## Azure Data Factory Trigger Types

Azure Data Factory Triggers are used to schedule a Data Pipeline runs without any interventions. In other words, an Azure Data Factory Trigger is a processing unit that determines when to begin or invoke an end-to-end pipeline execution in Azure Data Factory.

Besides the obvious benefit of scheduling the Data Pipeline for future runs, the Azure Data Factory Trigger also provides the ability to choose and process data from the past. Based on the type of the trigger and criteria defined in the specific trigger, the Azure Data Factory Trigger determines when the pipeline execution will be invoked to create a Data-Driven Workflow.

Azure Data Factory Triggers come in three different types: Schedule Trigger, Tumbling Window Trigger, and Event-based Trigger.

### Schedule Trigger

This Azure Data Factory Trigger is a popular trigger that can run a Data Pipeline according to a **predetermined schedule**. It provides extra flexibility by allowing for different scheduling intervals like minute(s), hour(s), day(s), week(s), or month(s).

You can set the start and end dates for the Schedule Trigger to be active so that it only runs a Pipeline based on the given time period. Furthermore, you can also use the Schedule Trigger to run on future calendar days and times, such as the 30th of each month, the first and third Monday of each month, and more.

The Schedule Azure Data Factory Triggers are built with a **“many to many” relationship** in mind, which implies that one Schedule Trigger can run several Data Pipelines, and a single Data Pipeline can be run by multiple Schedule Triggers.

### Tumbling Window Trigger

The Tumbling Window Azure Data Factory Trigger executes Data Pipelines at a **specified time slice** or **pre-determined periodic time interval**. It is significantly more advantageous than Schedule Triggers when working with historical data to copy or migrate data.

Consider the scenario in which you need to replicate data from a Database into a Data Lake on a regular basis, and you want to keep it in separate files or folders for every hour or day.

To implement this use case, you have to set a Tumbling Window Azure Data Factory Trigger for every 1 hour or every 24 hours. The Tumbling Window Trigger sends the start and end times for each time window to the Database, returning all data between those periods. Finally, the data for each hour or day can be saved in its own file or folder.

**Event-based Trigger**

The Event-based Azure Data Factory Trigger runs Data Pipelines i**n response to blob-related events**, such as generating or deleting a blob file present in an Azure Blob Storage. With the Event-based Triggers, you can schedule the Data Pipelines to execute in response to an event from Azure Blob Storage.

In addition, Event-based Triggers are not only compatible with blob, but also with ADLs. Similar to Schedule Triggers, Event Triggers can also work on **many-to-many relationships**, in which a single Event Trigger can run several Pipelines, and a single Pipeline can be run by multiple Event Triggers.

**Step 1**: Now, you are all set to create a new Schedule Trigger. Open the **Triggers** tab by clicking on the down arrow next to the **Factory Resources** section. Then, click on the “**+New**” button, as shown in the above image.

**Step 2**: A “**New Trigger**” dialogue box will open, where you have to enter the configuration parameters like Name, Type, and Start Date for your Azure Data Factory Trigger.

In the **Name** field, you have to provide a unique name to the Trigger. For this case, we named it “WeeklyTrigger\_ST.” Then, you have to select the **Type** of trigger. As we are about to create a Schedule Trigger, click on the corresponding radio button near the Schedule trigger.

In the **Start Date** field, enter the start date from when the Schedule Trigger should begin to execute your Pipeline. Then, in the **Recurrence** field, mention how frequently your trigger is supposed to run the Pipeline. For this case, we provided the recurrence or frequency as “Every 1 week” as shown in the image below:

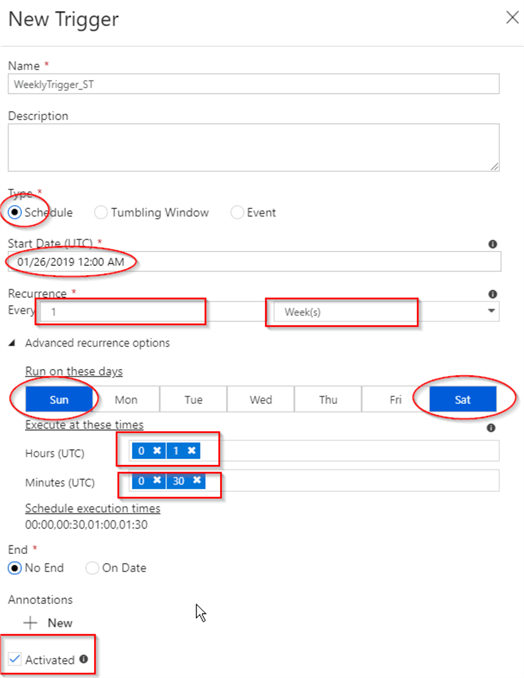
[](https://lh5.googleusercontent.com/nINegIwPFl_bYzV-2WOf9sYHtplJhY8ajEN9ESGZujdvRabxnK0cswnfIRoGoS4B8CWDqR-hLF4wq4MYX8VXl9d3YKJnLy77BsmIJSRw5ves5hVs5F7LETIjU2mprpTfNk4RBh7R)

Image Source: [MS SQL Tips](https://www.mssqltips.com/tipimages2/6062_azure-data-factory-automating-pipeline-executions-part-2.005.png)

It is not mandatory to set the **Advanced Recurrence** options. However, you can configure the Advanced Recurrence options to select a particular weekday to run the Pipeline.

You can also provide the specific timings to schedule your Pipeline executions. For setting a specific time, you have to enter values in the **Hours and Minutes** section, as shown in the image.

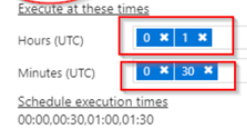
[](https://lh3.googleusercontent.com/8KhtLHhTL2WqwPuk3tquO2Ihi17_85ovwCe0yXeI0yNbnoYW5MvVqEQdGhZRxdB-lkOnuiefI3yjZZZP9AIrY2PQqdfYVeZUM751fs4Xpq3geB0jTrCiWp6qUYwZteujd3kbIhUj)

Image Source: [MS SQL Tips](https://www.mssqltips.com/tipimages2/6062_azure-data-factory-automating-pipeline-executions-part-2.005.png)

**Step 3**: Now, Azure Data Factory will calculate all possible hour/minute combinations and generate eight-time frames in UTC, such as 12:00 AM, 0:30 AM, 1:00 AM, and 1:30 AM, as shown in the image above.

**Step 4**: Next, select the **No End** radio button if you want the Schedule Trigger to continuously run the Pipelines based on the specified time period.

**Step 5**: Select the “**Activated**” checkbox to confirm that the Azure Data Factory Trigger gets activated as soon as it is published in the Azure Data Factory.

**Step 6**: In the next step, you have to assign pre-built pipelines to the newly created Schedule Azure Data Factory Trigger. Select the pipeline named “CopyPipeline\_l6c” from the Factory Resources section. Then, click on Triggers from the left side and select **New/Edit**, as shown in the image.

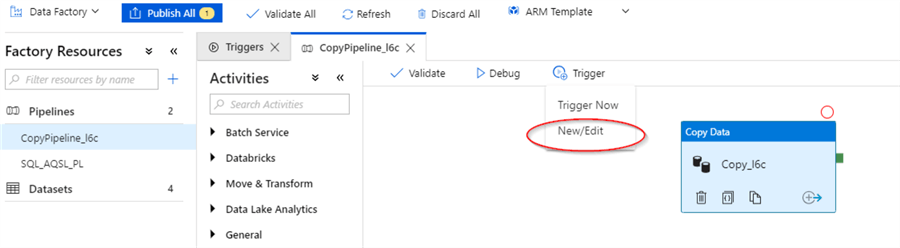
[](https://lh4.googleusercontent.com/SmQit6QMgN3S7Vdpxq9tjigtILUw41Ba0M6MtPC0CnDQ9LS9Cz5hnxwbNibsMxgp7jSzpFSE7ZkPiKvMpH49wPF618aKknYgfpbtCKtejsIX8tNuZkcN58TCsKXS4yN29mWe97aH)

Image Source: [MS SQL Tips](https://www.mssqltips.com/tipimages2/6062_azure-data-factory-automating-pipeline-executions-part-2.006.png)

**Step 7**: The “**Add Trigger**” dialog box appears, where you have to select the name of the newly created Schedule Trigger from the drop-down menu.

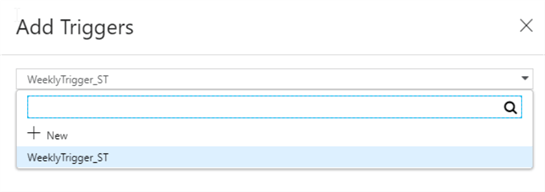
[](https://lh6.googleusercontent.com/1-mej1BqwLrUShdBjJhRoigWuSfFCmPH1OVnVpZ5h3bFXTWuVYAHni5I5i7cOynp6kBDfEMbYdGZgPbd0Pzx8EOW4zEBNeyIBwwaEV6_wA6nJpDjk5Z8wFMLB8A28Sf6vn9M1k9m)

Image Source: [MS SQL Tips](https://www.mssqltips.com/tipimages2/6062_azure-data-factory-automating-pipeline-executions-part-2.007.png)

**Step 8**: Since Schedule Triggers are capable of executing more than one pipeline, you can also assign the trigger named “WeeklyTrigger\_ST” to the pre-built Pipeline named “SQL\_AQSL\_PL.”

After adding a Schedule Trigger to the two different Pipelines, you have to make sure that the Azure Data Factory Trigger is set to **Active**.

[](https://lh3.googleusercontent.com/AN0t6eGY6ZOWpr66VLsUdD9hTbKwTTQ7VQP02j4TvYYmlDVG-wDveLcvmI07jMKiCi-Mp9S6pMOQE8lKLTo3Y1CTbpq3BxoiSkqwVaJF1co8LCLm3nirRBVXrpaU-fRLnt-NiqlG)

Image Source: [MS SQL Tips](https://www.mssqltips.com/tipimages2/6062_azure-data-factory-automating-pipeline-executions-part-2.010.png)

**Step 9**: Navigate to the Triggers tab, where you can see the newly created Schedule Trigger named “WeeklyTrigger\_ST” which is set to active.

<https://hevodata.com/learn/azure-data-factory-trigger/#:~:text=The%20Event%2Dbased%20Azure%20Data,event%20from%20Azure%20Blob%20Storage>.