LAB - Incrementally load data from Azure SQL Database to Azure Blob storage using the Azure portal

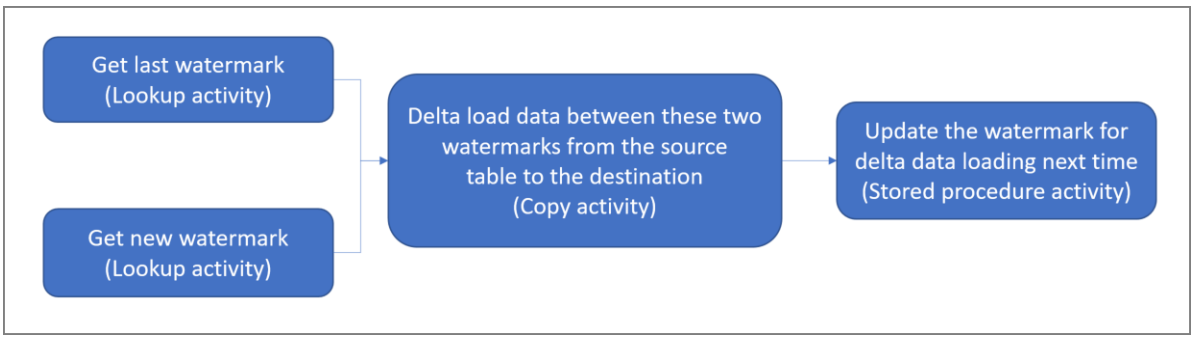
In this tutorial, you create an Azure Data Factory with a pipeline that loads delta data from a table in Azure SQL Database to Azure Blob storage.

You perform the following steps in this tutorial:

- Prepare the data store to store the watermark value.
- Create a data factory.
- Create linked services.
- Create source, sink, and watermark datasets.
- Create a pipeline.
- Run the pipeline.
- Monitor the pipeline run.
- Review results
- Add more data to the source.
- Run the pipeline again.
- Monitor the second pipeline run
- Review results from the second run

Overview

Here is the high-level solution diagram:



Here are the important steps to create this solution:

1. **Select the watermark column**. Select one column in the source data store, which can be used to slice the new or updated records for every run. Normally, the data in this selected column (for example, last_modify_time or ID) keeps increasing when rows are created or updated. The maximum value in this column is used as a watermark.
2. **Prepare a data store to store the watermark value**. In this tutorial, you store the watermark value in a SQL database.
3. **Create a pipeline with the following workflow**:

   The pipeline in this solution has the following activities:

- Create two Lookup activities. Use the first Lookup activity to retrieve the last watermark value. Use the second Lookup activity to retrieve the new watermark value. These watermark values are passed to the Copy activity.
- Create a Copy activity that copies rows from the source data store with the value of the watermark column greater than the old watermark value and less than the new watermark value. Then, it copies the delta data from the source data store to Blob storage as a new file.
- Create a StoredProcedure activity that updates the watermark value for the pipeline that runs next time.

Prerequisites

- **Azure SQL Database**. You use the database as the source data store. If you don't have a database in Azure SQL Database, see Create a database in Azure SQL Database for steps to create one.
- **Azure Storage**. You use the blob storage as the sink data store. If you don't have a storage account, see Create a storage account for steps to create one. Create a container named adftutorial.

Create a data source table in your SQL database

1. Open SQL Server Management Studio. In **Server Explorer**, right-click the database, and choose **New Query**.
2. Run the following SQL command against your SQL database to create a table named `data_source_table` as the data source store:

```
create table data_source_table
(
    PersonID int,
    Name varchar(255),
    LastModifytime datetime
);

INSERT INTO data_source_table
(PersonID, Name, LastModifytime)
VALUES
(1, 'aaaa','9/1/2017 12:56:00 AM'),
(2, 'bbbb','9/2/2017 5:23:00 AM'),
(3, 'cccc','9/3/2017 2:36:00 AM'),
(4, 'dddd','9/4/2017 3:21:00 AM'),
(5, 'eeee','9/5/2017 8:06:00 AM');
```

In this tutorial, you use LastModifytime as the watermark column. The data in the data source store is shown in the following table:

```
PersonID | Name | LastModifytime
-------- | ---- | ----------------------
1        | aaaa | 2017-09-01 00:56:00.000
2        | bbbb | 2017-09-02 05:23:00.000
3        | cccc | 2017-09-03 02:36:00.000
4        | dddd | 2017-09-04 03:21:00.000
5        | eeee | 2017-09-05 08:06:00.000
```

Create another table in your SQL database to store the high watermark value

1. Run the following SQL command against your SQL database to create a table named `watermarktable` to store the watermark value:

```
create table watermarktable
(

TableName varchar(255),
WatermarkValue datetime,
);
```

2. Set the default value of the high watermark with the table name of source data store. In this tutorial, the table name is data_source_table.

```
INSERT INTO watermarktable
VALUES ('data_source_table','1/1/2010 12:00:00 AM')
```

3. Review the data in the table `watermarktable`.

```
Select * from watermarktable
```

Output:

```
TableName         | WatermarkValue
----------------- | ----------------------
data_source_table | 2010-01-01 00:00:00.000
```

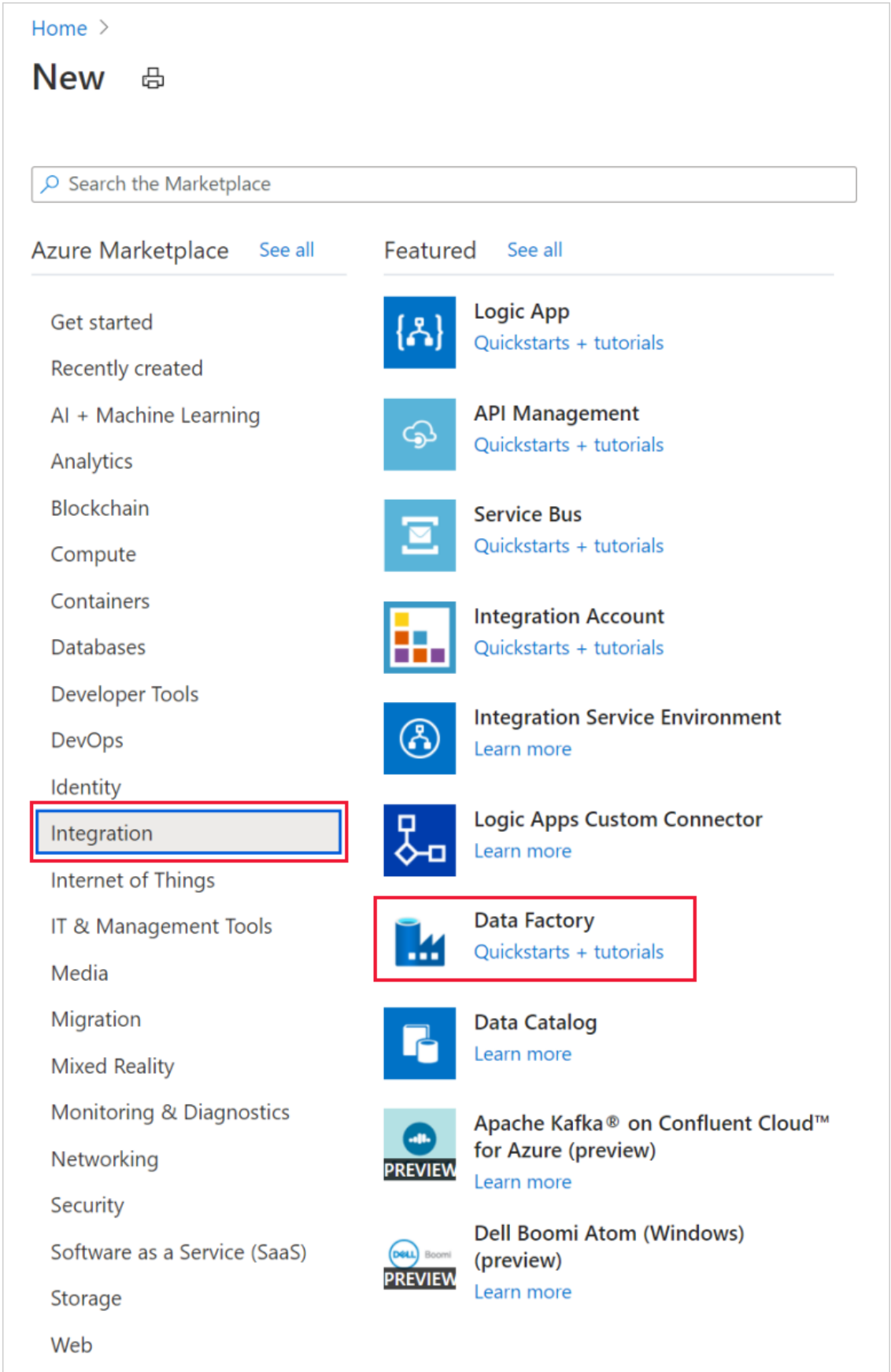
Create a stored procedure in your SQL database

Run the following command to create a stored procedure in your SQL database:

```
CREATE PROCEDURE usp_write_watermark @LastModifiedtime datetime, @TableName
varchar(50)
AS

BEGIN

UPDATE watermarktable
SET [WatermarkValue] = @LastModifiedtime
WHERE [TableName] = @TableName

END
```

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. On the left menu, select **Create a resource** > **Integration** > **Data Factory**:



3. In the **New data factory** page, enter **ADFIncCopyTutorialDF** for the **name**.

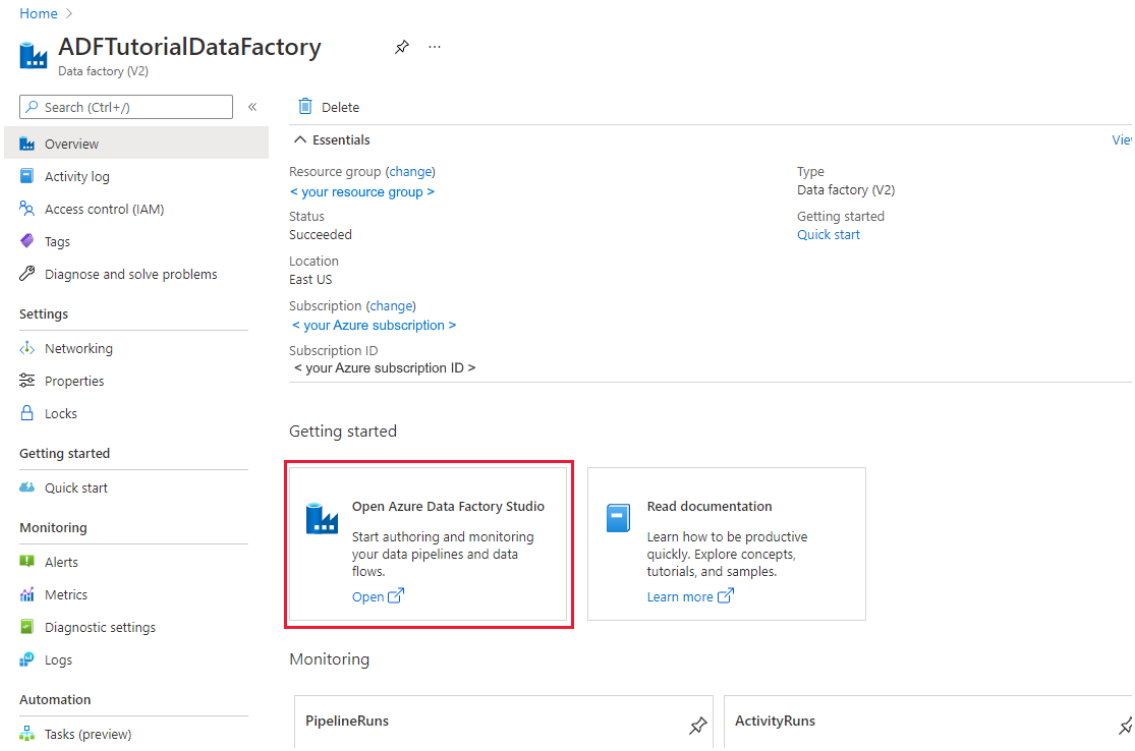
   The name of the Azure Data Factory must be **globally unique**. If you see a red exclamation mark with the following error, change the name of the data factory (for example, yournameADFIncCopyTutorialDF) and try creating again. See Data Factory - Naming Rules article for naming rules for Data Factory artifacts.

   Data factory name "ADFIncCopyTutorialDF" is not available

4. Select your Azure **subscription** in which you want to create the data factory.

5. For the **Resource Group**, do one of the following steps:
    - Select **Use existing**, and select an existing resource group from the drop-down list.
    - Select **Create new**, and enter the name of a resource group.

      To learn about resource groups, see Using resource groups to manage your Azure resources.
6. Select **V2** for the **version**.
7. Select the **location** for the data factory. Only locations that are supported are displayed in the drop-down list. The data stores (Azure Storage, Azure SQL Database, Azure SQL Managed Instance, and so on) and computes (HDInsight, etc.) used by data factory can be in other regions.
8. Click **Create**.
9. After the creation is complete, you see the **Data Factory** page as shown in the image.

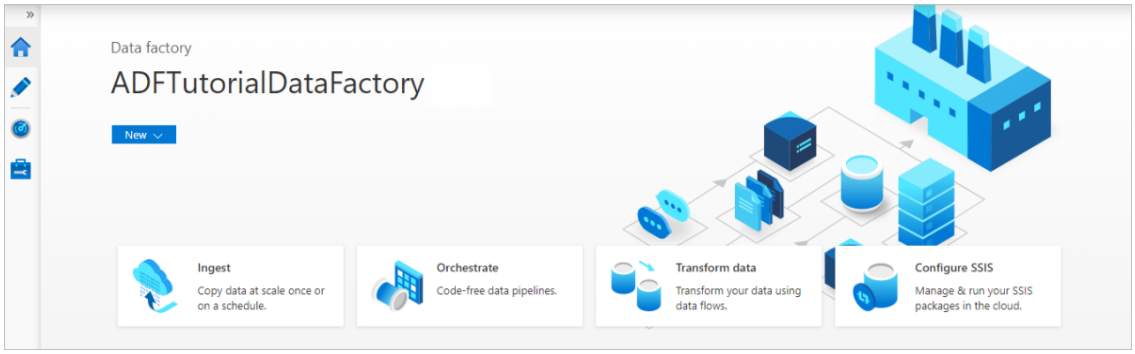


10. Select **Open** on the **Open Azure Data Factory Studio** tile to launch the Azure Data Factory user interface (UI) in a separate tab.

Create a pipeline

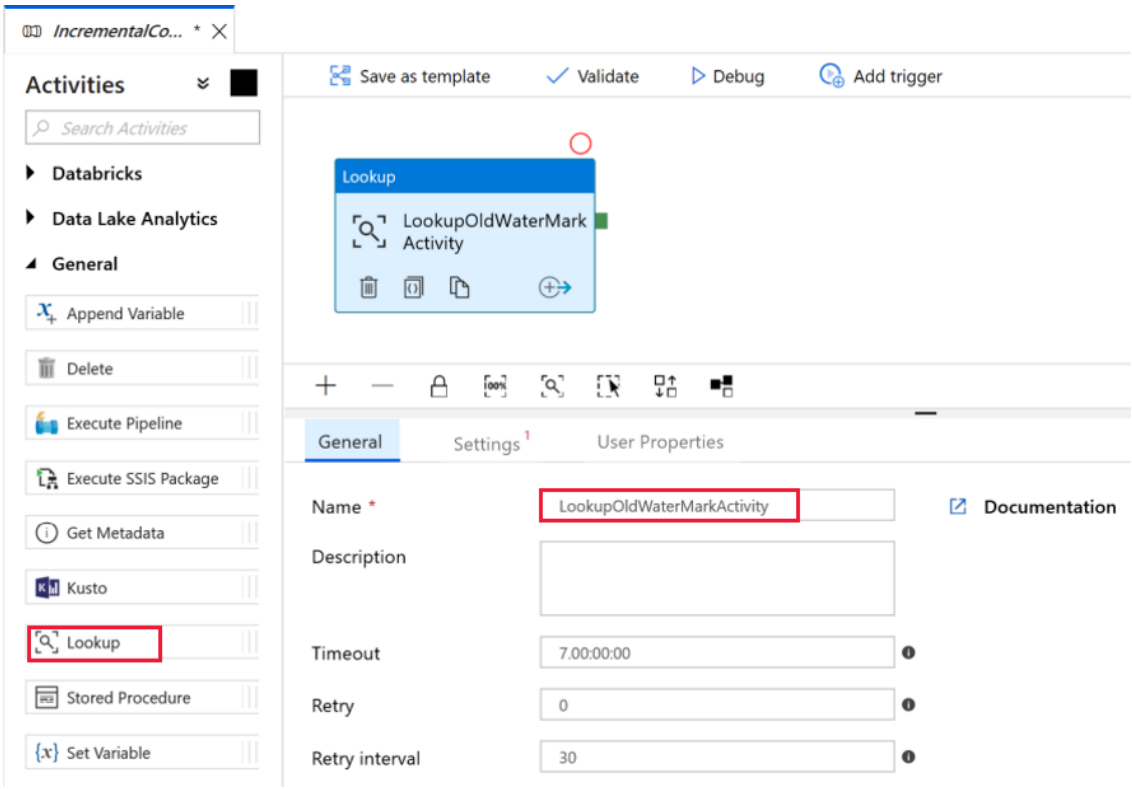
In this tutorial, you create a pipeline with two Lookup activities, one Copy activity, and one StoredProcedure activity chained in one pipeline.

1. On the home page of Data Factory UI, click the **Orchestrate** tile.

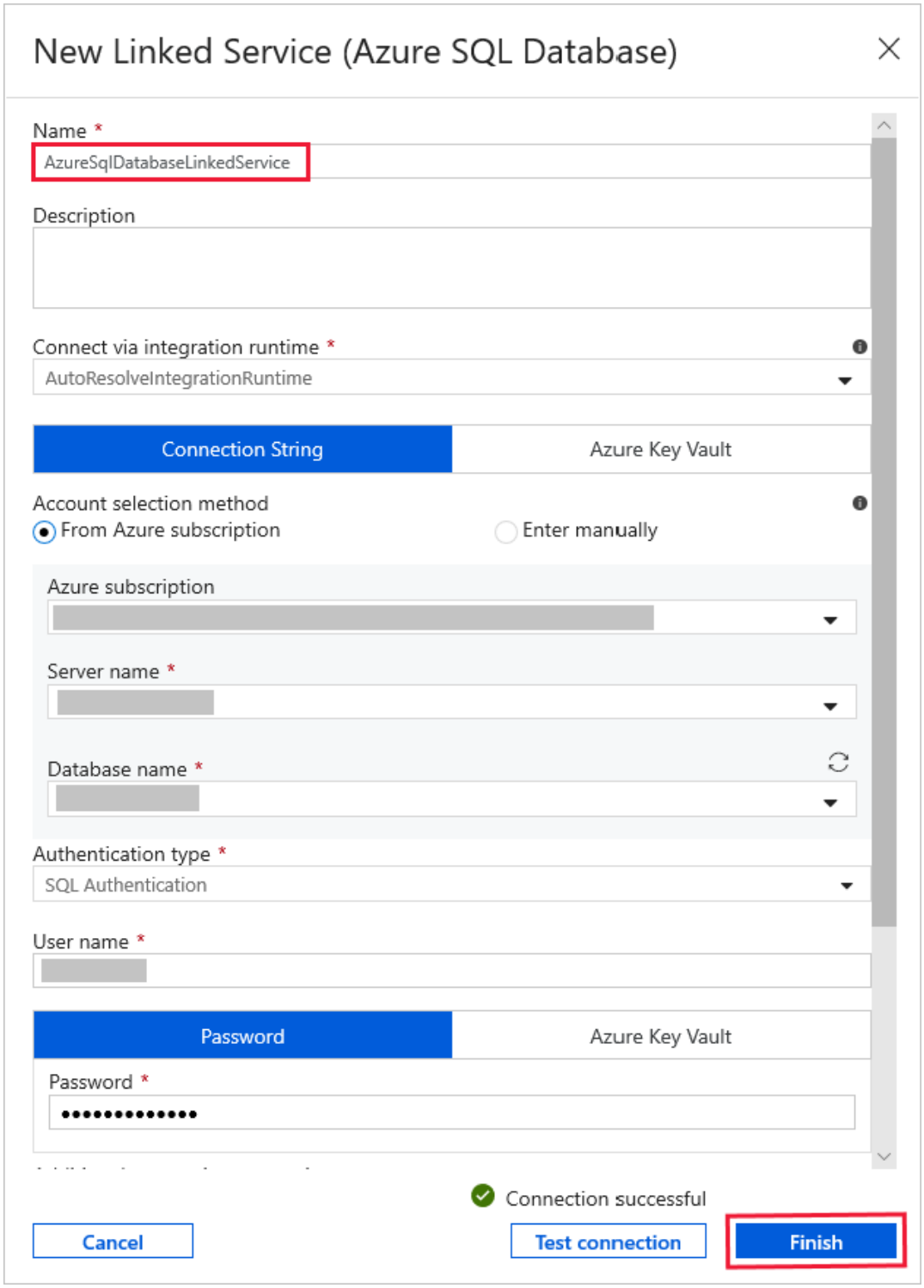


2. In the General panel under **Properties**, specify **IncrementalCopyPipeline** for **Name**. Then collapse the panel by clicking the Properties icon in the top-right corner.

3. Let's add the first lookup activity to get the old watermark value. In the **Activities** toolbox, expand **General**, and drag-drop the **Lookup** activity to the pipeline designer surface. Change the name of the activity to **LookupOldWaterMarkActivity**.

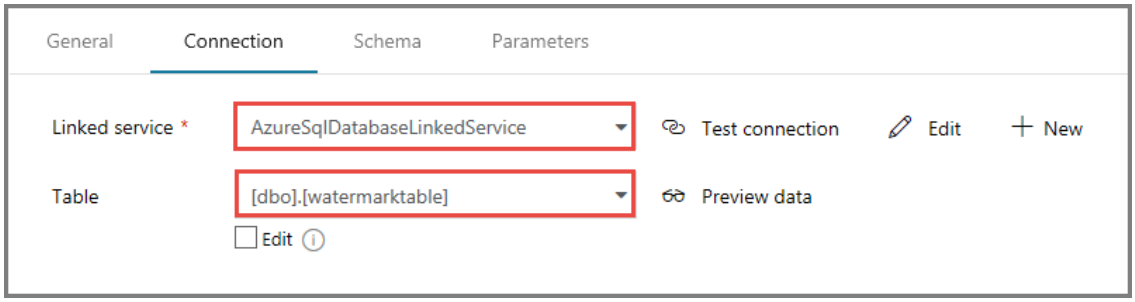


4. Switch to the **Settings** tab, and click **+ New** for **Source Dataset**. In this step, you create a dataset to represent data in the **watermarktable**. This table contains the old watermark that was used in the previous copy operation.
5. In the **New Dataset** window, select **Azure SQL Database**, and click **Continue**. You see a new window opened for the dataset.
6. In the **Set properties** window for the dataset, enter **WatermarkDataset** for **Name**.
7. For **Linked Service**, select **New**, and then do the following steps:
    1. Enter **AzureSqlDatabaseLinkedService** for **Name**.
    2. Select your server for **Server name**.
    3. Select your **Database name** from the dropdown list.
    4. Enter your **User name** & **Password**.
    5. To test connection to the your SQL database, click **Test connection**.
    6. Click **Finish**.
    7. Confirm that **AzureSqlDatabaseLinkedService** is selected for **Linked service**.



8. Select **Finish**.

8. In the **Connection** tab, select **[dbo].[watermarktable]** for **Table**. If you want to preview data in the table, click **Preview data**.

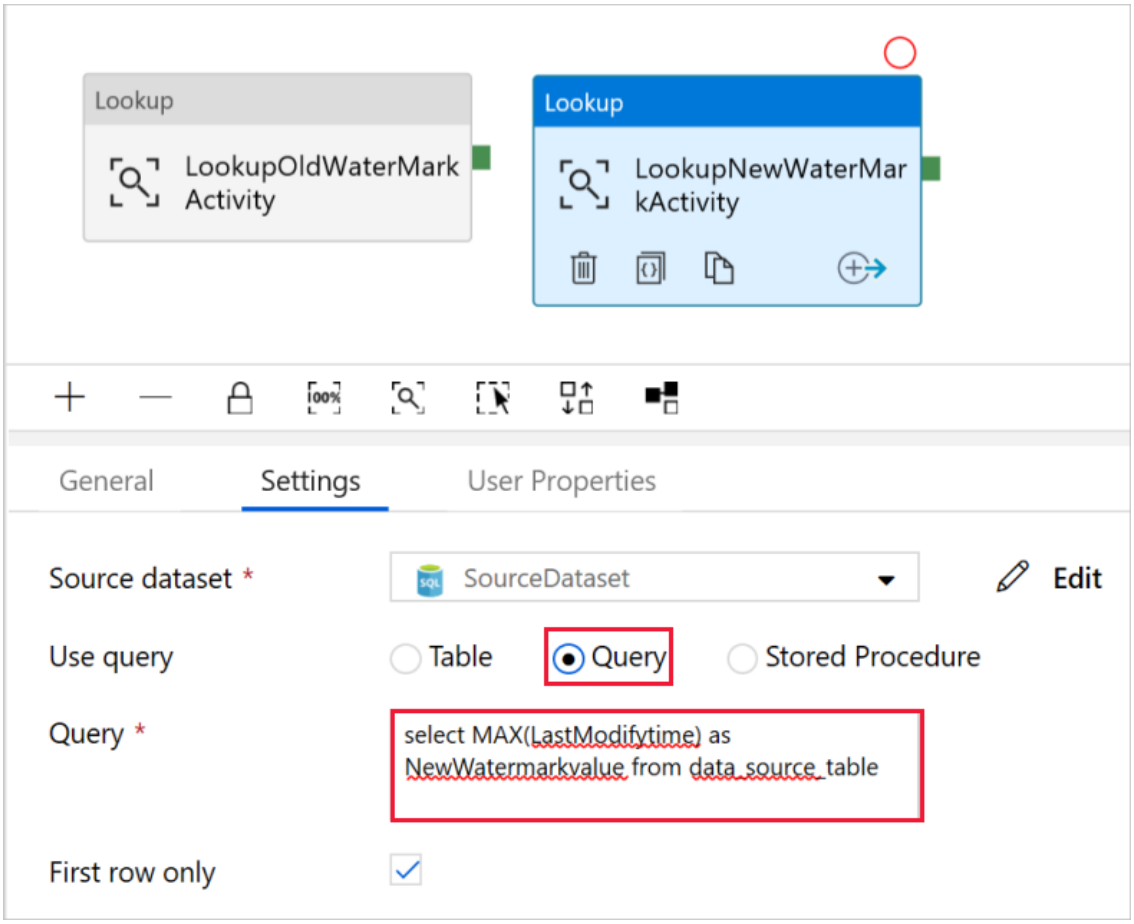


9. Switch to the pipeline editor by clicking the pipeline tab at the top or by clicking the name of the pipeline in the tree view on the left. In the properties window for the **Lookup** activity, confirm that **WatermarkDataset** is selected for the **Source Dataset** field.

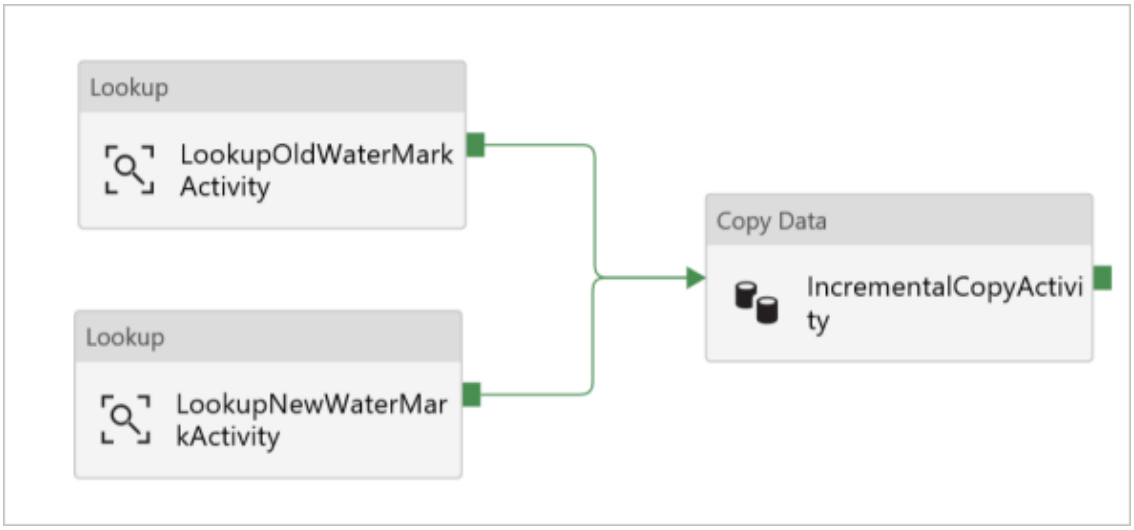
10. In the **Activities** toolbox, expand **General**, and drag-drop another **Lookup** activity to the pipeline designer surface, and set the name to **LookupNewWaterMarkActivity** in the **General** tab of the properties window. This Lookup activity gets the new watermark value from the table with the source data to be copied to the destination.
11. In the properties window for the second **Lookup** activity, switch to the **Settings** tab, and click **New**. You create a dataset to point to the source table that contains the new watermark value (maximum value of LastModifyTime).
12. In the **New Dataset** window, select **Azure SQL Database**, and click **Continue**.
13. In the **Set properties** window, enter **SourceDataset** for **Name**. Select **AzureSqlDatabaseLinkedService** for **Linked service**.
14. Select **[dbo].[data_source_table]** for Table. You specify a query on this dataset later in the tutorial. The query takes the precedence over the table you specify in this step.
15. Select **Finish**.
16. Switch to the pipeline editor by clicking the pipeline tab at the top or by clicking the name of the pipeline in the tree view on the left. In the properties window for the **Lookup** activity, confirm that **SourceDataset** is selected for the **Source Dataset** field.
17. Select **Query** for the **Use Query** field, and enter the following query: you are only selecting the maximum value of **LastModifytime** from the **data_source_table**. Please make sure you have also checked **First row only**.

    SQLCopy

```
select MAX(LastModifytime) as NewWatermarkvalue from data_source_table
```



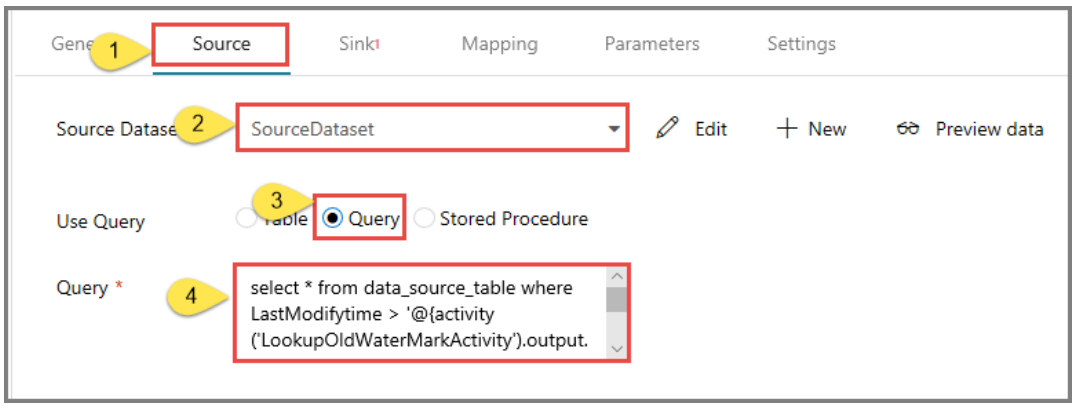
18. In the **Activities** toolbox, expand **Move & Transform**, and drag-drop the **Copy** activity from the Activities toolbox, and set the name to **IncrementalCopyActivity**.
19. **Connect both Lookup activities to the Copy activity** by dragging the **green button** attached to the Lookup activities to the Copy activity. Release the mouse button when you see the border color of the Copy activity changes to blue.



20. Select the **Copy activity** and confirm that you see the properties for the activity in the **Properties** window.
21. Switch to the **Source** tab in the **Properties** window, and do the following steps:
    1. Select **SourceDataset** for the **Source Dataset** field.
    2. Select **Query** for the **Use Query** field.
    3. Enter the following SQL query for the **Query** field.

       SQLCopy

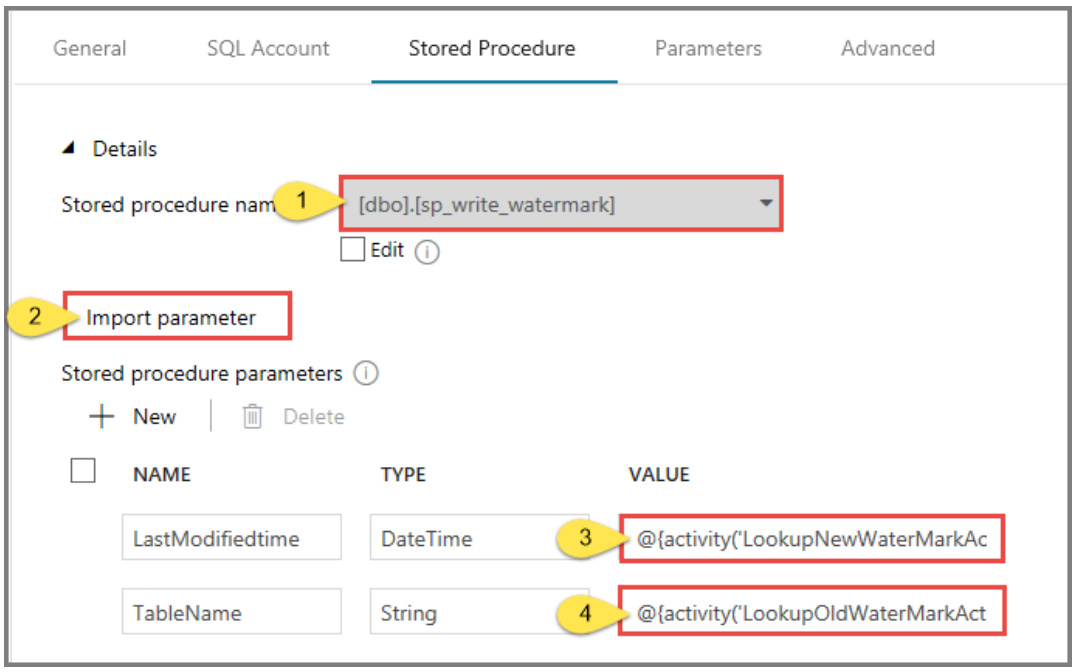
```
select * from data_source_table where LastModifytime >
'@{activity('LookupOldWaterMarkActivity').output.firstRow.WatermarkValue}
' and LastModifytime <=
'@{activity('LookupNewWaterMarkActivity').output.firstRow.NewWatermarkval
ue}'
```



22. Switch to the **Sink** tab, and click **+ New** for the **Sink Dataset** field.
23. In this tutorial sink data store is of type Azure Blob Storage. Therefore, select **Azure Blob Storage**, and click **Continue** in the **New Dataset** window.
24. In the **Select Format** window, select the format type of your data, and click **Continue**.

25. In the **Set Properties** window, enter **SinkDataset** for **Name**. For **Linked Service**, select **+ New**. In this step, you create a connection (linked service) to your **Azure Blob storage**.
26. In the **New Linked Service (Azure Blob Storage)** window, do the following steps:
    1. Enter **AzureStorageLinkedService** for **Name**.
    2. Select your Azure Storage account for **Storage account name**.
    3. Test Connection and then click **Finish**.
27. In the **Set Properties** window, confirm that **AzureStorageLinkedService** is selected for **Linked service**. Then select **Finish**.
28. Go to the **Connection** tab of SinkDataset and do the following steps:
    1. For the **File path** field, enter **adftutorial/incrementalcopy**. **adftutorial** is the blob container name and **incrementalcopy** is the folder name. This snippet assumes that you have a blob container named adftutorial in your blob storage. Create the container if it doesn't exist, or set it to the name of an existing one. Azure Data Factory automatically creates the output folder **incrementalcopy** if it does not exist. You can also use the **Browse** button for the **File path** to navigate to a folder in a blob container.
    2. For the **File** part of the **File path** field, select **Add dynamic content [Alt+P]**, and then enter `@CONCAT('Incremental-', pipeline().RunId, '.txt')` in the opened window. Then select **Finish**. The file name is dynamically generated by using the expression. Each pipeline run has a unique ID. The Copy activity uses the run ID to generate the file name.
29. Switch to the **pipeline** editor by clicking the pipeline tab at the top or by clicking the name of the pipeline in the tree view on the left.
30. In the **Activities** toolbox, expand **General**, and drag-drop the **Stored Procedure** activity from the **Activities** toolbox to the pipeline designer surface. **Connect** the green (Success) output of the **Copy** activity to the **Stored Procedure** activity.
31. Select **Stored Procedure Activity** in the pipeline designer, change its name to **StoredProceduretoWriteWatermarkActivity**.
32. Switch to the **SQL Account** tab, and select **AzureSqlDatabaseLinkedService** for **Linked service**.
33. Switch to the **Stored Procedure** tab, and do the following steps:
    1. For **Stored procedure name**, select **usp_write_watermark**.
    2. To specify values for the stored procedure parameters, click **Import parameter**, and enter following values for the parameters:

Name	Type	Value
LastModifiedtime	DateTime	@{activity('LookupNewWaterMarkActivity').output.firstRow.NewWatermarkvalue}
TableName	String	@{activity('LookupOldWaterMarkActivity').output.firstRow.TableName}



34. To validate the pipeline settings, click **Validate** on the toolbar. Confirm that there are no validation errors. To close the **Pipeline Validation Report** window, click >>.
35. Publish entities (linked services, datasets, and pipelines) to the Azure Data Factory service by selecting the **Publish All** button. Wait until you see a message that the publishing succeeded.

Trigger a pipeline run

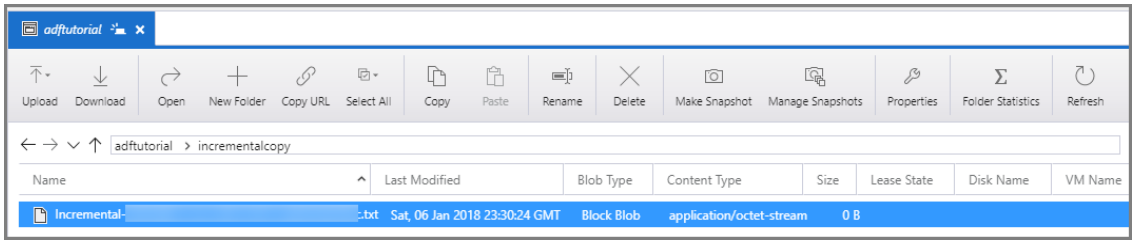
1. Click **Add Trigger** on the toolbar, and click **Trigger Now**.
2. In the **Pipeline Run** window, select **Finish**.

Monitor the pipeline run

1. Switch to the **Monitor** tab on the left. You see the status of the pipeline run triggered by a manual trigger. You can use links under the **PIPELINE NAME** column to view run details and to rerun the pipeline.
2. To see activity runs associated with the pipeline run, select the link under the **PIPELINE NAME** column. For details about the activity runs, select the **Details** link (eyeglasses icon) under the **ACTIVITY NAME** column. Select **All pipeline runs** at the top to go back to the Pipeline Runs view. To refresh the view, select **Refresh**.

Review the results

1. Connect to your Azure Storage Account by using tools such as Azure Storage Explorer. Verify that an output file is created in the **incrementalcopy** folder of the **adftutorial** container.



2. Open the output file and notice that all the data is copied from the **data_source_table** to the blob file.

```
1,aaaa,2017-09-01 00:56:00.0000000
2,bbbb,2017-09-02 05:23:00.0000000
3,cccc,2017-09-03 02:36:00.0000000
4,dddd,2017-09-04 03:21:00.0000000
5,eeee,2017-09-05 08:06:00.0000000
```

3. Check the latest value from `watermarktable`. You see that the watermark value was updated.

```
Select * from watermarktable
```

Here is the output:

TableName	WatermarkValue
data_source_table	2017-09-05 8:06:00.000

Add more data to source

Insert new data into your database (data source store).

```
INSERT INTO data_source_table
VALUES (6, 'newdata','9/6/2017 2:23:00 AM')

INSERT INTO data_source_table
VALUES (7, 'newdata','9/7/2017 9:01:00 AM')
```

The updated data in the your database is:

```
PersonID | Name | LastModifytime
-------- | ---- | --------------
1 | aaaa | 2017-09-01 00:56:00.000
2 | bbbb | 2017-09-02 05:23:00.000
3 | cccc | 2017-09-03 02:36:00.000
4 | dddd | 2017-09-04 03:21:00.000
5 | eeee | 2017-09-05 08:06:00.000
6 | newdata | 2017-09-06 02:23:00.000
7 | newdata | 2017-09-07 09:01:00.000
```

Trigger another pipeline run

1. Switch to the **Edit** tab. Click the pipeline in the tree view if it's not opened in the designer.
2. Click **Add Trigger** on the toolbar, and click **Trigger Now**.

Monitor the second pipeline run

1. Switch to the **Monitor** tab on the left. You see the status of the pipeline run triggered by a manual trigger. You can use links under the **PIPELINE NAME** column to view activity details and to rerun the pipeline.
2. To see activity runs associated with the pipeline run, select the link under the **PIPELINE NAME** column. For details about the activity runs, select the **Details** link (eyeglasses icon) under the **ACTIVITY NAME** column. Select **All pipeline runs** at the top to go back to the Pipeline Runs view. To refresh the view, select **Refresh**.

Verify the second output

1. In the blob storage, you see that another file was created. In this tutorial, the new file name is `Incremental-<GUID>.txt`. Open that file, and you see two rows of records in it.

   ```
   6,newdata,2017-09-06 02:23:00.0000000
   7,newdata,2017-09-07 09:01:00.0000000
   ```

2. Check the latest value from `watermarktable`. You see that the watermark value was updated again.

   ```
   Select * from watermarktable
   ```

   sample output:

TableName	WatermarkValue
data_source_table	2017-09-07 09:01:00.000

Conclusion

You performed the following steps in this tutorial:

- Prepare the data store to store the watermark value.
- Create a data factory.
- Create linked services.
- Create source, sink, and watermark datasets.
- Create a pipeline.
- Run the pipeline.
- Monitor the pipeline run.
- Review results
- Add more data to the source.

- Run the pipeline again.
- Monitor the second pipeline run
- Review results from the second run

In this tutorial, the pipeline copied data from a single table in SQL Database to Blob storage.