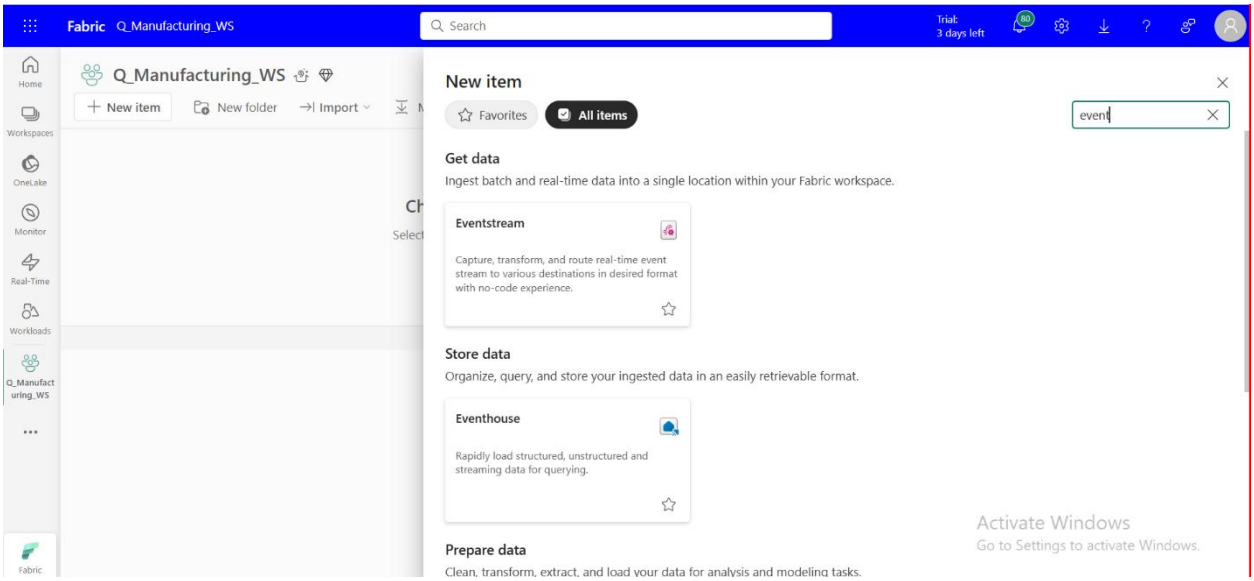
Creating a Real Time Intelligence Solution and Dashboard on Microsoft Fabric

1. Creating an Eventstream service

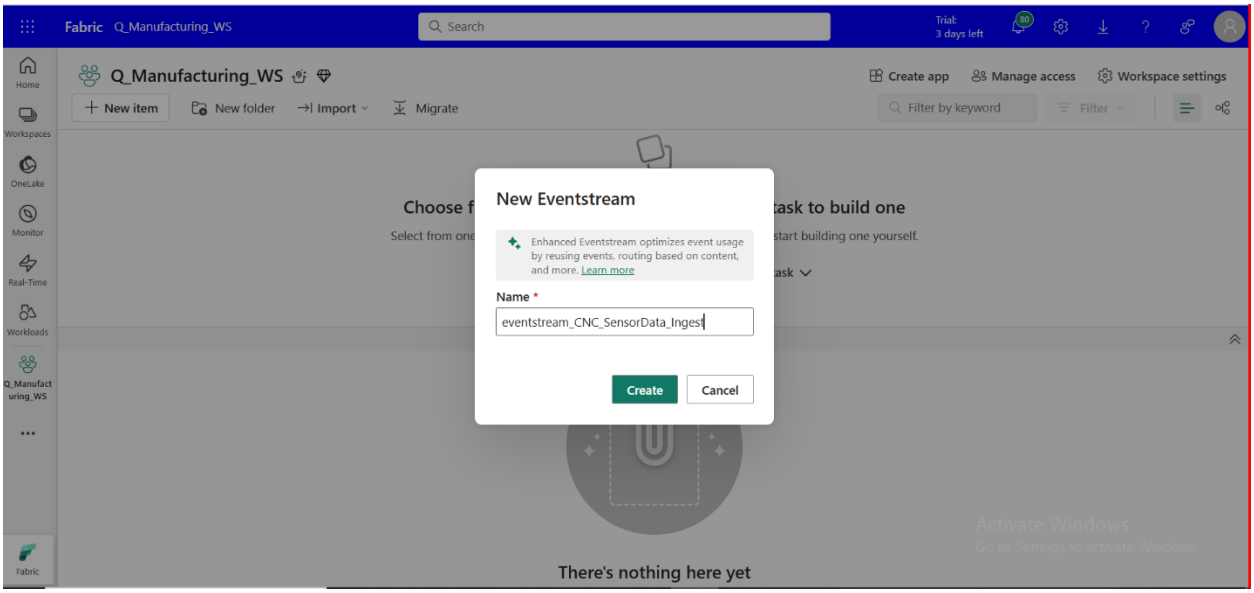
An Eventstream allows you to capture, transform, and route streaming data to one or more destinations in real time.

1. In your Fabric workspace, click on the **“New”** button (represented by a “+” icon).
2. In the search bar, type **“Eventstream”**.
3. From the search results, select **Eventstream** to begin the creation process



Give your Eventstream a descriptive name that reflects its purpose. It is strongly recommended to use a naming convention that indicates:

- The **source/destination** of the data (e.g., IoTDevice, APIFeed, KQLDatabase)
- Or the **function** of the Eventstream (e.g., TransformSensorData, RouteAlerts)
- Here we are naming the Eventstream as ‘eventstream_CNC_SensorData_Ingest’
- Once you’ve entered a name, click **Create** to initialize the Eventstream

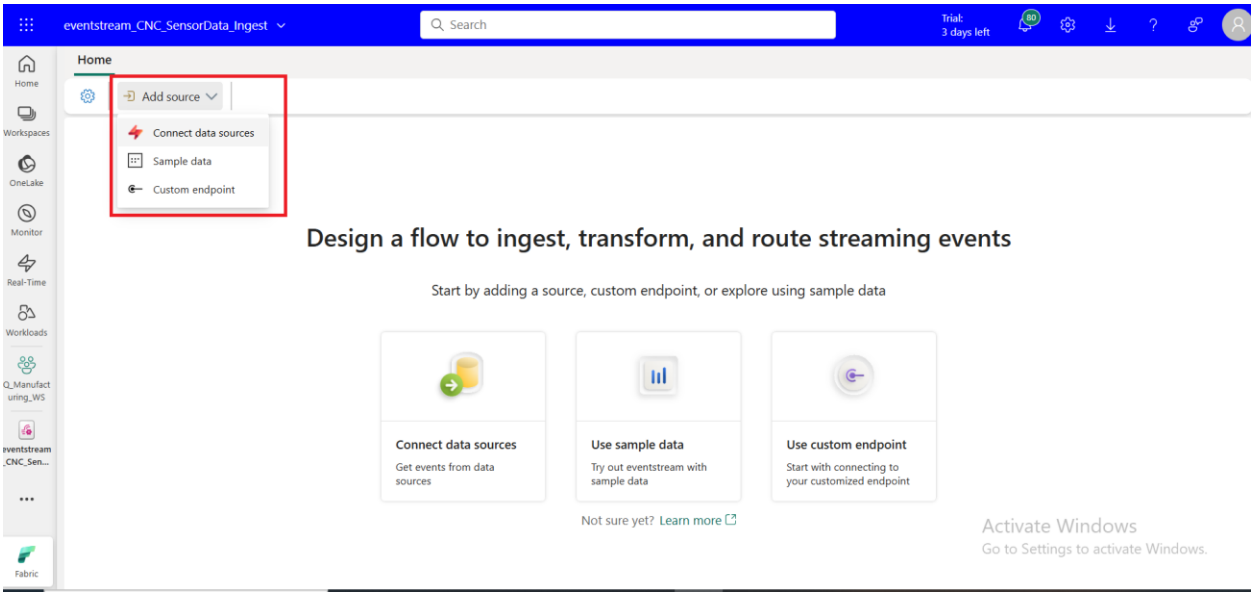


2. Configuring the Eventstream

After the Eventstream is created, you need to configure where the streaming data will come from.

1. In the Eventstream canvas, click on **“Add source.”**
2. From the menu, select **“Connect data sources.”**

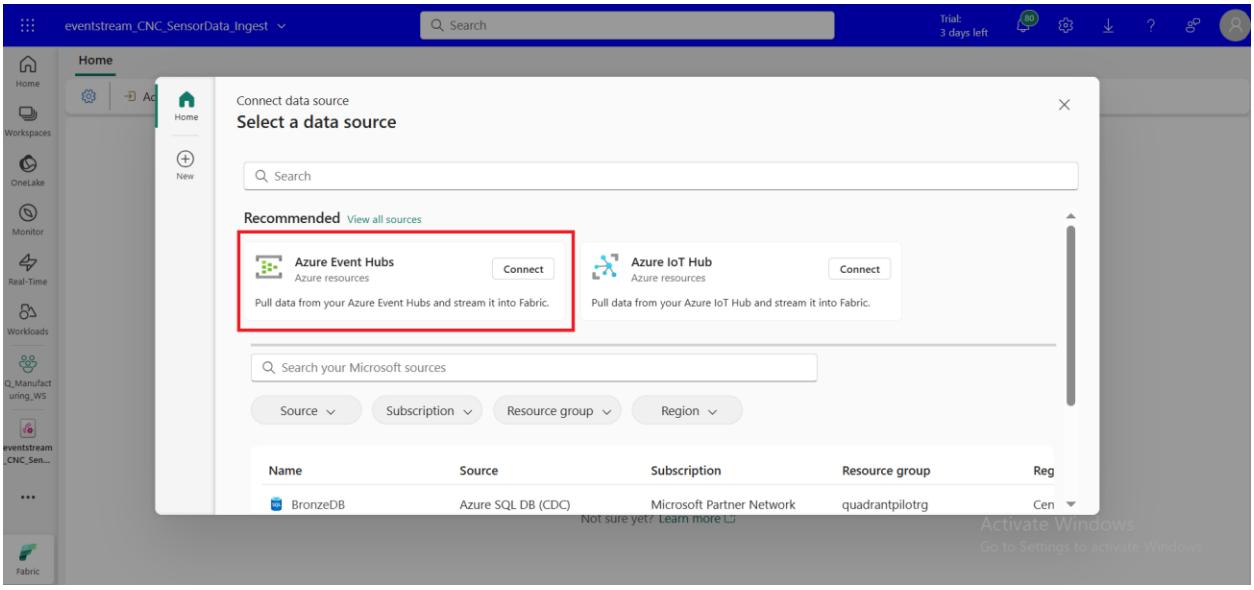
This will allow you to choose the type of streaming data source you want to connect (e.g., Event Hub, IoT Hub, or custom sources).



3. Connect to Data Source

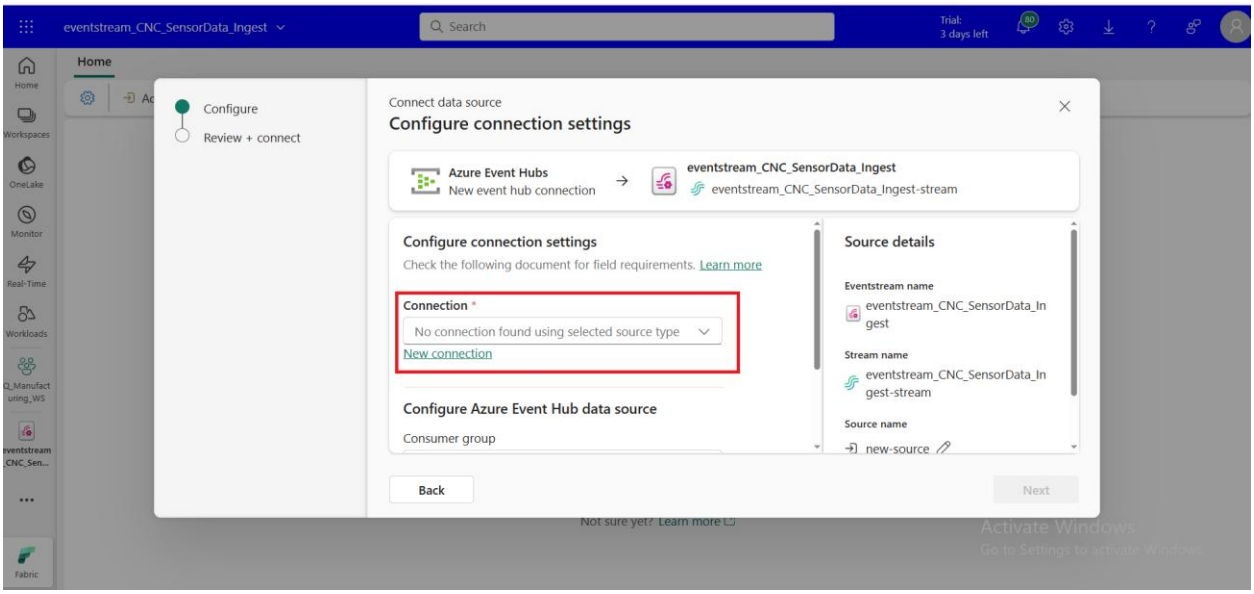
In this example, the streaming data originates from **Azure Event Hub**.

1. In the **“Connect data sources”** pane, use the search bar to find and select **“Azure Event Hubs.”**
2. Click **“Connect.”**

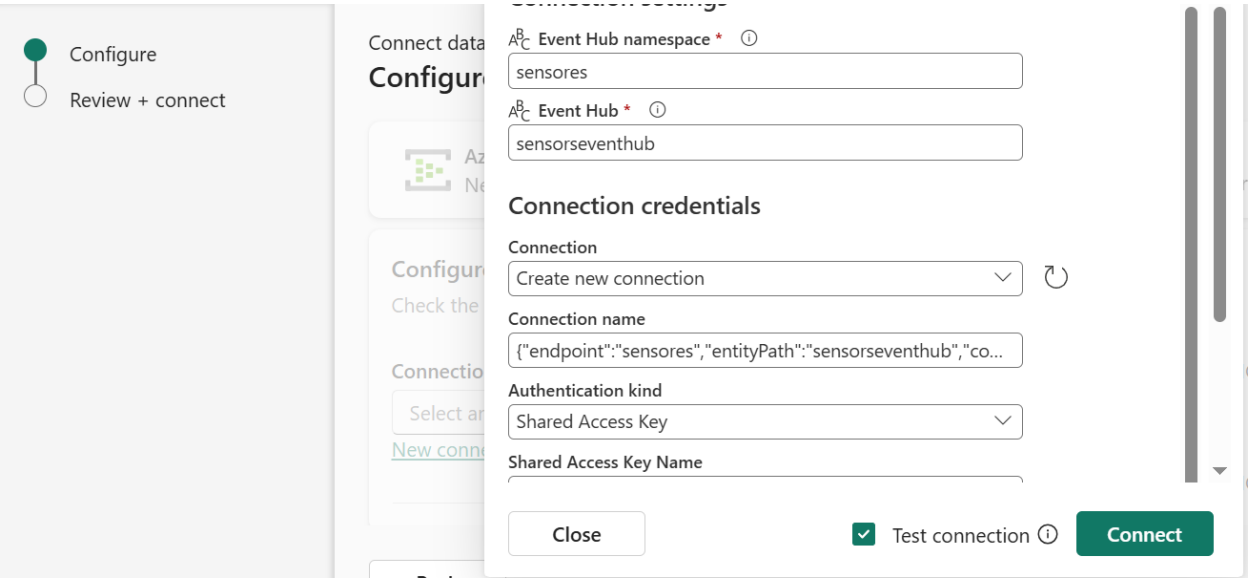


In the **“Configure connection settings”** pane that appears, choose **“New connection.”**

This will allow you to establish a connection to your Event Hub instance.



4. Establish Connection to Data Source

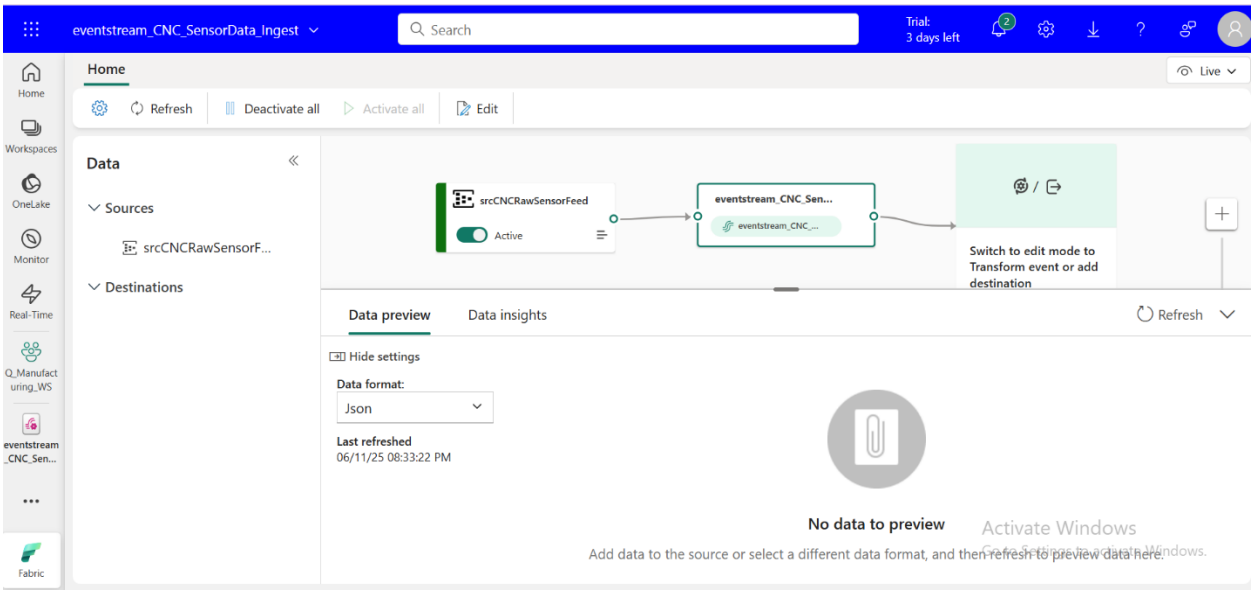


To establish a new connection with your data source, you need to provide the required authentication details and configuration parameters of the data source.

For this example, use the following parameters of the Event Hub to populate the New connection pane:

- **Event Hub namespace:** *sensores*
- **Event Hub:** *sensorseventhub*
- **Connection:** 'Create new connection'
- **Connection name:** This gets auto-populated, so leave as is
- **Authentication Kind:** Shared Access Key
- **Shared Access Key Name:** SensorSAS
- **Shared Access Key:** ic+bx/gEKRz+eDF8NpPRur+7efedMSrQV+AEhA2Vih0=
- **Privacy Level:** None

Once all fields are filled in, click **"Next"** or **"Connect"** (as applicable) to establish the connection. The connection will show up in the Eventstream canvas as shown below.

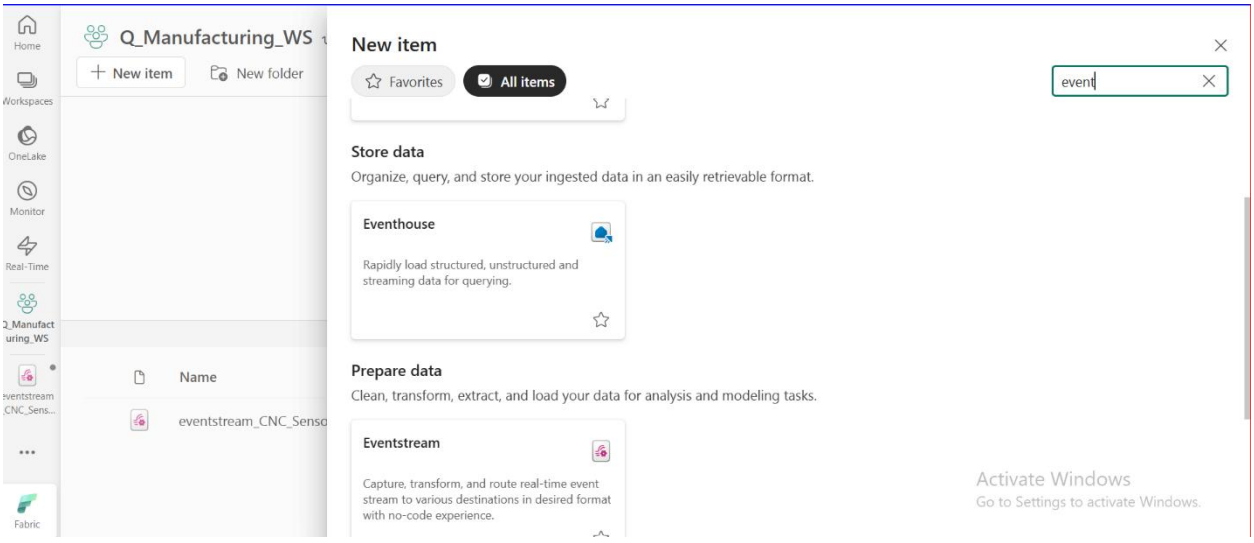


5. Creating destinations

Now that the source is configured, the next step is to set up **destinations** where your streaming data will be stored, routed, or visualized.

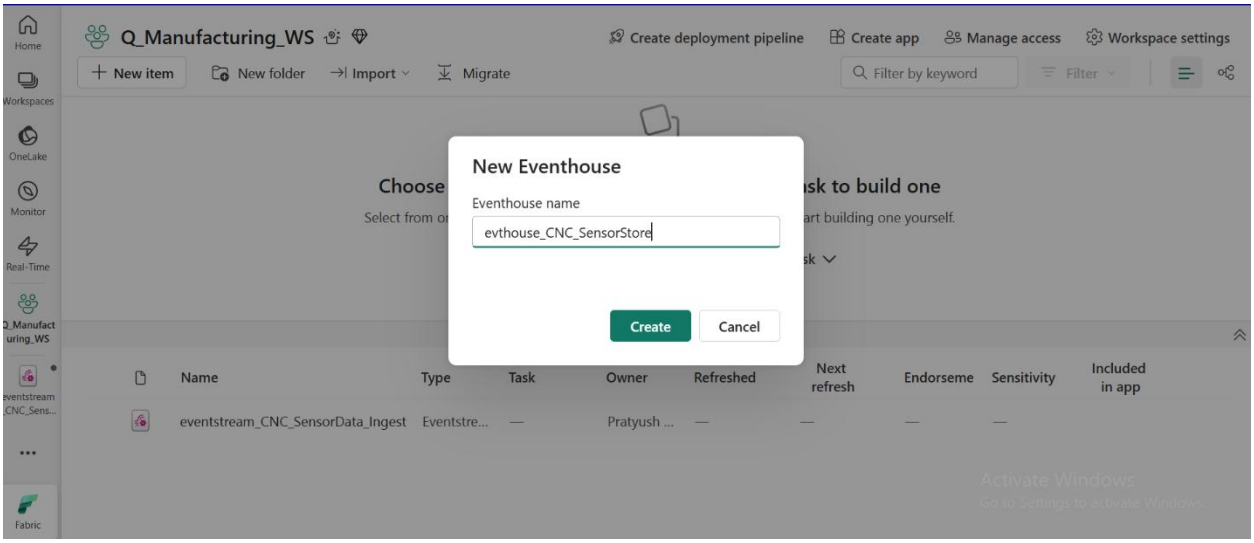
In this example, we will store the data in an **Eventhouse and a lakehouse (for long term storage)**.

1. Go to your **Fabric workspace**.
2. Click the **"New Item"** button and use the search bar to find **"Eventhouse."**

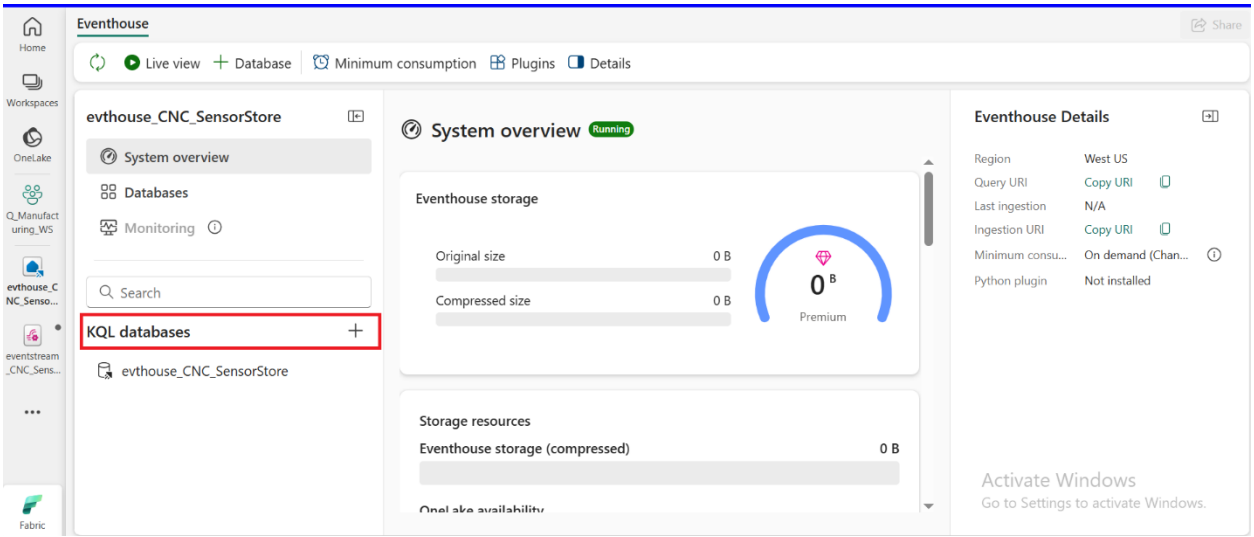


3. Select **Eventhouse** from the search results.

4. Provide a descriptive name for the Eventhouse.
    - For this example, name it: evthouse_CNC_SensorStore.



5. Click **Create** to provision the Eventhouse. Once created, the Eventhouse resource will appear in your workspace, and its structure will look similar to the image shown below. Click on the '+' icon to create a new database. Provide a unique name to the database. In the current example we name it 'kqldb_CNC'

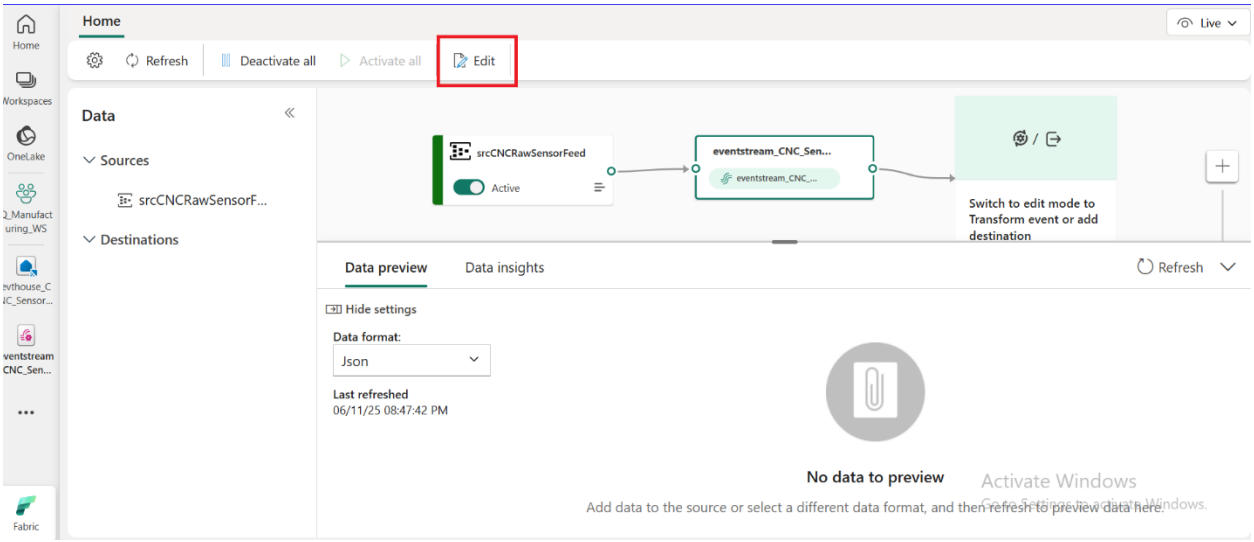


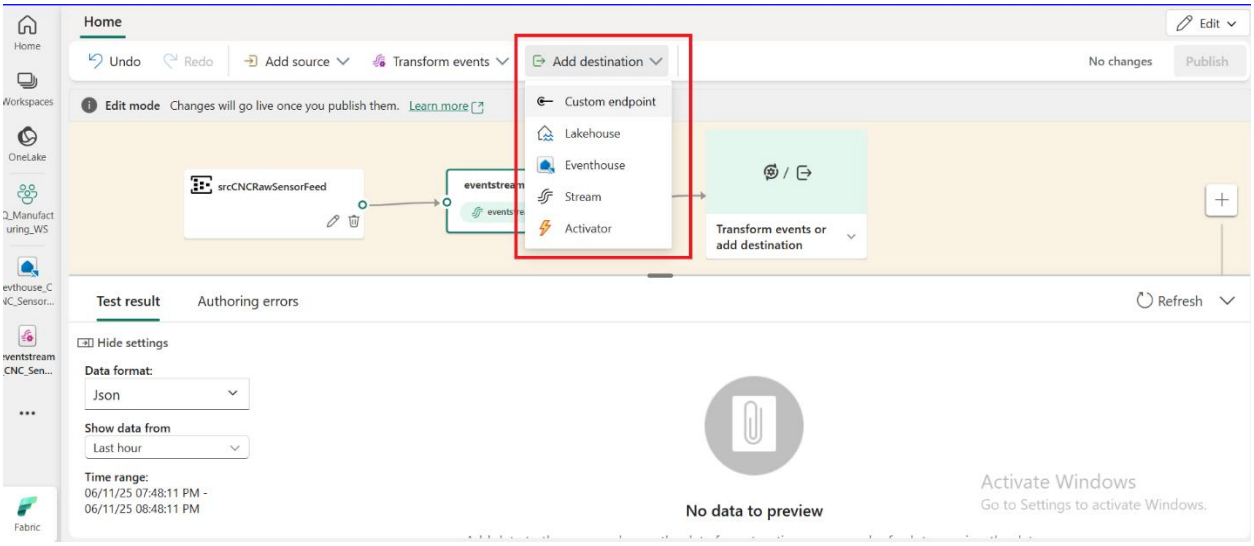
In the similar fashion, create a new Lakehouse from your workspace. In the current example, we named the Lakehouse 'lakeCNCProdSensorData.'

6. Configuring Destinations to the Eventstream

Now that the Eventhouse has been created, configure your Eventstream to route data to it.

1. Navigate back to your **Eventstream** resource.
2. Open the Eventstream and click on the **“Edit”** button to open the canvas view.
3. In the canvas, click on the **“Add destination”** dropdown.
4. From the list of options, select **“Eventhouse.”**
5. In the configuration pane that appears, fill in the following parameters:
    - **Data ingestion mode:**
      Event processing before ingestion
    - **Destination name:**
      evthouseCNCSensorStore *(The name of the Eventhouse created earlier)*
    - **Workspace:**
      Q_Manufacturing_WS *(Your Fabric workspace name)*
6. Click **“Apply”** or **“Save”** to complete the destination setup.
7. Now navigate to the Eventstream and select ‘Edit’ and click on ‘Add Destination’. Choose ‘Eventhouse’ as the destination.





8. Set the configuration parameters as follows,

   Data Ingestion Mode: Event processing before ingestion

   Destination Name: Eventhouse

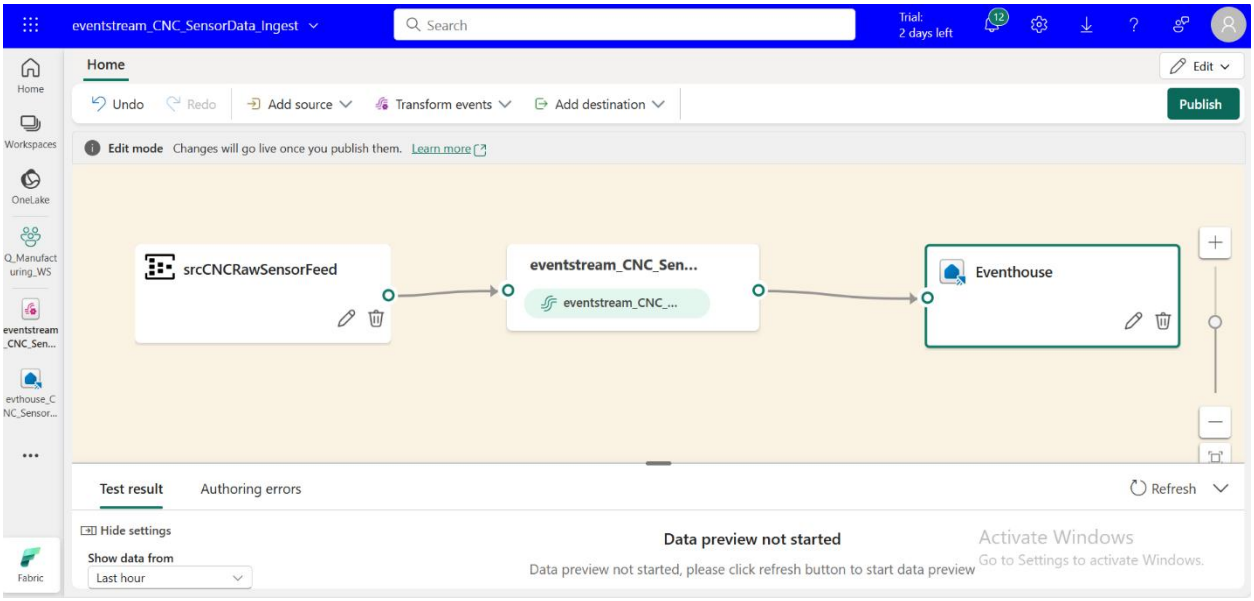
   Workspace: The name you gave. In this case it is ‘Q_Manufacturing_WS’

   Eventhouse: Choose the eventhouse you created. In this case it is ‘evthouse_CNC_SensorStore’

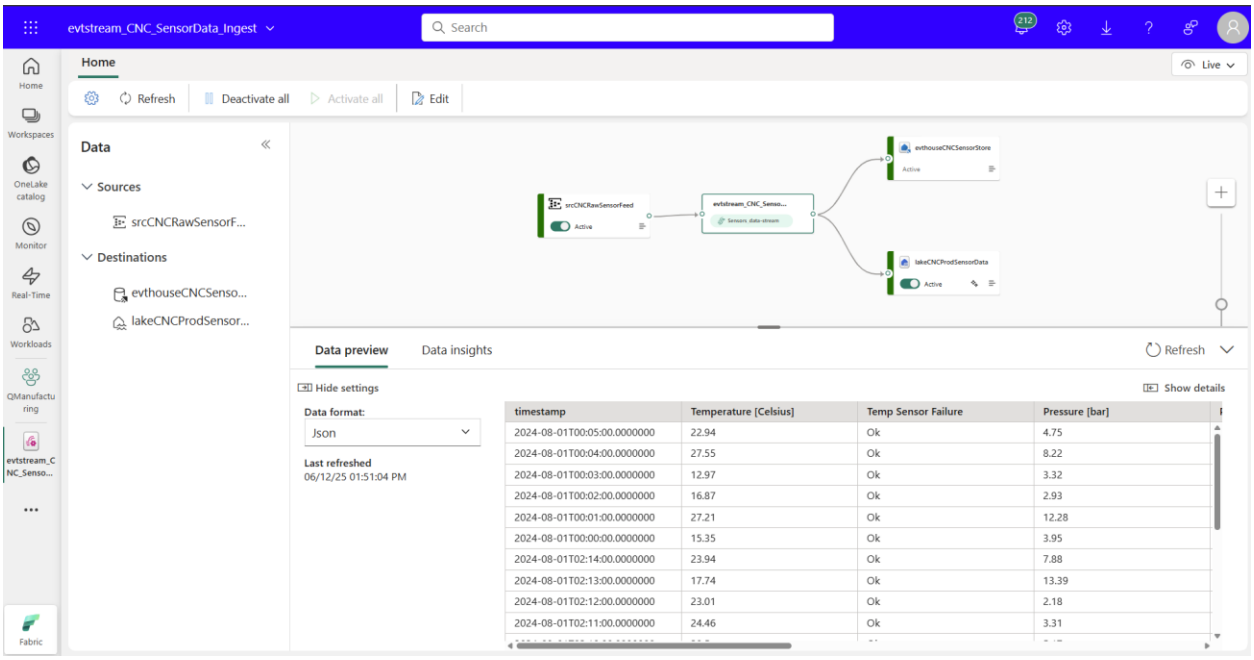
   KQL Database: kqldb_CNC

   KQL Destination Table: Create New: ‘sensortab’

   Once the parameters are defined select ‘Save’ and then Publish.



Repeat the same to add Lakehouse as the destination. The final canvas is as shown below



7. Explore the Eventhouse and KQL Database (Analyze & Transform)

Once the Eventstream is running, sensor data is ingested into the Eventhouse for analysis.

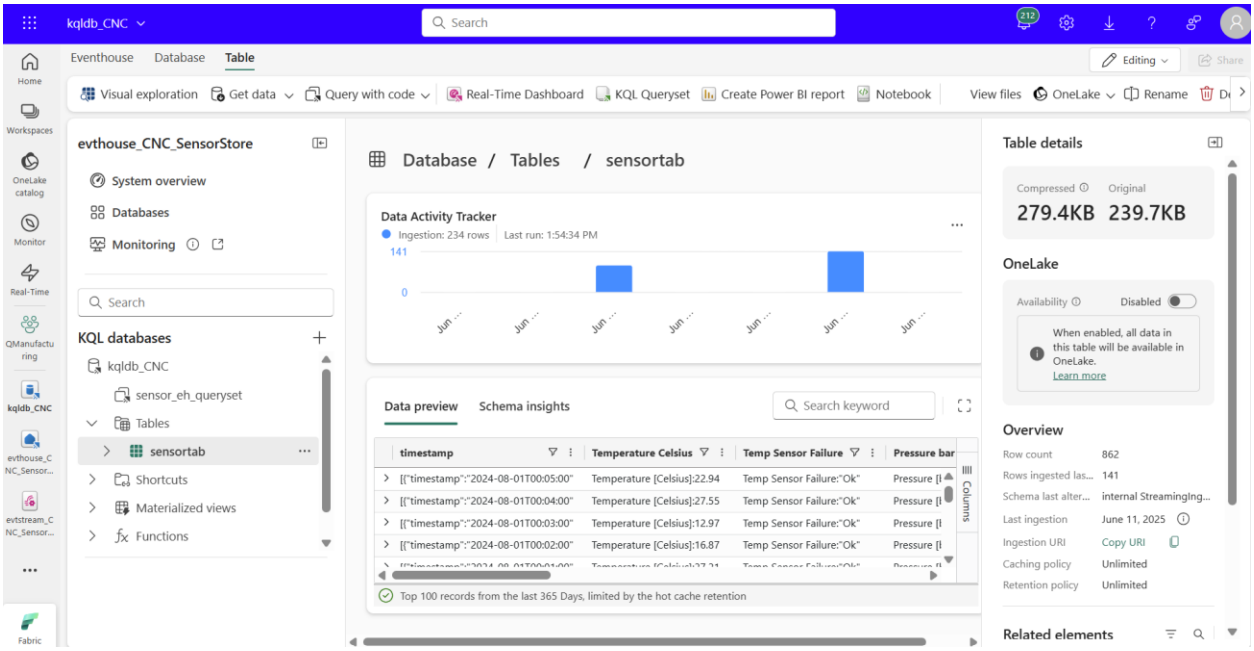
1. **Access the Eventhouse**:
   From your workspace, open the Eventhouse (e.g., evthouse_CNC_SensorStore).
   In the image, the KQL database **kqldb_CNC** is shown under this Eventhouse.

2. **Understand the Eventhouse Role**:
   Eventhouses power real-time data processing in Microsoft Fabric. They handle high-speed ingestion and querying of structured and semi-structured data from continuous streams.

3. **Explore the KQL Database**:
   Inside the Eventhouse, the **KQL database** (e.g., kqldb_CNC) manages your data. It contains:

   - Tables like **sensortab** where ingested sensor data is stored.
   - Data activity metrics, row counts, and ingestion details.

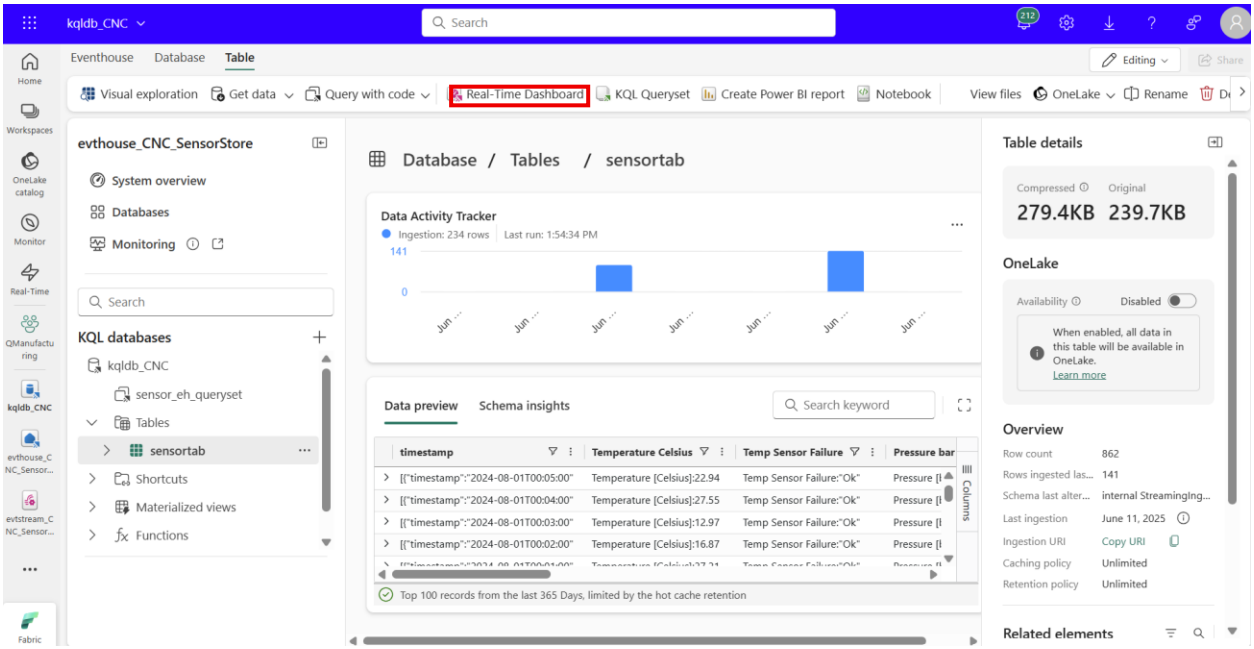
4. **Analyze and Transform the Data**:
   Use tools like *Data preview*, *Schema insights*, and *KQL Queryset* to inspect and analyze your data in real time.
   As shown, the sensortab table stores columns like timestamp, temperature, sensor status, and pressure.



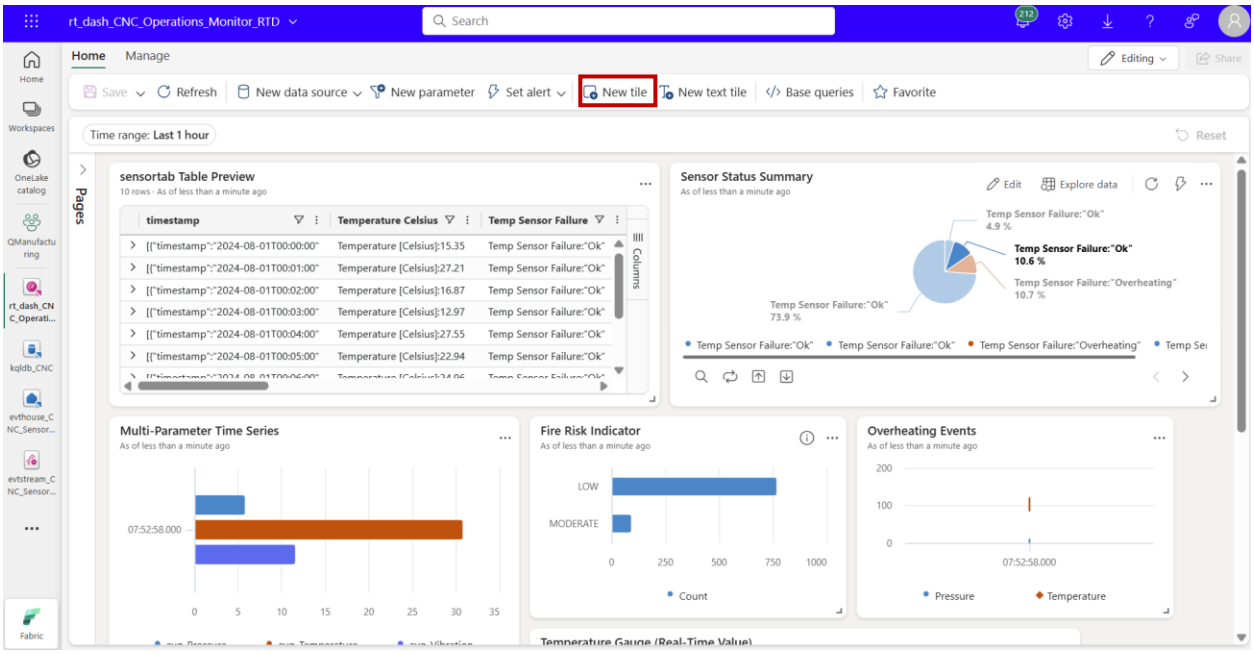
8. Create a Real-Time Dashboard

Once your sensor data is flowing into the KQL database, you can visualize it using real-time dashboards.

1. **Click on “Real-Time Dashboard”**:
   In the toolbar at the top of the Eventhouse interface, select **"Real-Time Dashboard"** to begin building a live visual report.

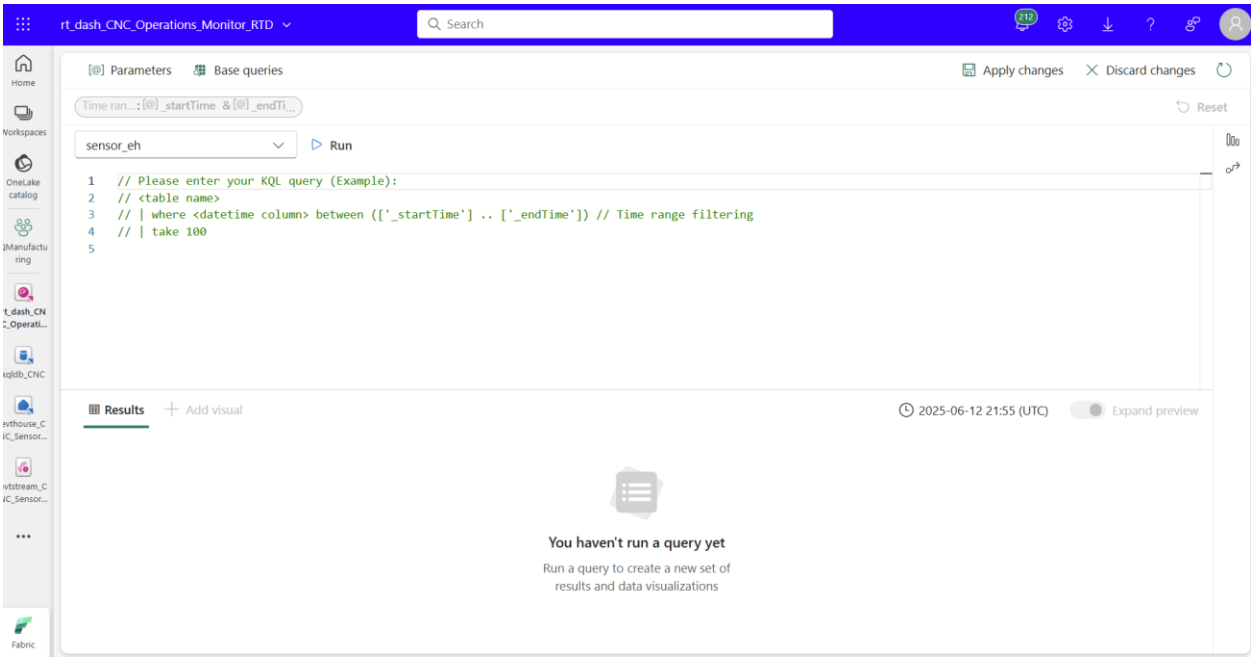


2. Click the **“New tile”** button located at the top center of the dashboard toolbar to begin creating a custom visualization for displaying specific data insights.



3. Write the query according to business need
   eg.
   sensortab

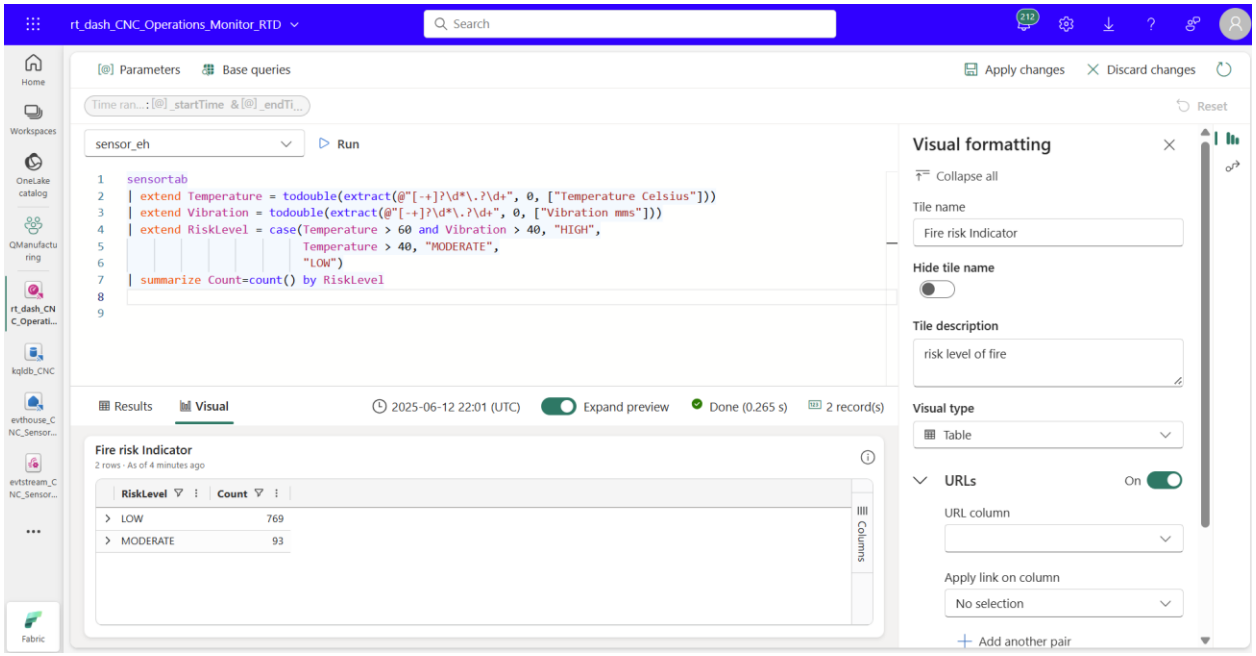
```
| extend Temperature = todouble(extract(@"[-+]?\d*\.?\d+", 0, ["Temperature Celsius"]))

| extend Vibration = todouble(extract(@"[-+]?\d*\.?\d+", 0, ["Vibration mms"]))

| extend RiskLevel = case(Temperature > 60 and Vibration > 40, "HIGH",

                         Temperature > 40, "MODERATE",

                         "LOW")

| summarize Count=count() by RiskLevel
```



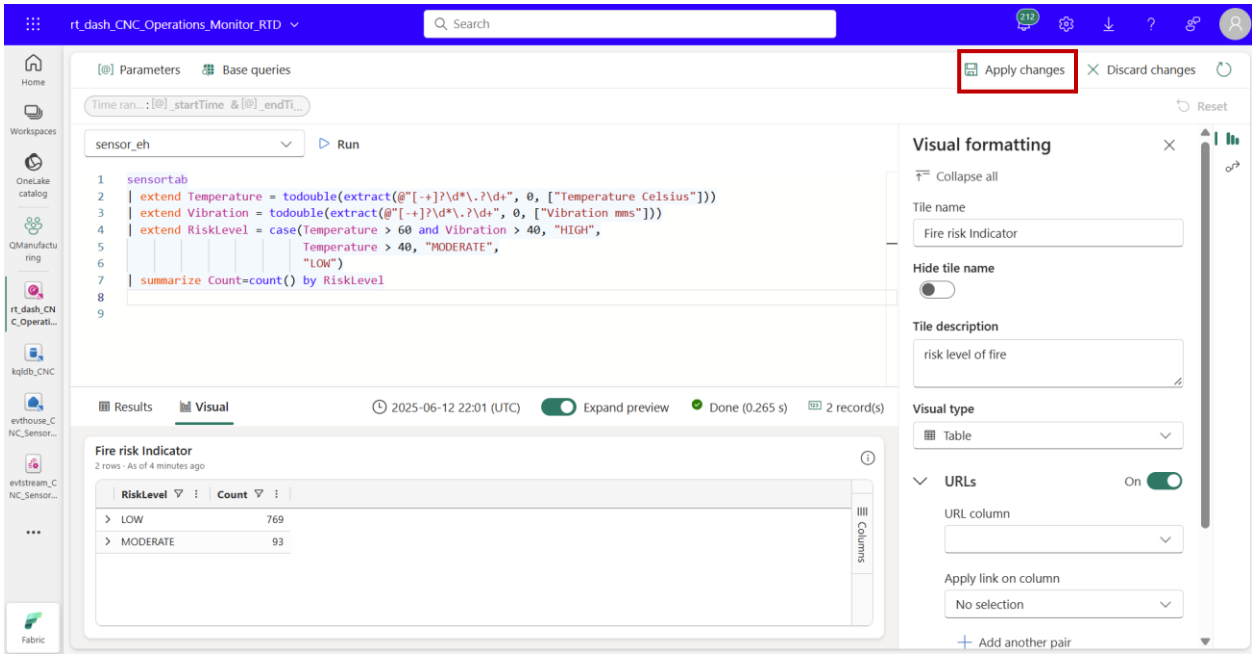
4. Click on run

5. Add Visual, Icon Present beside the results

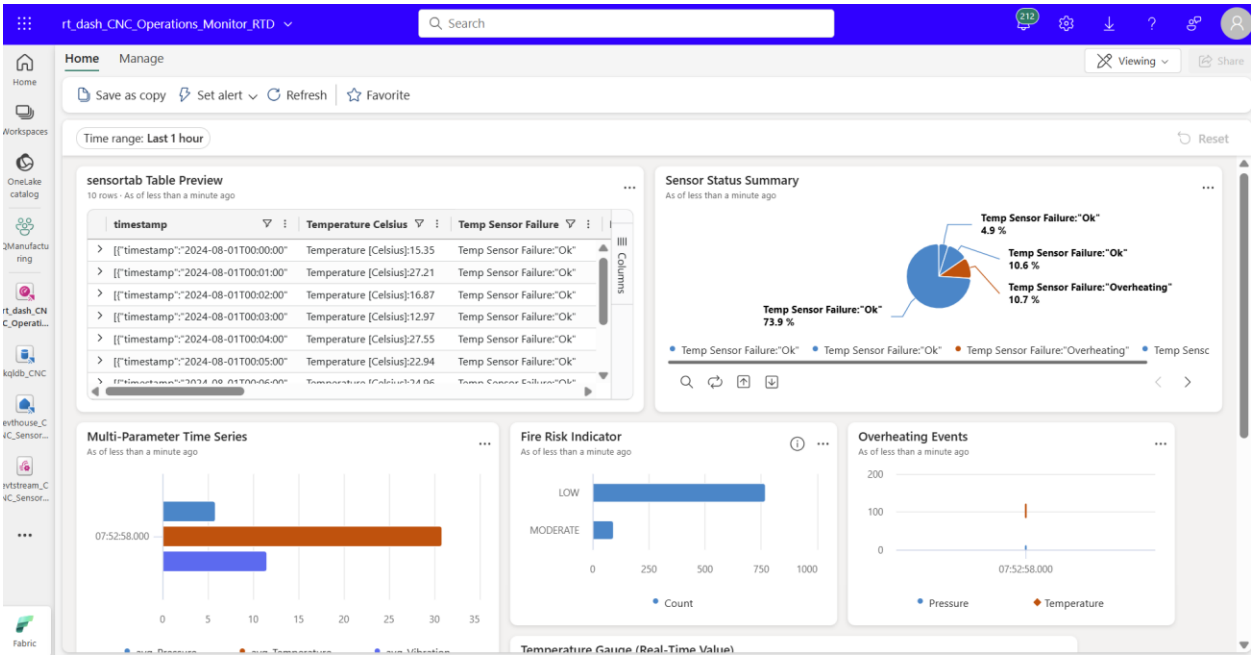
6. Write Tile name, description, URLs and apply conditional formatting



7. click on Apply changes

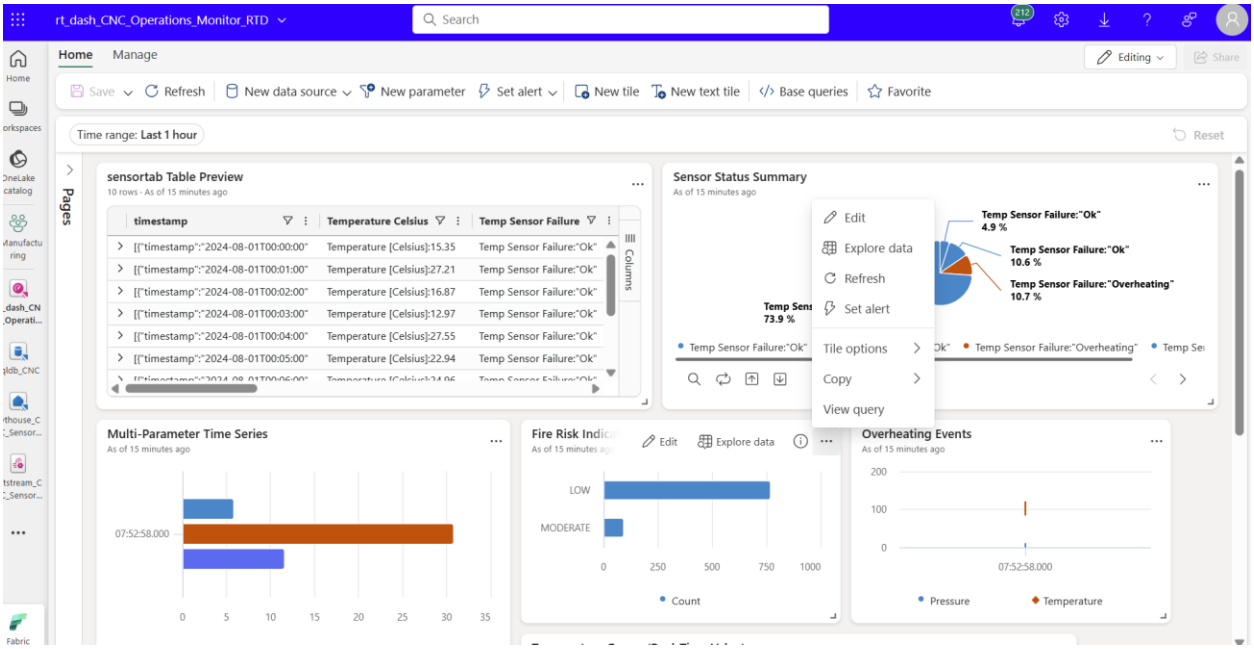


10. Follow the same steps (clicking **“New tile”** and configuring the desired data view) to create and customize **various tiles** across the dashboard



11. Set an Alert on a Tile

Click the **ellipsis (...)** at the top-right corner of the tile, then select **“Set alert”** from the dropdown menu to configure threshold-based notifications for that specific visualization.

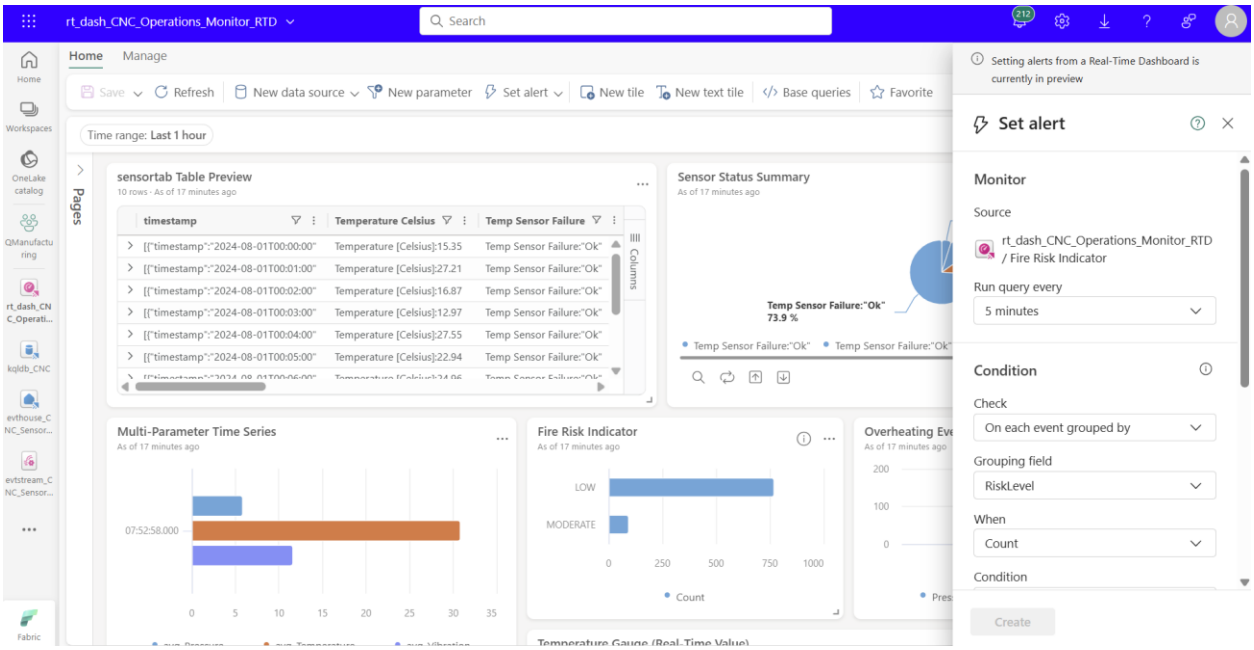


- 10. **Source:**
  rt_dash_CNC_Operations_Monitor_RTD / Fire Risk Indicator
  (This specifies the tile you're monitoring — in this case, the Fire Risk Indicator chart.)
- **Run query every:**
  5 minutes
  (Defines how frequently the system will evaluate the condition for alerts.)

📋 Condition Section

- **Check:**
  On each event grouped by
  (This means the alert checks the grouped values — e.g., by risk level.)
- **Grouping field:**
  RiskLevel
  (It evaluates each unique value in the RiskLevel field — such as "LOW", "MODERATE", etc.)
- **When:**
  Count
  (You are triggering based on how many events occur per group.)
- **Condition:**
  (The next part would specify a threshold, such as "Count is greater than 500". This section is partially hidden in your image, but it will let you pick conditions like:
  - is greater than

- is less than
- equals
  *... and then enter a numeric value.)



11. click on create