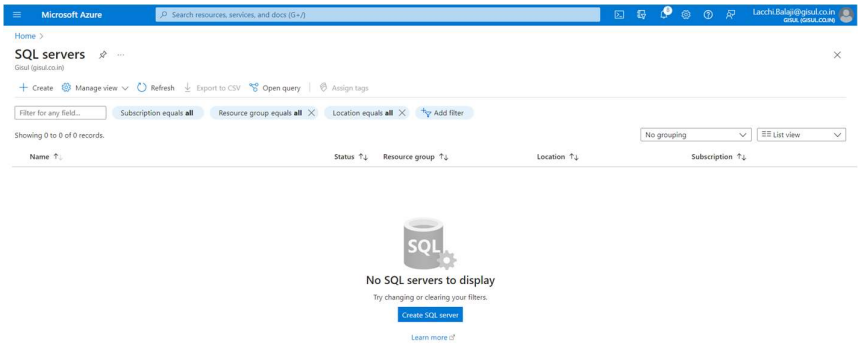
Context

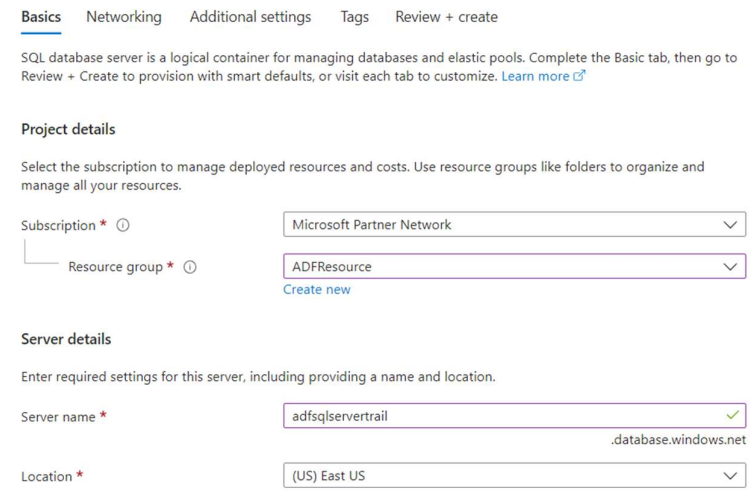
Create SQL Server

1. On the Home page click on SQL servers.

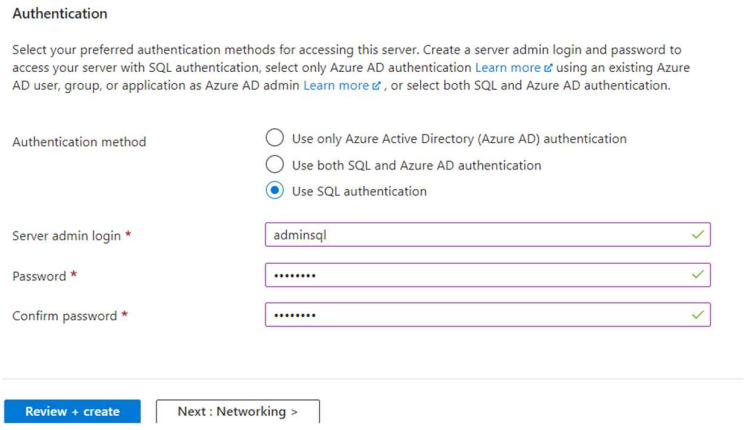
2. Click on Create SQL Server as shown below.



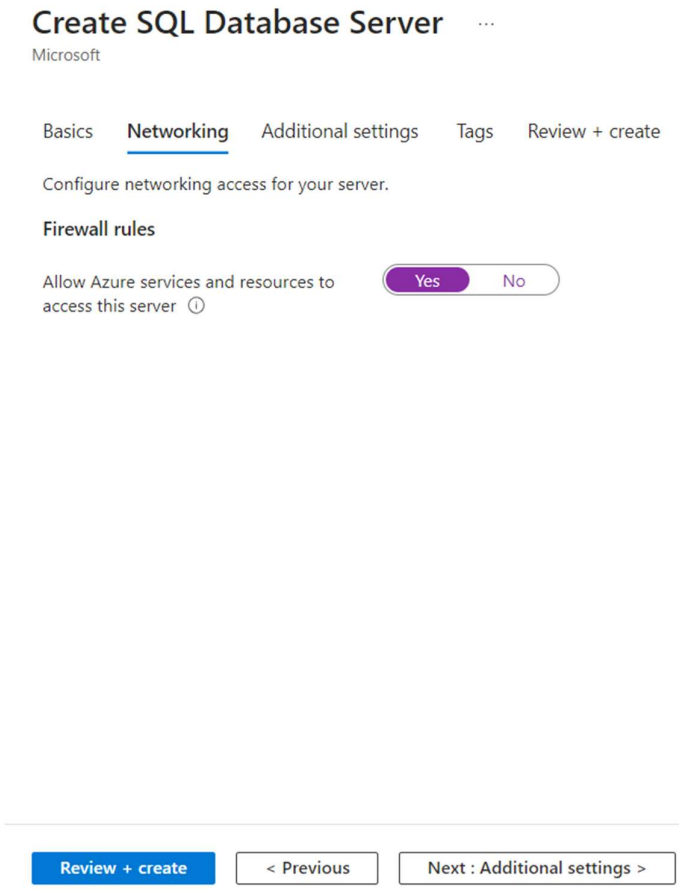
3. Under basic give the below properties as shown below.



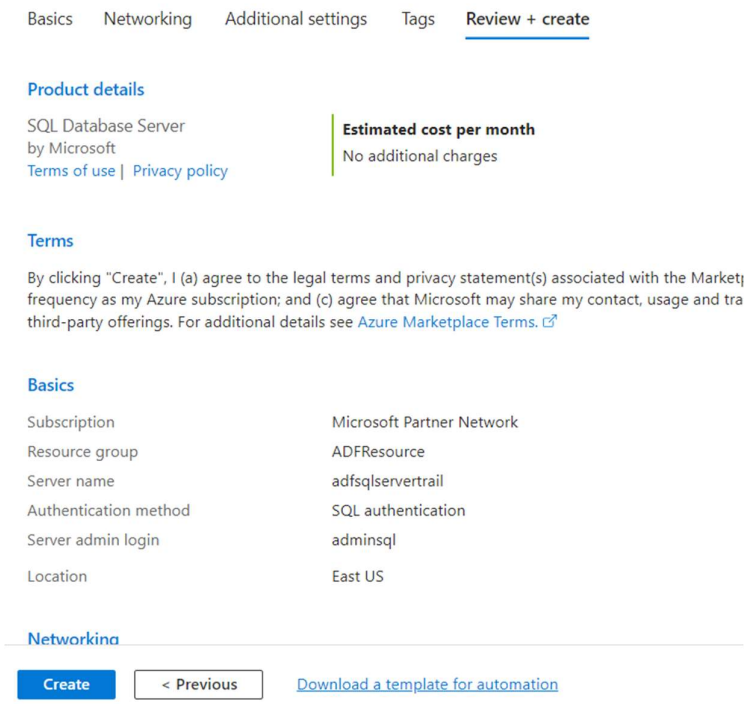
4. Scroll down and give the login name and password then click on the Next option.



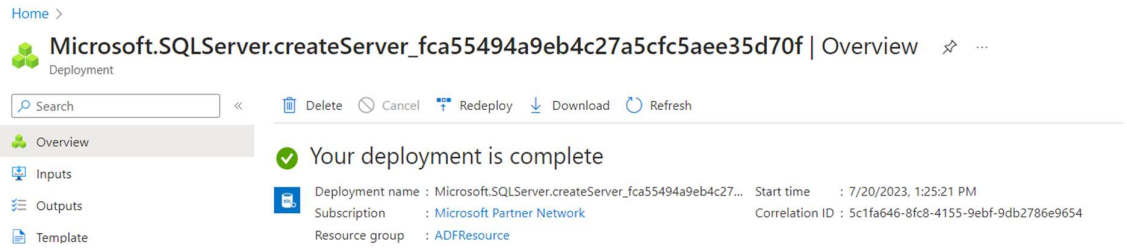
5. Set the below properties and click on Review+Create



6. Next click on Create.

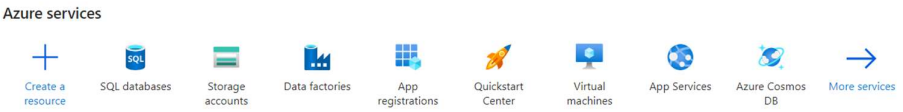


7. After some time you will see a screen like below.

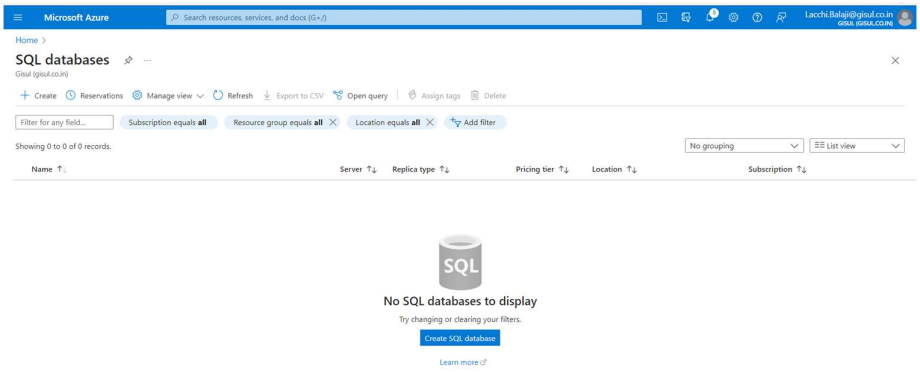


Create SQL database

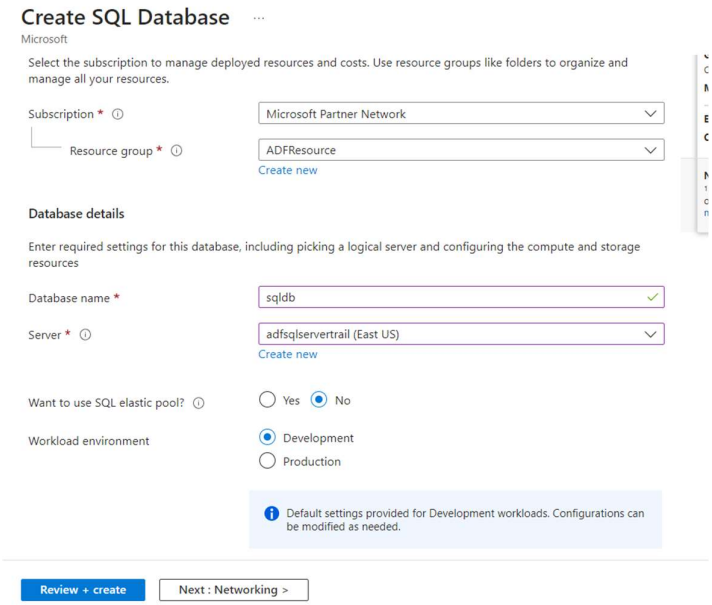
1. On the Home page click on SQL database.



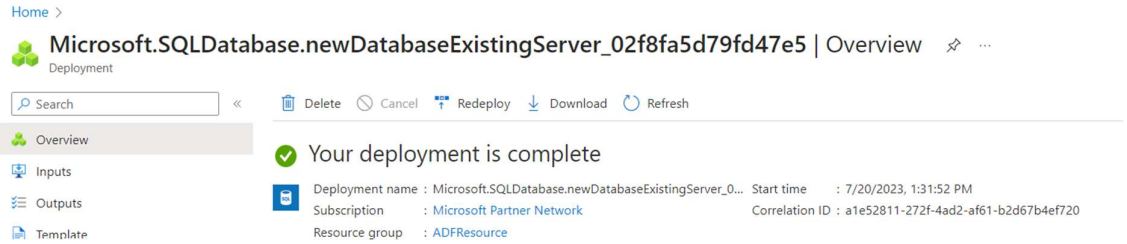
2. Next click on create sql database.



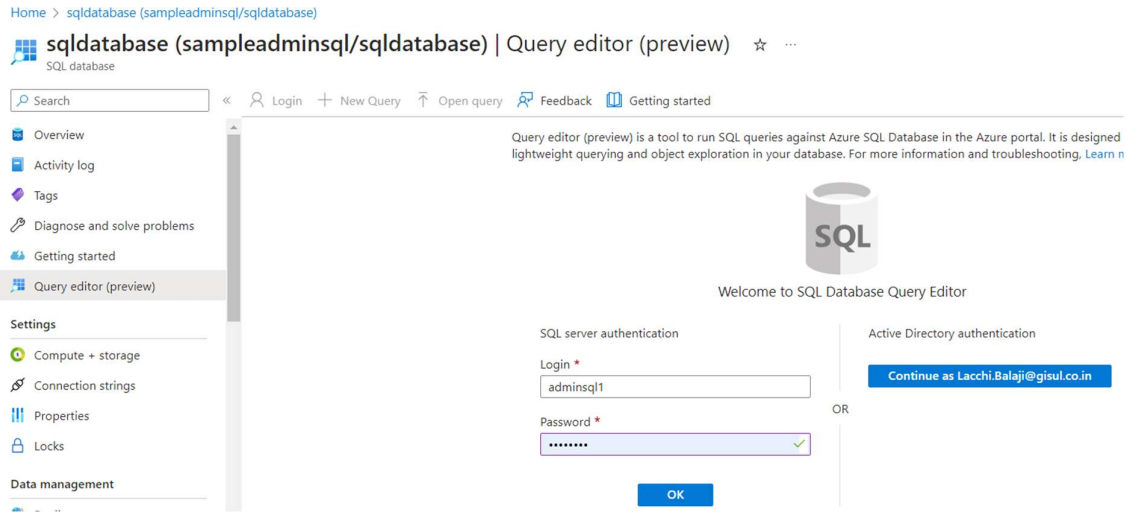
3. Give the Database name as sqldb and select the server that we created before then set the properties as shown below. And click on Reviw+Create.



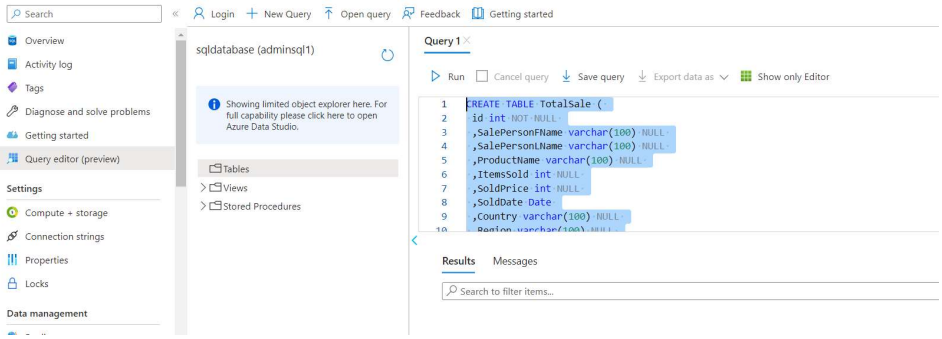
4. Next click on Create.
5. After some time you will see the below screen as shown.



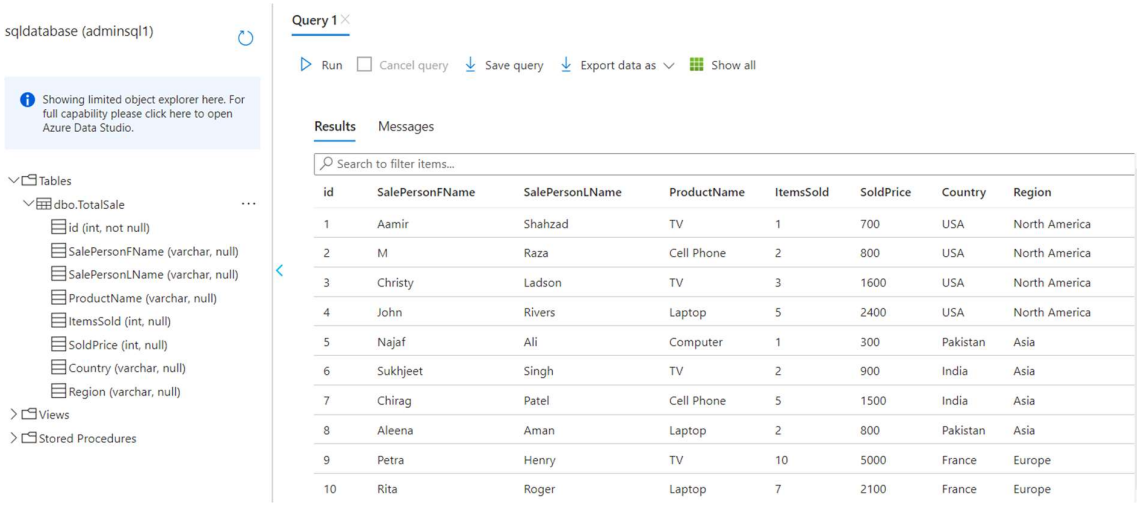
6. Now click on the SQL Database that we created just now.
7. Go to Query editor, give login credentials that we created in the sql server click on Ok.
8. If it asks to allow access to an IP address allow it.



9. A Query editor will open so that we can create tables and export tables also.
10. I have created a table using sql query.



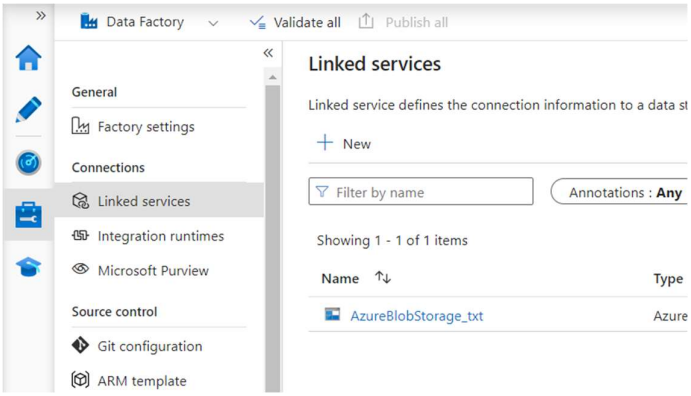
11. Below is the table I have created.



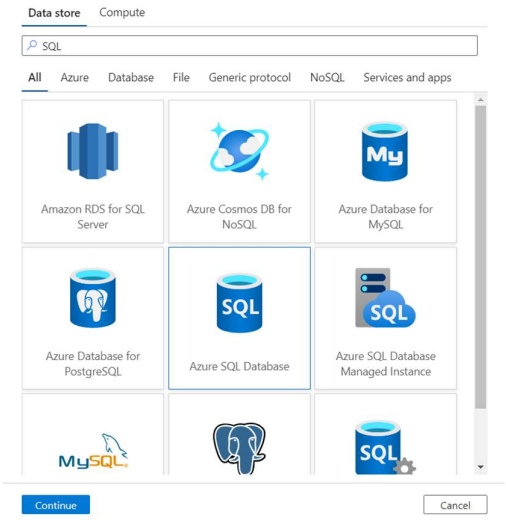
id	SalePersonFName	SalePersonLName	ProductName	ItemsSold	SoldPrice	Country	Region
1	Aamir	Shahzad	TV	1	700	USA	North America
2	M	Raza	Cell Phone	2	800	USA	North America
3	Christy	Ladson	TV	3	1600	USA	North America
4	John	Rivers	Laptop	5	2400	USA	North America
5	Najaf	Ali	Computer	1	300	Pakistan	Asia
6	Sukhjeet	Singh	TV	2	900	India	Asia
7	Chirag	Patel	Cell Phone	5	1500	India	Asia
8	Aleena	Aman	Laptop	2	800	Pakistan	Asia
9	Petra	Henry	TV	10	5000	France	Europe
10	Rita	Roger	Laptop	7	2100	France	Europe

Create a Linked Services

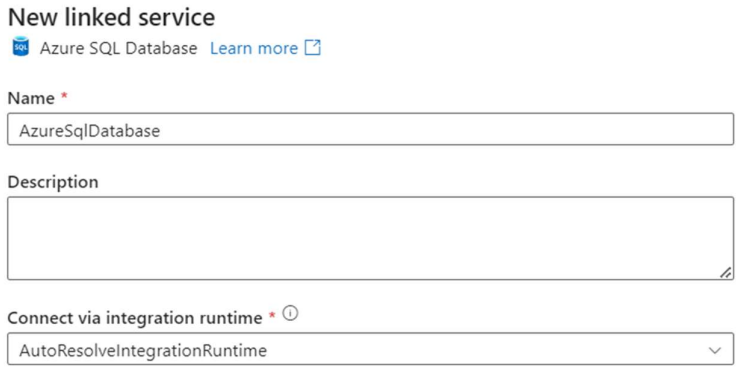
1. Go to Azure Data Factory.
2. Go to the Manage tab and select Linked Services then click on the plus symbol to create.



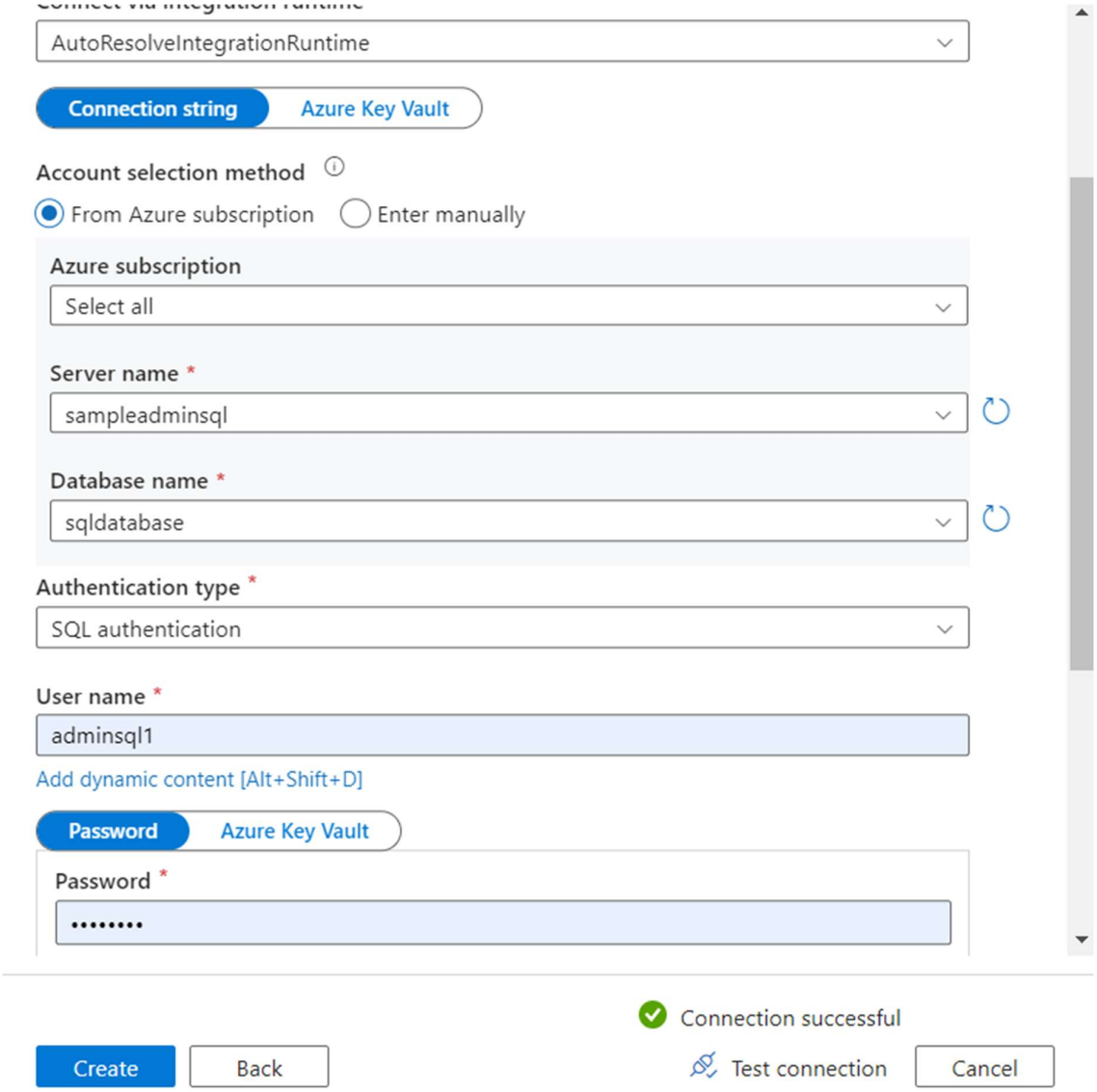
3. Select the SQL Azure Database and click on Continue.



4. Give the Name and Runtime.

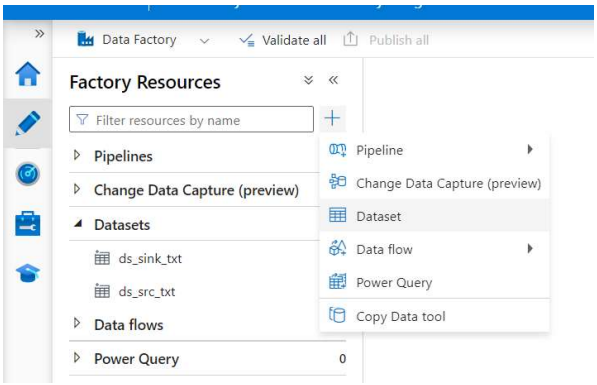


5. And give the below properties and test the connection then click on create.

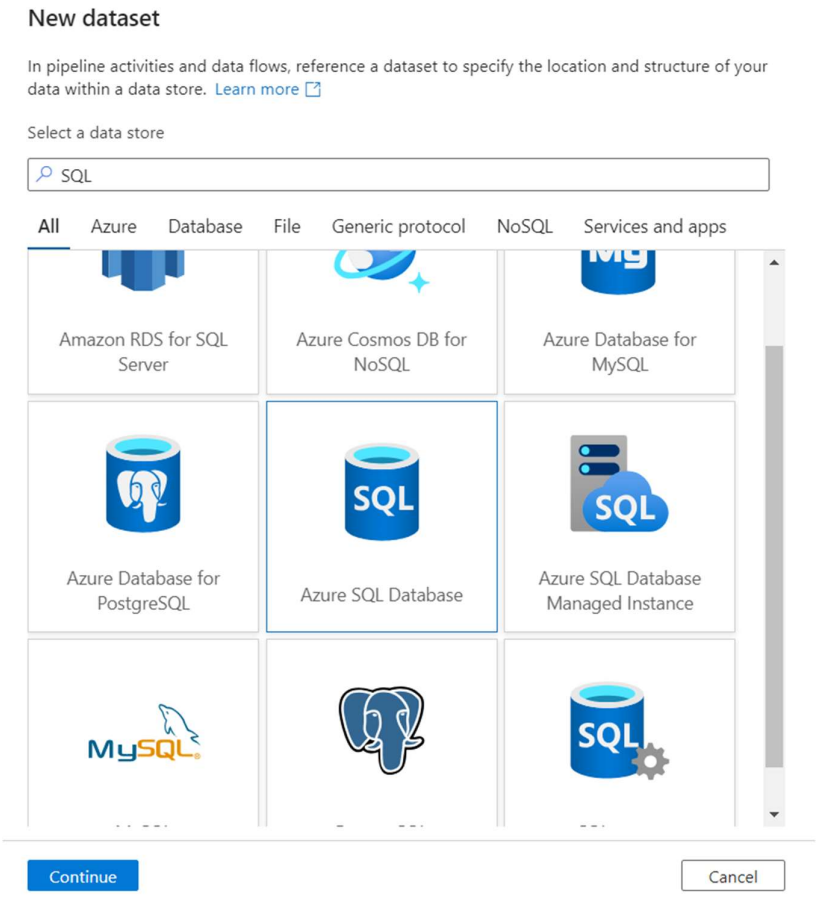


Create Dataset

1. Go to the Author tab and click on Plus symbol then click on Dataset.

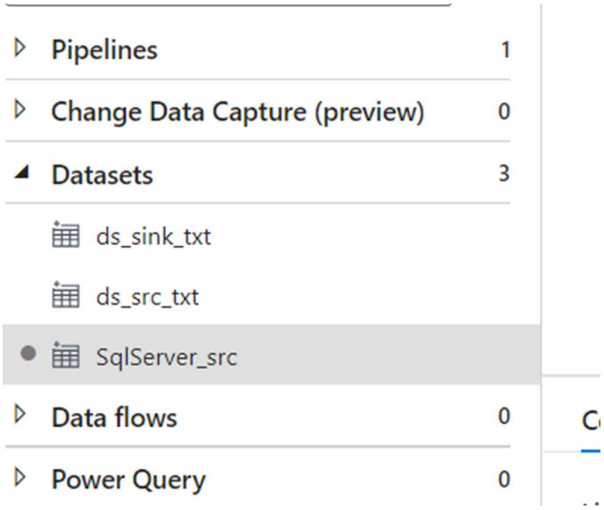


2. Select Azure SQL Database and click on Continue.



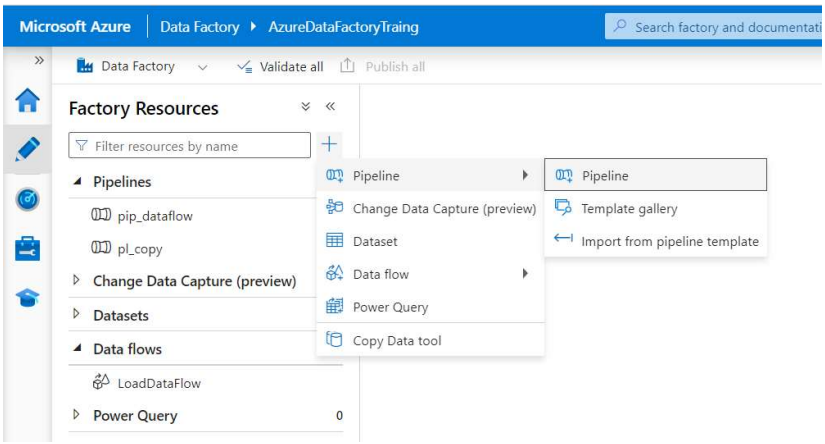
3. Give the Name and select the linked service and table that we created before. Then check the none option and click on Ok.

4. Under the dataset you can see our sql server dataset as shown below.

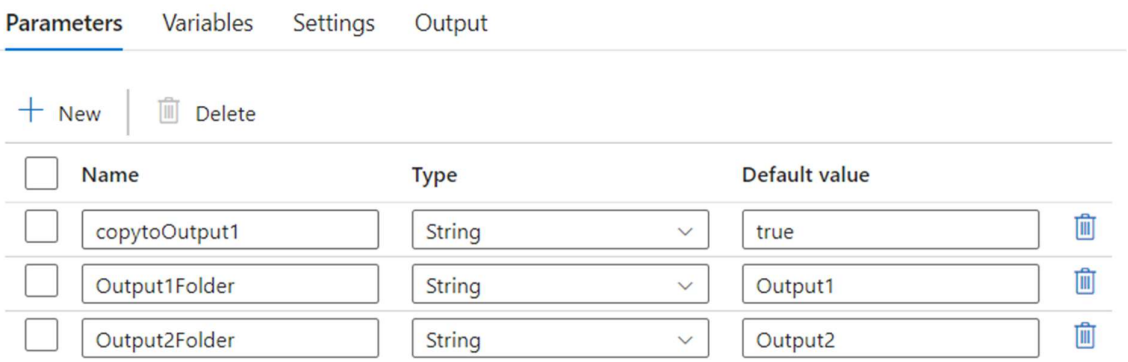


If Condition Activity

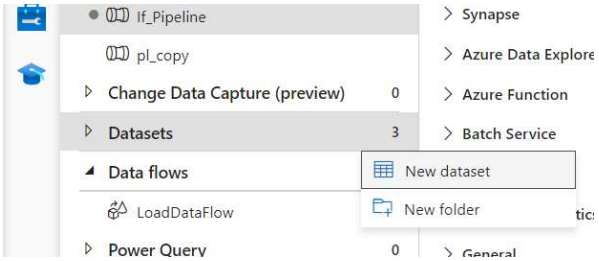
1. Go to Azure Data Factory and create a Pipeline.



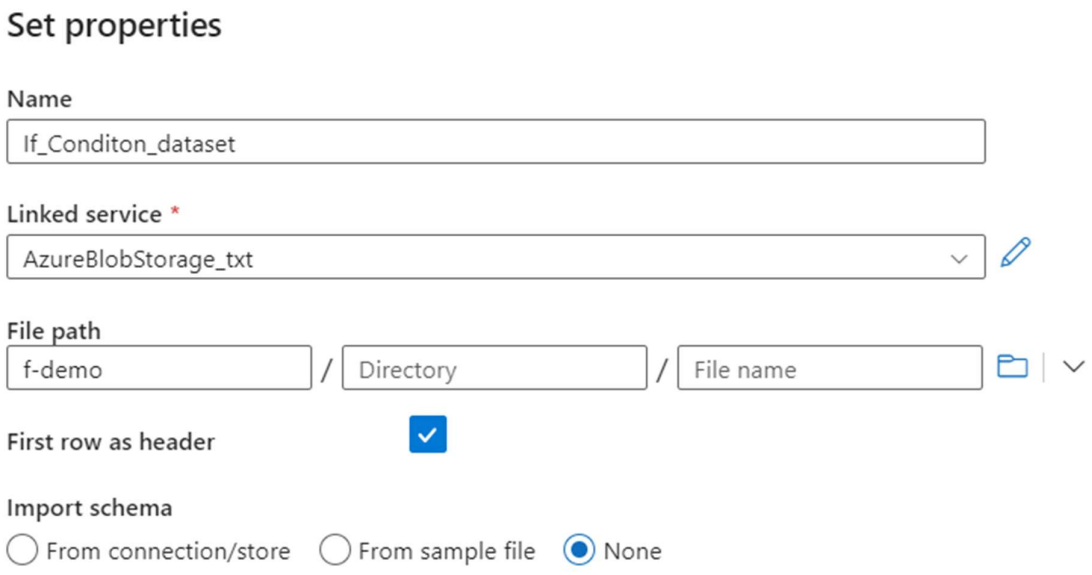
2. Under the Parameter create three parameters as shown below.



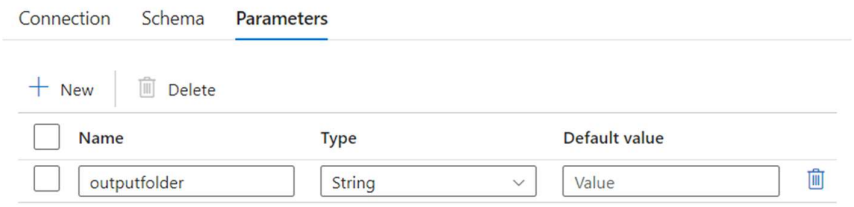
3. Now create a Dataset for the destination.



4. Select Azure Blob Storage and click on continue. Select the Delimited text and click on continue.
5. Give the Name, select the Linked Service, and under the file path give the container name that we created in the previous session. Then check on the None option.

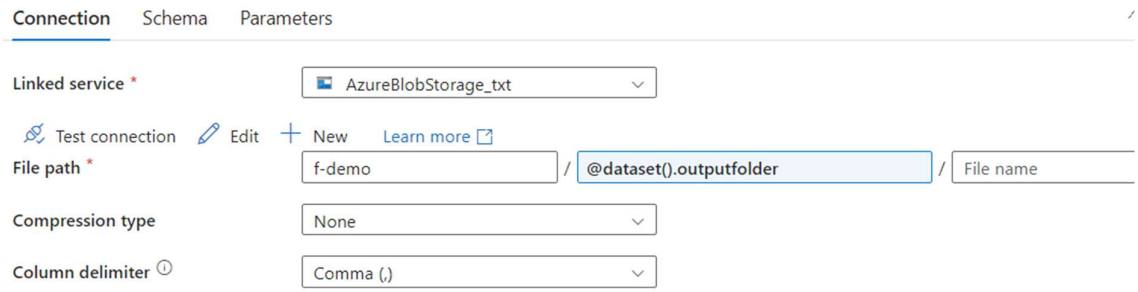


6. Under Parameter create a below parameter.

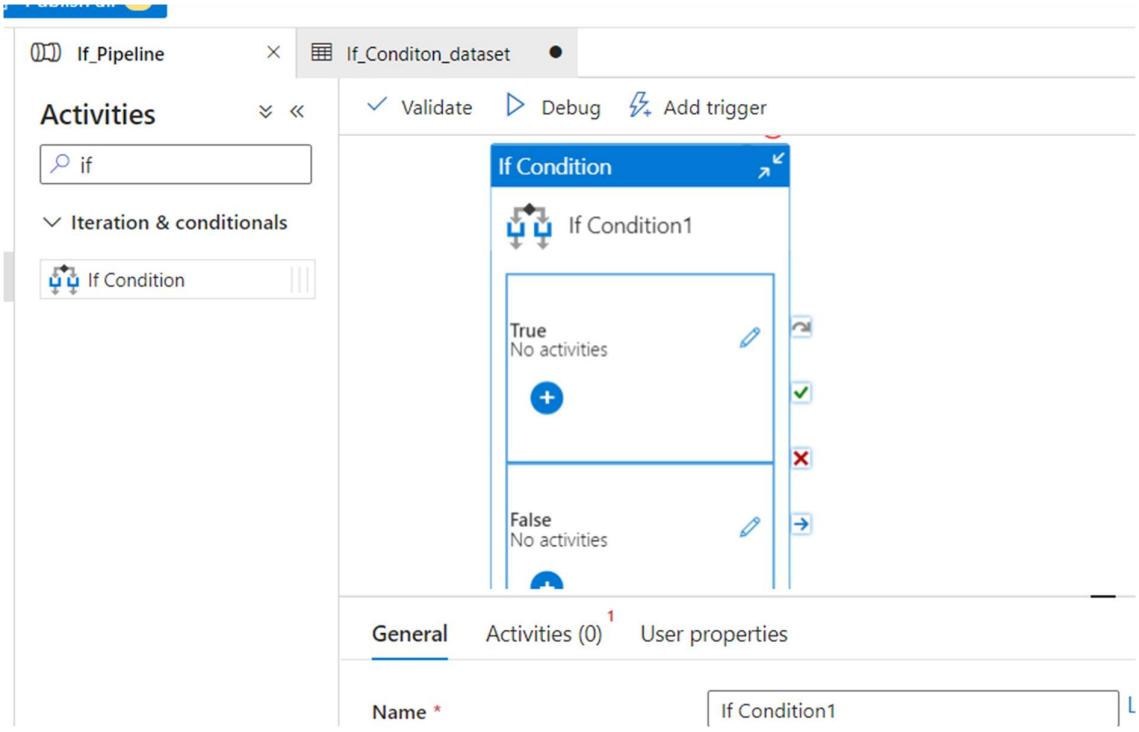


7. Under Connection, in the Directory give the below expression.

Exp: @dataset().outputfolder

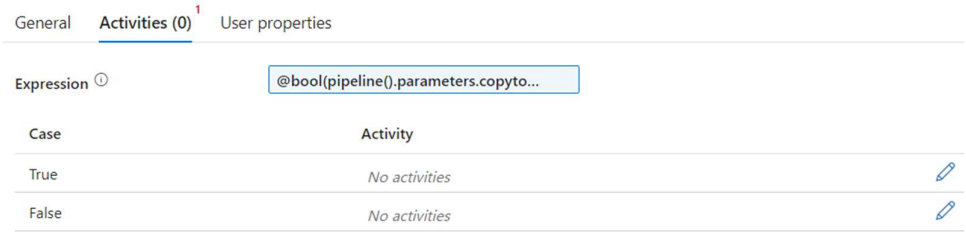


8. Now go to pipeline drag and drop the If condition activity as shown below.

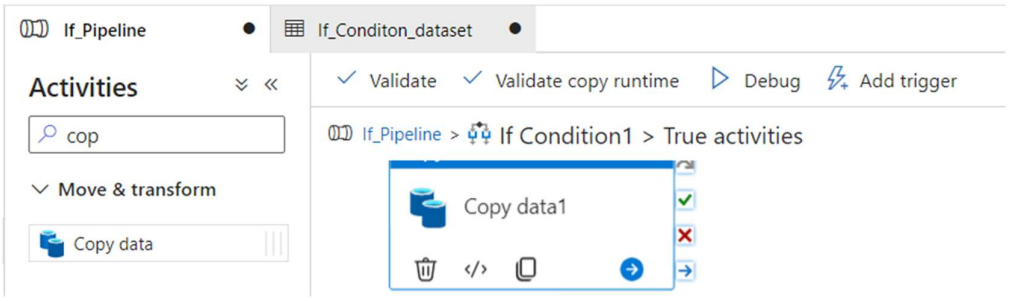


9. Under Activity, in the expression section give the below expression.

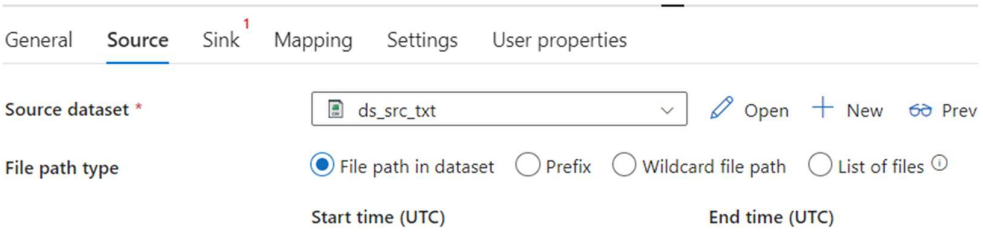
Exp: @bool(pipeline().parameters.copytoOutput1)



10. Next click on the True case pencil symbol.
11. Drag and drop the Copy data as shown below.

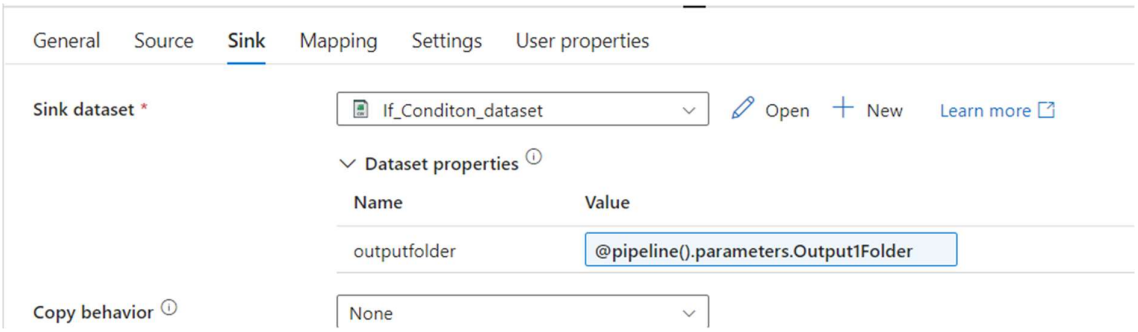


12. Under the Source select the source dataset that we created in the previous session.

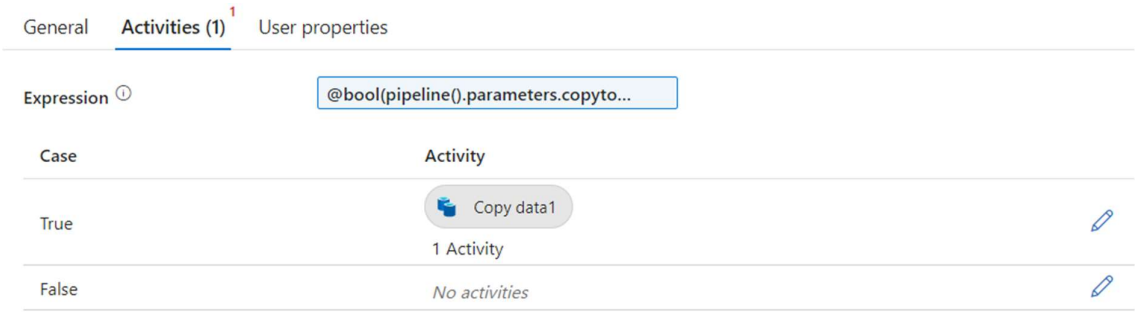


13. Under sink select the destination target that we created before.
14. Under the outputfolder value give the below expression.

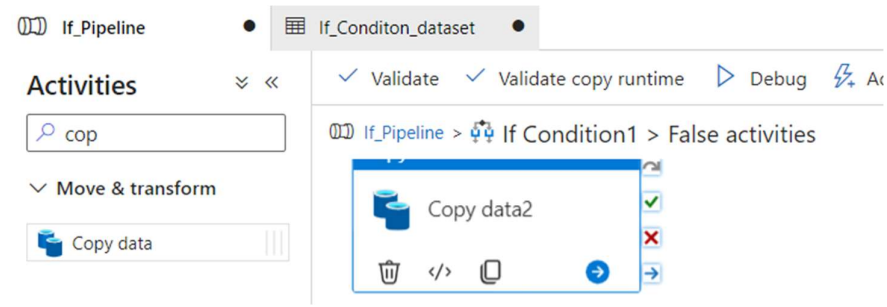
Exp: @pipeline().parameters.Output1Folder



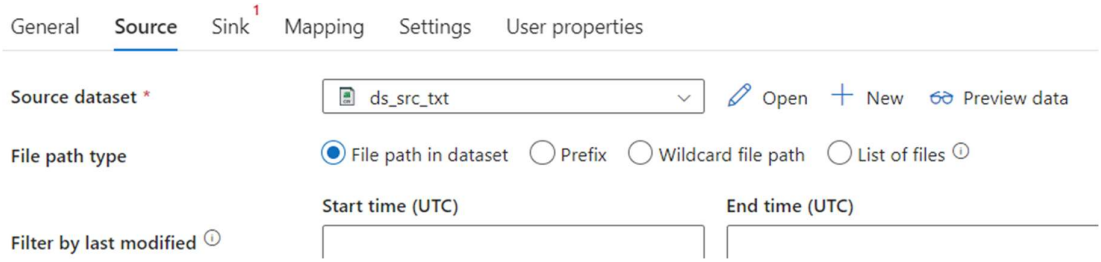
15. Next go to pipeline and click on the False pencil symbol as shown below.



16. Next drag and drop the copy data as shown below.

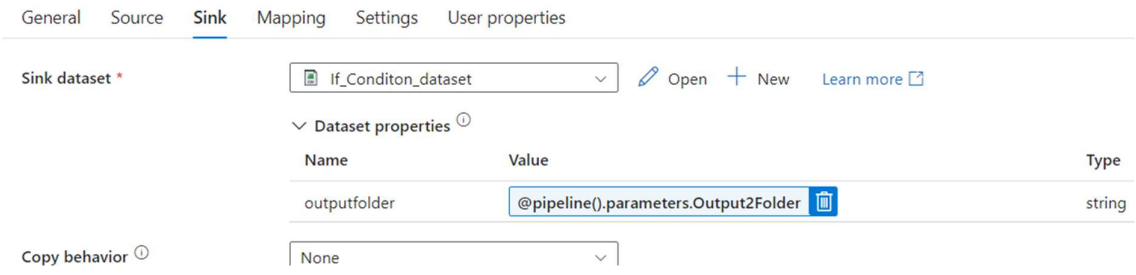


17. Under the source select the source dataset.

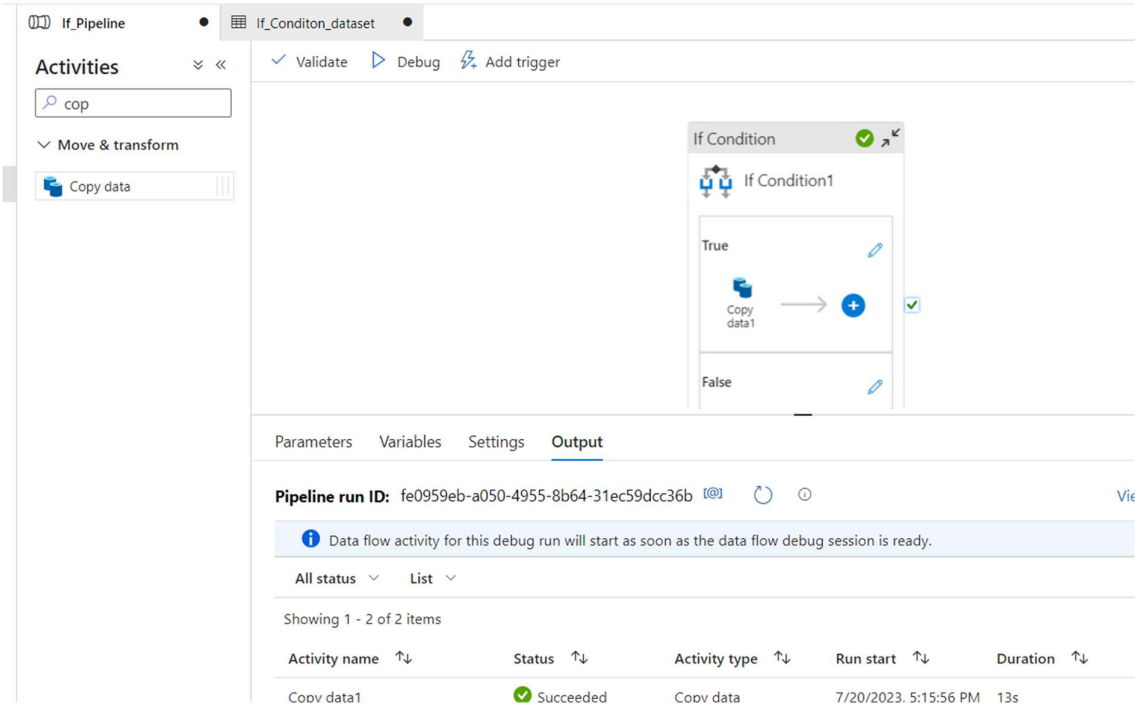


18. Under sink select the destination target that we created before.
19. Under the outputfolder value give the below expression.

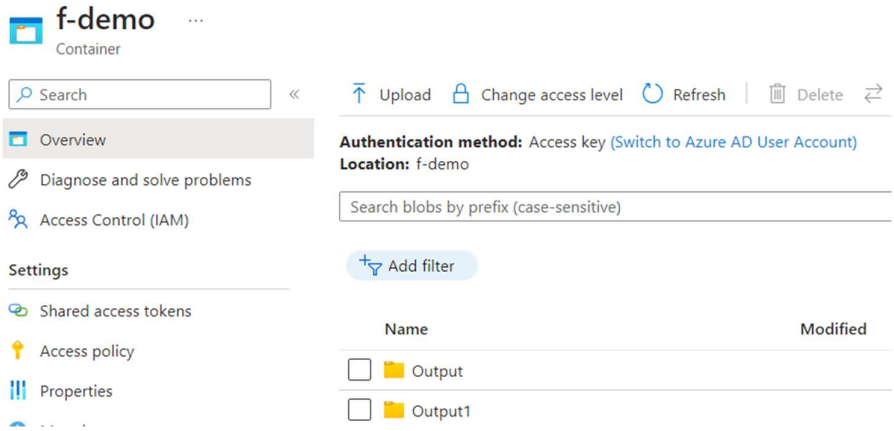
Exp: @pipeline().parameters.Output2Folder



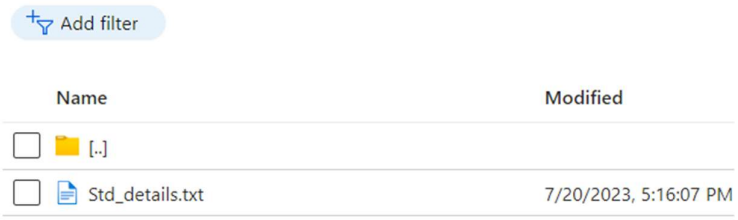
20. Go to Pipeline validate and click on Debug.
21. Our pipeline execute successfully.



22. As we all know in the pipeline parameter we have given the value as true.
23. So, it means the data will be stored in the output 1 folder.
24. Go to your container you will see an Output1 folder created.

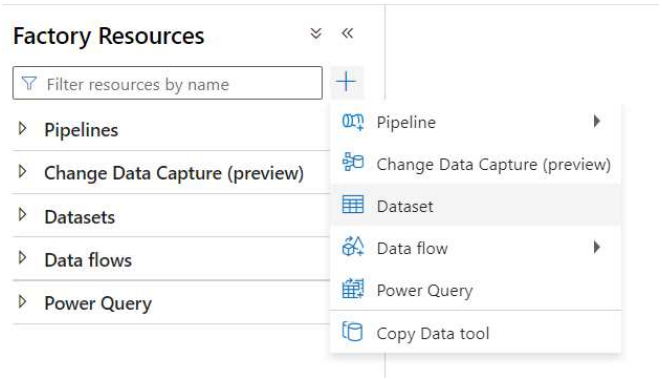


25. If you go inside the folder you will see a Source file.

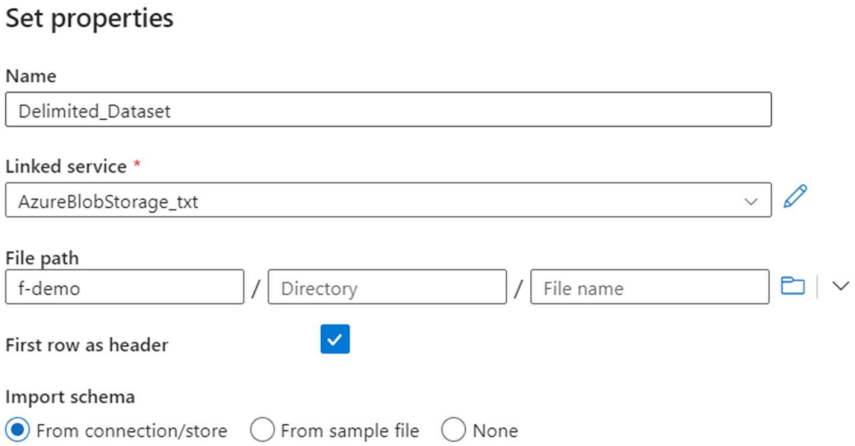


For Each Activity

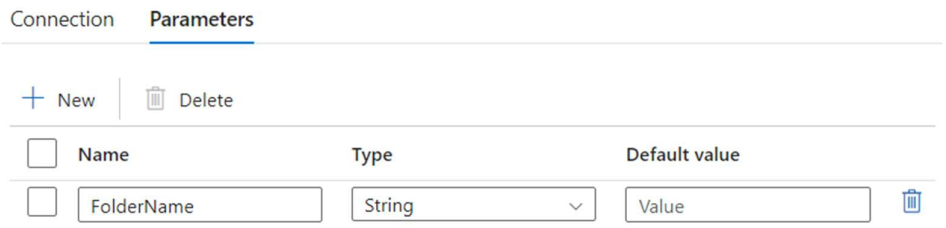
1. Create a Dataset.



2. Select Azure Blob Storage and click continue. Then select Delimited Text and click on Continue.
3. Give name, linked service, and container name then click ok.

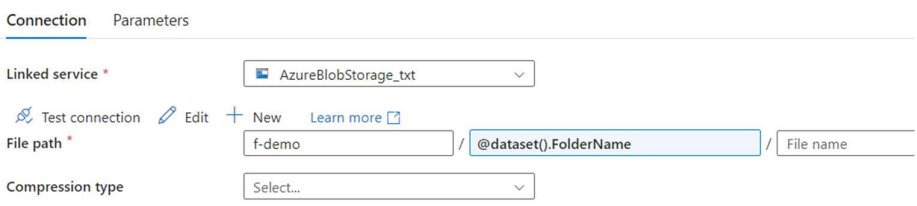


4. Under parameters create a New Parameter as shown below.

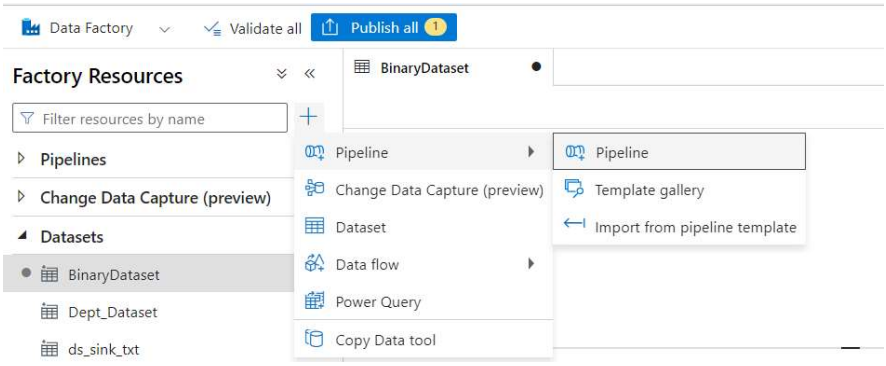


5. Under connection in the directory give the below expression.

Exp: @dataset().FolderName

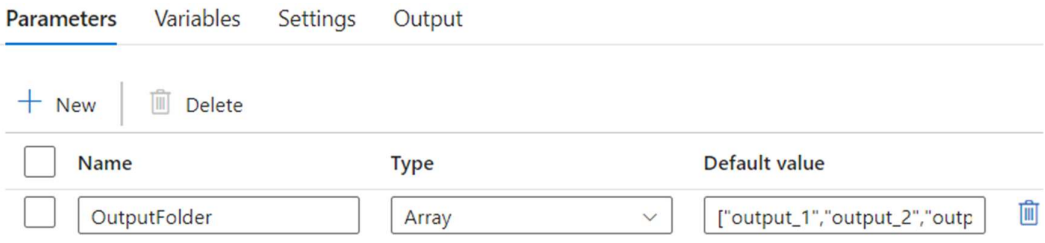


6. Create a Pipeline.

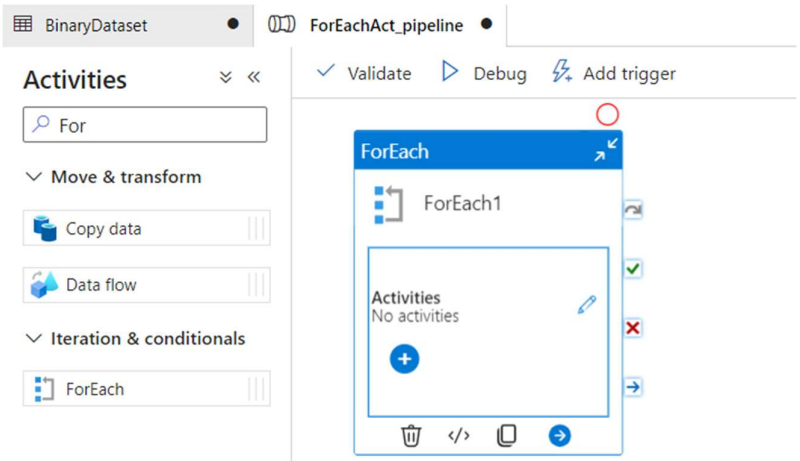


7. Under parameter create a Parameter with the below value.

Value: ["output_1","output_2","output_3"]

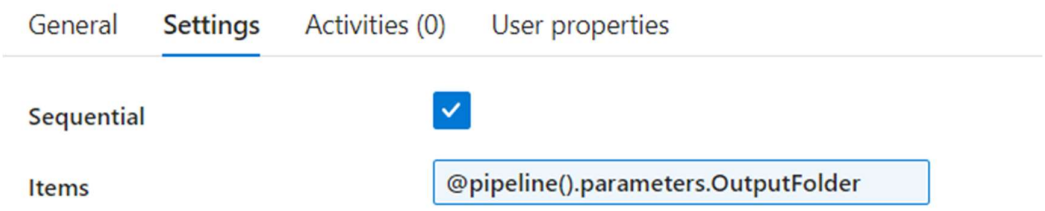


8. Drag and drop the ForEach Activity as shown below.

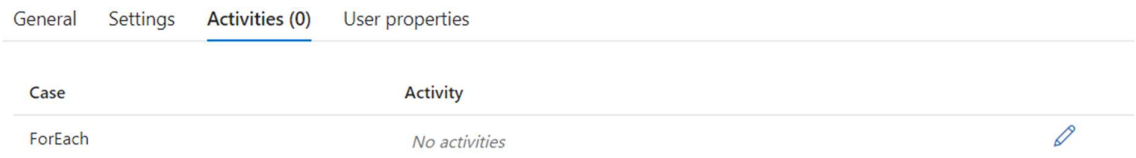


9. Under Settings give the below Expression.

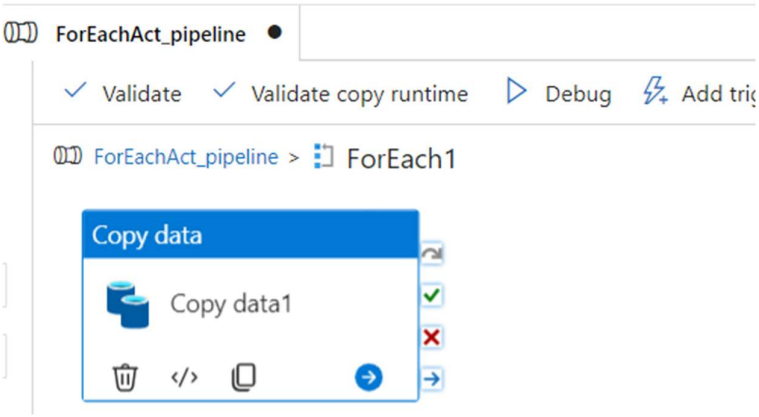
Exp: @pipeline().parameters.OutputFolder



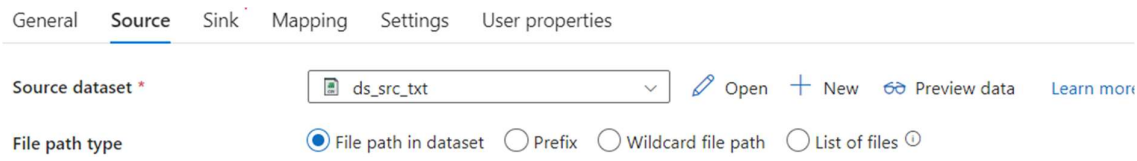
10. Under Activity click on the pencil symbol.



11. Drag and drop the copy data activity.

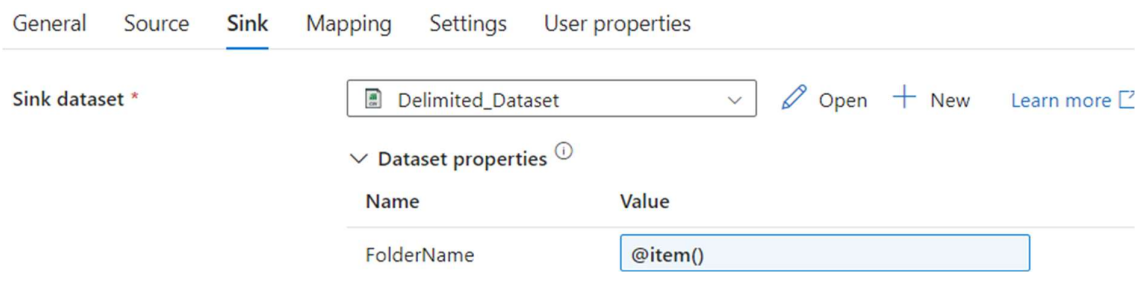


12. Under the source select the Employee Dataset.

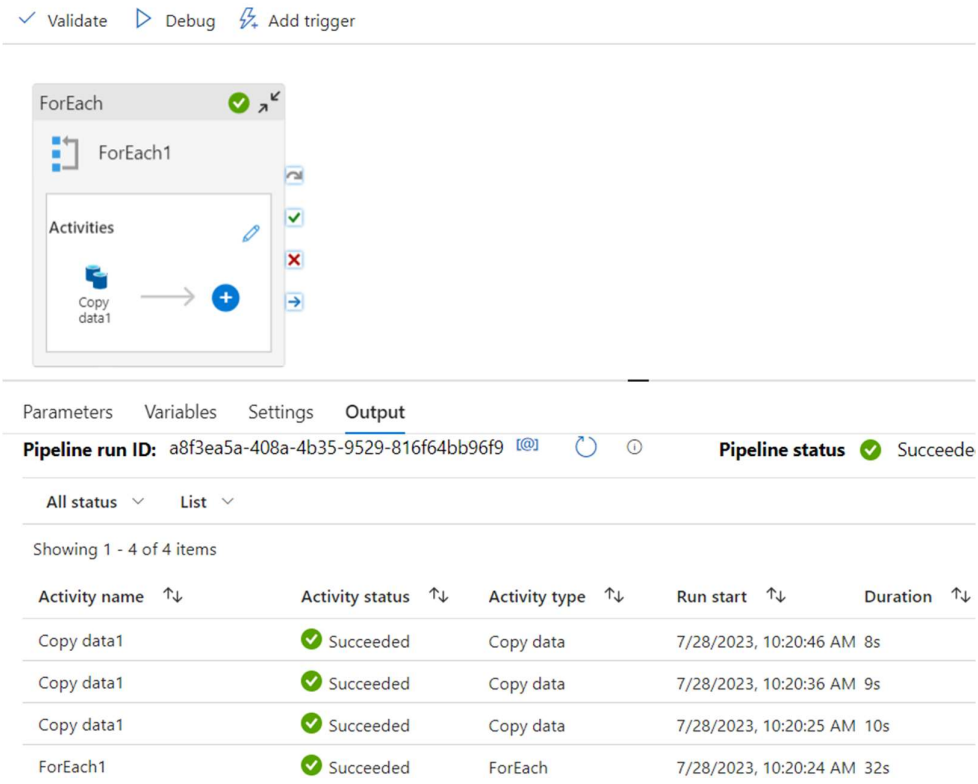


13. Under Sink select the Binary Dataset that we created before and in the FolderName give the below expression.

Exp: @item()



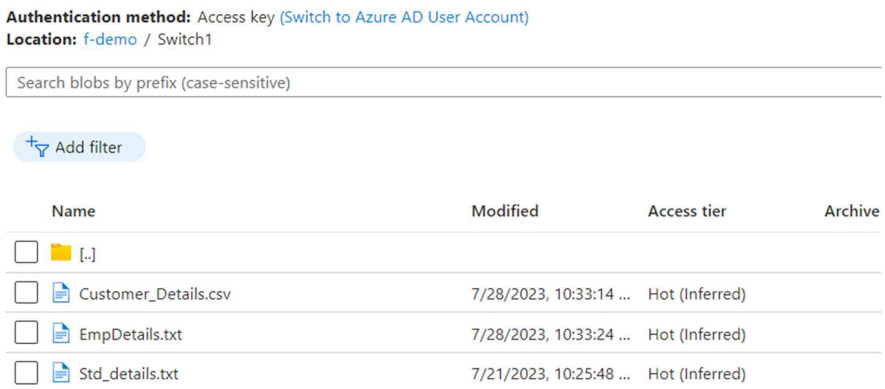
14. Next go to Pipeline and validate it then click on debug.
15. Here Our Pipeline was executed successfully.



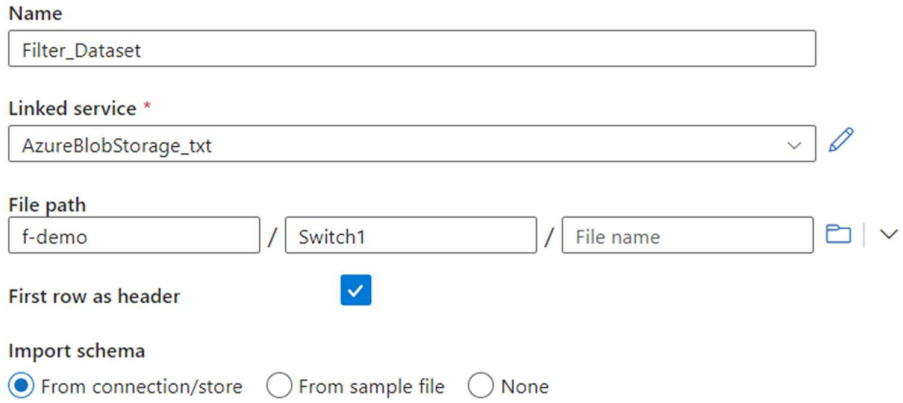
16. Go to Destination and check whether the folders are created or not.

Filter Activity

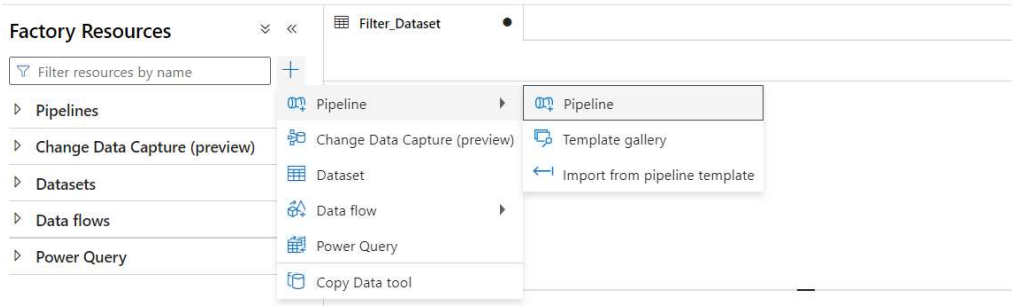
1. Here in the folder I have 2 Txt files and 1 CSV file. So, I need to get the txt files.



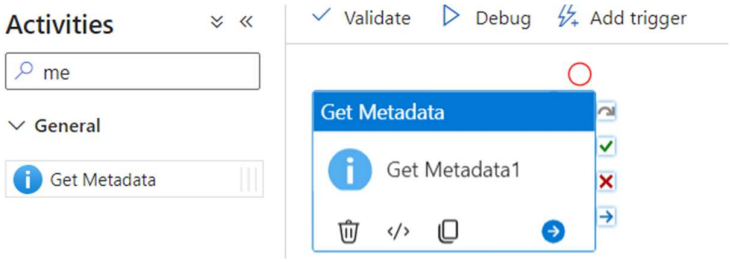
2. Create a Dataset for this folder location and do not select any file.
3. Select Azure Blob Storage and click on continue. Then select Delimited Text and click on Continue.
4. Give the name, linked service, and container then folder.



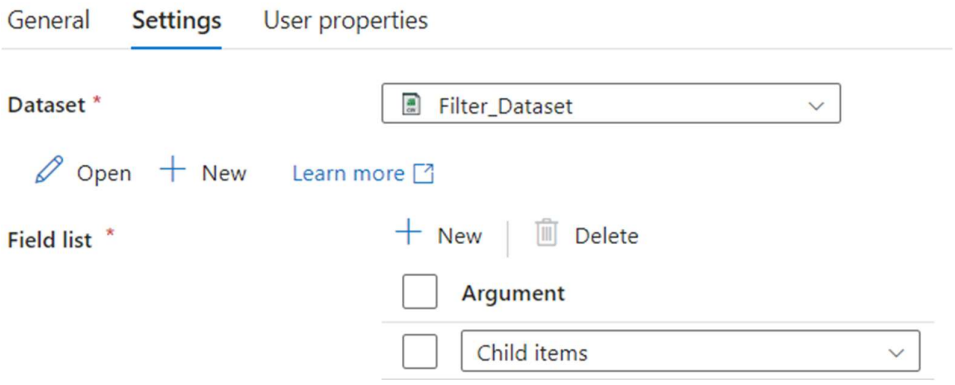
5. Create a pipeline.



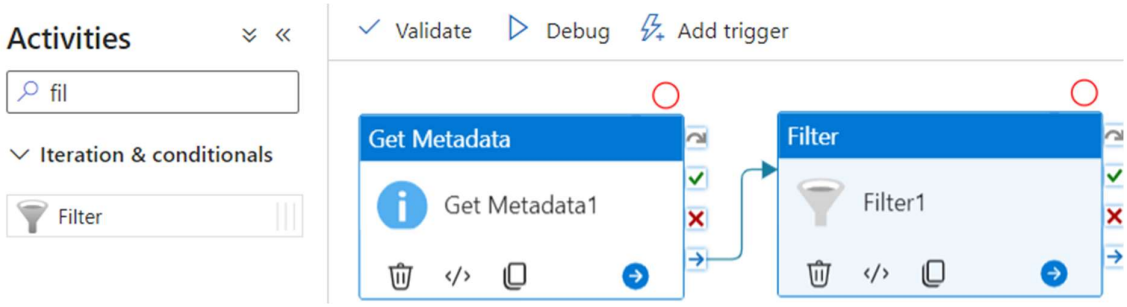
6. Drag and drop the Get Metadata.



7. Under Settings select the Dataset that we created before and select the field list.



8. Drag and drop the filter activity and make the connection as shown below.

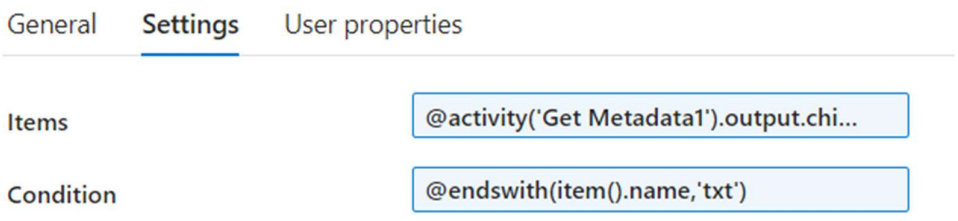


9. Under settings for the Item give the below expression.

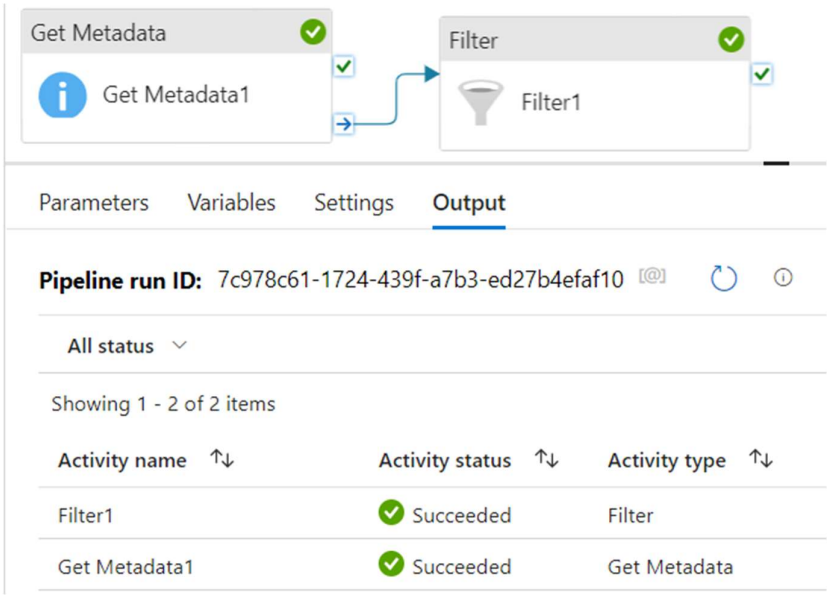
Exp: @activity('Get Metadata1').output.childItems

10. For the Condition give the below expression.

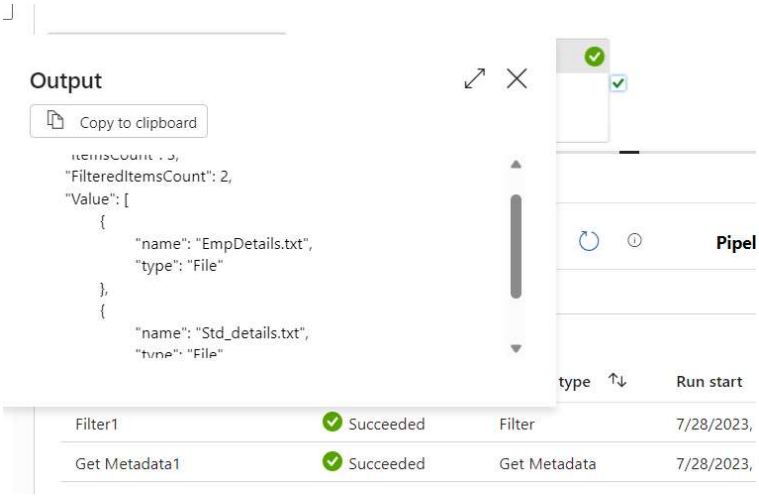
Exp: @endswith(item().name,'txt')



11. Now validate the pipeline and click on debug.
12. Here Our Pipeline was executed successfully.

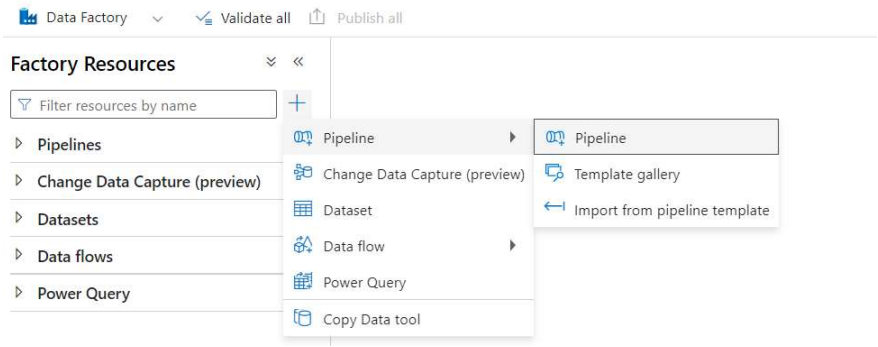


13. Now click on filter output and check the txt file names.



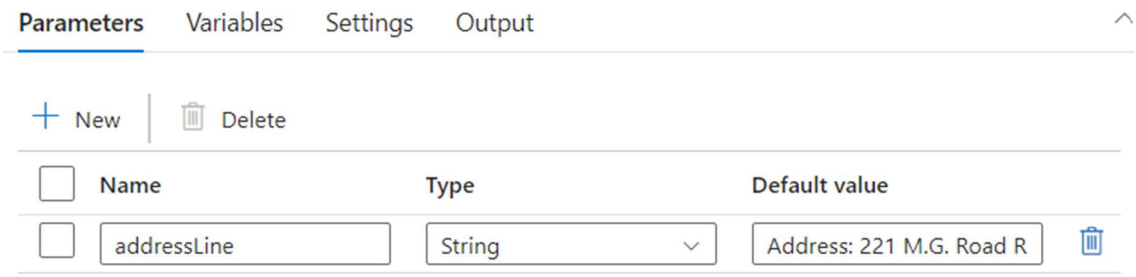
Set Variable Activity

1. Create a Pipeline.



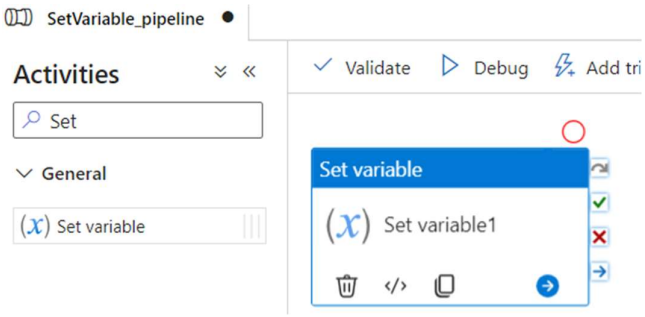
2. Create a parameter with the below value.

Value: Address: 221 M.G. Road Road. City: Kolkata. State: West Bengal



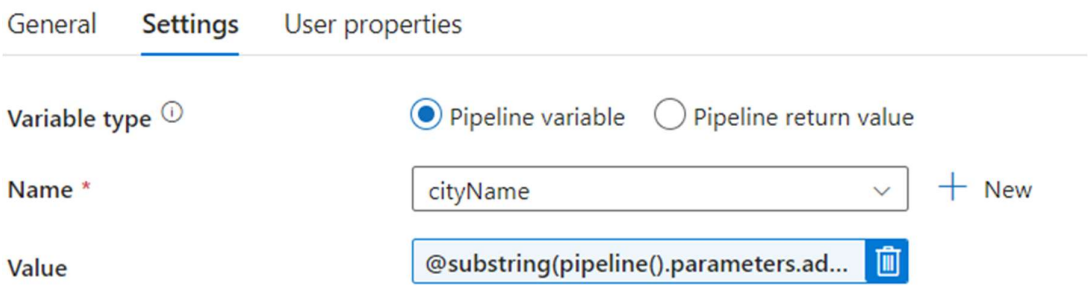
3. Under Variables create a variable as shown below.

4. Drag and drop the set variable.

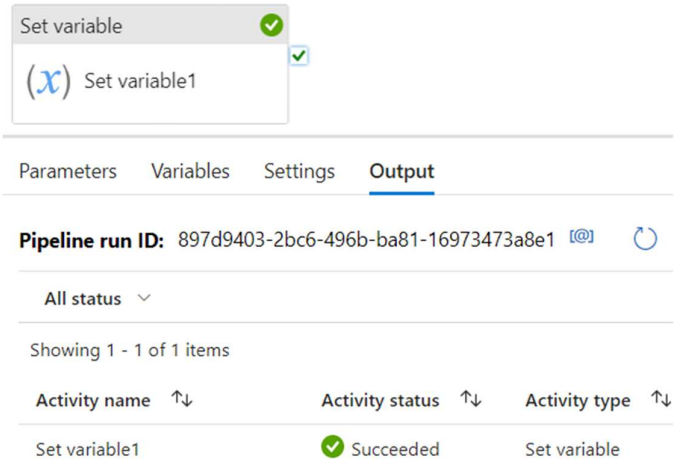


5. Under setting select the name and give the below expression.

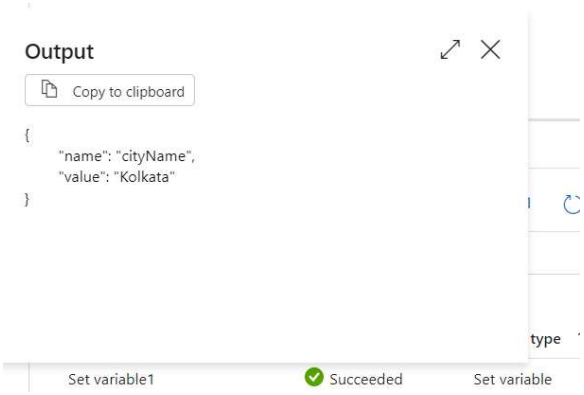
Exp:@substring(pipeline().parameters.addressLine,add(indexof(pipeline().parameters.addressLine, 'City:'), 6),sub(sub(indexof(pipeline().parameters.addressLine, 'State:'), 2),add(indexof(pipeline().parameters.addressLine, 'City:'), 6)))



6. Next validate the pipeline and click on debug.
7. Here our pipeline was executed successfully.



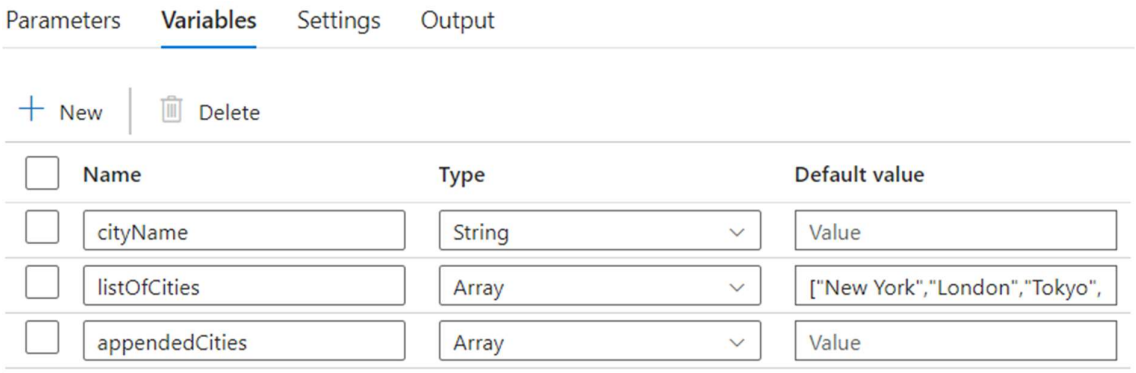
8. In the “Output” tab, the output is displayed as “Kolkata”.



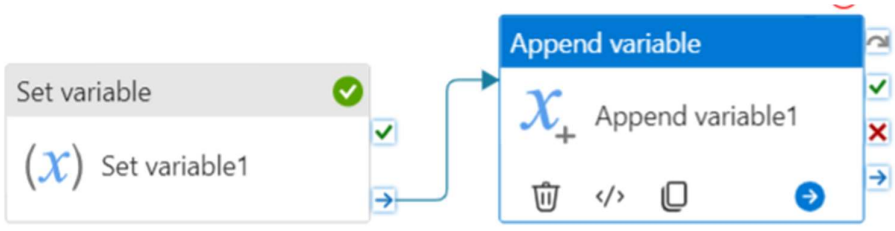
Append Variable

1. In this example we are going to use the same pipeline that we created before for the set variable active.
2. So open the set variable pipeline that we created just before. Under variables add two more variables.
3. For the listofCities give the below value.

Value: ["New York","London","Tokyo","Singapore City"]

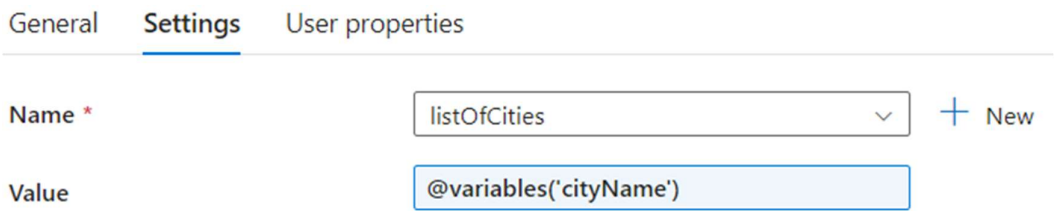


4. Drag and drop the Append Variable Activity and make the connection as shown below.

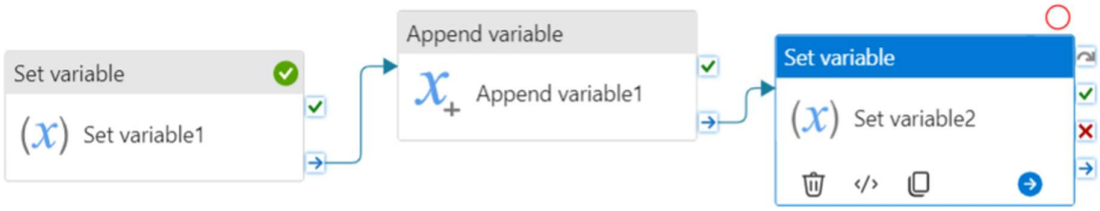


5. Under Settings select the name and give the value.

Value: @variables('cityName')

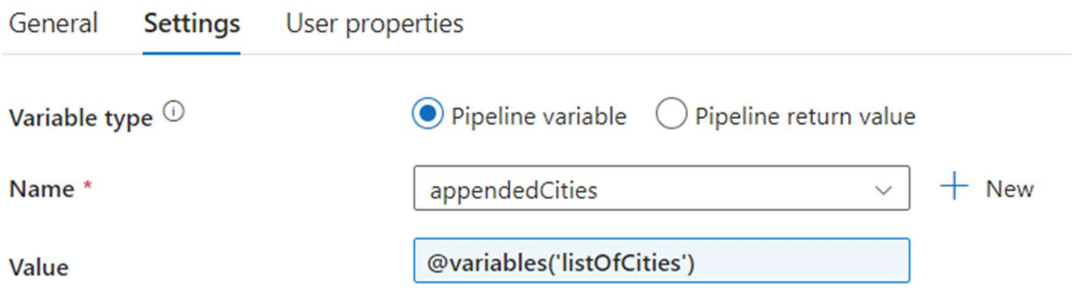


6. Drag and drop the set variable activity and make connections as shown below.

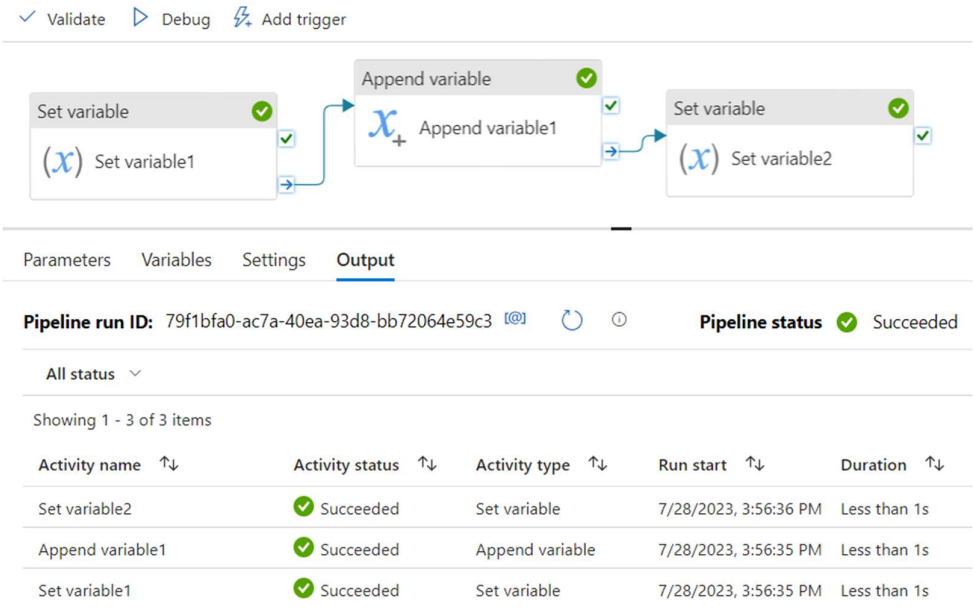


7. Under Settings give the name and value.

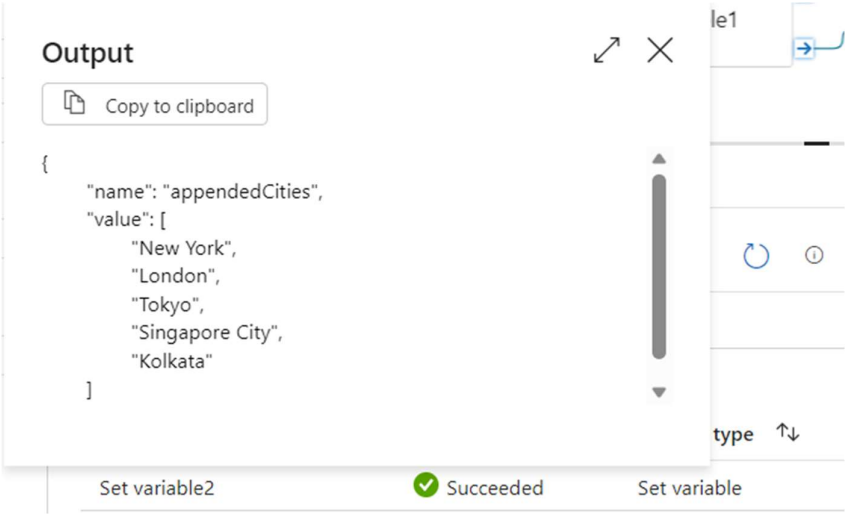
Value: @variables('listOfCities')



8. Next validate the Pipeline and click on Debug.
9. Here our Pipeline was executed successfully.

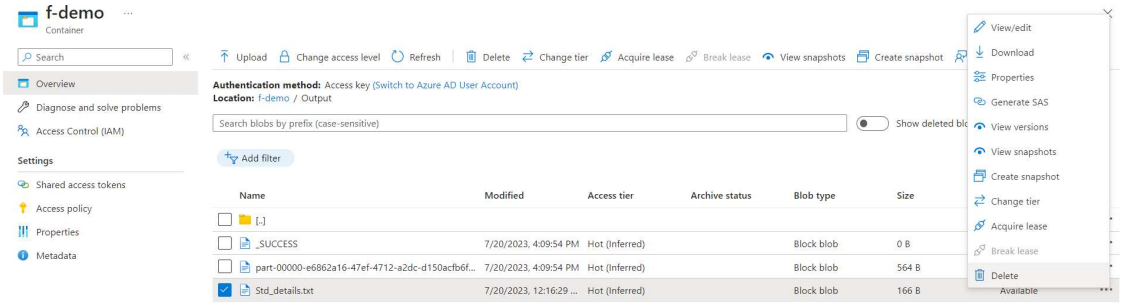


10. In the Output tab, the output is displayed as the value of the cityName Variable, appended to the Variable listOfCities of Array Type.

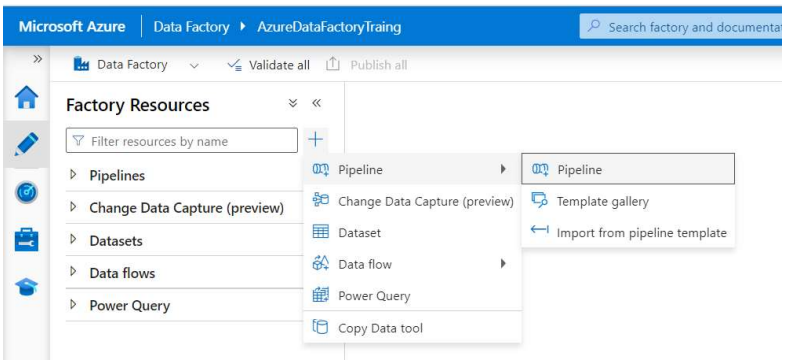


Until Activity

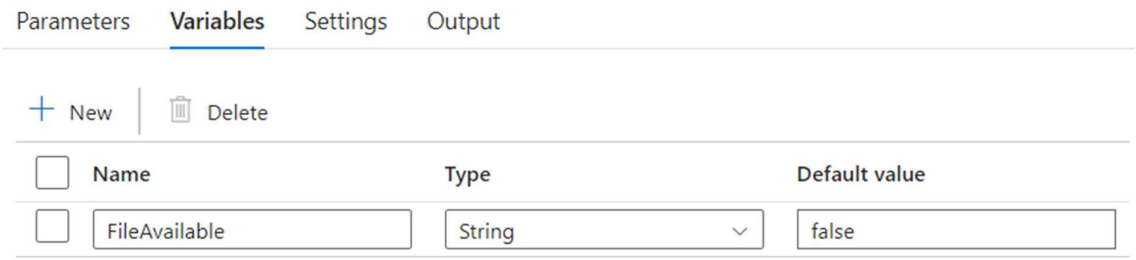
1. Go to Container and go to Output folder then delete the file.



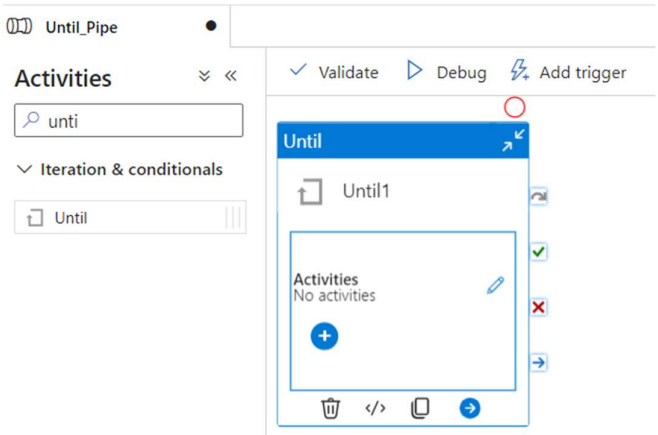
2. Create a Pipeline.



3. Under variable create a variable as shown below.

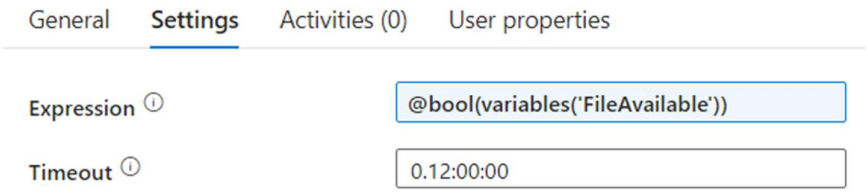


4. Drag and drop the Until activity.

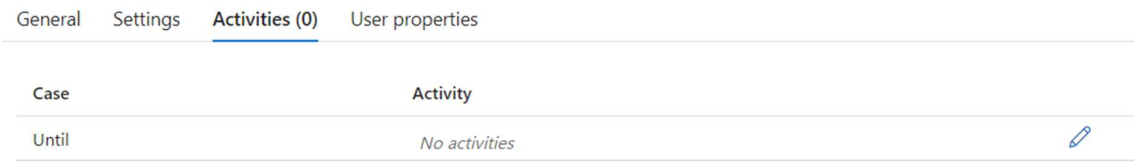


5. Under settings, in the expression give the below expression.

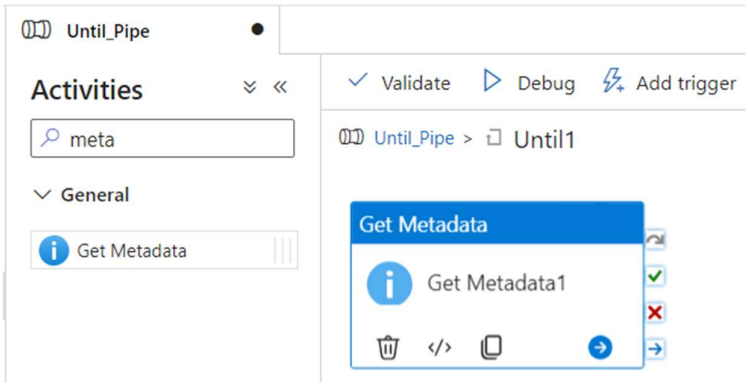
Exp: @bool(variables('FileAvailable'))



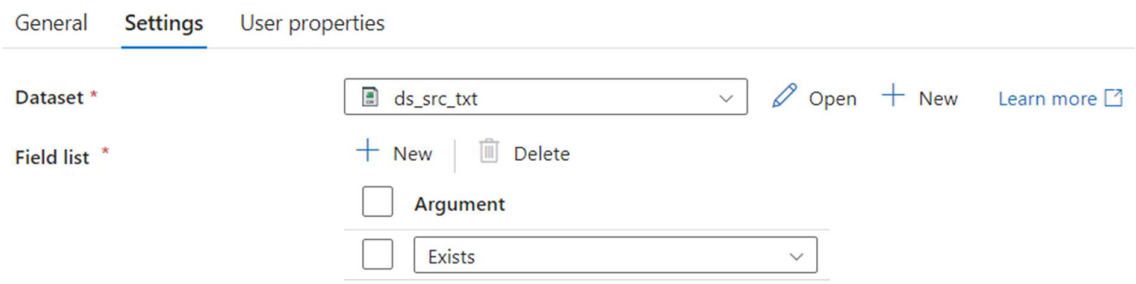
6. Under Activity click on the pencil symbol.



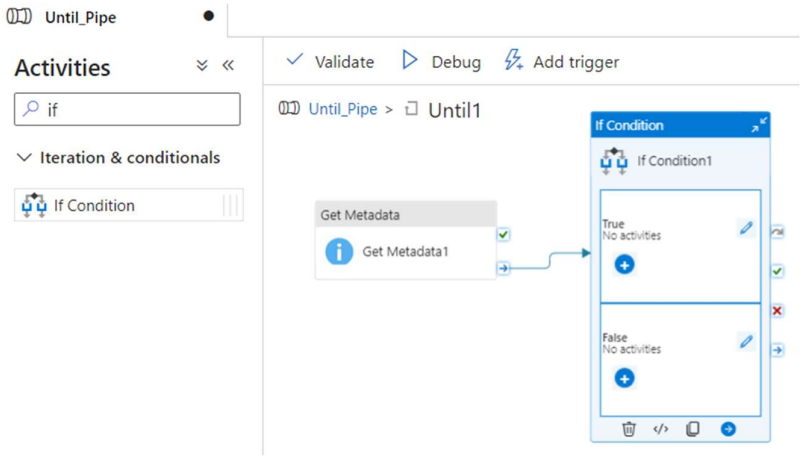
7. Drag and drop the get metadata activity.



8. Under setting select the source dataset. And click on new to add an Argument.

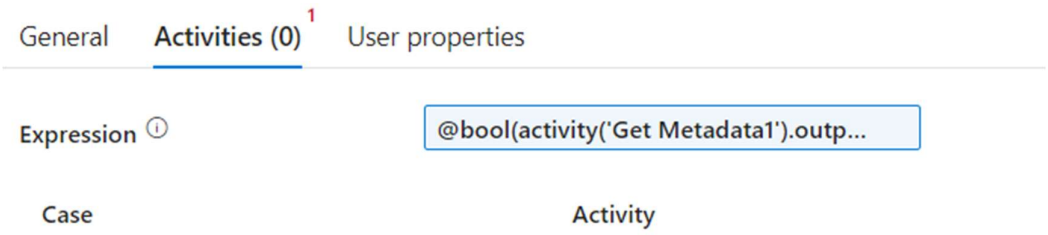


9. Drag and drop the If condition and make the connection as shown below.

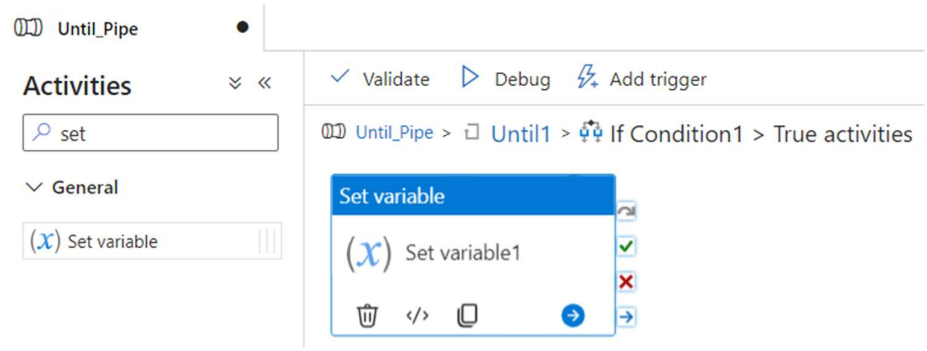


10. Under the activity, in the expression give the below expression.

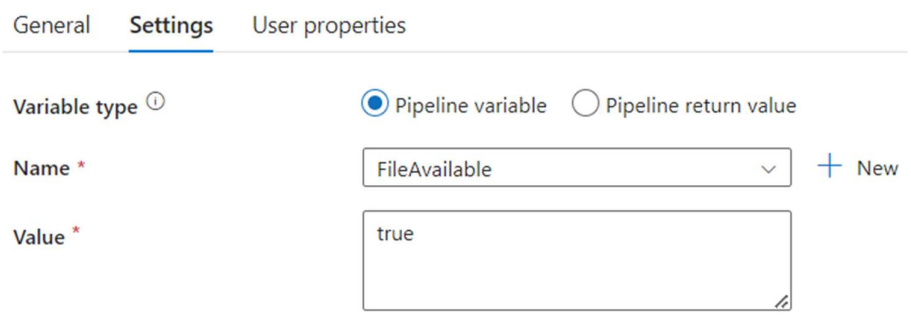
Exp: @bool(activity('Get Metadata1').output.exists)



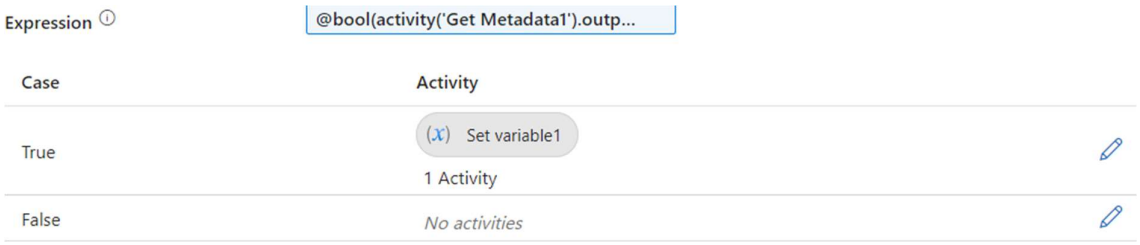
11. Click on the True condition pencil symbol.
12. Drag and drop the set variable activity as shown below.



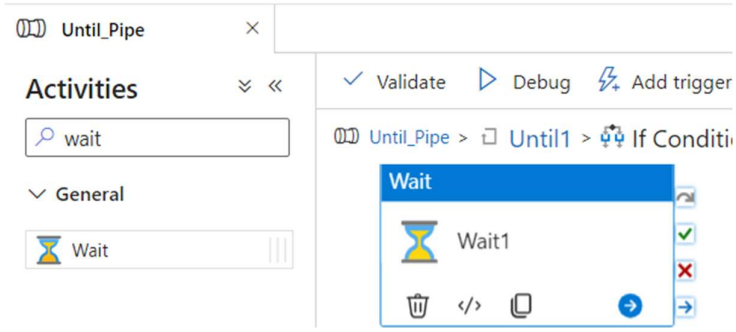
13. Under setting give the Name and value as shown below.



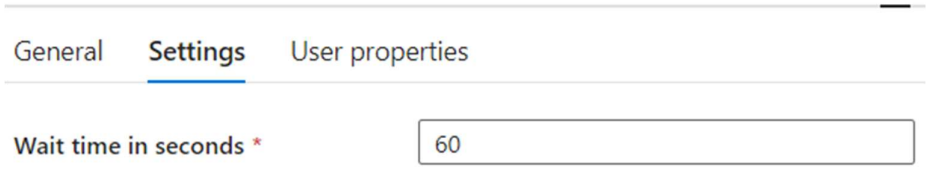
14. Now go back to Until and click on the False activity pencil symbol.



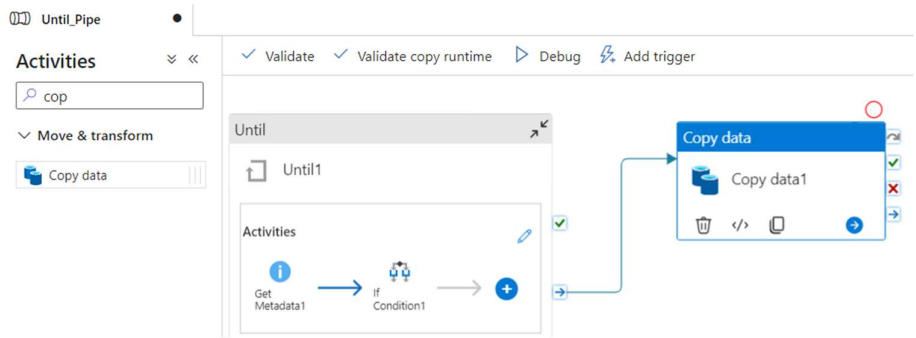
15. Drag and drop the wait activity.



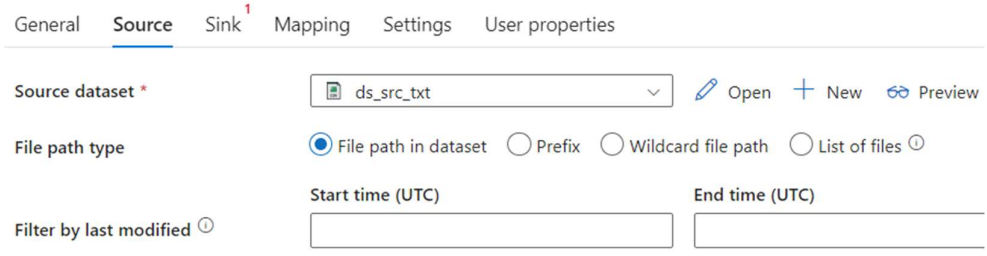
16. Under the settings give the time as 60 sec.



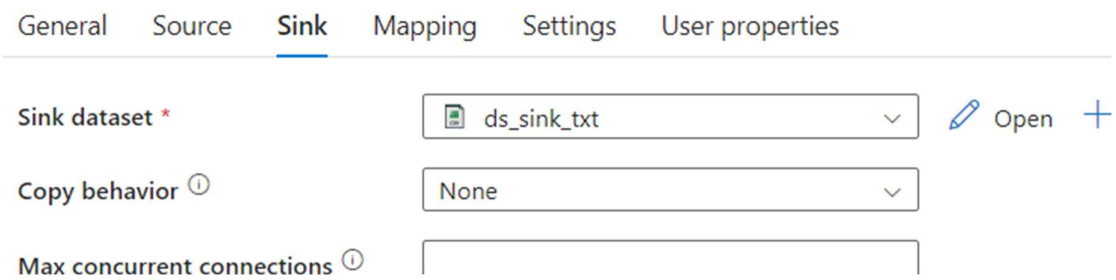
17. Go back to the pipeline and drag and drop the Copy data activity and make the connection.



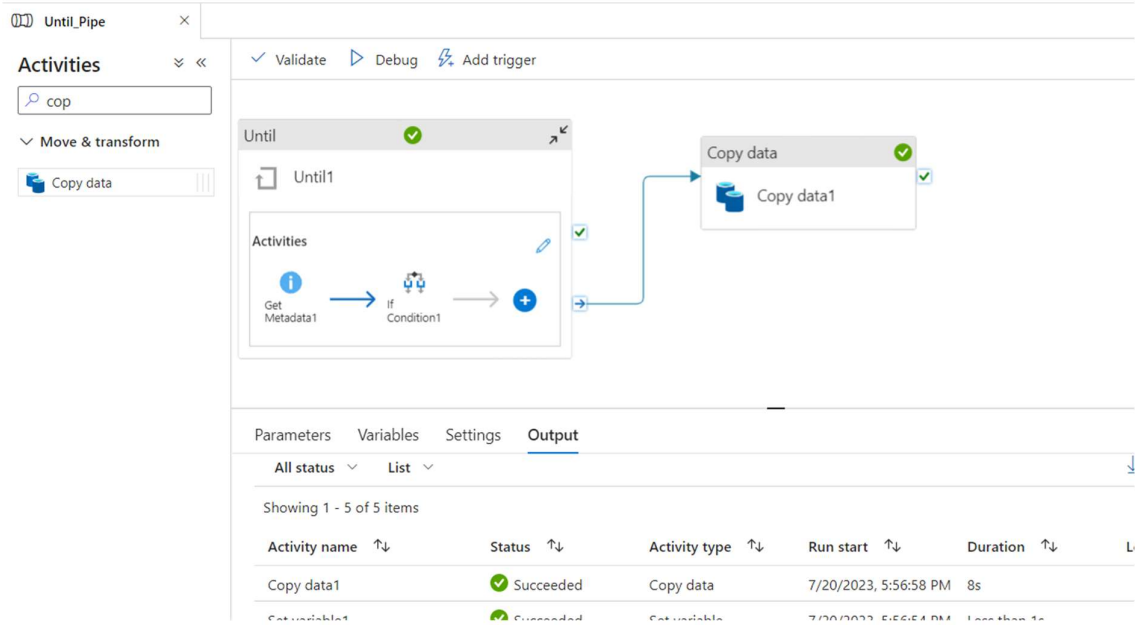
18. Under the Source select the source.



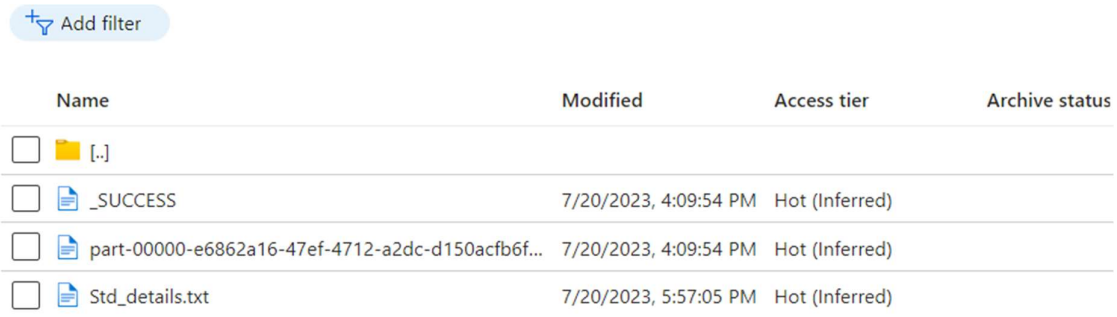
19. Under sink select the destination dataset.



20. Now validate the pipeline and click on Debug. Here our pipeline runs successfully as shown below.

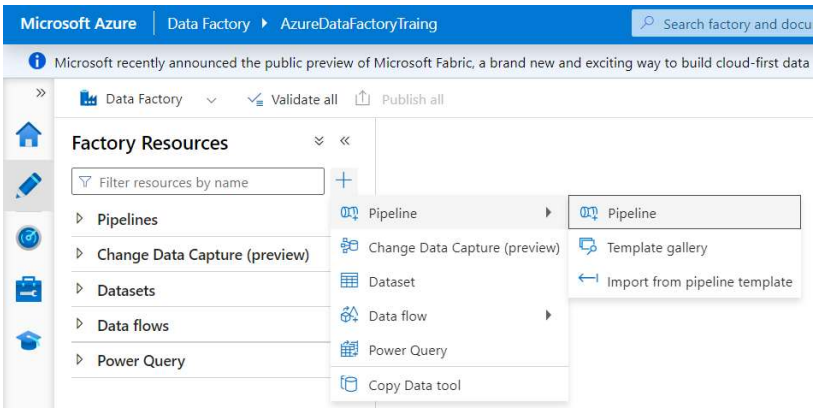


21. So in our case we already have a file in our source so it will directly copy into the target folder.

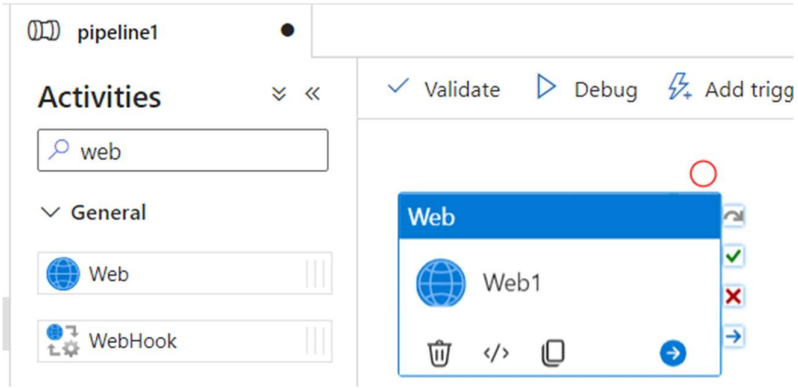


Web Activity

1. Go to Azure Data Factory and go to Author then create a pipeline.

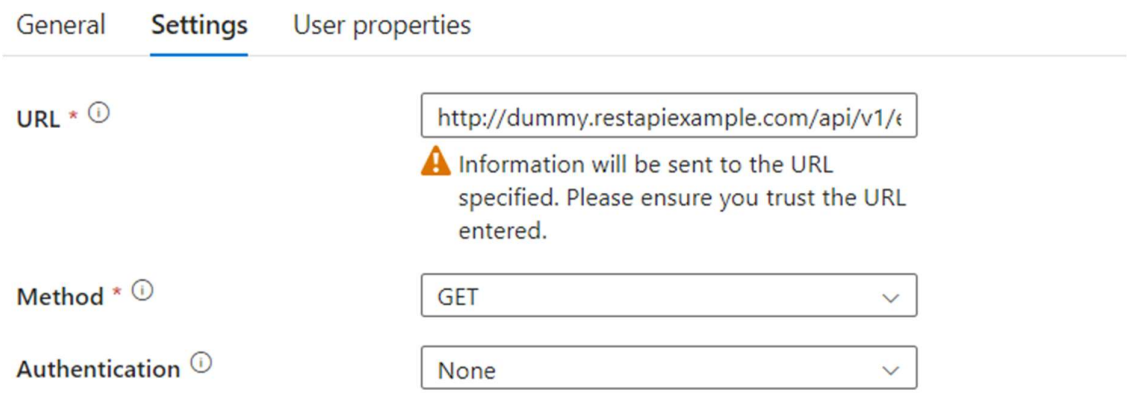


2. Drag and drop the web activity as shown below.

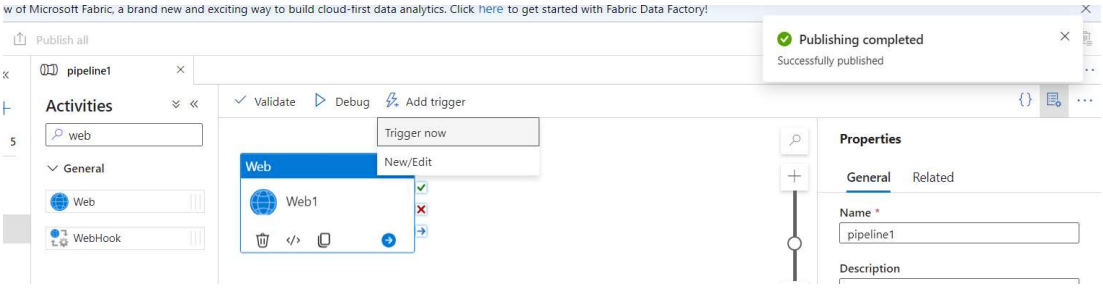


3. Under setting give the URL and set the method.

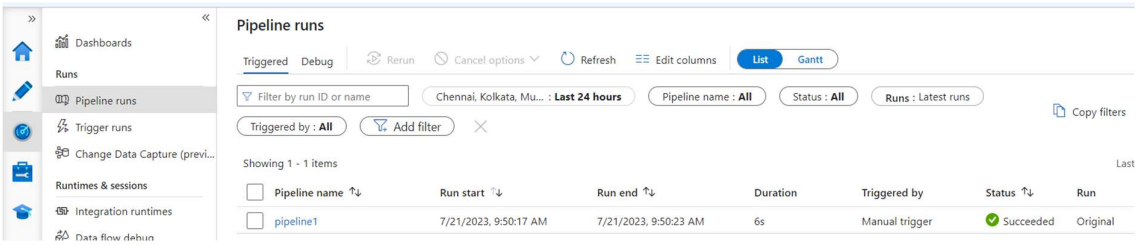
URL: http://dummy.restapiexample.com/api/v1/employees



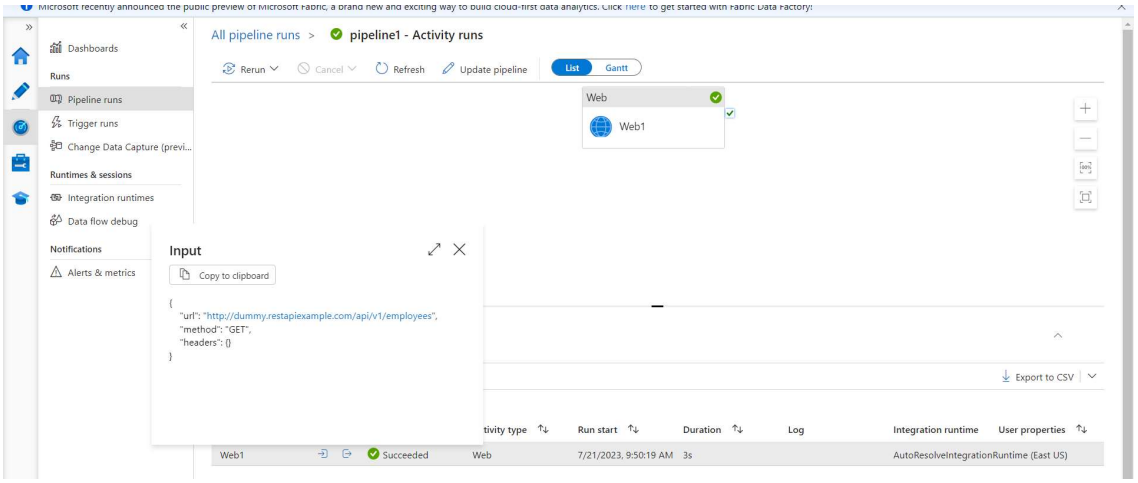
4. Publish the pipeline and click on Add trigger then click on Trigger now.



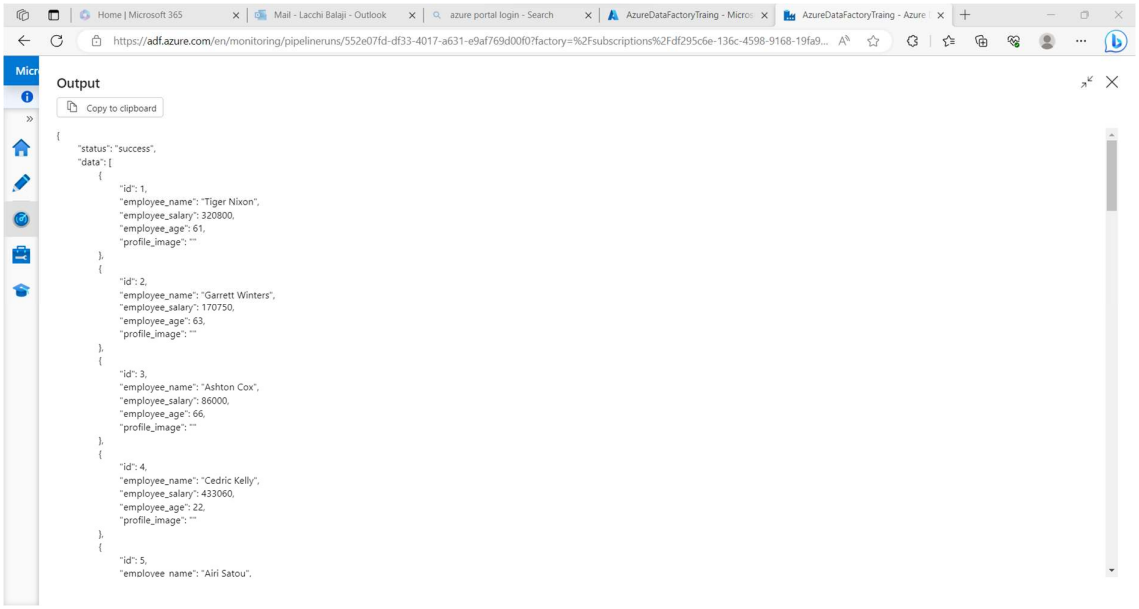
5. Next click Ok.
6. Now go to the Monitor tab and you can see our pipeline status is succeeded.



7. Next click on the Pipeline. Inside that click on Input you will see below input.

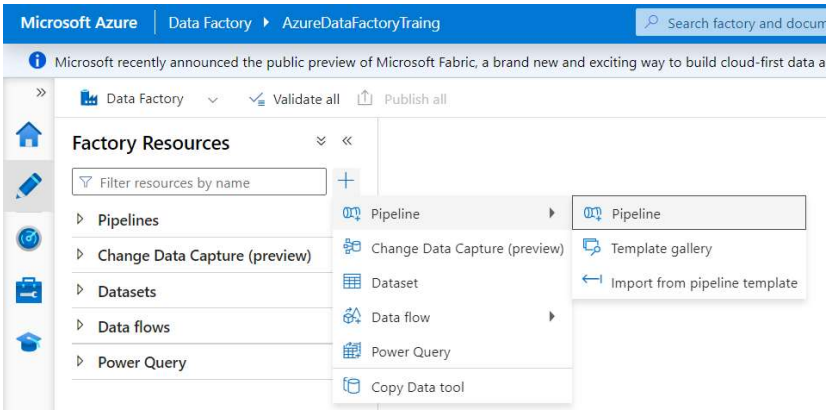


8. Now if you click on Output you will see an output as shown below.

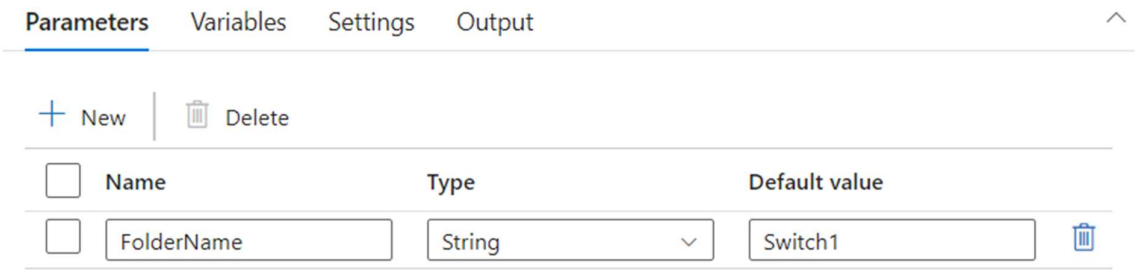


Switch Activity

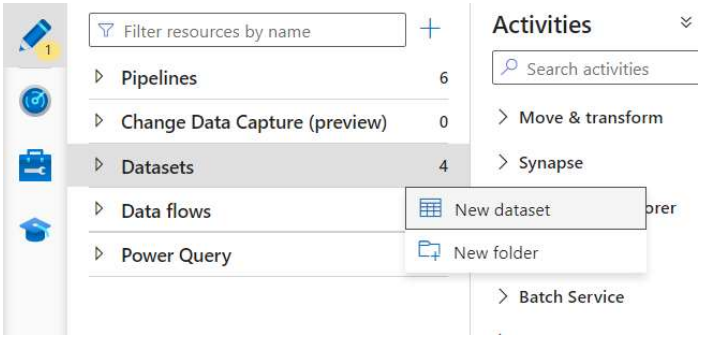
1. Create a pipeline.



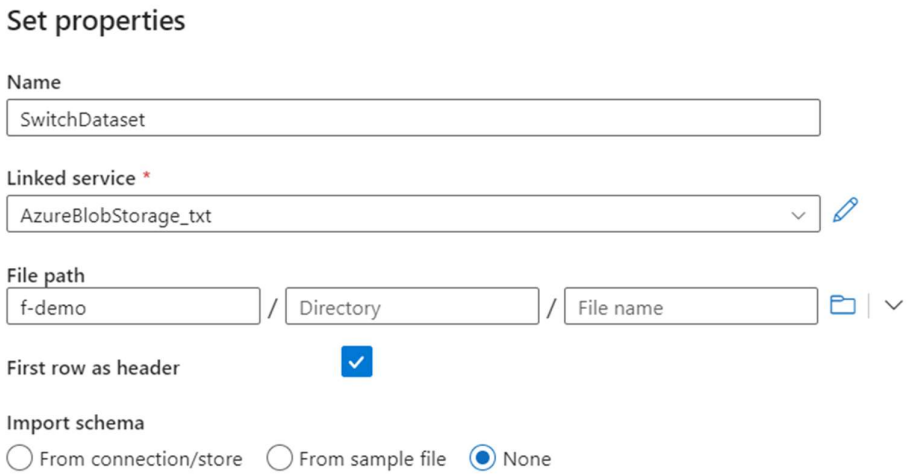
2. Under the parameter create the below parameter.



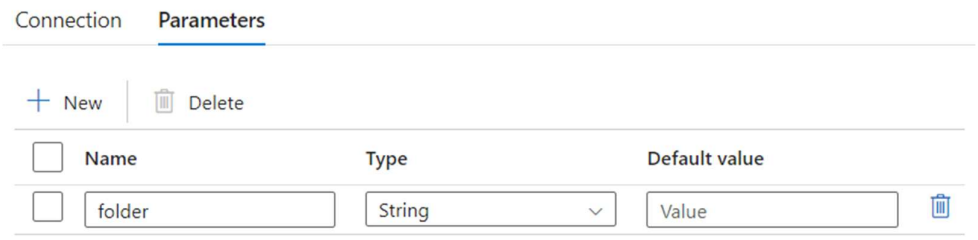
3. Create a Dataset.
4. Select the Azure Blob Storage and click on Continue. Then select the Binary and click on continue.



5. In the properties give the name, linked service, and the container name as shown below.

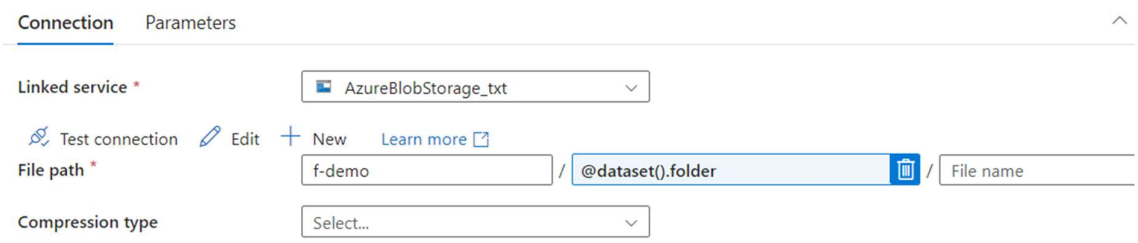


6. Now under parameter create a below parameter.

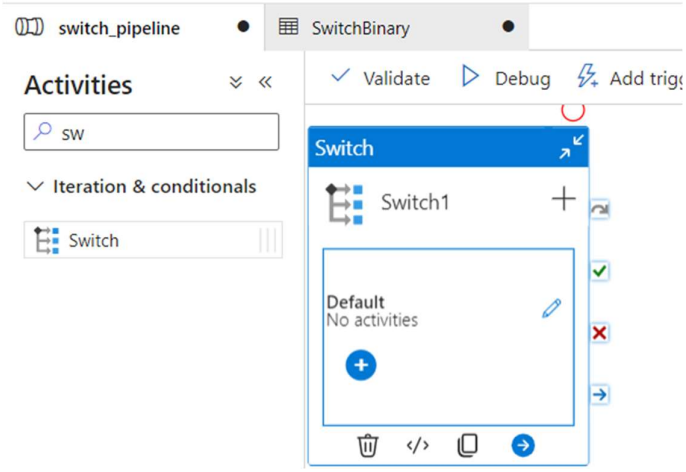


7. Under connection, in the directory give the below expression as shown below.

Exp: @dataset().folder

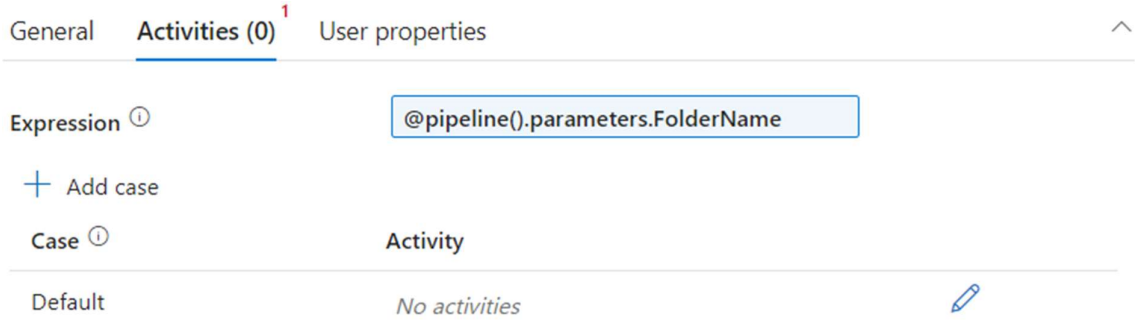


8. Go to Pipeline drag and drop the switch activity.

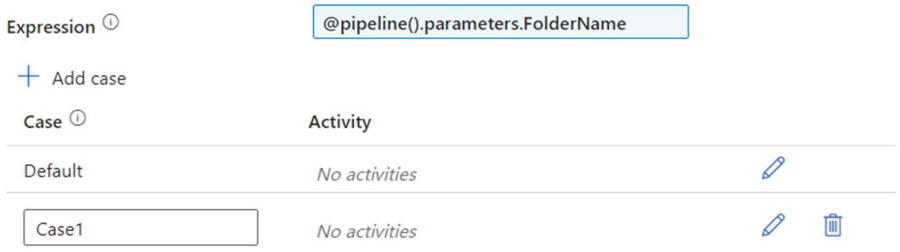


9. Under Activity give the below expression.

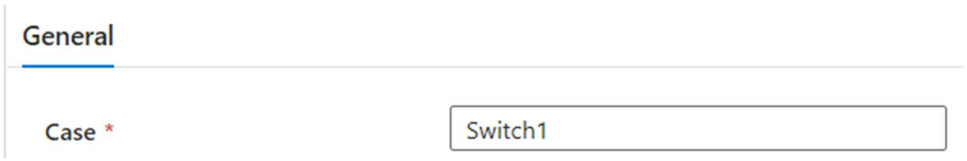
Exp: @pipeline().parameters.FolderName



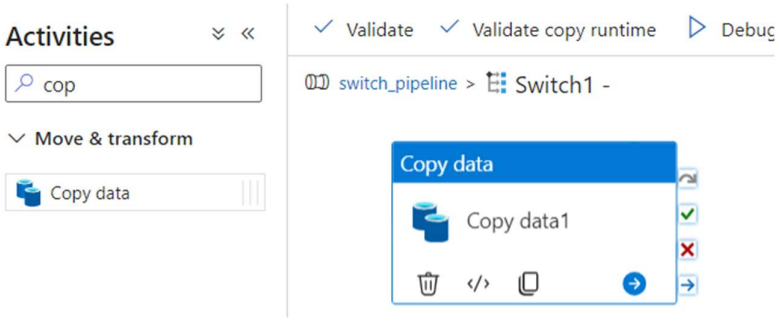
10. Now click on the plus symbol to add a case. Then click on the pencil symbol.



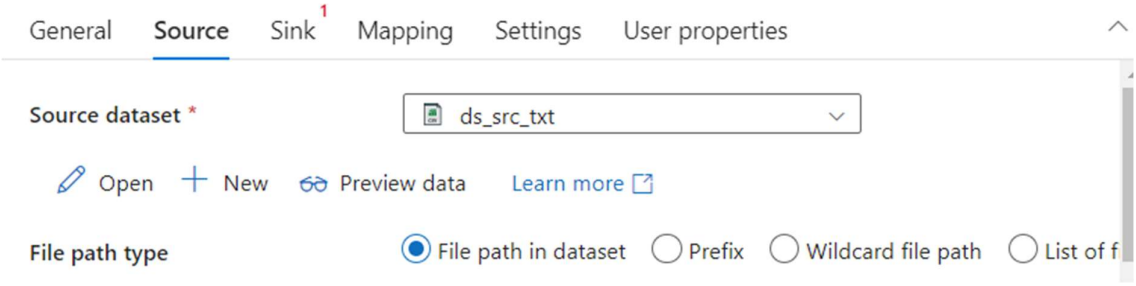
11. Give the case name Switch1 as shown below.



12. Drag and drop the copy data activity as shown below.

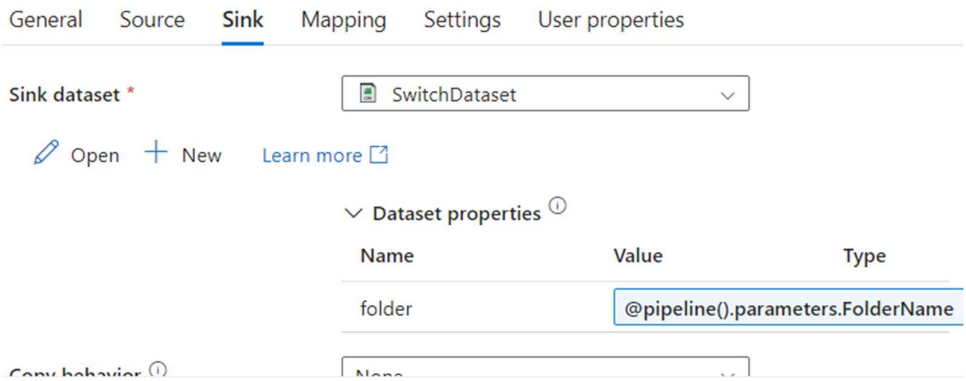


13. Under the Source select the source dataset that we created in the day 1 session.

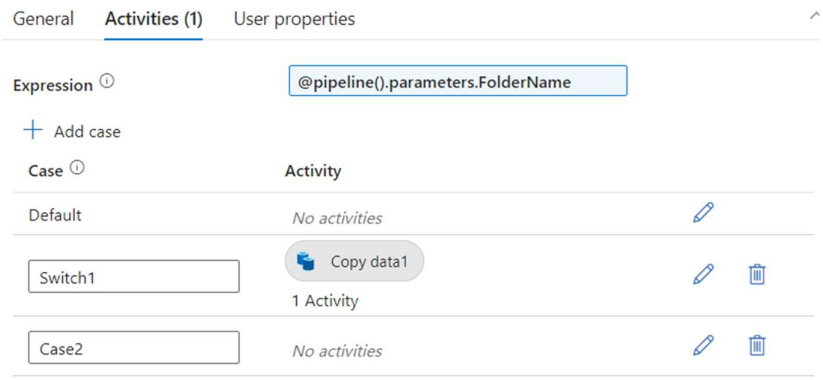


14. Under the sink select the dataset that we created before and under the folder name give the below expression.

Exp: @pipeline().parameters.FolderName

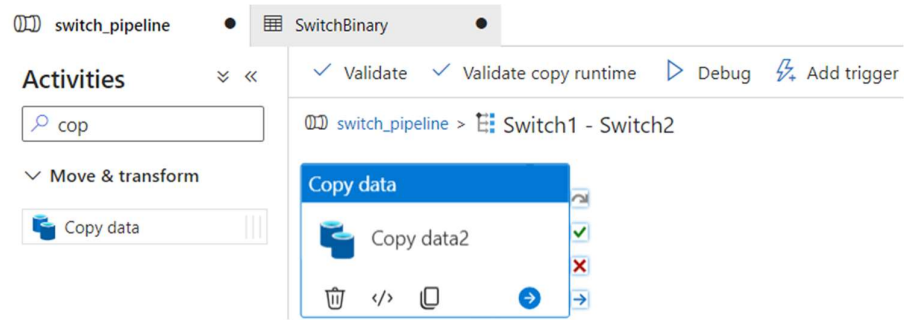


15. Now go to pipeline and add another case.

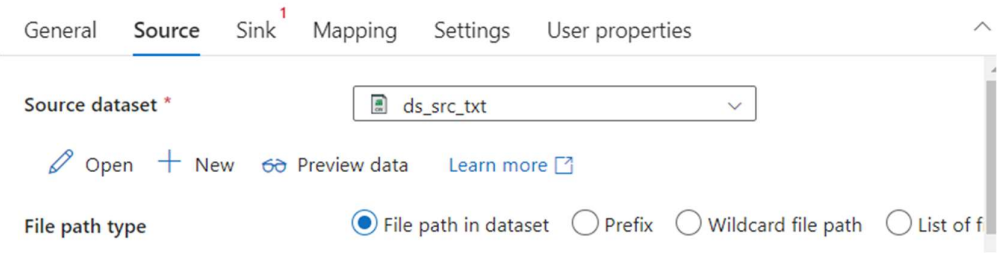


16. Name it Switch2.

17. Drag and drop the copy data activity as shown below.

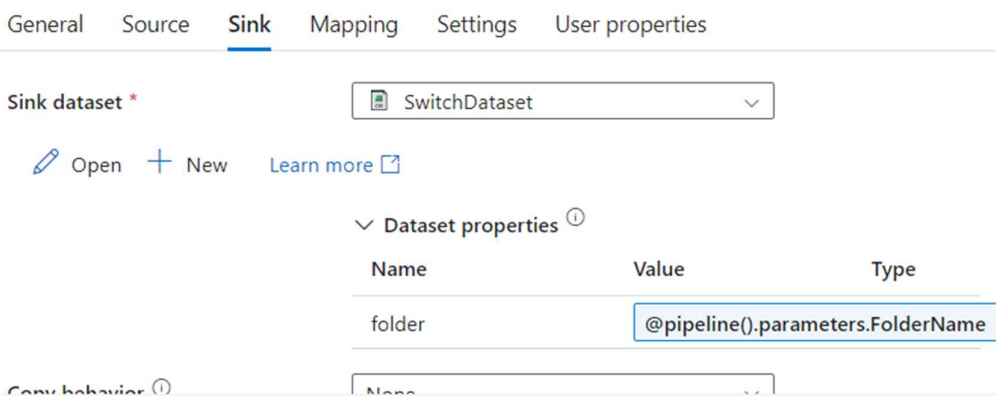


18. Under the Source select the source dataset that we created in the day 1 session.

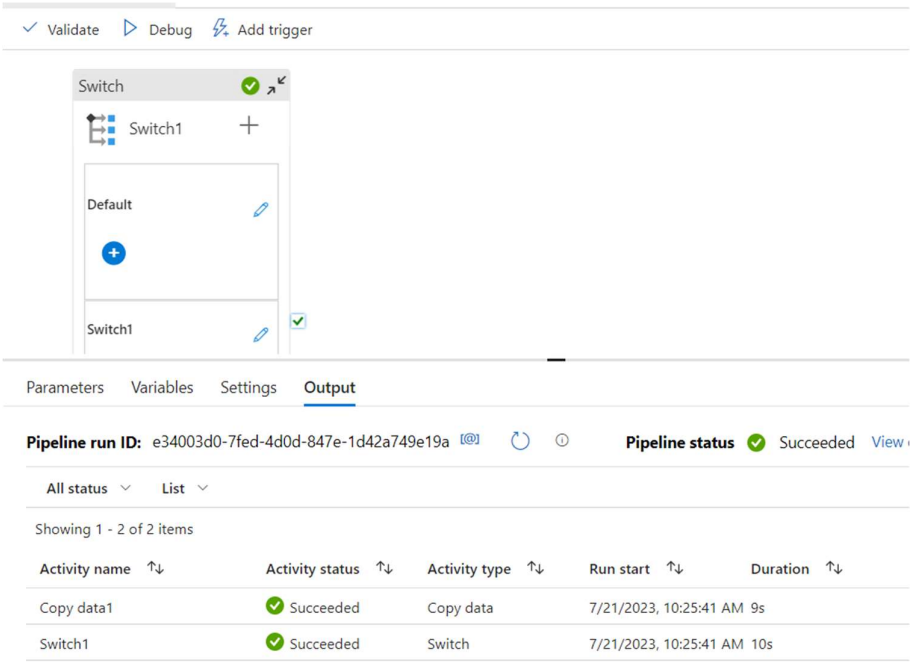


19. Under the sink select the dataset that we created before and under the folder name give the below expression.

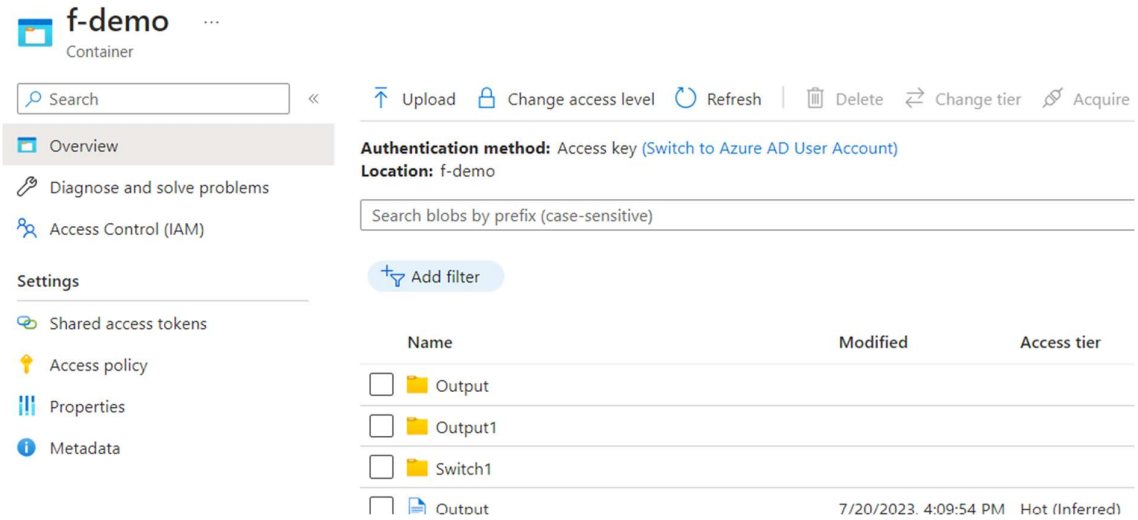
Exp: @pipeline().parameters.FolderName



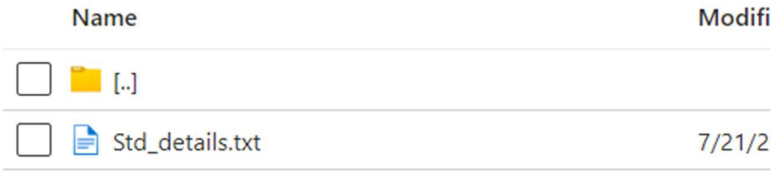
20. Next go to Pipeline and validate it and click on debug.
21. Here our pipeline executes successfully.



22. As initially give the parameter value as switch1 so if you go to your container you will see it as a Switch1 folder.

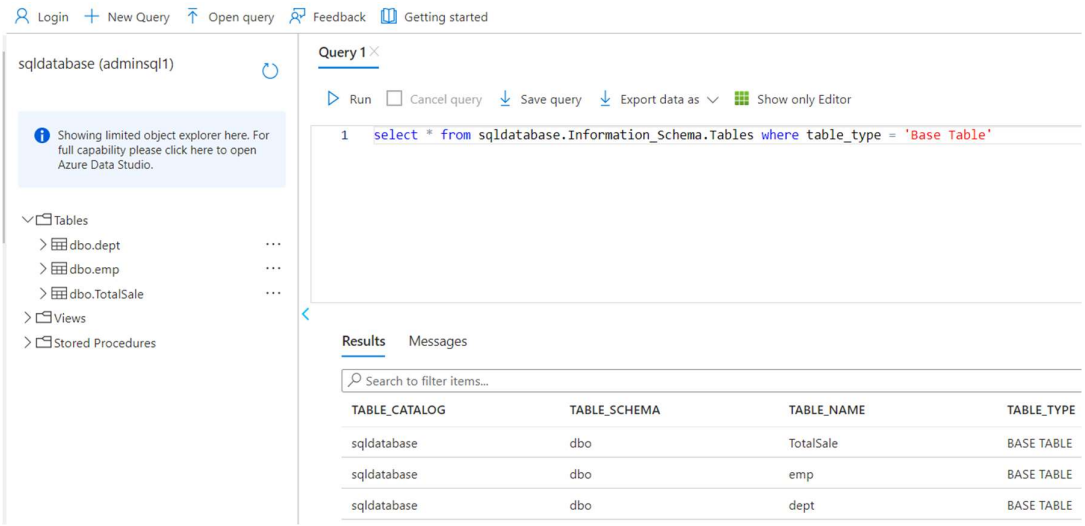


23. Inside the folder the file is copied from source to destination.

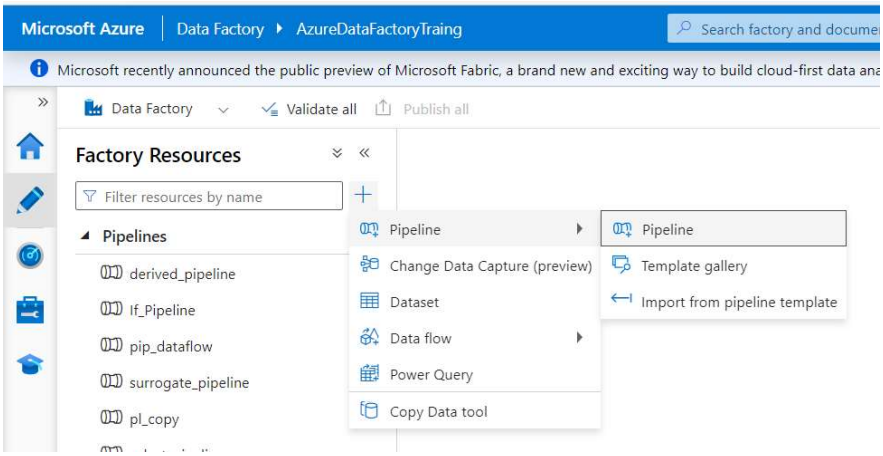


Lookup Activity

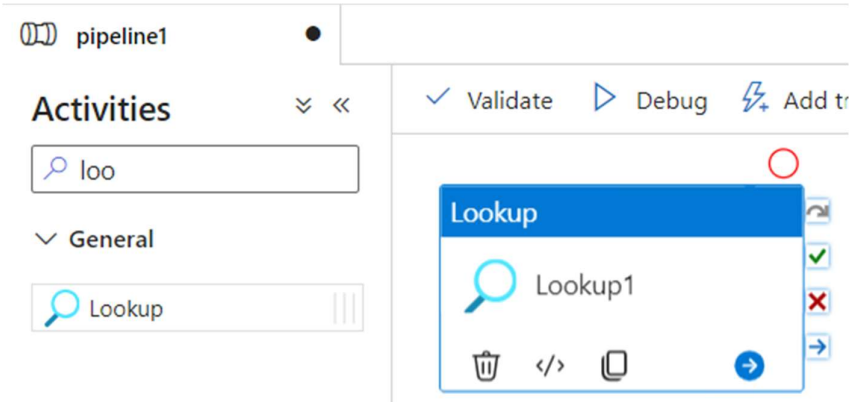
1. Open the SQL database that we created in the previous session.
2. Here we have three tables. Here dynamically we will store these tables in the Azure Blob storage.



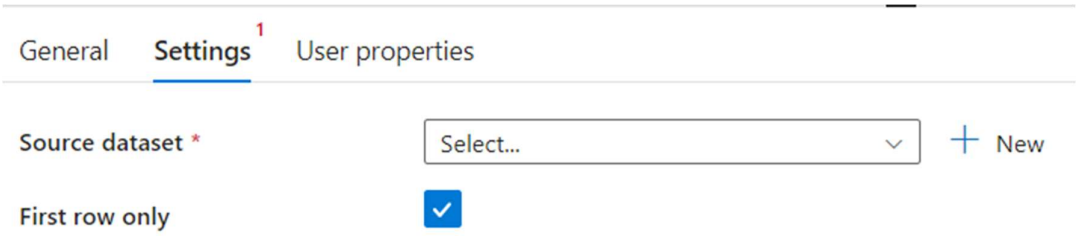
3. Create a Pipeline.



4. Drag and drop the lookup activity as shown below.



5. Under settings click on New.

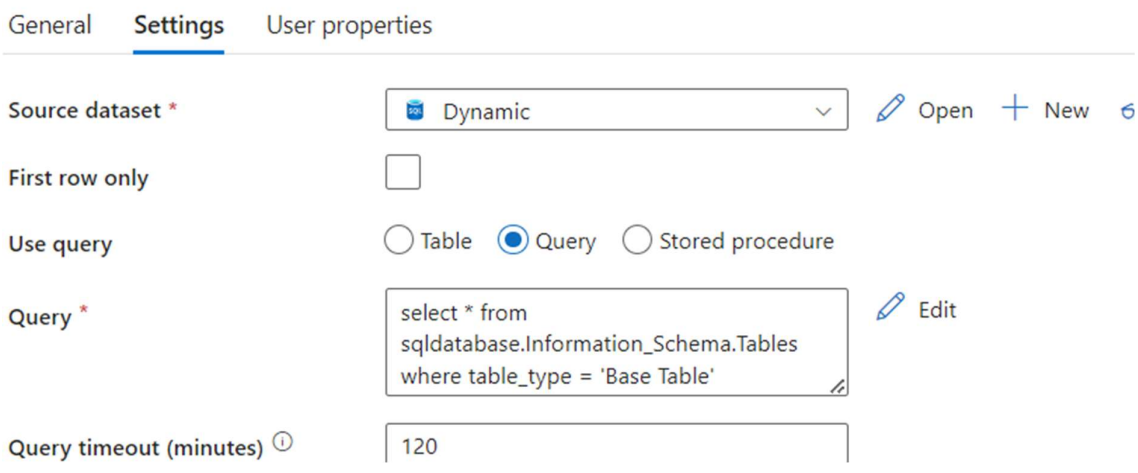


6. Select the Azure SQL Database and click continue.
7. Just give the name and select the linked service and click ok.

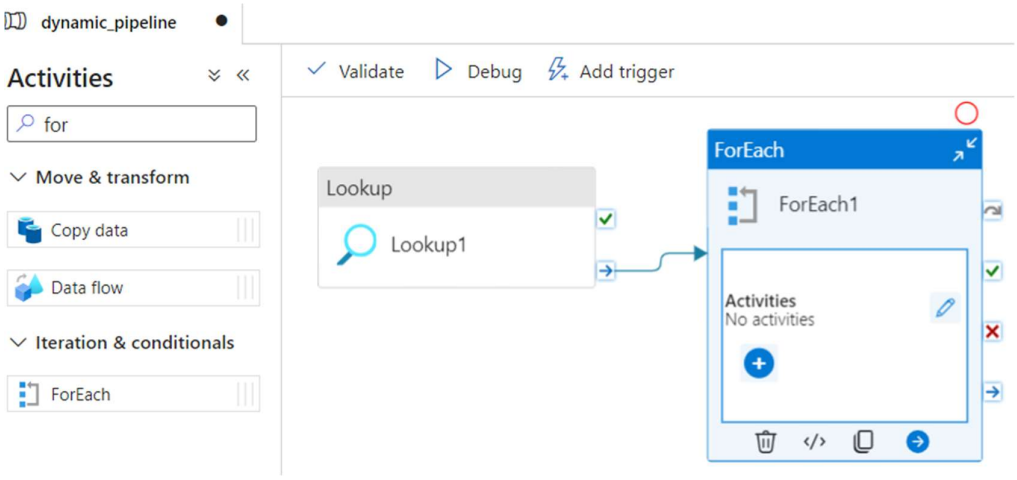
8. Next check the query option and give the below query.

Query: select * from sqldatabase.Information_Schema.Tables where table_type = 'Base Table'

9. Uncheck the First row-only option.

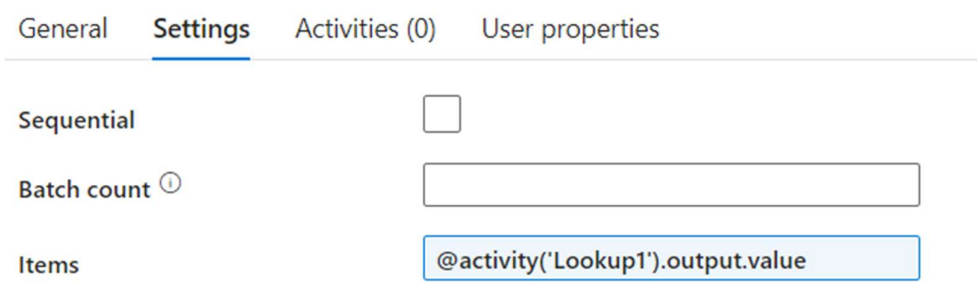


10. Drag and drop the foreach activity and make the connection as shown below.

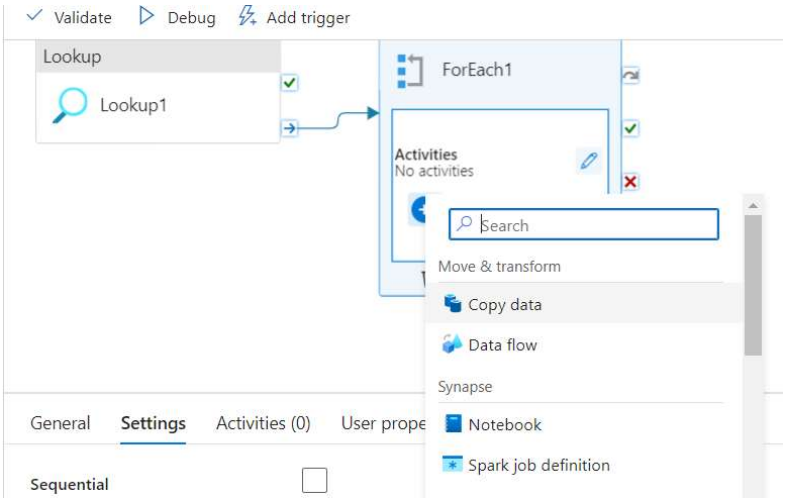


11. Under setting in the Items give the below expression as shown below.

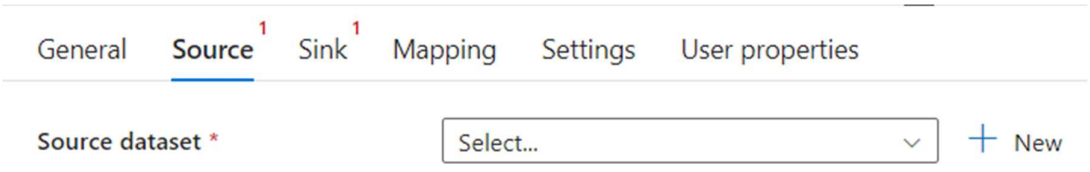
Exp: @activity('Lookup1').output.value



12. Click on the plus symbol in the foreach loop and click on copy data option.



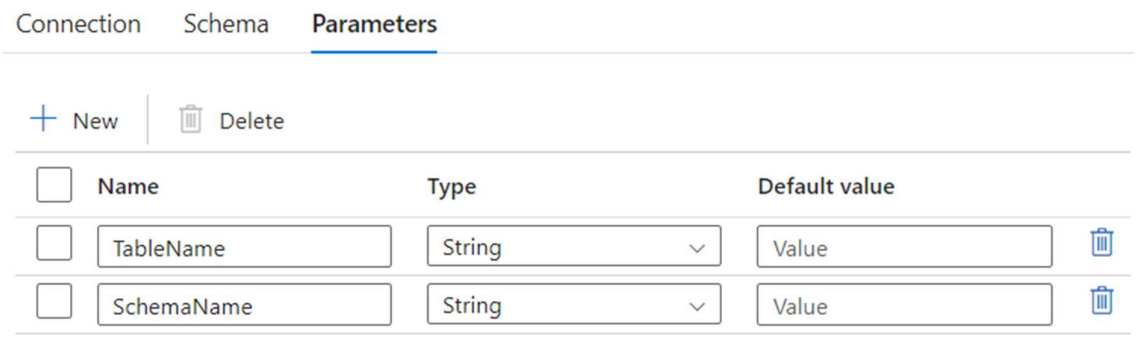
13. Under the source click on New.



14. Select the Azure SQL Database and click on continue.
15. Then give the name and linked service then click on Ok.

16. Click on Open.

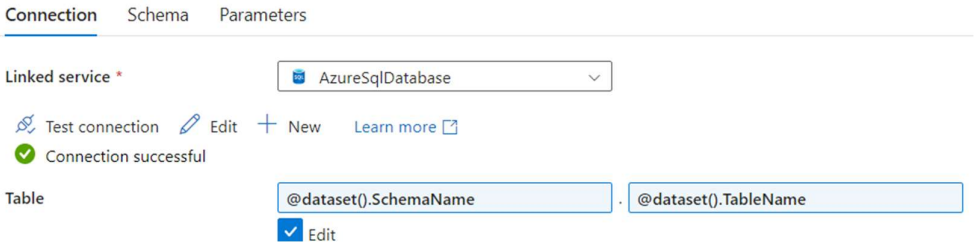
17. Under parameters add two parameters as shown below.



18. Now go to connections and check the edit option and give the expression as shown below.

Exp: @dataset().SchemaName

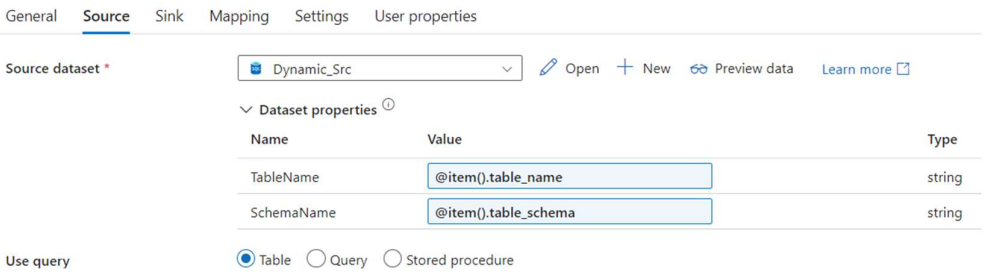
Exp: @dataset().TableName



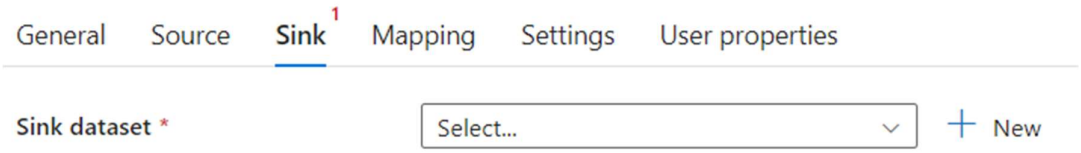
19. Under source for the TableName and SchemaName give the below expression.

Exp: @item().table_name

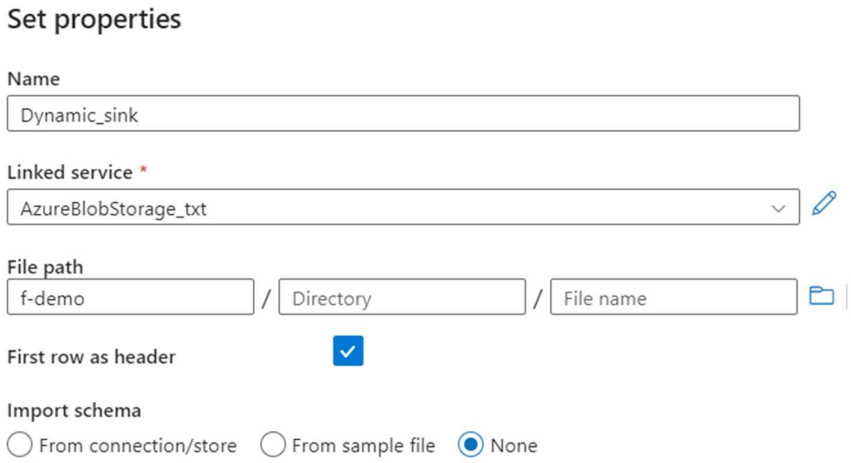
Exp: @item().table_schema



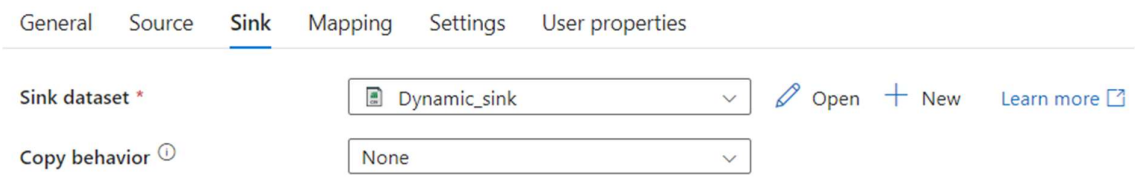
20. Under sink click on new.



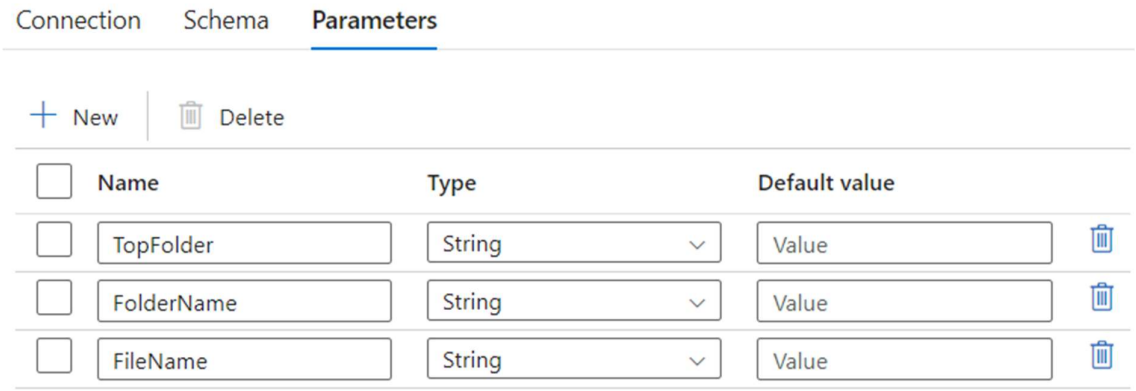
21. Select Azure Blob Storage and click on continue. Then click on delimited text and click on continue.
22. Give the Name, Linked service, and Container name.



23. Click on Open.



24. Under parameter create below three parameters.

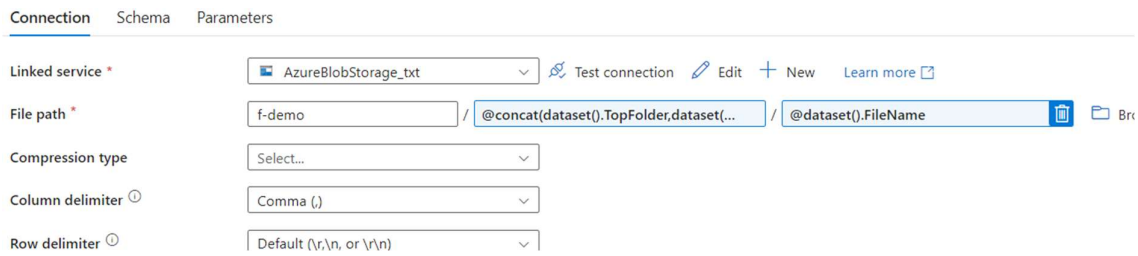


25. Under connection, in the directory give the below expression.

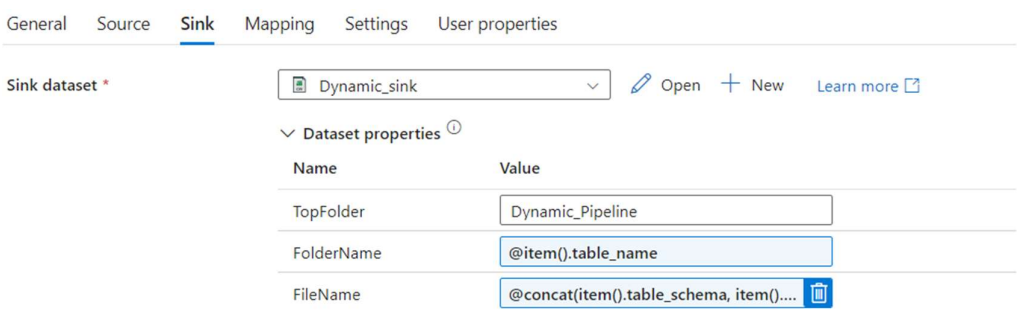
Exp: @concat(dataset().TopFolder,dataset().FolderName)

26. And in the File Name give the below expression.

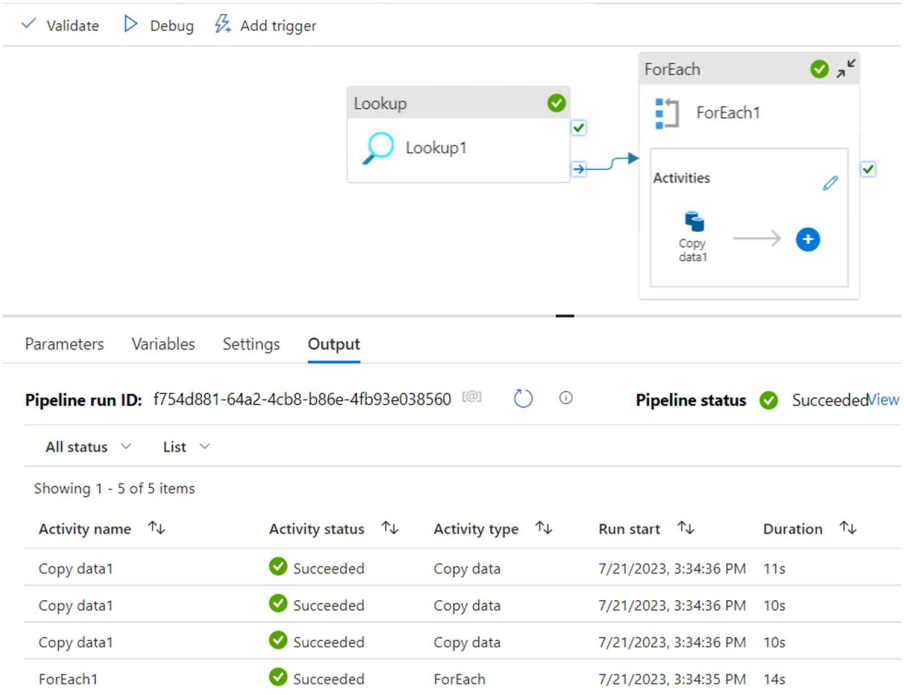
Exp: @dataset().FileName



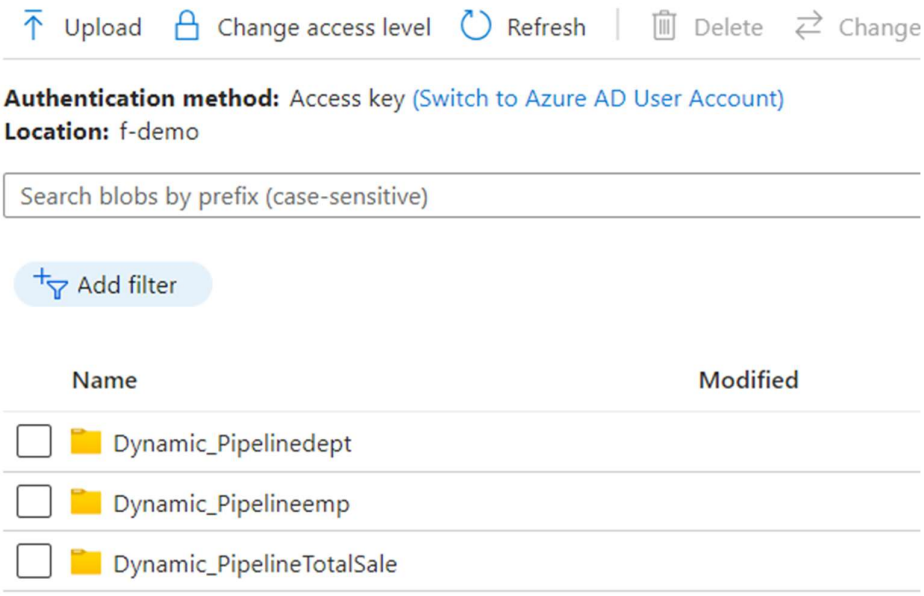
27. Now go to the pipeline under the sink for the top folder and give Dynamic_Pipeline.
28. For the Folder Name give this expression: @item().table_name
29. For the File Name give this expression: @concat(item().table_schema, item().table_name)



30. Now validate the pipeline and click on debug.
31. Our pipeline was executed successfully.

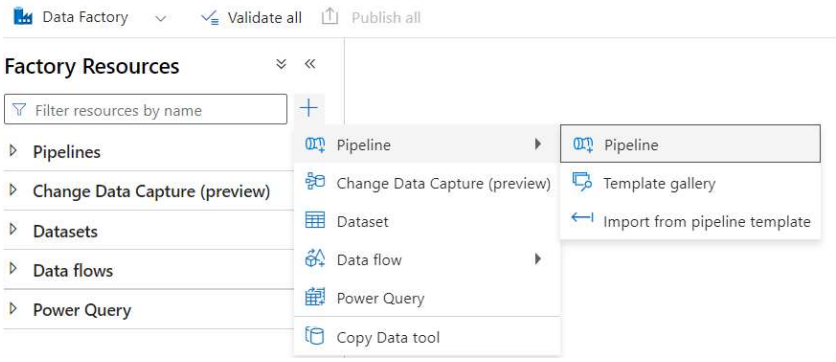


32. Now go to container and you will see three folders as shown below. Each folder has a table.

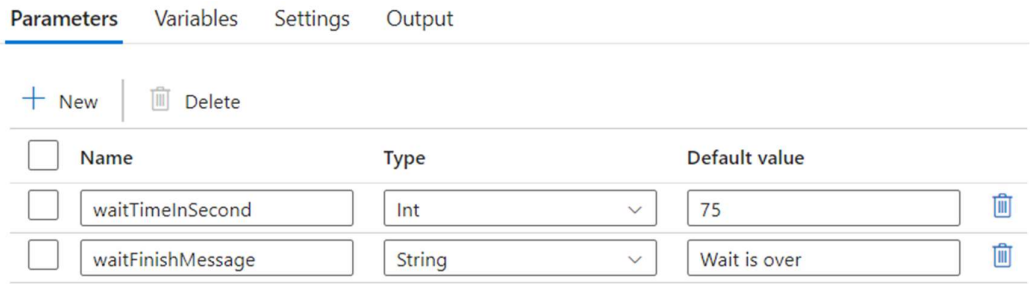


Wait Activity

1. Create a Pipeline.

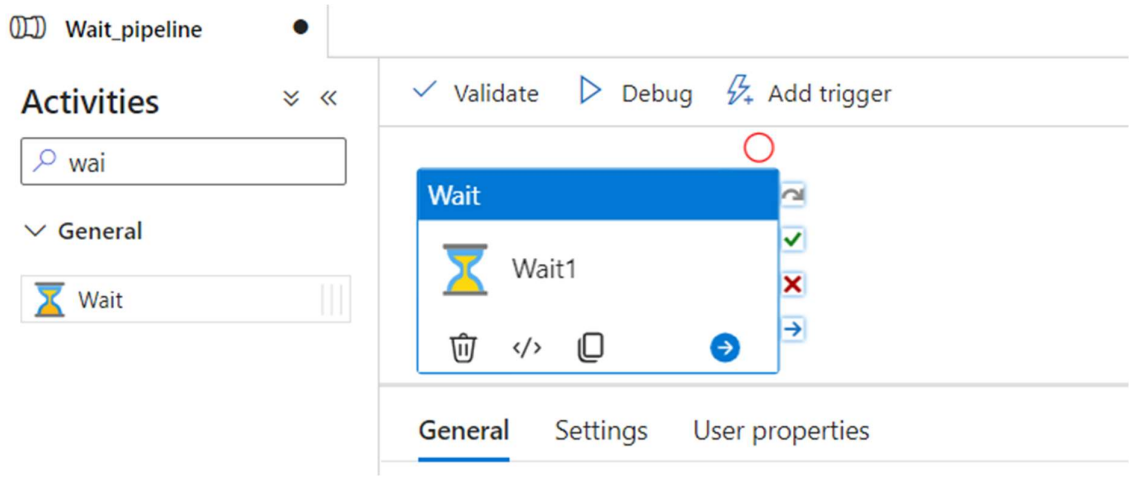


2. Create Parameters as shown below.



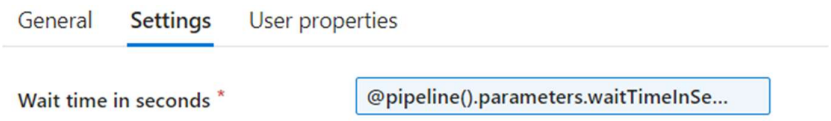
3. Create a Variable as shown below.

4. Drag and drop the wait as shown below.

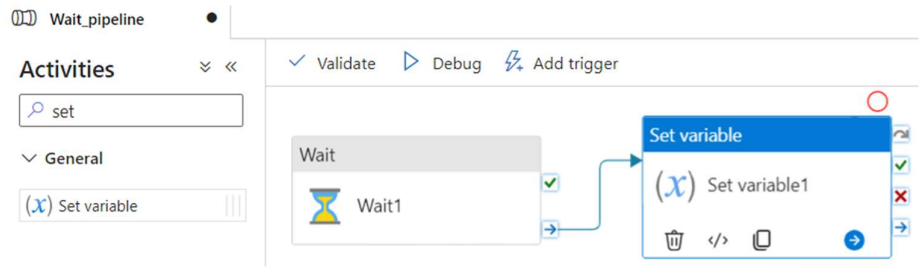


5. Under settings give the below expression.

Exp: @pipeline().parameters.waitTimeInSecond

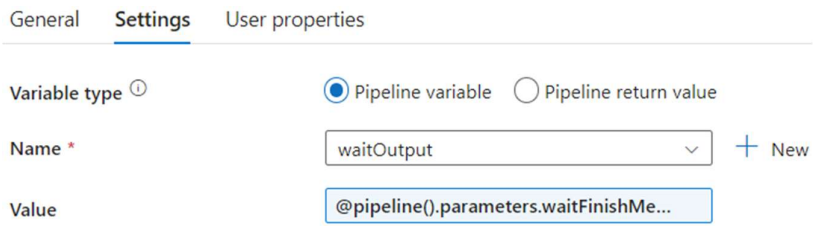


6. Drag and drop the Set Variable and make connections as shown below.

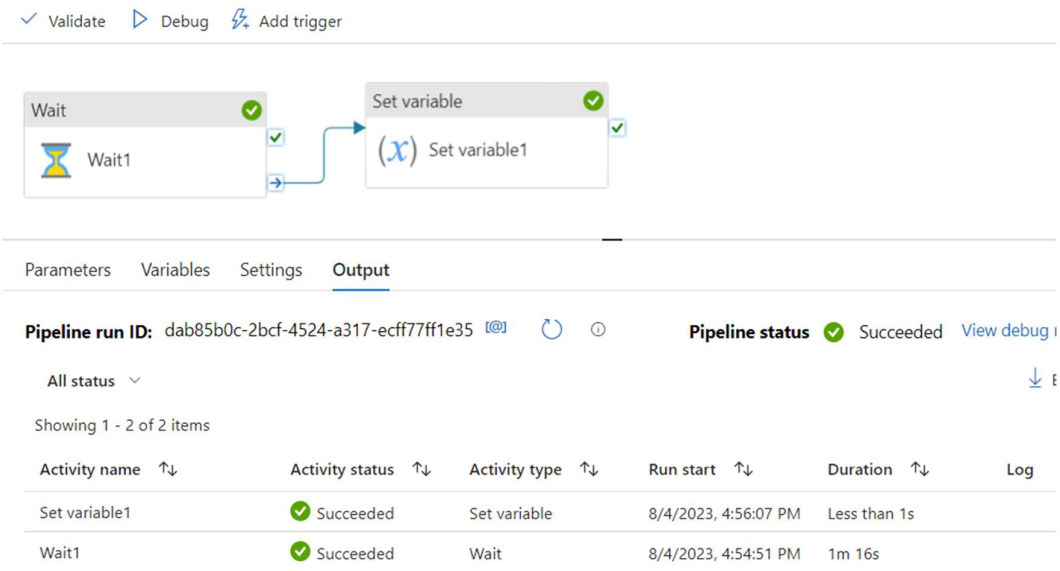


7. Under settings, select the name and give the below expression.

Exp: @pipeline().parameters.waitFinishMessage



8. Now validate the pipeline and click on Debug.
9. Our Pipeline was executed successfully.
10. It can be seen that the “Wait” Activity executed for 75 seconds, i.e., 1 minute and 15 seconds.
11. The “Wait” Activity transferred control to the “Set variable” Activity on the 76th second, i.e., when the time spent by the “Wait” Activity was 1 minute 16 seconds.



Script Activity

1. Create below tables as shown below.

Code: create table emp_det(
id int,
name varchar(20),
gender varchar(20))

insert into emp_det values(1, 'Mahesh', 'Male');
insert into emp_det values(2, 'Anu', 'Female');

Create table dept_det(
id int,
name varchar(20))

insert into dept_det values(1, 'IT');
insert into dept_det values(2, 'HR');
insert into dept_det values(1, 'Payroll');

Select * from emp_det;
Select * from dept_det;

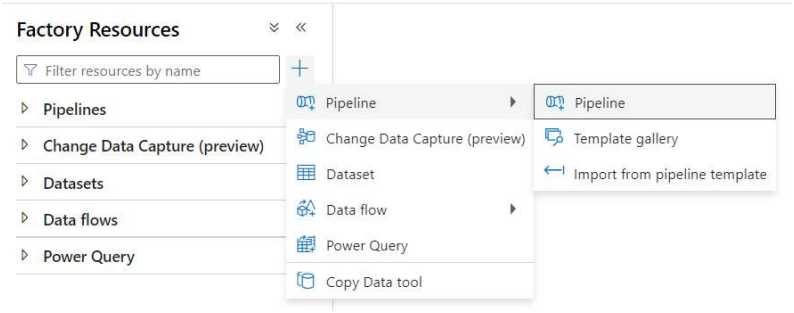
```
16
17  Select * from emp_det;
```

Results Messages

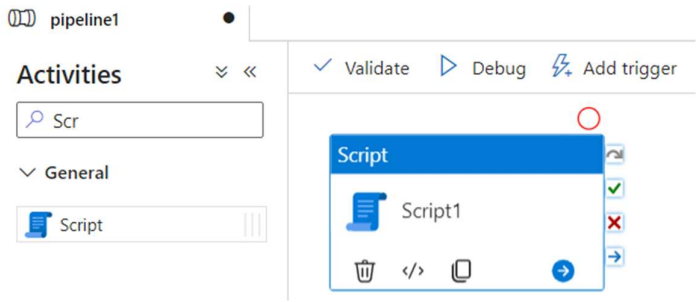
Search to filter items...

id	name	gender
1	Mahesh	Male
2	Anu	Female

2. Create a Pipeline.

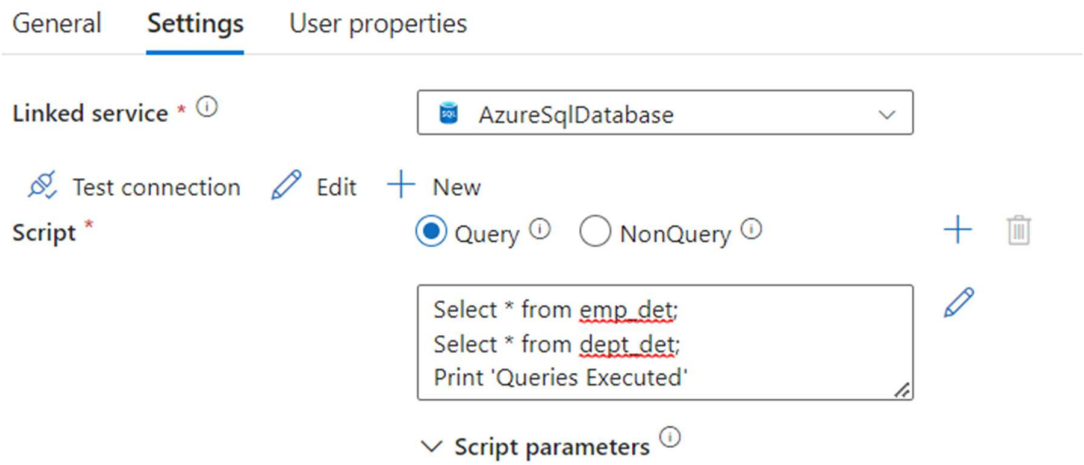


3. Drag and drop the Script Activity.

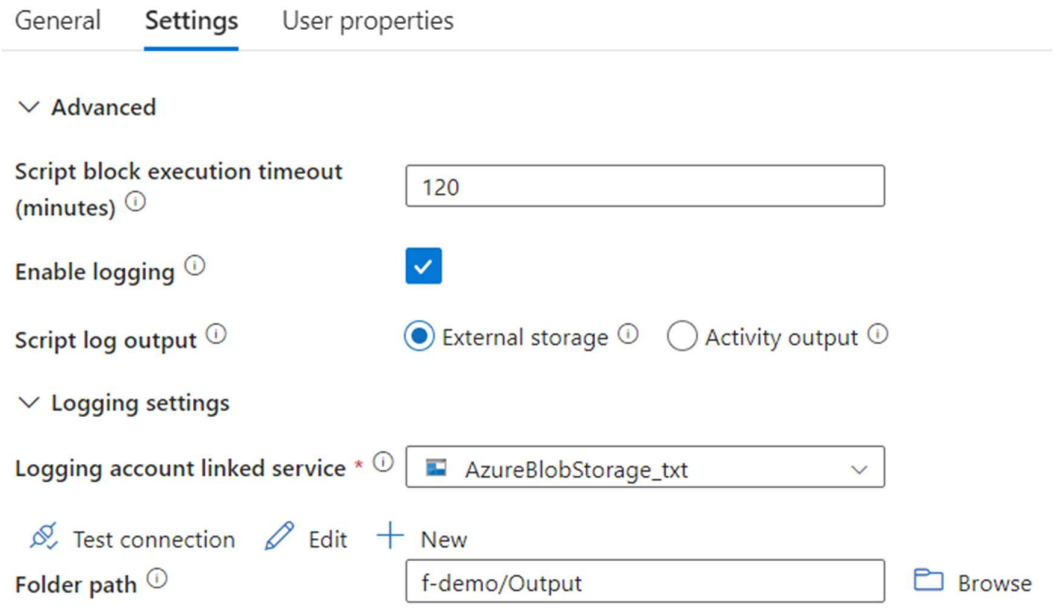


4. Under setting, give the below script.

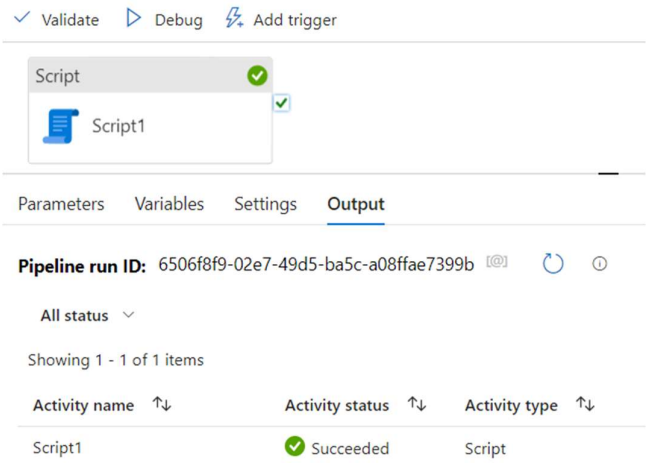
Code: Select * from emp_det;
Select * from dept_det;
Print 'Queries Executed'



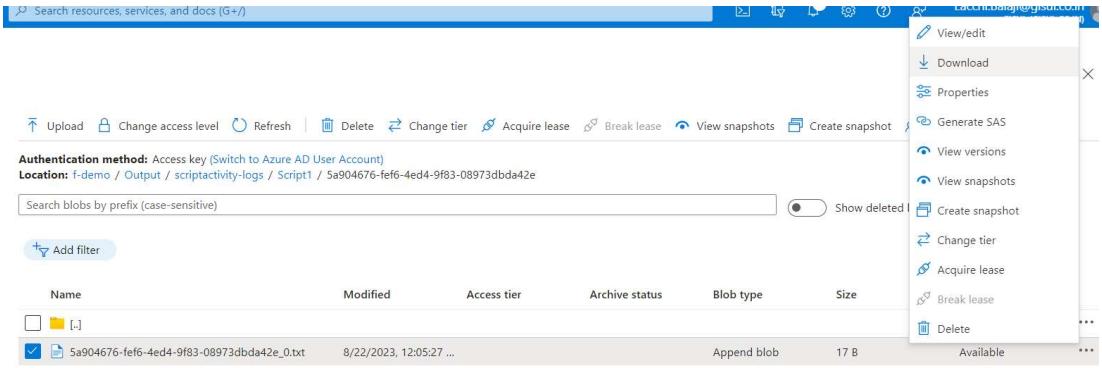
5. Scroll down under Advanced, check the Enable logging and select the linked service then set the location.



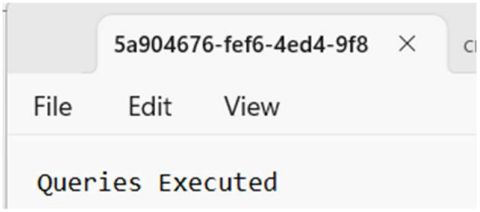
6. Validate the pipeline and click on Debug.
7. Here our pipeline was executed successfully.



8. Go to the location and download the file as shown below.

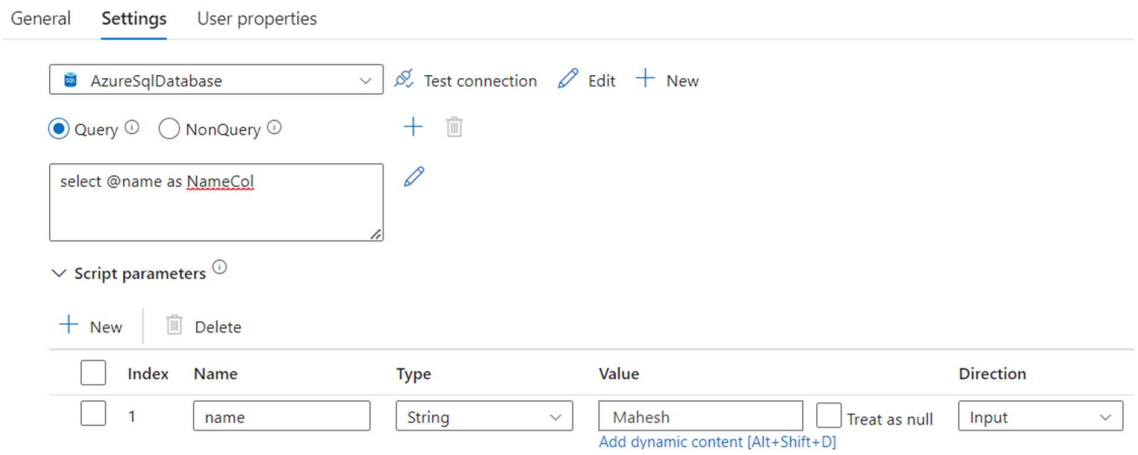


9. When you open the document, you will see the below message.

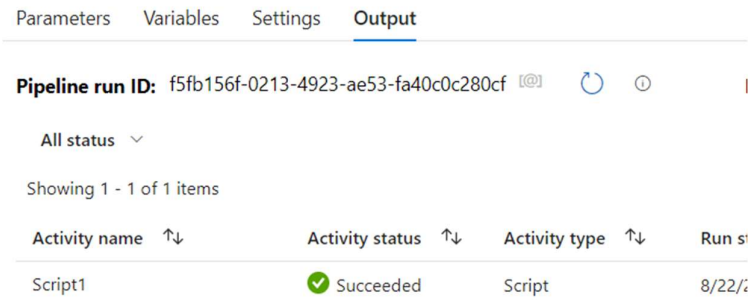


10. Now go back to Script Activity settings and change the query then add the below script parameter.

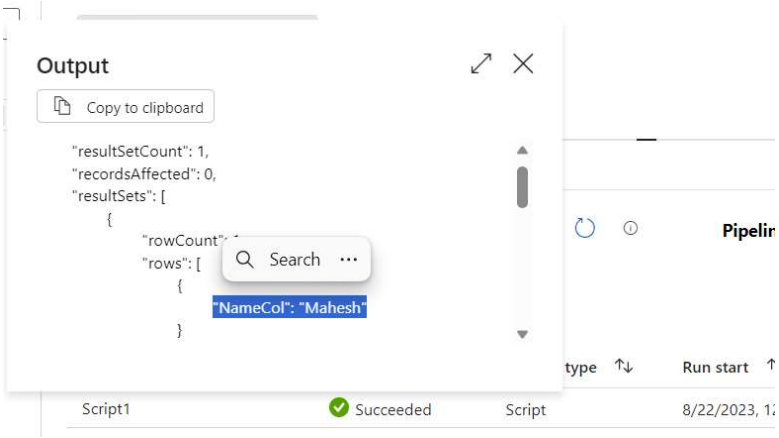
Query: select @name as NameCol



11. Validate the pipeline and click on Debug.
12. Here our pipeline was executed successfully.



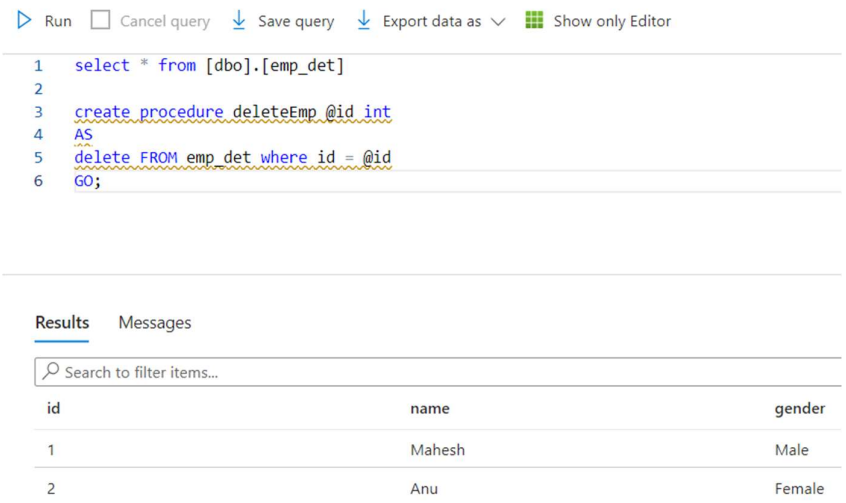
13. In the Output you will see the Name as shown below.



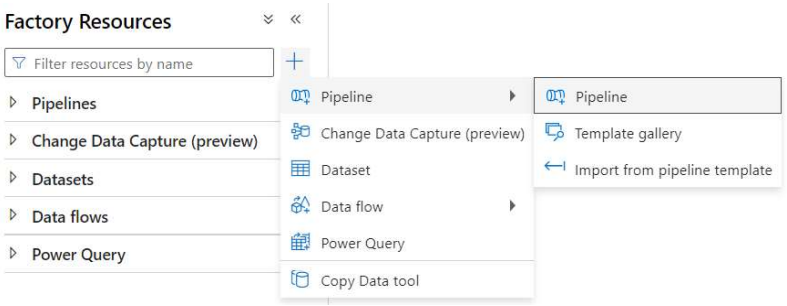
Stored Procedure Activity

1. In this example we are using the below table. Create a Procedure for the Table.

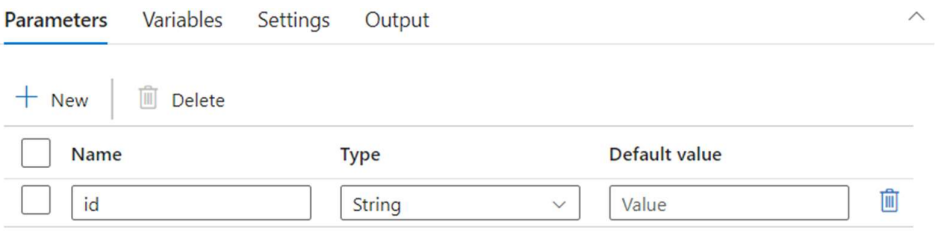
Code: create procedure deleteEmp @id int
AS
delete FROM emp_det where id = @id
GO;



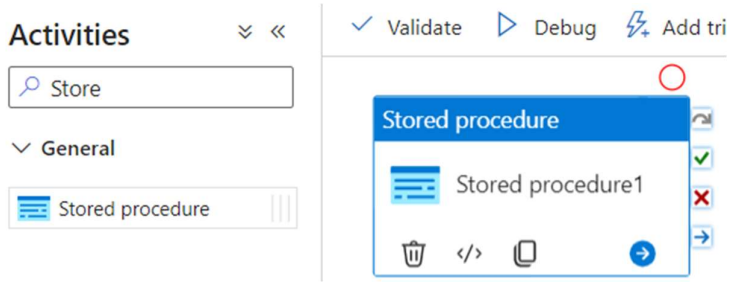
2. Create a Pipeline.



3. Under parameter create a below parameter.

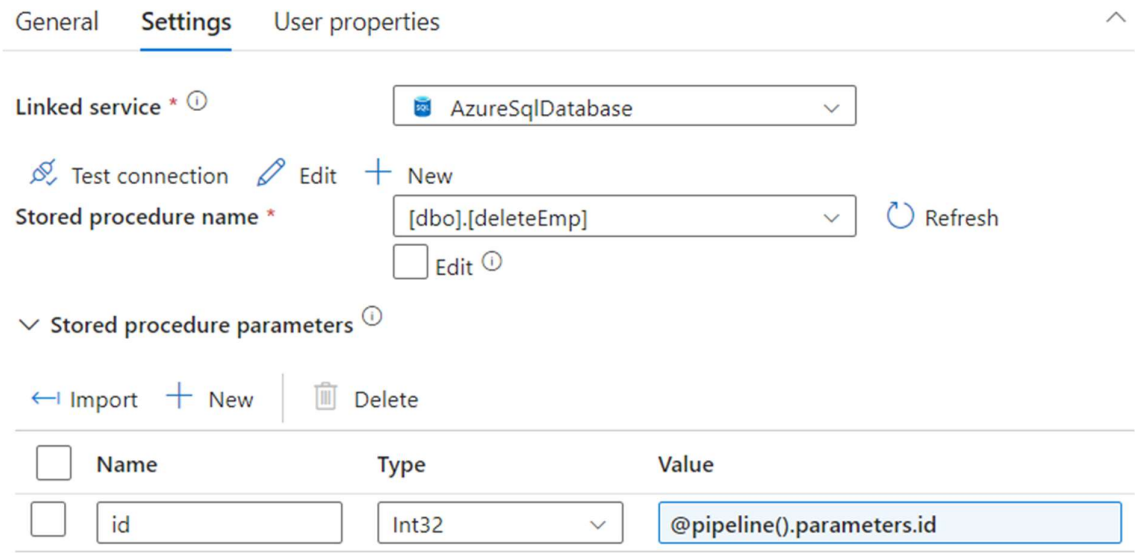


4. Drag and drop the stored procedure activity.



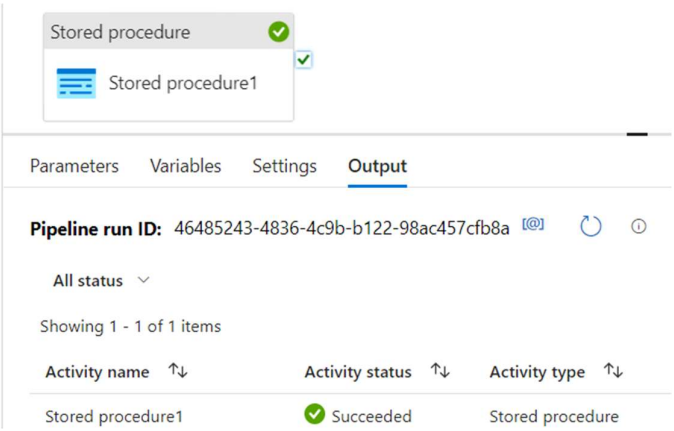
5. Now select the SQLServer linked service and select the Stored Procedure that we created before.
6. Then under parameters click on import and give the below expression.

Exp: @pipeline().parameters.id



7. Validate the pipeline and click on debug.
8. It will ask the id value, give 1.

9. Our pipeline executed successfully.



10. Now go to SQL database and run the table, id 1 row has been deleted.