



Yellowbrick

Stacking up
versus
Azure Synapse /
Microsoft Fabric



Azure Synapse / Microsoft Fabric

Azure Synapse is a collection of related but independent data technologies such as Dedicated SQL Pools (SQL Data Warehouse), SQL Serverless (Fabric Warehouse), Synapse Spark (Fabric Data Engineering), Data Explorer (Fabric Real-Time Analytics), and Data Factory. Yellowbrick competes primarily with Synapse SQL Dedicated Pools and SQL Serverless components, and to a lesser extent with Data Explorer.

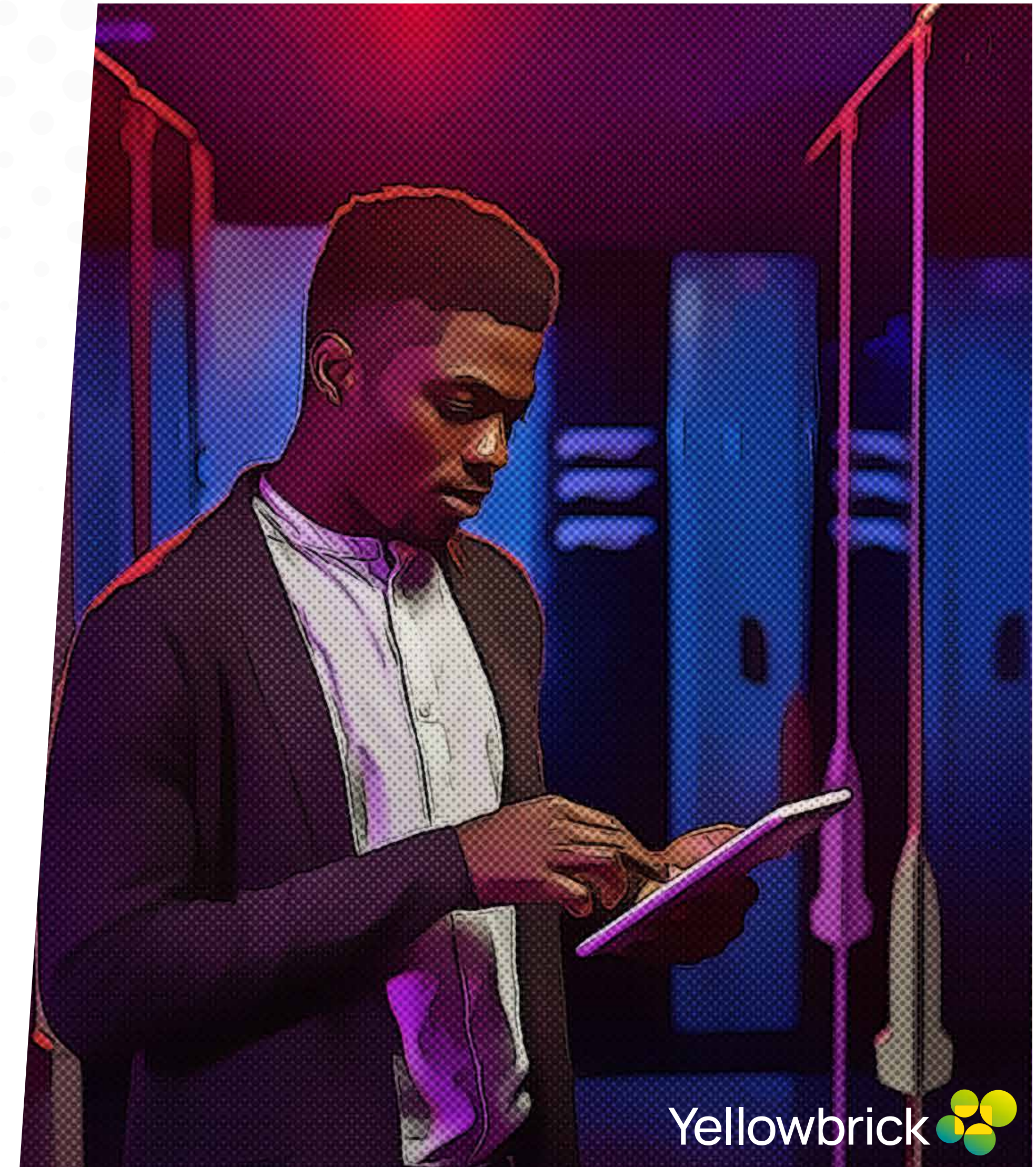
Both Yellowbrick and Synapse SQL capabilities use MPP techniques to distribute SQL query workloads across multiple nodes to execute queries against large volumes of data, but the similarities end there.

The elephant in the room is that after many years of underinvestment, Azure Synapse has finally been discontinued by Microsoft in favor of Microsoft Fabric. Microsoft Fabric Warehouse is a brand-new service requiring migration and a rethink of security and operating procedures.

Azure Synapse does not deliver the modern elastic multi-cluster experience that customers need and expect. Unlike Yellowbrick, SQL Pools are limited to a single database with no cross-database queries. Despite many years of promising multi-cluster support, SQL Pools only support a single dedicated compute Pool/Cluster. The result – Synapse customers require costly over-provisioning.

Yellowbrick supports multiple independent compute clusters – for example for different teams – facilitating cross-charge and right-sizing.

SQL Pools use opaque DWU units, with increments in factors of 60. Yellowbrick allows each cluster to grow in single node increments and distributes data properly between active nodes.



Azure Synapse / Microsoft Fabric

Yellowbrick is also completely transparent about CPU, memory, and cache utilization.

Compute for Synapse is combined into DWU (or capacities) and requires Synapse-specific pre-purchase agreements, whereas Yellowbrick uses standard Azure Compute allowing the use of more flexible Azure Reserved Instances. The result – better utilization of enterprise discounts and no opaque markup of compute, network, or storage.

Fabric shifts customers to a new Lakehouse model based around Synapse SQL Serverless. The performance and cost unpredictability of SQL Serverless make it useful only for occasional ad hoc queries over the data lake where performance is not a concern.

Yellowbrick runs completely in your VNET (or VPC) simplifying security and firewall configuration and can be set up without any public IP endpoints. Synapse requires complex managed endpoints and routing to deliver a secure implementation. Parts of the Synapse service are multi-tenant and run in Microsoft Managed, requiring complex security audits and lengthy approval processes from your security organization.

Managing security for Synapse requires complex coordination of Azure RBAC, Synapse RBAC, Storage RBAC, and in-database object security. With Yellowbrick all security is managed through standard SQL roles and privileges – both simplifying operations and governance. Yellowbrick supports AAD authentication and OAuth as well as database authentication.

If you have been using Azure Synapse or are looking to upgrade your SQL Server-based data warehouse, Yellowbrick provides a more capable, more modern, more scalable, and more

cost-effective database platform in either cloud or on-premises. Yellowbrick also offers the flexibility to enable BI users to run tens of concurrent simpler queries with fewer resources assigned to maximize the interactive query performance, e.g., in Tableau or Power BI.

	Yellowbrick	Synapse
Support for OAuth and AAD	✓	✓
Highly compressed columnstore storage format	✓	✓
Full separation of storage and compute	✓	✓
Hybrid row / columnstore	✓	✗
No secondary indexes and materialized views for performance	✓	✗
Scale one node at a time	✓	✗
Multi-cluster support	✓	✗
Load-balanced scale-out cluster support	✓	✗
Designed for high concurrency	✓	✗
Full transparency of storage and compute resources	✓	✗
Runs on modern cloud-native Kubernetes architecture	✓	✗
Long-term backup retention	✓	✗
Basic resource governance: CPU / memory	✓	✓
Advanced workload management using compile time or run-time metrics	✓	✗
Support for Kafka ingest without Stream Analytics	✓	✗
Queries compiled to machine code	✓	✗
Avoids inefficient in-memory buffer pool	✓	✗
Ingest data from data lake in CSV and Parquet	✓	✓
Serverless query over file	✗	✓