

## Loading 10 GB into Azure SQL Data Warehouse with Data Factory

### Introduction:

Fabrikam has successfully used Azure Data Factory and created a copy pipeline to copy their data from Azure SQL Database to Azure SQL Data Warehouse. Now, they are looking for another scenario where they want to load TBs of data that they have in their on-premises servers in table format. They will be able to move the data to Azure storage accounts using AZCopy. After moving the data to the storage account, the team is expecting to load the data at a rate of 1Gb/min. They are expecting to achieve this target by using the Azure Data Factory by integrating it with a large sized SQL Data Warehouse. They are now trying to perform this migration by using some datasets of their own that are of 10 GB size. To do this demo, they are using Azure SQL Datawarehouse and Azure Data Factory. On a successful migration of the data to Azure SQL Datawarehouse, they will be adopting to this service.

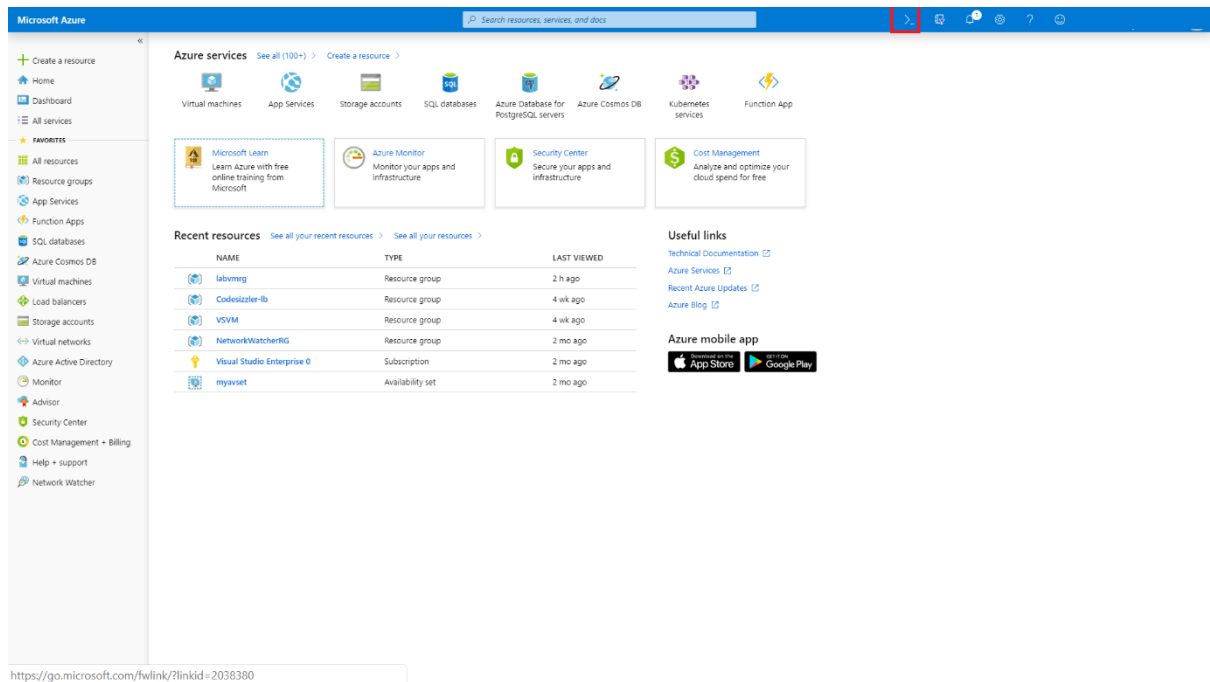
### Prerequisites:

To perform this demo user must have a valid Azure subscription and basic knowledge on

- Azure SQL Data Warehouse
- Data Factory
- Storage account

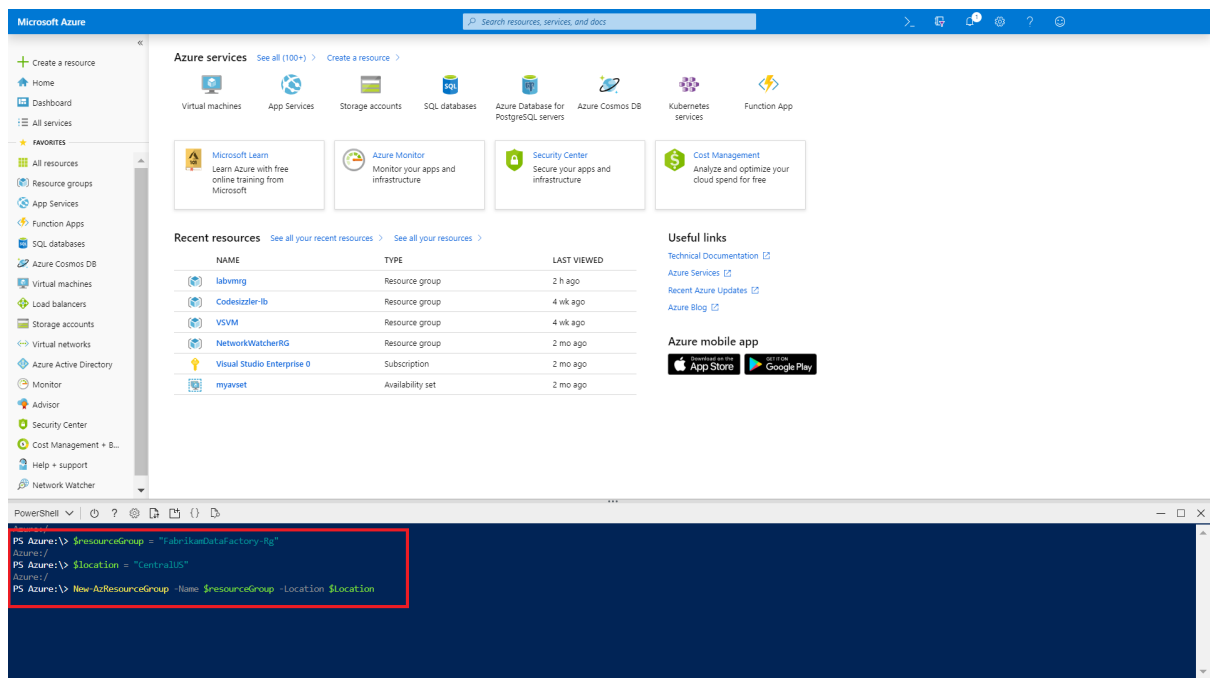
### Demo:

Log-in to Azure portal with your account using [www.portal.azure.com](http://www.portal.azure.com). In Azure portal start a PowerShell session to create a Storage account.



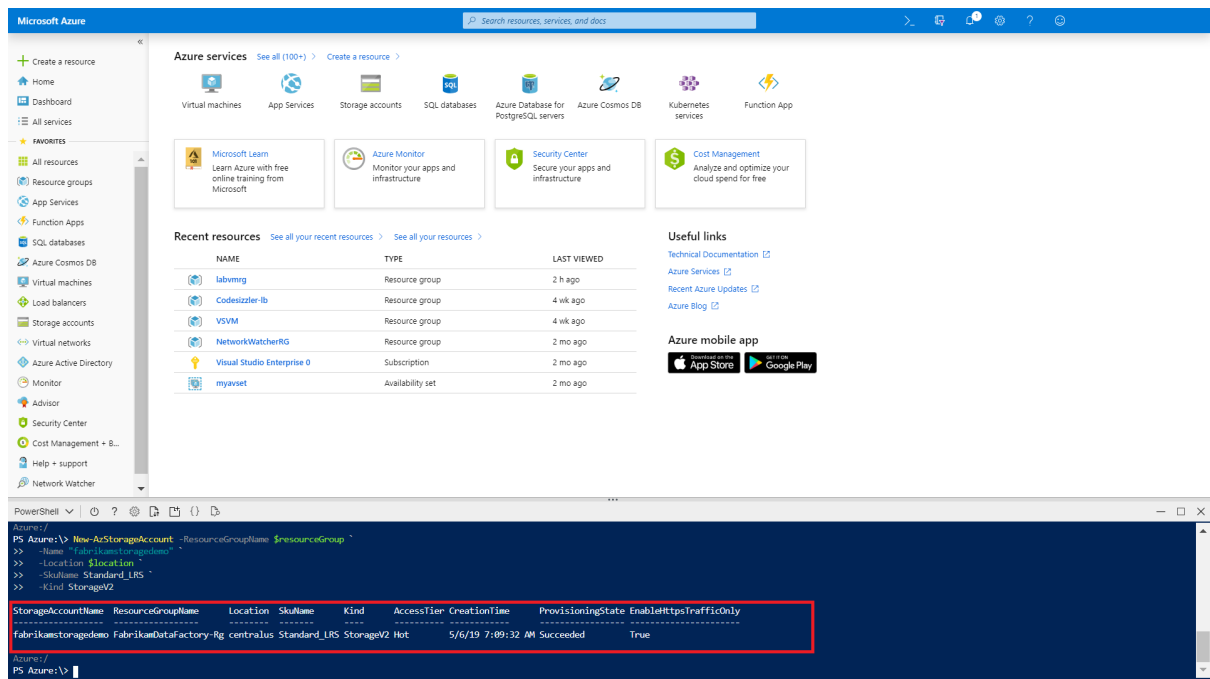
In the PowerShell session run the following command to create a resource group.

```
$resourceGroup = "fabrikam-rg"
$location = "CentralUS"
New-AzResourceGroup -Name $resourceGroup -Location $location
```



After creating the resource group run the following by replacing the **account-name** string with a unique name.

```
New-AzStorageAccount -
ResourceGroupName
$resourceGroup `
  -Name <account-name> `
  -Location $location `
  -SkuName Standard_RAGRS `
  -Kind StorageV2
```



NAME	TYPE	LAST VIEWED
labvmrg	Resource group	2 h ago
Codesizler-lb	Resource group	4 wk ago
VSVVM	Resource group	4 wk ago
NetworkWatcherRG	Resource group	2 mo ago
Visual Studio Enterprise 0	Subscription	2 mo ago
myavset	Availability set	2 mo ago

StorageAccountName	ResourceGroupName	Location	SkuName	Kind	AccessTier	CreationTime	ProvisioningState	EnableHttpsTrafficOnly
FabrikamStorageDemo	FabrikamDataFactory-Rg	centralus	Standard_LRS	StorageV2	Hot	5/6/19 7:00:32 AM	Succeeded	True

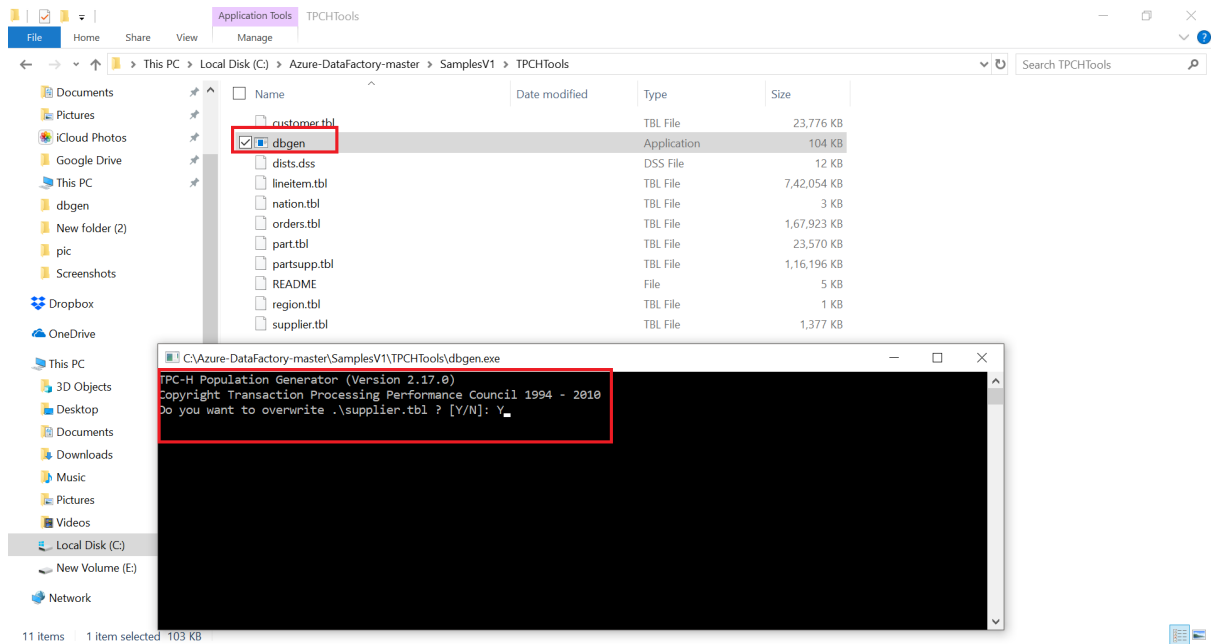
After creating the storage account navigate to [http://www.tpc.org/tpc\\_documents\\_current\\_versions/current\\_specifications.asp](http://www.tpc.org/tpc_documents_current_versions/current_specifications.asp) and download the **TCP-H** tool kit and compile it yourself. Once you compile the file run the **dbgen.exe** and the following command ten times by changing the number.

```
Dbgen -s 1000 -S **1** -C 10 -T L -v

Dbgen -s 1000 -S **2** -C 10 -T L -v

...

Dbgen -s 1000 -S **10** -C 10 -T L -v
```



Note the file sizes changes every time when you run the command. It will rewrite the table files when you run the command.

After generating the data files move the **lineitem.tbl** file to your blob storage account. Then start a PowerShell session and run the following command to create multiple data files. Replace the respective values with your values and change the number string to how many times that you need to multiple the data file.

#Declarations

\$resourceGroup = "fabrikam-rg"

\$location = "SouthIndia"

#Creating new RG

New-AzureRmResourceGroup -Name \$resourceGroup -Location \$location

#Creating storage account

\$storageAccount = New-AzureRmStorageAccount -ResourceGroupName  
\$resourceGroup -Name "fabrikamst03" -Location \$location -SkuName Standard\_LRS -  
Kind Storage

#Creating Context for storage account

\$ctx = \$storageAccount.Context

#creating Container

\$containerName = "fabrikamcontainer1"

New-AzureStorageContainer -Name \$containerName -Context \$ctx -Permission  
container

\$container = Get-AzureStorageContainer -Name fabrikamcontainer1 -Context \$ctx

For (\$i=0; \$i -le 10; \$i++) {

    #Copying blobs - Simple blob copy

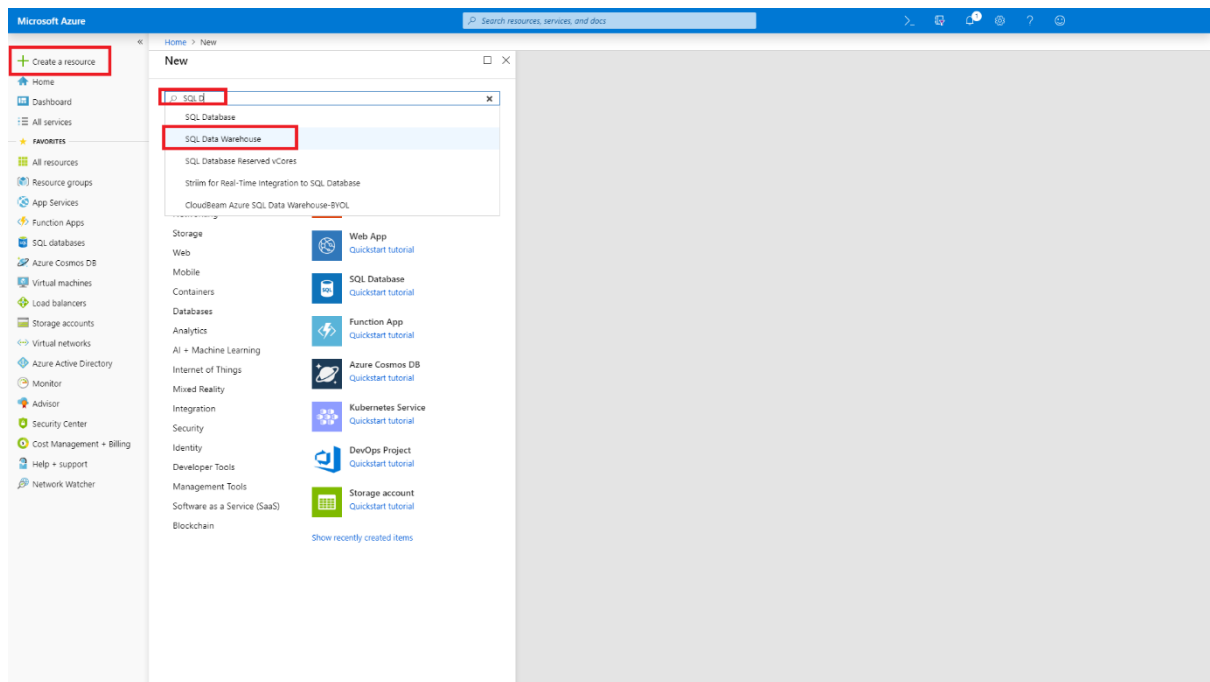
    \$blobName = "lineitem.tbl"

    \$newblobname = \$i

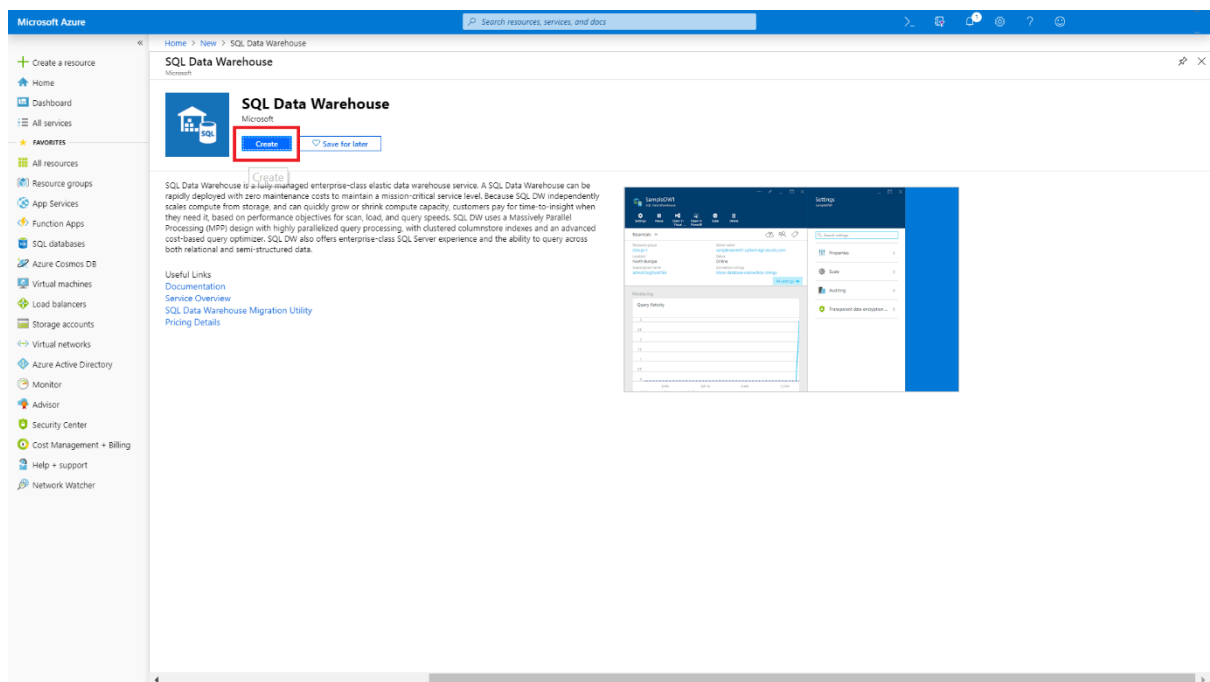
    Start-AzureStorageBlobCopy -SrcBlob \$blobName -SrcContainer fabrikamcontainer1  
    -DestContainer fabrikamcontainer1 -DestBlob \$newblobname -Context \$ctx -verbose

}

In Azure portal click on create a new resource and search for SQL Data Warehouse.



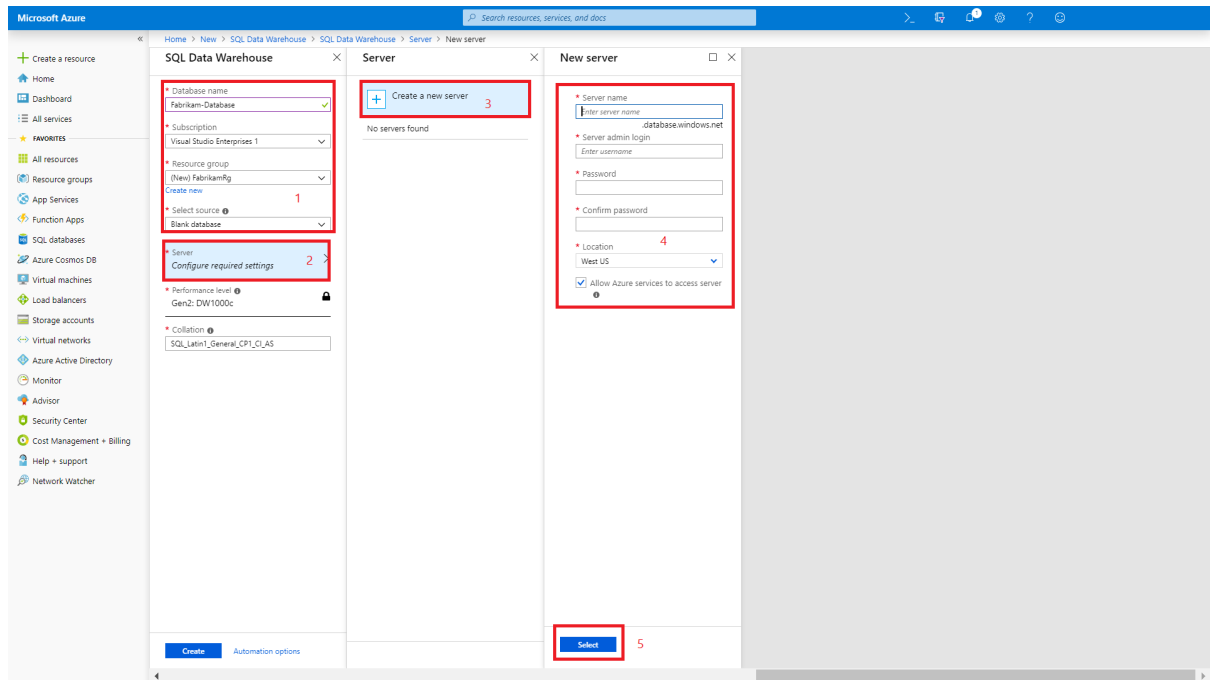
Use the search list to create a SQL Data Warehouse. Click on create.



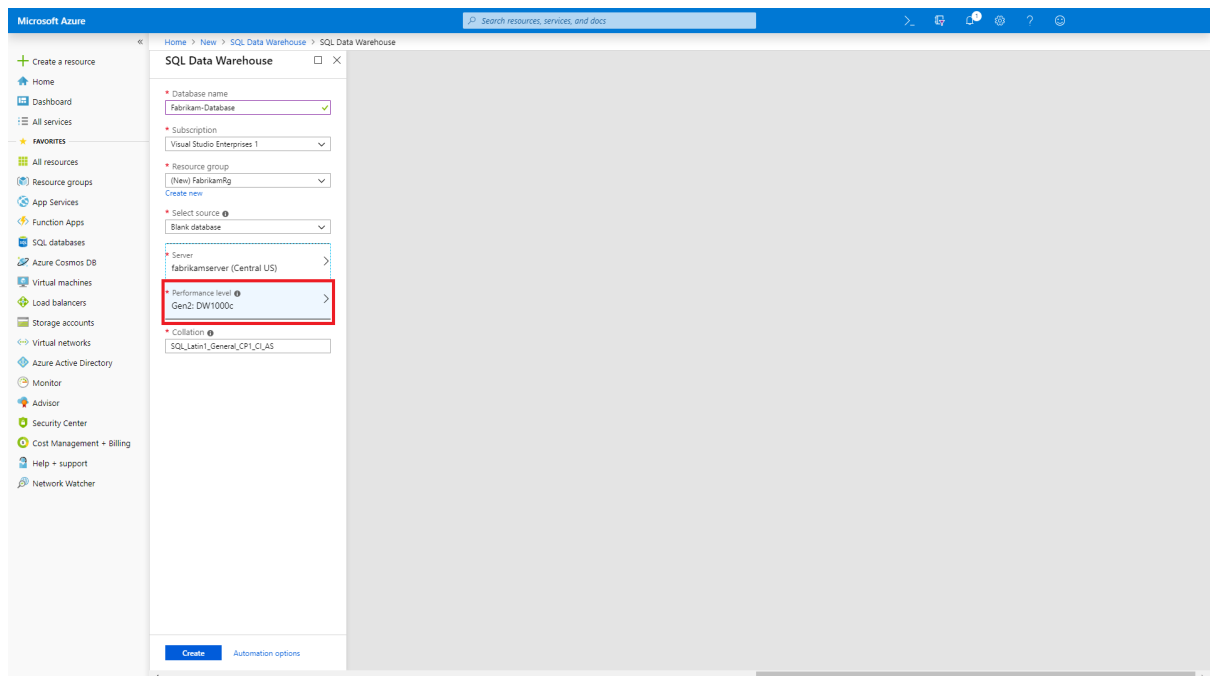
When it prompts configure the following settings and click on create.

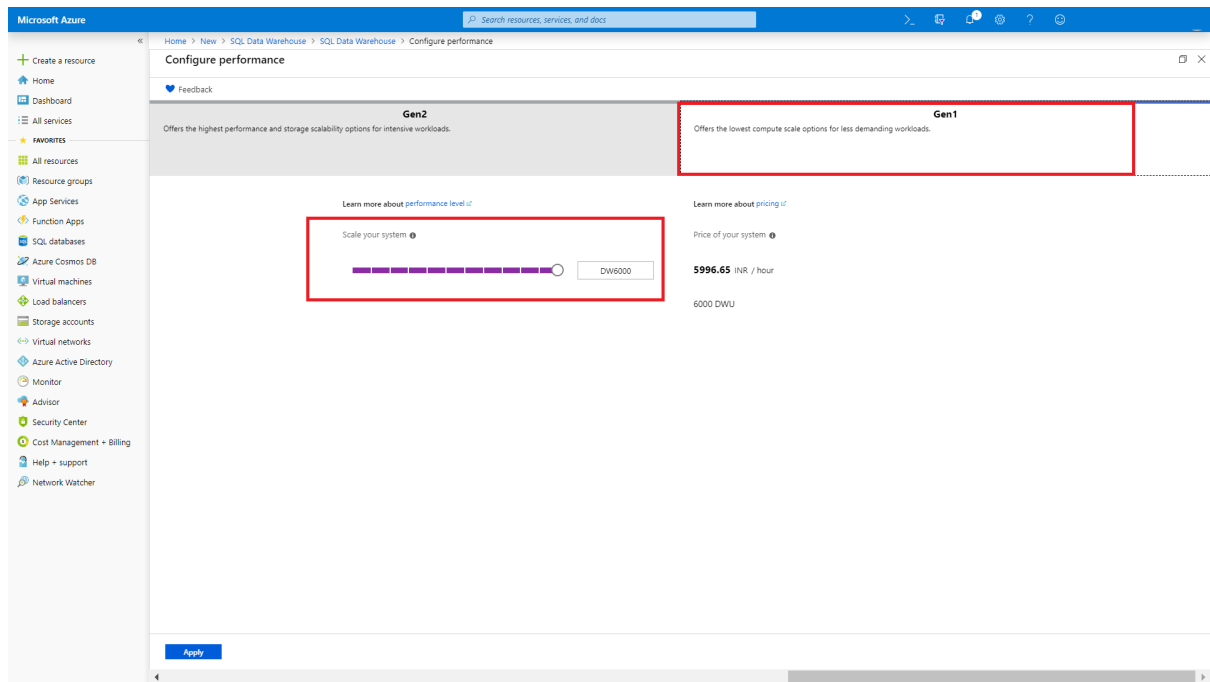
- Database name: Fabrikam-Database
- Subscription: Select a valid one
- Resource group: Create a new resource group **fabrikam-rg**
- Select Source: Blank Database
- Server: Configure a server with following configurations

- Server name: fabrikamserver
- Server admin log-in: FabrikamUser
- Password: Fabrikam@12345

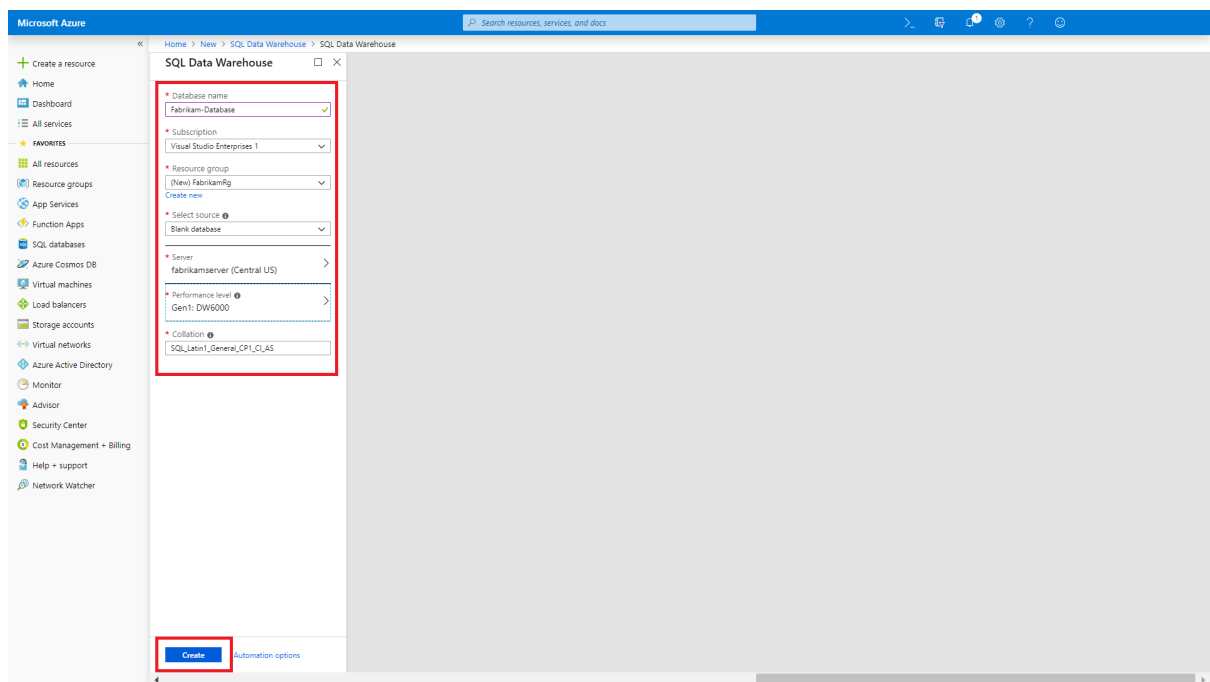


Once you configured the server settings the performance level panel will get unlocked. Configure the scale level to **DW6000**.





After configuring the scale level leave the collation to default and click on create.



After the deployment gets succeeded navigate to **SQL Server Management Studio** and log in with the respective credentials.

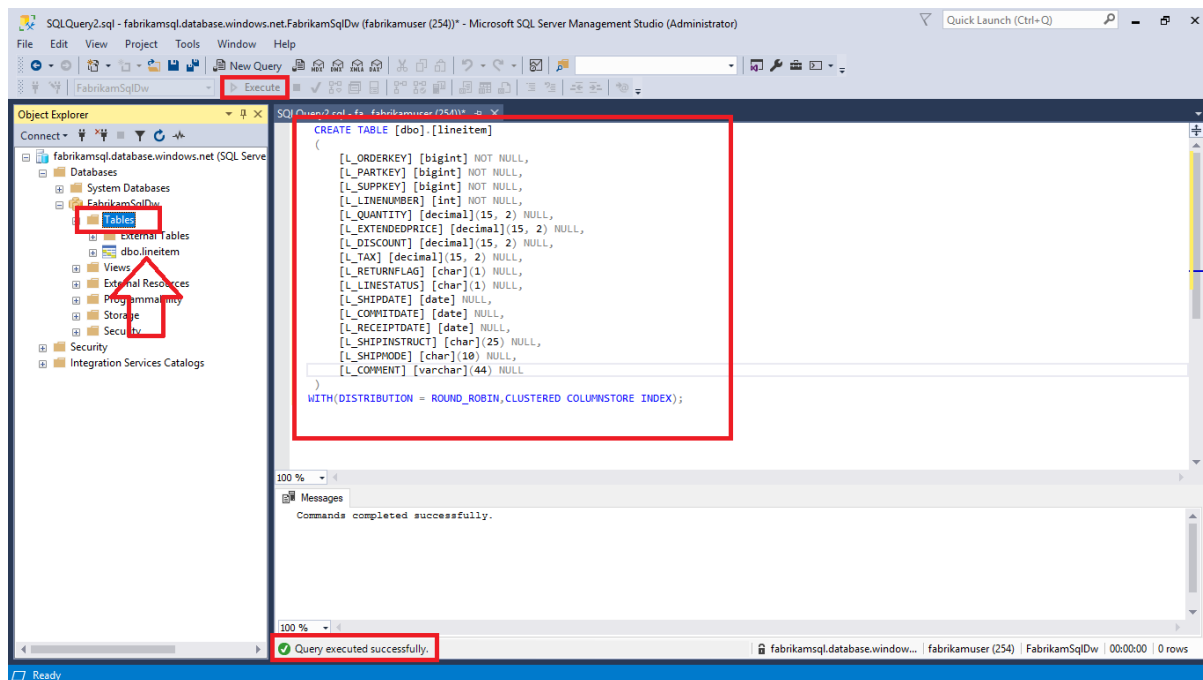
In the SQL Server Management Studio run the following script with selecting the table to create a new table.



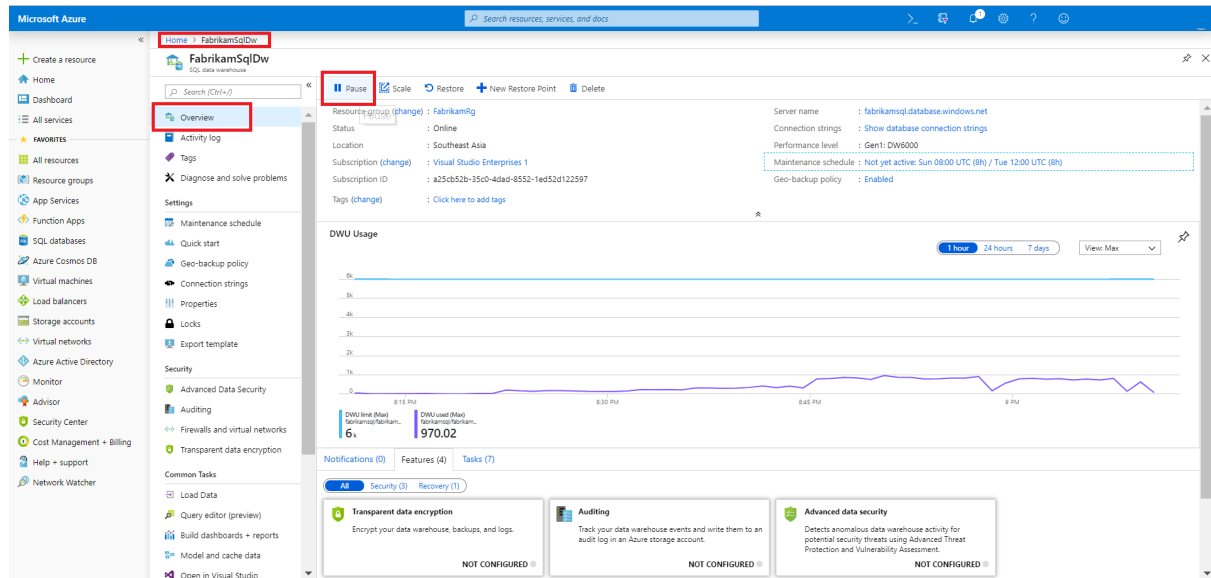
```

CREATE TABLE [dbo].[lineitem]
(
    [L_ORDERKEY] [bigint] NOT NULL,
    [L_PARTKEY] [bigint] NOT NULL,
    [L_SUPPKEY] [bigint] NOT NULL,
    [L_LINENUMBER] [int] NOT NULL,
    [L_QUANTITY] [decimal](15, 2) NULL,
    [L_EXTENDEDPRICE] [decimal](15, 2) NULL,
    [L_DISCOUNT] [decimal](15, 2) NULL,
    [L_TAX] [decimal](15, 2) NULL,
    [L_RETURNFLAG] [char](1) NULL,
    [L_LINESTATUS] [char](1) NULL,
    [L_SHIPDATE] [date] NULL,
    [L_COMMITDATE] [date] NULL,
    [L_RECEIPTDATE] [date] NULL,
    [L_SHIPINSTRUCT] [char](25) NULL,
    [L_SHIPMODE] [char](10) NULL,
    [L_COMMENT] [varchar](44) NULL
)
WITH
(
    DISTRIBUTION = ROUND_ROBIN,
    CLUSTERED COLUMNSTORE INDEX
)

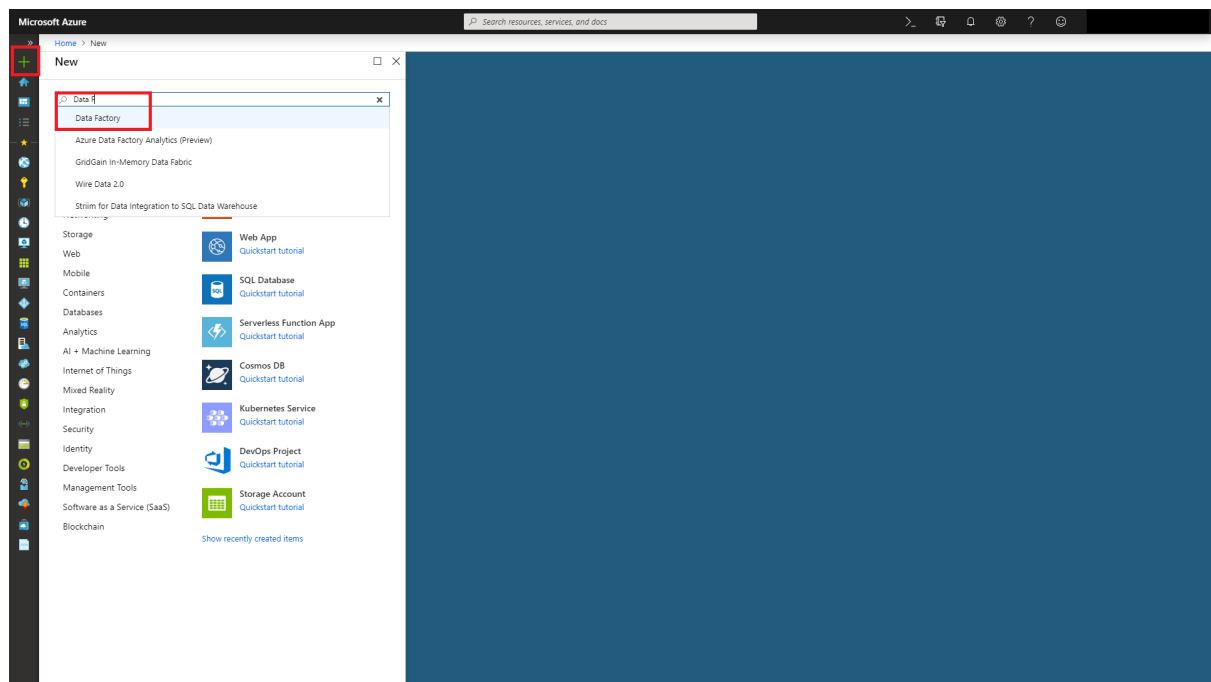
```



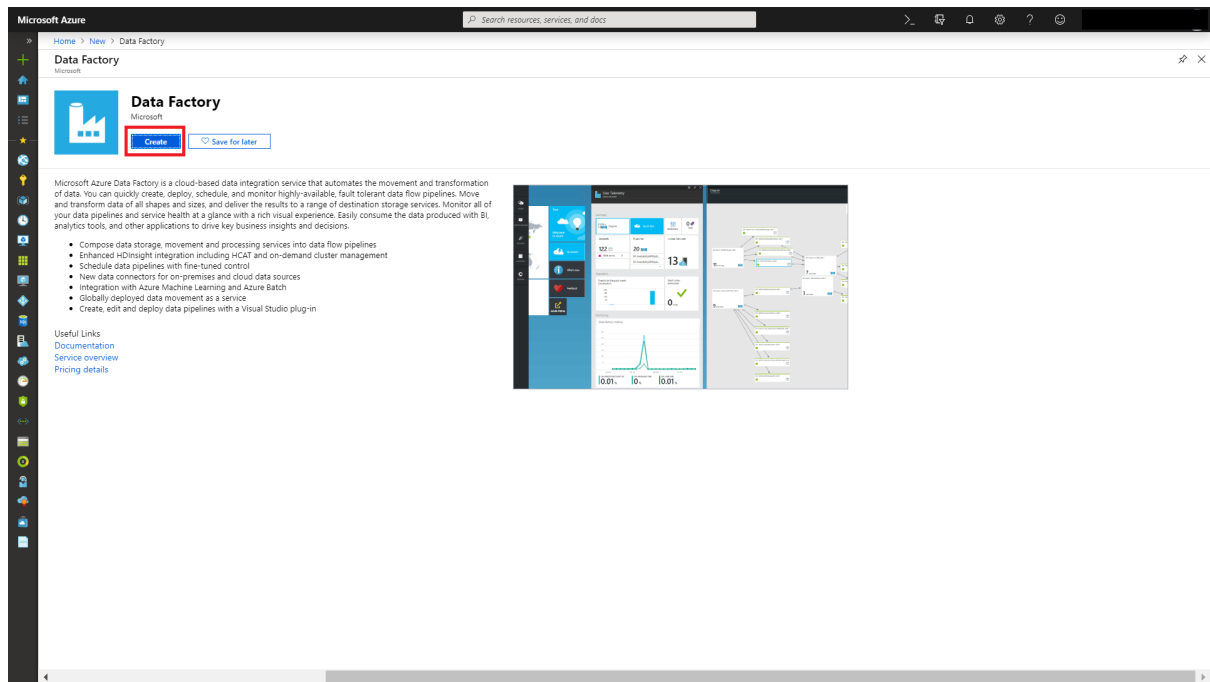
After creating the table pause you SQL Data Warehouse.



In Azure portal click on create a new resource and search for **Data Factory**.

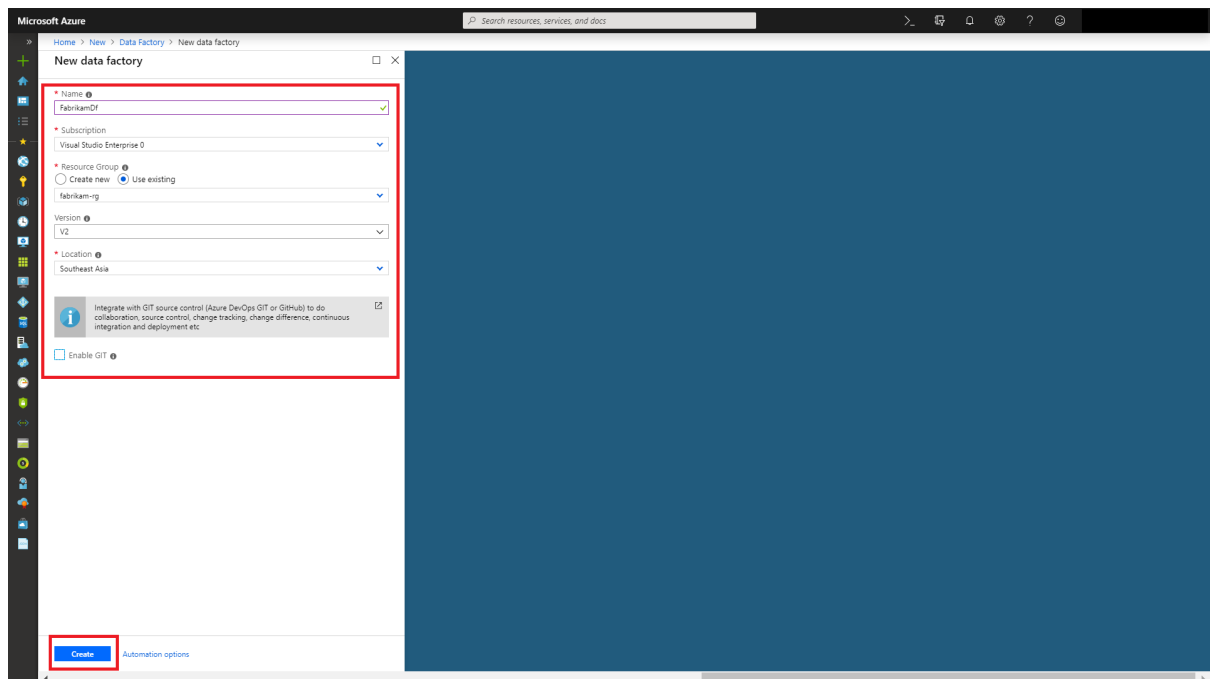


Use the search list to create a Data Factory.

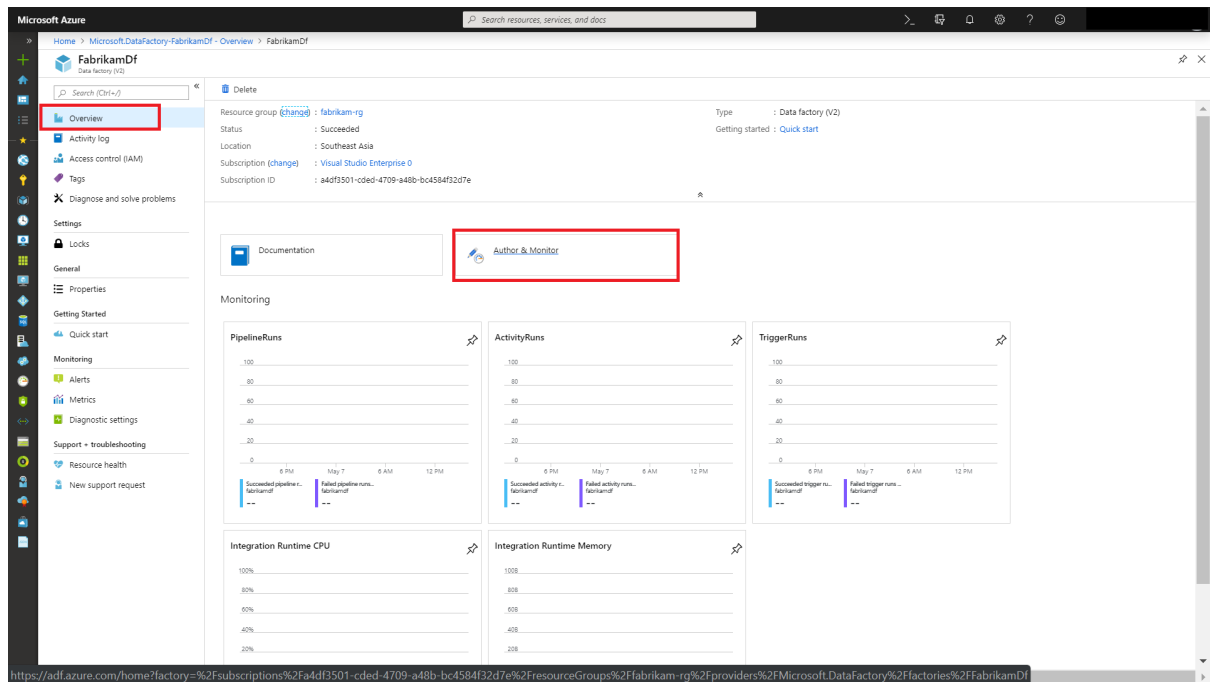


In the create Data Factory panel configure the following settings and click on create.

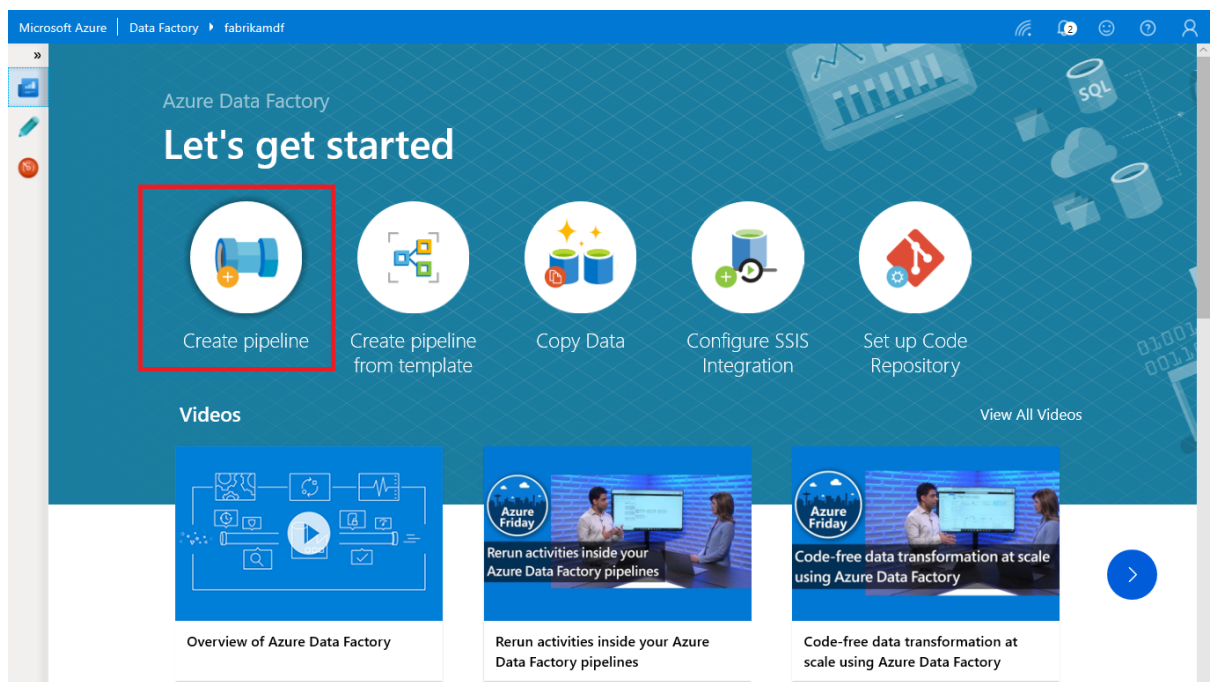
- Name: FabrikamDf
- Subscription: Select a valid one
- Resource group: Select the fabrikam-rg
- Version: V2
- Location: Select a valid location



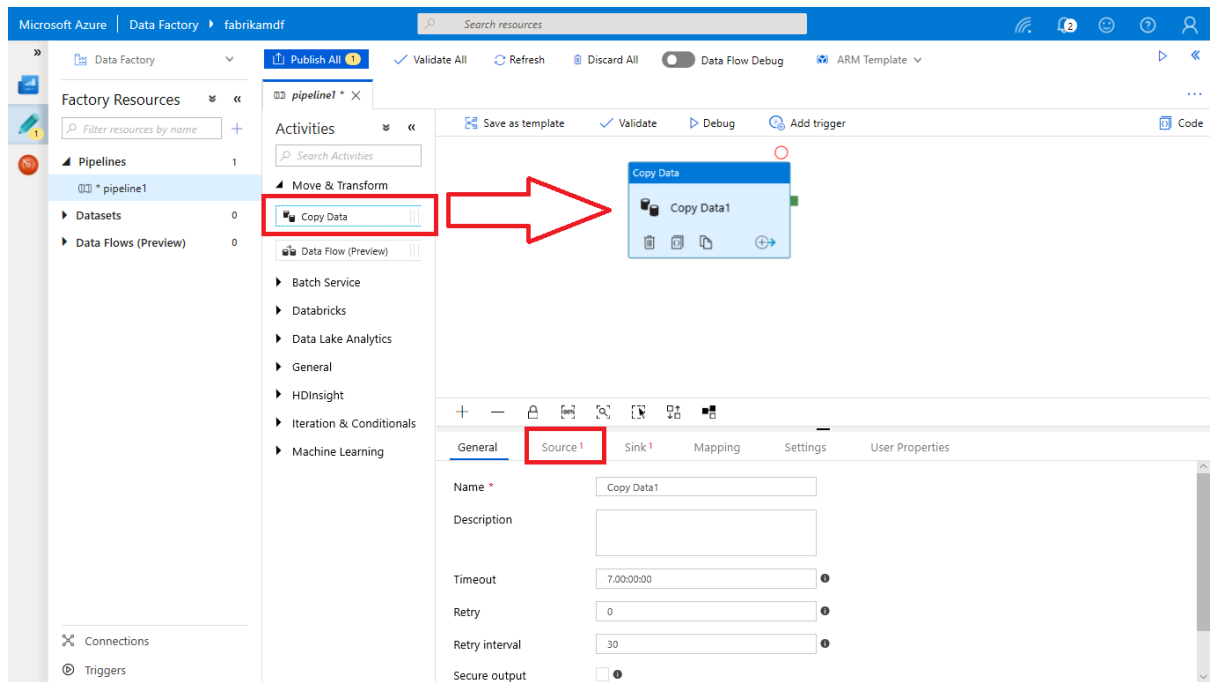
When the deployment got succeed navigate to the deployed resource group and in the overview panel, click on **Author & Monitor**.



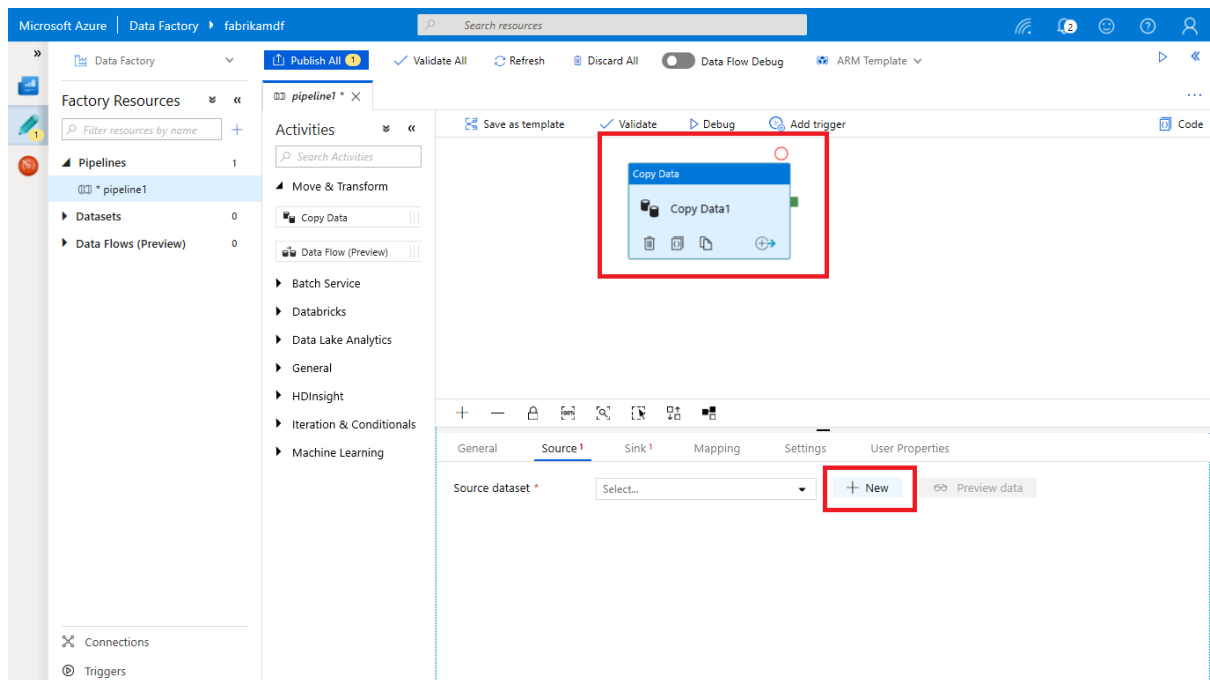
When it prompts click on **Create Pipeline**.



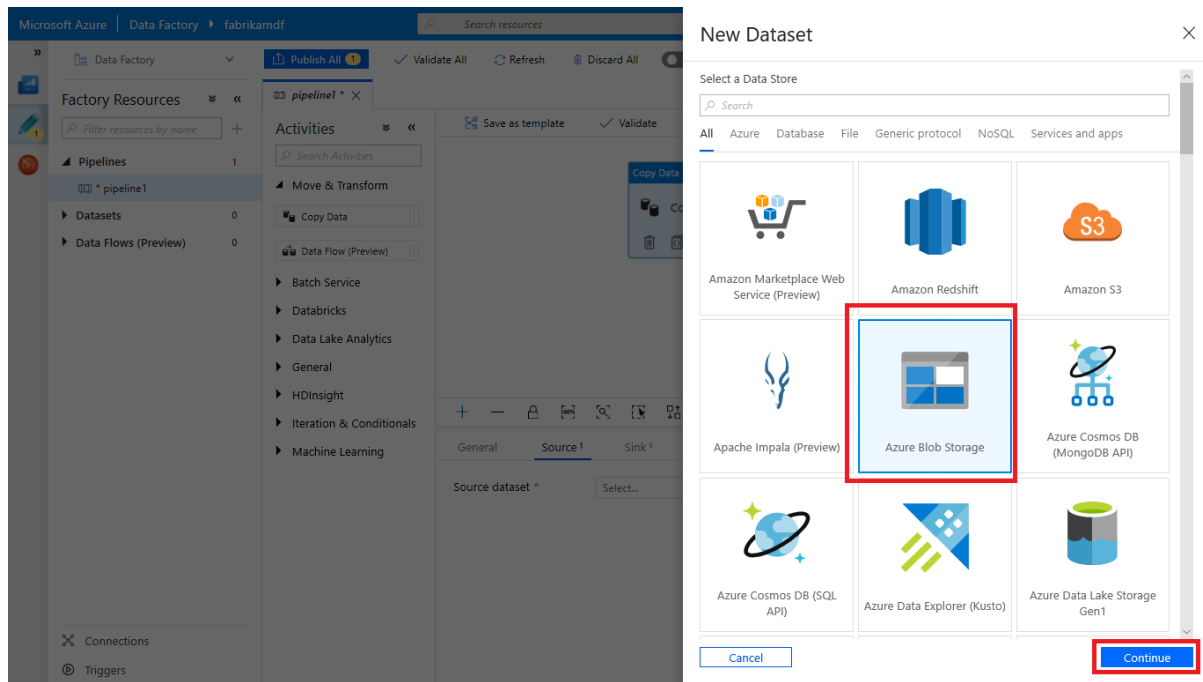
In the pipeline panel drag the copy data to the work space and click on source.



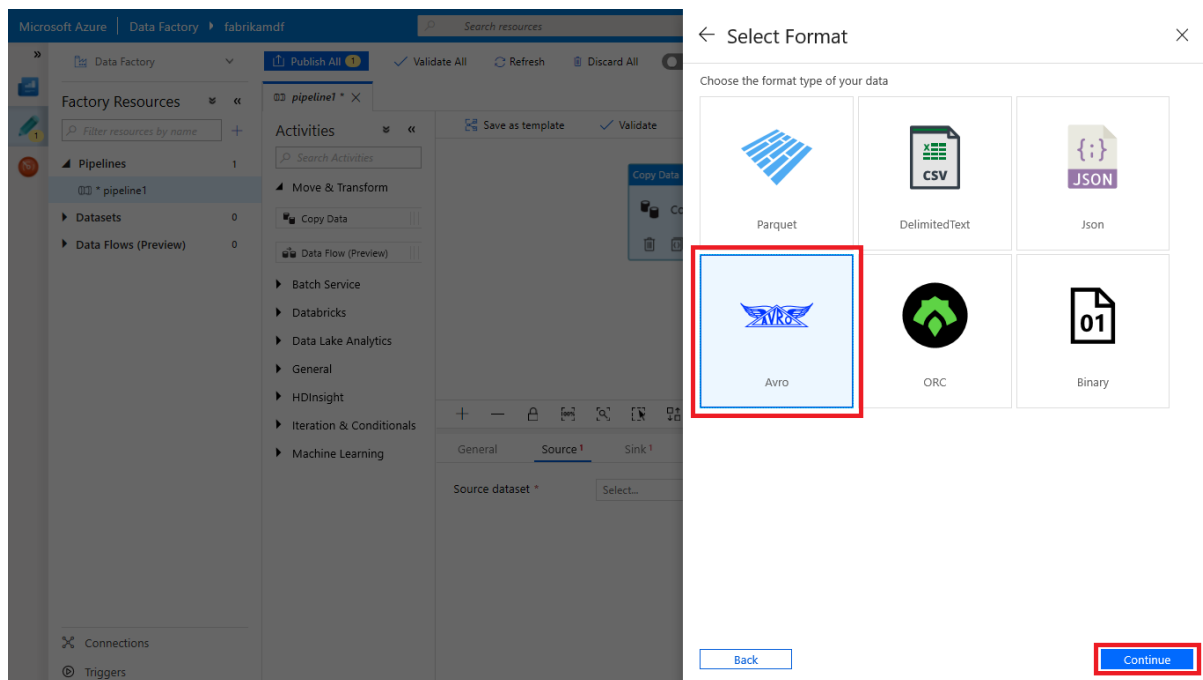
In the source panel click on **New**.

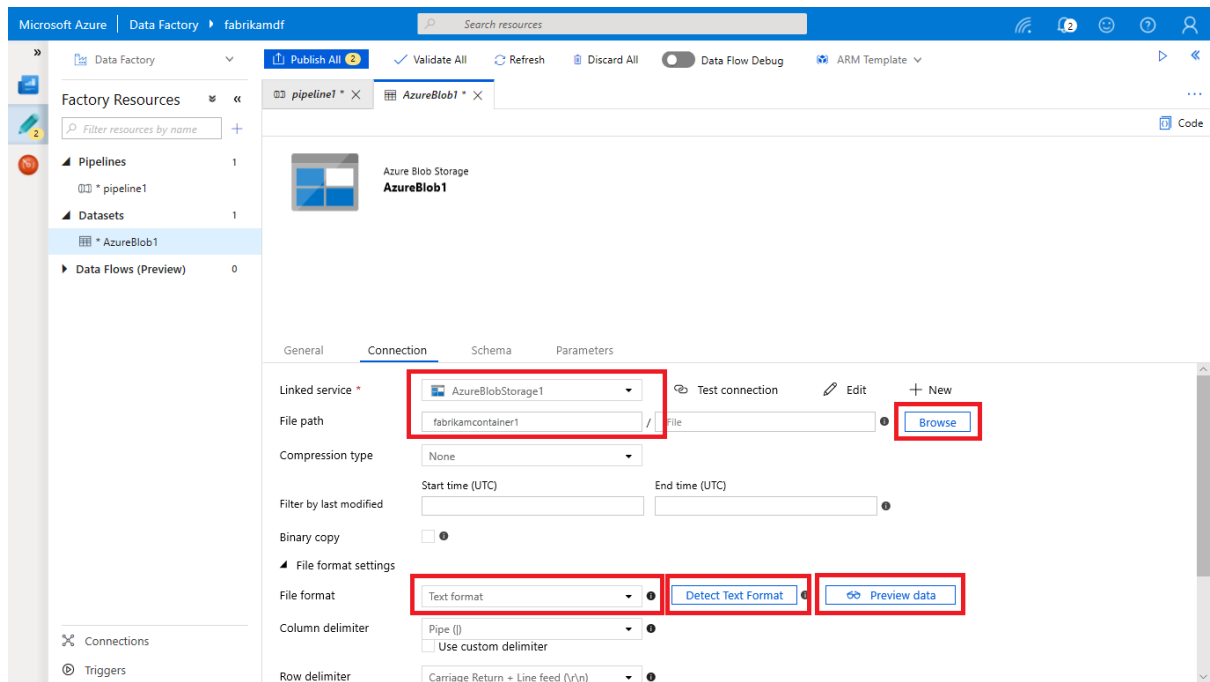


When it prompts select Azure Blob Storage and click on **Continue**.

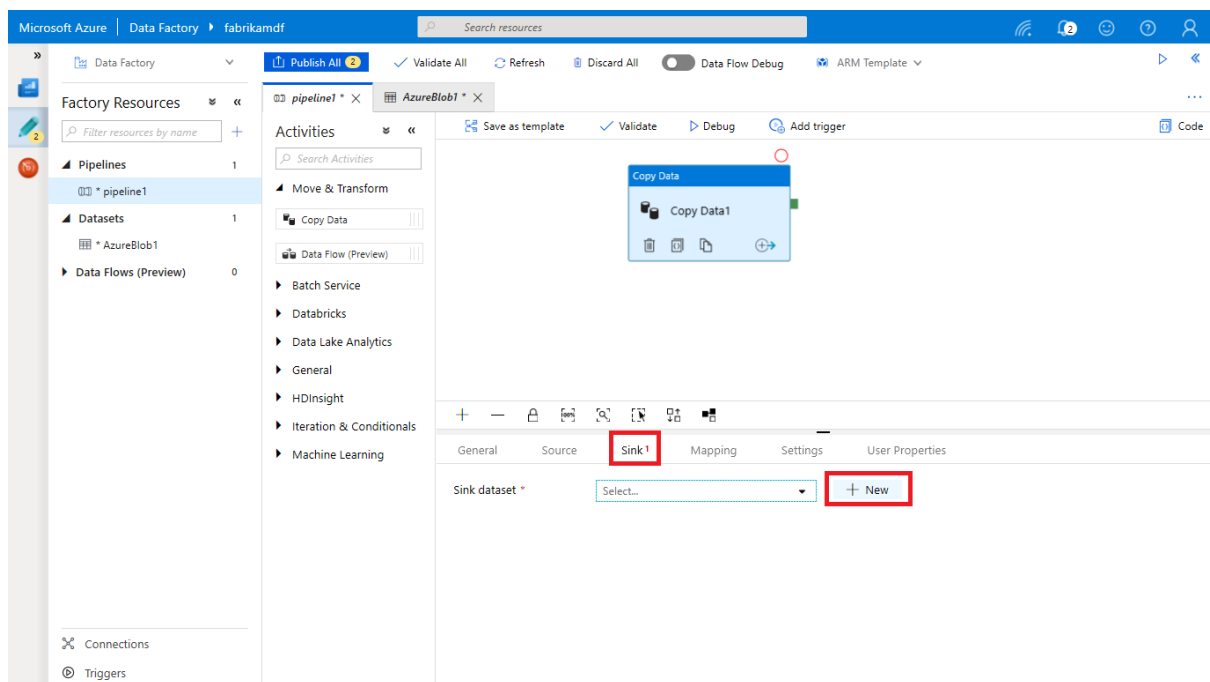


When it prompts click on **Avro** and configure the required settings. After configuring the required settings click on preview data.

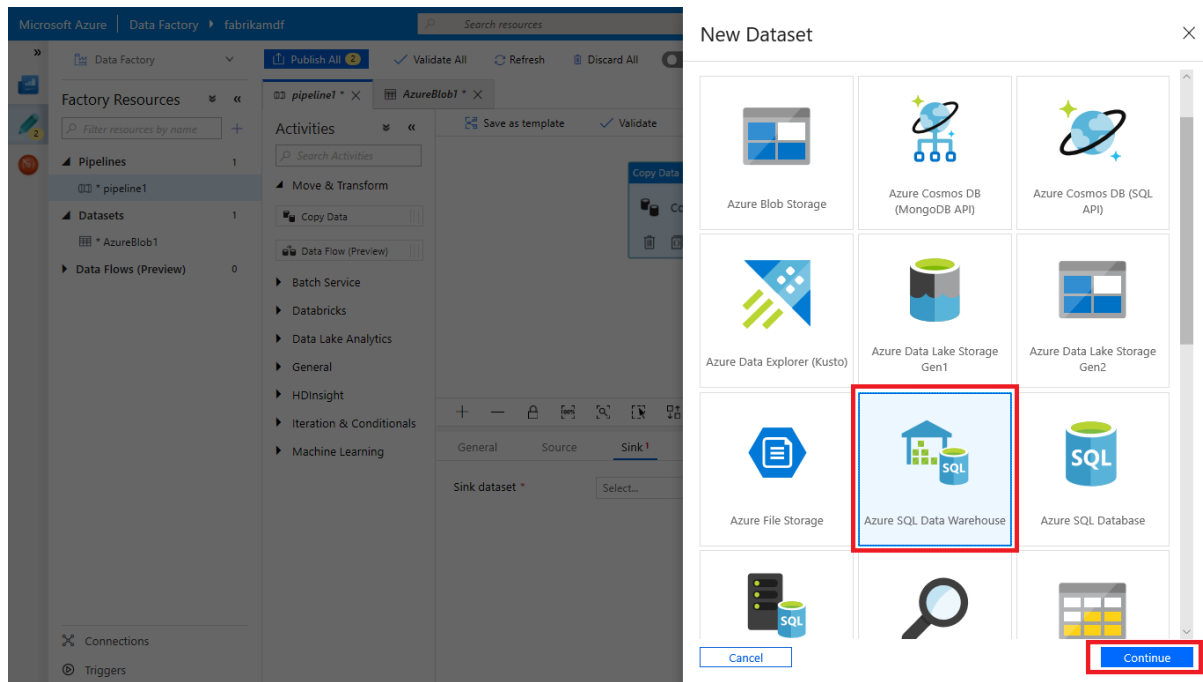




Monitor the data preview and navigate to **pipeline>sink**. In the sink panel click on new.



When it prompts select Azure SQL Data warehouse and click on continue.

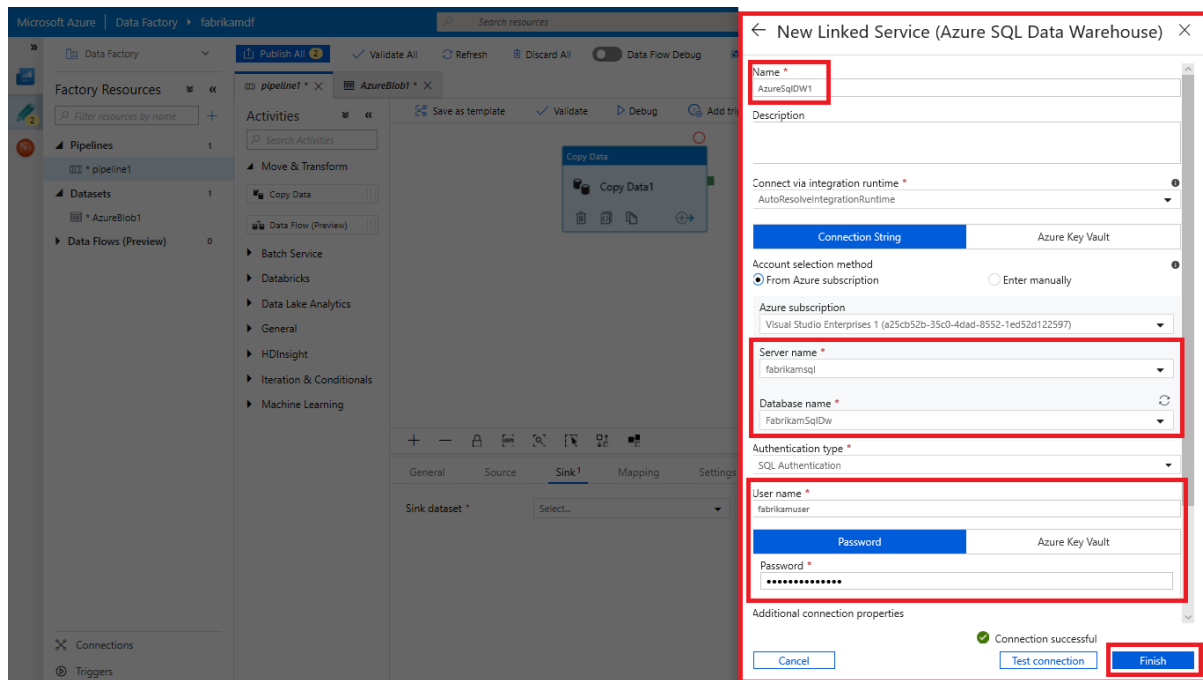


In that panel link the SQL Data warehouse that you have created previously in this exercise.

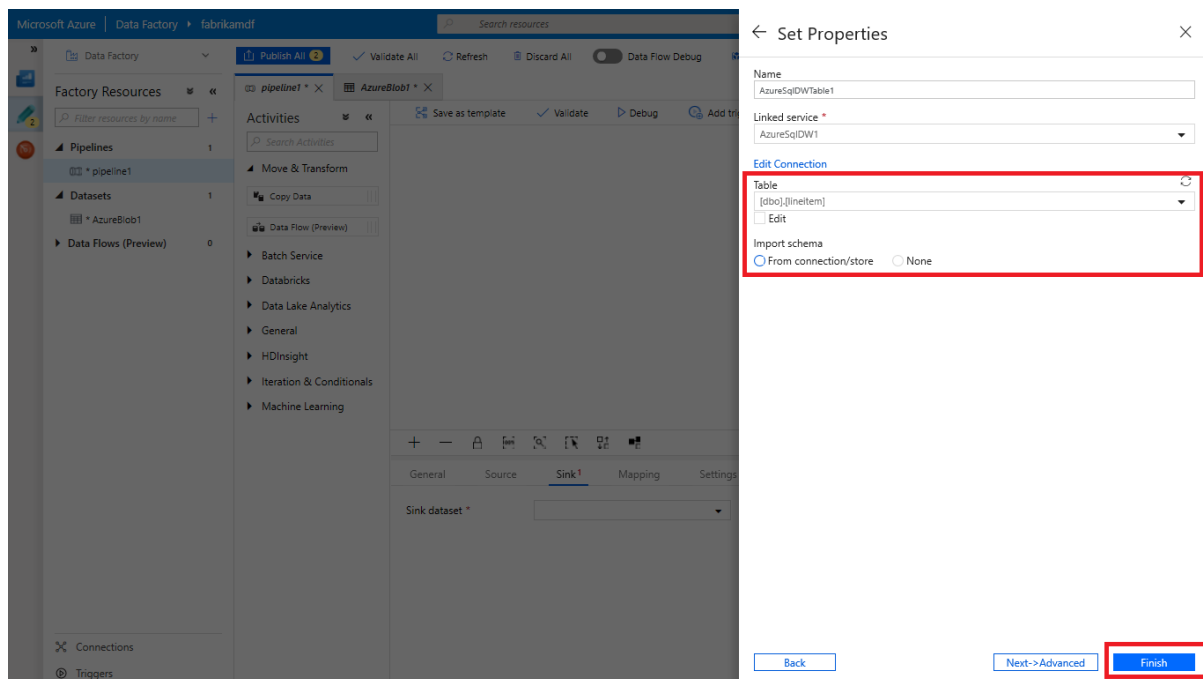
- Name: AzureSqlDw1
- Connect via integration runtime: Default
- Account selection method: From Azure subscription
- Azure subscription: Select a valid one
- Server name: fabrikamsql
- Database name: FabrikamSqlDw
- Authentication type: SQL Authentication
- User name: FabrikamUser
- Password: Fabrikam@12345

After configuring all the settings click on test connection and note that it shpws connection successful. Then click on **Finish**.

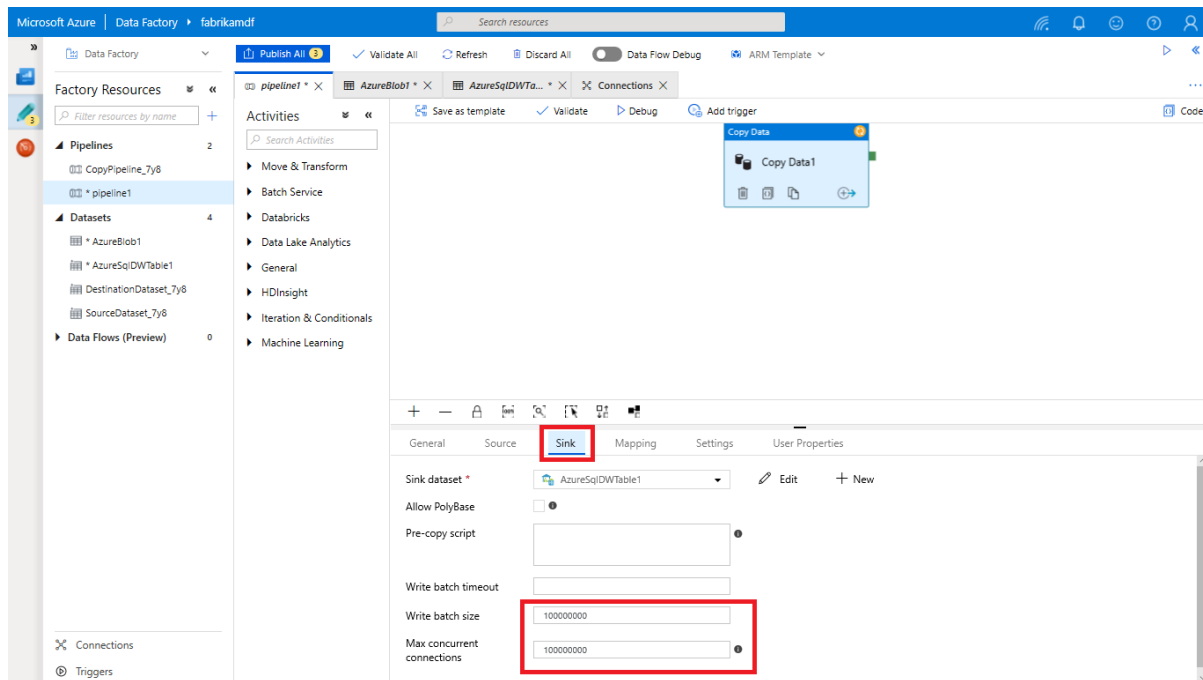




When you click on finish it will navigate you to set properties panel. In that panel select the table which you have created using the SQL Server Management Studio and click on finish.

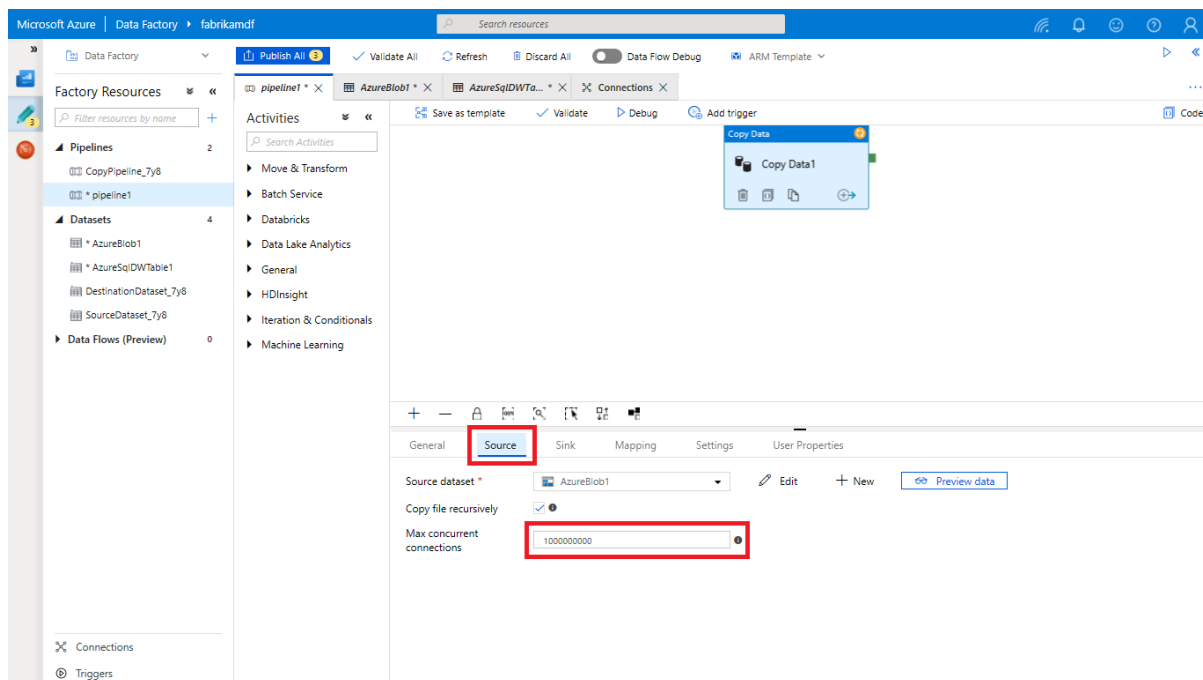


In the sink panel write the batch size and maximum concurrent connection to 100000000.

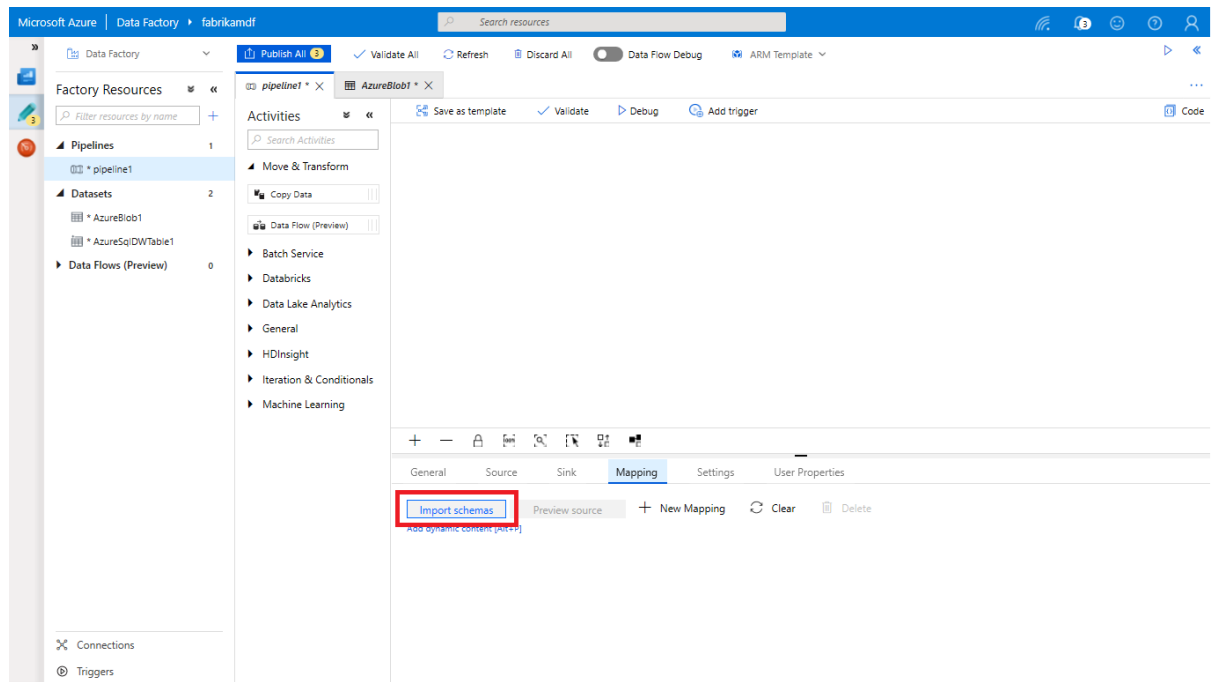


After setting the sink panel navigate to **Source** panel and configure the following settings

- Source Dataset: The storage account that you have created previously in this demo.
- Max concurrent connections: 1000000000

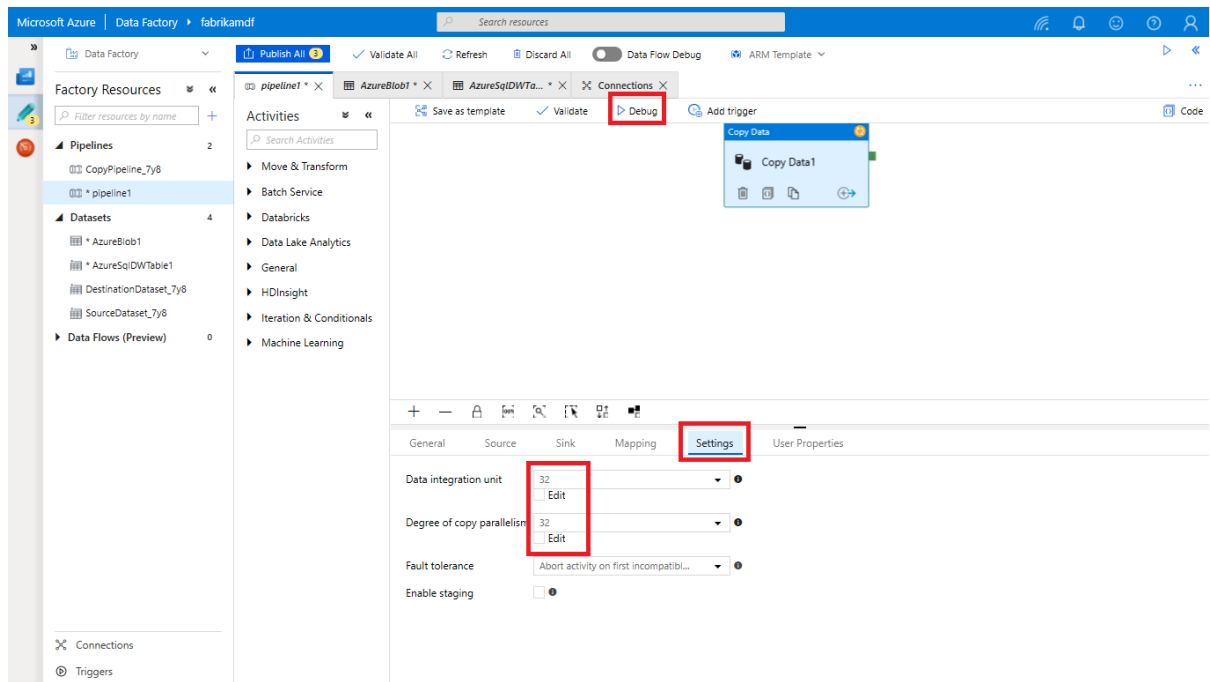


Then navigate to **Mapping** and click on import schemas. When it prompts select the source tables in the blob storage and map them to the schema in the Azure SQL Data Warehouse destination.

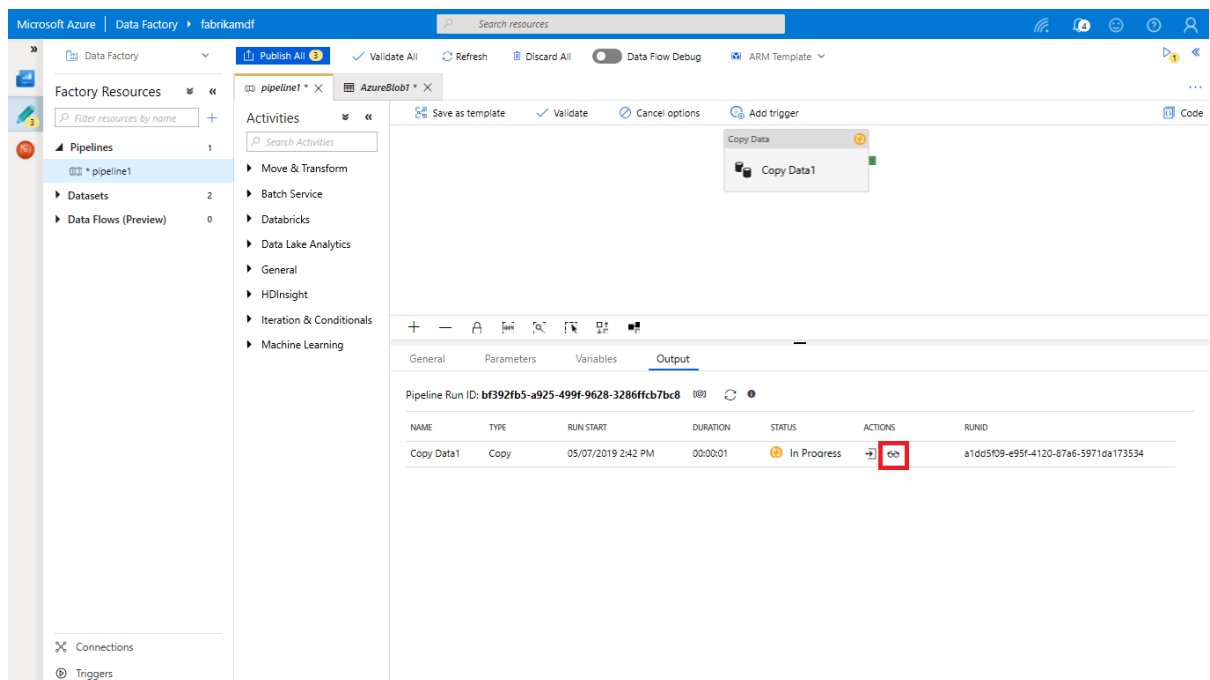


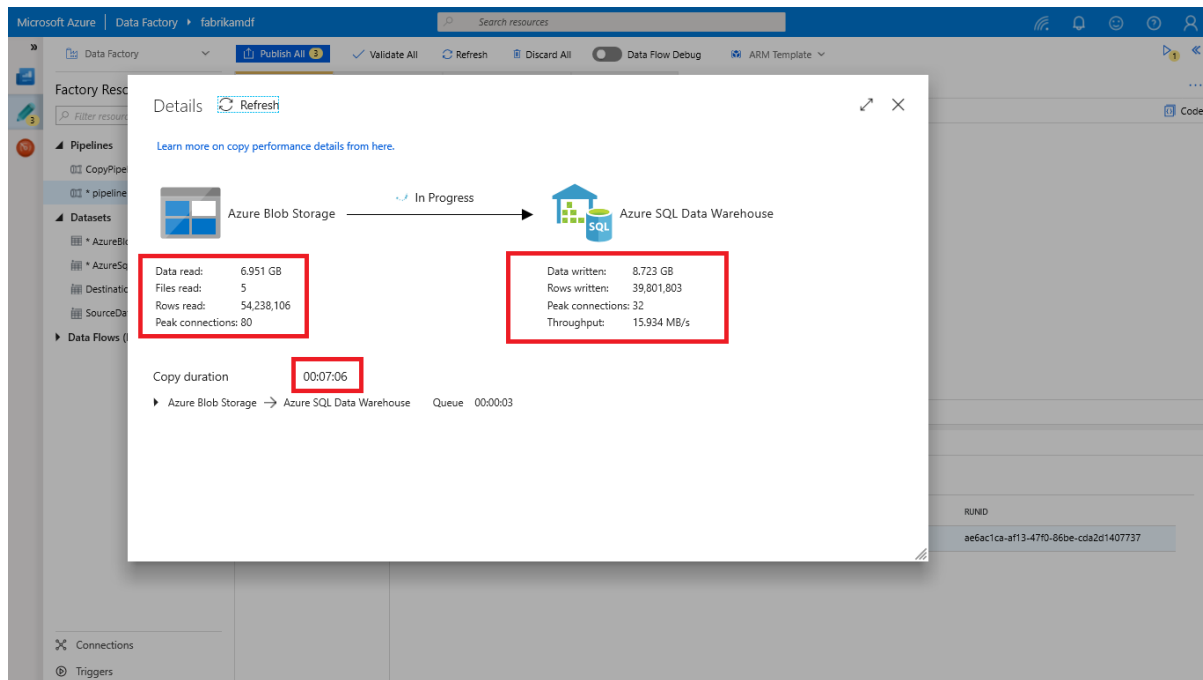
After setting the mapping panel navigate to settings and configure the following settings and click on **Debug**.

- Data integration unit: 32
- Degree of copy parallelism: 32



After clicking the debug icon, the process will be started. Navigate to panel and click on the glass icon to monitor the copy process.





## Summary:

As expected, Fabrikam was able to load all the data from Azure Storage Account to Azure SQL Data Warehouse that too at a rate of 1Gb/min. Now, this made the team of Fabrikam to go with Azure SQL Data Warehouse and Data Factory for migration. The team is further looking to understand how they can make use of Azure Blob Storage to store their semi-structured data that is required for performing analytics.