

#FABCONSQLCON2026

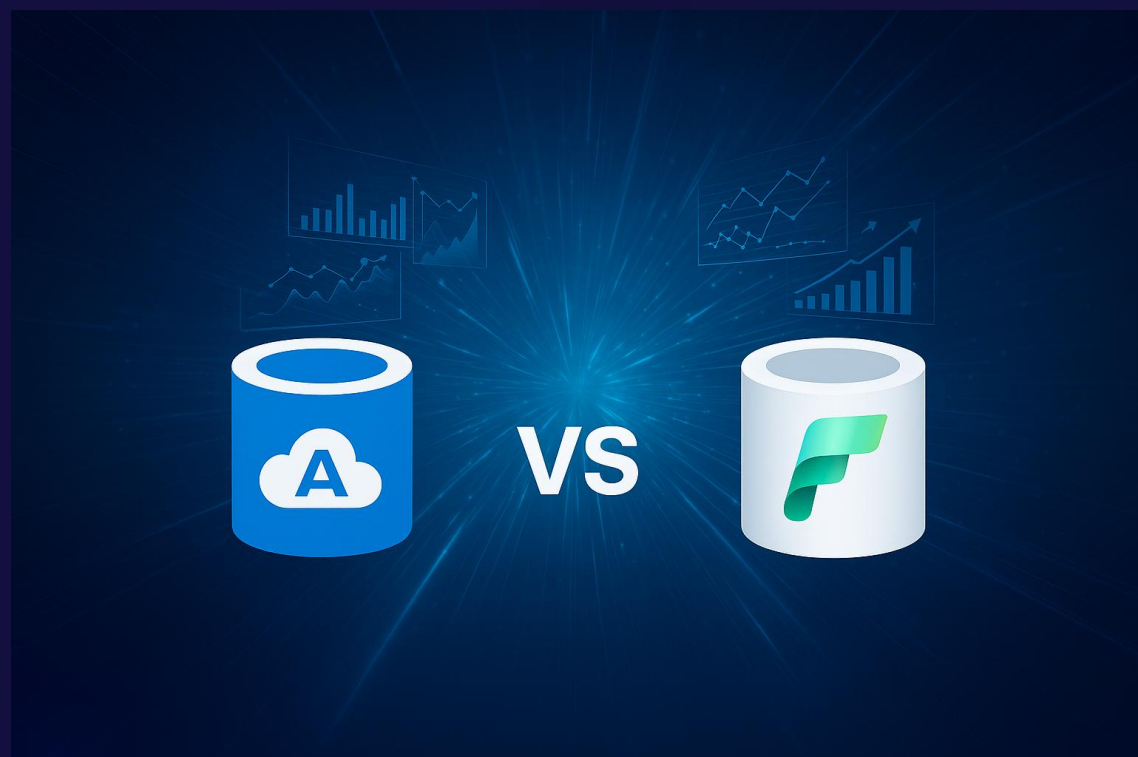
FABCON

Microsoft Fabric
COMMUNITY CONFERENCE

SQLCON

Microsoft SQL
COMMUNITY CONFERENCE

ATLANTA MARCH 16 - 20, 2026



From Dev to Dashboard - Azure SQL vs Fabric SQL in Practice

Matthias Nohl | b.telligent
March 18th, 2026

Your Speaker



Strategic Partner Manager &
Microsoft Community Lead @
b.telligent Deutschland GmbH

My Job



Munich

My Hometown

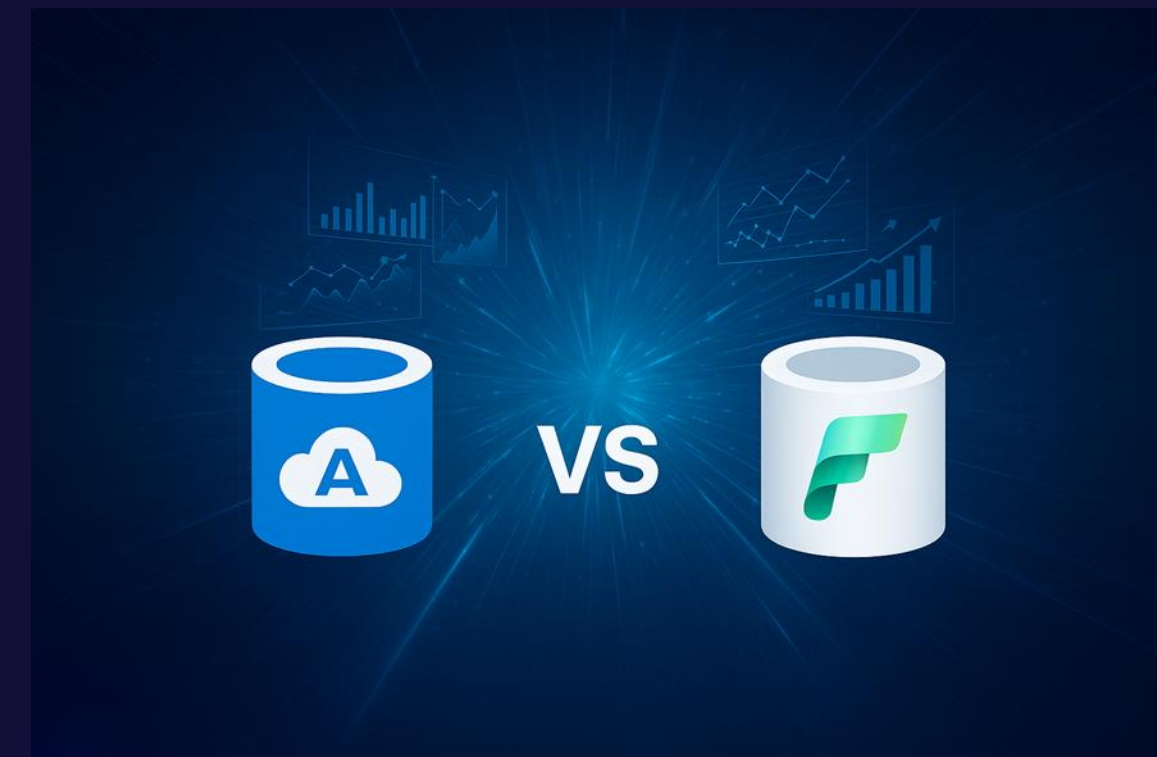


in [linkedin.com/in/matthiasnohl](https://www.linkedin.com/in/matthiasnohl)

Let's Connect

Agenda

- | | |
|----|--|
| 01 | Introduction |
| 02 | Architectures:
Azure SQL vs Fabric SQL |
| 03 | Features <ul style="list-style-type: none">• Security• Governance• Integration• Development |
| 04 | Demo – Let's start the journey |
| 05 | Costs & Billing |
| 06 | Decision framework |



01 - Introduction



The “Why”

Azure SQL

Databases

Includes: single databases, elastic pools, and database servers



Managed instances

Includes: single instances, instance pools



SQL virtual machines

Includes: 60+ available images combining SQL Server 2008-2019 and a variety of available OS and license types



Fabric SQL



Announcing SQL database in Fabric (Generally Available)

AI

Announcements

Databases



November 18, 2025 by [Anna Hoffman](#)

25,138 Views

in Share

Tweet

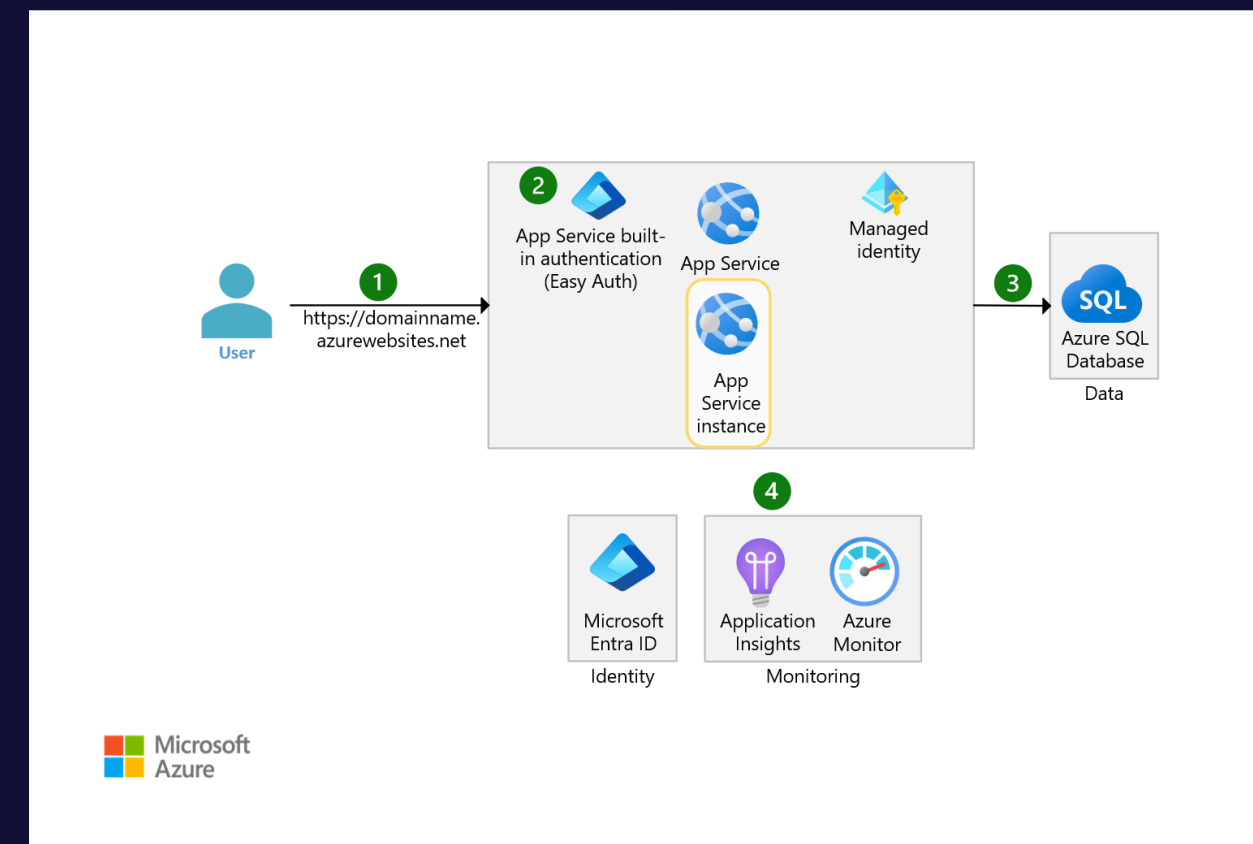
Like

The “Why” – Different Architecture Patterns

Basic BI Workload

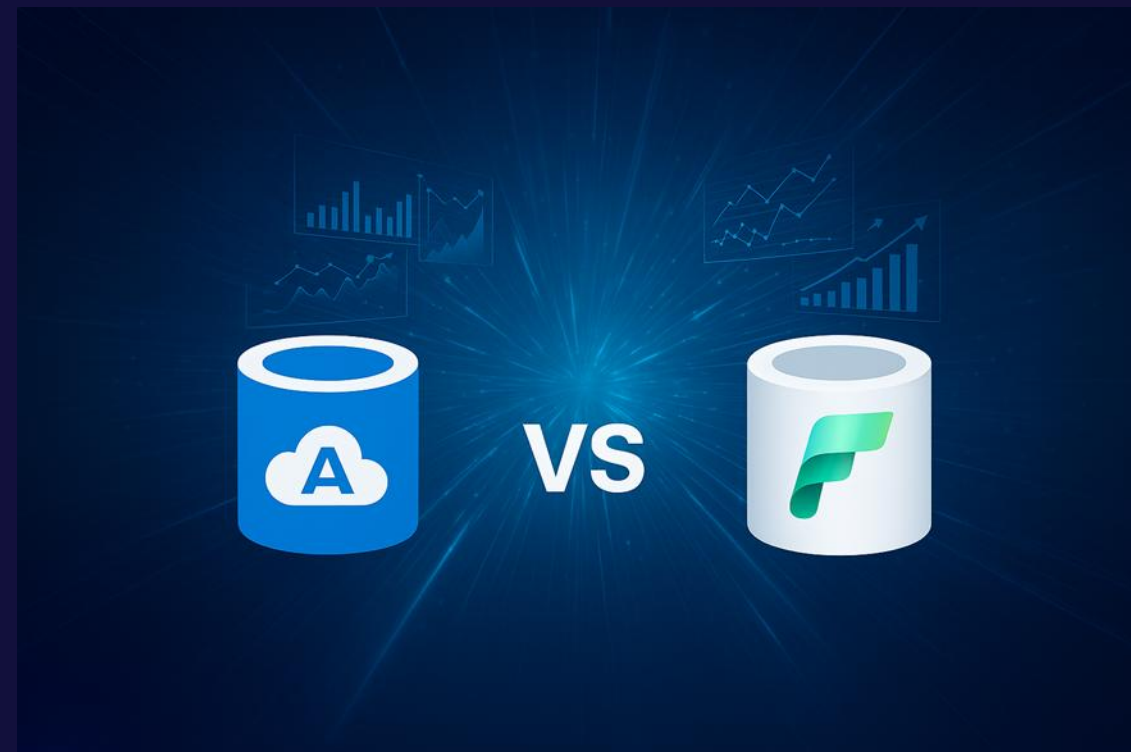


Basic web application



Source: [Basic Web Application - Azure Architecture Center](#) | [Microsoft Learn](#)

02 - Architectures: Azure SQL vs Fabric SQL



Architecture - Facts First

Azure SQL

„Build applications using a fully managed SQL database with built-in performance and reliability.”

vCore-based:

- General Purpose (serverless / provisioned) – Gen 5 (80/128 vCores, 240/625GB RAM, 4TB), FSv2, DC
- Hyperscale (serverless / provisioned) – Gen 5 (80 vCores, 240GB RAM)
- Business Critical – Gen 5 (128 vCores, 625GB RAM, 4 TB), DC

DTU-based purchasing model:

- Basis
- Standard
- Premium

Source: [What Is a Single Database? - Azure SQL Database | Microsoft Learn](#)

Architecture - Facts First

Fabric SQL

„Build modern cloud apps that scale on an intelligent, fully managed database.”

vCore-based:

- Up to 32 vCores
- 4 TB Storage
- 1,024 GB Tempdb
- 50 MB/s Lograte
- Read-only Replica: 1 (SQL analytics endpoint)
- 150 Databases per workspace

General Purpose standard-series (Gen5) (part 3 of 3)

Compute sizes (service level objectives, or SLOs) for General Purpose standard-series databases follow the naming convention GP_Gen5_ followed by the number of vCores.

The following table covers these SLOs: GP_Gen5_24, GP_Gen5_32, GP_Gen5_40, GP_Gen5_80, and GP_Gen5_128:

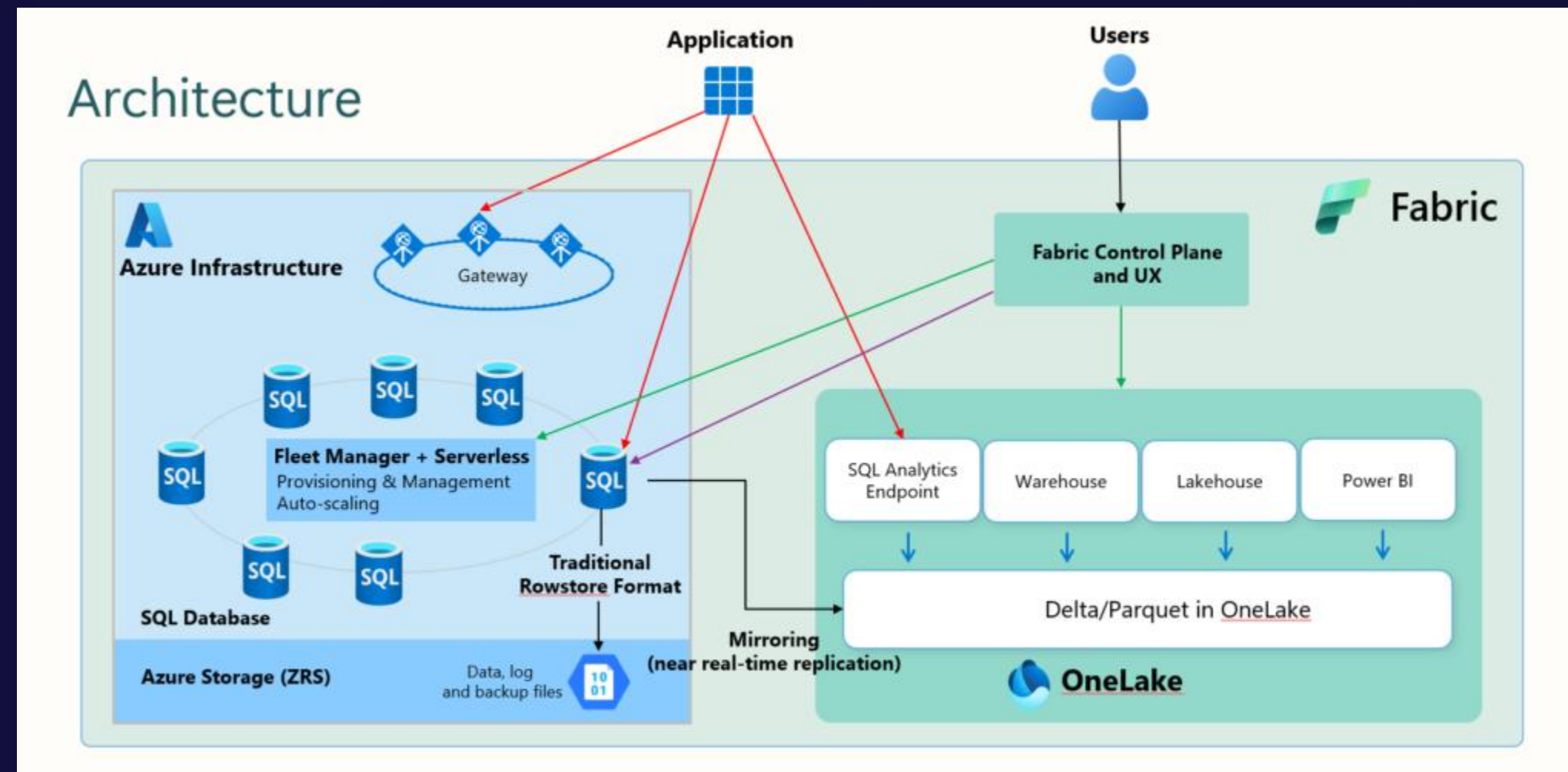
Expand table

vCores	24	32	40	80	128
Hardware	Gen5	Gen5	Gen5	Gen5	Gen5
Memory (GB)	124.6	166.1	207.6	415.2	625
Columnstore support	Yes	Yes	Yes	Yes	Yes
In-memory OLTP storage (GB)	N/A	N/A	N/A	N/A	N/A
Max data size (GB)	4096	4096	4096	4096	4096
Max log size (GB) ¹	1024	1024	1024	1024	1024
Tempdb max data size (GB)	768	1024	1280	2560	2560
Storage type	Remote SSD	Remote SSD	Remote SSD	Remote SSD	Remote SSD
Read IO latency ⁴	5-10 ms	5-10 ms	5-10 ms	5-10 ms	5-10 ms
Write IO latency ⁴	5-7 ms	5-7 ms	5-7 ms	5-7 ms	5-7 ms
Max data IOPS ²	7,680	10,240	12,800	12,800	16,000
Max log rate (MiB/s)	50	50	50	50	50
Max concurrent workers	2,400	3,200	4,000	8,000	12,800
Max concurrent logins	2,400	3,200	4,000	8,000	12,800
Max concurrent external connections ³	150	150	150	150	150
Max concurrent sessions	30,000	30,000	30,000	30,000	30,000
Number of replicas	1	1	1	1	1
Multi-AZ	Yes	Yes	Yes	Yes	Yes
Read Scale-out	N/A	N/A	N/A	N/A	N/A
Included backup storage	1X DB size	1X DB size	1X DB size	1X DB size	1X DB size

Source:
[Limitations for SQL database - Microsoft Fabric | Microsoft Learn](#)
[Single database vCore resource limits - Azure SQL Database | Microsoft Learn](#)

Architecture - Facts First

What do you get in Fabric?



Source: [SQL Database in Fabric - What, Why, and How? - Data Mozart](#)

Architecture - Facts First

Fabric SQL

vCores to CUs Mapping

“1 Fabric capacity unit corresponds to 0.383 Database vCores, or, 1 Database vCore corresponds to 2.611 Fabric capacity unit.”

SKU	Capacity Units (CU)	SQL database vCores per second
F2	2	0.766
F4	4	1.532
F8	8	3.064
F16	16	6.128
F32	32	12.256
F64	64	24.512
F128	128	49.024
F256	256	98.048
F512	512	196.096
F1024	1024	392.192
F2048	2048	784.384

Source: [Billing and Utilization Reporting - Microsoft Fabric](#) | [Microsoft Learn](#)

Architecture - Facts First

Deep-Dive? Reliability in Microsoft Fabric

- Availability zone support
- BCDR → Phase 2: Disaster failover → Not defined for “Fabric SQL”
 - But: **Experience-specific disaster recovery guidance**
 - *“To protect against a regional failure, users of SQL databases can take proactive measures to periodically export their data and use the exported data to recreate the database in a new workspace when needed.”*
- SLA 99,9% Uptime

Maximum Available Minutes: The sum of all minutes that a given, individual Capacity has been instantiated during an Applicable Period for a given tenant.

Downtime Minutes*: The total accumulated minutes in an Applicable Period for a given Capacity, after its creation, or before it is deprovisioned when the Capacity is unable to be utilized in all applicable Fabric features listed below:

SQL Database

 - *"SQL Database" means any SQL database created in Microsoft Fabric workspace.*
 - *Open Database: Open and view a SQL database in the service.*

Source:

[Reliability in Microsoft Fabric | Microsoft Learn](#)

[Microsoft Fabric disaster recovery experience specific guidance - Microsoft Fabric | Microsoft Learn Licensing Documents](#)

Architecture - Facts First

How does it work?

- SQL database in Fabric autoscales compute and provisions a minimum of 2 GB memory, billed as compute while the database is online.
- After 15 minutes of inactivity, all compute resources including CPU and memory are released, and the compute bill is zero.

Be careful: Scales means...Fabric F64 → 24.512 vCores Max

Source: [Billing and Utilization Reporting - Microsoft Fabric](#) | [Microsoft Learn](#)

03 - Features



Security

Feature / Capability	◆ Fabric SQL Database (SaaS)	◆ Azure SQL Database (PaaS)
Always Encrypted	✗ Not available	✓ Available
Transparent Data Encryption (TDE)	✓ Available (with CMK)	✓ Available
Application Roles	✗ Not available	✓ Available
EXECUTE AS	✗ Not available	✓ Available (EXECUTE AS USER)
Authentication	Entra ID only; no SQL logins	Entra and SQL logins supported
Network Security	Private Links (tenant-level/workspace-level) for SQL Endpoint	Private Links, VNet integration

Source: [Supported scenarios for workspace private links - Microsoft Fabric | Microsoft Learn](#)

Governance

Feature / Capability	◆ Fabric SQL Database (SaaS)	◆ Azure SQL Database (PaaS)
Geo-Replication / Failover Groups	✗ Not available	✓ Available
Auditing	✓ Audit logs to OneLake (preview)	✓ Audit logs to Azure Monitor / Log Analytics
Compliance	✓ ISO 27001/17/18, SOC 1/2/3, HIPAA (BAA)	✓ ISO, SOC, HIPAA, PCI, FedRAMP, etc.
Backup & Restore	Automatic, 7-day retention (35 max), no geo or cross-workspace-restore	Automatic + manual restore, LTR, geo-restore

- **Access Control**

- Access via Fabric workspace roles (Admin, Member, Contributor, Viewer) and item-level permissions
- Native SQL permissions and database roles (“Manage SQL Security”) also supported

- **Data Classification & Labels**

- Integration with Microsoft Purview
- Sensitivity labels and protection policies supported

Integration

Feature / Capability	◆ Fabric SQL Database (SaaS)	◆ Azure SQL Database (PaaS)
Power BI Direct Lake	✓ Native support for real-time analytics without import	✗ Only via Import or DirectQuery
Power BI Write Back	✓ via Translytical Task Flows (User Data Functions)	✗ Not available
Spark / Notebooks / Real-Time Analytics	✓ Seamless integration via OneLake	✗ Requires separate services
Vector Data Type & RAG Support	✓ Available (for AI/ML use cases)	✓ Available (for AI/ML use cases)
GraphQL API Generation	✓ Available (auto-generated)	✗ Not available

- **SQL Analytics Endpoint**

- Read-only TDS endpoint for analytical workloads (MPP engine)

- **Cross-Database Queries**

- JOINS across Fabric SQL DBs, Warehouses, Lakehouses via 3-part naming

- **Data Virtualization (Ext. Files)**

- Query external files (Parquet, CSV, JSON) via OPENROWSET & External Tables through OneLake Shortcuts

- **Spark & Data Engineering**

Access SQL data via Delta (OneLake) or JDBC in notebooks & pipelines

- **Mirroring & Pipelines**

- Near real-time sync from external SQL sources (e.g., on-prem, Azure SQL MI) via CDC & pipelines

Development

Project Structure & Source Control

-  Azure SQL: Uses SQL Database Projects (SSDT or VS Code) → builds a .dacpac. No built-in Git integration.
-  Fabric SQL: Optional built-in Git integration at workspace level (Azure DevOps/GitHub). Changes can be committed or pulled directly via UI.

Build & Validation (CI)

-   Both: Build SQL project to generate a .dacpac (via dotnet build or VS/VS Code). Validates syntax, object references, and target platform compatibility (Azure SQL or Fabric).

Deployment (CD)

-  Azure SQL: Deploy .dacpac via SqlPackage or Azure DevOps task (SqlAzureDacpacDeployment). Auth via SQL login or Entra (incl. service principal).
-  Fabric SQL: Only Entra authentication (e.g., service principal with workspace access). Deployment via SqlPackage with Entra connection string and e.g. fabric-cicd Python tool (artefact)

Configuration & Secrets

-  Azure SQL: Connection strings, passwords, etc., managed via Azure Key Vault or pipeline secrets.
-  Fabric SQL: Workspace-specific server names, Entra SPN credentials via Key Vault. No SQL logins supported.

Testing

-   Both: tSQLt unit tests in Docker/local SQL Server/SQL MI; integration tests against Azure / Fabric SQL.

Rollback

-  Azure SQL: Point-in-time restore (7–35 days), DACPAC rollback possible.
-  Fabric SQL: PITR (7 days), no overwrite – restore creates new DB. Rollback typically via re-deploying previous DACPAC.

Environments

-  Azure SQL: Separate servers/DBs per environment.
-  Fabric SQL: Separate workspaces (Dev/Test/Prod), Git branch per workspace, deployment via merge + update

Monitoring

Built-in Metrics

-  Azure SQL: Exposed via Azure Monitor: CPU, DTU/vCore usage, storage, sessions, etc.
-  Fabric SQL: Capacity-level metrics via Fabric Capacity Metrics App; limited per-DB granularity

Query Performance Insights

-  Azure SQL: Query Store, DMVs, and Query Performance Insight in Azure Portal
-  Fabric SQL: DMVs available; Performance Dashboard for SQL database

Auditing

-  Azure SQL: Full auditing to Azure Monitor, Log Analytics, Event Hub, or Storage
-  Fabric SQL: SQL auditing in preview; logs to OneLake; limited configuration options

Alerts & Diagnostics

-  Azure SQL: Alerts via Azure Monitor; diagnostic settings configurable per DB
-  Fabric SQL: Performance Dashboard supports Alerts (ongoing & pending); relies on capacity-level monitoring or external tools

Telemetry Access

-  Azure SQL: Rich telemetry via Azure Monitor, Application Insights, and Log Analytics
-  Fabric SQL: Limited telemetry; some logs available via Fabric Admin APIs or audit preview

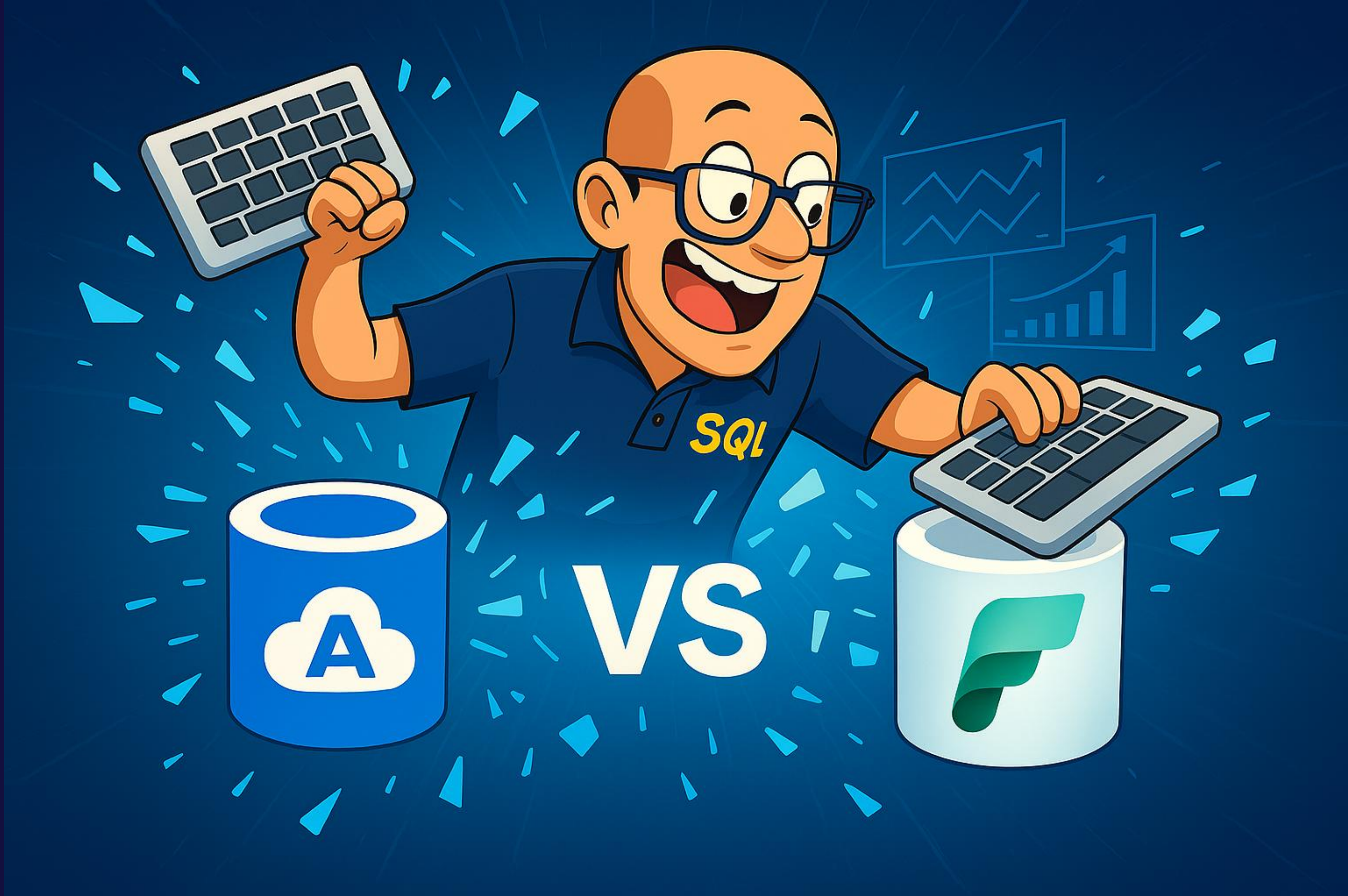
Performance Tuning Insights

-  Azure SQL: Automatic tuning recommendations, index suggestions, plan regression detection
-  Fabric SQL: Auto-indexing enabled by default; no tuning dashboard yet

Third-party Integration

-  Azure SQL: Supported via Azure Monitor APIs, Log Analytics, and diagnostic logs
-  Fabric SQL: Limited; APIs emerging, but not yet mature for full external observability

Source: [Monitor Performance Using DMVs - Azure SQL Database & SQL database in Microsoft Fabric | Microsoft Learn](#)







04 - Demo – Let's **start** the journey

05 - Costs & Billing



Costs & Billing

- $I_i \in \{0,1\}$: online indicator $I_i = 0$ (inactivity $\geq 15\text{min}$)
- Constant k : 2.611 - CU per Database vCore
- Δt_i : interval duration in seconds
- v_i : vCores used each second
- m_i : Memory (GB) used each second
- Normalization $r = 3\text{GB}$ per vCore for comparing memory to CPU
- Minimum online memory $m_{min} = 2\text{GB}$

Then the billed CU-seconds for interval i are:

$$CU_i = \begin{cases} 0, & I_i = 0 \\ k\Delta t_i \max\left(v_i, \frac{\max(m_i, m_{min})}{r}\right), & I_i = 1 \end{cases}$$

Source: [Billing and Utilization Reporting - Microsoft Fabric](#) | Microsoft Learn

Costs Example

Time Interval (min)	vCores used each second	Memory GB used each second	Compute dimension billed	CU Seconds Billed (Simple Explanation)
00:00 – 0:05	2	3	vCores used	Used 2 vCores for 300 seconds. CPU was higher than memory, so billing is based on vCore usage. Calculation: $2 \times 300 \times 2.611 = 1,567$ CU seconds.
0:05 – 0:15	1	6	Memory used	Used 6 GB memory for 600 seconds. Memory exceeded CPU, so billing is based on memory usage. Memory-to-vCore ratio: 6 GB = 2 vCores. Calculation: $2 \times 600 \times 2.611 = 3,133$ CU seconds.
0:15 – 0:30	0	2	Minimum memory allocated	Minimum 2 GB memory is allocated to keep the database ready, even when idle. Memory-to-vCore ratio: 2 GB = 0.6666 vCore. Calculation: $0.6666 \times 900 \times 2.611 = 1,567$ CU seconds.
0:30 – 0:60	0	0	No compute billed after 15 minutes of inactivity	No activity for 15 minutes, so all compute, including CPU and memory, is released. Compute billing stops. Calculation: 0 CU seconds.

Total Estimated CU seconds: 6,267 CU seconds

Source: [Billing and Utilization Reporting - Microsoft Fabric | Microsoft Learn](#)

Cost Comparison

Assumptions:

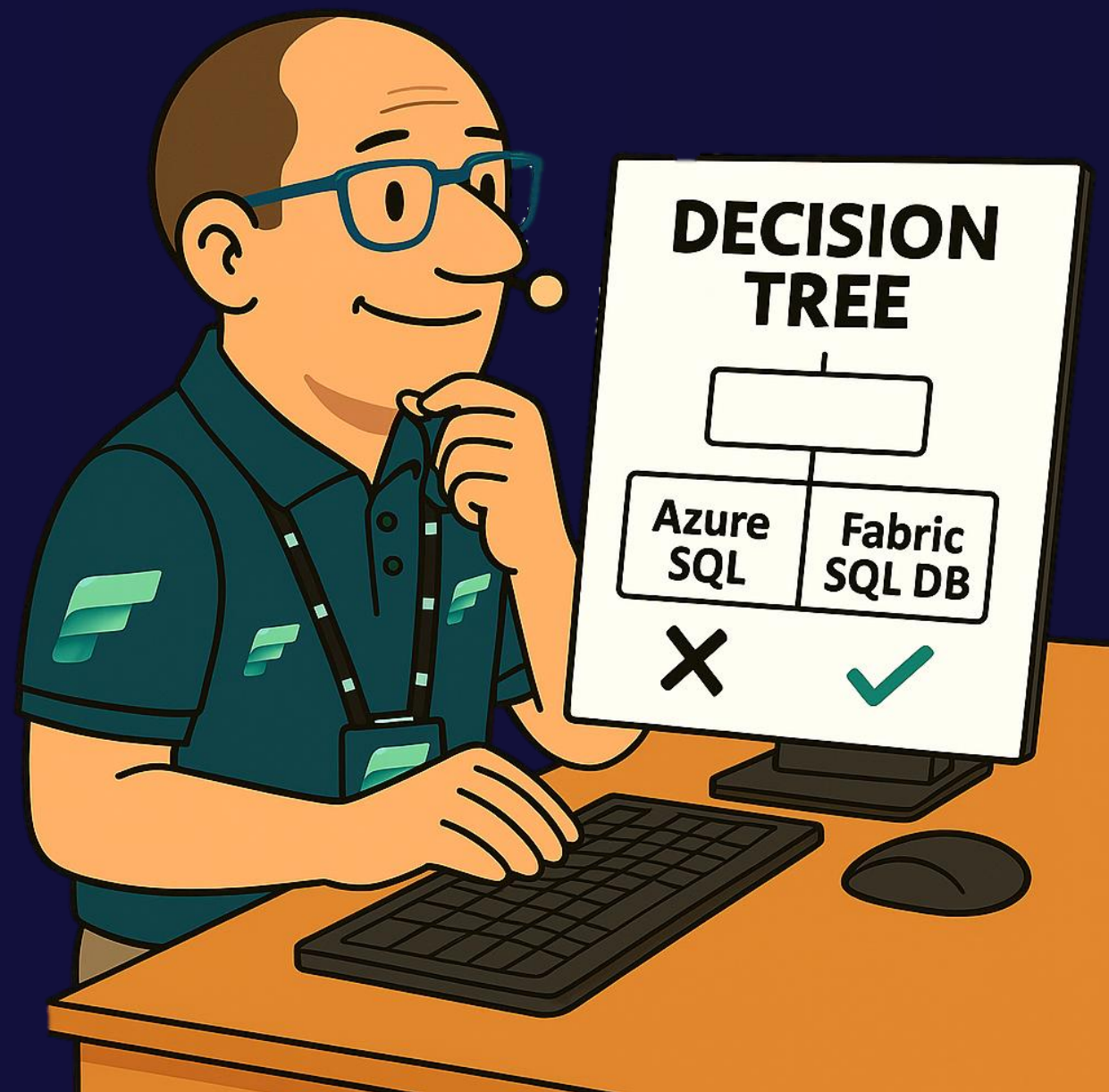
- 1 month = 730 hours
- Datacenter: US EAST, ZRS, Gen 5
- DB Size: 100GB, 30GB Log, 100GB Backup Size
- 12.256 vCores F32 approx. = 12 vCores Azure SQL

n

Service	Fabric SQL	Azure SQL	Azure SQL	Azure SQL	
SKU	F32	General Purpose Provisioned	Business Critical Provisioned	Hyperscale Standard Series + 1 Replica	
\$/month	\$ 4.204,80	\$ 3.009,17	\$ 5.951,85	\$ 3.200,20	PAYG
\$/hour	\$ 5,76	\$ 4,12	\$ 8,15	\$ 4,38	
\$/month	\$ 2.501,33	\$ 2.262,70	\$ 5.018,00	\$ 2.080,00	OneYear Reserved
\$/hour	\$ 3,43	\$ 3,10	\$ 6,87	\$ 2,85	
Storage	\$ 5,20	\$ 29,90	\$ 32,50	\$ 25,00	

Source: [Billing and Utilization Reporting - Microsoft Fabric | Microsoft Learn](#)

06 - Decision framework



The Choice is Yours!

✓ Use Azure SQL if:

- Your workload is very small, highly variable or needs massive throughput
- You need extensive DR/Backup retention
- You require features not (yet) in Fabric
- You need standalone database access patterns

✓ Use SQL Database in Fabric if:

- Your app's value is tied to real-time insights and you want to avoid complex ETL
- Your app could leverage the “new stuff” (GraphQL, AI, etc.)
- You need to serve both end-user applications and analytical workloads from the same data
- You are building a new solution on Microsoft Fabric anyway



**HERZLICHEN
DANK!**

Sound off.
The mic is all yours.
Influence the product roadmap.

Join the Fabric User Panel



Share your feedback directly with our
Fabric product group and researchers.

<https://aka.ms/JoinFabricUserPanel>

Join the SQL User Panel



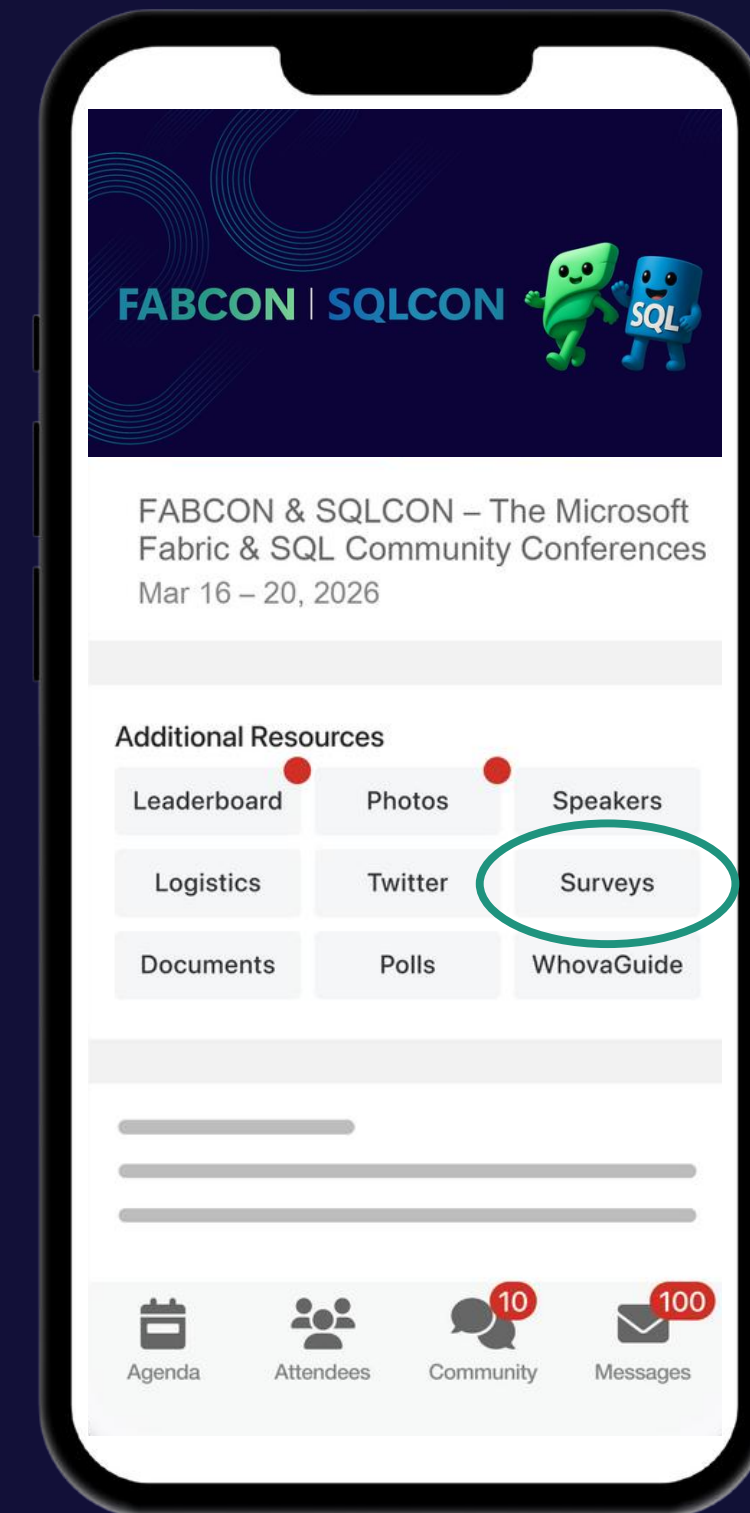
Influence our SQL roadmap and ensure
it meets your real-life needs

<https://aka.ms/JoinSQLUserPanel>

How was the session?



Complete Session Surveys in
Ukova for your chance to WIN
PRIZES!



Get Two Fabric Certifications for FREE

Attendees of FABCON can take the Fabric Analytics Engineer or Fabric Data Engineer exam for free. Be part of the 2 fastest growing role-based certifications in Microsoft history.

Request your voucher by March 23, 2026.

<https://aka.ms/fabcon/cert100>

