

Reasons to Modernize Data Workloads to Azure

Speaker Name



Reasons to Modernize



Scale seamlessly with growing demands



Enhanced **Security** and **Compliance**



Automated **High Availability & Disaster Recovery**



Integrated **AI capabilities**

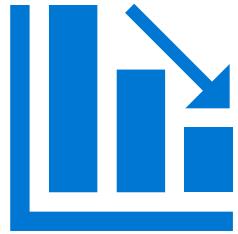
Scale on Demand



Vertical Scaling



Upgrade to a Higher Service Tier



Downgrade to a Lower Service Tier



Change the Performance Level

Changing Performance Levels (vCore)

PowerShell

- Set-AzSqlDatabase

REST

- Update database

Azure CLI

- az sql db update

T-SQL

- ALTER DATABASE ... MODIFY (EDITION = ...)

Service and compute tier

Select from the available tiers based on the needs of your workload. The vCore model provides a wide range of configuration controls and offers Hyperscale and Serverless to automatically scale your database based on your workload needs. Alternately, the DTU model provides set price/performance packages to choose from for easy configuration. [Learn more](#)

Service tier

General Purpose (Scalable compute and storage options) 

vCore-based purchasing model

General Purpose (Scalable compute and storage options)

Hyperscale (On-demand scalable storage)

Business Critical (High transaction rate and high resiliency)

Compute tier

Compute Hardware

Select the hardware configuration based on confidential computing hardware depends on the service tier.

Hardware Configuration

DTU-based purchasing model

Basic (For less demanding workloads)

Standard (For workloads with typical performance requirements)

Premium (For IO-intensive workloads)

up to 80 vCores, up to 408 GB memory

[Change configuration](#)

Save money

Already have a SQL Server License? Save with a license you already own with Azure Hybrid Benefit. Actual savings may vary based on region and performance tier. [Learn more](#)

Yes No

vCores [Compare vCore options](#)

2

Data max size (GB) 

32

Changing Compute Tier and Hardware (vCore)

Service and compute tier

Select from the available tiers based on the needs of your workload. The vCore model provides a wide range of configuration controls and offers Hyperscale and Serverless to automatically scale your database based on your workload needs. Alternately, the DTU model provides set price/performance packages to choose from for easy configuration. [Learn more](#)

Service tier

General Purpose (Scalable compute and storage options)

[Compare service tiers](#)

Compute tier

- Provisioned** - Compute resources are pre-allocated. Billed per hour based on vCores configured.
- Serverless** - Compute resources are auto-scaled. Billed per second based on vCores used.

Compute Hardware

Select the hardware configuration based on your workload requirements. Availability of compute optimized, memory optimized, and confidential computing hardware depends on the region, service tier, and compute tier.

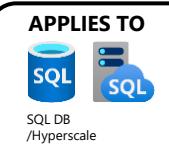
Hardware Configuration

Gen5

up to 80 vCores, up to 408 GB memory

[Change configuration](#)

Impact of Database Changes



Connections to the database may be temporarily dropped

The duration depends on both size and service tier

No data is lost during this process

Changing to, from, within Standard Tier of 250 GB database < 6 hours

Changing within Premium Tier of 250 GB database < 3 hours

Impact of Database Changes (continued)

Downgrading the service tier can cause a change in the backup retention periods.

Changing your database pricing tier does not automatically change the max database size.

Enterprise Security



Enterprise Ready



Security

Identity

Encryption

Networking



High Availability

Availability Zones

RPO = 0

Automatic Failovers



Disaster Recovery

Backups

Geo-Redundancy

Long Term Retention

Enterprise Security

Azure Database for PostgreSQL is **the only Postgres offering** with support for Entra Id Authentication

Service or Customer Managed encryption keys to **protect data at rest**

Secure network connectivity via Private Endpoints



Entra Id

Enterprise Identity built in to Azure SQL, PostgreSQL, Cosmos DB offerings



Customer Managed Keys

Key Vault integration with optional HSM support



Private Endpoints

Network interface that uses a private IP address from your virtual network

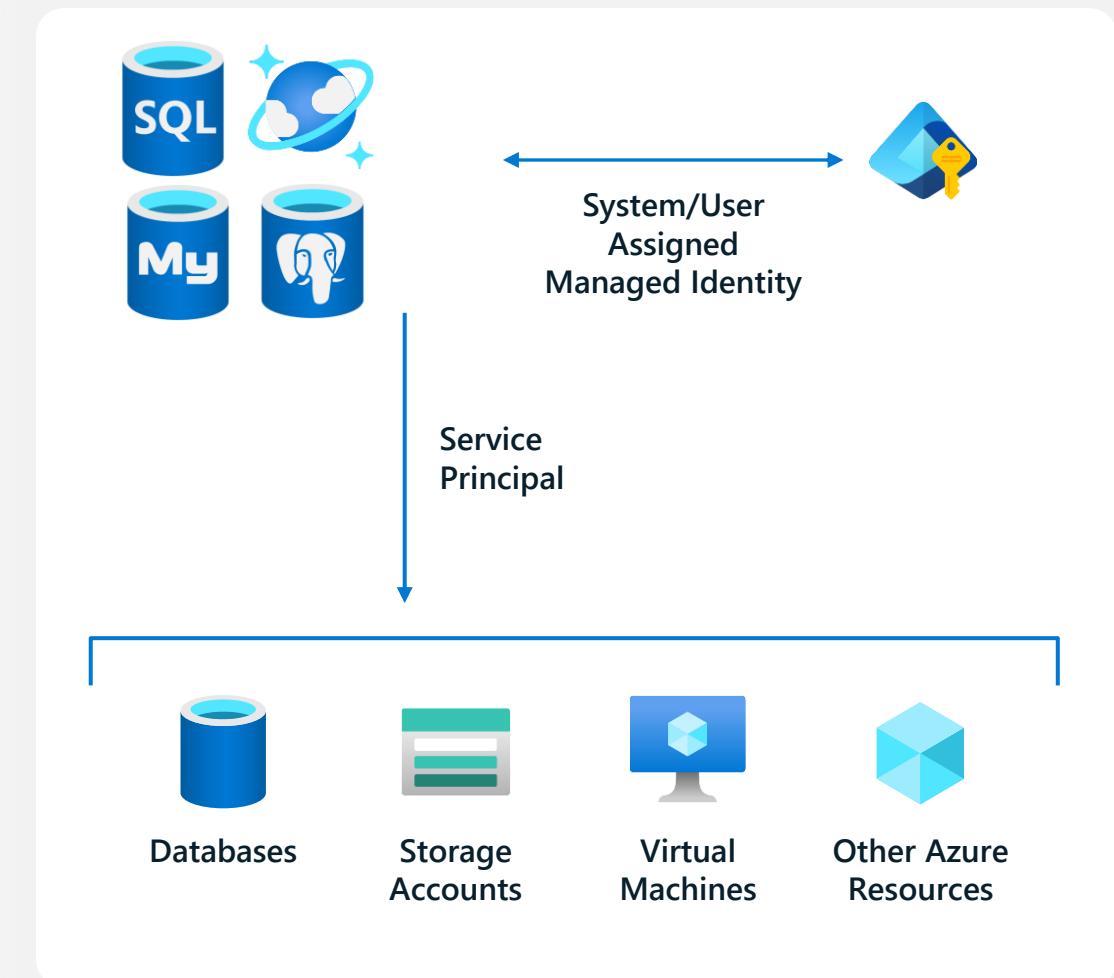
Enterprise Security

Enterprise Identity

Use Managed Identities to **connect to any Azure resource** that supports Entra ID

Eliminates the need to store and manage secrets within Postgres service

Support for both **User and System Assigned** Managed Identities



Enterprise Security

Customer Managed Encryption Keys

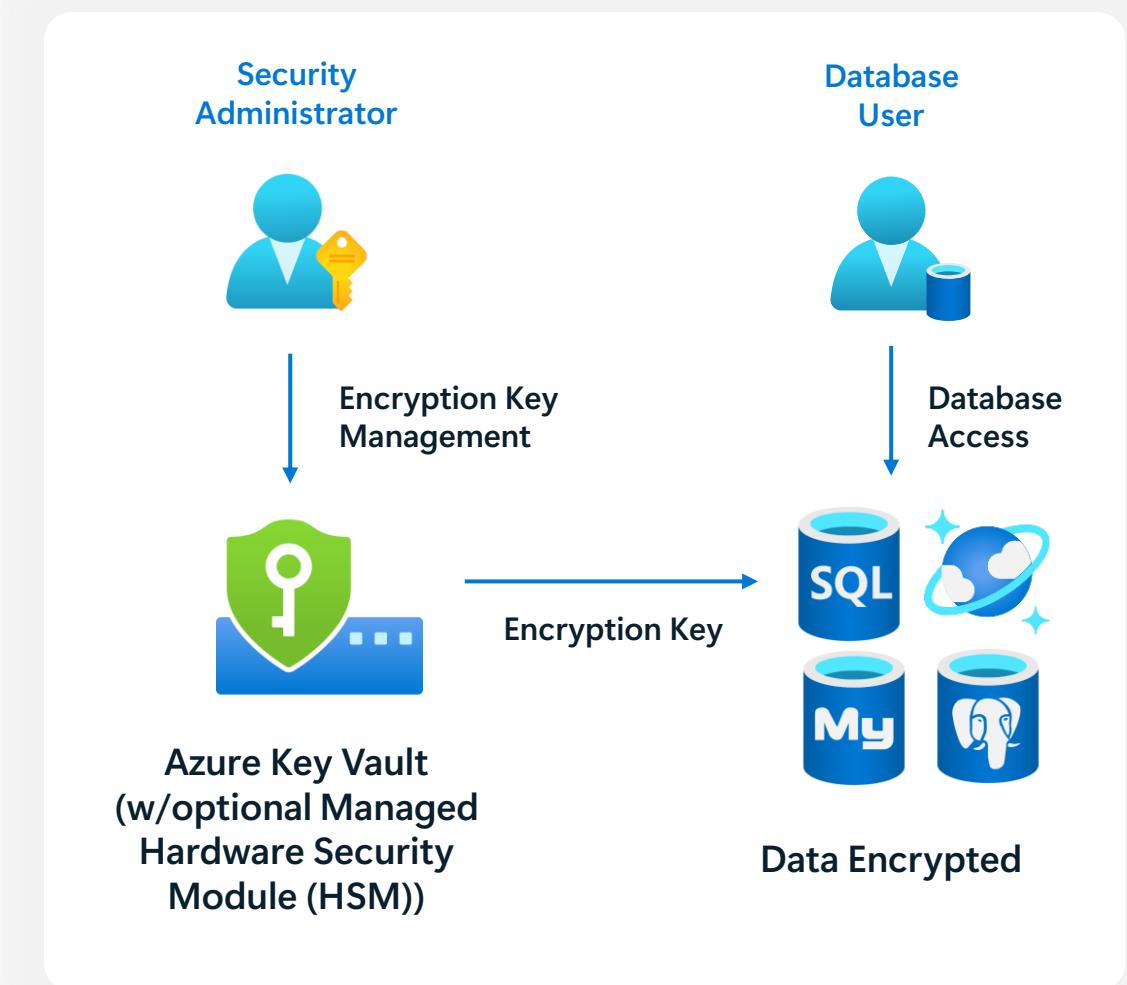
Flexible Server supports **Customer Managed Encryption Keys** for securing data

Rotate and revoke keys for as needed

Meet regulatory and compliance requirements by managing **key lifecycle and access** policies

Provides **separation of duties** to key management from data management

Centralized management of keys in Azure Key Vault or Managed HSM



Enterprise Security

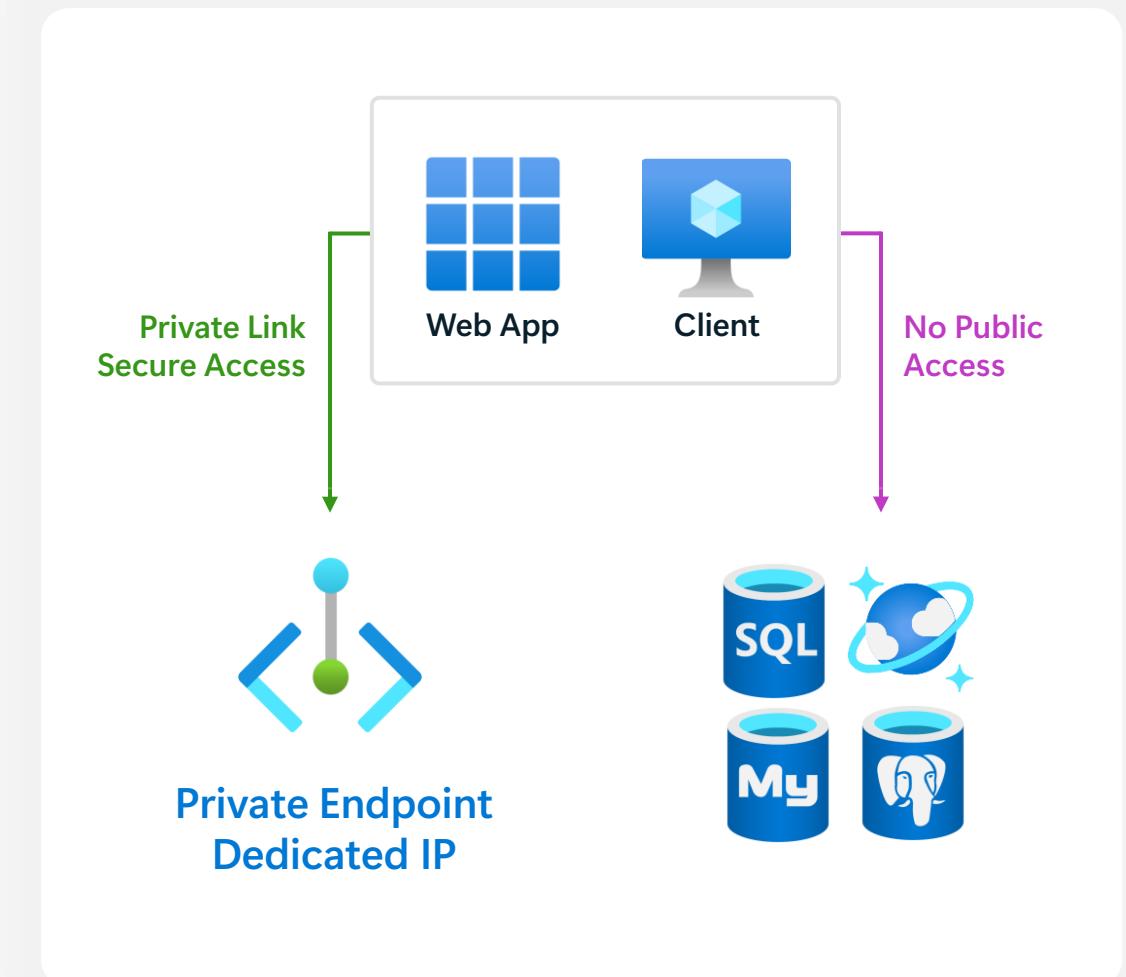
Private Endpoints

Provides a **private IP address** from your virtual network for access.

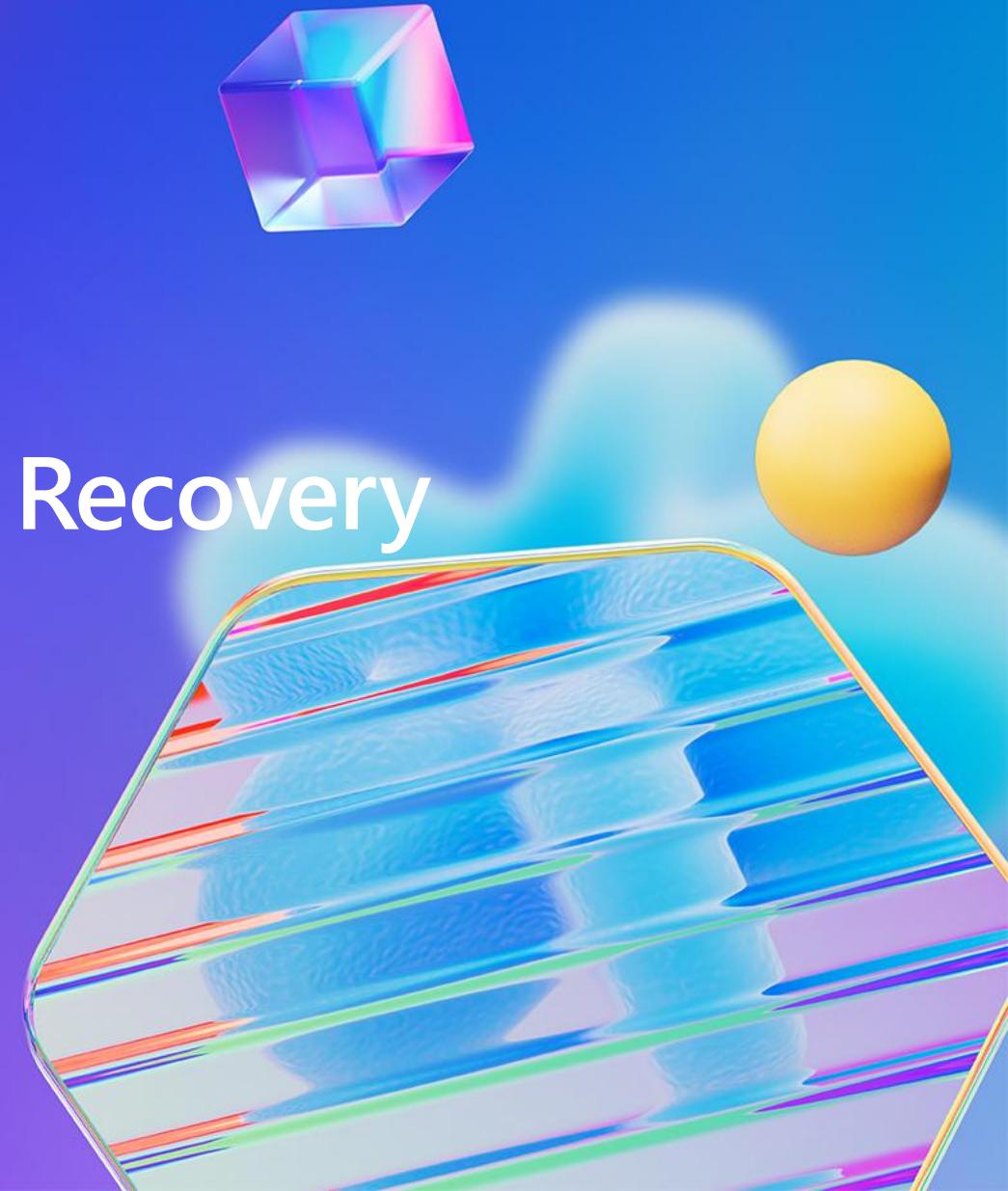
Prevents **data exfiltration** from your virtual network

Improved performance using a direct route to Azure resources vs. connecting via Internet

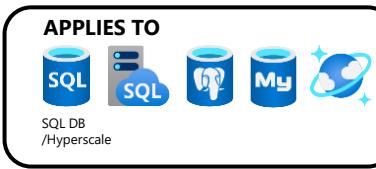
Fine-grained access control using **network security groups (NSG's)** to restrict access to the private endpoint



High Availability and Disaster Recovery



High Availability & Disaster Recovery features



| |  Azure SQL DB |  Azure SQL DB Hyperscale |  Azure SQL Managed Instance |  Azure Cosmos DB |  Azure Database for PostgreSQL |  Azure MySQL |
|--------------------|--|---|--|---|---|---|
| High Availability | Built-in 99.995% | Built-in 99.95% | Built-in 99.99% | Built-In 99.999% | Built-In 99.99% | Built-In 99.99% |
| Disaster Recovery | Built-in (including Read Replica/Geo Replica) | Built-in (including Read Replica/Geo Replica) | Built-in (including Read Replica / Geo Replica) | Built-In | Built-In (including Read Replica/Geo Replica) | Built-In (including Read Replica/Geo Replica) |
| Multi-region Write | Multi-region failover read/write only | Multi-region failover read/write only | Multi-region failover read/write only | Built-In | Multi-region failover read/write only | Multi-region failover read/write only |
| Scale-out Read | Read Replica | Read Replica | Read Replica | Built-In | Read Replica | Read-Replica |
| CDC / Replication | Supported | supported | Supported | Supported | Supported (logical decoding / PG_logical) | Supported (Debezium, Kafka, Event Hubs, Binlog Replication) |

Availability vs High Availability vs Disaster Recovery

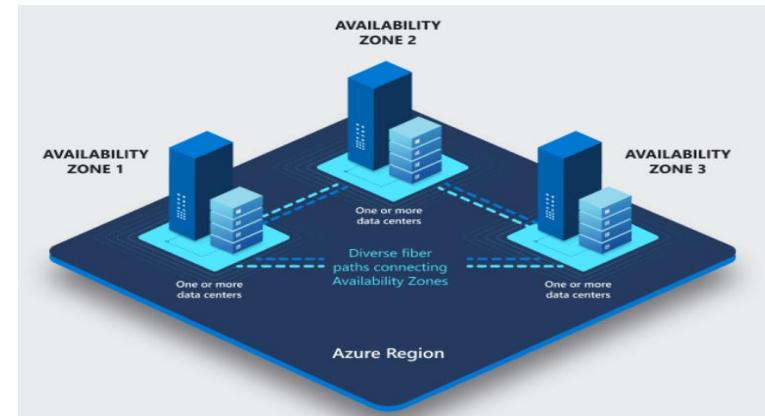


Availability

- Every database comes with core resiliency and reliability, that protects it against software or hardware failures.
- Automated backups protect against data deletion or corruption.
- Industry leading SLA across Azure Database Offerings

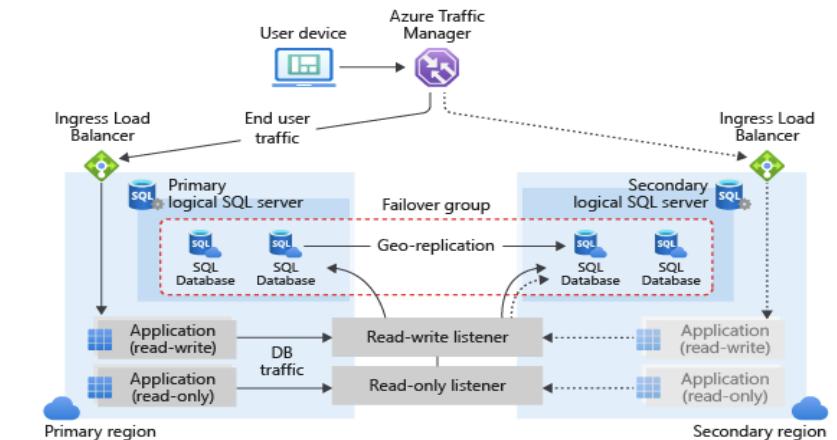
High Availability (Zone Redundancy) (Customer configured)

- Database is resilient to zonal failures.
- Recovery is
 - Automatic
 - Quick
 - Transparent
- Higher SLA



Disaster Recovery (Customer configured)

- Ability to quickly recover the database from a catastrophic regional failure to provide business continuity.
- Options:
 - Failover Groups*
 - Active Geo Replication
 - Geo Restore



*No additional price for Premium & Business Critical high availability

High Availability - Zone Redundant Configuration



Replicas are automatically created across Zones

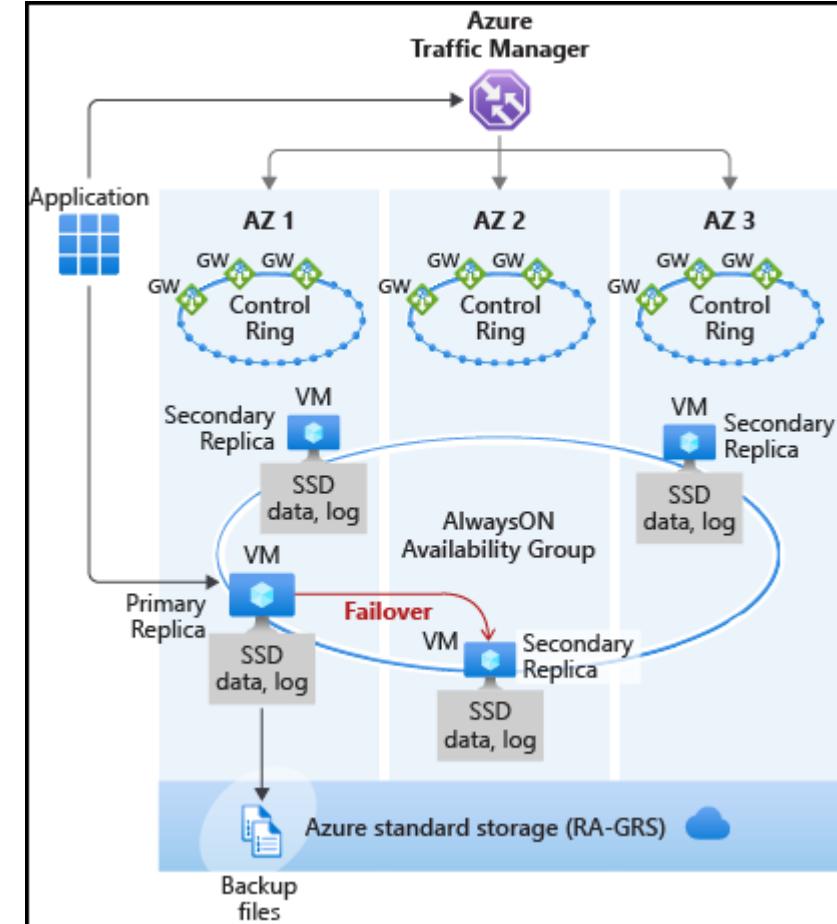
Ability to place different replicas of the database to different availability zones in the same region

The routing is automatically controlled by Azure Traffic Manager (ATM)

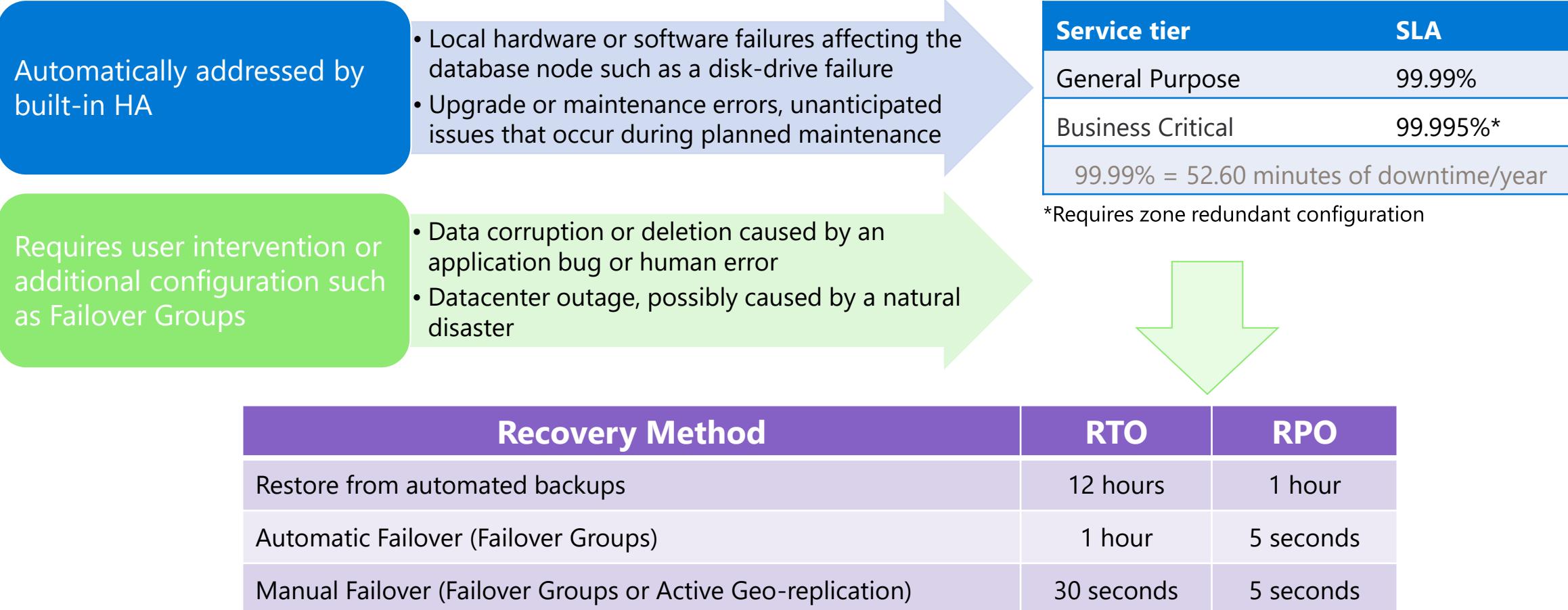
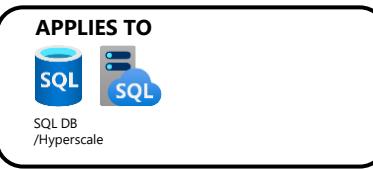
Increased network latency may impact the performance

Zone redundant configurations are GA for BC tier (both) and SQL DB GP tier, preview for MI GP tier

Synchronous replication



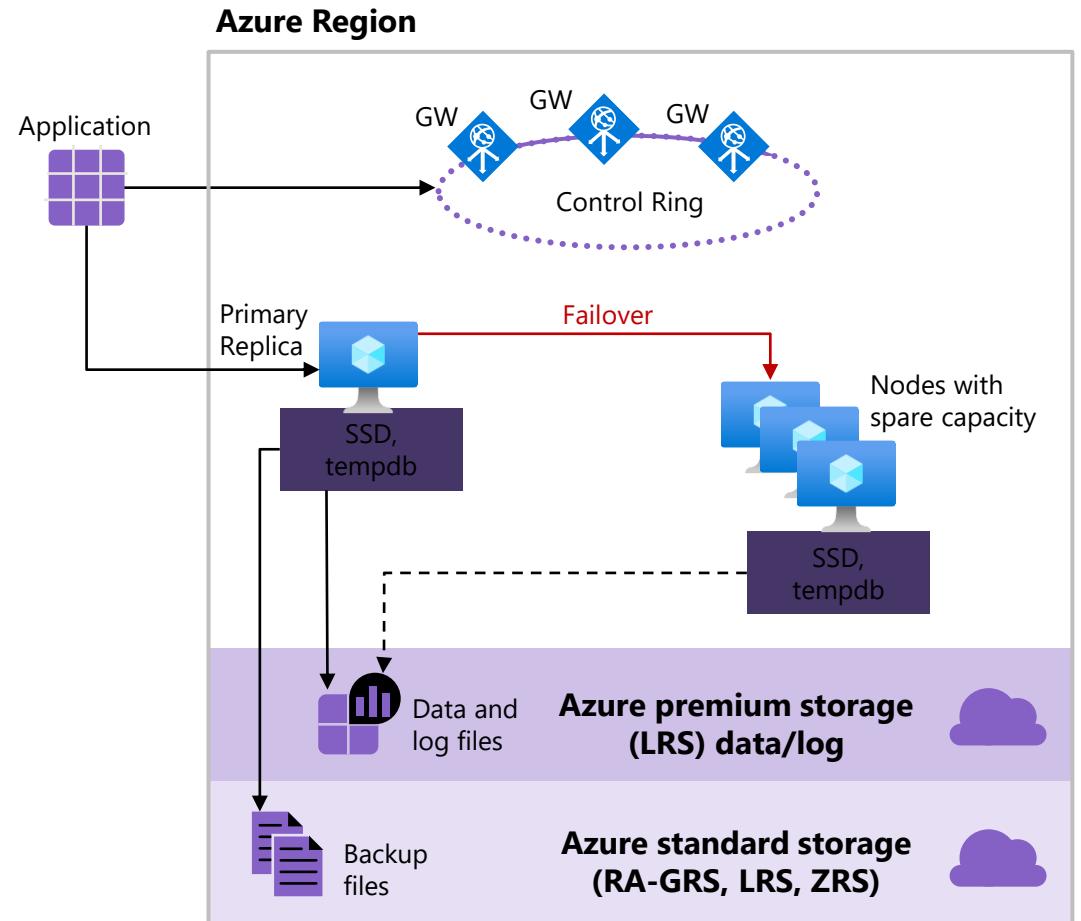
High Availability (HA) in Azure SQL PaaS



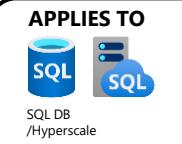
Azure SQL High Availability in Basic, Standard, and General-Purpose Service Tiers



- Behaves like SQL server failover cluster
- Remote storage provides data redundancy within a datacenter
- Backup files are in a different location with geo-redundancy
- Failover decisions based on SQL and Service Fabric
- Recovery time depends on spare capacity
- Connectivity redirection built-in
- Zone redundancy is in public preview
- 99.99% SLA



High Availability in Premium and Business Critical Service Tier SQL DB



Behaves like Always On Availability Groups

3 secondary replicas automatically created

At least one secondary must sync for commits

System database replicated

Backup files in a different location with geo-redundancy

Automatic failover based on SQL and Service Fabric

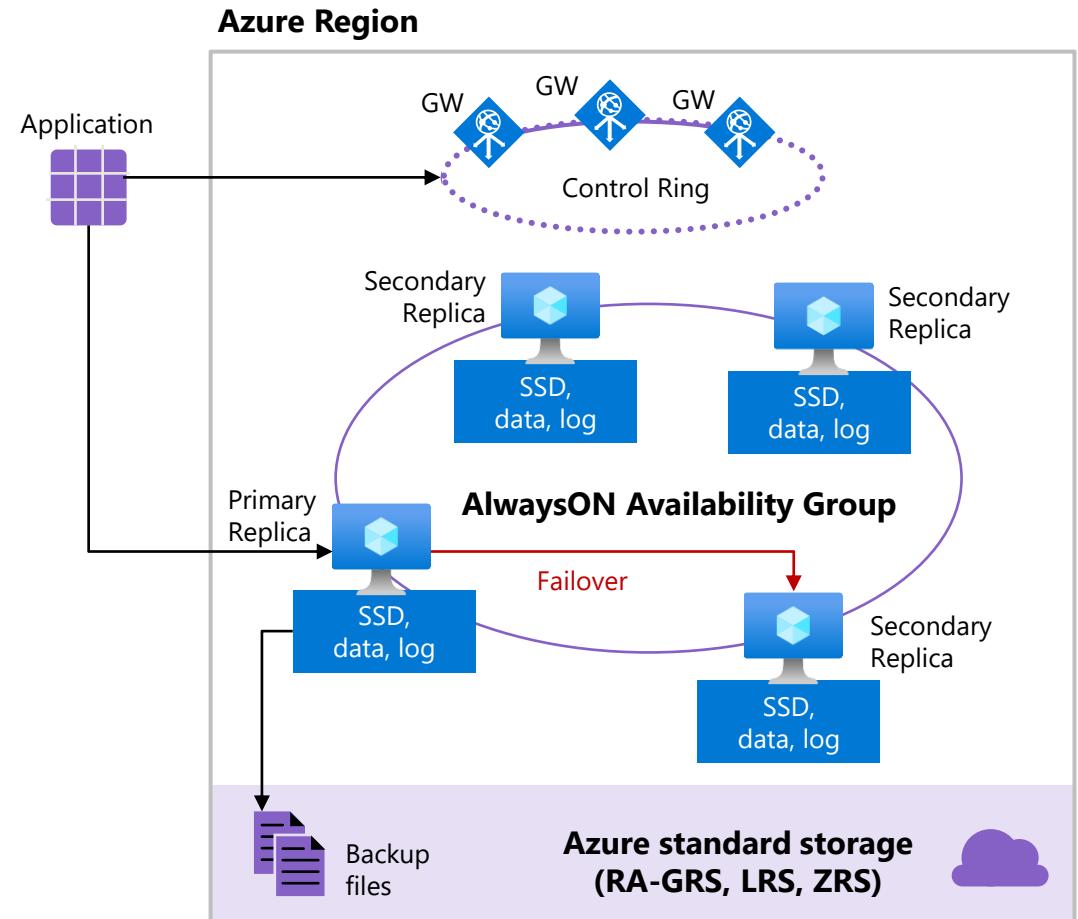
Recovery time extremely fast

Connectivity redirection built-in

Read Scale-Out from one of the replicas

Zone redundant option is GA

99.99% SLA in single zone



High Availability Considerations in Managed Instance

APPLIES TO
SQL DB /Hyperscale

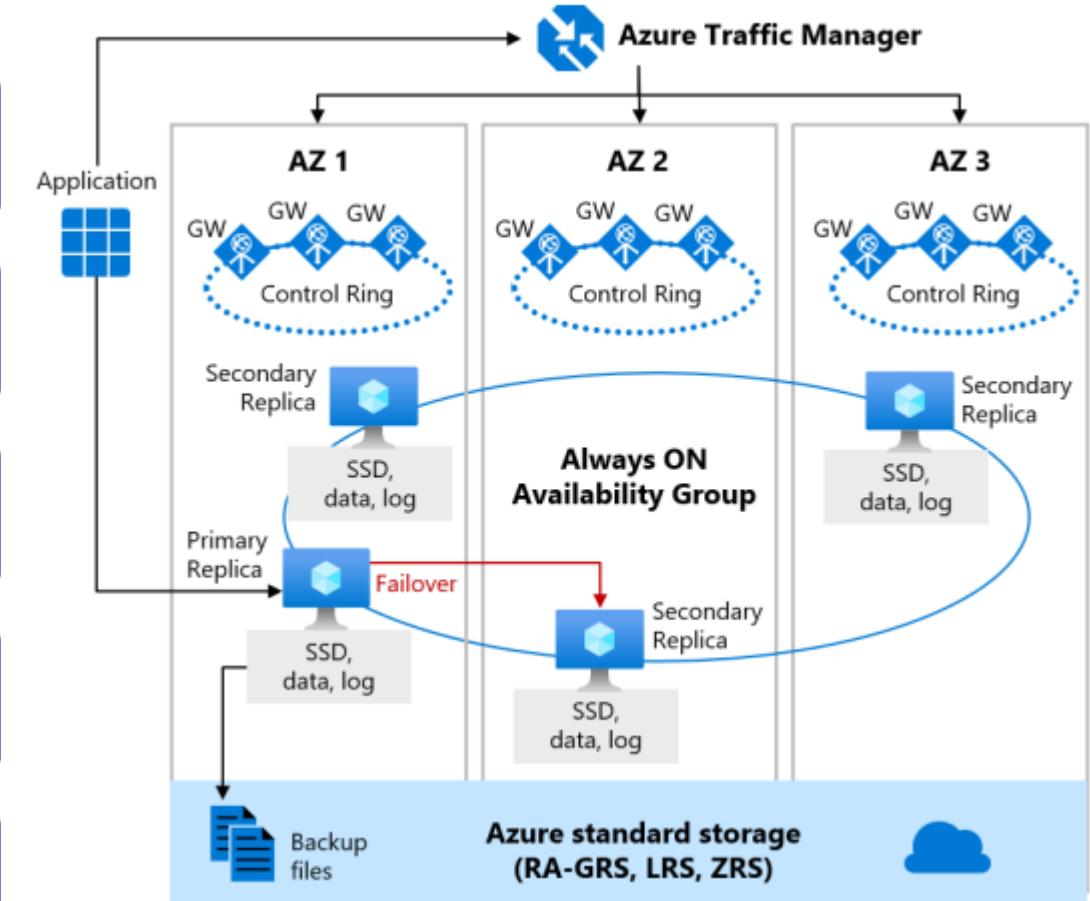
In Business-Critical tier, there is 1 readable and at least 1 non-readable replica

Log Shipping and Database Mirroring are not supported to/from MI

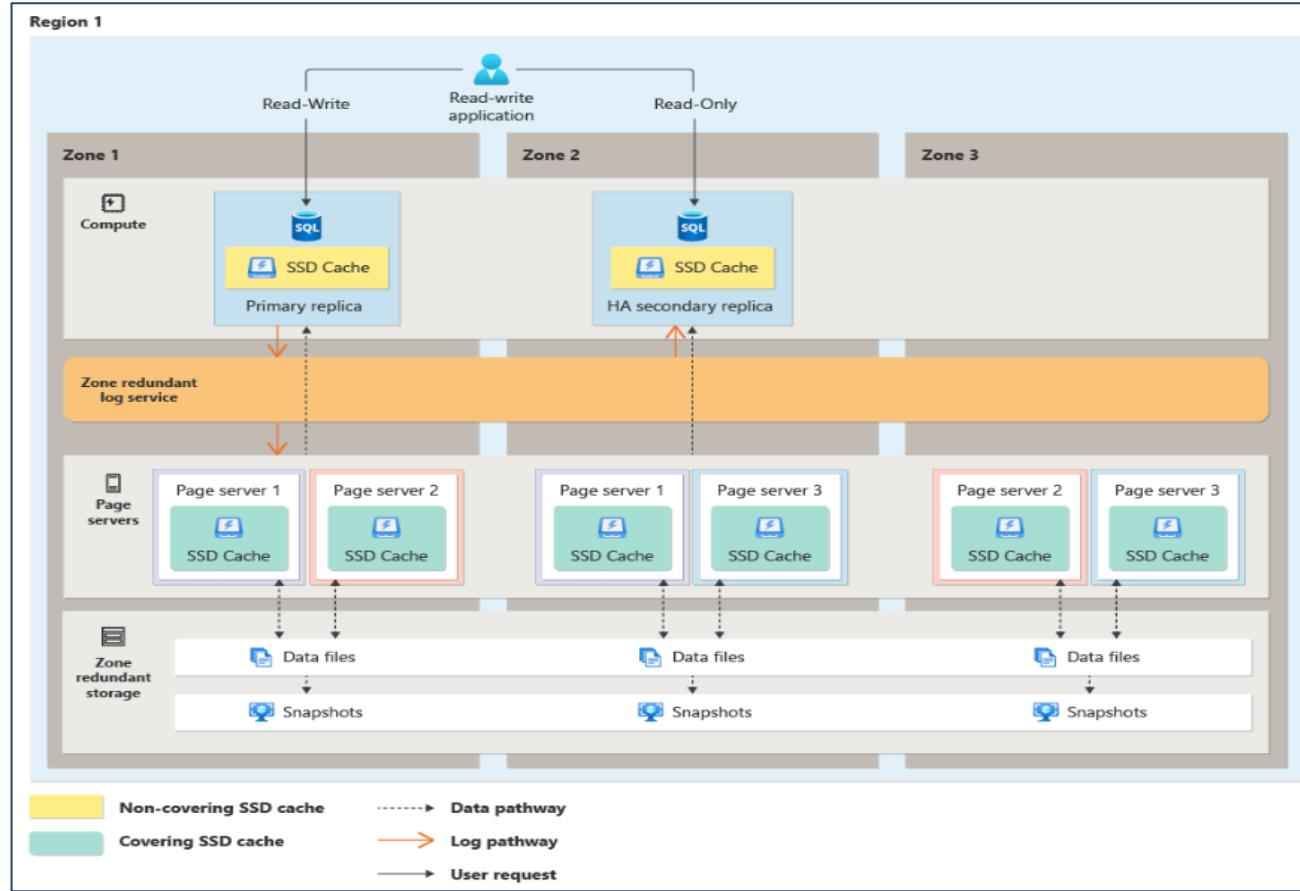
Replication is supported between databases in SQL Managed Instance, SQL Database, or SQL Server

New link feature - connects your SQL Servers hosted anywhere to SQL Managed Instance

Zone redundancy for SQL Managed Instance is GA in Business Critical tier, preview in General Purpose tier



High Availability - Hyperscale Service Tier



Data resiliency is provided at the storage level

When the compute replica is down, a new replica is created automatically with no data loss

If there's only one compute replica, it may take some time to build the local cache in the new replica after failover

Log service is zone redundant.

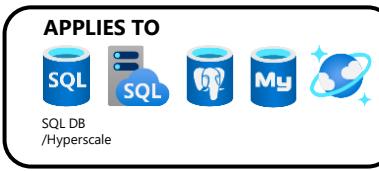
During the cache rebuild phase, the database fetches data directly from the Page servers, resulting in higher storage latency and degraded query performance

99.995% uptime SLA

All layers have zone resiliency.

For mission-critical apps that require high availability with minimal failover impact, you should **provision at least 2 compute replicas** including the primary compute replica. That way there is a **hot-standby** replica available that serves as a failover target

Active Geo-Replication Capabilities



Asynchronous
Replication

Readable secondary
databases

Multiple Replicas

Configurable
performance level of
secondary database

User-controlled failover
and fallback

Must manually keep
credentials and firewall
rules in sync

Failover Group

Incidents that cannot be self-mitigated by the Azure service fabric built-in automatic high availability operations

Provides read-write and read-only **listener** end-points that remain unchanged during failovers

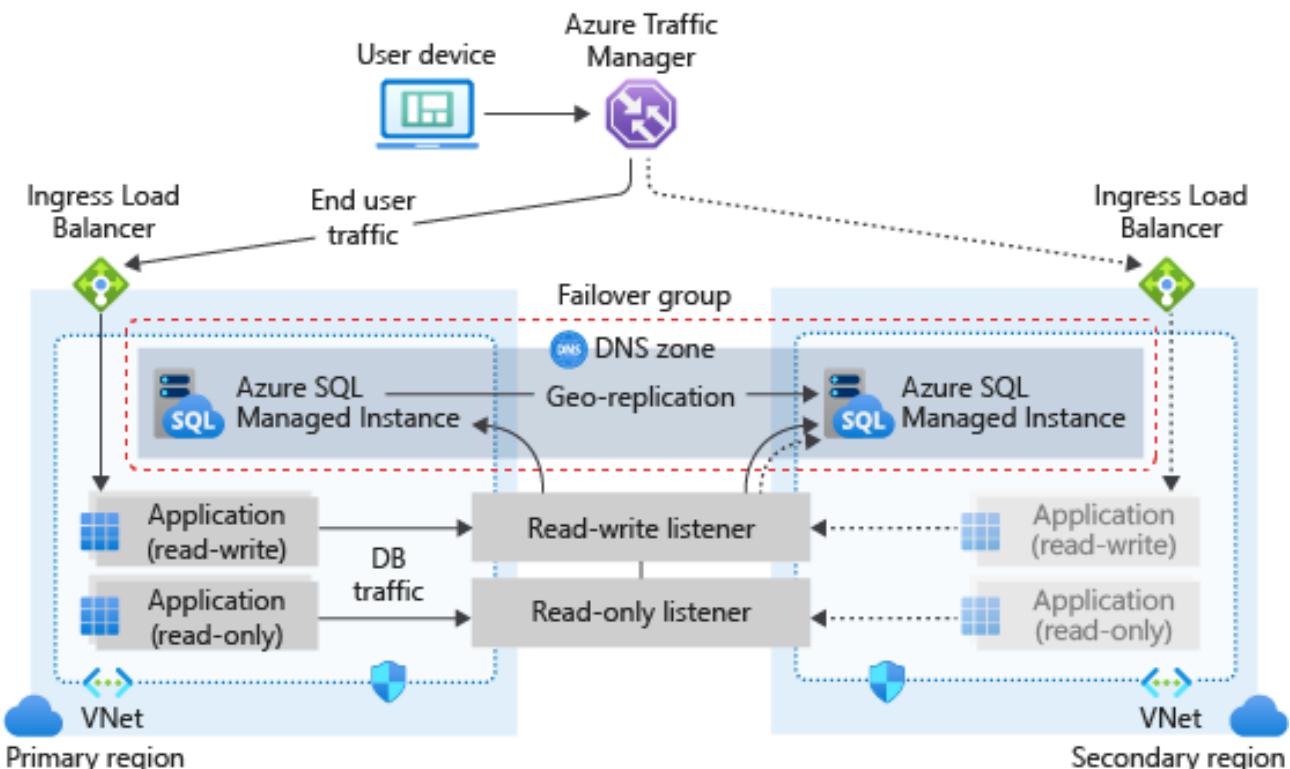
For Managed Instances

- A Failover Group contains all user databases in the instance and therefore only one Failover Group can be configured on a Managed Instance
- Both the primary and secondary instances must be in the same DNS zone
- [Deploy both managed instances to paired regions for performance reasons](#)
- System databases are not replicated
- Manually synchronize instance properties and retention policies instances

Failover Group architecture

The auto-failover group must be configured on the primary instance and will connect it to the secondary instance in a different Azure region. All user databases in the instance will be replicated to the secondary instance.

System databases like master and msdb will not be replicated.



Failover Group

How to prevent data loss

- Failover Groups use an asynchronous replication mechanism, making the possibility of data loss unavoidable
- Call the **sp_wait_for_database_copy_sync** stored procedure immediately after committing the transaction to block the calling thread until the transaction is hardened on the Secondary replica

Initial Seeding

- Possible seeding speed is up to 360 GB an hour for MI and 500 GB an hour for SQL Database

Limitations

- Failover groups cannot be created between two servers/instances in the same Azure region
- Failover groups cannot be renamed
- Database rename is not supported for databases in failover group

Failover Group

Read-Write Listener - <fog-name>.<zone_id>.database.windows.net

Read-Only Listener - <fog-name>.secondary.<zone_id>.database.windows.net

The Read-Write listener and Read-Only listener cannot be reached via the public endpoint for MI

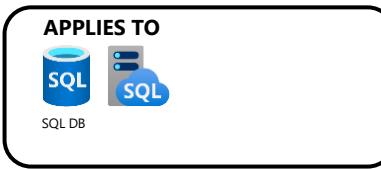
Connection redirection with **ApplicationIntent=ReadOnly** connection string property

- To connect to the built-in read-only replica in the primary location, use ApplicationIntent=ReadOnly and <fog-name>.<zone_id>.database.windows.net
- To connect to a read-only replica in the secondary location, use ApplicationIntent=ReadOnly and <fog-name>.secondary.<zone_id>.database.windows.net

Microsoft managed failover policy

- Azure Service fabric detects failure that cannot be mitigated by the built-in high availability infrastructure due to the scale of the impact and initiates the failover
- With **GracePeriodWithDataLossHours**, you can control how long the system waits before initiating the failover that is likely to result in data loss
 - Failover only occurs after minimum grace period has expired – **minimum 1 hour**
- You can choose customer managed failover to have more control on the failover to the **secondary** site
- Read Only Listener disabled after failover by default to avoid performance impact on the new Primary, but you can change it by configuring the **AllowReadOnlyFailoverToPrimary** property

Failover policy in failover groups



Customer managed (Recommended)

- On rare occasions, built in availability or high availability (zone redundancy) isn't enough for business continuity.
- In these situations, customers can perform forced failover during unexpected outage.
- This policy puts customer in control of restoring business continuity for their application.
- In command line tools such as PowerShell or CLI and Rest API, failover policy value for customer managed is *manual*.

Microsoft managed

- In the event of a widespread outage that impacts a primary region, Microsoft initiates forced failover of all failover groups.
- For Microsoft to initiate forced failovers,
 - Many databases in the region must be impacted.
 - Grace period of at least one hour is expired as humans are involved to verify scale of outage.
- Microsoft managed failover won't be initiated for individual failover groups or a subset of failover groups in a region.
- In command line tools such as PowerShell or CLI and Rest API, failover policy value for microsoft managed is *automatic*.

Active geo-replication vs auto-failover groups

| | Active Geo-replication | Failover groups |
|--|------------------------|-----------------|
| Continuous data synchronization between primary and secondary | Yes | Yes |
| Fail over multiple databases simultaneously | No | Yes |
| Connection string remains unchanged after failover | No | Yes |
| Supports read-scale | Yes | Yes |
| Multiple replicas | Yes | No |
| Can be in same region as primary | Yes | No |

Disaster Recovery

Backup protection for major disruptions

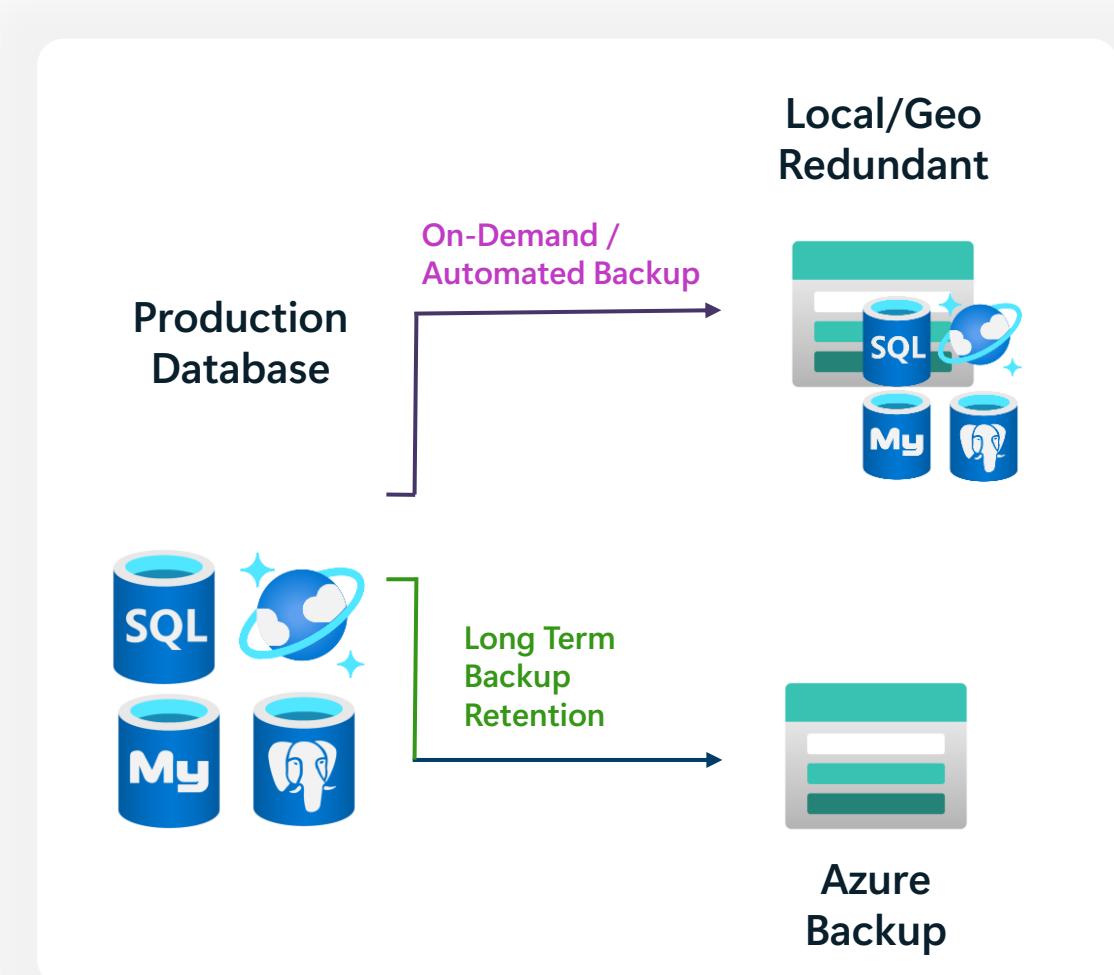


On-Demand or Automated backups protect data from loss with 30 Day retention

Configure backups for **geo-redundant storage**, and restore to different regions in event of failure

Long-term backup retention for **up to 10 years** supports compliance requirements for data retention

RPO of 5 minutes



Disaster Recovery - Geo-Redundant Backups

Replicated to paired region

Can be restored to any region in the same subscription

Creates a new database on a Managed Instance

RPO up to 1 hour

RTO usually less than 12 hours, depending on size and activity

Link Feature for Managed Instance

APPLIES TO

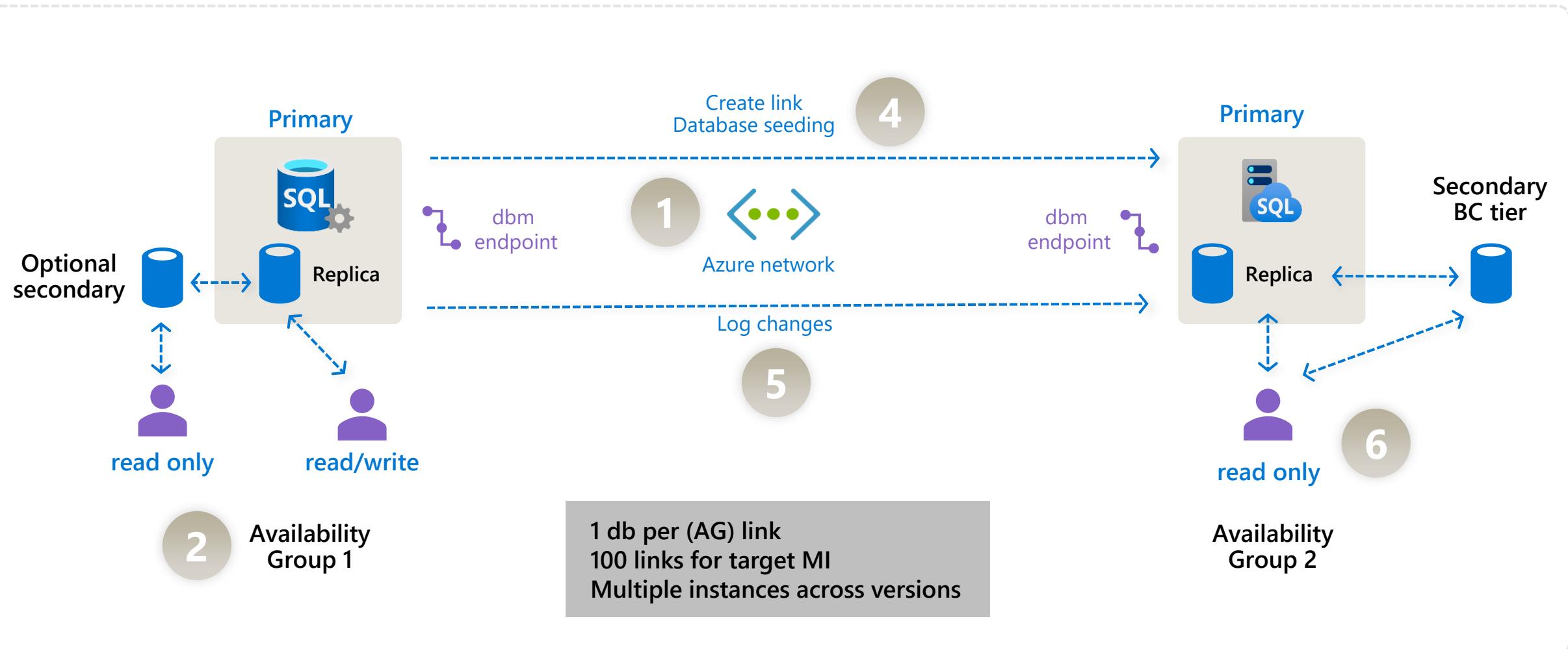


Applies to: Managed Instance

Distributed Availability Group (DAG) async

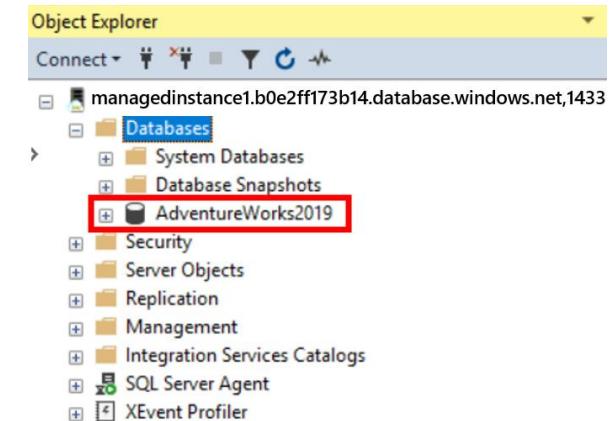
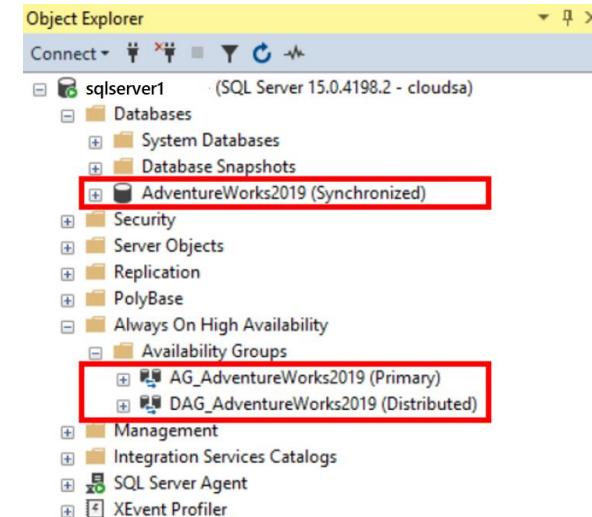
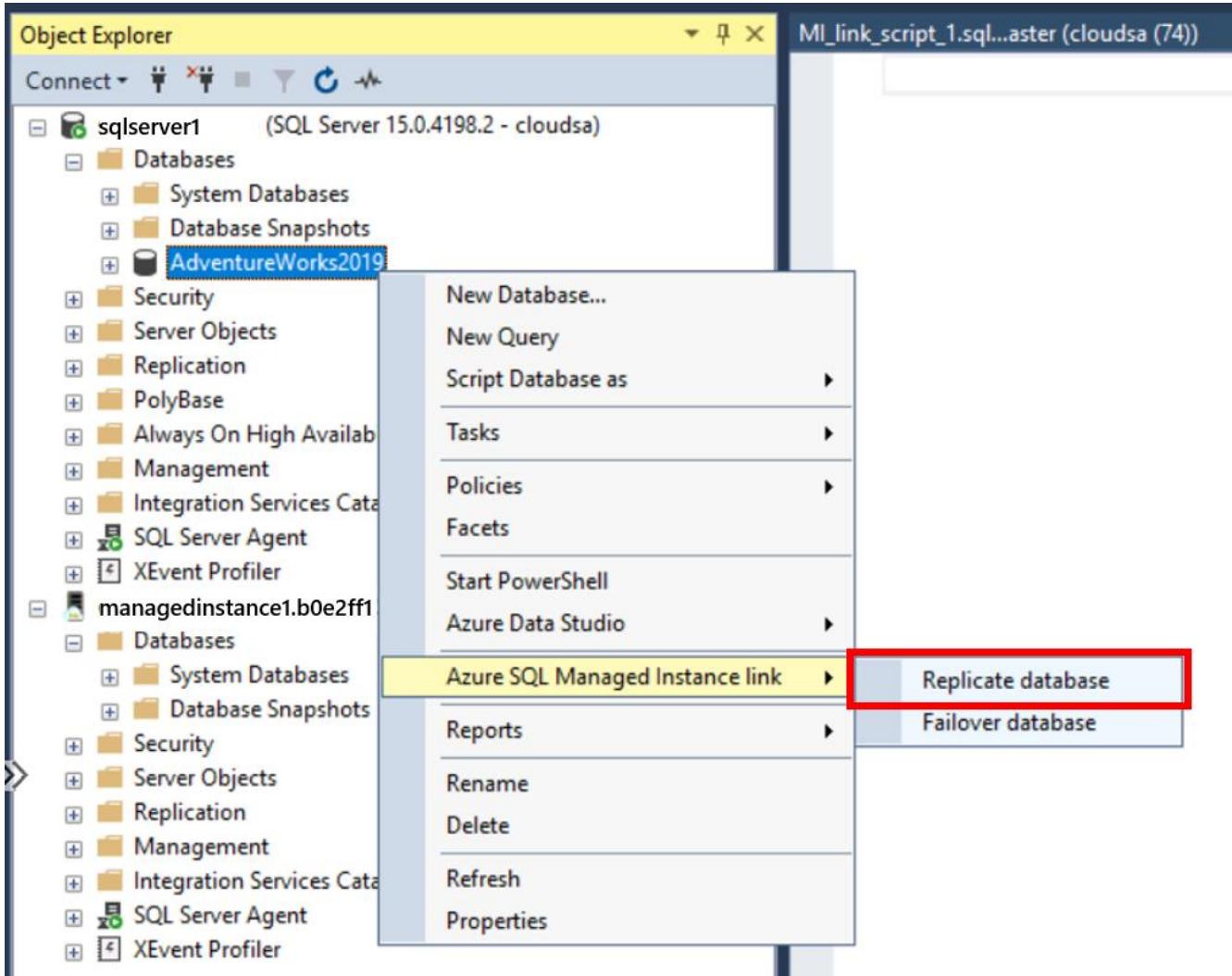
3

Single step with SSMS
using the replicate wizard



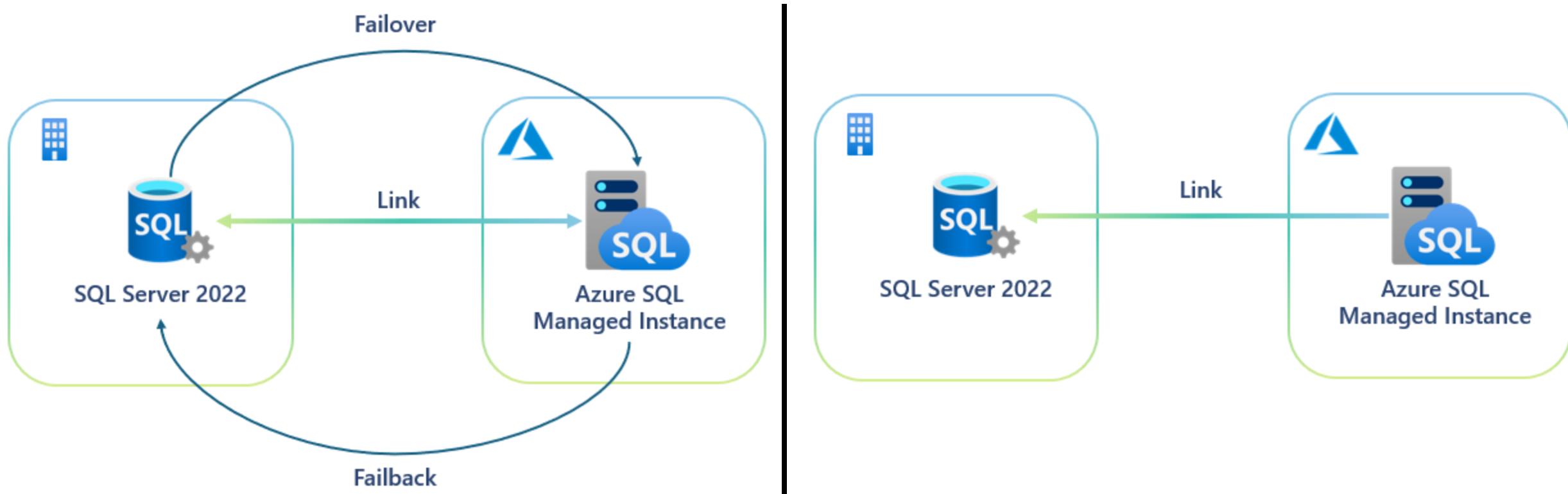
Managed Instance Link

APPLIES TO





Managed Instance Link Feature from Azure SQL Managed Instance to SQL Server 2022



Bi-directional failover between SQL Server 2022 and SQL Managed Instance

Creating the link from Azure SQL Managed Instance to SQL Server 2022

- [Create a resource](#)
- [Home](#)
- [Dashboard](#)
- [All services](#)
- [★ FAVORITES](#)
- [Resource groups](#)
- [All resources](#)
- [Recent](#)
- [Azure SQL](#)
- [App Services](#)
- [SQL databases](#)
- [Virtual machines \(classic\)](#)
- [Virtual machines](#)
- [Cloud services \(classic\)](#)
- [Subscriptions](#)
- [Azure Active Directory](#)
- [Monitor](#)
- [Help + support](#)
- [Advisor](#)
- [Cost Management + Billing](#)

sqlmidr

SQL managed instance

[New database](#) [Reset password](#) [Delete](#) [Feedback](#)

Your managed instance is ready. Click here to learn about next steps. →

[View Cost](#) | [JSON View](#)

Essentials

Resource group : [sqlmidrrg](#)

Managed instance admin : sqladmin

Status : Online

Host : sqlmidr.426638509636.database.windows.net

Location : East US

Pricing tier : Business Critical Standard-series (Gen 5) (256 GB, 8 vCores, Geo...

Subscription : [AzureSQL bobward](#)Instance pool : [Not in an instance pool](#)

Subscription ID : 0efc44aa-c965-420f-aac4-fff305dbcc97

Virtual network / subnet : [vnet-sqlmidr/ManagedInstances](#)**Tags (edit)** : SQL2022CompatibilityOn : trueVirtual cluster : [Not available](#)

0 managed instance databases

| Name | Status | Creation time (UTC) | Earliest restore time (UTC) |
|------|--------|---------------------|-----------------------------|
|------|--------|---------------------|-----------------------------|

No databases found

CPU utilization

Show data for last:

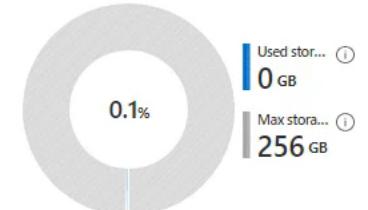
[1 hour](#) [24 hours](#) [7 days](#)

Aggregation type:

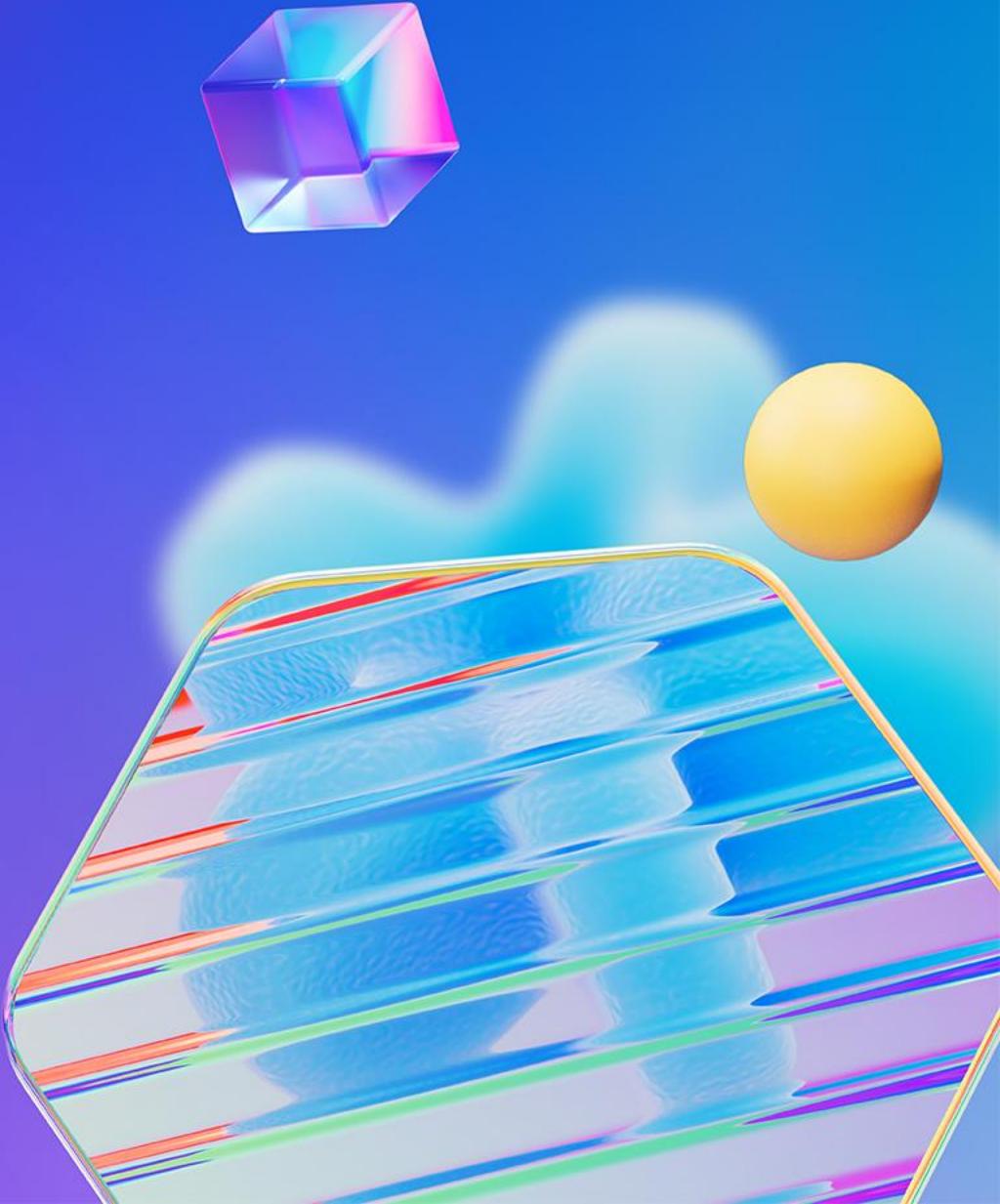
[Avg](#)

100%
90%
80%
70%

Storage utilization



Reduced TCO



Built in Optimization - Automatic Tuning



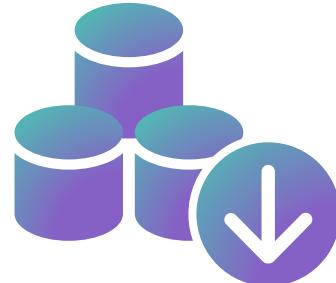
Index Recommendations

Automatically determine optimal indexes based on user activity

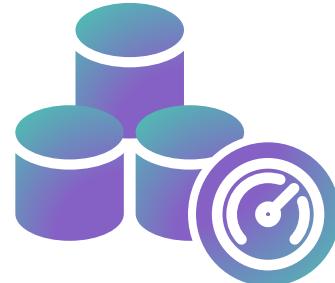


Server Parameter Tuning

Refines server parameters to maximize workload performance



Minimize Server
Resources



Maximize Workload
Performance

Autonomous Optimization

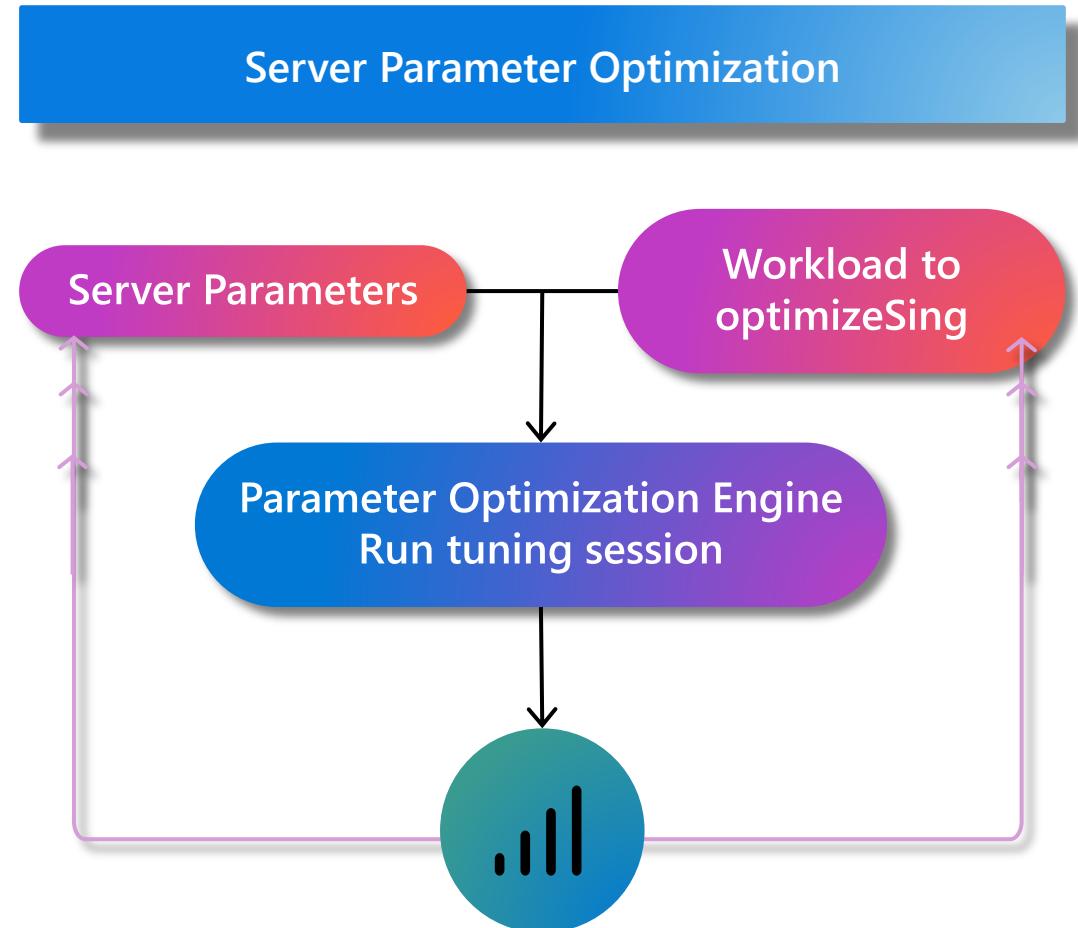
Index recommendations & server parameter tuning

Optimize index and server configuration with autonomous tuning

Improves workload performance by analyzing queries tracked by Query Store and providing index recommendations

Suggests indexes that could be added or removed during an index tuning session that can improve or reduce performance impact

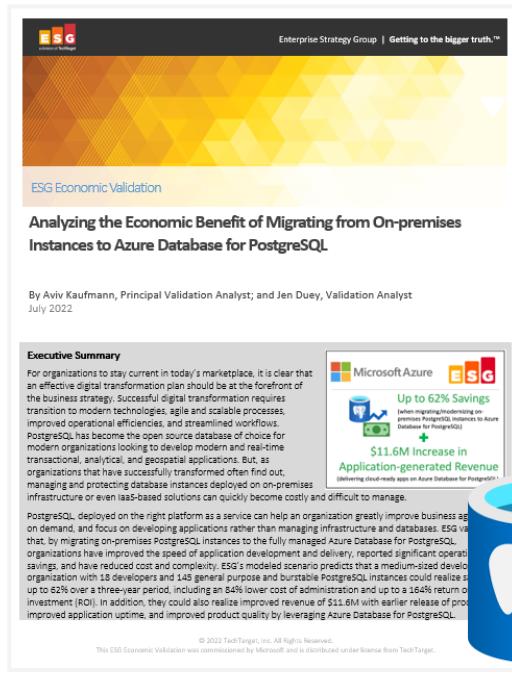
Identifies indexes not used in a configurable period that could be removed to **reduce unnecessary consumption**



Reduced resource usage
Improved Throughput

Significantly reduce cost when migrating workloads to Azure PostgreSQL Flex

Microsoft commissioned Enterprise Strategy Group to conduct an [Economic Value study](#) to examine potential cost savings and business benefits enterprises would achieve from migrating on-premises workloads to Azure Database for PostgreSQL.



Key report benefits and findings¹

Up to 62%

Cost savings when migrating on-premises PostgreSQL instances to Azure Database for PostgreSQL

Modernizing on Azure Database for PostgreSQL yields even **greater** benefits...

+\$11.6M

Additional revenue due to faster time to market

84%

Reduction in operational cost

164%

based on the expected savings and benefits

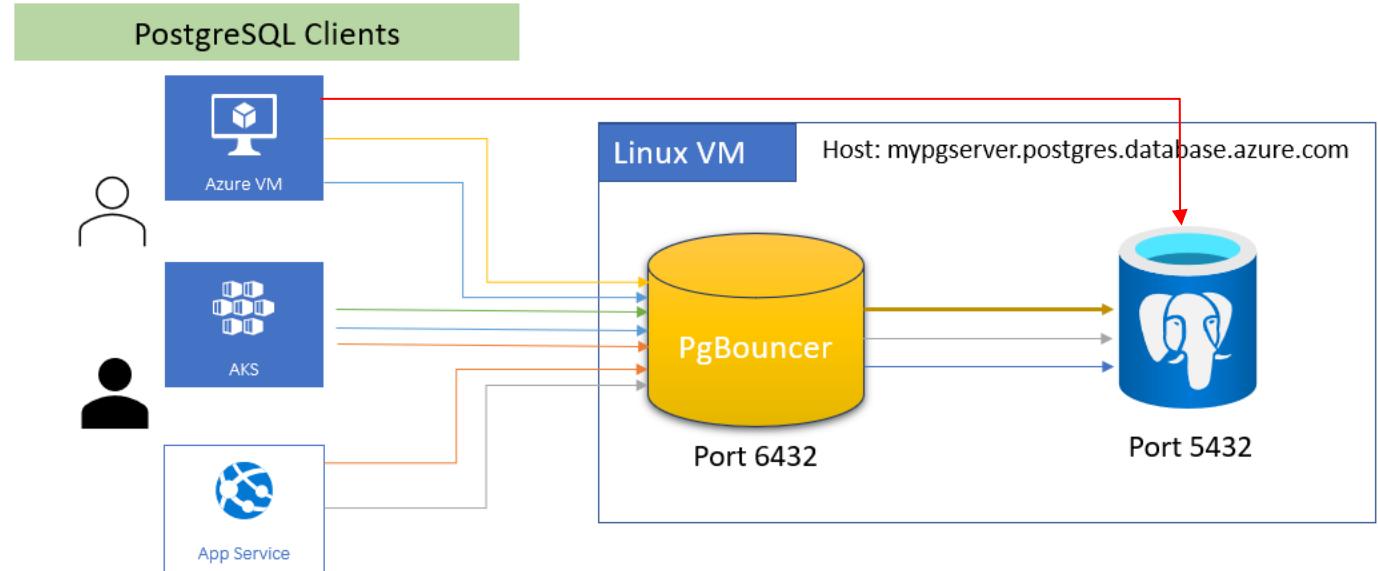
71%

Lower cost of systems administration

¹ Results are based on interviewed customers. "Analyzing the Economic Benefit of Migrating from On-premises Instances to Azure Database for PostgreSQL," a commissioned study conducted by The Enterprise Strategy Group on behalf of Microsoft, July 2022.

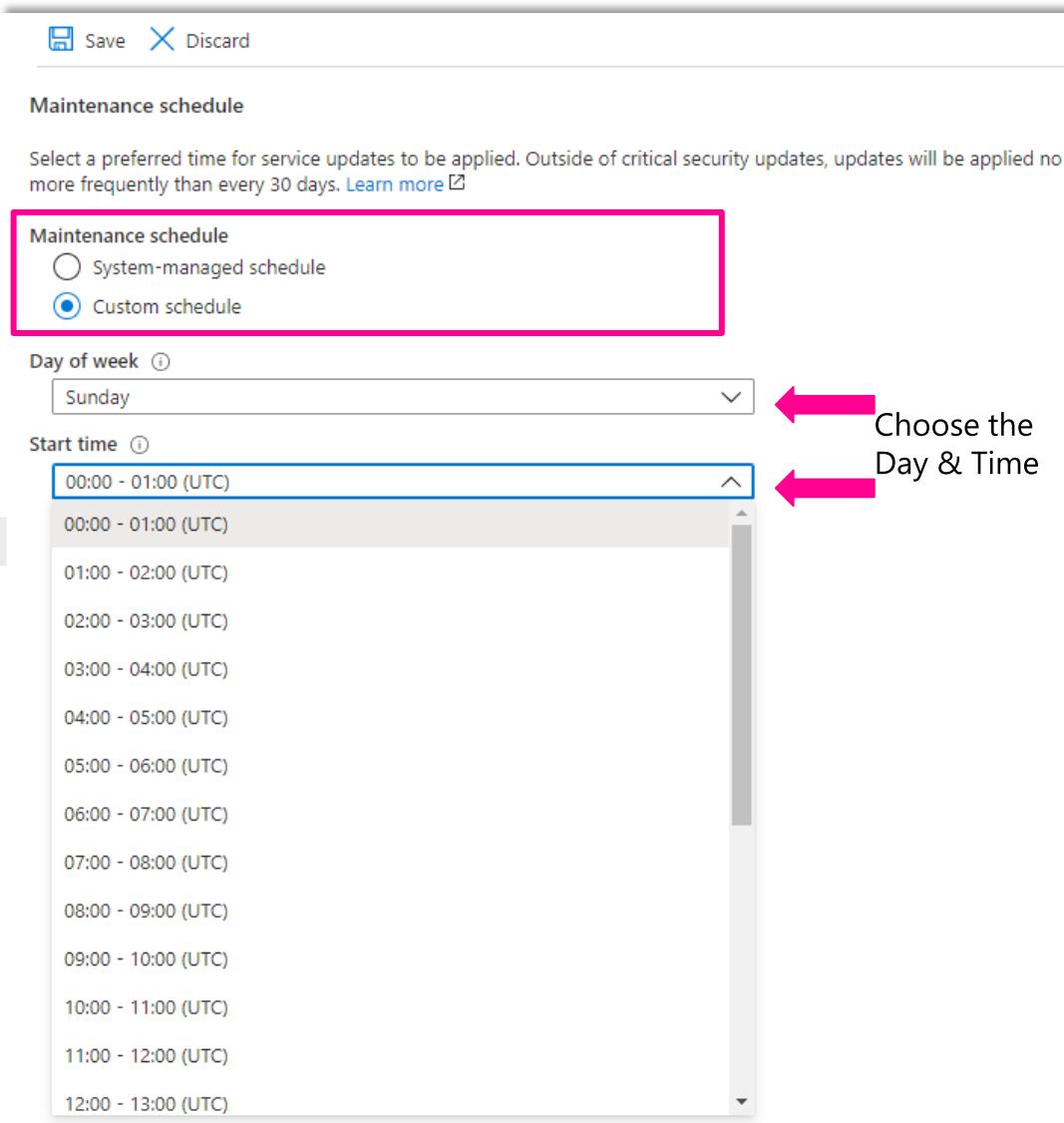
Connection Pooling - Built-in PgBouncer

- Postgres connections are expensive
- Support for managed connection pooler – PgBouncer (ver 1.22.1)
 - General Purpose & Memory optimized SKUs only
- Opt-in feature
- No need to restart
 - Dynamic
- Connection to the database server via port 6432 routes via PgBouncer
- You can connect to 5432 as well



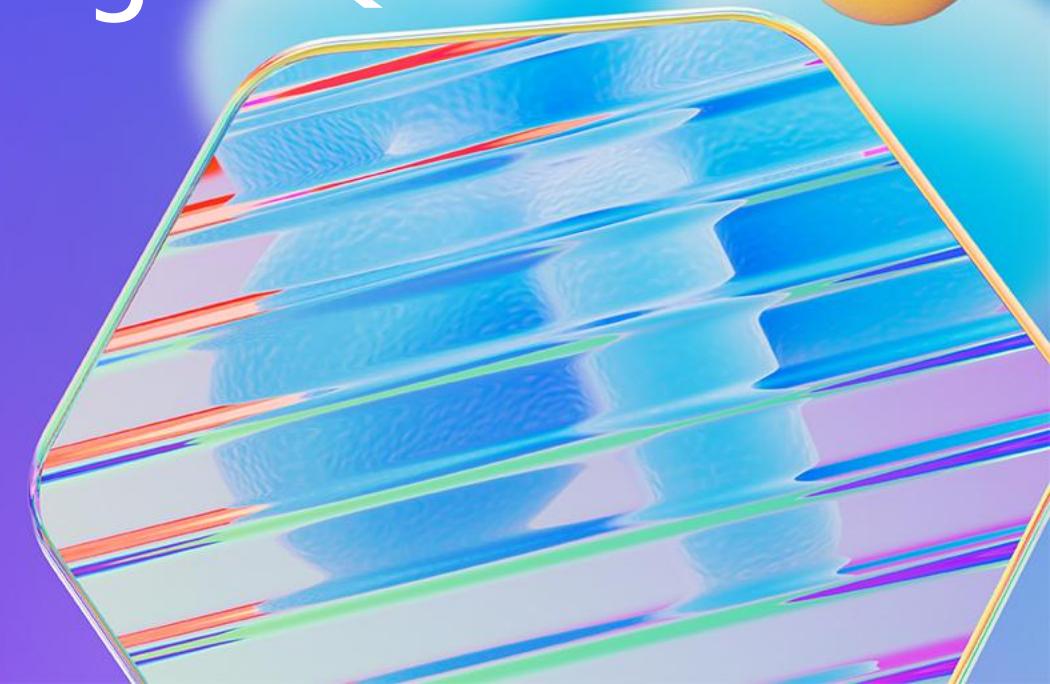
| Parameter name | VALUE | Description |
|-------------------------------------|-------------|---|
| pgbouncer.default_pool_size | 50 | How many server connections to allow per user/database pair. |
| pgbouncer.enabled | TRUE | Denotes if PgBouncer service is enabled. |
| pgbouncer.ignore_startup_parameters | | Comma-separated list of parameters that PgBouncer can ignore because they are going .. |
| pgbouncer.max_client_conn | 5000 | Maximum number of client connections allowed. |
| pgbouncer.min_pool_size | 0 | Add more server connections to pool if below this number. |
| pgbouncer.pool_mode | TRANSACTION | Specifies when a server connection can be reused by other clients. |
| pgbouncer.query_wait_timeout | 120 | Maximum time (in seconds) queries are allowed to spend waiting for execution. If the qu.. |
| pgbouncer.stats_users | | Comma-separated list of database users that are allowed to connect and run read-only ... |

Control your planned maintenance

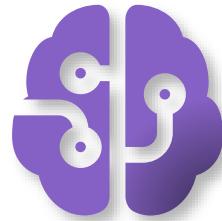
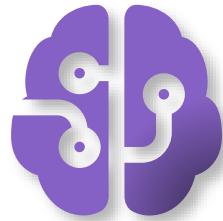
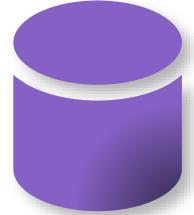


- System defined schedule or Custom schedule for patching
- Schedule your maintenance per server
- 1-hour maintenance window
- Expected downtime to be less than 60 seconds
- Next planned maintenance schedule only after 30 days
- 5 days advance notification

Integrated AI capabilities - PostgreSQL



Azure Database for PostgreSQL – AI features



Vector data type

pgvector extension – version 0.7

Native vector data type – store embeddings

Vector indexing for performant searches

Efficient ANN searches within the DB

Azure AI extension

SQL Interface to Azure OpenAI

Create embeddings from SQL Statements

SQL interface to Azure AI Language services

Invoke Azure ML models from within the DB

Integrations

LangChain

Semantic Kernel

LlamaIndex

New Cloud Apps

AI-Ready and Azure ecosystem integrated

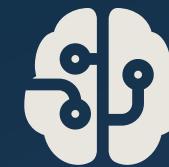
New Apps

Azure OpenAI

Azure AI Services

Azure Storage

Azure Machine Learning



Azure AI Services



Azure Database
for PostgreSQL



Open AI

AI Services integrated into Azure Postgres

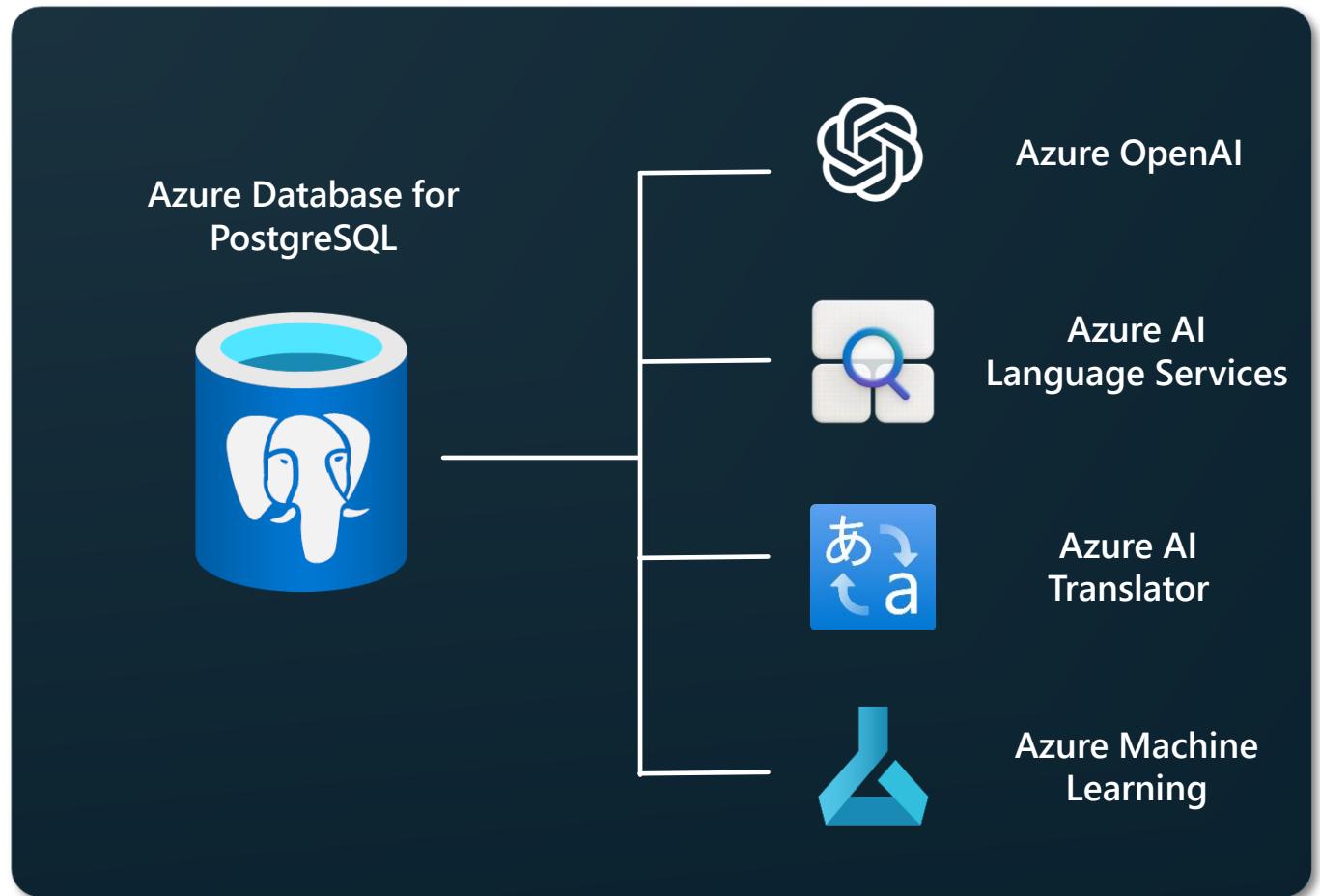
Make remote calls directly from PostgreSQL

azure_ai extension

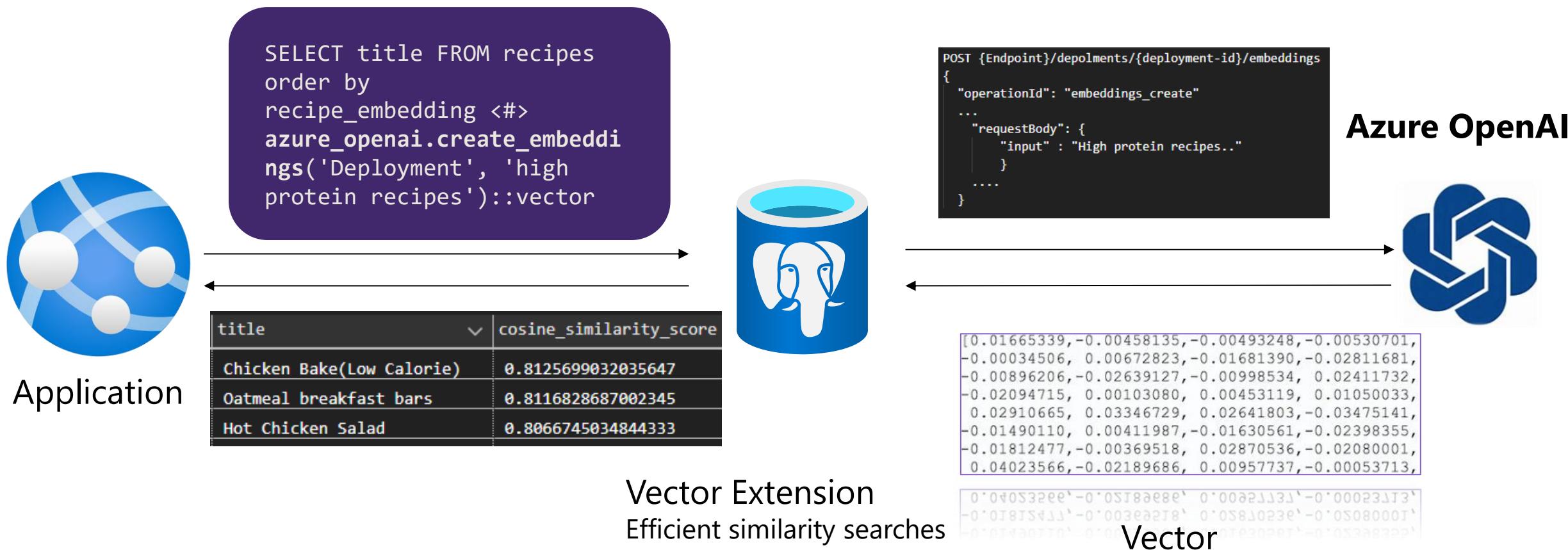
Exceptional simplicity out of the box

- Azure OpenAI
 - Azure AI Language Services
 - Azure AI Translator
 - Azure Machine Learning
-

Enables developers to rapidly adopt new AI capabilities in their solution without complex re-architecture or refactoring

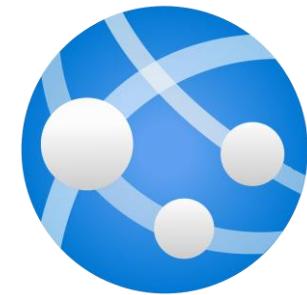


Azure AI extension – Azure Open AI



Azure Database for PostgreSQL Semantic search

Azure AI extension – Language Services



```
select b.* , a.*  
from recipe_reviews a,  
azure_cognitive.analyze_sentiment  
(review_text, 'en') b;
```



```
POST {Endpoint}/language/:analyze-text?api-version=2023-04-01  
{  
    "kind": "SentimentAnalysis",  
    "parameters": {},  
    "modelVersion": "latest"  
},  
    "analysisInput": {  
        "documents": [  
            {  
                "id": "1",  
                "language": "en",  
                "text": "Easy to make, awesome healthy recipe."  
            }  
        ]  
    }  
}
```

Azure AI
Language
Services



Application

| sentiment | positive_score | neutral_score | negative_score | review_text |
|-----------|----------------|---------------|----------------|--|
| positive | 0.99 | 0 | 0 | This was delicious, I loved it and so very easy. |
| negative | 0.25 | 0.1 | 0.65 | I did not care for it, there are better recipes. |
| neutral | 0.33 | 0.63 | 0.04 | I cannot decide, it is easy to make, but was ave |

- Sentiment Analysis
- Summarization
- Language detection
- Key phrase extraction
- PII/PHI detection

Azure AI Language Services integration

Vector Generation

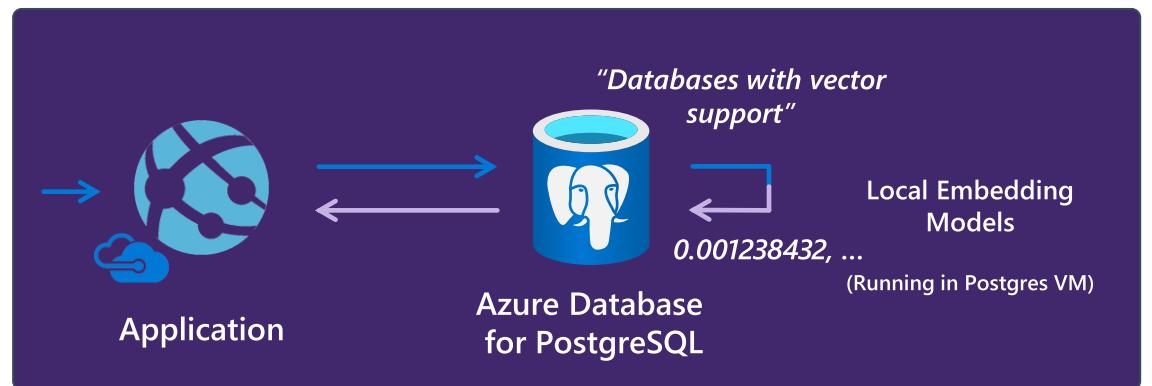
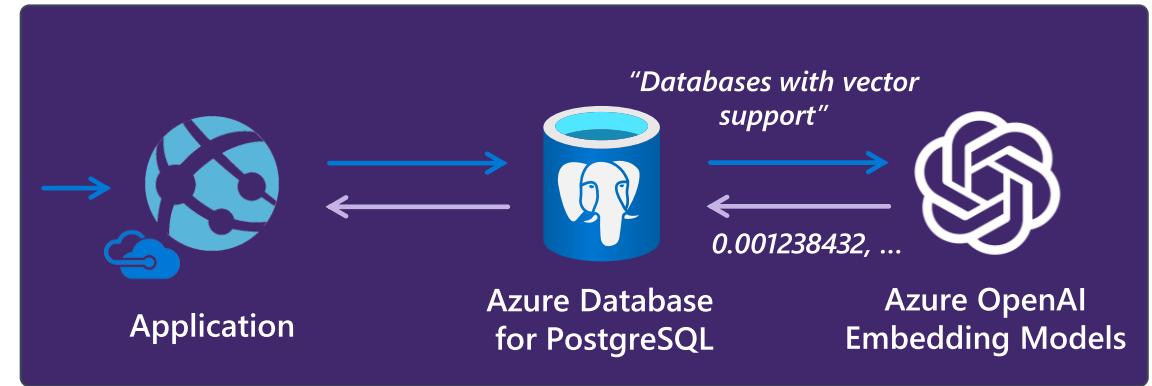
Unique Remote + In-Database Embedding Models

Remote Embedding Models

```
SELECT * FROM <table>
ORDER BY
database_description <->
azure_openai.create_embeddings(
'text-embedding-ada-002',
'Databases with vector support')
```

In-Database Embedding Models (Preview)

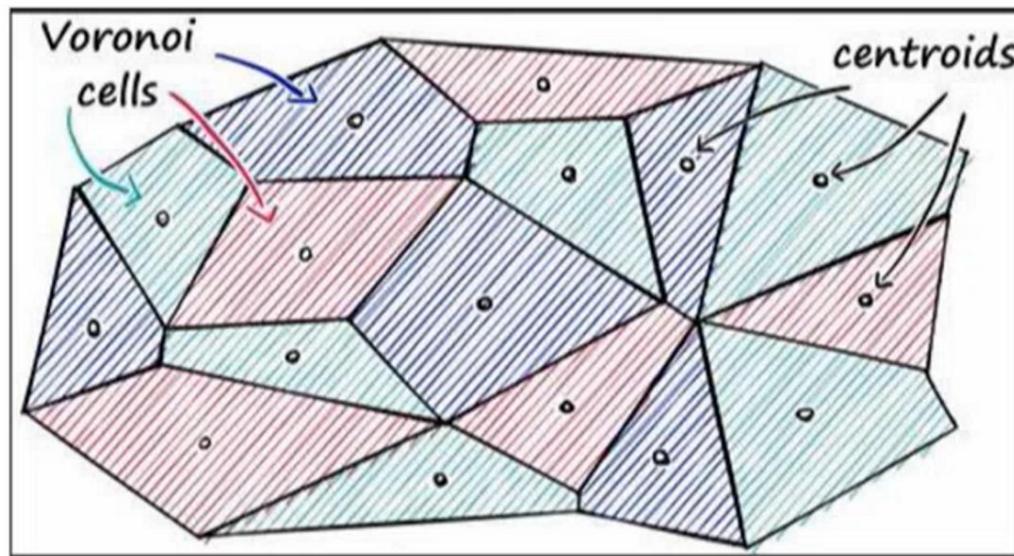
```
SELECT * FROM <table>
ORDER BY
recipe_embedding <#>
azure_local_ai.create_embeddings(
'multilingual-e5-small:v1',
'Databases with vector support')
```



Vector indexes supported today - PGVector

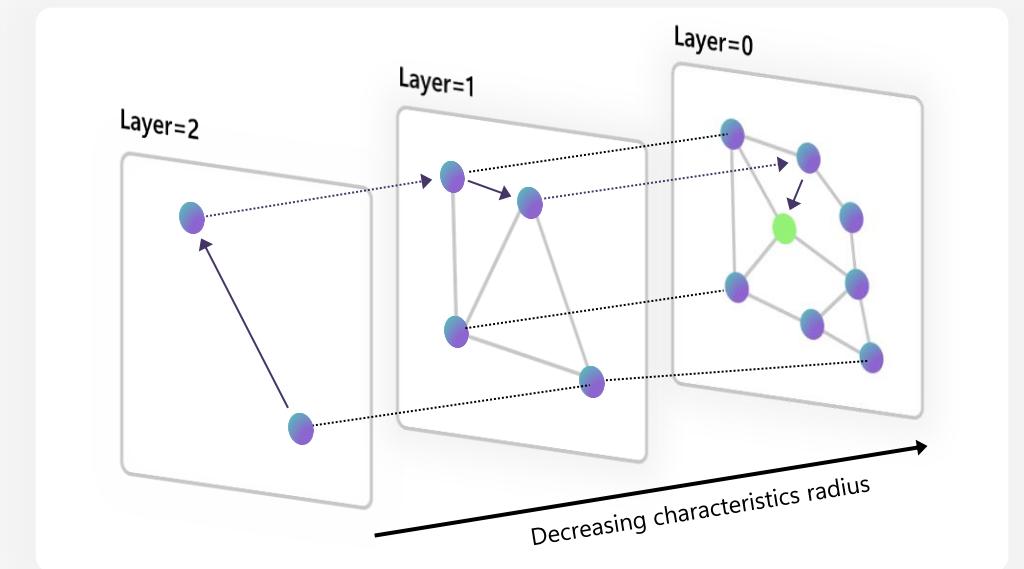
IVFFlat

- Clusters vectors by applying k-means clustering.
- Memory efficient but requires index rebuilds.



HNSW

- Builds a multi-layer graph with long and short connections between the vectors.
- The graph can be incrementally updated.



DiskANN Vector Index

Preview



Unique to
Azure

Highly **performant, scalable, and accurate** index for vectors

Superior to IVFLAT and HNSW

Reduced memory footprint by storing vectors on SSD

Compression and quantization **improve speed and accuracy** of vector search

Accuracy retained as data changed

Vector compression

Large Vectors

{ D1, D2, D3, D4, D5, ..., D99, D100 }

Quantization

Compressed Vectors

{ D1, D2 .., D10 }

Optimized storage

RAM

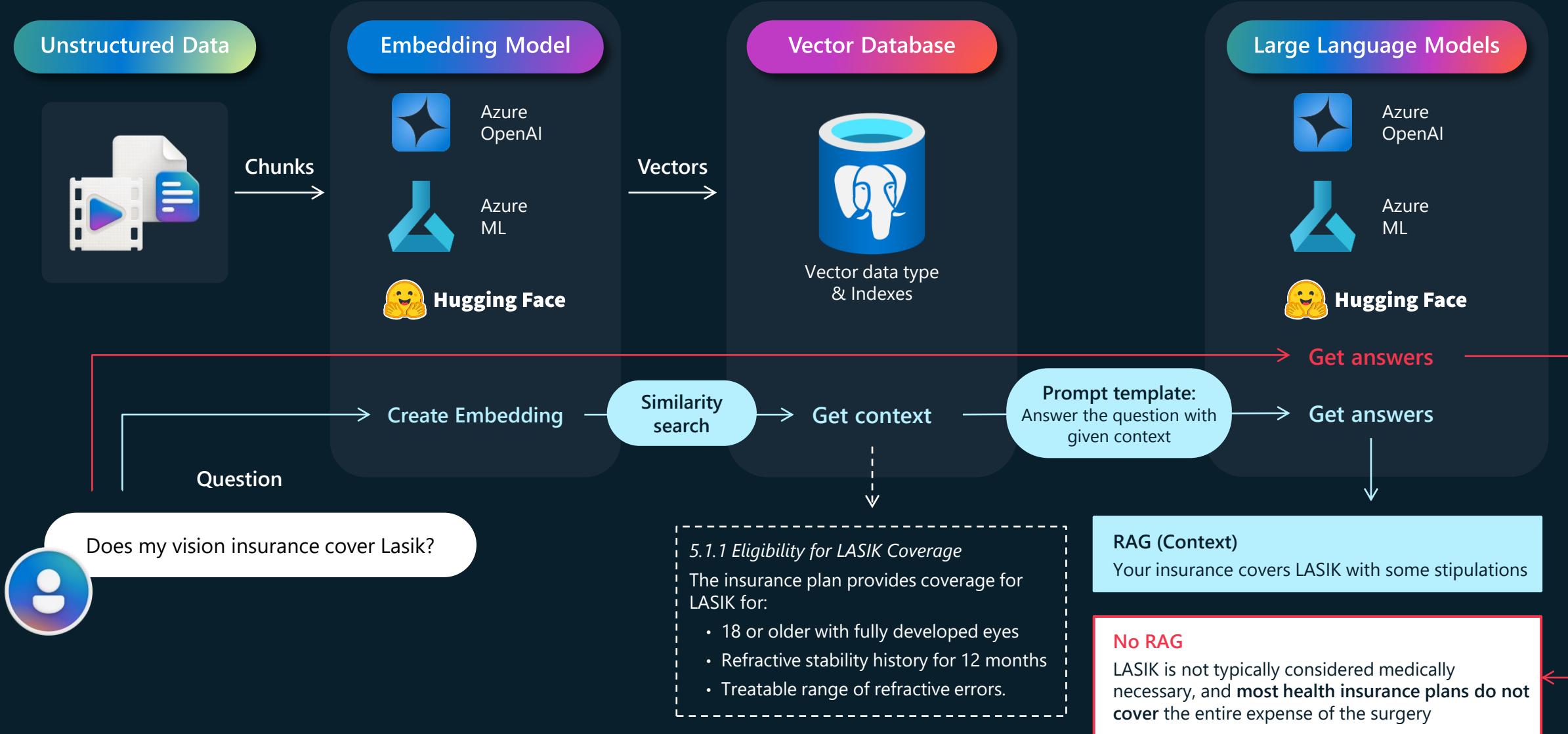
Compressed vectors

Optimized for minimal SSD reads

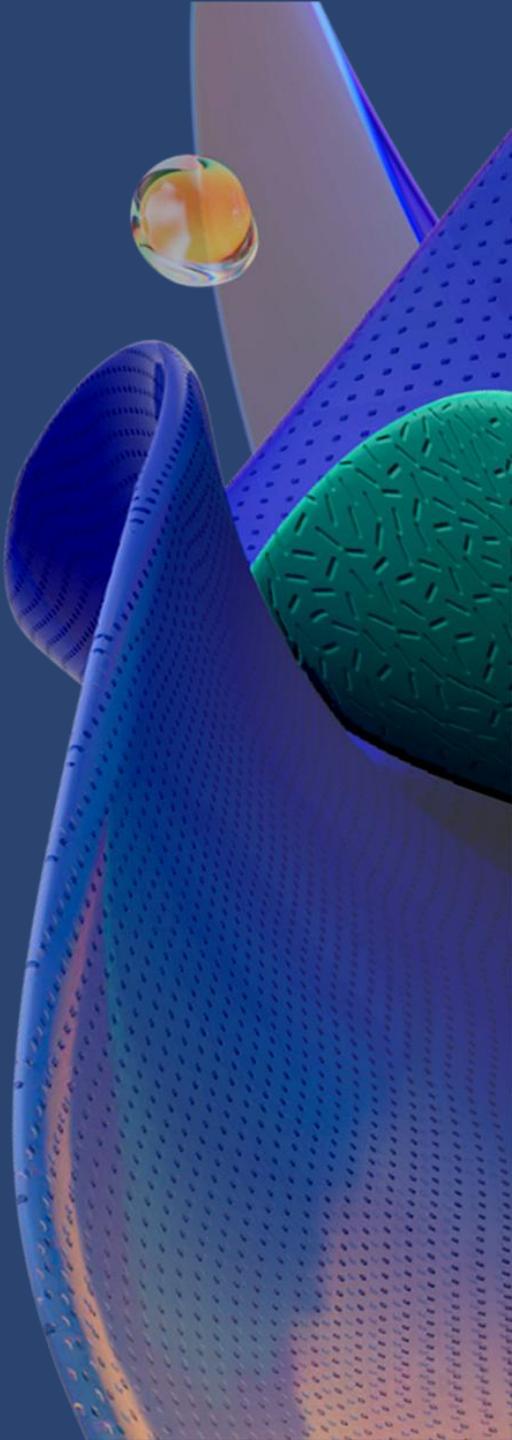
SSD

Full vectors + graph

RAG Pattern with Azure Database for PostgreSQL



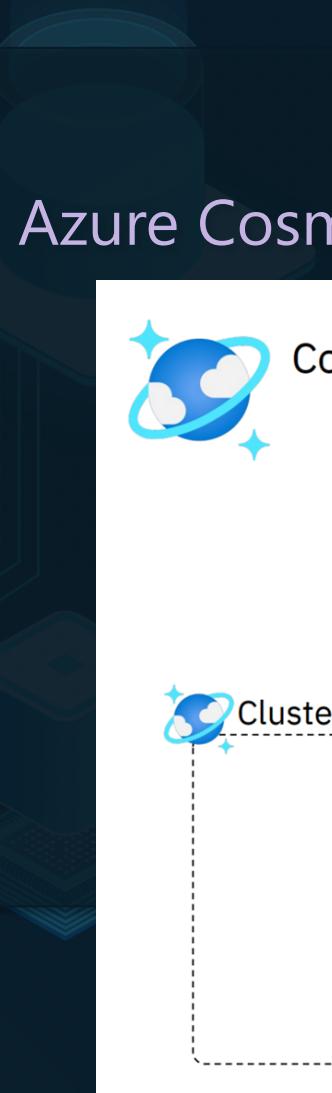
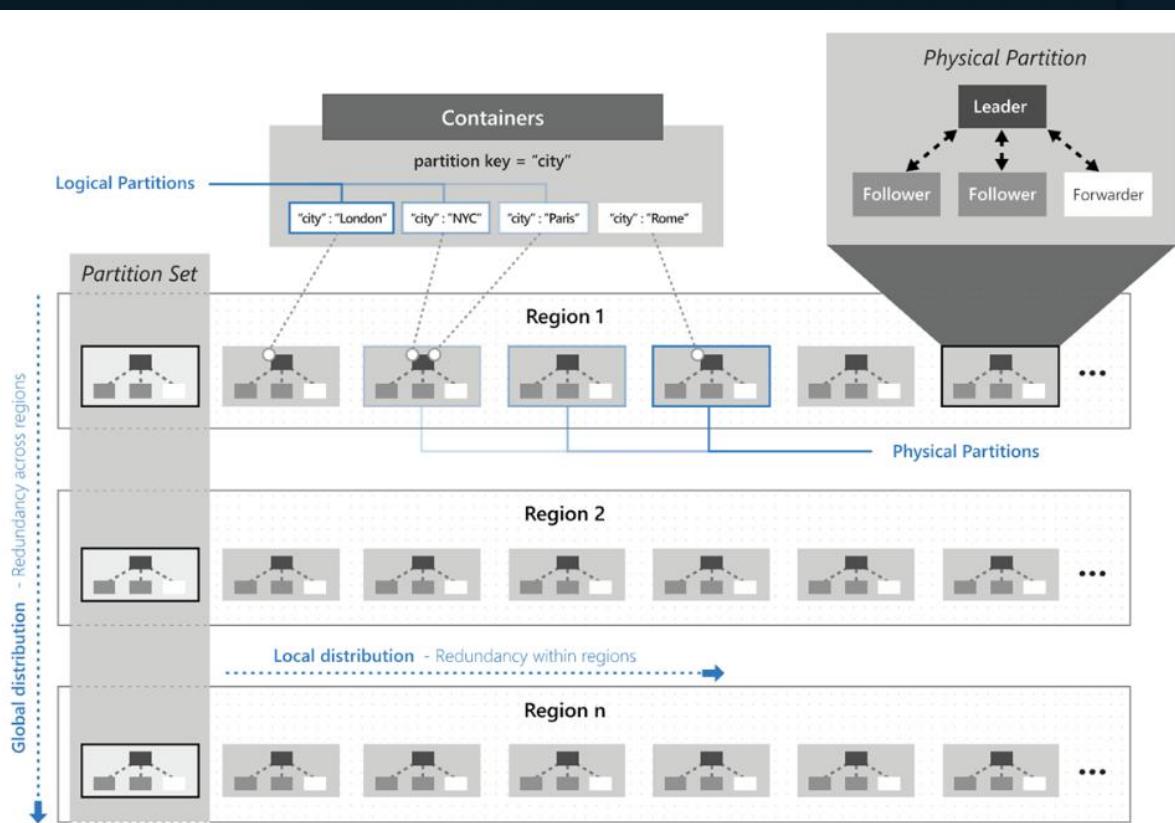
Integrated AI capabilities – Cosmos DB



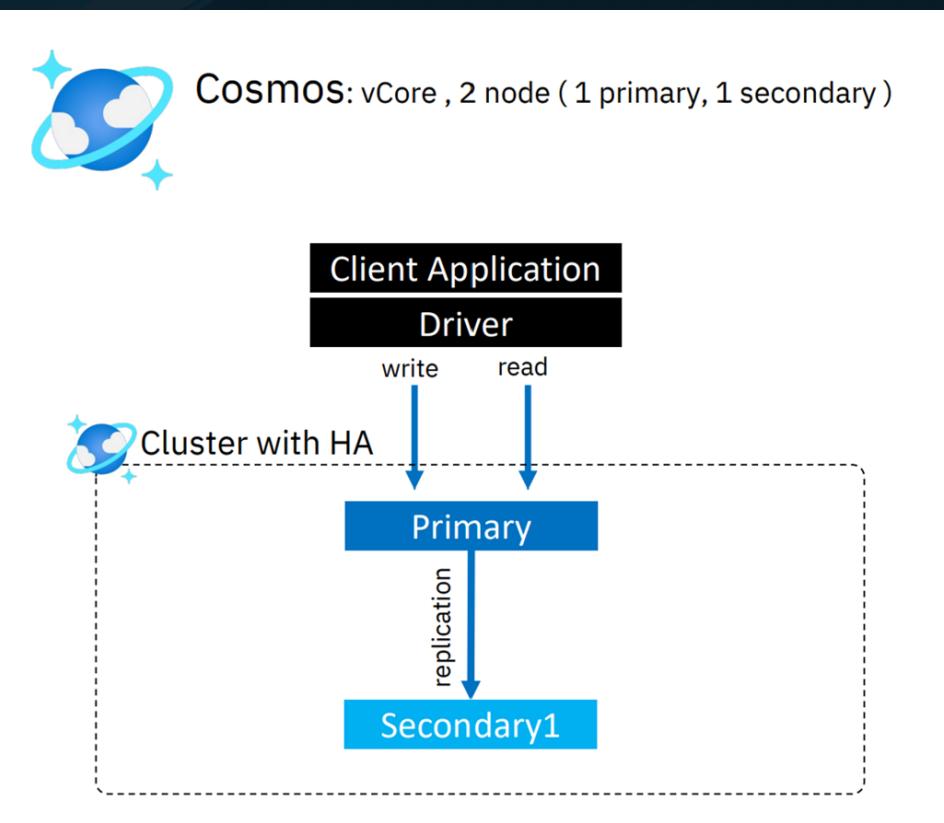
Azure Cosmos DB is a set of highly-scalable & AI-ready databases



Azure Cosmos DB NoSQL



Azure Cosmos DB for MongoDB vCore



Azure Cosmos DB is a set of highly-scalable & AI-ready databases



Azure Cosmos DB NoSQL

- High elasticity. Instant Autoscale/Serverless
- Low latency, real-time data transactions
- Mission critical reliability (99.999%)
- Built-in vector index and search (ft. DiskANN)



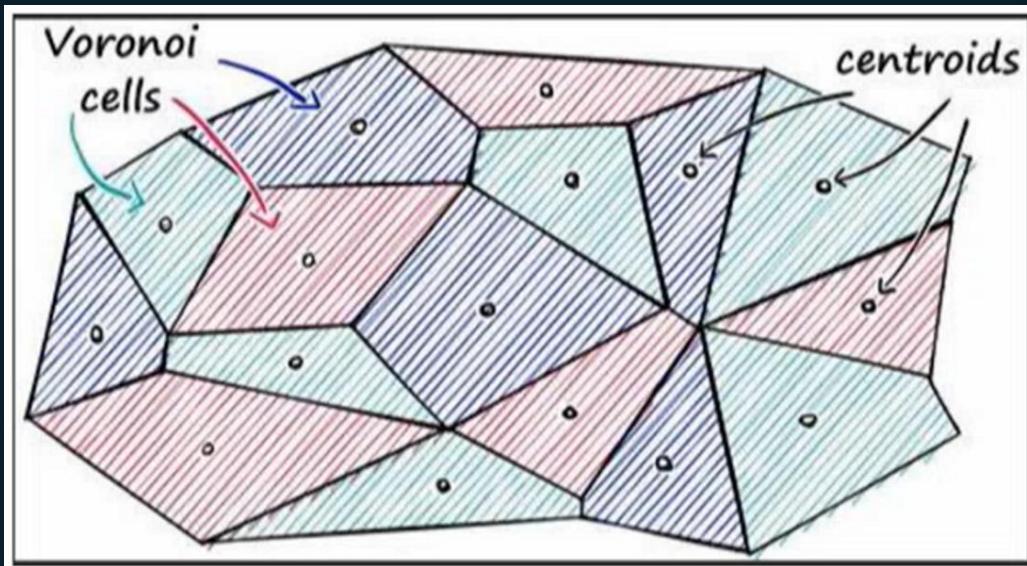
Azure Cosmos DB for MongoDB

- Provision compute + storage
- Store data + vectors together, keep consistent.
- High reliability (99.995%)
- Built-in vector index and search (ft. IVF & HNSW)
- DiskANN (preview)

Vector indexes – IVFFlat and HNSW

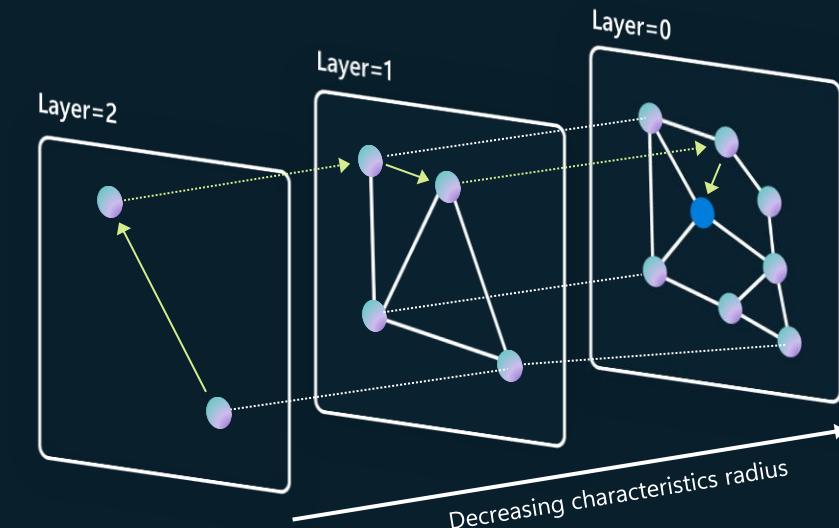
IVFFlat

- Clusters vectors by applying k-means clustering.
- Memory efficient but requires index rebuilds.



HNSW

- Builds a multi-layer graph with long and short connections between the vectors.
- The graph can be incrementally updated.



Vector indexing – MongoDB vCore

IVF (Inverted File Index)

- Partitions vectors into clusters and assigns each vector to one cluster.
- **Building the index is fast and memory-efficient**
- Requires a separate clustering step before indexing (slow)
- **Tuning parameters is important.** Can be very accurate if configured properly

HNSW (Hierarchical Navigable Small World)

- Builds a multi-layer graph with long and short connections between the vectors.
- **Robust and accurate at scale**
- No-preprocessing step.
- **Can support many inserts/deletes efficiently.**
- **Larger memory footprint**
- It also has many parameters (such as the number of layers and neighbors) that need to be tuned carefully.

Cosmos DB NoSQL vector indexing and search

DiskANN is a suite of ANN indexing algorithms that interfaces a disk-based index with an in-memory index for efficient querying.

- **In-Memory Index:** DiskANN maintains a small in-memory index that acts as a gateway to the larger on-disk index, allowing for quick access to the most relevant data points.
- **On-Disk Index:** The bulk of the data is stored in an optimized on-disk index, which is designed to minimize disk seeks and read operations during a query to maximize performance.
- **Query Processing:** When a query is received, DiskANN first consults the in-memory index to identify the best entry points into the on-disk index, then efficiently finds the nearest vectors.



DiskANN in Azure Cosmos DB for NoSQL

Compression of vectors

Vector with D dimensions
{1.23342, -1.31292, 2.39247, -1.27285, ... }

Quantization

{ q₁, q₂, q₃, q₄, ... }

Compressed vector with m << D dimensions

Storage and graph construction

RAM

{Compressed vector 1}
 {Compressed vector 2}
 ⋮

Optimized for minimal
SSD reads

SSD

Low-diameter graph
+ full-precision vectors



Algorithms

Vamana (index)

Search

Pruning

Document Embedding and processing pipeline

Integrate multiple Azure services and handles chunking/vectorization
Azure/document-vector-pipeline



Full Text search & ranking

Full Text Search & Ranking

- Search for text in documents
- Improved search relevancy and efficiency with a new Full Text Index
- Rank text search results by relevancy with BM25



Full Text Search

Quickly and efficiently find keywords and terms in properties



Text Analyzers

Tokenization
Stopword removal
Stemming



Text Ranking with BM25

Find the documents most relevant to your search

Hybrid search

Hybrid Search for improved search relevancy

Reciprocal Rank Fusion (RRF)

Vector Similarity Score

+

Full Text Scoring (BM25)

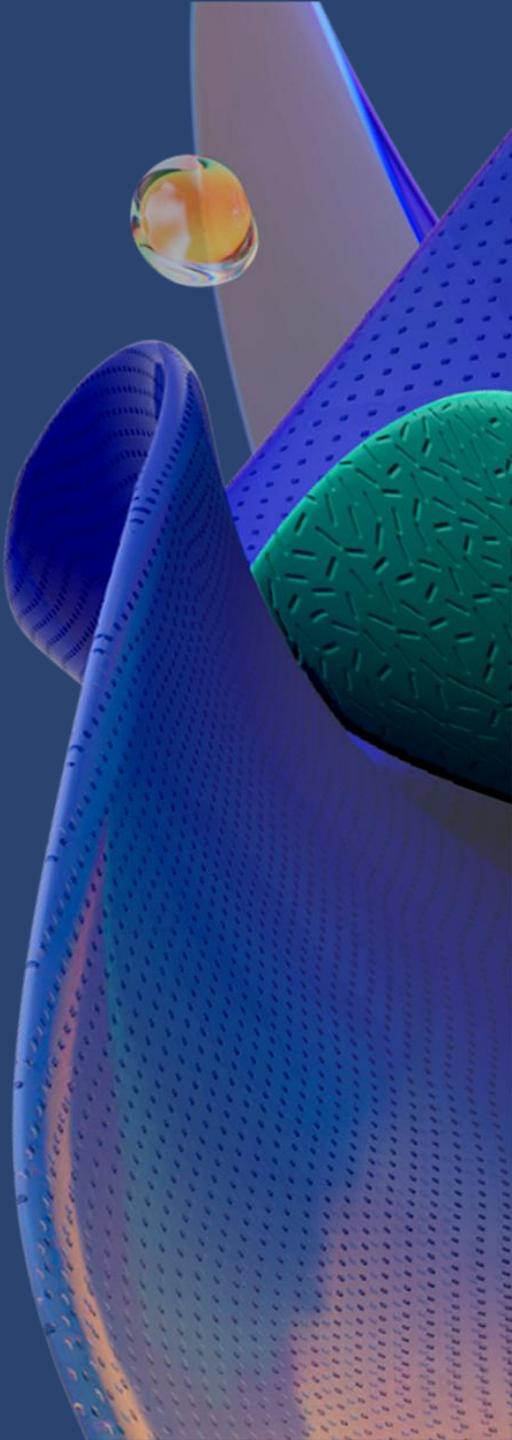
=

More Relevant Search Results

Example Query

```
SELECT TOP 10 *
FROM c
ORDER BY RANK RRF(FullTextScore(c.text,['keyword1','keyword2']),
VectorDistance(c.vector, @vector))
```

Integrated AI capabilities – Azure SQL





Azure SQL Database

Fast, flexible, secure modern app development for workloads big or small



Optimize application
performance



Deliver AI-ready
apps faster

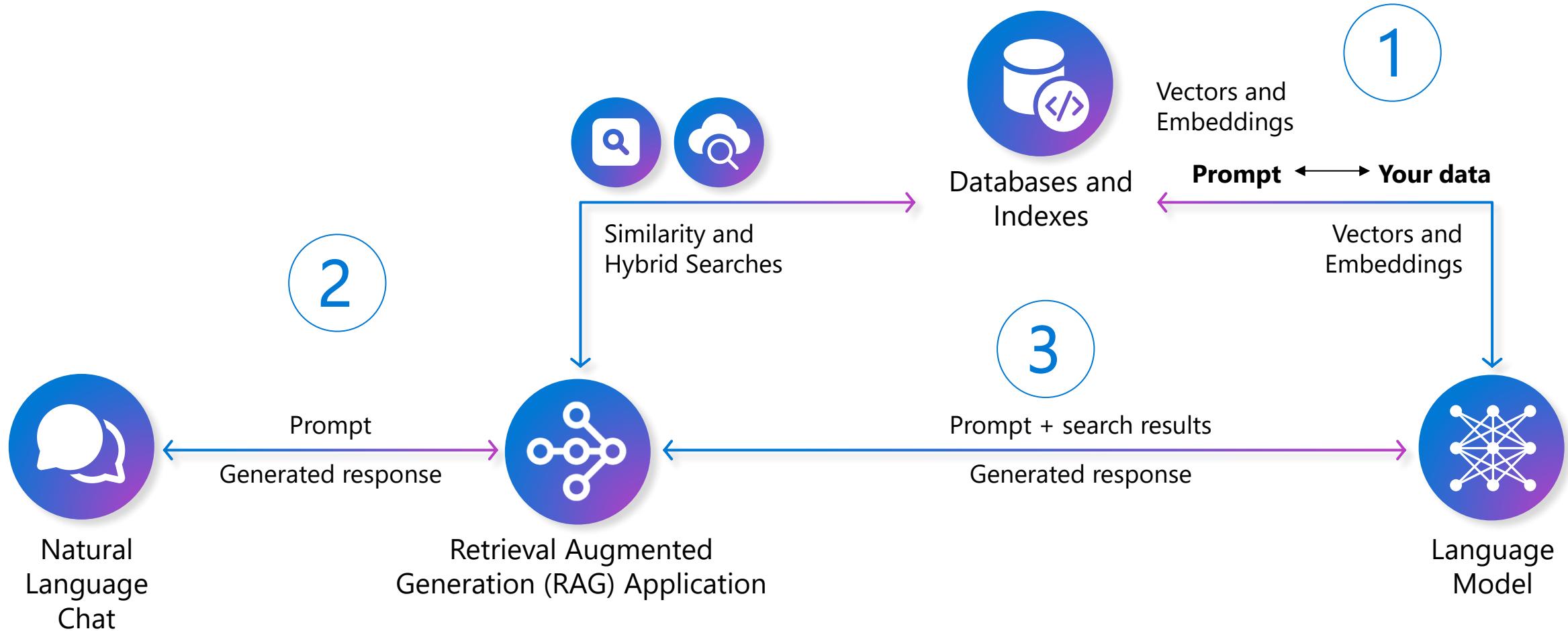


Build applications
that grow with you



Keep your data secure
and available

Why GenAI with your database?



New Developer Value In Azure SQL Database

Native JSON type
and aggregates in
Azure SQL Database

Public Preview

MSSQL
SDK Style Projects VS

Preview

MSSQL Extension for
Visual Studio Code
(v1.25.0)

Preview

Native vector type
and functions in
Azure SQL Database

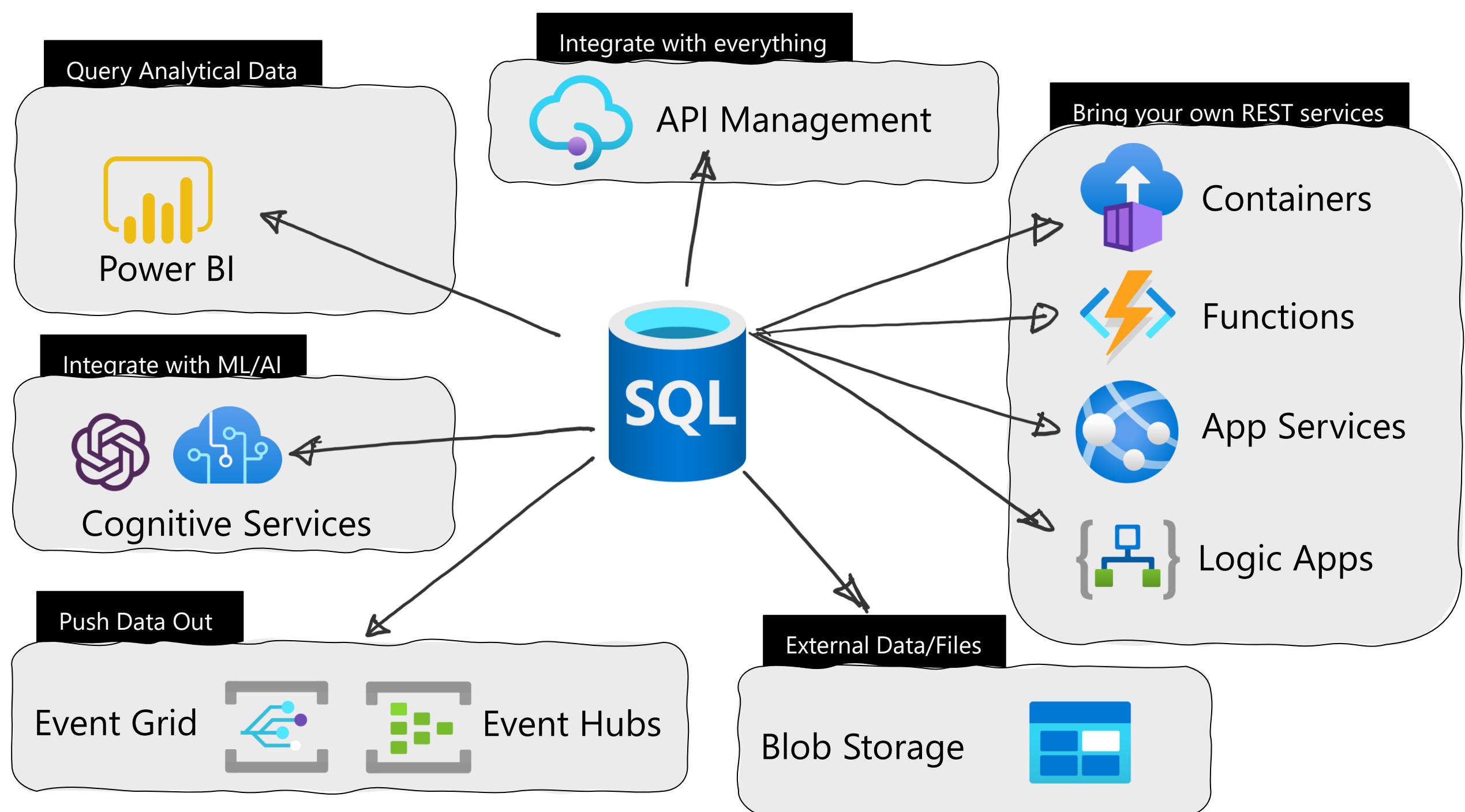
Public Preview

Vector Integration
with LangChain, EF
Core, Semantic
Kernel and Logic
Apps

Public Preview

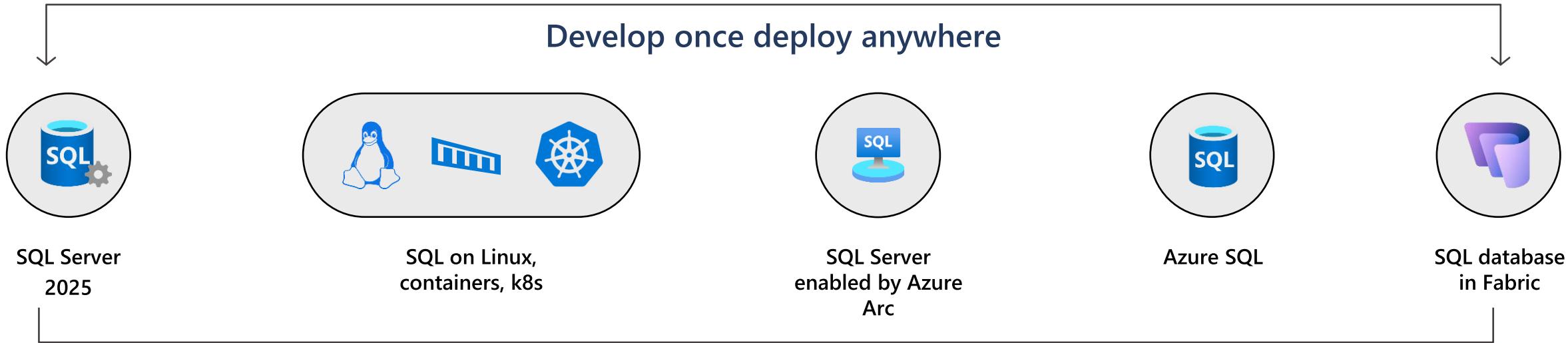
Vector Index with
DiskANN

Coming Soon

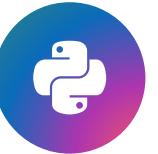


Microsoft SQL – ground to cloud to fabric

The next generation enterprise ready AI database



Building on a foundation



SQL Server 2017

SQL Server on Linux
Containers
Adaptive query processing
Automatic Tuning
Graph database
Machine learning services



SQL Server 2019

Data virtualization
Intelligent query processing
Accelerated database recovery
Data classification



SQL Server 2022

Cloud Connected
IQP NextGen
Ledger
Data Lakes
T-SQL enhancements

SQL Server 2025

Enterprise AI-ready database from ground to cloud

