Portfolio Project:
Real-Time Intelligence in Microsoft Fabric

Project Overview

This project focused on utilizing Real-Time Intelligence capabilities in Microsoft Fabric to ingest, analyze, and visualize real-time data streams, specifically stock market data. The goal was to create an analytical solution that captures and displays real-time data effectively.

Objectives:

1. **Create a Workspace:**
   - Set up a workspace with Fabric capacity enabled.
2. **Create an Eventstream:**
   - Ingest real-time stock market data from a streaming source.
3. **Create an Eventhouse:**
   - Store the captured data in a table.
4. **Query the Captured Data:**
   - Analyze the ingested data using KQL queries.
5. **Create a Real-Time Dashboard:**
   - Visualize the data in a dashboard.
6. **Create an Alert:**
   - Set up an alert to notify when stock prices change significantly.

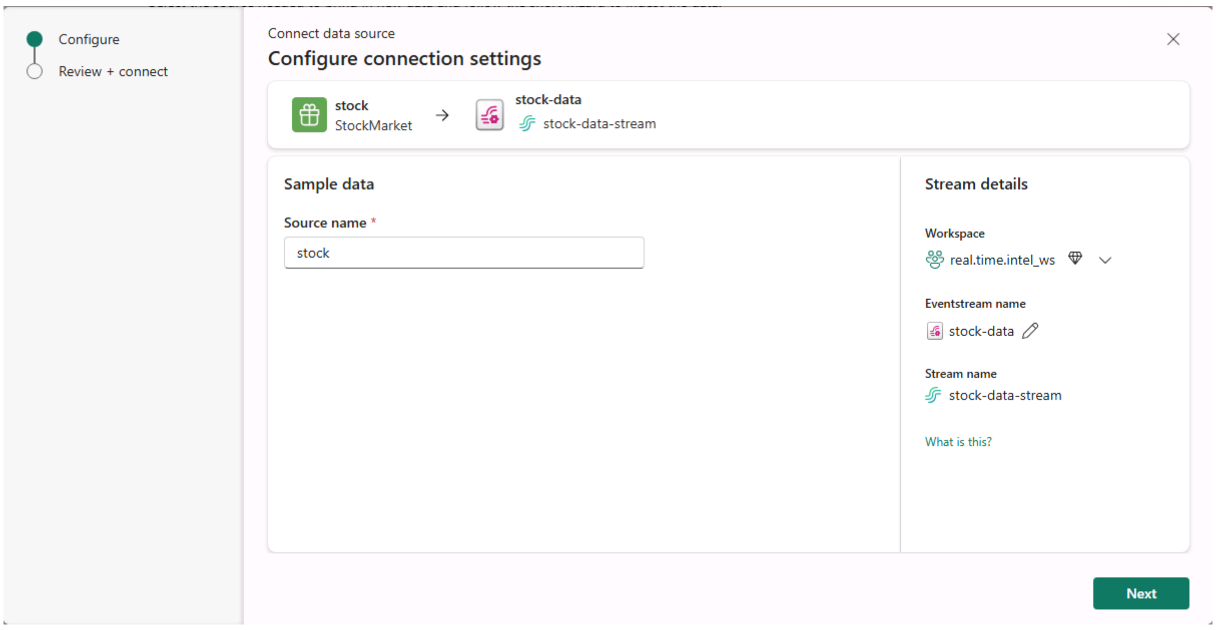
Experience

Create a Workspace

- Navigated to Microsoft Fabric Home and signed in.
- Selected Workspaces and created a new workspace with a name.

Create an Eventstream

- In the menu bar, selected the Real-Time hub.
- In the Real-Time hub, selected Data sources and connected to the Stock market sample data source.
- Named the source "stock" and changed the default eventstream name to "stock-data".
- Completed the connection wizard and opened the eventstream.



Connect data source

Configure connection settings

stock
StockMarket → stock-data
stock-data-stream

Sample data

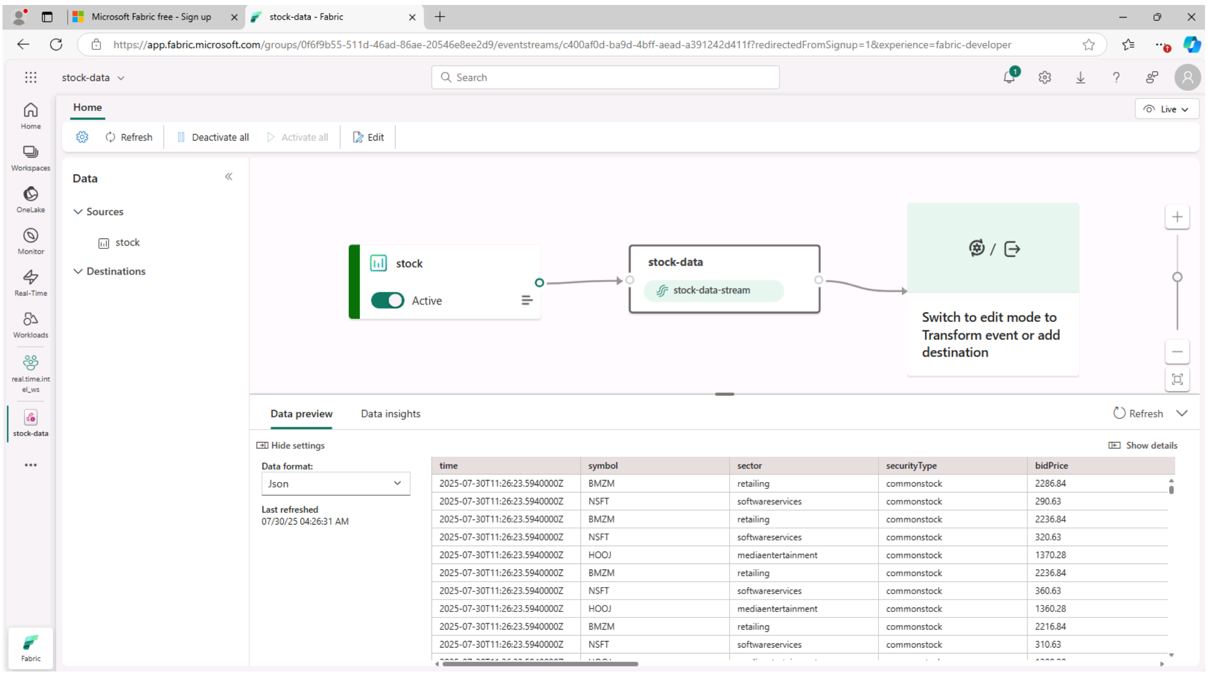
Source name *
stock

Stream details

Workspace
real.time.intel_ws

Eventstream name
stock-data

Stream name
stock-data-stream



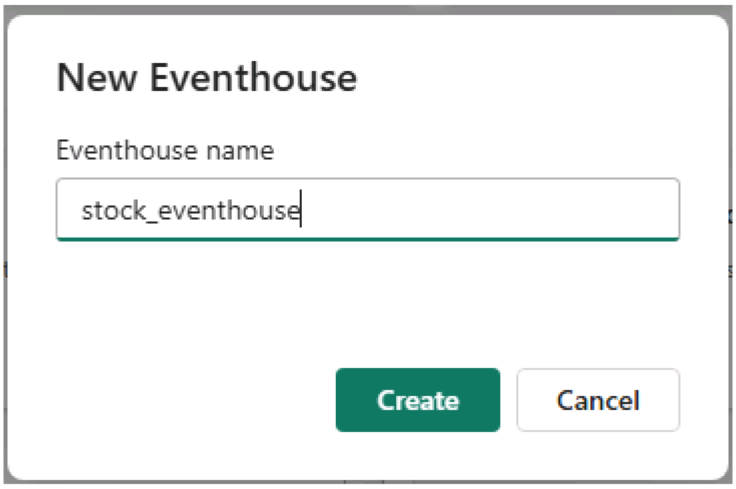
Data format: Json

Last refreshed
07/30/25 04:26:31 AM

time	symbol	sector	securityType	bidPrice
2025-07-30T11:26:23.5940000Z	BMZM	retailing	commonstock	2286.84
2025-07-30T11:26:23.5940000Z	NSFT	softwareservices	commonstock	290.63
2025-07-30T11:26:23.5940000Z	BMZM	retailing	commonstock	2236.84
2025-07-30T11:26:23.5940000Z	NSFT	softwareservices	commonstock	320.63
2025-07-30T11:26:23.5940000Z	HOOJ	mediaentertainment	commonstock	1370.28
2025-07-30T11:26:23.5940000Z	BMZM	retailing	commonstock	2236.84
2025-07-30T11:26:23.5940000Z	NSFT	softwareservices	commonstock	360.63
2025-07-30T11:26:23.5940000Z	HOOJ	mediaentertainment	commonstock	1360.28
2025-07-30T11:26:23.5940000Z	BMZM	retailing	commonstock	2216.84
2025-07-30T11:26:23.5940000Z	NSFT	softwareservices	commonstock	310.63

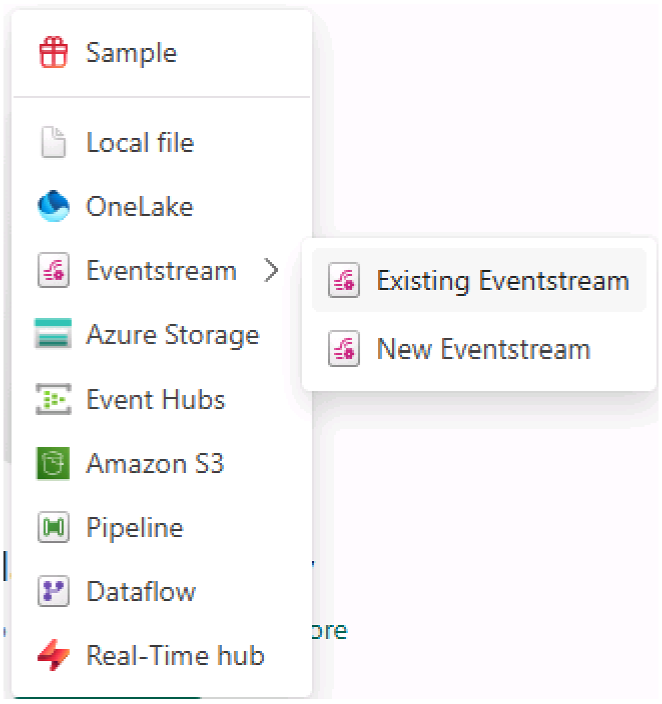
Create an Eventhouse

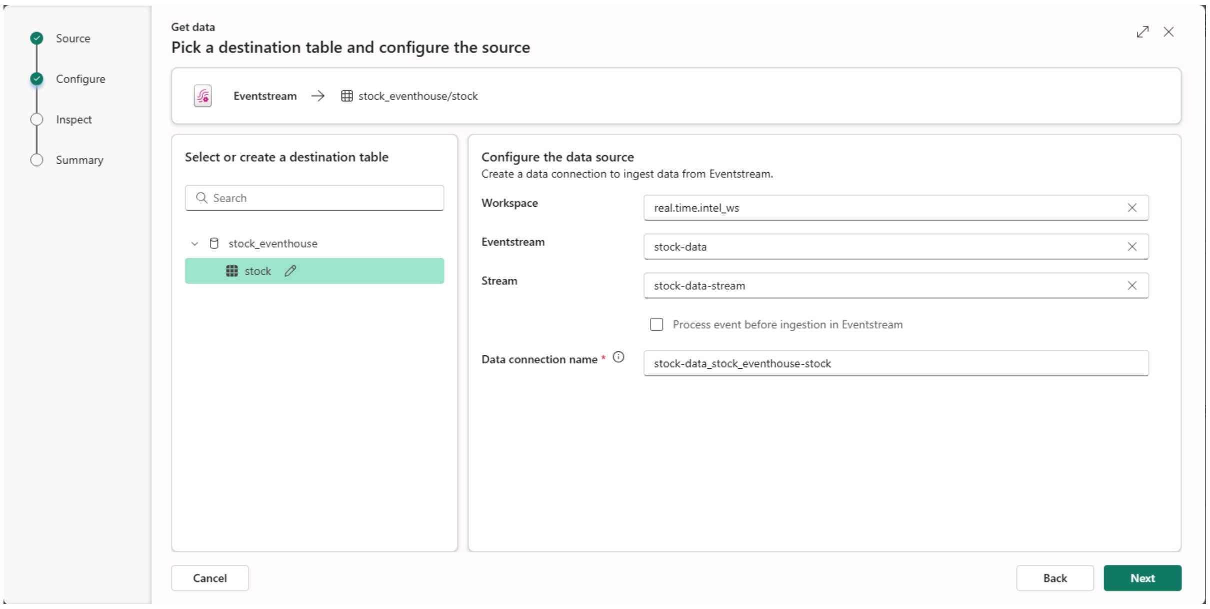
- Selected Create, then Eventhouse, and gave it a unique name.
- Selected Get data, chose Eventstream > Existing eventstream, and created a new table named "stock" from the stock-data eventstream.



New Eventhouse

Eventhouse name: stock_eventhouse





Get data

Pick a destination table and configure the source

Eventstream → stock_eventhouse/stock

Select or create a destination table

Search

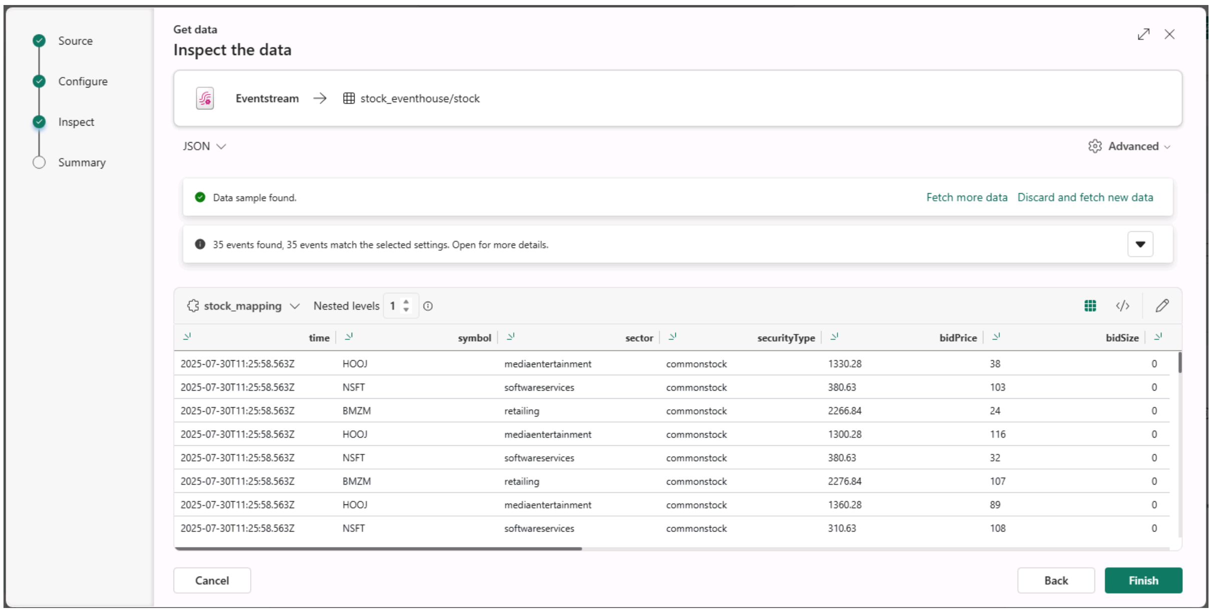
stock_eventhouse

stock

Configure the data source

Create a data connection to ingest data from Eventstream.

Workspace	real.time.intel_ws
Eventstream	stock-data
Stream	stock-data-stream
	☐ Process event before ingestion in Eventstream
Data connection name *	stock-data_stock_eventhouse-stock



Source
Configure
Inspect
Summary

Get data

Inspect the data

Eventstream → stock_eventhouse/stock

JSON

Advanced

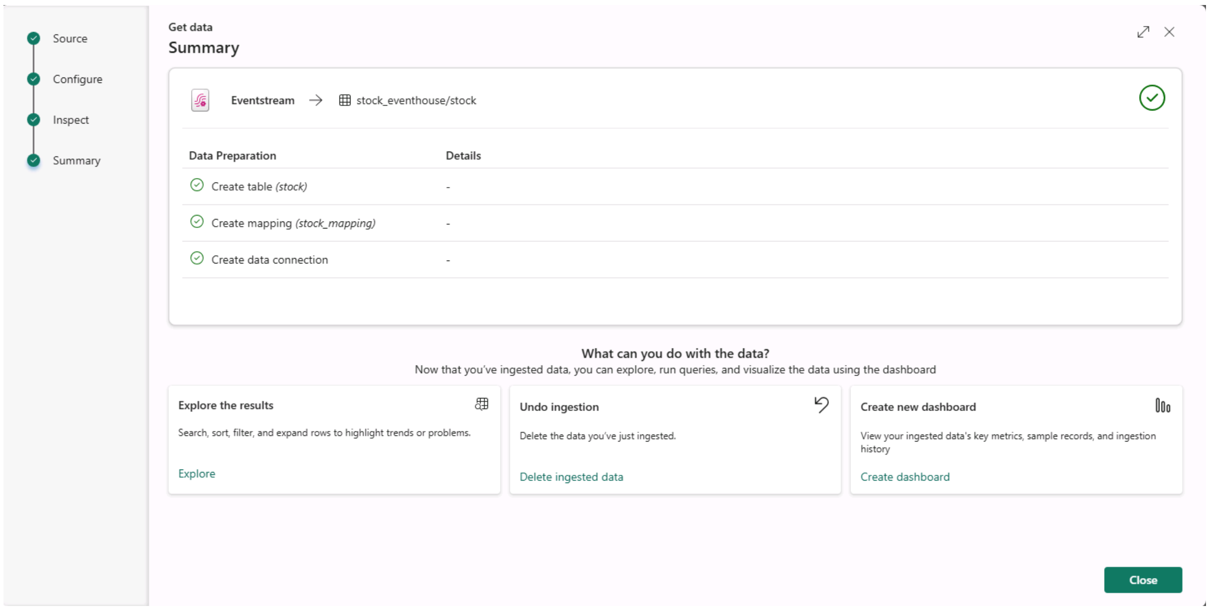
Data sample found.   Fetch more data   Discard and fetch new data

35 events found, 35 events match the selected settings. Open for more details.

stock_mapping   Nested levels 1

time	symbol	sector	securityType	bidPrice	bidSize	
2025-07-30T11:25:58.563Z	HOOJ	mediaentertainment	commonstock	1330.28	38	0
2025-07-30T11:25:58.563Z	NSFT	softwareservices	commonstock	380.63	103	0
2025-07-30T11:25:58.563Z	BMZM	retailing	commonstock	2266.84	24	0
2025-07-30T11:25:58.563Z	HOOJ	mediaentertainment	commonstock	1300.28	116	0
2025-07-30T11:25:58.563Z	NSFT	softwareservices	commonstock	380.63	32	0
2025-07-30T11:25:58.563Z	BMZM	retailing	commonstock	2276.84	107	0
2025-07-30T11:25:58.563Z	HOOJ	mediaentertainment	commonstock	1360.28	89	0
2025-07-30T11:25:58.563Z	NSFT	softwareservices	commonstock	310.63	108	0

Cancel   Back   Finish



Source
Configure
Inspect
Summary

Get data

Summary

Eventstream → stock_eventhouse/stock

Data Preparation	Details
Create table *(stock)*	-
Create mapping *(stock_mapping)*	-
Create data connection	-

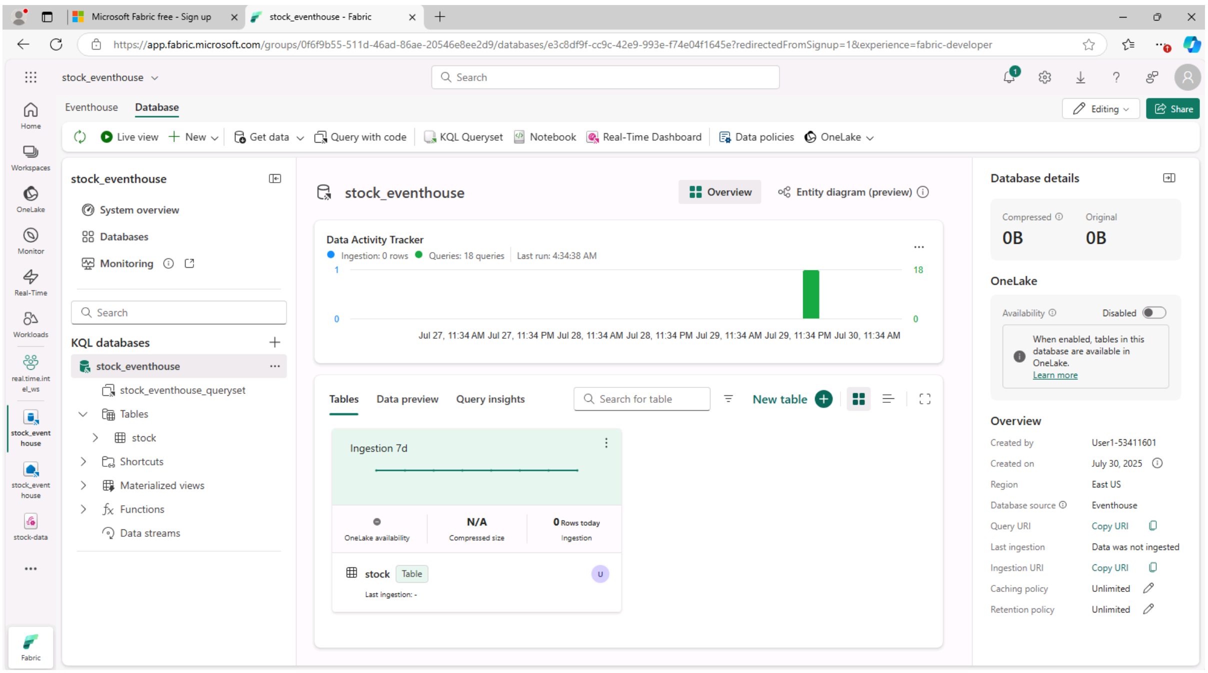
What can you do with the data?
Now that you've ingested data, you can explore, run queries, and visualize the data using the dashboard

Explore the results
Search, sort, filter, and expand rows to highlight trends or problems.
Explore

Undo ingestion
Delete the data you've just ingested.
Delete ingested data

Create new dashboard
View your ingested data's key metrics, sample records, and ingestion history
Create dashboard

Close



Database details

Compressed	0B
Original	0B

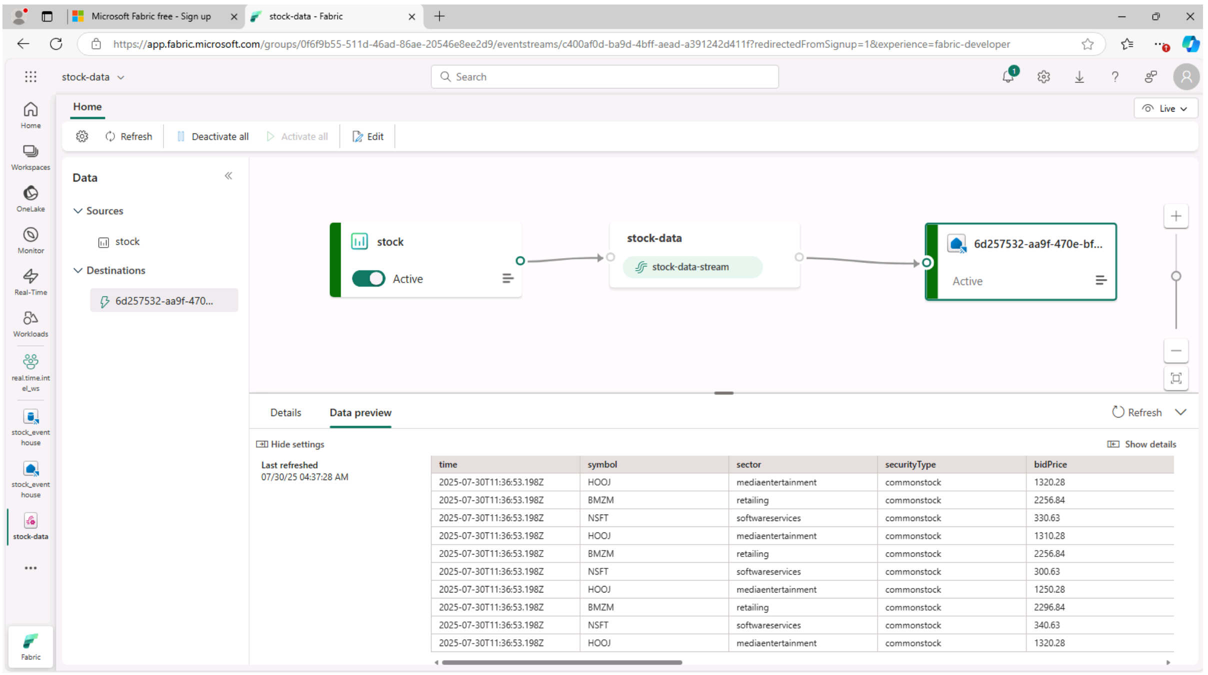
OneLake

Availability: Disabled

When enabled, tables in this database are available in OneLake. Learn more

Overview

Created by	User1-53411601
Created on	July 30, 2025
Region	East US
Database source	Eventhouse
Query URI	Copy URI
Last ingestion	Data was not ingested
Ingestion URI	Copy URI
Caching policy	Unlimited
Retention policy	Unlimited

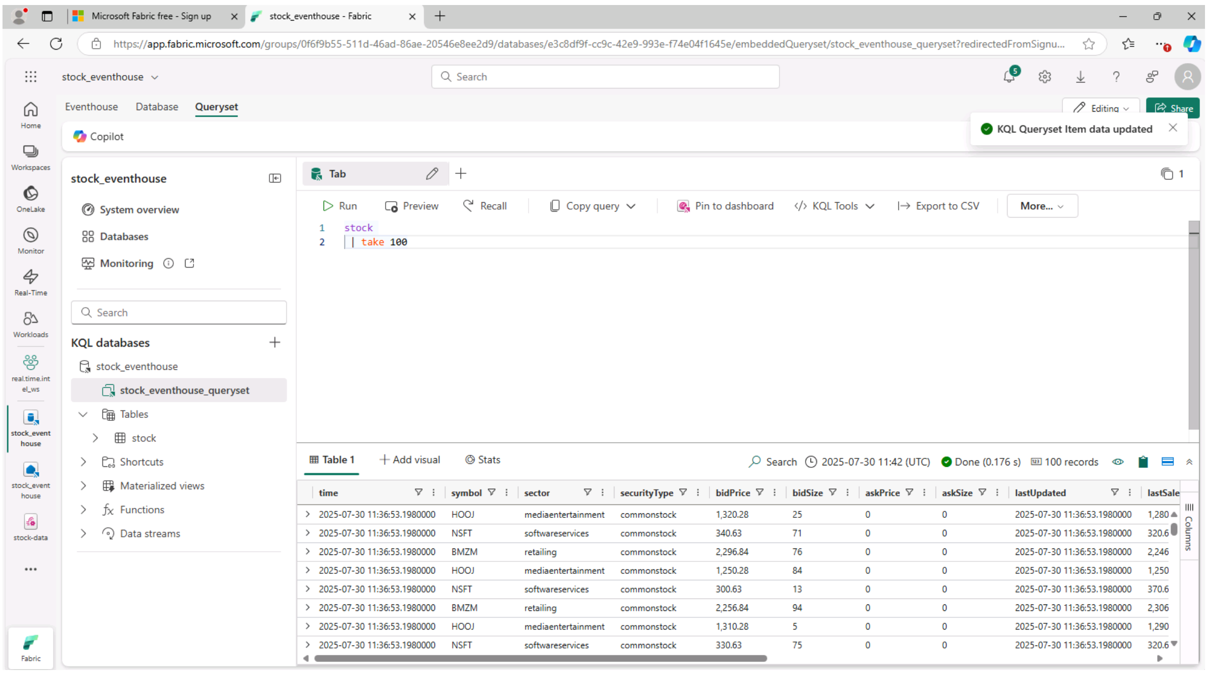


Last refreshed
07/30/25 04:37:28 AM

time	symbol	sector	securityType	bidPrice
2025-07-30T11:36:53.198Z	HOOJ	mediaentertainment	commonstock	1320.28
2025-07-30T11:36:53.198Z	BMZM	retailing	commonstock	2256.84
2025-07-30T11:36:53.198Z	NSFT	softwareservices	commonstock	330.63
2025-07-30T11:36:53.198Z	HOOJ	mediaentertainment	commonstock	1310.28
2025-07-30T11:36:53.198Z	BMZM	retailing	commonstock	2256.84
2025-07-30T11:36:53.198Z	NSFT	softwareservices	commonstock	300.63
2025-07-30T11:36:53.198Z	HOOJ	mediaentertainment	commonstock	1250.28
2025-07-30T11:36:53.198Z	BMZM	retailing	commonstock	2296.84
2025-07-30T11:36:53.198Z	NSFT	softwareservices	commonstock	340.63
2025-07-30T11:36:53.198Z	HOOJ	mediaentertainment	commonstock	1320.28

Query the Captured Data

- In the menu bar, selected the eventhouse database and opened the queryset.
- Ran the following KQL query to retrieve 100 rows of data:



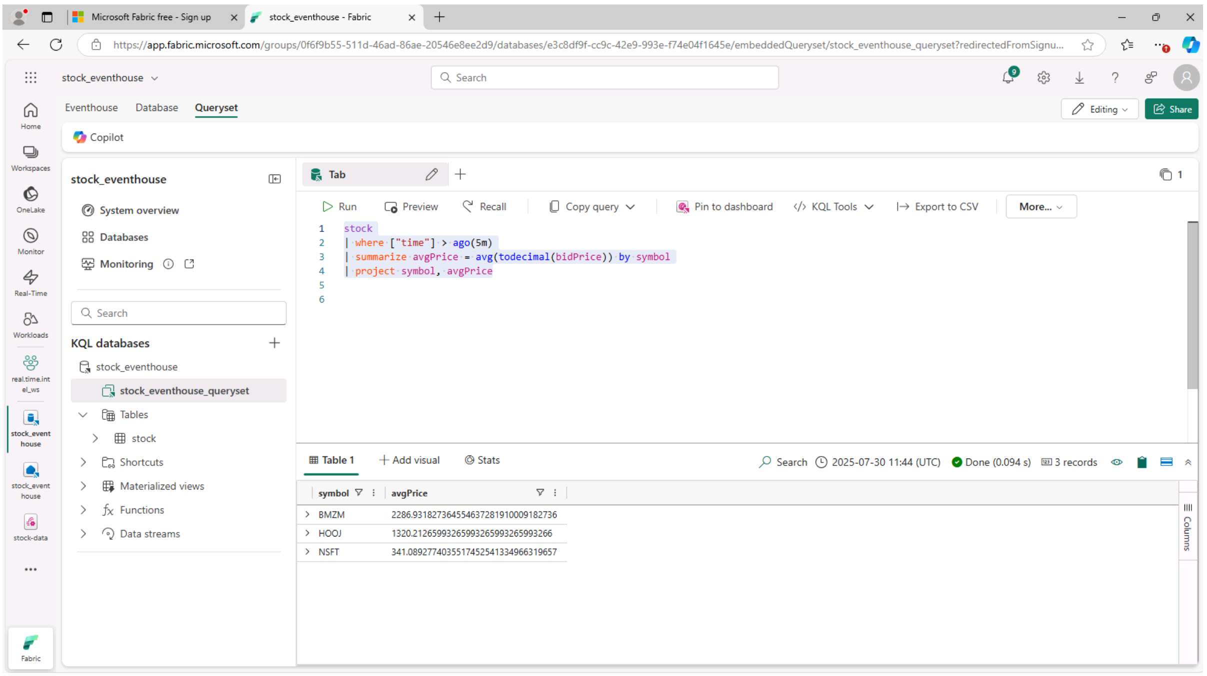
KQL Queryset Item data updated

```
stock
| take 100
```

2025-07-30 11:42 (UTC) Done (0.176 s) 100 records

time	symbol	sector	securityType	bidPrice	bidSize	askPrice	askSize	lastUpdated	lastSale
2025-07-30 11:36:53.1980000	HOOJ	mediaentertainment	commonstock	1,320.28	25	0	0	2025-07-30 11:36:53.1980000	1,280
2025-07-30 11:36:53.1980000	NSFT	softwareservices	commonstock	340.63	71	0	0	2025-07-30 11:36:53.1980000	320.6
2025-07-30 11:36:53.1980000	BMZM	retailing	commonstock	2,296.84	76	0	0	2025-07-30 11:36:53.1980000	2,246
2025-07-30 11:36:53.1980000	HOOJ	mediaentertainment	commonstock	1,250.28	84	0	0	2025-07-30 11:36:53.1980000	1,250
2025-07-30 11:36:53.1980000	NSFT	softwareservices	commonstock	300.63	13	0	0	2025-07-30 11:36:53.1980000	370.6
2025-07-30 11:36:53.1980000	BMZM	retailing	commonstock	2,256.84	94	0	0	2025-07-30 11:36:53.1980000	2,306
2025-07-30 11:36:53.1980000	HOOJ	mediaentertainment	commonstock	1,310.28	5	0	0	2025-07-30 11:36:53.1980000	1,290
2025-07-30 11:36:53.1980000	NSFT	softwareservices	commonstock	330.63	75	0	0	2025-07-30 11:36:53.1980000	320.6

- Modified the query to calculate the average price for each stock symbol in the last 5 minutes:

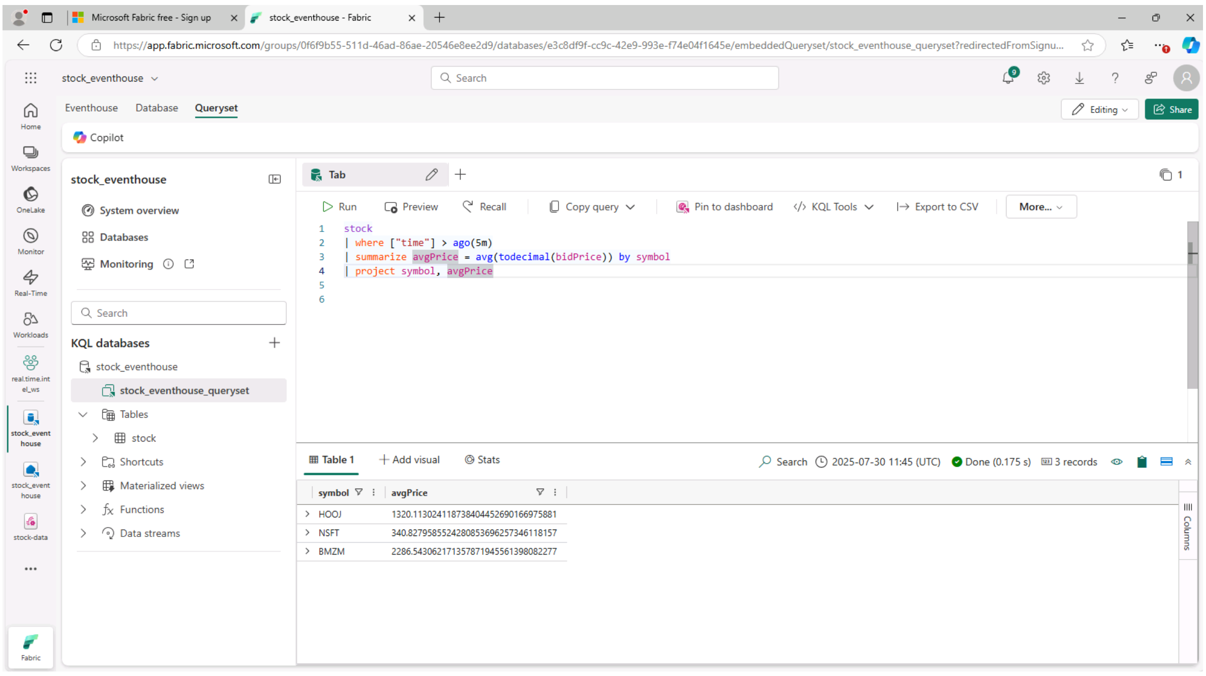


```
stock
| where ["time"] > ago(5m)
| summarize avgPrice = avg(todecimal(bidPrice)) by symbol
| project symbol, avgPrice
```

2025-07-30 11:44 (UTC) Done (0.094 s) 3 records

symbol	avgPrice
BMZM	2286.931827364554637281910009182736
HOOJ	1320.21265993265993265993265993266
NSFT	341.0892774035517452541334966319657

- Ran the modified query and observe the results.

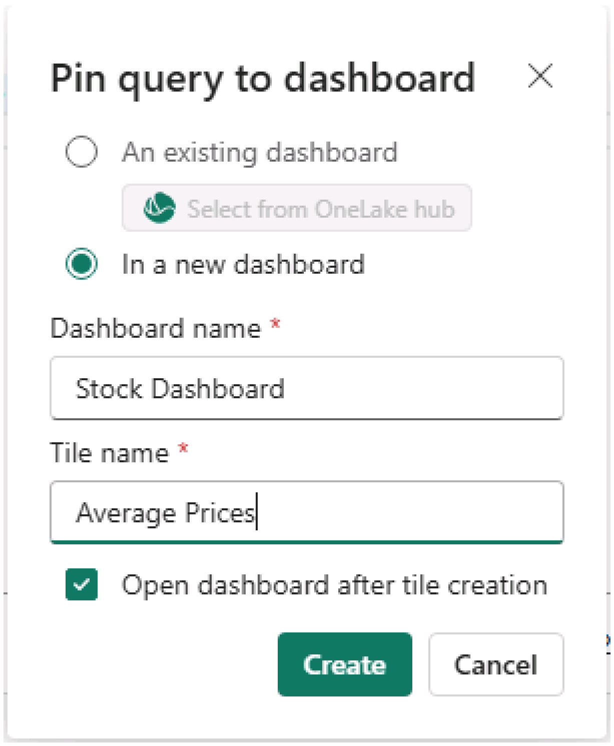


```
stock
| where ["time"] > ago(5m)
| summarize avgPrice = avg(todecimal(bidPrice)) by symbol
| project symbol, avgPrice
```

symbol	avgPrice
HOOJ	1320.113024118738404452690166975881
NSFT	340.8279585524280853696257346118157
BMZM	2286.543062171357871945561398082277

Create a Real-Time Dashboard

- Pinned the KQL query for average stock prices to a new dashboard named "Stock Dashboard" with the tile name "Average Prices".
- Switched to Editing mode and changed the visual from Table to Column chart.
- Applied changes to view the modified dashboard.



Pin query to dashboard

An existing dashboard

Select from OneLake hub

[✓ In a new dashboard]

Dashboard name *

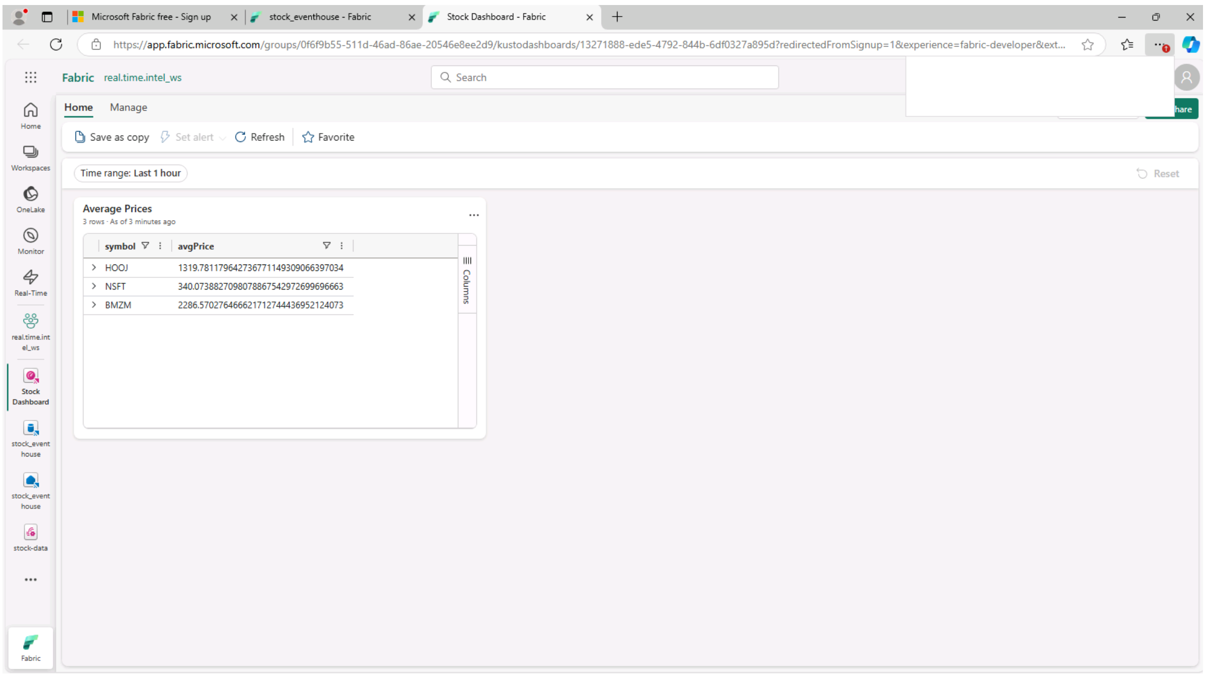
Stock Dashboard

Tile name *

Average Prices

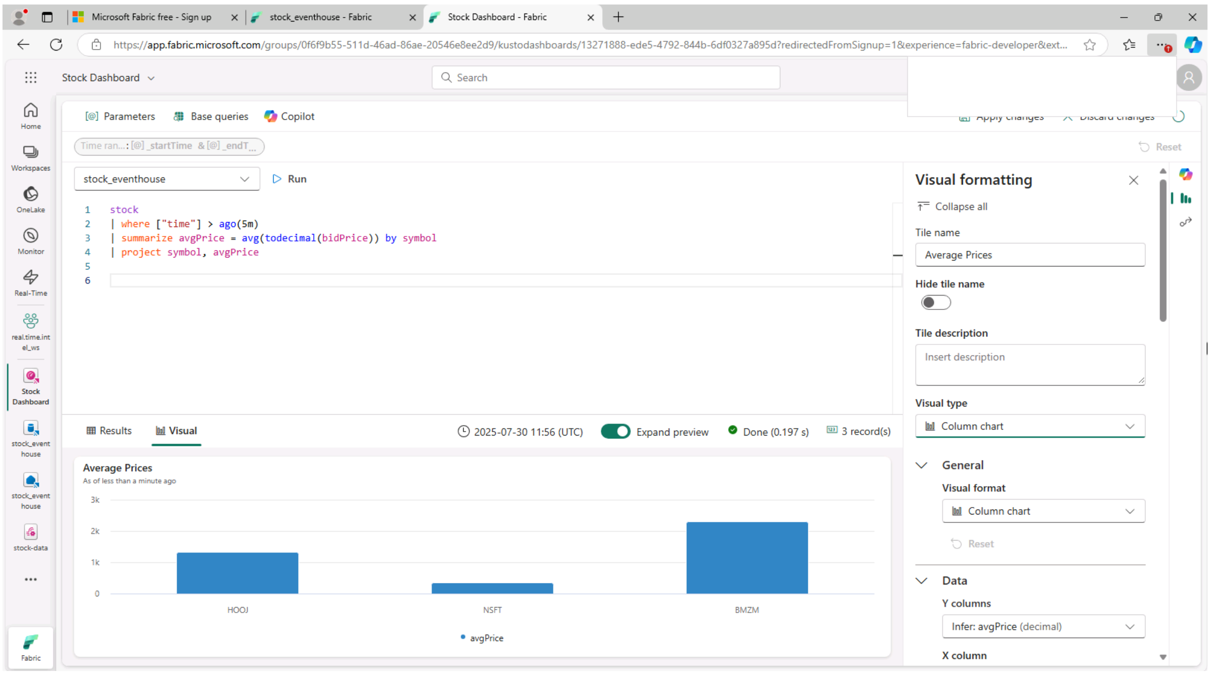
[✓ Open dashboard after tile creation]

Create Cancel


Average Prices

3 rows · As of 3 minutes ago

symbol	avgPrice
HOOJ	1319.781179642736771149309066397034
NSFT	340.0738827098078867542972699696663
BMZM	2286.570276466621712744436952124073


```
stock
| where ["time"] > ago(5m)
| summarize avgPrice = avg(todecimal(bidPrice)) by symbol
| project symbol, avgPrice
```

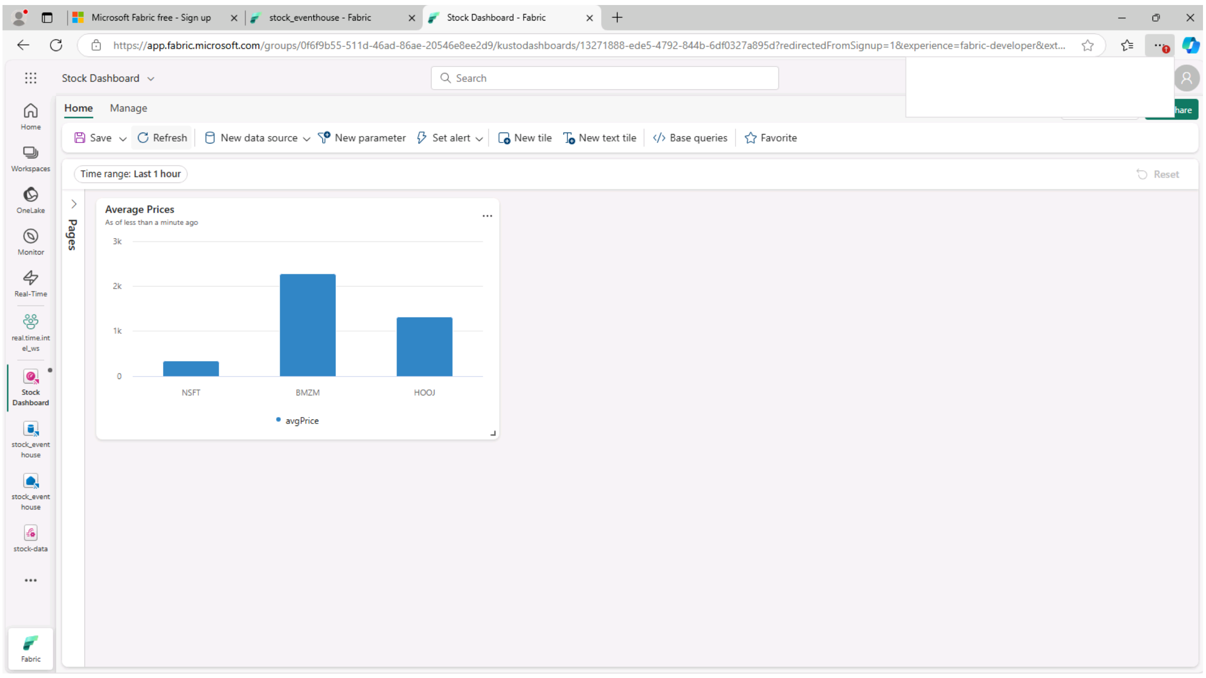
Visual formatting

Tile name: Average Prices

Visual type: Column chart

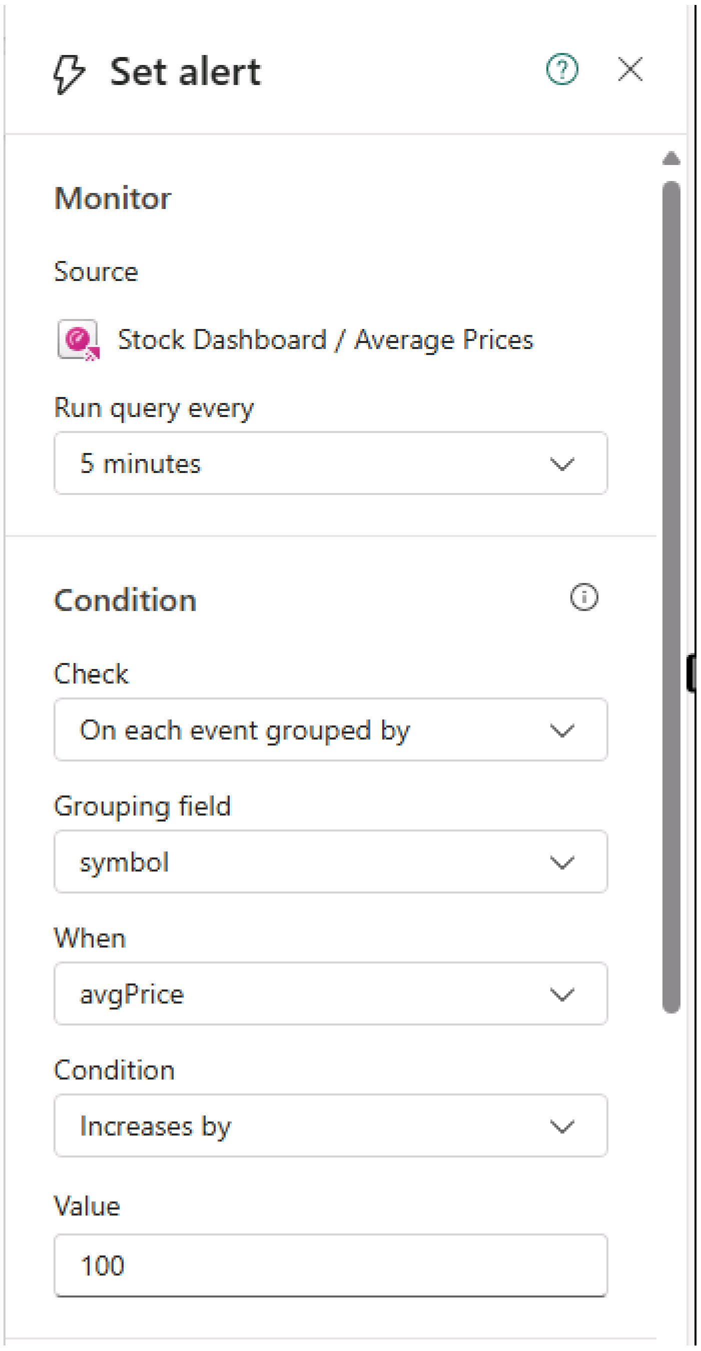
Visual format: Column chart

Y columns: Infer: avgPrice (decimal)



Create an Alert

- In the dashboard, selected Set alert.
- Configured the alert with the following settings:
  - ➢ Run query every: 5 minutes
  - ➢ Check: On each event grouped by
  - ➢ Grouping field: symbol
  - ➢ When: avgPrice
  - ➢ Condition: Increases by
  - ➢ Value: 100
  - ➢ Action: Send me an email
  - ➢ Save location: Your workspace
  - ➢ Item: Create a new item with a unique name
- Created the alert and confirm it has been saved.



Set alert

Monitor

Source

Stock Dashboard / Average Prices

Run query every: 5 minutes

Condition

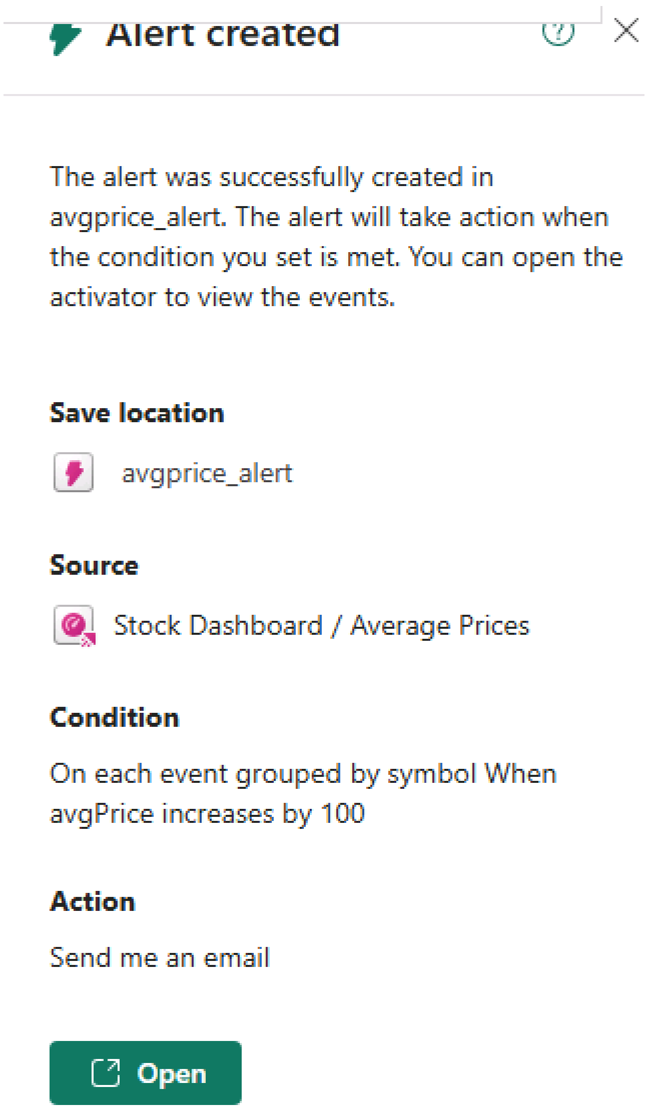
Check: On each event grouped by

Grouping field: symbol

When: avgPrice

Condition: Increases by

Value: 100


Alert created

The alert was successfully created in avgprice_alert. The alert will take action when the condition you set is met. You can open the activator to view the events.

Save location

avgprice_alert

Source

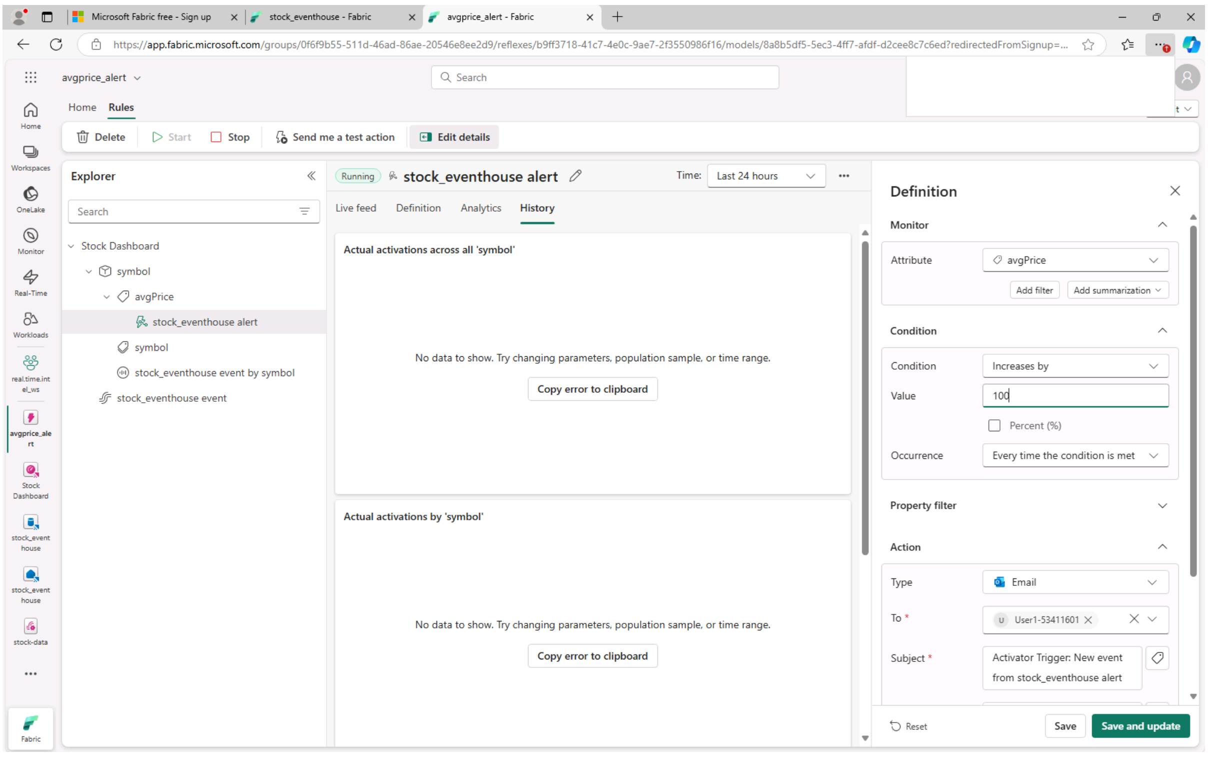
Stock Dashboard / Average Prices

Condition

On each event grouped by symbol When avgPrice increases by 100

Action

Send me an email



Time: Last 24 hours

Actual activations across all 'symbol'

No data to show. Try changing parameters, population sample, or time range.

Actual activations by 'symbol'

No data to show. Try changing parameters, population sample, or time range.

Definition

Monitor

Attribute: avgPrice

Condition

Condition: Increases by

Value: 100

Percent (%): ☐

Occurrence: Every time the condition is met

Property filter

Action

Type: Email

To *: User1-53411601

Subject *: Activator Trigger: New event from stock_eventhouse alert

Results

- ✓ A workspace and eventstream were successfully created in Microsoft Fabric.
- ✓ Real-time stock market data was ingested and stored in an eventhouse.
- ✓ KQL queries were executed to analyze the ingested data, including retrieving average stock prices.
- ✓ A real-time dashboard was created to visualize stock data, and an alert was configured to notify when stock prices change significantly.

Conclusion

This project provided a practical introduction to Real-Time Intelligence in Microsoft Fabric. Key features such as eventstreams, eventhouses, KQL querying, real-time dashboards, and alerts were effectively utilized to create an analytical solution for real-time data streams. This exercise demonstrated the capabilities of Microsoft Fabric in managing and visualizing real-time data efficiently.

Resources

GitHub profile: https://github.com/ThatoMTNG/Microsoft-Fabric-Analytics-Engineer-DP-600-

Mentions

Project Author: Thato Metsing (https://www.linkedin.com/in/thatometsing/)

Project Mentor: Maureen Direro (https://www.linkedin.com/in/maureen-direro-46a6b220/)