Architectural Summary – Azure Data Explorer (ADX): An Azure Real Time Analytics Service

# Version Information

Satya Komatineni, 12/26/23

Version 1

Contents

[Version Information 1](#_Toc154562747)

[Goal: Architecturally understand ADX 2](#_Toc154562748)

[What is ADX? 2](#_Toc154562749)

[ADX Architecture 3](#_Toc154562750)

[Ingestion 4](#_Toc154562751)

[Storage 5](#_Toc154562752)

[Use 5](#_Toc154562753)

[Positioning of ADX 6](#_Toc154562754)

[ADX UI Tools 7](#_Toc154562755)

[Kusto Explorer 8](#_Toc154562756)

[ADX Web UI 9](#_Toc154562757)

[Azure Notebooks 9](#_Toc154562758)

[Options ADX Dashboards 10](#_Toc154562759)

[What is real time analytics and how do you do that with ADX? 10](#_Toc154562760)

[Characteristics of Real time analytics 10](#_Toc154562761)

[Examples of Real time use cases. 11](#_Toc154562762)

[How to execute real time analytics in ADX 12](#_Toc154562763)

[Traditional Batch Analytics 12](#_Toc154562764)

[Bottom line, ADX, Relevance, Why, Where? 12](#_Toc154562765)

[Questions and Further Research 13](#_Toc154562766)

[Competing technologies to ADX 13](#_Toc154562767)

[Key References 14](#_Toc154562768)

[Credits 14](#_Toc154562769)

# Goal: Architecturally understand ADX

There are way too many back-end data architectures in IT organizations. To quote a few names

1. Data warehouses
2. Data marts
3. Data lakes
4. Delta lakes
5. Master data
6. Big data
7. Hadoop
8. Spark
9. Real time analytics
10. Batch analytics
11. Various forms of reporting

In all this where does “Azure Data Explorer” (ADX) fit?

It seems comprehensive enough that this document takes a deeper architectural look at what ADX is.

You will know in this document:

1. What is ADX?
2. What are its features?
3. How comprehensive is it?
4. What is its eco-system?

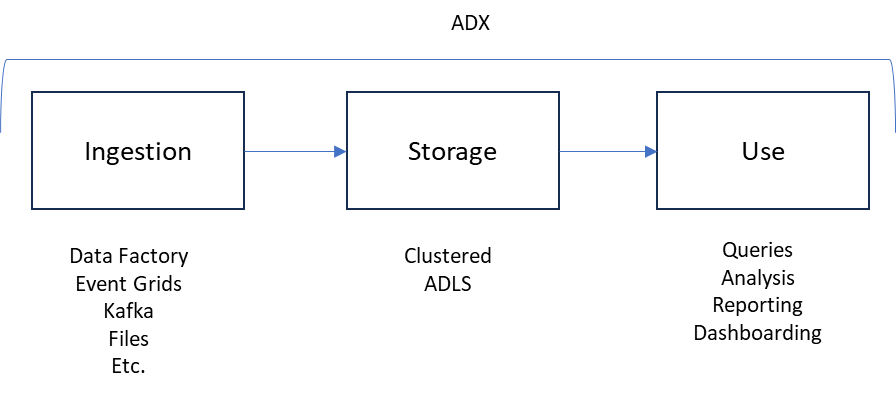
# What is ADX?

Its name “Azure Data Explorer” is a bit of a misnomer when encountering it first. It is just not an explorer of data like a query language, that it is as well.

It is all the following:

1. It is a database (it uses ADLS – Azure data lake Storage)
2. It has connectors to ingest data into the defined logical database.
3. It has a query and programming language called KQL.
4. Functions (like stored procedures) can be written in KQL that can be used for transformations and other needs.
5. It provides interfaces for real time dashboards like PowerBi and Grafana
6. It has connectors for big data processing via Spark.
7. It is a system optimized for real time processing.
8. It is clustered in its architecture.
9. It uses optimized storage mechanisms like columnar data.
10. It is managed, in the sense that you don’t have to explicitly manage the underlying storage.
11. So, it acts like a properly managed database.

# ADX Architecture



Although this picture is a bit misleading it gives a better idea of what ADX is without overly complicating it.

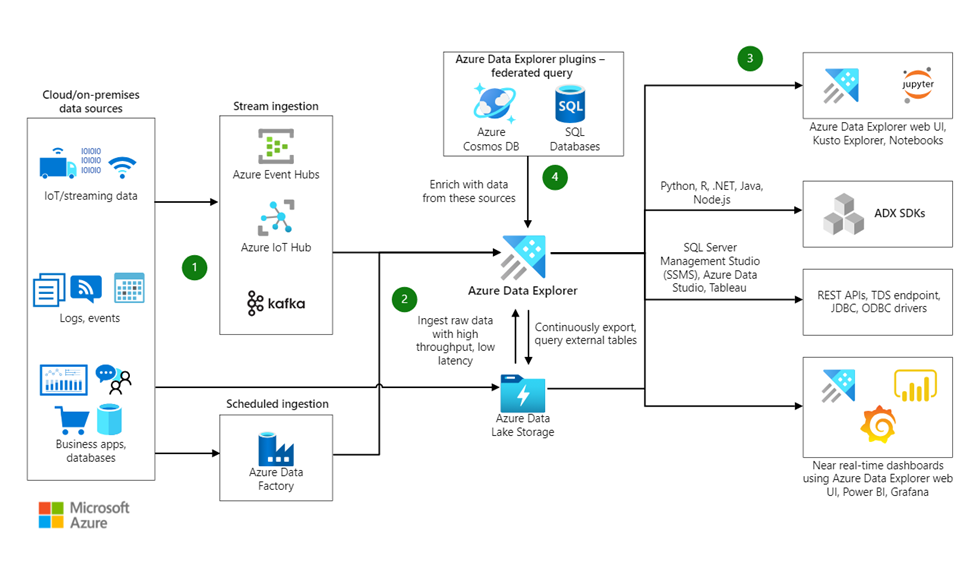
This figure is pointing out to the comprehensive (all together) and managed nature of ADX.

That is, it consists of all 3 parts.

The misleading part is this: The storage is done via ADLS. However, any processing of the data during the ingestion or during the querying is done through a clustered computing that is “outside” of the storage and is essentially what is ADX clustered processing engine.

So, unlike a traditional database ADX separates processing from storage.

Now let’s look at the architecture picture that is on Microsoft learning site.



## Ingestion

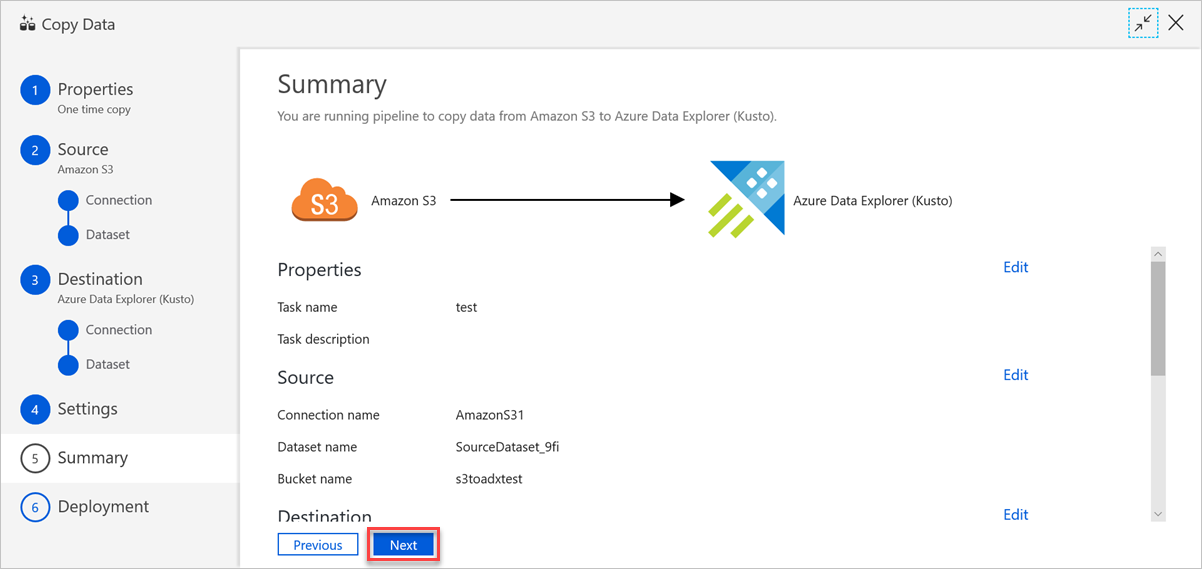
As you can see ADX allows ingestion from a variety of sources.

The way each ingestion is carried out can be different and documented separately.

Imagine ADX as a specialized database into which data is ingested.

Here is an example of how this is done through ADF (Azure Data Factory) via an ADX connector.

Here the ADF pipeline has two components. One reads from the S3 storage, and the data then is fed to the ADX ingest component. The ADX component is configured with details of the ADX cluster, database, table etc.



## Storage

Storage is fully managed by ADX in the underlying ADLS. ADX developers only need to define the high level details of clusters, databases, and tables.

Like any other data storage admins can optimize the data storage in ADLS as allowed and needed by ADLS.

## Use

ADX exposes much of its functionality through KQL: Kusto Query Language.

It is similar to SQL.

Some key qualities are

1. Simpler than SQL
2. Analyze and visualize data
3. Custom functions
4. Time series
5. Temporal data
6. Statistical functions
7. Math operations
8. Readability for expressive queries
9. Performance
10. Data ingestion
11. Data transformation
12. Scale for very large data sets

Here are some samples of KQL

// 1. Select all columns from the "MyTable" table

MyTable

// 2. Count the number of records in "MyTable" where the value in the "Status" column is "Completed"

MyTable

| where Status == "Completed"

| count

**// 3. Calculate the average CPU usage over time**

//This query performs time-series analysis on the "Perf" table, calculating the average CPU usage over 1-hour intervals.

Perf

| summarize AvgCpuUsage = avg(Usage)

by bin(TimeGenerated, 1h)

// 4. Join two tables on a common column

Table1

| join kind=inner (Table2) on CommonColumn

// 5. Summarize and aggregate data by grouping on a specific column

Sales

| summarize TotalSales = sum(Amount) by ProductCategory

Here is an example of a KQL transform and load function

1. // Define a custom function named TransformAndCopy

2. .create function TransformAndCopy() {

3. // Define the transformation logic inside the function

4. SourceTable

5. | extend TransformedColumn = Col1 + Col2 // Sum of numerical columns Col1 and Col2

6. | project TransformedColumn, OtherColumns // Select the transformed and other columns

7. | into TargetTable // Insert the result into the target table

8. }

9.

10. // Execute the custom function

11. TransformAndCopy()

In line 5, “extend” operator is used to add an extra column.

Project operator is used to select the total list of output columns.

# Positioning of ADX

This is how Azure is positioning ADX

1. Fully managed data analytics service
2. Real-time and time-series analysis on large volumes of data streams
3. Expected domains of use include:
   1. Business activities,
   2. Human operations,
   3. Applications,
   4. Websites
   5. Internet of Things (IoT) devices

Among its uses, Azure states:

1. Iteratively explore data on the fly.
2. Improve services and products.
3. Enhance customer experiences.
4. Monitor devices.
5. Boost operations.
6. Quickly identify patterns, anomalies, and trends in your data.
7. Explore new questions and get answers in minutes.
8. Run as many queries as you need.
9. Optimized cost structure.

# ADX UI Tools

The following tools are available:

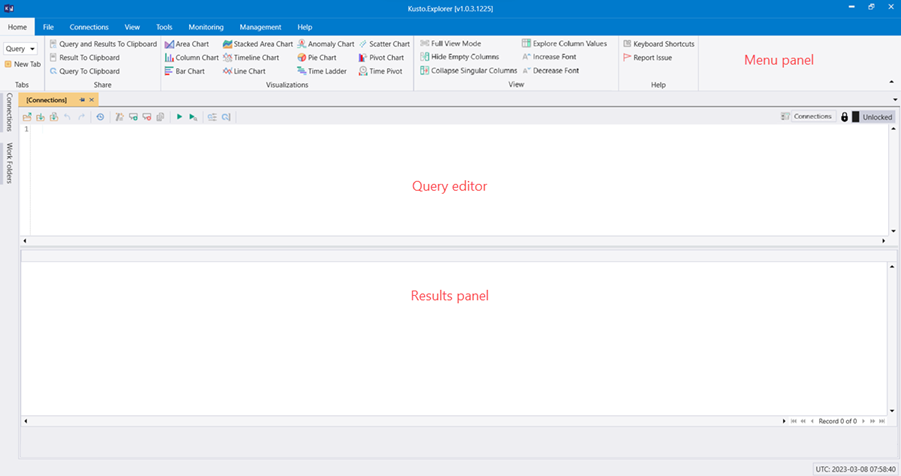
1. Kusto Explorer (Windows Desktop)
2. Azure Data Explorer WebUI
3. ADX Explorer (Jupyter) notebooks

## Kusto Explorer

This is like any other database explorer UI on windows desktop.

Some of its features are:

1. Query Editor: Writing and executing KQL queries.
2. Query Results: Displays query results in a tabular format with options for visualization.
3. Object Explorer: Allows users to explore databases, tables, and other ADX resources in a tree-like structure.
4. Charting and Visualization: Supports basic charting and visualization of query results.
5. Query History: Maintains a history of executed queries.

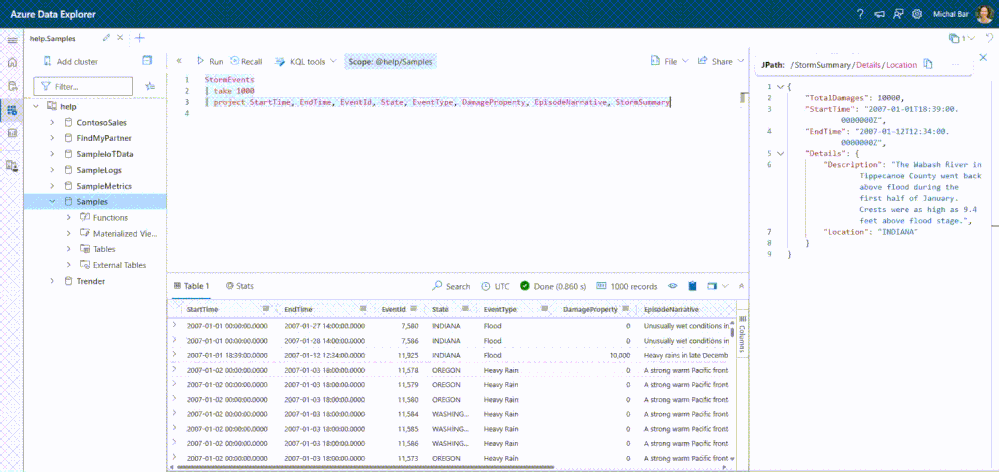


## ADX Web UI

The Azure Data Explorer Web UI is the web-based interface for interacting with Azure Data Explorer.

Its features are

1. Query Editor: Provides a web-based environment for writing and executing KQL queries.
2. Query Results: Displays the results of executed queries in a tabular format.
3. Object Explorer: Offers a tree-like structure to navigate and explore databases, tables, and functions.
4. Charting and Visualization: Supports basic charting for visualizing query results.
5. Query History: Maintains a history of executed queries.
6. Azure Portal Integration: Accessible through the Azure Portal for managing ADX resources.



## Azure Notebooks

The following is from “learn.microsoft.com”.

Quote:

Kqlmagic (extension) extends the capabilities of the Python kernel in Jupyter Notebook so you can run Kusto Query Language (KQL) queries natively.

You can combine Python and KQL to query and visualize data using the rich Plot.ly library integrated with the render operator.

The kqlmagic extension is compatible with Jupyter Lab, Visual Studio Code Jupyter extension, and Azure Data Studio, and supported data sources include Azure Data Explorer, Azure Monitor logs, and Application Insights.

End quote.

# Options ADX Dashboards

Why would you want to collect so much data if you don’t want to summarize and have analytical dashboards, especially for logs, monitoring, and operations.

Although you can use Kusto Explorer to create some charts and graphs or even sophisticated dashboards, these needs to be given access to end users.

The following dedicated tools are more appropriate:

1. **Grafana**: Widely used for monitoring and visualizing time-series data from various sources. Often integrated with specific data stores or monitoring solutions. Highly customizable with a focus on real-time monitoring. Flexible alerting capabilities for monitoring metrics.
2. **Power BI**: Well-suited for business intelligence, data analysis, and reporting with a focus on visualizations and insights.
3. **Kibana:** Kibana is commonly used with Elasticsearch for log and metric visualization. ADX can be integrated with Kibana to analyze and visualize data stored in ADX using Kibana's powerful dashboarding capabilities.
4. **Azure Monitor:** Azure Monitor provides a centralized platform for monitoring and managing Azure resources. ADX integrates seamlessly with Azure Monitor, allowing users to create dashboards, set up alerts, and gain insights into the performance of Azure services.
5. **Microsoft Azure Dashboards:** Azure's built-in dashboards providing a customizable and interactive way to visualize ADX data in real-time.
6. **GCP Data Studio:** Google Cloud Platform's data visualization tool, integrated with ADX for creating interactive dashboards.
7. **Custom Web Applications:** ADX's REST APIs enable developers to build custom web applications tailored for real-time monitoring and visualization.

# What is real time analytics and how do you do that with ADX?

It is the process of analyzing and making sense of data as it is generated or received, providing insights and actionable information instantly or near-instantly.

It involves the rapid processing and interpretation of data streams to extract valuable insights, allowing organizations to respond quickly to changing conditions, make informed decisions, and take immediate actions.

## Characteristics of Real time analytics

**Low Latency:** Real-time analytics systems operate with minimal delay, delivering insights and results in near real-time or with low latency. This is crucial for time-sensitive decision-making.

**Continuous Data Processing:** Data is processed and analyzed continuously as it is generated, rather than in batches. This enables organizations to stay current with the latest information and respond to events as they happen.

**Event-Driven Architecture:** Real-time analytics often involves event-driven architectures, where systems respond to events or triggers in real-time, enabling automated actions or alerts based on specific conditions.

**Streaming Data:** Real-time analytics frequently deals with streaming data, which is a continuous flow of data generated from various sources. Analyzing data in motion allows organizations to gain insights without the need to store and process large volumes of historical data.

**In-Memory Processing:** To achieve low latency, real-time analytics systems often leverage in-memory processing, where data is stored and processed in the system's memory, reducing the need for disk I/O and improving performance.

**Decision Automation:** Real-time analytics can be integrated with decision automation systems, enabling organizations to automate responses to specific conditions or events without manual intervention.

**Monitoring and Visualization:** Real-time analytics systems often include monitoring dashboards and visualizations that allow users to observe and understand the data in real-time. This supports rapid decision-making and trend identification.

**Predictive Analytics:** Some real-time analytics systems incorporate predictive modeling and machine learning algorithms to anticipate future events or trends based on the current data stream.

## Examples of Real time use cases.

Another way of understanding real time analytics is looking at the use cases.

Here are a few examples:

**IoT (Internet of Things) Applications:** Event-driven architectures are extensively used in IoT scenarios, where devices generate events (sensor readings, status updates) that trigger real-time processing and actions.

**E-commerce and Retail:** Order processing, inventory management, and customer interactions in e-commerce platforms often utilize event-driven architectures for scalability and responsiveness.

**Financial Services:** Fraud detection systems use event streams to analyze transactions in real time, identifying and responding to potentially fraudulent activities.

**Healthcare:** Patient monitoring, health record updates, and real-time alerts in healthcare systems benefit from event-driven architectures for timely responses and data synchronization.

**Telecommunications:** Call detail record (CDR) processing, network monitoring, and automated responses to network events are common in the telecommunications industry.

**Logistics and Supply Chain:** Tracking shipments, managing inventory, and coordinating logistics operations involve event-driven systems to handle real-time updates and changes.

**Media and Entertainment:** Content delivery networks (CDNs) leverage event-driven architectures to efficiently distribute and update content, adapting to changing demand dynamically.

**Social Media Platforms:** Real-time notifications, updates, and user interactions on social media platforms rely on event-driven models for responsiveness and scalability.

**Gaming:** Online gaming platforms use event-driven architectures for handling in-game events, matchmaking, and real-time player interactions.

**Energy and Utilities:** Smart grid applications use event-driven systems to monitor and manage energy distribution, detect faults, and respond to changes in demand.

**Travel and Hospitality:** Booking systems, reservation updates, and travel itinerary management benefit from event-driven architectures to provide timely information to users.

**Manufacturing:** Predictive maintenance in manufacturing plants uses event-driven systems to monitor equipment health in real time and schedule maintenance proactively.

**Government and Public Services:** Emergency response systems use event-driven architectures to process and respond to events such as natural disasters, accidents, or public safety incidents.

**Education Technology:** Learning management systems (LMS) and educational platforms use event-driven architectures to handle user interactions, content updates, and progress tracking.

## How to execute real time analytics in ADX

The steps are:

1. Ingest data, events, and messages using any number of low-latency techniques
2. Curate, aggregate, count data in real time using KQL processes and functions.
3. Deal with results
   1. Save the results in target decision tables.
   2. Send them to dashboards.
   3. React with real time recommendations.

## Traditional Batch Analytics

Although Spark has a connector to ADX it is not clear if ADX is a suitable choice for traditional analytics.

Azure has a traditional warehousing solution based on SQL server called Synapse that serves this need.

Synapse has connectors to load data from ADX to synapse.

# Bottom line, ADX, Relevance, Why, Where?

Now I am at a point where I can make some calls.

One holy grail in backend data is how to avoid copying data between “different schemas”.

Same schema copies are ok. As it is often done for fault tolerance, reporting, and scaling.

Different schemas are also ok if they are virtual. For example defining on top of a delta lake.

“copy between different schemas” is a problem because that is going to be explicit ETL that needs to be hand coded and maintained.

ADX does not particularly solve the problem where you like to use the ingested data for both real time, and batch analytics without further modifying the schema. It may also not allow as easily virtual schemas.

However, once the data is ingested by ADX, perhaps it can solve for the following:

1. Near real time queries
2. Portal APIs
3. Reporting
4. Poorman’s warehouse or a delta lake

Two complementary or competing technologies one must look at are:

1. Synapse analytics (A managed warehouse like snowflake)
2. Delta lake (The new kid on the block that can serve both real time and batch?)

Why then ADX and where?

For analyzing numerous event centric use cases there doesn’t seem to be as compelling a system as ADX.

The con is that such a solution is a dedicated one requiring its own investment!

# Questions and Further Research

1. ADX is a solution that is in Azure cloud. What do you do if on premise? What are good alternatives for real time analytics that are as comprehensive?
2. What does an eco-system centered around a stream processing technologies like Flink look like?
3. Are there implementations of Delta Lake that is mature enough to address this space?
4. How does AWS Kinesis compare?
5. How does Digital strategies of today gain from Real time analytics? What tools are used in this space?
6. Compare and contrast with Azure Stream Analytics

# Competing technologies to ADX

1. Apache Flink: A stream processing engine at its core. Need to see how to provide and manage storage.
2. Elastic: A proven large volume search engine and eco-system. A final solution still have to be stitched together. It is not clear if there are better solutions on premise!
3. InfluxDB: Time series database widely used for monitoring metrics and events, real-time analytics, and other IoT and analytics applications.
4. SAP HANA: A high performance in memory database
5. Redshift: More of a batch analytics warehouse
6. Oracle stream analytics
7. Apache Druid: Time series database
8. Azure Stream Analytics: Run complex queries on data in motion. This is similar to Apache Flink
9. BigQuery: It integrates real-time analytics, managed storage layer, and can ingest streaming data dynamically.
10. Snowflake: Popular optimized, managed, cloud-based warehouse. May be supporting delta lake like semantics

# Key References

1. ADX Home page: [Link](https://azure.microsoft.com/en-us/products/data-explorer/)
2. ADX Documentation page: [Link](https://learn.microsoft.com/en-us/azure/data-explorer/)
3. Integration example of how ADX interacts with ingestions tools. An Azure Data Factory Example: [Link](https://learn.microsoft.com/en-us/azure/data-explorer/data-factory-load-data)
4. How managed storage works in ADX. An architectural note: [Link](https://learn.microsoft.com/en-us/azure/data-explorer/how-it-works)
5. How to use Jupyter notebooks for exploring ADX data: [Link](https://learn.microsoft.com/en-us/azure/data-explorer/kqlmagic?tabs=code)
6. KQL Home page to learn: [Link](https://learn.microsoft.com/en-us/azure/data-explorer/kusto/query/)

# Credits

1. ChatGPT
2. Microsoft Learn