Microsoft | Microsoft Fabric

Fabric
Real-Time Analytics in a Day

AGENDA

Presentation Part 1

Fabric Introduction
Real Time Analytics Overview
KQL Introduction

Lab 1

KQL Database Creation, Data Ingestion and Exploration

Presentation Part 2

KQL Advanced, Data Science

Lab 2

Advanced KQL, Policies, Visualization

Q&A

1

Presentation Part 1

Microsoft Fabric

Data analytics for the era of AI

Complete Analytics Platform

- Everything, unified
- SaaS-ified
- Secured and governed

Lake Centric and Open

- OneLake
- One copy
- Open at every tier

Empower Every Business User

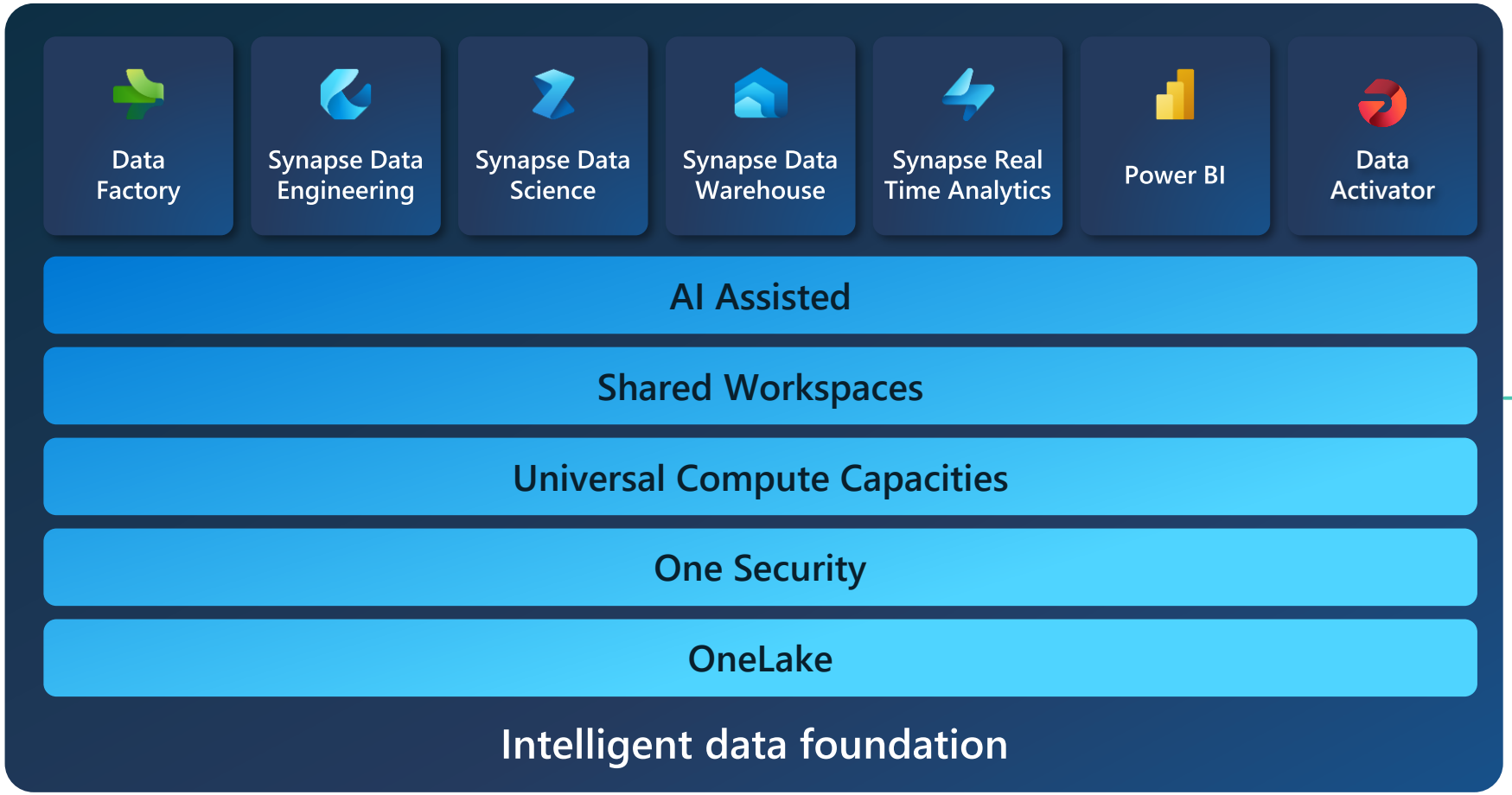
- Familiar and intuitive
- Built into Microsoft 365
- Insight to action

AI Powered

- Copilot accelerated
- GPT on your data
- AI-driven insights

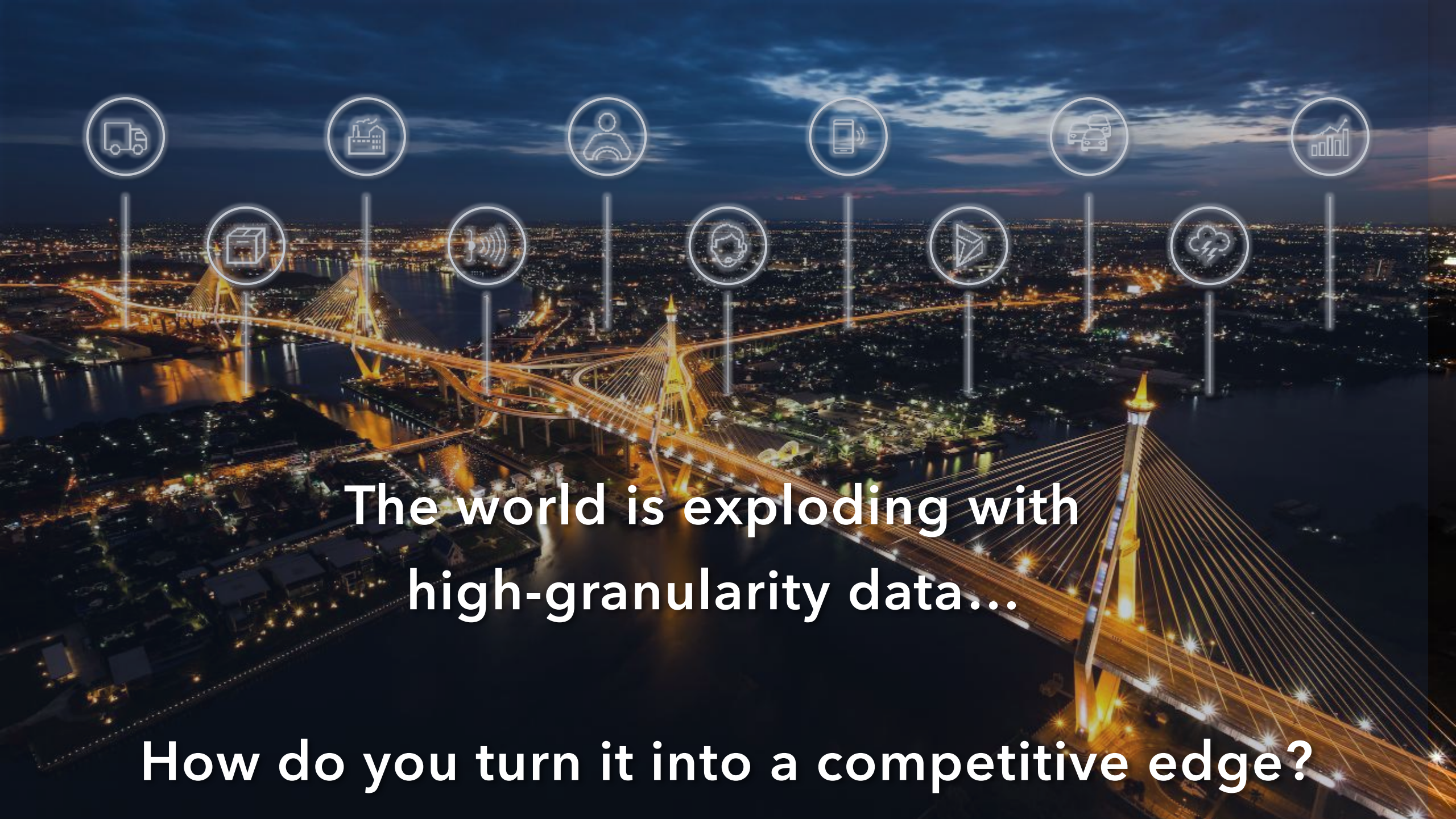
Microsoft Fabric

An end-to-end analytics platform that brings together all the data and analytics tools that organizations need to go from the data lake to the business user



Single...

Onboarding and trials
Sign-on
Navigation model
UX model
Workspace organization
Collaboration experience
Data Lake
Storage format
Data copy for all engines
Security model
CI/CD
Monitoring hub
Data Hub
Governance & compliance


The world is exploding with
high-granularity data...
How do you turn it into a competitive edge?

Observational Analytics

Why is it challenging to analyze?

Frequently changing business questions

Constantly changing schema

Near real time visibility required

Analytics systems costs are often prohibitive

Looking for unpredictable phenomena

Real-time analytics in Fabric

- Simplicity
- SaaS
- One unified portfolio for all data and AI solutions
- All Experiences are fully integrated
- One Logical Copy

- By-default streaming ingestion
- By-default indexing everything
- Time and Hash-based data partitioning
- Data movement
- Torrents of data
- Take action on real-time data
- Native support for structured, semi-structured and free text data
- In-place transformation

Poll: Do you work with use cases involving analytics on telemetry data?

 Yes

 Not yet

Every Vertical

Highlighted industries scenarios

Transportation and logistics

Smart cities/ buildings/home

Manufacturing operation

Automotive industry

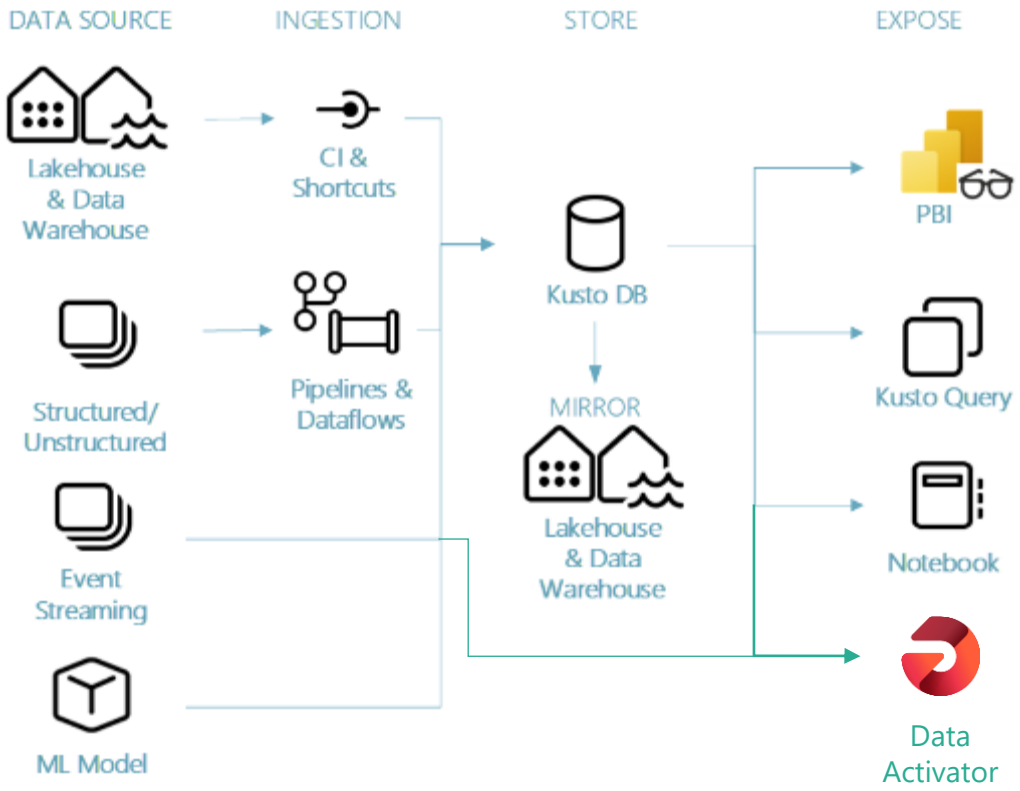
Energy oil and gas

IoT Analytics

Log Analytics

Real-time analytics in Fabric

Common Architecture



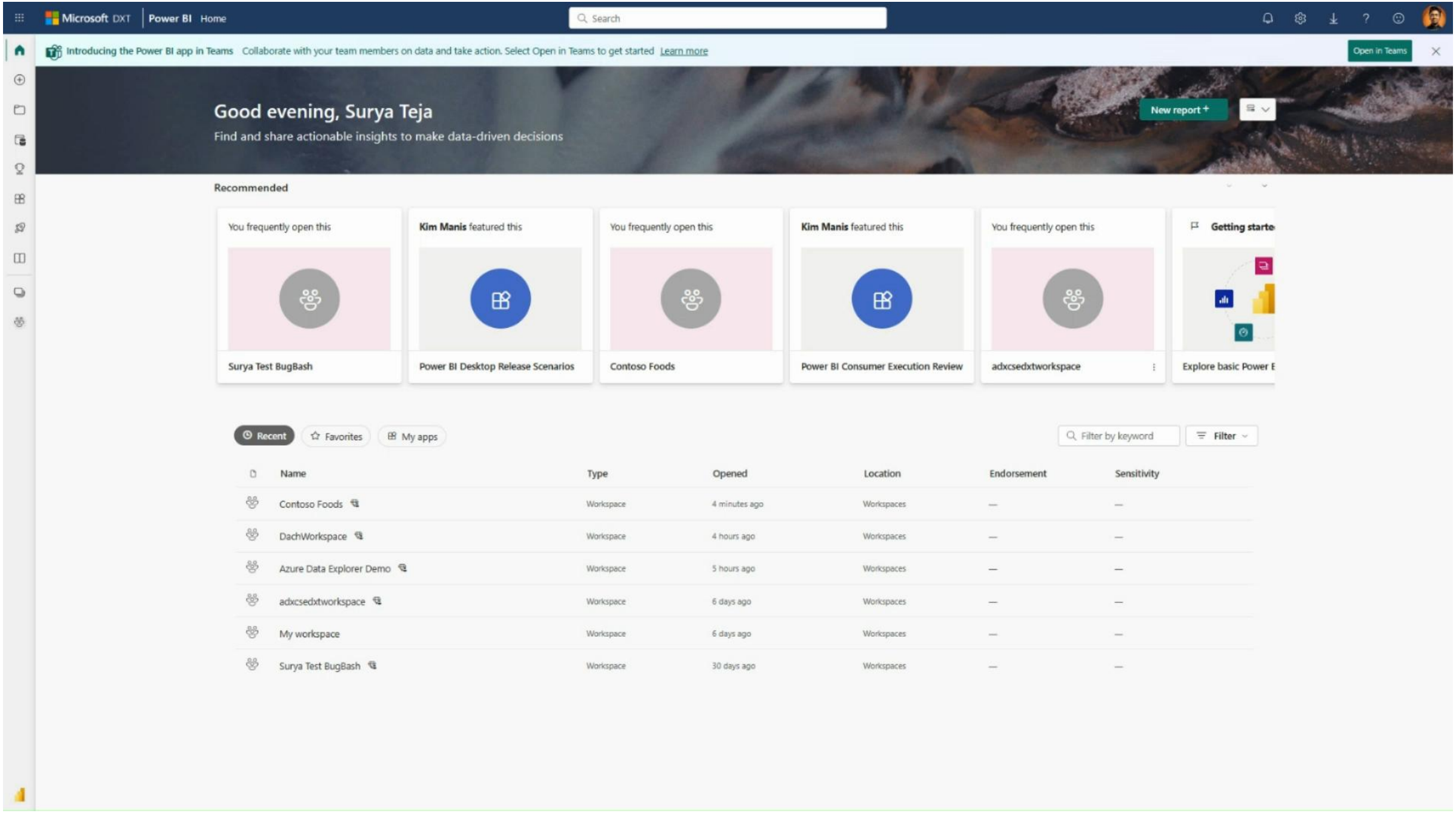
Ingestion: Event streaming through Event Hubs/IoT Hub/OneLake/Pipelines & Dataflows/Notebooks/Open-source ecosystem (Kafka, Logstash, Open Telemetry, etc.)

Storage: Kusto DB, and Lakehouse via mirroring

Train and test: ML Models and Experiments

Exposure: Power BI through see-through mode, KQL, Notebook via Spark/KQL, Data Activator detects & alerts.

RTA Demo E2E - Fundamental



Fabric Real-Time Analytics (RTA)

Find insights, track progress, and make decisions faster.

- KQL Database
- KQL Queryset
- Real-Time Dashboards
- Eventstreams
- Power BI
- Data Activator

KQL database key capabilities

Unlimited Scale (query, ingestion, storage)	Any data source	Any data format
Structured Semi-Structured Free-text	Real-time data Transformation of complicated data structure	Streaming analytics in NRT
High performance **Low latency** **High freshness**	Timeseries database	Everything is indexed and partitioned

KQL Database
recently released

Graph query semantics
Reduced the time to create a new eventstream down to a few seconds.
Introducing Graph Semantics, a new way to contextualize time series data

In-place data sharing
In place, real time, data sharing cross tenant
Create a database shortcut in Real-Time Analytics

Delta table shortcuts
Easily enable querying every delta table in OneLake or Azure Storage

Inline python in KQL database
Easily enable querying every delta table in OneLake or Azure Storage

Vector search for AI embeddings
Leverage the distributed nature of Kusto for high performant embedding selection
Optimizing Vector Similarity Searches at Scale (microsoft.com)

Sample databases
Create new sample databases or consume sample data

KQL Queryset

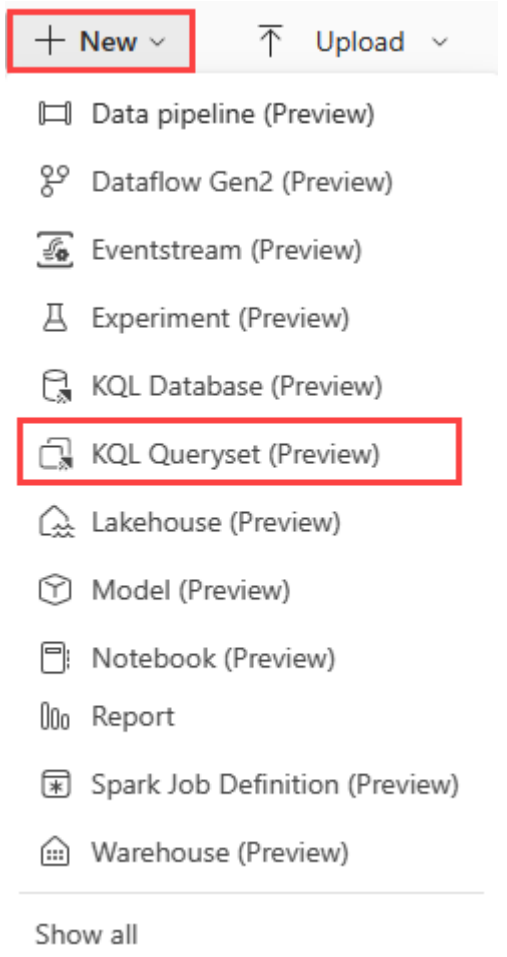
Overview

Uses the Kusto Query Language (KQL) for **creating queries**, views, functions, control commands, customize results and also supports many SQL functions.

Tabs in the KQL queryset can be associated with a different KQL database.

Lets your **save queries** for later use or **share with others** to collaborate on data exploration.

You can also change the **KQL database** associated with any tab, allowing you to run the same query on data in different states.



Event Streams

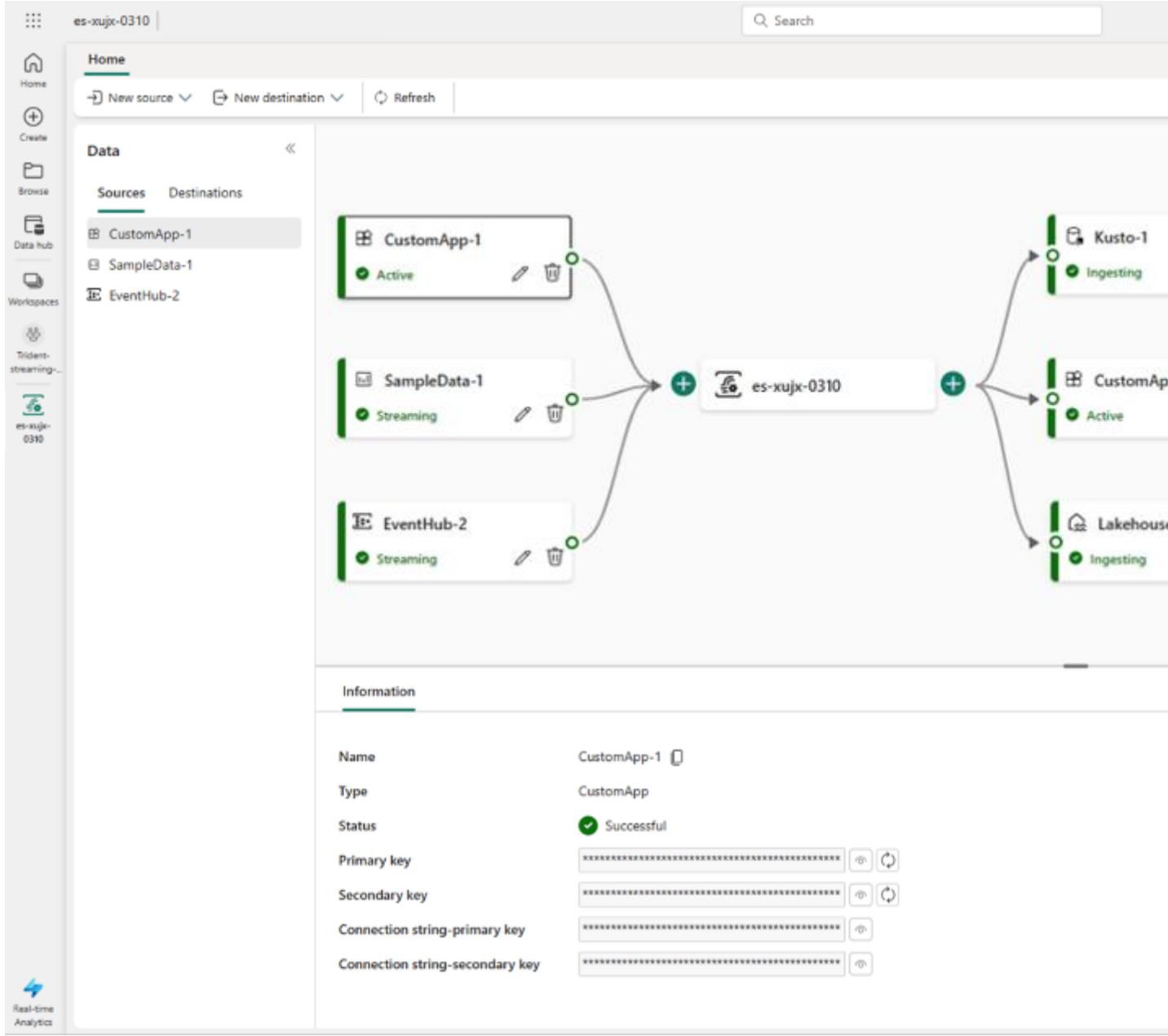
Overview

Centralized place for event data
Capturing, transforming, and routing event data.

Various source connectors
Fetch event data from diverse source, such as Custom Apps to push into eventstreams, Azure Event Hub or Samples.

No-code experience
Drag and drop makes it intuitive and easy to use. Build an end-to-end data flow diagram.

Multiple destinations
Lakehouse, KQL Database, Custom App, Data Activator (Reflex)



Event Streams
recently released

IoT Hub Source
Consume events from IoT Hub as a source of events for your eventstream

Stream processing before sending to KQL Database
Users can now do stream processing/transformation on incoming events from event stream before sending the events to the KQL Database that allows you to route, validate, filter, transform or aggregate the events.

AMQP format connection string support
Eventstream now offers support for AMQP format connection strings in both Custom App source and destination, allowing for seamless integration with external message brokers like RabbitMQ with Shovel.

Improved no-code stream processing designer
Improved experiences for editing event streams adding more intuitive gestures in the no-code designer.

Create Eventstream faster
Reduced the time to create a new eventstream down to a few seconds.

Real-Time Dashboards

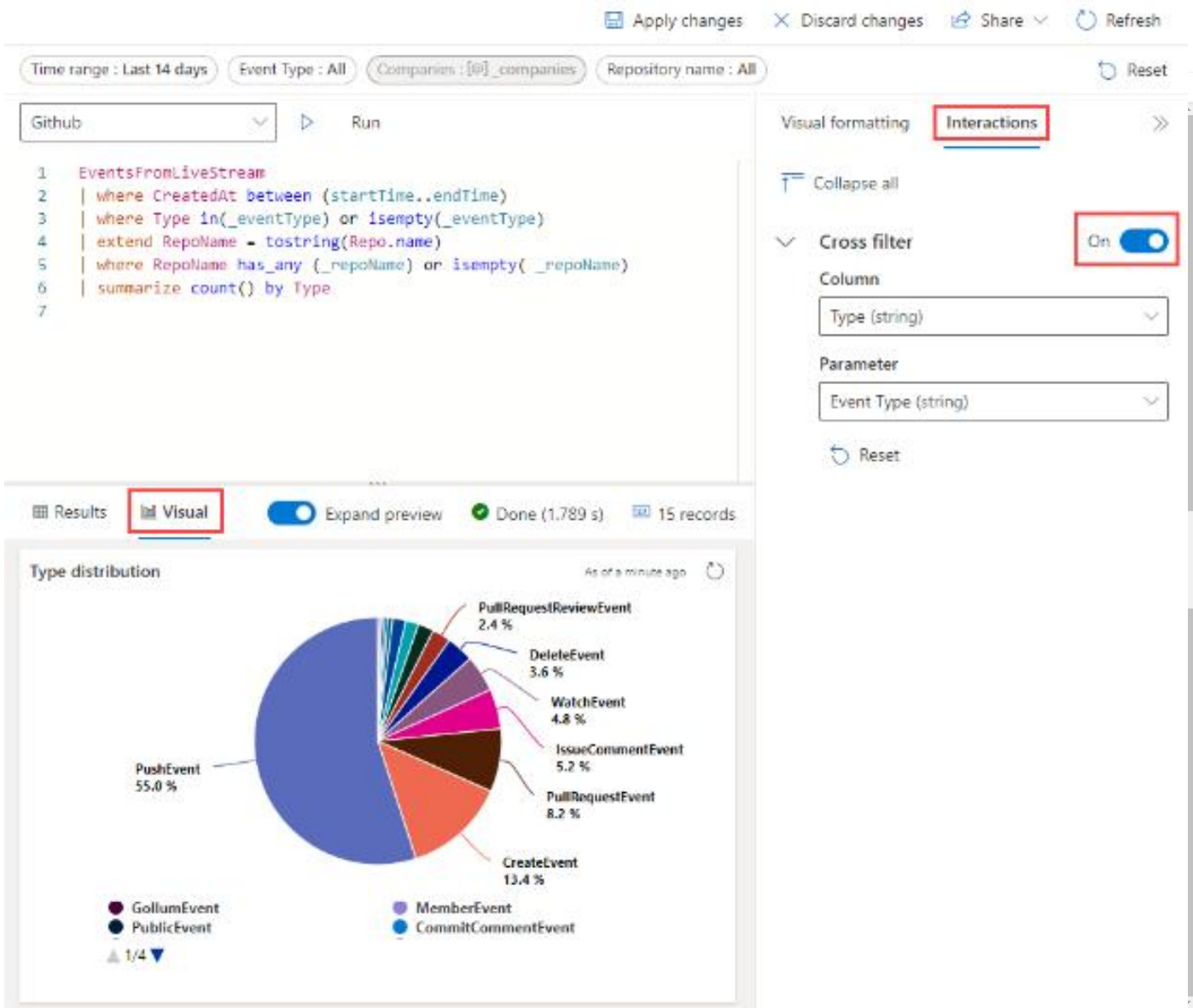
Overview

Collection of tiles
Each tile has an underlying query with a visual representation. Can be organized in pages. Supports parameters & cross-filter (drill-through).

Real Time
All live-queries with a min of 30s auto-refresh

Native KQL
Export to dashboard and modify seamless by toggling from analyst to pro-dev or vise-versa.

Visuals
Ease of data exploration. Fully integrated experience, improved query and visual performance with light, flexible modeling on high-granularity low latency data.



Power BI

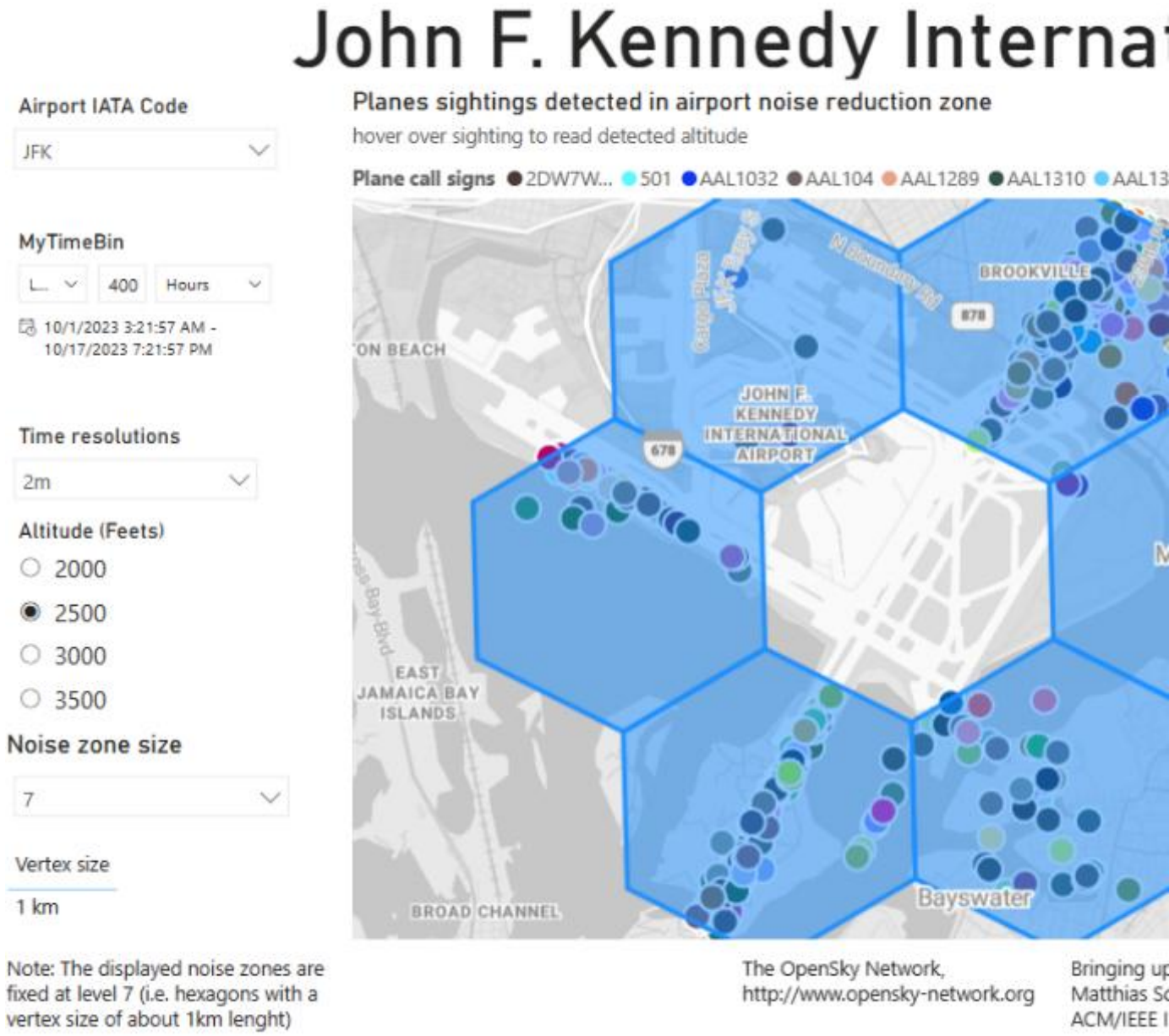
Best Practices

Travel light
Bring only the data that you need.

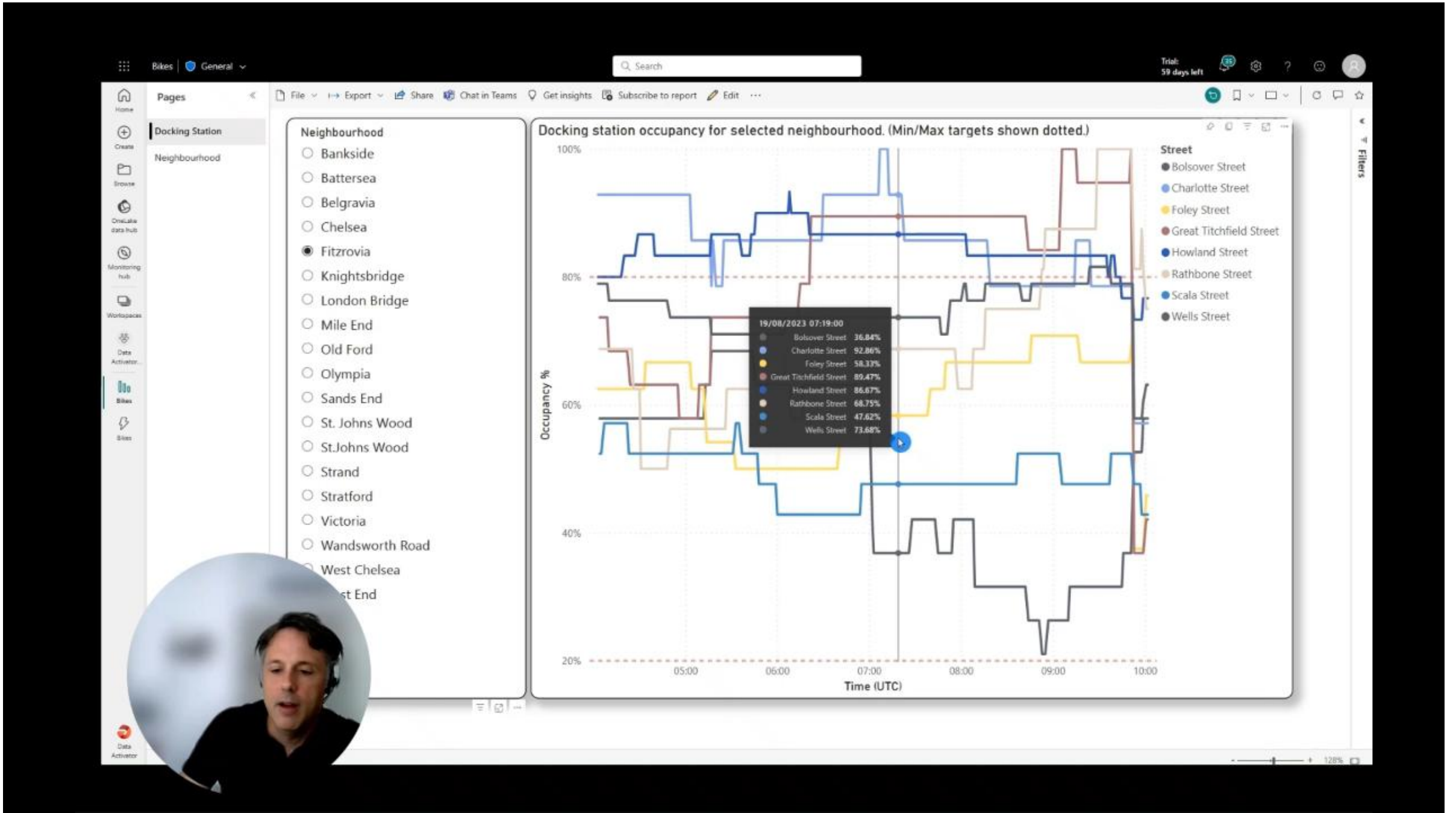
Composite model
Combine aggregated data for top-level dashboards with filtered operational raw data.

Parallelism
Increase concurrent connections in DQ or use weak consistency to improve the performance of dashboard rendering.

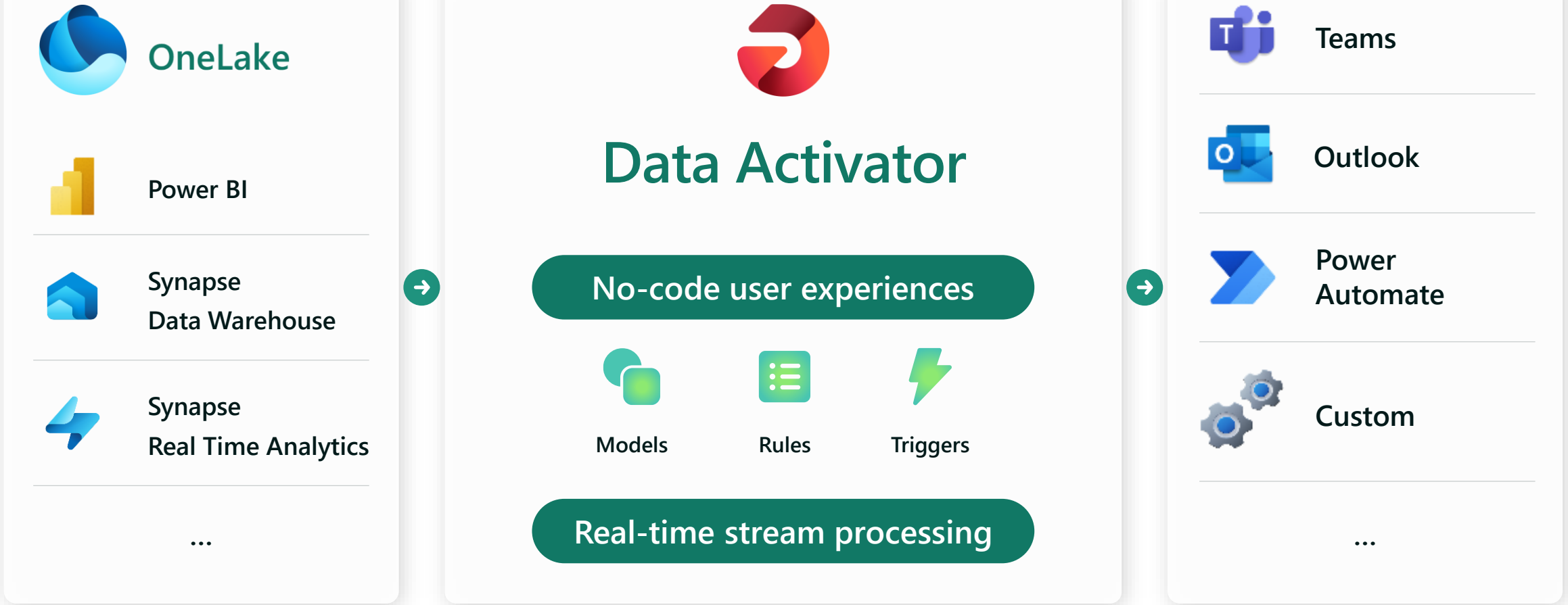
Additionally
Use effective slicers, filters and visuals for your data



RTA Demo – Data Activator & Power BI



Introducing Data Activator



Demo Scenario: London Bicycle Hire

- Use real-time data from docking station API
- Monitor 800 docking stations across London
- Alert if docking station has too few bikes, or too many

Resources

Explore the product here: https://aka.ms/try-fabric-rta

Get your questions answered in the Microsoft Fabric webinar series: https://aka.ms/fabric-webinar-series

See the latest announcements in the Microsoft Fabric blog site: https://aka.ms/fabric-tech-blog

Reading the official documentation on Real-Time Analytics: https://aka.ms/fabric-docs-rta

What is a Kusto query?

A Kusto query is a read-only request to process data and return results.

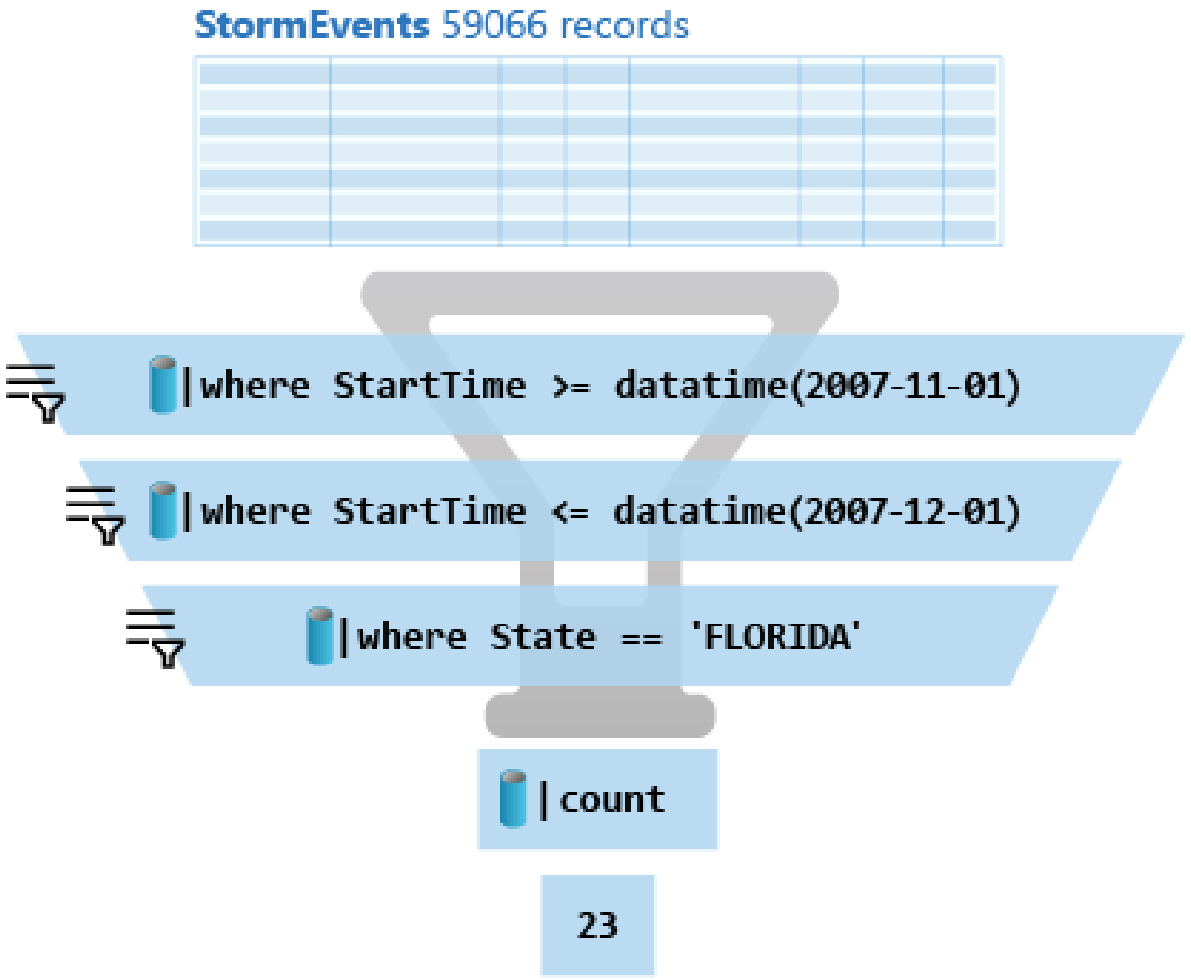
Has one or more query statements and returns data in a tabular or graph format.

Statements are sequenced by a pipe (|).
Data flows, or is piped, from one operator to the next.

It's like a funnel, where you start out with an entire data table.

The data is filtered/manipulated at each step and then fed into the following step.

Each time the data passes through another operator, it's filtered, rearranged, or summarized.



KQL Concepts

Relational operators
filters, union, joins, aggregations, …
Each operator consumes **tabular input** and produces a **tabular output**

Commands
Can be combined with '|' (**pipe**)

Similarities
OS shell, Linq, functional SQL…

Agility
Queries are easy to **write, read, change**

Statements
Single statement query
Use 'let' for reusing statements
Multi-statement (';') queries

KQL Basic Operators for data exploration

… | **count**
- Counts records in input table (e.g. T)

… | **take** 10
- Get few records - to familiarize yourself with the data. No actual order ensured.

… | **where** Timestamp > ago(1) and UserId == 'abdcdef'
- Filtering on a specific fields

… | **project** Col1, Col2, …
- Choose some columns (great if input table has dozens of columns)

…| **extend** NewCol1=Col1+Col2
- Introduces new calculated columns

… | **render** timechart
- Render data into a visual plot while exploring

SQL to KQL

Try the ‘EXPLAIN’ operator as follows:

EXPLAIN SELECT COUNT_BIG(*) as C FROM LogisticsTelemetry

Use SQL to KQL Cheat Sheet

Growth mindset 😇

Schema

Schema is
Relational, Lightweight, Dynamic

Databases
Authorization boundary
Transaction boundary
But not query boundary!
Supporting cross-database and cross-cluster queries

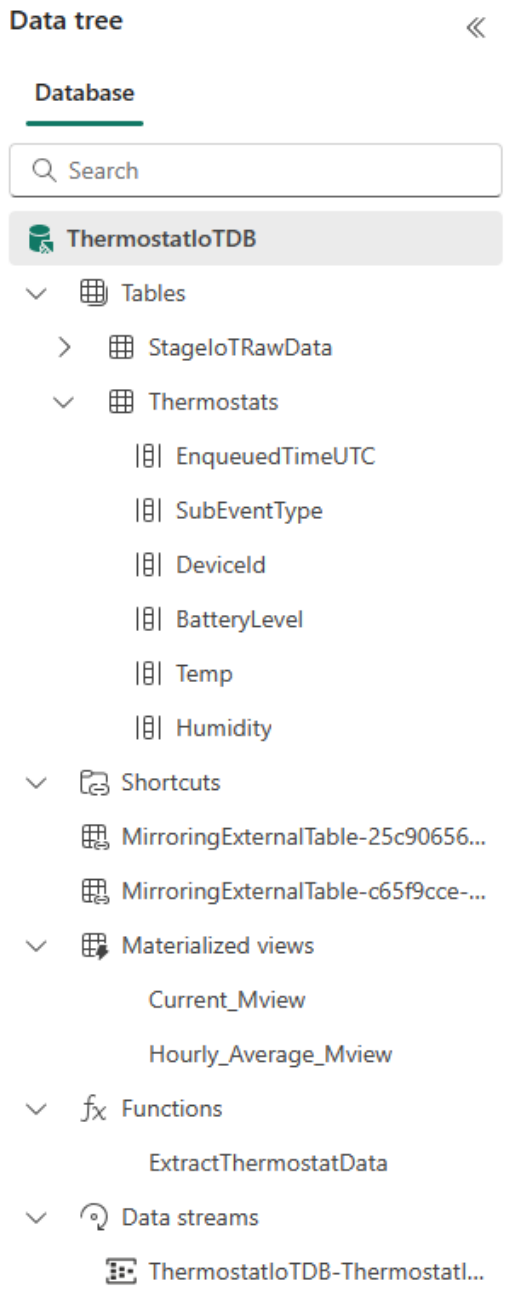
Tables
Rectangular

Columns
Supported types: boolean, integer, real, decimal, dates, timespan, string, dynamic (JSON)

Shortcuts (external data)

Stored functions (views)

Materialized views



Stored Functions
(stored procedures/views)

```
//Function creations
.create function AzureGithubEvents()
{
    GithubEvent
    | where Repo has 'Azure'
}
```

Functions
Reusable script, defined and used in a Database scope

View
parameter-less function

Schema
Scalar or tabular

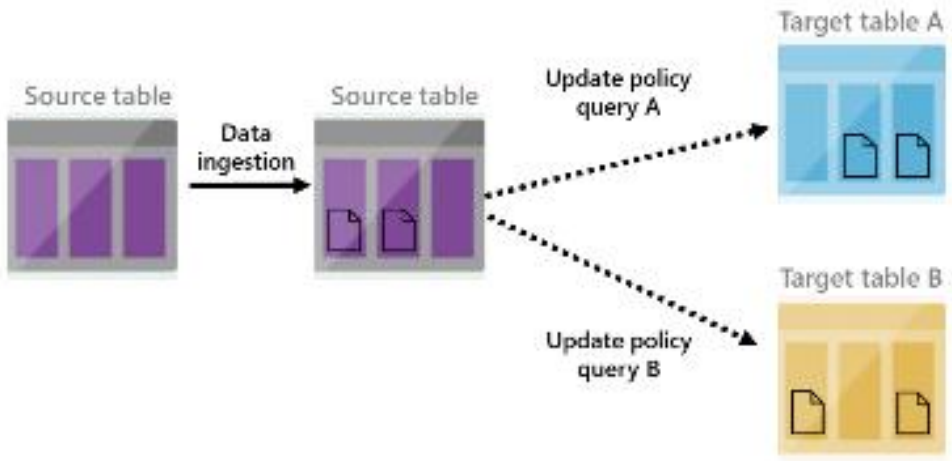
Special powers
Can override table with the same name.

Safe and Secure
Control commands are forbidden.

Applications
- Sharing queries between users/applications
- Abstracting complex logic from other applications
- SQL-compat tools connecting to Kusto via Views to run high-performance queries

Update Policy (mini-ETL/Trigger)

```
.alter table TargetTable policy update
'['
   '{"IsEnabled": true,'
   '"Source":"SourceTable",'
   '"Query": "DeriveFunction()"}'
']'
```



Essentially

Triggered ingestion into another table.

Semantics

Attached on "*target*" table and points to "*source*" table.
Transformation is an arbitrary Kusto query

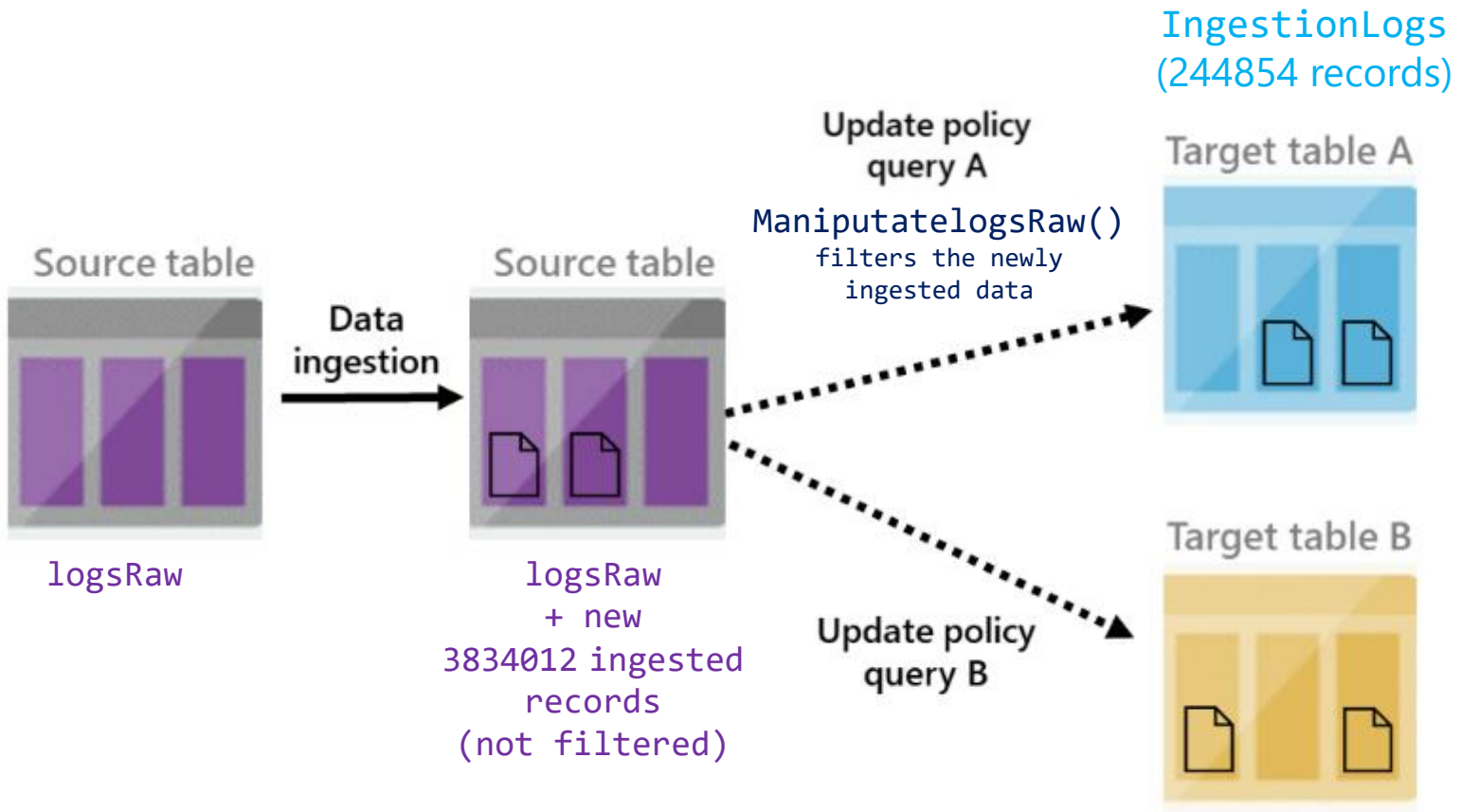
Special powers

Source table is scoped to the newly ingested data only

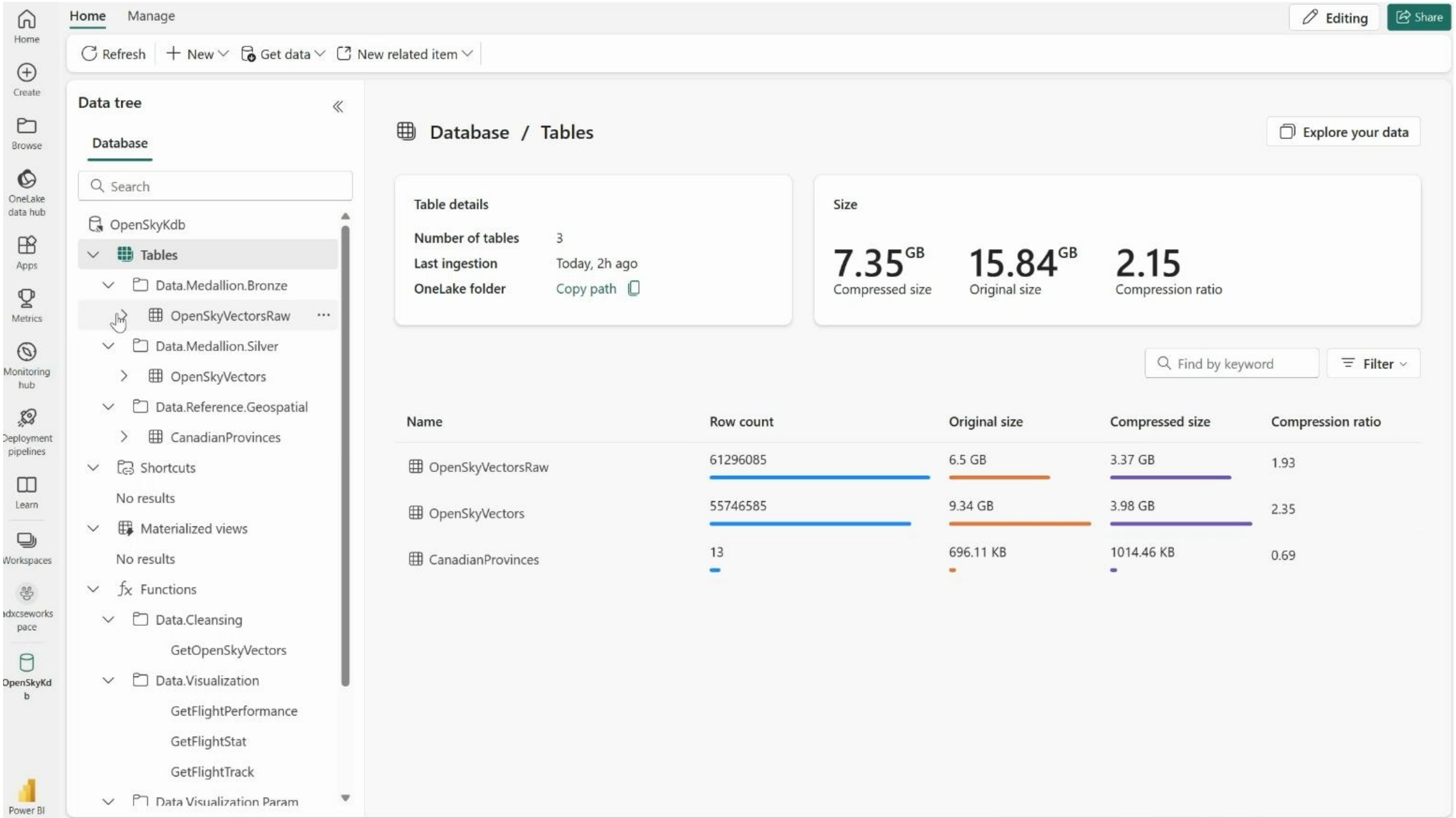
Applications

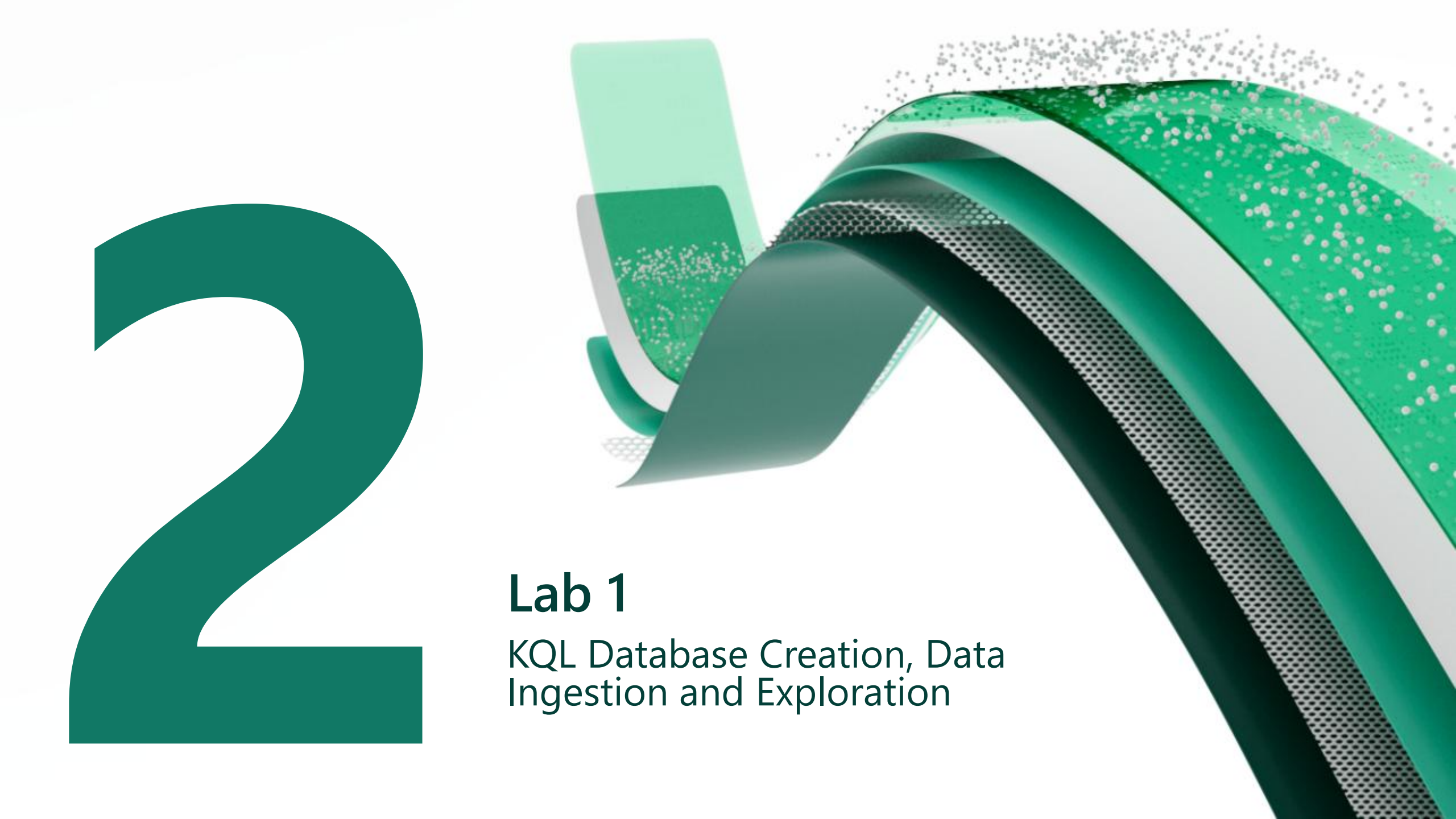
- Transform data schema (lightweight ETL)
- De-multiplexing data stream into several tables
- Use target tables with longer retention
- Reduce duplicates

Update Policy



RTA Demo E2E - Intermediate





2

Lab 1

KQL Database Creation, Data Ingestion and Exploration

Lab Guidelines

Combination of labs and open-hack. Use the links, hints, **try to solve**, or raise your hands.

Screenshots are just examples. Your output may not exactly match the screenshots provided.

Please **read the instructions** carefully.

Badges will be issued within 24 hours.

We will help everyone complete the Labs.

https://aka.ms/FabricRTAinaDay > start with **Lab1.md**

aka.ms/FabricRTAinADay

Lab 1

Lab 2

Complete the quizzes to earn the badge.

Quizzes can be re-attempted, and will help guide your answers

Lab 1 - Quiz

Lab 2 - Quiz

3

Presentation Part 2

Let statements

```
//let statement
let AzureGithubEvents =
    GithubEvent
    | where Repo has 'Azure';
//Expression here
AzureGithubEvents
| count
```

Essentially
Named expression or function, defined and used in a query scope

Usage
let foo = T | where … ;
let bar = (val:string) { … };

Schema
Scalar or tabular

Special powers
Can override table / stored function with the same name

Applications
- Reusing language expressions
- Mid-tier exposing Logical Model (see also: restrict)

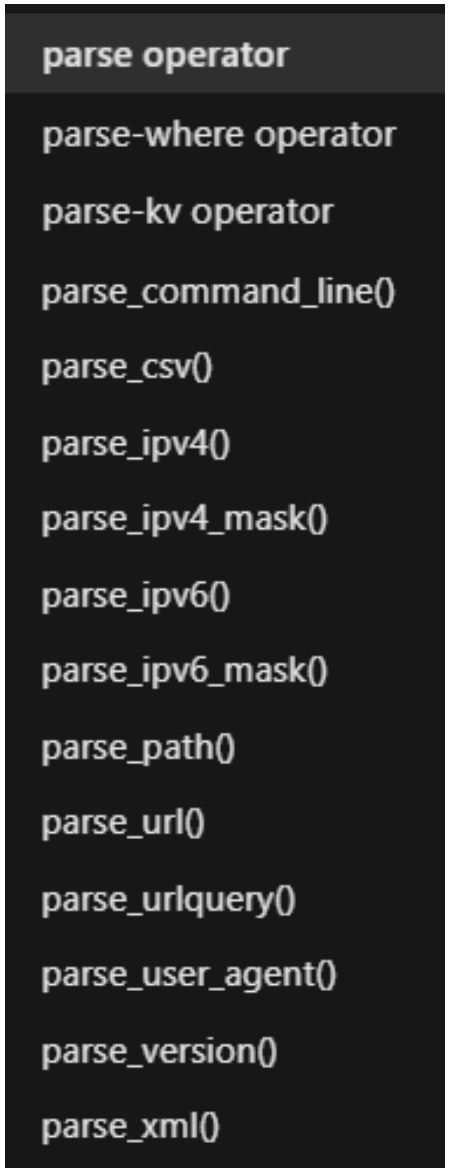
Parsing

Parsing Can be done with regular expressions as well as with native functions for specific data types and patterns

Parsing operators
Return data in columns

Parsing functions
Return dynamic objects using JSON notation

Applications
- Parse complex JSON, XML documents
- Extract IP addresses, URLs
- Extract key-value pairs



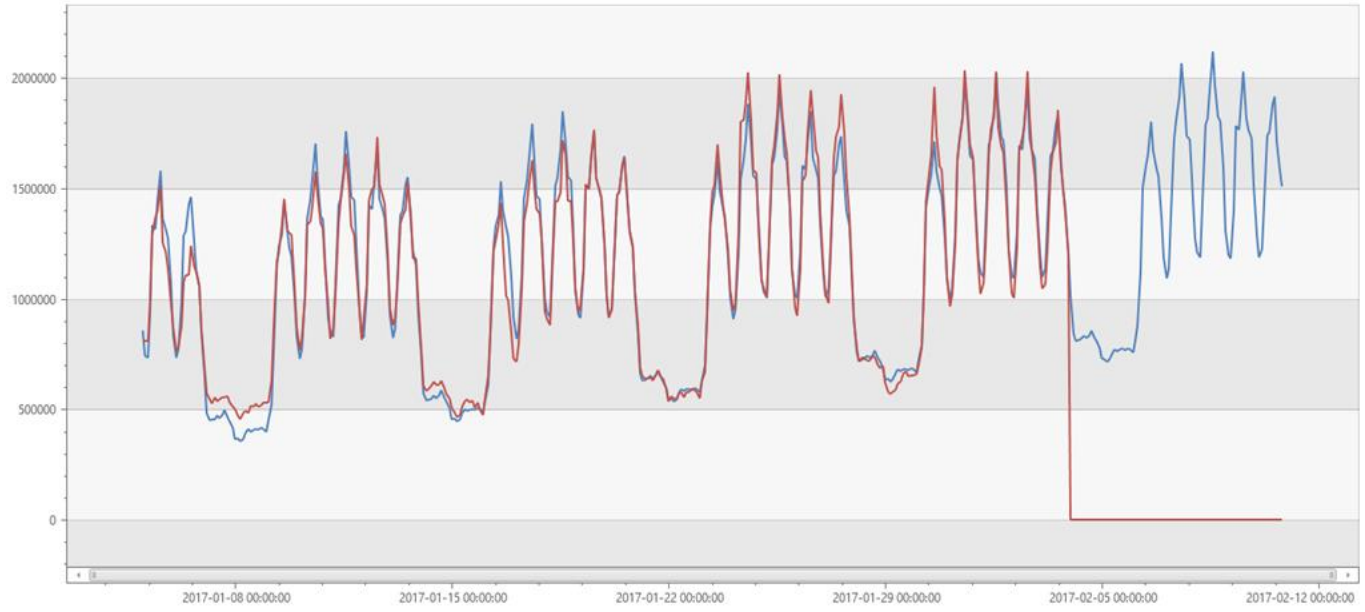
Time Series Analysis

make-series
Operator for creating set of (time) series

Large family of functions can be applied
- Element-wise operations
- Filtering (signal processing)
- Statistics
- Regressions
- Seasonality detection/validation
- Anomaly Detection
- Forecasting
- mv-apply operator

Optimized for bulks of time series

avg_shock_series	enqueuedTime
[-0.014102389298512921, 0.054822081643445778, -0.024335543043808073, 0.0073609799437805252, -0.034215766864223458, -0.031322885028220365, 0.007474701289064049,	["2022-01-25T16:53:16.9056584Z", "2022-01-25T17:03:16.9056584Z", "2022-01-25T17:13:16.9056584Z", "2022-01-25T17:23:16.9056584Z", "2022-01-25T17:33:16.9056584Z", "2022-01-25T17:43:16.9056584Z", "2022-01-25T17:53:16.9056584Z",



Python plugin

KQL extensibility with inline Python

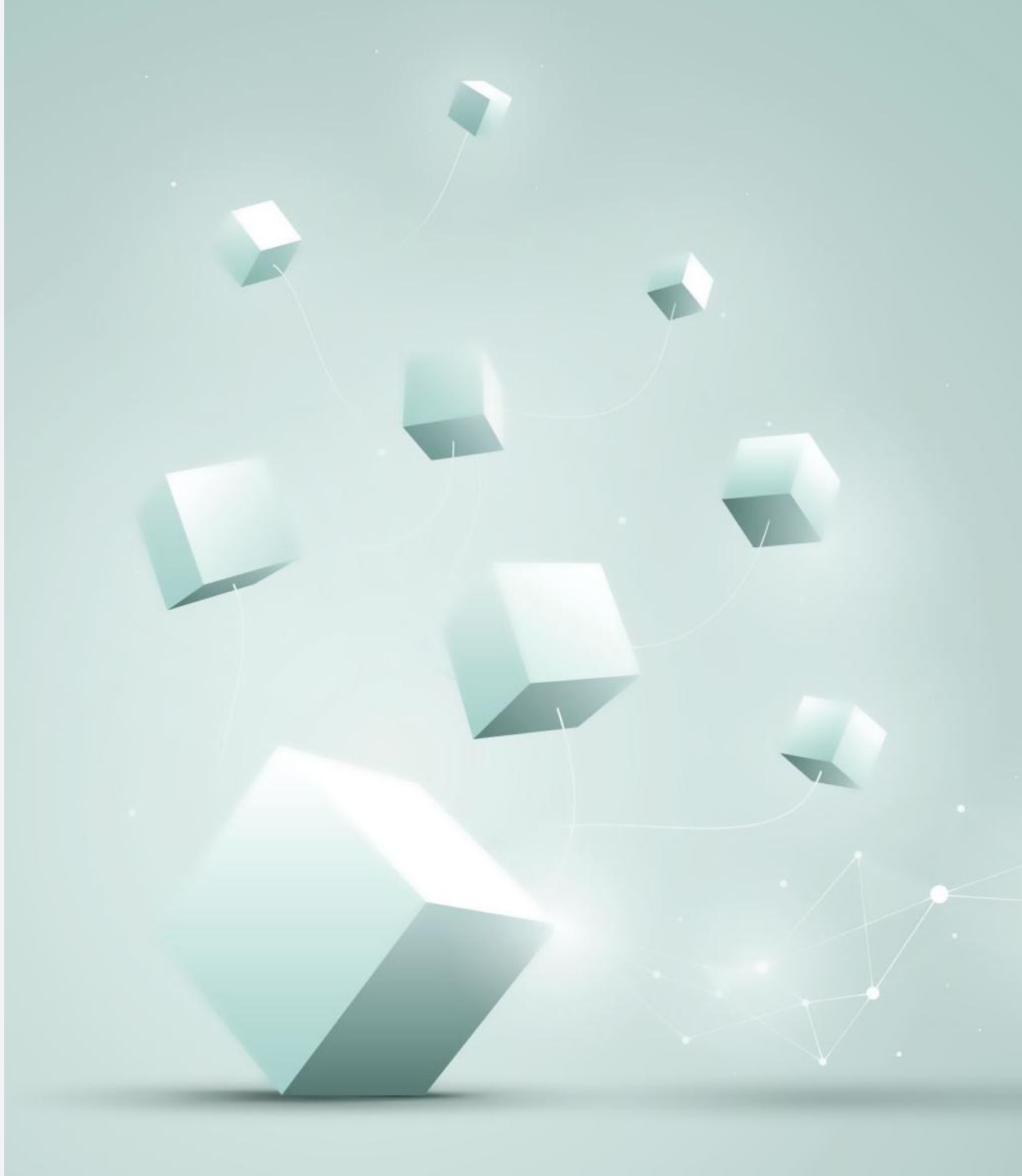
Python engine (sandbox)
Runs on secure local containers on same Kusto engine nodes

Image
Based on Anaconda 5.2.0 + popular packages (tensorflow, keras, torch and more)

Execution
Single node (default) or distributed processing

Extensible
Ability to install custom packages.
Edit & debug in VSCode (client-side integration)

Scalability
Support for scoring with big ML models



ML Connectivity

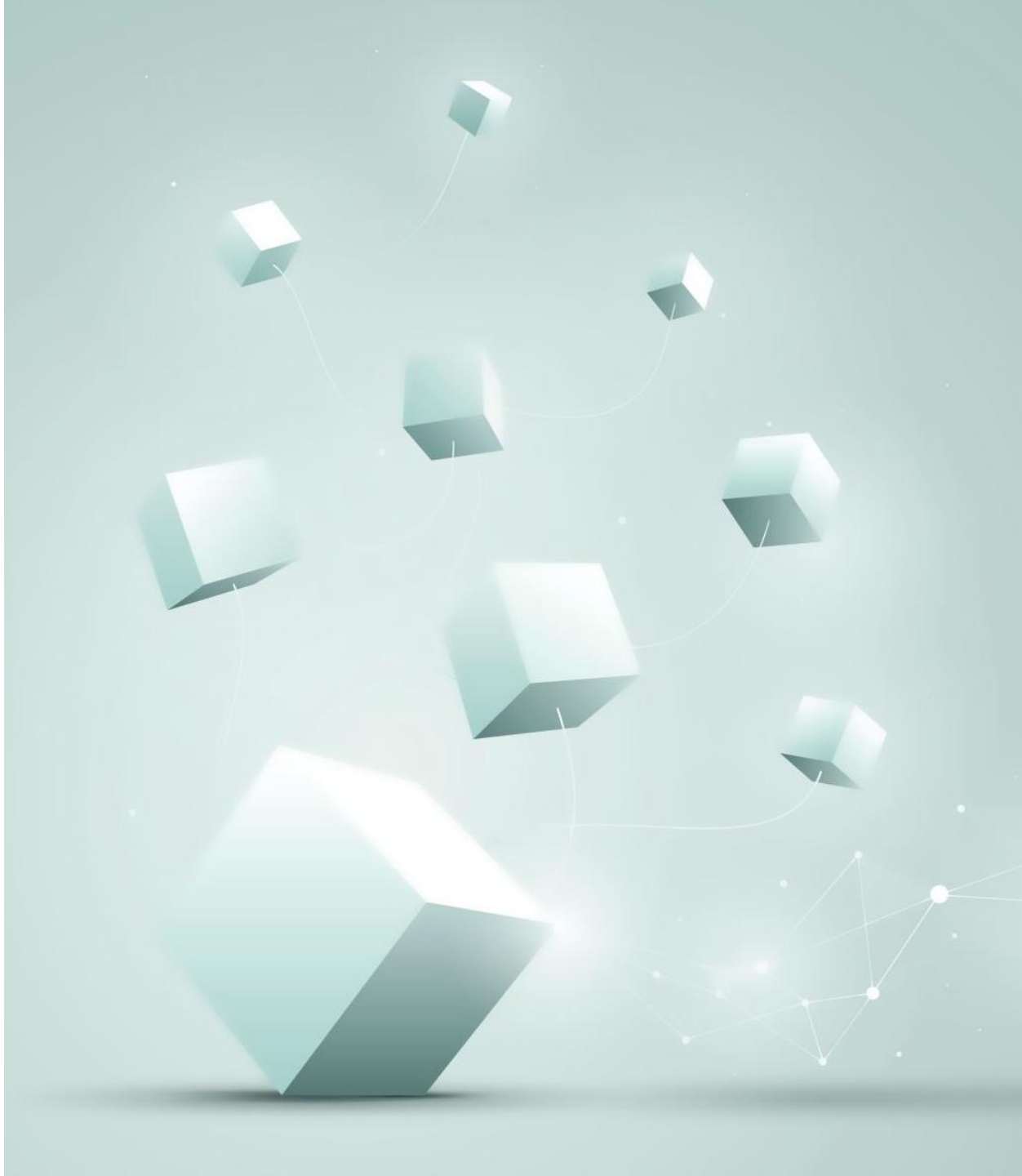
Notebooks with Kqlmagic library
Connect multiple clusters (Kusto, Log Analytics, App Insights). Create rich interactive notebook with Kusto and Python. Use out-of-the-box integration with Python libraries.

Python & R Client SDKs
Access data from Kusto using Python or R libraries

Spark connector
Read data or write data to Kusto using Scala and PySpark

Parquet support
Ingest parquet files from Blob/Data Lake. Export data to Blob/Data Lake as parquet files (with partitioning support)

External tables
Query parquet files from data lake as external



4

Lab 2

Advanced KQL, Policies, Visualization

aka.ms/FabricRTAinADay

Lab 1

Lab 2

Complete the quizzes to earn the badge.

Quizzes can be re-attempted, and will help guide your answers

Lab 1 - Quiz

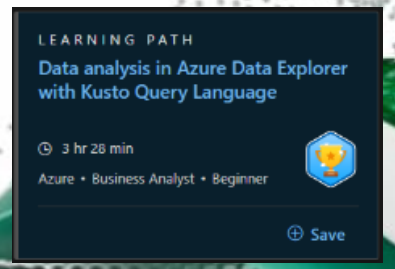
Lab 2 - Quiz

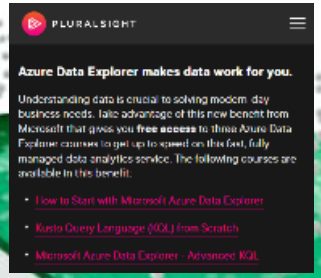
5

Q&A

Open discussion

6





Next steps

Continue your learning

L300, L400, POC, Migration, etc

Microsoft

Thank you