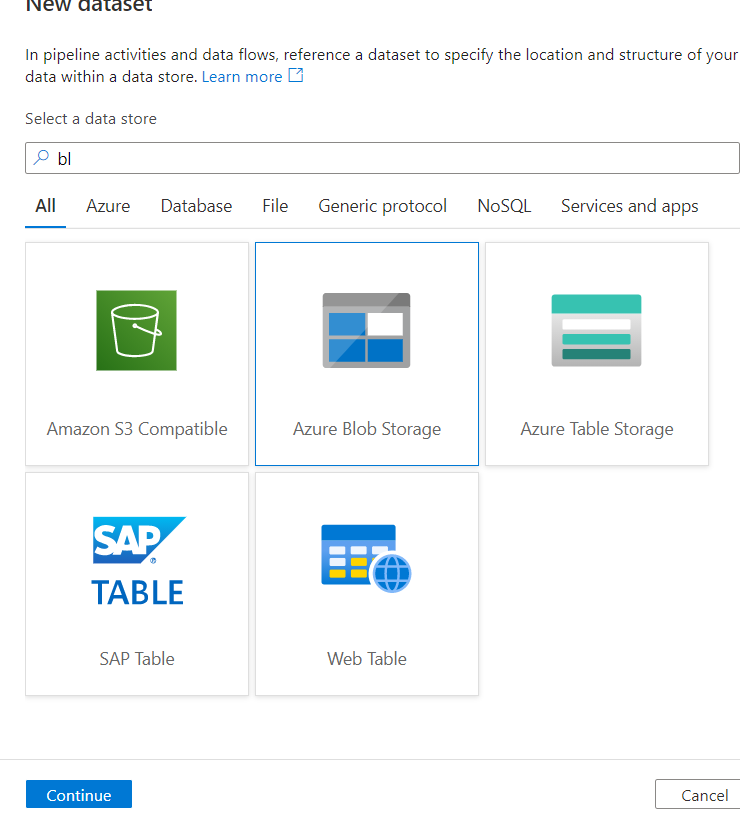
The input dataset represents the source data in the input folder. In the input dataset definition, you specify the blob container (**blobcontainer**), the folder (**input**), and the file (**cars.csv**) that contain the source data.

The output dataset represents the data that's copied to the destination.

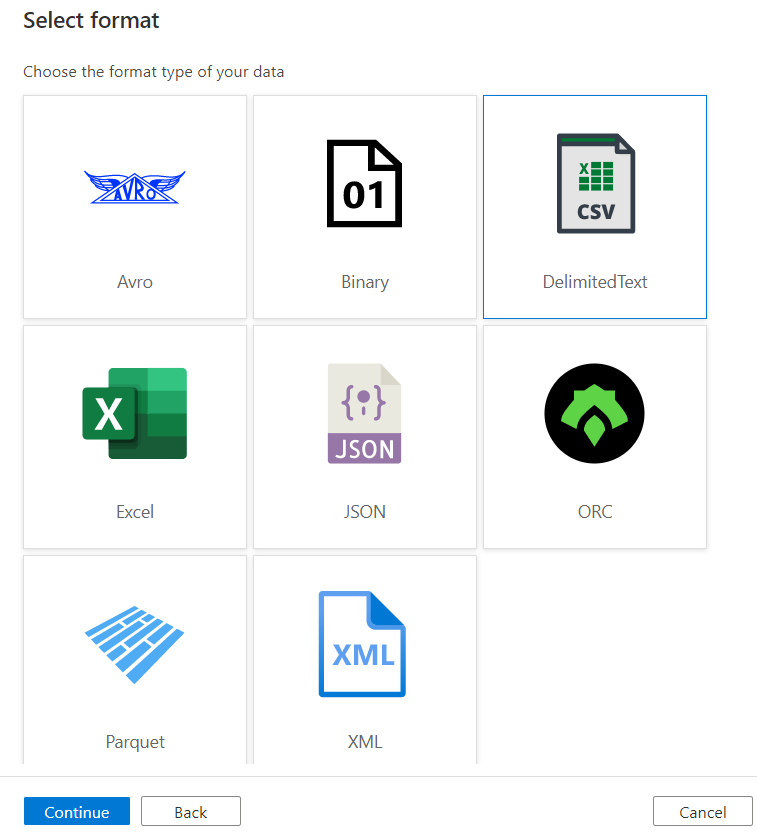
1. Select **Dataset** > **New** **Dataset** from the left pane on **Author** page.



1. On the **New Dataset** page, select **Azure Blob Storage**, and then select **Continue**.



1. On the **Select Format** page, choose the format type of your data, and then select **Continue**.



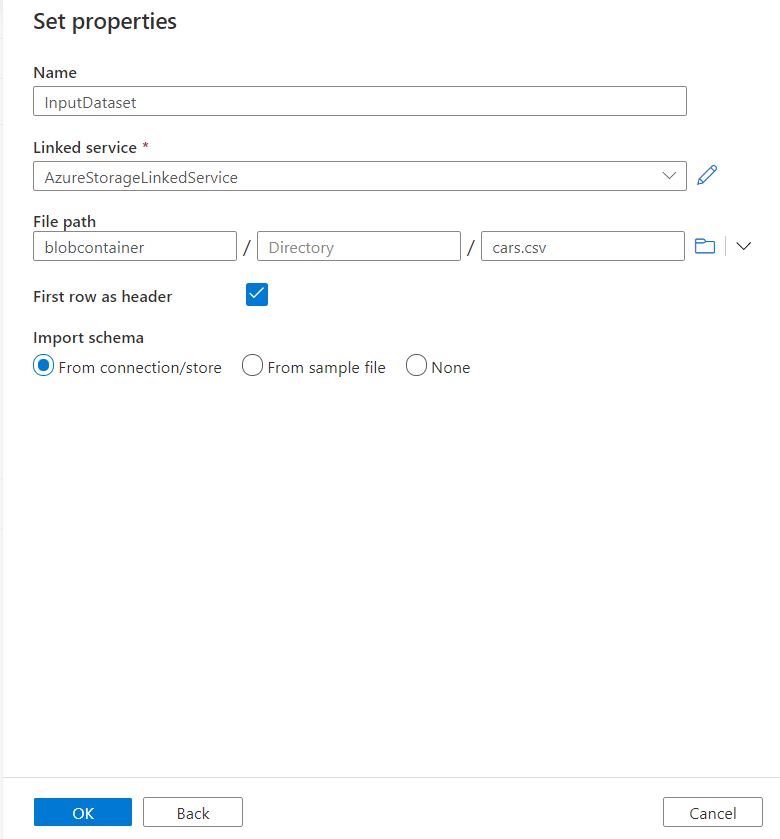
1. On the **Set Properties** page, complete following steps:

a. Under **Name**, enter **InputDataset**.

b. For **Linked service**, select **AzureStorageLinkedService**.

c. For **File path**, select the **Browse** button.

d. In the **Choose a file or folder** window, browse to the **input** folder in the **blobcontainer** container, select the **cars.csv** file, and then select **OK**.



1. Repeat the steps to create the output dataset:

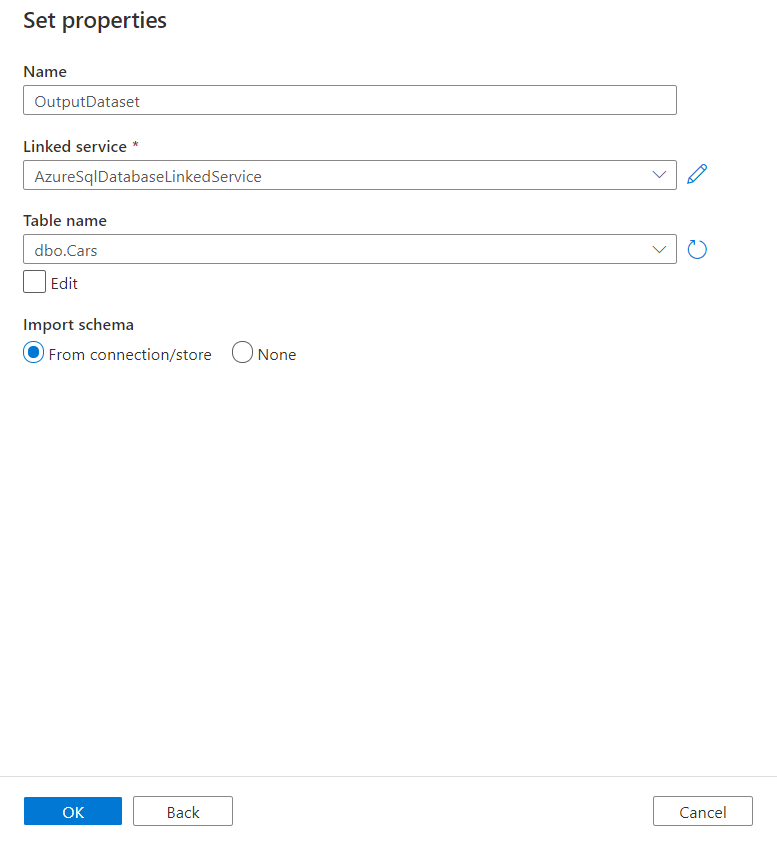
a. Select the **+** (plus) button, and then select **Dataset**.

b. On the **New Dataset** page, select **Azure SQL Database**, and then select **Continue**.

c. On the **Set Properties** page, specify **OutputDataset** for the name. Select **AzureSqlDatabaseLinkedService** as linked service.

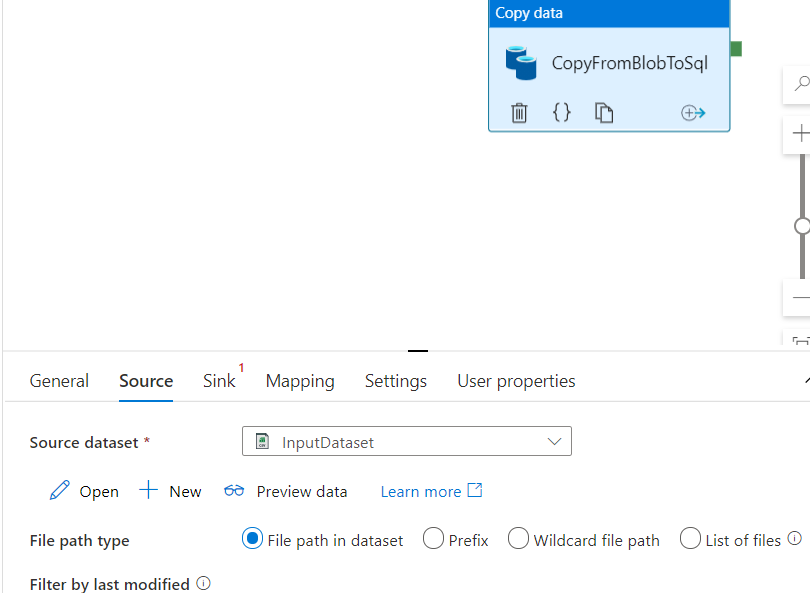
d. Under **Table name**, enter **dbo.Cars**. In the **import schema,** select **connection** **store**.

e. Select **OK**.

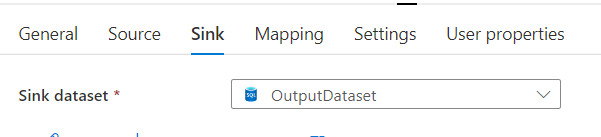


### Configure source

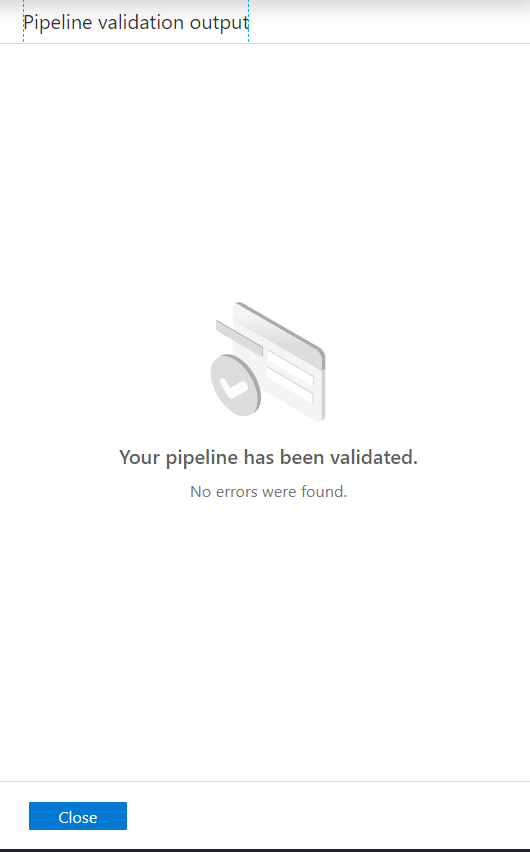
* 1. Select the Copy Data category
  2. Switch to the **Source** tab in the copy activity settings, and select **InputDataset** for **Source Dataset**.



* 1. Switch to the **Sink** tab in the copy activity settings, and select **OutputDataset** for **Sink Dataset**. Also, write a Pre-copy script



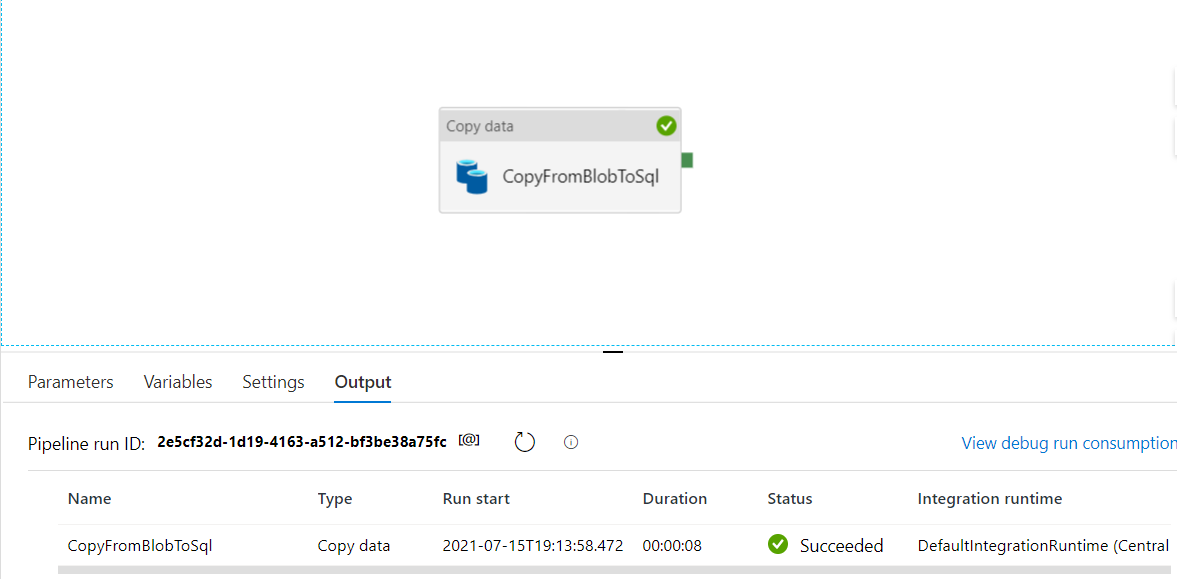
* 1. Click **Validate** on the pipeline toolbar above the canvas to validate the pipeline settings. Confirm that the pipeline has been successfully validated. To close the validation output, select the Validation button in the top-right corner.



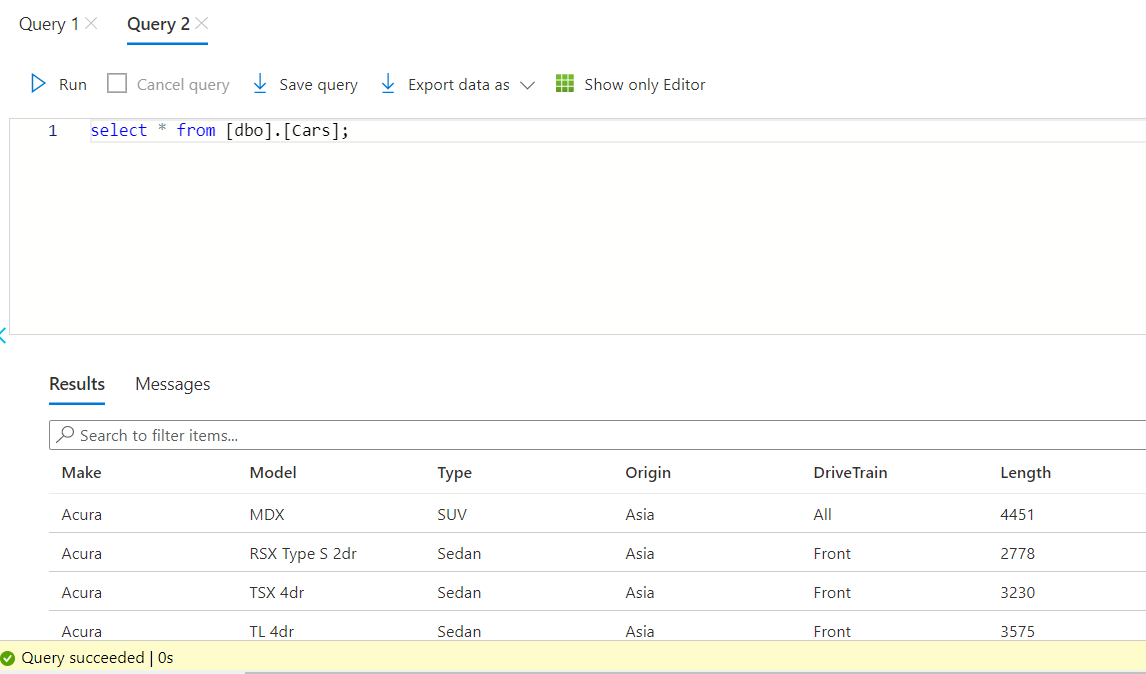
## Debug the pipeline

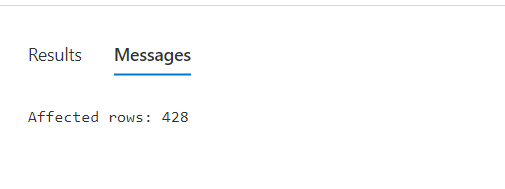
In this step, you debug the pipeline before deploying it to Data Factory.

1. On the pipeline toolbar above the canvas, click **Debug** to trigger a test run.
2. Confirm that you see the status of the pipeline run on the **Output** tab of the pipeline settings at the bottom.



1. In SQL database, run “**select**” query in query editor to confirm that data has been copied.





**NOTE**: Specify Pre-copy script “***delete from cars***” for Copy Activity to execute data from writing into sink table in each run. Used to clean pre-loaded data.

**CONCLUSION:**

Configured Azure Data Factory and created pipelines to take data from Azure Blob and inserted in into Azure SQL**.**