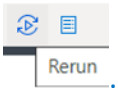
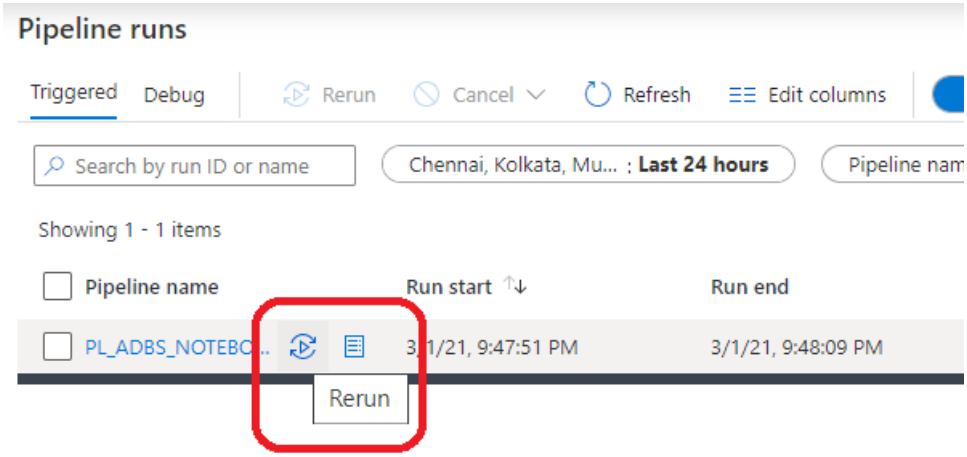
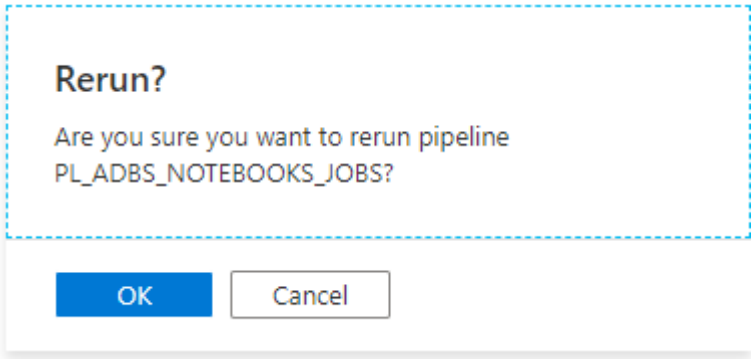
Azure Data Factory Advanced Interview Questions And Answers **Youtube:** [pyspark telugu]

1) How to rerun a pipe line from Data Factory Monitor.

Ans: Simply navigate to the 'Monitor' => Pipeline Runs => Triggered => Click below red circle rerun icon

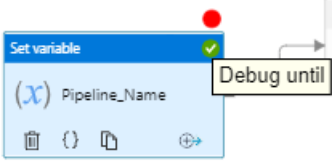






2) If you have 15 activities in pipeline, if we want to debug only first 10 activities how do we do that?

Ans: Every Activity top red Circle will be there. If you select that "Debug Until" it will debug until that activity.

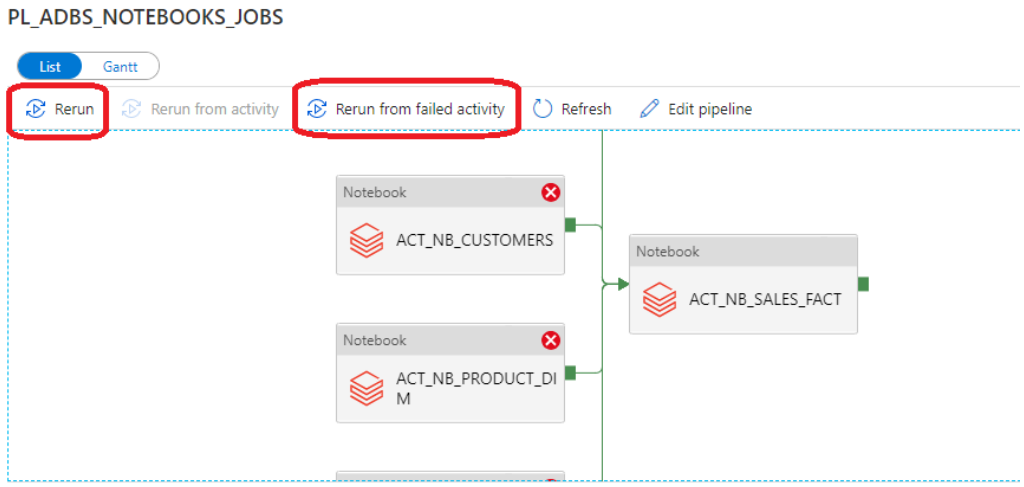


3) How to restart failed pipe line jobs in Azure data factory.

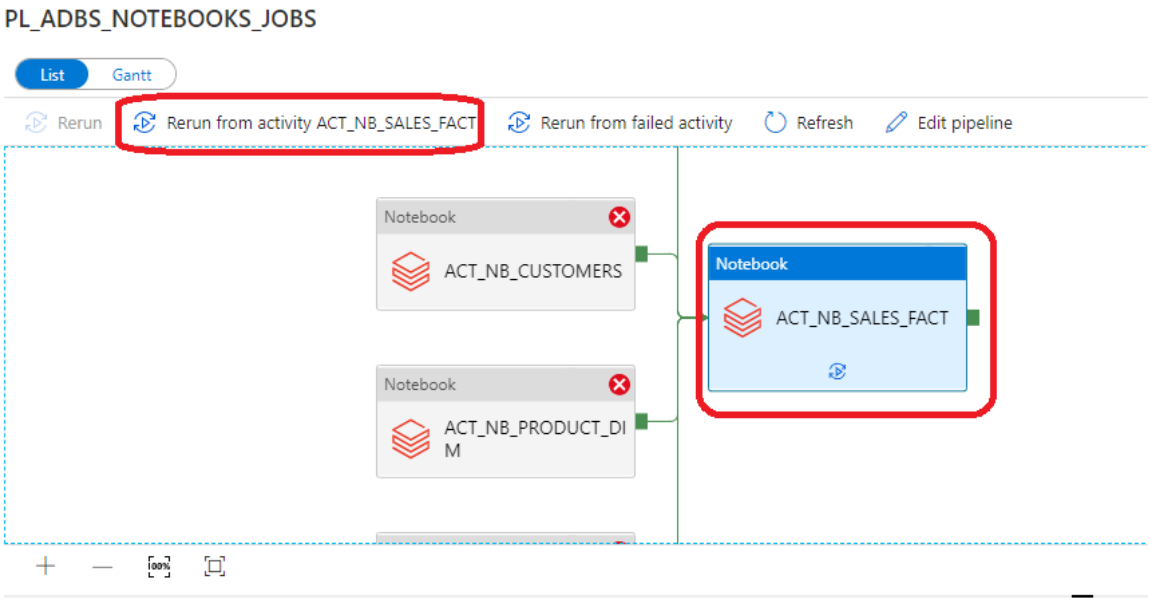
Filed pipelines will have multiple options.

i) **Rerun**: if you select rerun option it will return or restart from first step. If some steps are succeeded, those will be retrigger. So before triggering rerun option verify succeeded steps will be retriggered. There are some changes of reloading data again if its having copy activities.

ii) **Rerun From Failed Activity**: If an activity fails, times out, or is canceled, you can rerun the pipeline from that failed activity by selecting Rerun from failed activity.



iii) **Rerun from Activity**: If you wish to rerun starting at a specific point, you can do so from the activity runs view. Select the activity you wish to start from and select Rerun from activity.



4) How to make activity dependency in Pipeline Multiple Activities and those types.

Activity Dependency defines how subsequent activities depend on previous activities, determining the condition of whether to continue executing the next task. An activity can depend on one or multiple previous activities with different dependency conditions.

The different dependency conditions are: **Succeeded**, **Failed**, **Skipped**, and **Completed**.

For example, if a pipeline has Activity A -> Activity B, the different scenarios that can happen are:

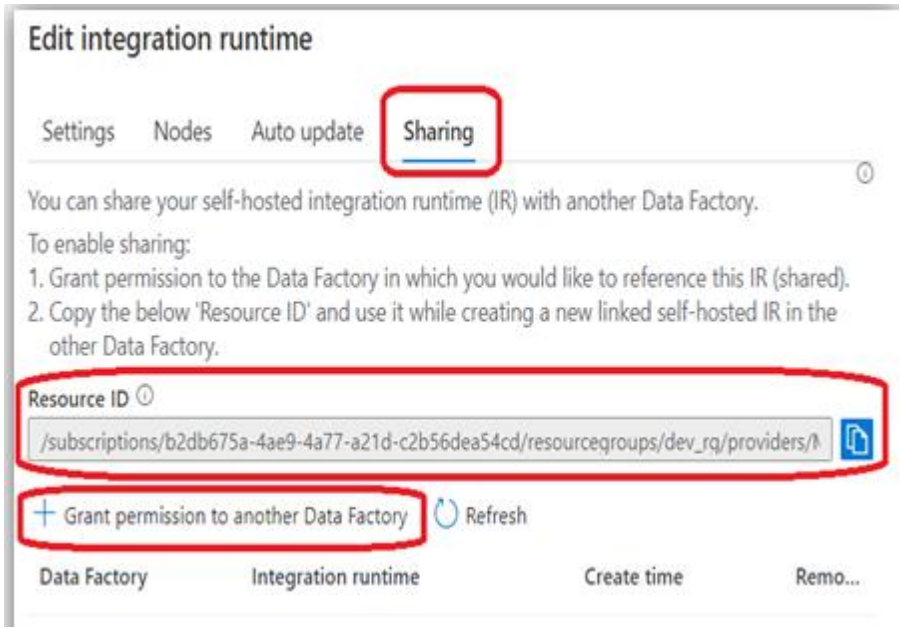
- **succeeded** : Activity B has dependency condition on Activity A with **succeeded**: Activity B only runs if Activity A has a final status of succeeded
- **failed** : Activity B has dependency condition on Activity A with **failed**: Activity B only runs if Activity A has a final status of failed
- **completed** : Activity B has dependency condition on Activity A with **completed**: Activity B runs if Activity A has a final status of succeeded or failed
- **skipped**: Activity B has a dependency condition on Activity A with **skipped**: Activity B runs if Activity A has a final status of skipped. Skipped occurs in the scenario of Activity X -> Activity Y -> Activity Z, where each activity runs only if the previous activity succeeds. If Activity X fails, then Activity Y has a status of "Skipped" because it never executes. Similarly, Activity Z has a status of "Skipped" as well.

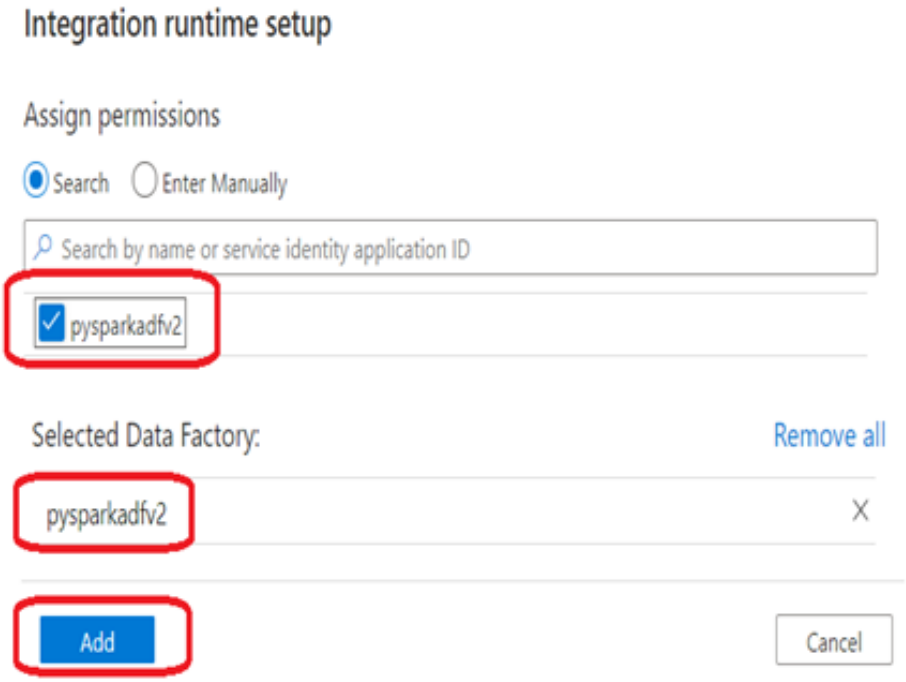
5) What is shared self-hosted Integration Runtime or How to Share Integration Runtime from One Data Factory to another Data Factory?

Shared IR: An original self-hosted IR that runs on a physical infrastructure.
Linked IR: An IR that references another shared IR. The linked IR is a logical IR and uses the infrastructure of another shared self-hosted IR.

While Creating Self-Hosted Integration Runtime we will have separate option to Create Shared

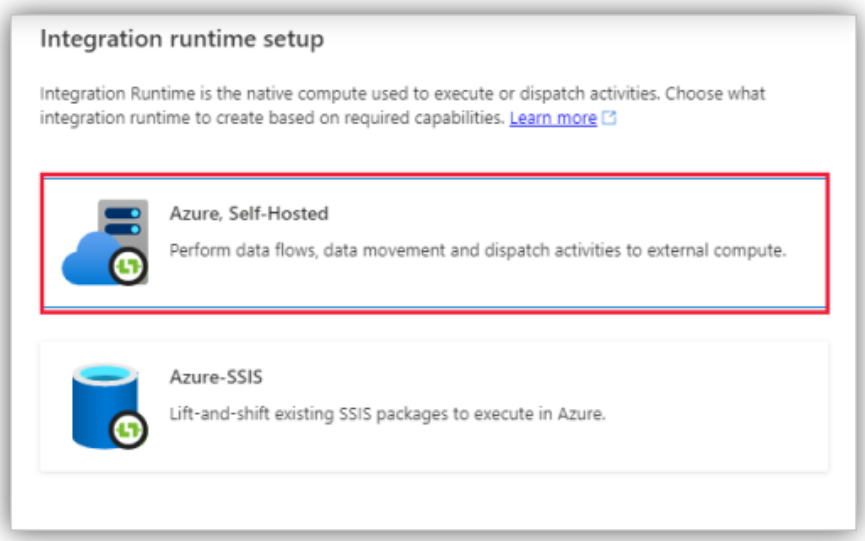
In the self-hosted IR to be shared, select **Grant permission to another Data factory** and in the "Integration runtime setup" page, select the Data factory in which you want to create the linked IR.

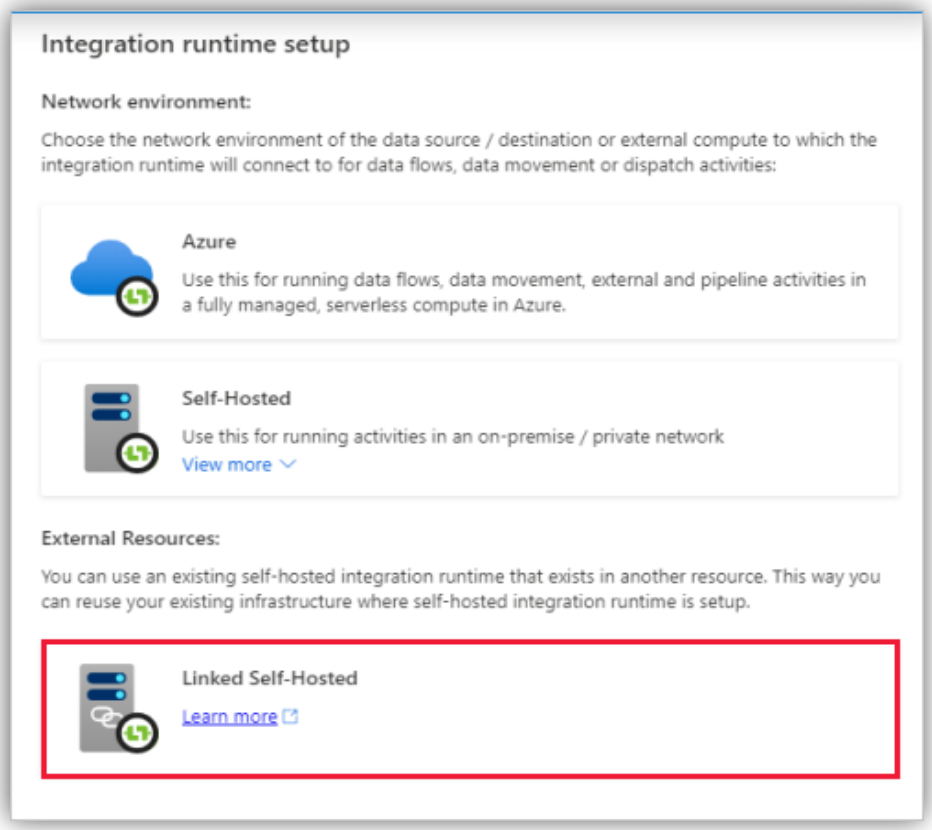




In the data factory to which the permissions were granted, create a new self-hosted IR (linked) and enter the resource ID.

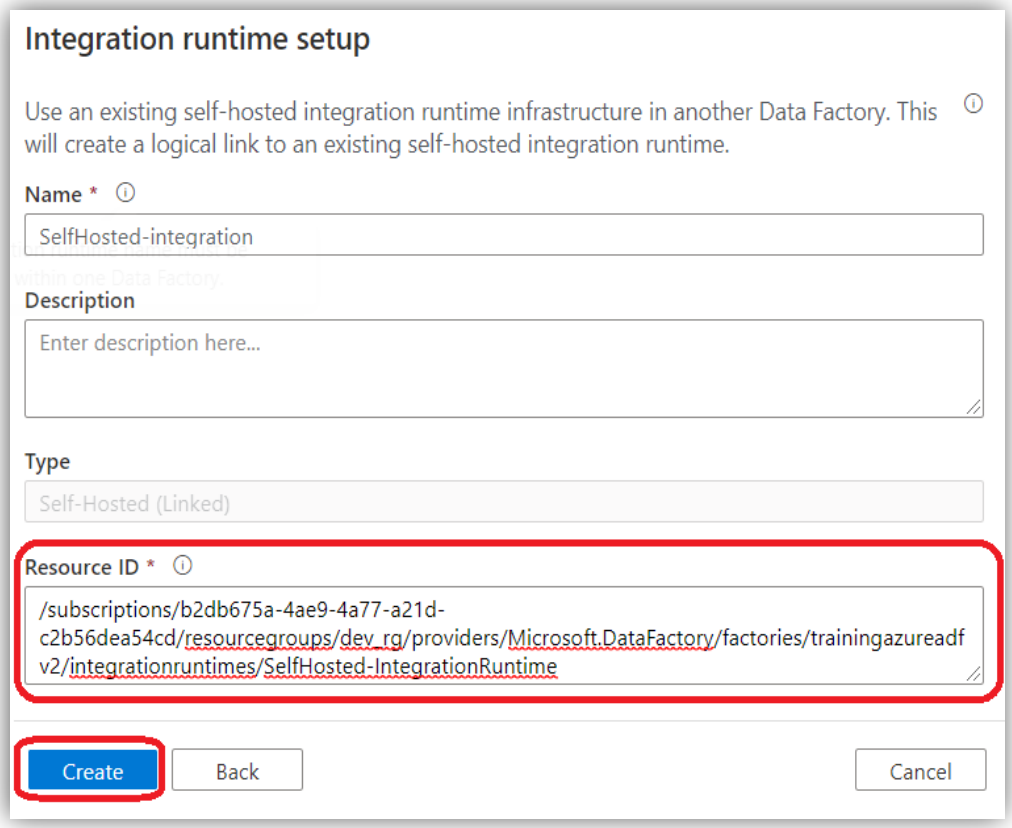
Azure Data Factory Advanced Interview Questions And Answers **Youtube: pyspark telugu**

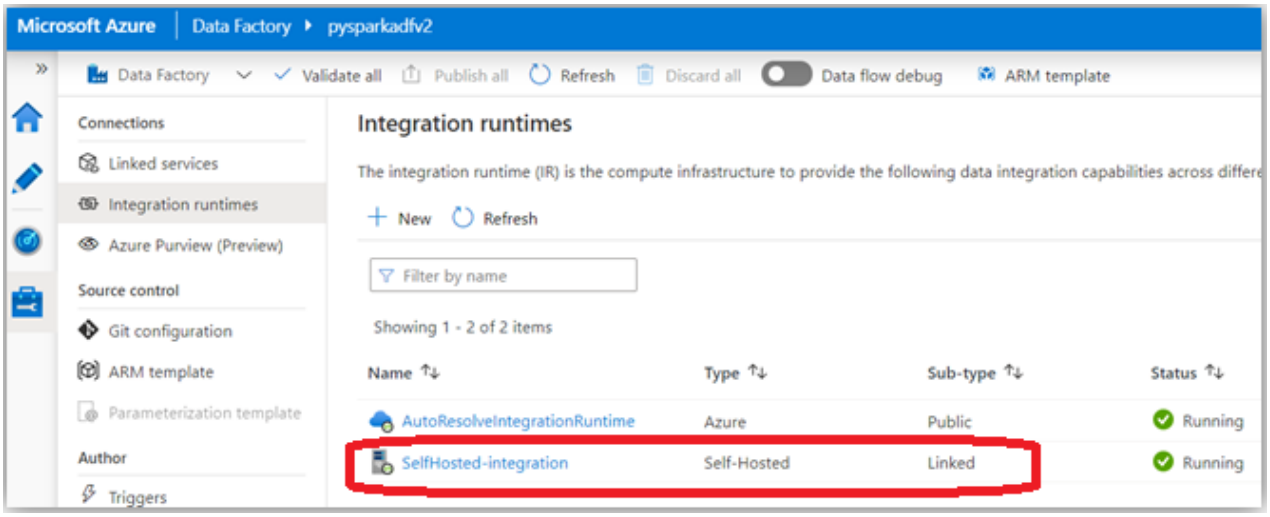




Rename Integration runtime name as " Self-Hosted-Integration " And Paste Resource ID in down window. Then Create. It will be created new Share self-hosted Integration runtime in another Data factory

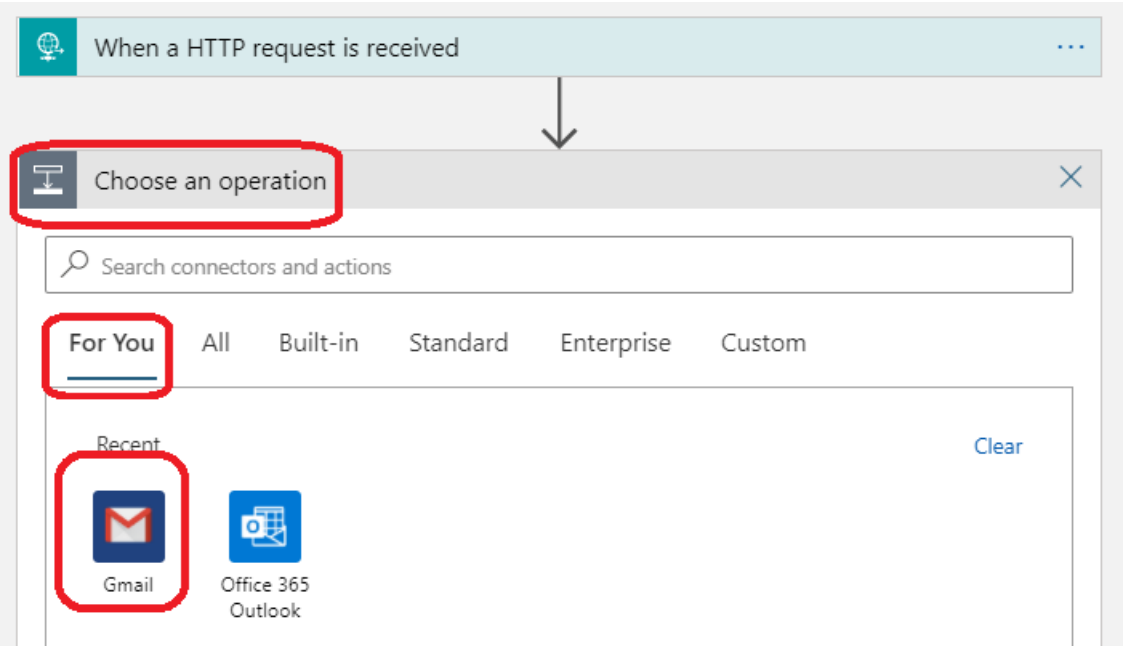
Azure Data Factory Advanced Interview Questions And Answers Youtube: pyspark telugu

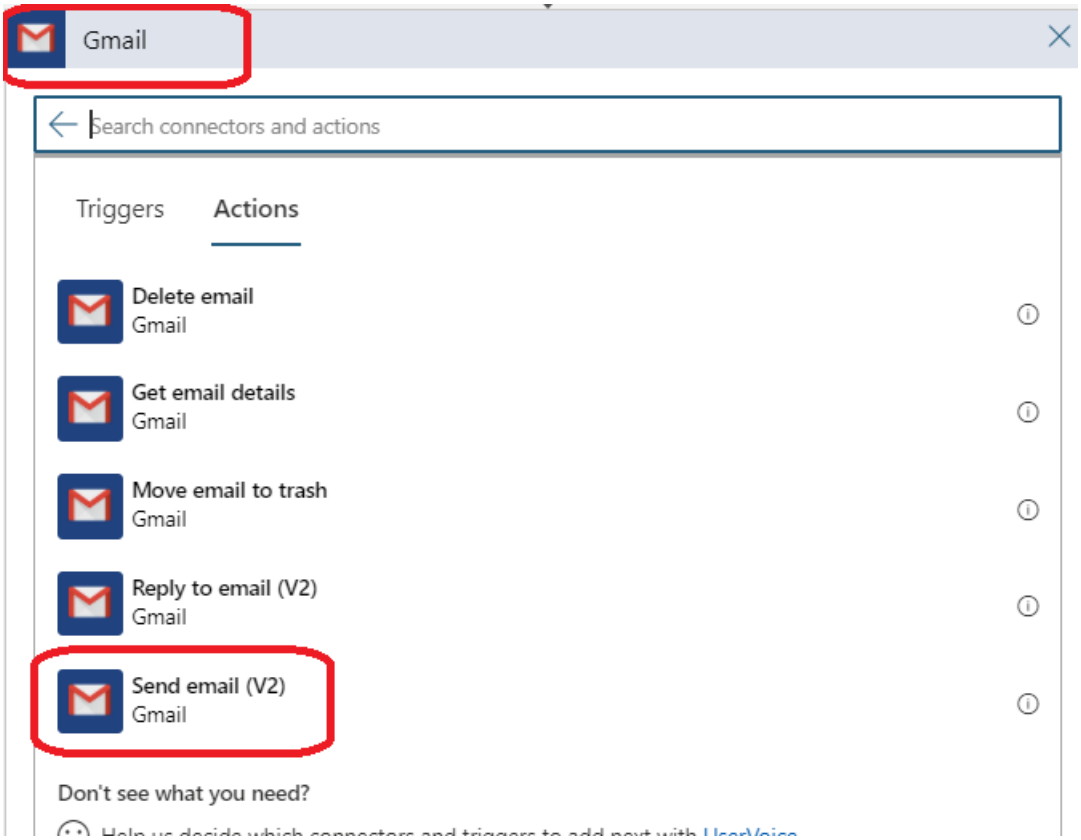




6) What is Logic App? Or How to send an email notification in Azure Data Factory? Or what is WEB Activity and when we can use this activity?

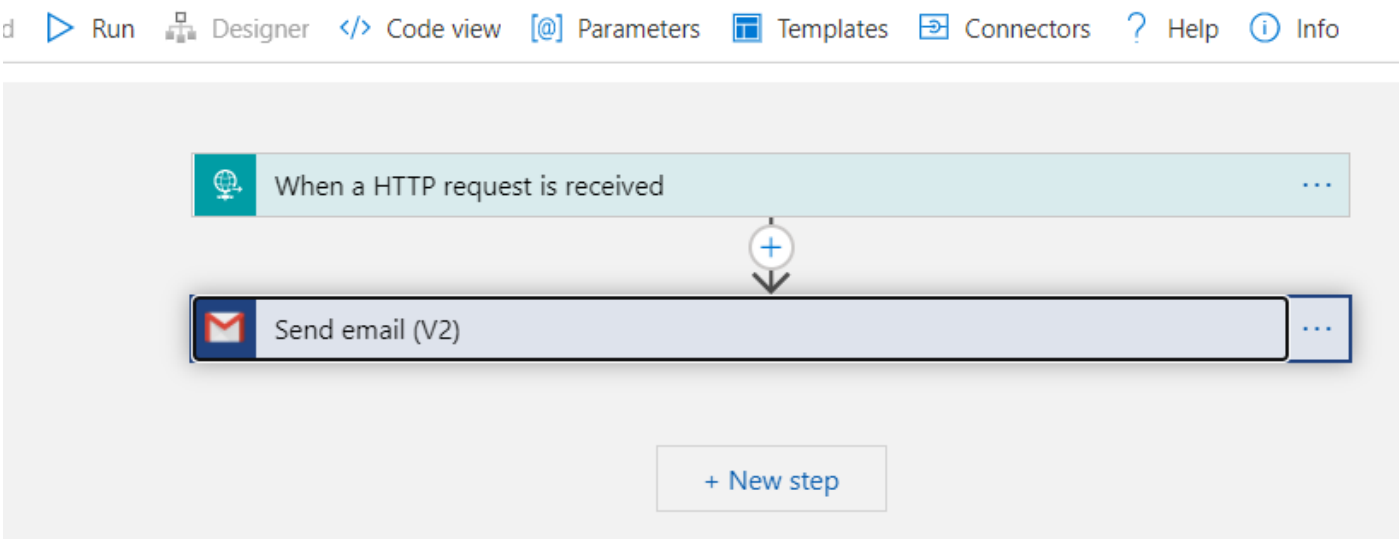
WE can use this for multiple scenarios. Logic App (email) and ADF (error handling). The communication between these two Azure parts is done with a JSON message via an HTTP request (post). The JSON message contains the name of the Data Factory and the pipeline that failed, an error message and an email address. You could of course hardcode the email address in Logic Apps, but now you can reuse the Logic App for various pipelines or data factories and notify different people.



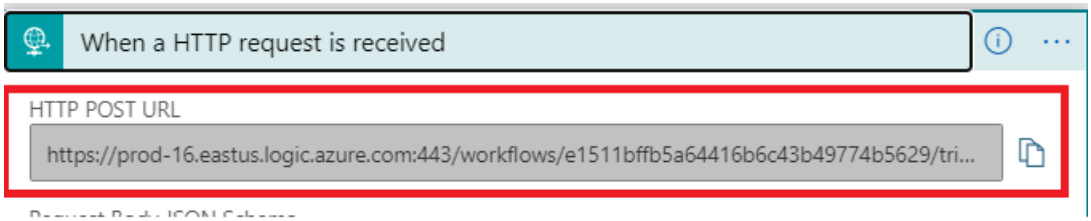


Next, we will add a new step to our Logic App, called “Send an email”. I will use gmail but if you want to use another email provider pick that one.

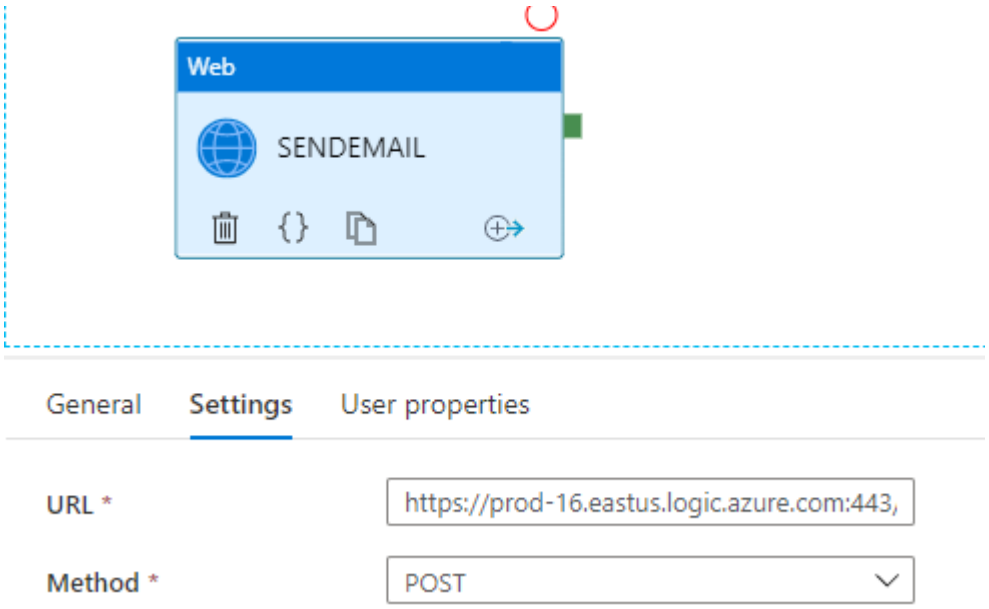
It’s the first time you connect Gmail account on Azure? Then you need to connect your Gmail account to Azure by signing in. (Note: allow pop-ups in your browser.)



After creation of Azure Logic App and saving the Logic App, Azure created an endpoint URL for our Logic Apps, you'll find in the first step. Copy this URL to a notepad, we'll need this later.



Now add an Web activity to the pipeline and rename it.

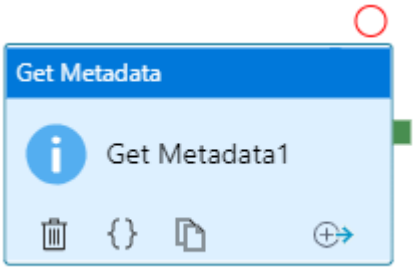


7) What is Get Metadata Activity? When we can use this? OR How to get folders and filenames at dynamically?

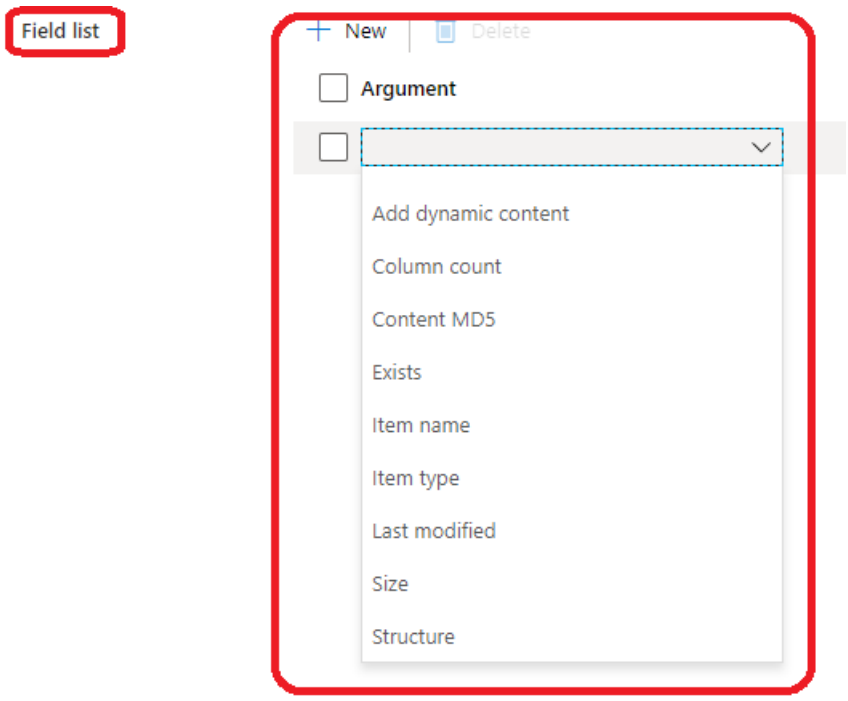
You can use the Get Metadata activity to retrieve the metadata of any data in Azure Data Factory. You can use this activity in the following scenarios:

Validate the metadata of any data.
Trigger a pipeline when data is ready/available.
The following functionality is available in the control flow:

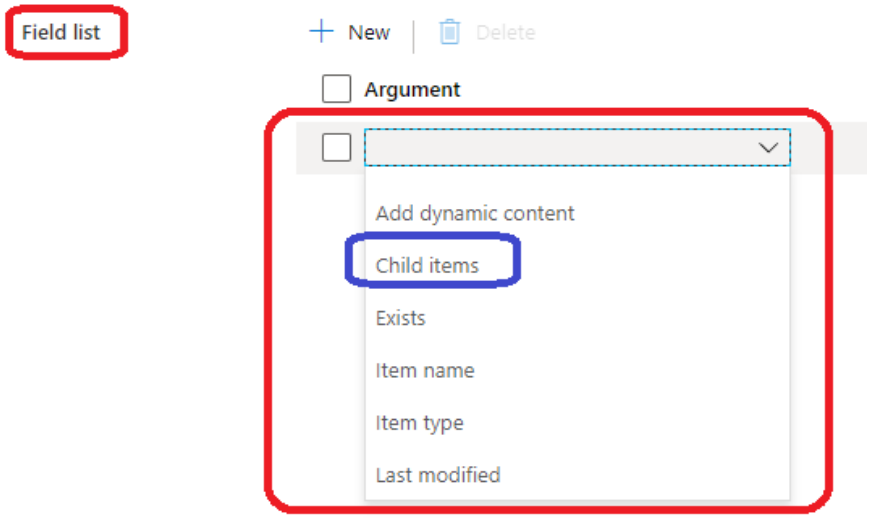
You can use the output from the Get Metadata activity in conditional expressions to perform validation.
You can trigger a pipeline when a condition is satisfied via Do Until looping.



The Get Metadata activity takes a dataset as an input and returns metadata information as output. Currently, the following connectors and corresponding retrievable metadata are supported. The maximum size of returned metadata is around 4 MB.



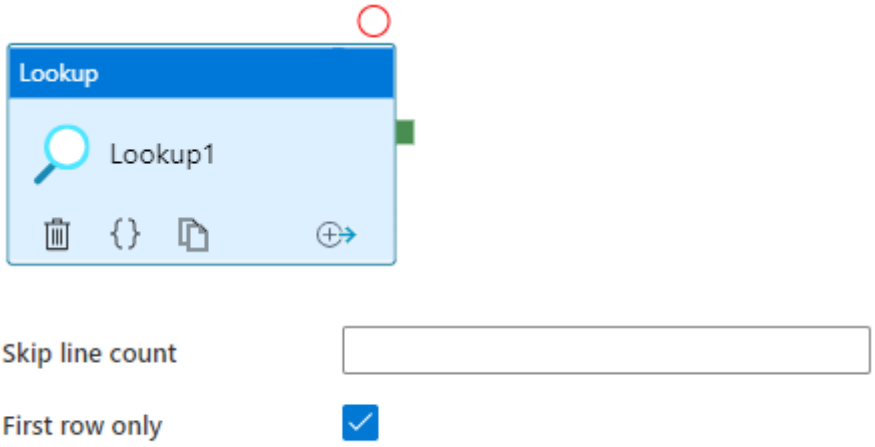
child Items List of subfolders and files in the given folder. Applicable only to folders. Returned value is a list of the name and type of each child item.



8) What is Lookup Activity and when we can use this activity?

Lookup activity can retrieve a dataset from any of the Azure Data Factory-supported data sources. Use it in the following scenario:

Dynamically determine which objects to operate on in a subsequent activity, instead of hard coding the object name. Some object examples are files and tables.



Use the Lookup activity result
The lookup result is returned in the output section of the activity run result.

When firstRowOnly is set to true (default), the output format is as shown in the following code. The lookup result is under a fixed firstRow key. To use the result in subsequent activity, use the pattern of @{activity('LookupActivity').output.firstRow.table}.

9) If you are running more no of pipelines and its taking longer time to execute. How to resolve this type of issues?

We can go with splitting pipelines batches wise and create multiple integration runtimes. Then those loads will be shared by multiple integration runtimes and we can improve the load performance of more no of pipelines.

10) What is auto resolve integration runtime in azure data factory?

AutoResolveIntegrationRuntime. This is the default integration runtime, and the region is set to auto-resolve. That means that Azure Data Factory decides the physical location of where to execute activities based on the source, sink, or activity type.

11) Data Factory supports three types of triggers. Mention these types briefly

- The **Schedule trigger** that is used to execute the ADF pipeline on a wall-clock schedule
- The **Tumbling window** trigger that is used to execute the ADF pipeline on a periodic interval, and retains the pipeline state
- The **Event-based trigger** that responds to a blob related event, such as adding or deleting a blob from an Azure storage account

Pipelines and triggers have a **many-to-many relationship** (except for the tumbling window trigger).**Multiple triggers can kick off a single pipeline**, or **a single trigger can kick off multiple pipelines**.

12) Any Data Factory pipeline can be executed using three methods. Mention these methods

- Under **Debug mode**
- Manual execution using **Trigger now**
- Using an added **scheduled**, **tumbling window** or **event trigger**

13) How to load data whenever we receive a file in azure data factory? Or How to run a pipeline if we receive a file or if we delete a file?

Using Event-based trigger we can solve above requirement.

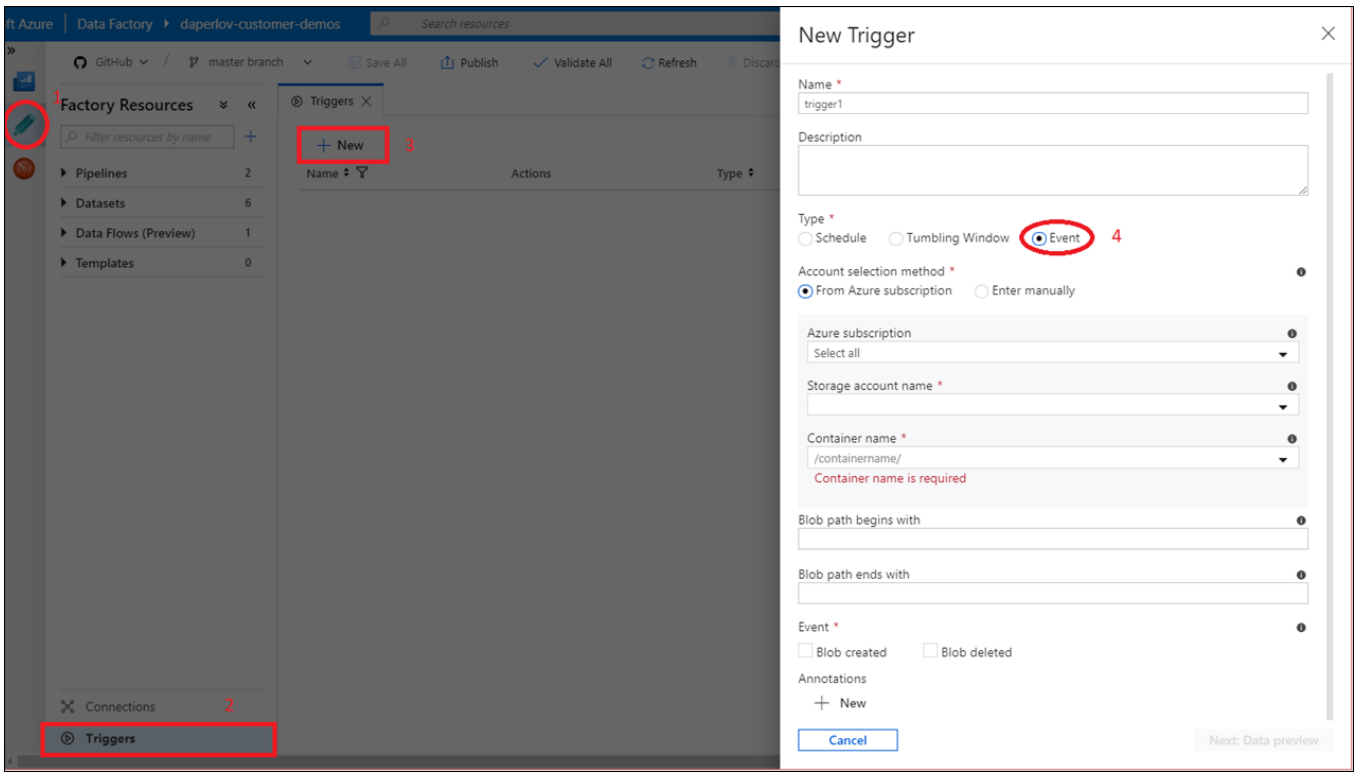
An event-based trigger runs pipelines in response to an event, such as the arrival of a file, or the deletion of a file, in Azure Blob Storage.

Steps to create Event Based Trigger:

In the bottom-left corner, click on the Triggers button

Click + New which will open up the create trigger side nav

Select trigger type Event

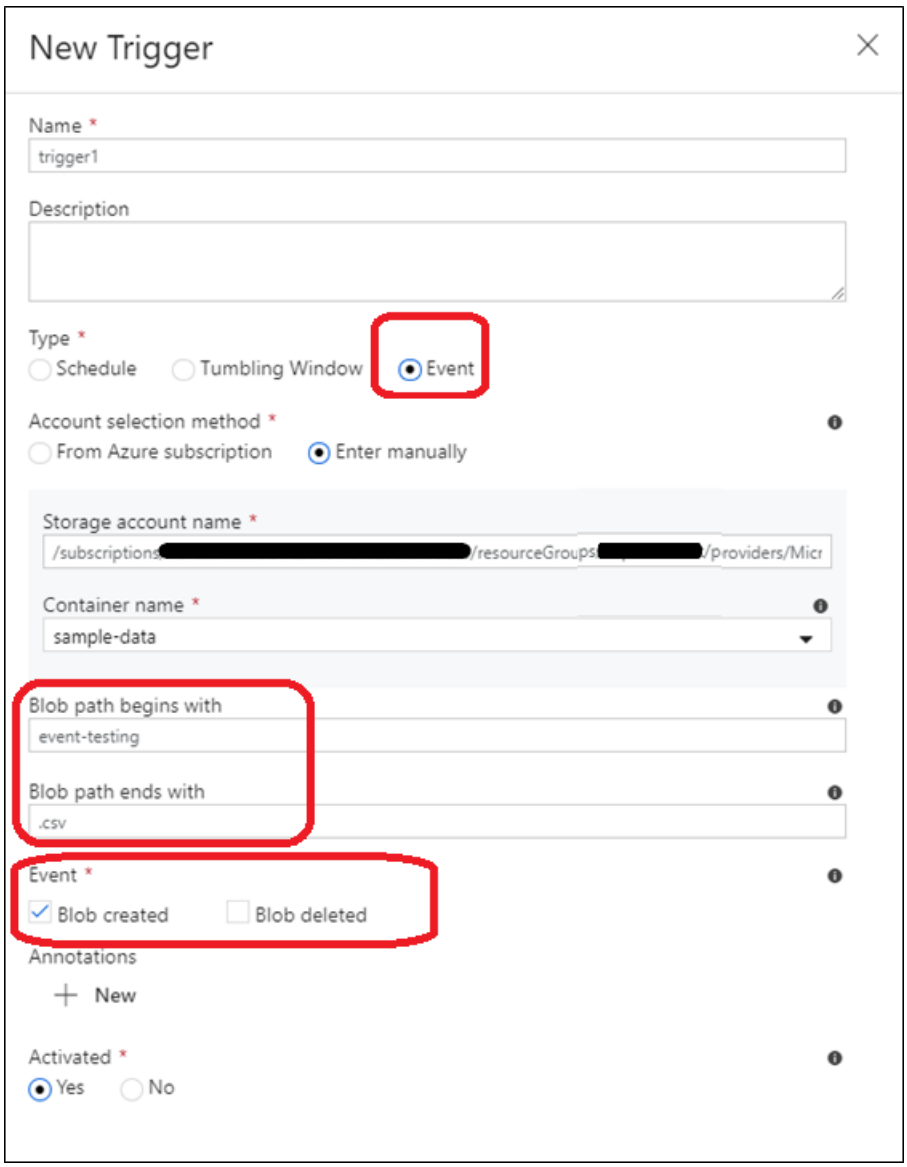


Blob path begins with: The blob path must start with a folder path. Valid values include `2018/` and `2018/april/shoes.csv`. This field can't be selected if a container isn't selected.

Blob path ends with: The blob path must end with a file name or extension. Valid values include `shoes.csv` and `.csv`. Container and folder name are optional but, when specified, they must be separated by a `/blobs/` segment. For example, a container named 'orders' can have a value of `/orders/blobs/2018/april/shoes.csv`. To specify a folder in any container, omit the leading '/' character. For example, `april/shoes.csv` will trigger an event on any file named `shoes.csv` in folder a called 'april' in any container.

Note: Blob path **begins with** and **ends with** are the only pattern matching allowed in Event Trigger. Other types of wildcard matching aren't supported for the trigger type.

Select whether your trigger will respond to a **Blob created** event, **Blob deleted** event, or both. In your specified storage location, each event will trigger the Data Factory pipelines associated with the trigger



14) Difference between Scheduled Trigger and Tumbling window trigger?

The tumbling window trigger and the schedule trigger both operate on time heartbeats. How are they different?

The **tumbling window trigger run** waits for the **triggered pipeline run** to finish. Its run state reflects the state of the triggered pipeline run. For example, if a triggered pipeline run is cancelled, the corresponding tumbling window trigger run is marked cancelled. This is different from the "fire and forget" behavior of the **schedule trigger**, which is marked successful as long as a pipeline run started.

Backfill scenarios

Tumbling Window: Supported. Pipeline runs can be scheduled for windows in the past.

Scheduled Trigger: Not supported. Pipeline runs can be executed only on time periods from the current time and the future.

Pipeline-to-trigger relationship

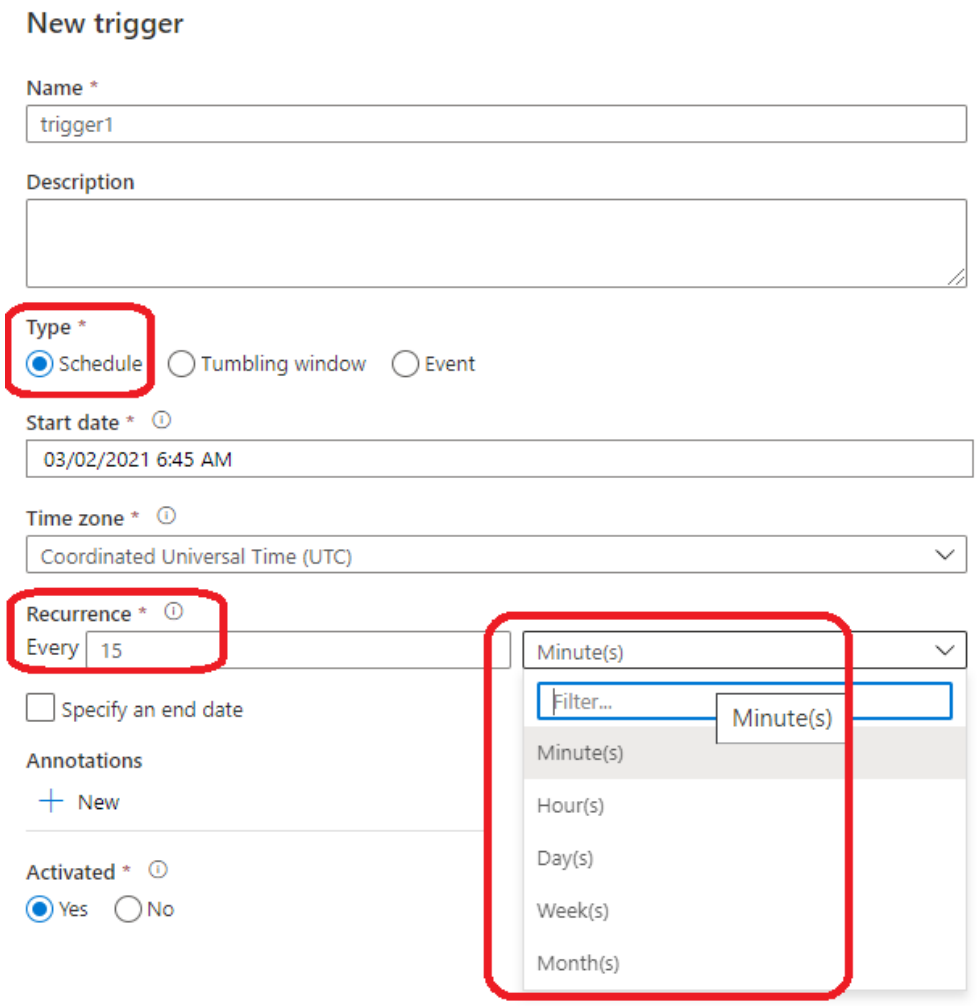
Tumbling Window: Supports a one-to-one relationship. Only one pipeline can be triggered.

Scheduled Trigger: Supports many-to-many relationships. Multiple triggers can kick off a single pipeline. A single trigger can kick off multiple pipelines.

Note:

In order to build a dependency chain and make sure that a trigger is executed only after the successful execution of another trigger in the data factory,

Azure Data Factory Advanced Interview Questions And Answers **Youtube:** [pyspark telugu]



Azure Data Factory Advanced Interview Questions And Answers **Youtube: pyspark telugu**