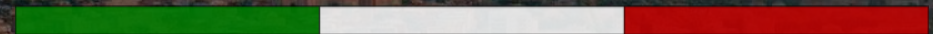


An aerial photograph of a city, likely Rome, showing a dense urban landscape with a river (the Tiber) winding through it. The image is used as a background for the conference title.

DELPHIDAY



italian conference

Azure SQL Database Essentials

From deployment to maintenance: initial
choices and daily operations



SERGIO GOVONI

CENTRO SOFTWARE S.P.A.



segovoni.medium.com



bit.ly/sgovoni-MVP



twitter.com/segovoni



github.com/segovoni



linkedin.com/in/sgovoni

DELPHIDAY

italian conference

9-10 Giugno 2026
Piacenza



wintech
italia

OPEN-SOURCE PROJECTS

github.com/segovoni

SQL command-line utility

github.com/segovoni/sqlcmdcli

Alter column with dependencies

github.com/segovoni/sp_alter_column

Conference demos

github.com/segovoni/sql-server-demos

DELPHIDAY

italian conference

9-10 Giugno 2026
Piacenza



wintech
italia



AGENDA

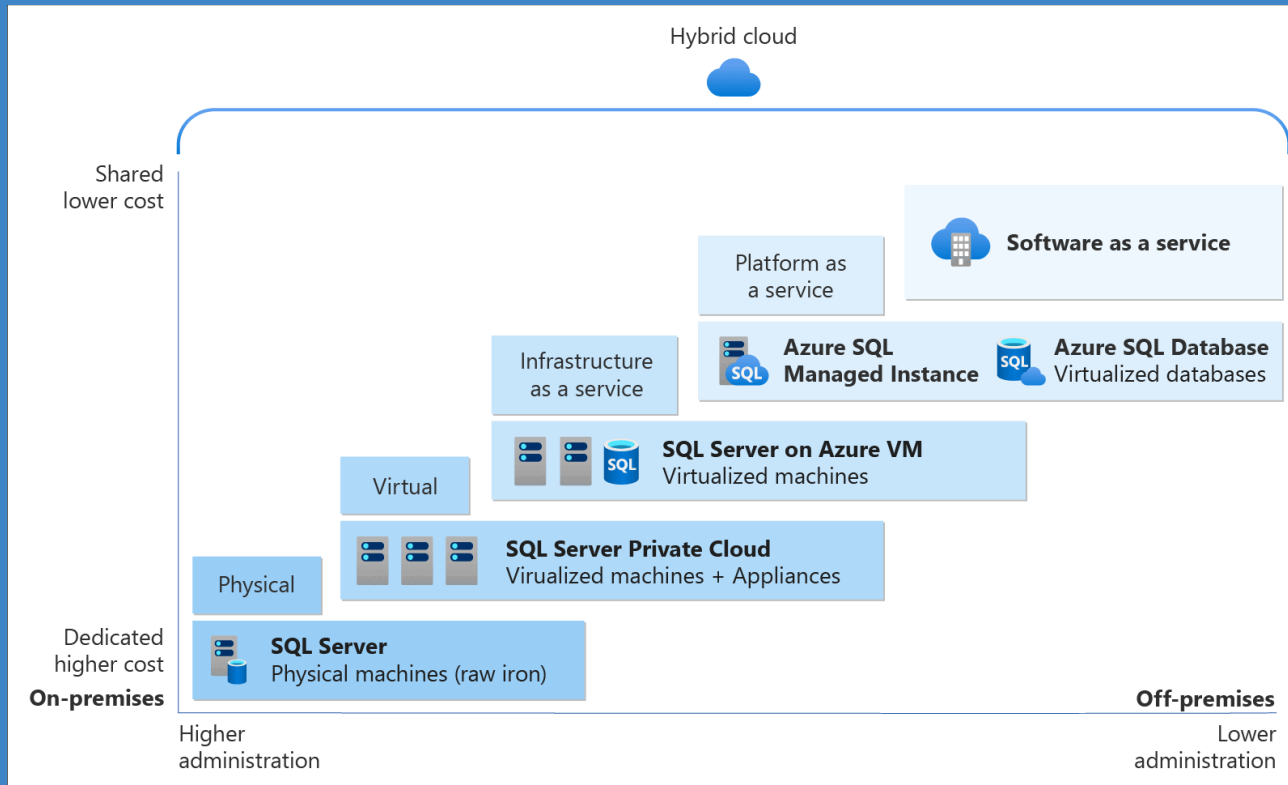
- Overview of SQL solutions in Azure
- Purchasing models
- Service tiers
- Compute tiers
- Provisioning and deploy
- Networking
- Backup and restore
- Maintenance activities
- High availability



Overview of SQL solutions in Azure



Overview of SQL solutions in Azure





Overview of SQL solutions in Azure

SQL Server on Azure Virtual Machines



Best for re-hosting and apps requiring
OS-level access and control
Automated manageability features and
OS-level access

Azure SQL Managed Instance



Best for modernizing existing apps
Offers high compatibility with SQL Server
and native VNET support

Azure SQL Database



Best for building new apps in the cloud
Pre-provisioned or serverless and
Hyperscale to meet demanding workload
requirements

Infrastructure as a Service (IaaS)

Platform as a Service (PaaS)



Familiar drivers, T-SQL, engine, and tools



Overview of SQL solutions in Azure

SQL virtual machines

Best for migrations and applications requiring OS-level access



SQL virtual machine

- SQL Server and OS server access
- Expansive SQL And OS version support
- Automated manageability features for SQL Server

Managed instances

Best for most lift-and-shift migrations to the cloud



Single instance

- SQL Server surface area (vast majority)
- Native virtual network support
- Fully managed service



Instance pool

- Pre-provision compute resources for migration
- Enables cost-efficient migration.
- Ability to host smaller instances (2Vcore)
- Currently in public preview

Databases

Best for modern cloud applications. Hyperscale and serverless options are available



Single database

- Hyperscale storage (up to 100TB)
- Serverless compute
- Fully managed service



Elastic pool

- Resource sharing between multiple databases to price optimize
- Simplified performance management for multiple databases
- Fully managed service



SQL Server on Azure VMs

- IaaS model with full control
- Best fit for full SQL Server compatibility
- Supports custom OS and SQL configuration
- You manage backups, patching, and monitoring
- Highest isolation and environment flexibility





Azure SQL Managed Instance

- PaaS with high SQL Server compatibility
- Best for lift-and-shift migrations
- Managed instance with reduced admin work
- Supports SQL Agent and instance-level features
- Works well with virtual network integration
- More isolated than Azure SQL Database





Azure SQL Database

- Fully managed PaaS database service
- Best for modern cloud applications
- Built for scalability and high availability
- Advanced security with less administration
- Supports single databases and elastic pools
- Backups, patching, and updates are automated





Azure SQL Database

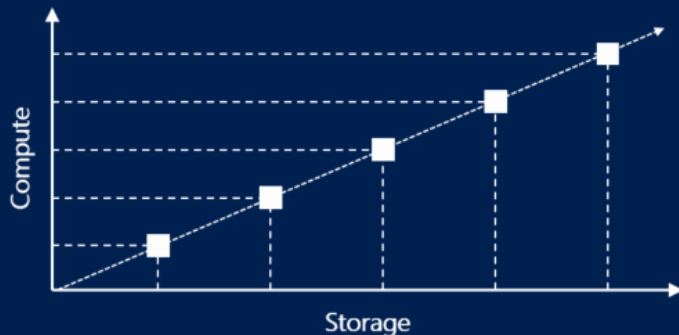


Purchasing models



Purchasing models

DTU model
Simple,
Preconfigured

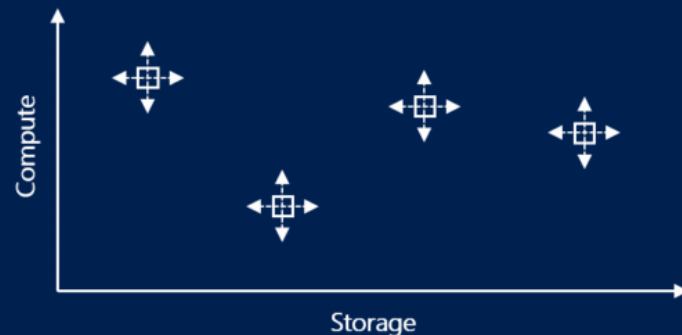


Database Transaction Unit (DTU)-based model

- Bundled measure of compute, storage and IO resources
- Best for customers who want simple, pre-configured resource options.

OR

vCore model
Independent scalability



vCore-based model

- Independent scaling of compute, storage and IO resources
- Best for customers who value flexibility, control and transparency
- Use with Azure Hybrid Benefit for SQL Server to gain cost savings



Purchasing models

Model	Description	Use case	Service level
DTU-based	Bundled compute, storage, and I/O model based on DTUs for single databases and eDTUs for elastic pools.	Customers who want simple, preconfigured resource options	Basic, Standard, Premium
vCore-based	This model allows you to independently choose compute and storage resources with Azure Hybrid Benefit support	Customers who value flexibility, control, and transparency	General Purpose, Business Critical, Hyperscale



Start for free

- <https://aka.ms/sqlfreeoffer>
- Monthly free limits
 - 100,000 vCore seconds
 - Max of 4 vCores
 - Max 32 GB of data per database
 - Up to 10 databases per Azure subscription
 - Long-term backup retention isn't available
 - Backup is local redundant only



Start for free

- <https://aka.ms/sqlfreeoffer>
- Free limit reached options
 - Auto-pause the database until next month
 - Continue using database for additional charges
 - Keep the database online
 - Standard General-Purpose tier serverless rates



Service tiers



Service tiers: DTU

- A database transaction unit (DTU) represents a blended measure of CPU, memory, reads, and writes
- For a single database at a specific compute size within a service tier, Azure SQL Database guarantees a certain level of resources for that database (independent of any other database)
- [Azure SQL Database DTU Calculator](#)



Service tiers: DTU

- DTU-based purchasing model offers three service tiers:
 - Premium for OLTP applications with high transaction rates, low-latency I/O, and the highest resilience through multiple isolated replicas
 - Standard for less demanding workloads with balanced, budget-oriented compute and storage
 - Basic for small applications with minimal performance requirements and lower cost.



Service tiers: vCore

- vCore-based purchasing model offers three service tiers:
 - Hyperscale (recommended) for most business workloads, with independently scalable compute and storage, high performance, high availability replicas
 - Business Critical for OLTP applications with high transaction rates and low-latency I/O, with high resilience through several isolated replicas
 - General Purpose for less demanding workloads with balanced, budget-oriented compute and storage



Compute tiers



Compute tiers

- vCore-based purchasing model provides two different compute tiers for Azure SQL Database:
 - Provisioned compute tier
 - Serverless compute tier
- DTU-based purchasing model provides just the provisioned compute tier



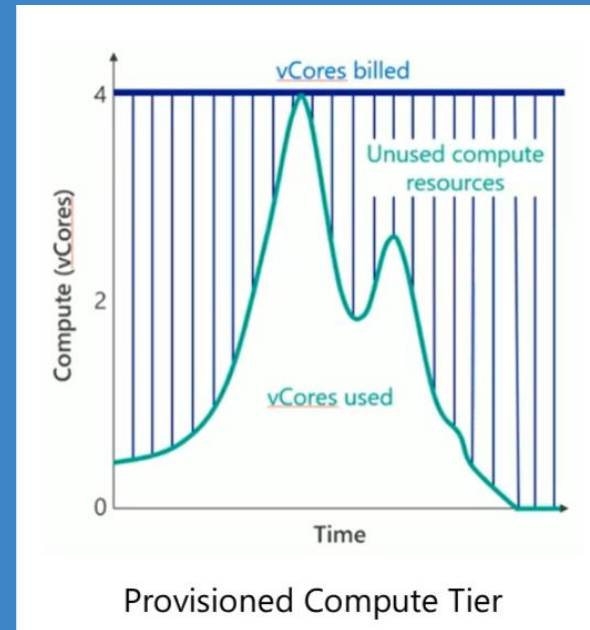
Compute tiers

- **Provisioned** compute tier provides a specific amount of compute that is continuously provisioned independent of workload activity, and bills for the amount of compute provisioned at a fixed price per hour
- **Serverless** compute tier automatically scales compute resources based on workload activity and bills for the amount of compute used, per second
 - Available in the vCore purchasing model, in the General Purpose and Hyperscale service tiers



Provisioned compute tiers

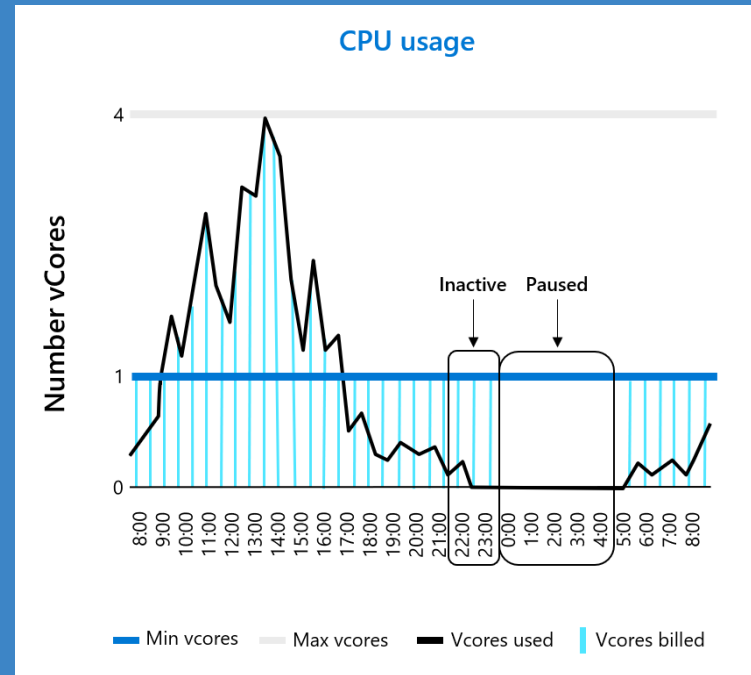
- Preallocation is effective for predictable patterns
- Inefficiencies
 - Overprovisioning during low-usage periods
 - Underprovisioning to reduce costs





Serverless compute tiers

- Sporadic or unpredictable usage
- Compute sizing uncertainty or workloads requiring frequent rescaling to reduce costs
- Dev/test workloads





Serverless compute tiers

- Automatically scales compute based on workload demand
- Billed per second for compute used
- Auto-pause and auto-resume reduce compute cost during idle periods
- Less suitable for workloads that need constant performance
- Available in vCore model only
 - Supported in General Purpose and Hyperscale
- Auto-pause/auto-resume currently supported only in General Purpose



Serverless auto-pause, auto-resume

- Supported only in the General-Purpose tier
- Auto-pause happens after the configured auto-pause delay
- Auto-pause requires zero sessions and zero user workload CPU
- When paused, compute cost is zero; only storage is billed
- The next login or activity automatically resumes the database
- The first connection attempt can return error 40613, then retry



Serverless auto-pause, auto-resume

- What prevents auto-pause:
 - Geo-replication and failover groups
 - Long-term backup retention (LTR)
 - SQL Data Sync sync database
 - DNS alias on the logical server
 - Elastic Jobs job database



Service and compute tiers

DTU

Standard

Premium

vCore

General Purpose

Provisioned

Serverless

Business Critical

Provisioned

Hyperscale

Provisioned

Serverless



Provisioning and deploy



Choose the right Azure SQL solution

→ Decision tree explained

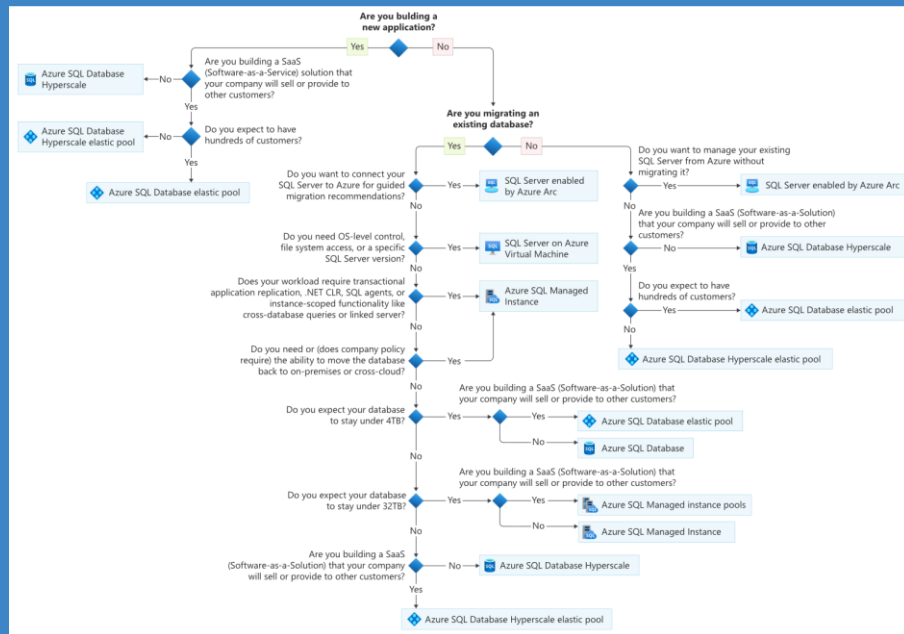
→ aka.ms/AzureSQLdecisionTree

→ Hands-on decision tree

→ aka.ms/azuresqlhub

→ Compare databases

→ azure-sql/database/features-comparison





Deployment models

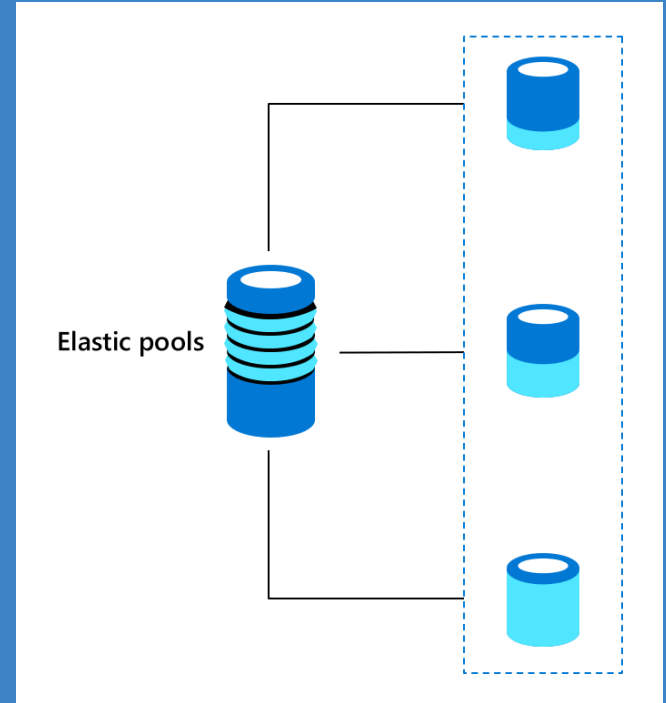
→ Deployment options

- Single database is a fully managed, isolated database for modern cloud applications and microservices that need a single reliable data source
- Elastic pool is a set of single databases that share resources such as CPU and memory, and databases can be moved in and out of the pool



Elastic pool

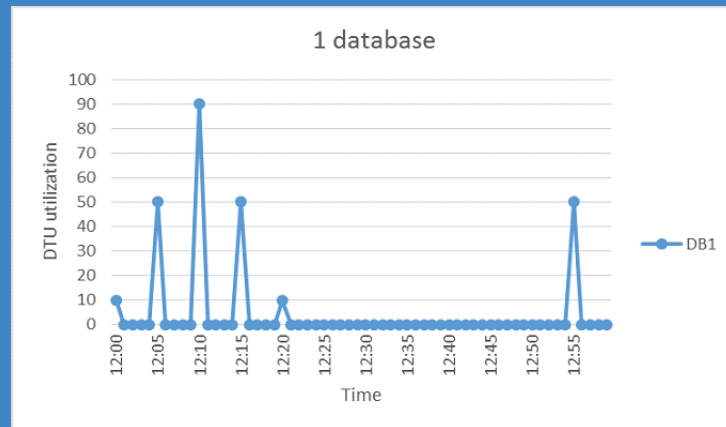
- When is elastic pools the right choice?
 - Multiple databases with unpredictable and intermittent usage patterns can be consolidated into a single server and use elastic pools for better price optimization





Elastic pool

- When is elastic pools the right choice?
 - A database that spends much of its idle time but periodically spikes with activity
 - This utilization pattern is suited for a pool





Elastic pool

- With elastic pools, you don't need to focus on dialing database performance up and down as demand for resources fluctuates
- The pooled databases consume the performance resources of the elastic pool as needed
- Pooled databases consume but don't exceed the limits of the pool, so your cost remains predictable even if individual database usage doesn't



Elastic pool

- The best size for a pool depends on the aggregate resources needed for all databases in the pool:
 - Maximum compute resources utilized by all databases in the pool (eDTUs or vCores)
 - Maximum storage bytes utilized by all databases in the pool
- How do I choose the correct pool size?



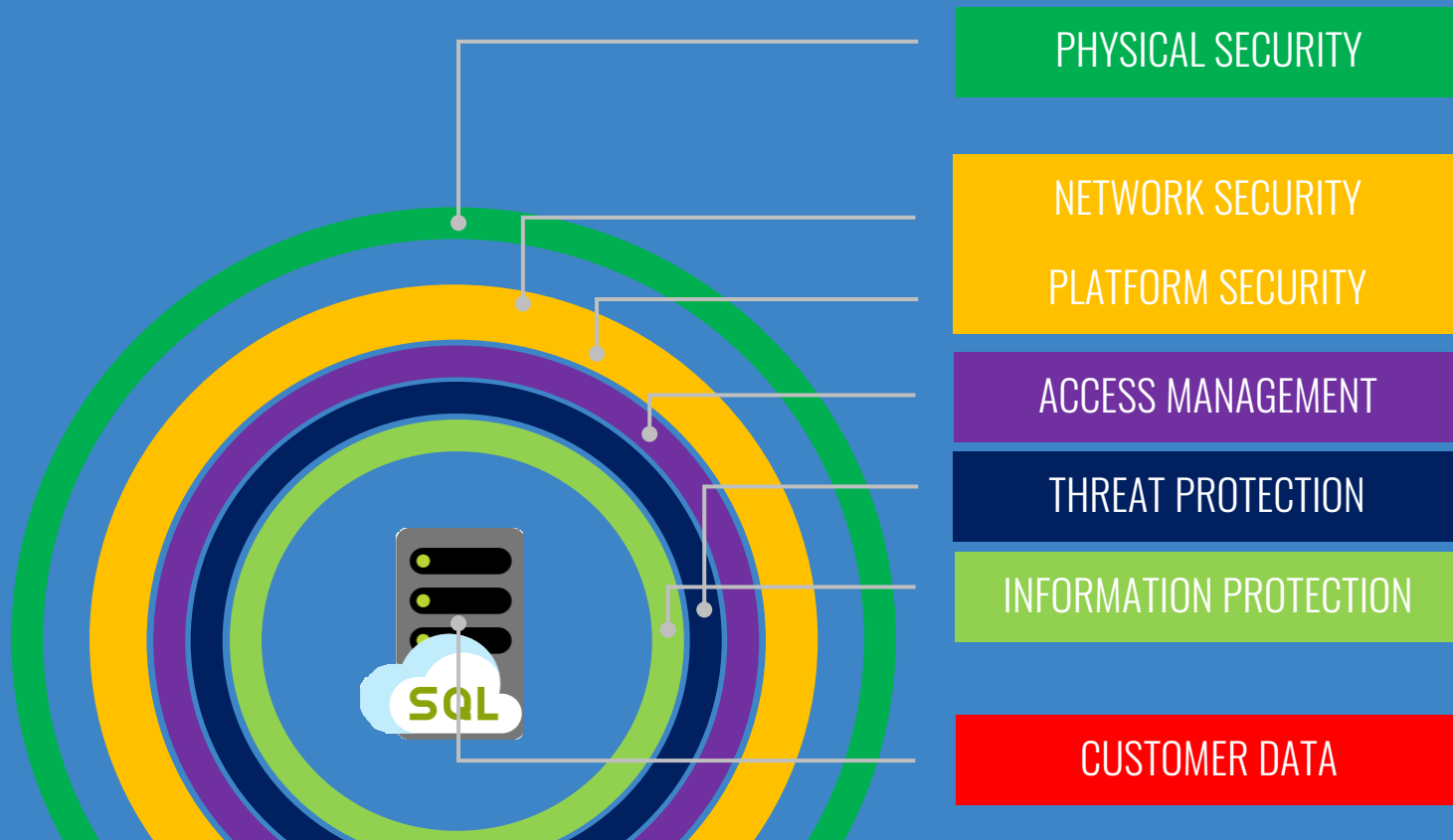
Demo



Azure SQL networking



Azure SQL security capabilities



PHYSICAL SECURITY

NETWORK SECURITY

PLATFORM SECURITY

ACCESS MANAGEMENT

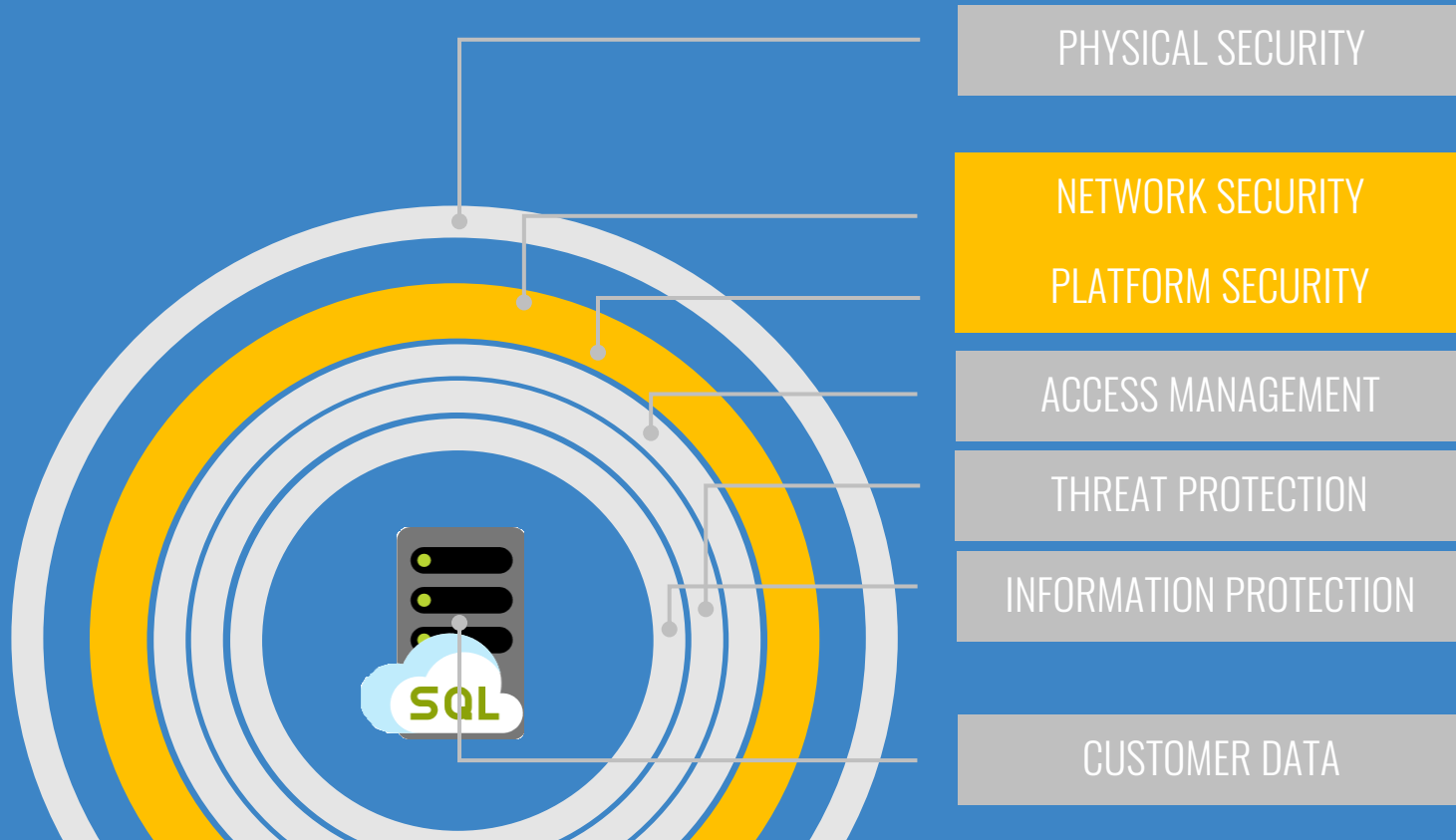
THREAT PROTECTION

INFORMATION PROTECTION

CUSTOMER DATA



Azure SQL networking security



PHYSICAL SECURITY

NETWORK SECURITY

PLATFORM SECURITY

ACCESS MANAGEMENT

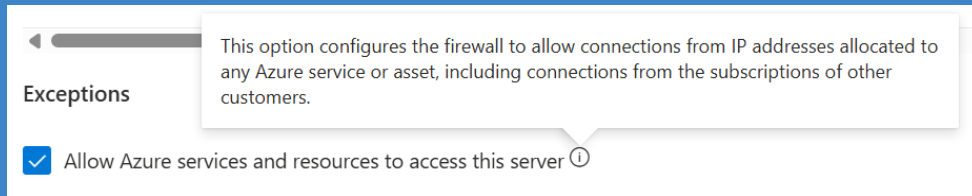
THREAT PROTECTION

INFORMATION PROTECTION

CUSTOMER DATA



Don't allow Azure services (IMHO)

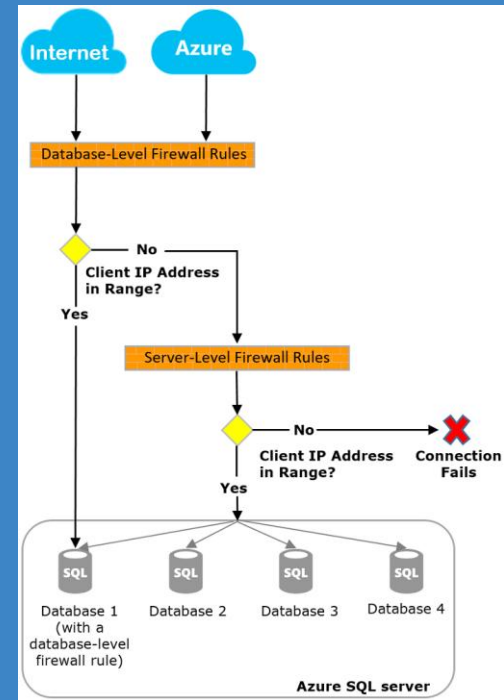


- Allow Azure services means:
 - It's not limited to your services
 - Any Azure service can bypass your firewall
- Prefer virtual networks instead
- Authentication and networking are separate configurations, but both affect database connectivity



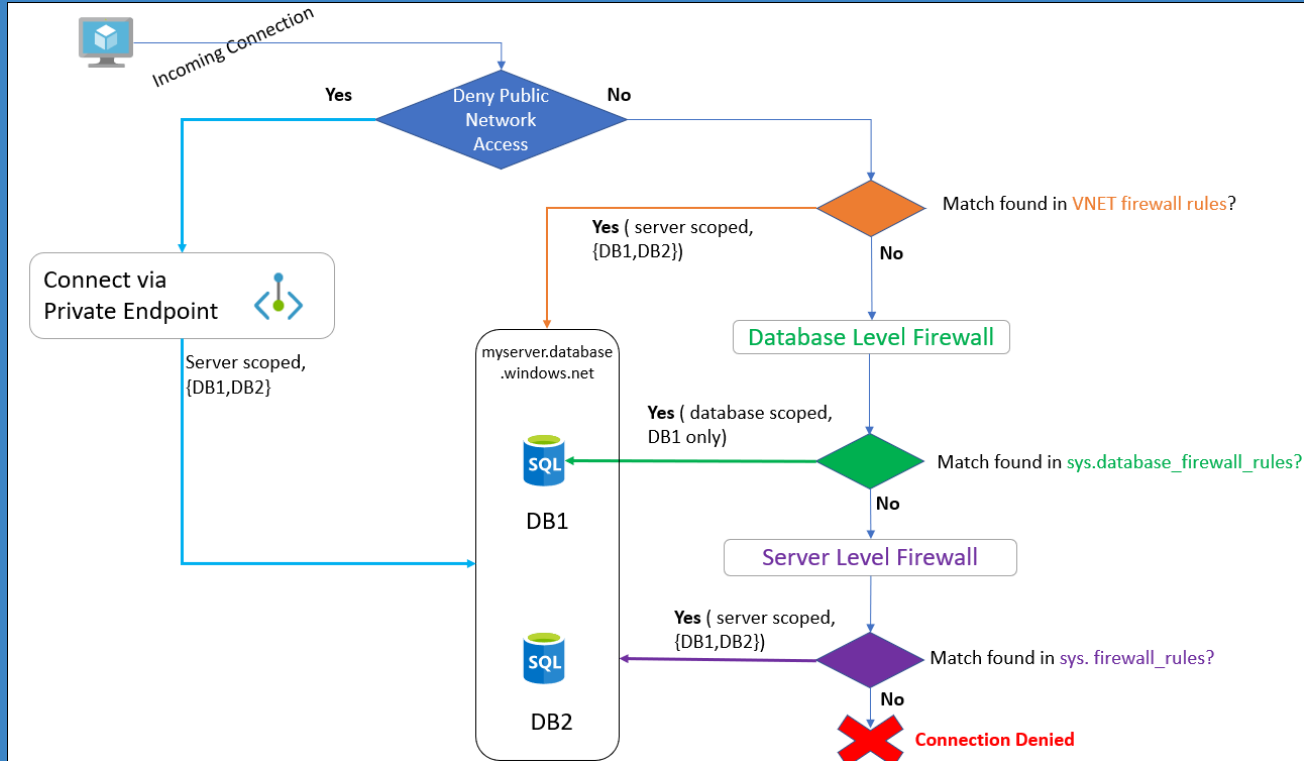
Server and database-level firewall

- Internet and Azure connections must pass through the firewall first
- Server-level rules allow access to all databases on the server
- Rules are stored in the master database
- A server supports up to 256 server-level IP firewall rules





Server and database level firewall





Demo



Connection policy

Connection Policy

Configure how clients communicate with your SQL database server. [Learn more](#)

Connection policy

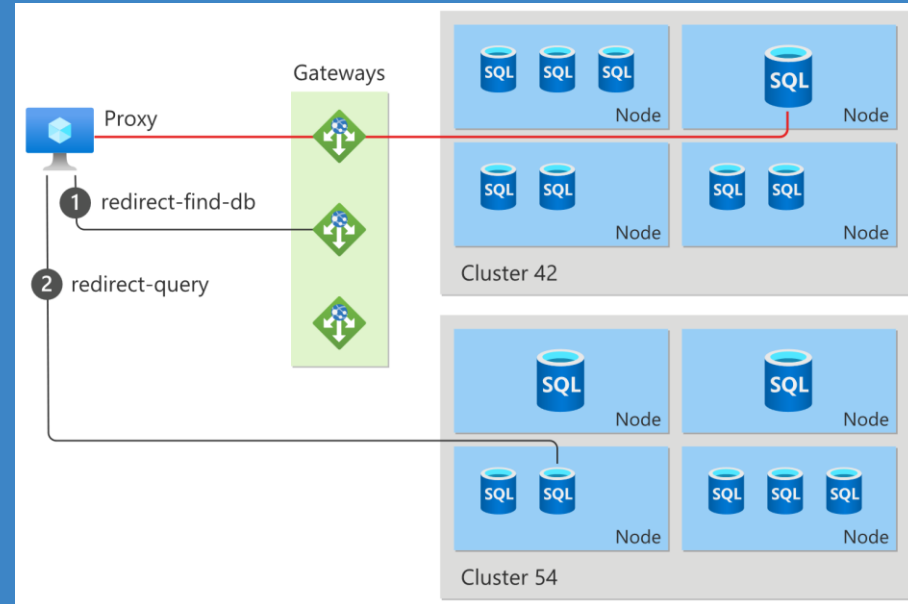
- ☐ Default - Uses Redirect policy for all client connections originating inside of Azure (except Private Endpoint connections) and Proxy for all client connections originating outside Azure
- ☐ Proxy - All connections are proxied via the Azure SQL Database gateways
- ☒ Redirect - Clients establish connections directly to the node hosting the database

- Redirect (recommended): Direct client-to-node connection, lower latency, higher throughput
- Proxy: All traffic goes through the Azure SQL Database gateway, increasing latency and reducing throughput
- Default: Redirect for connections from inside Azure, Proxy for connections from outside Azure



Connectivity architecture

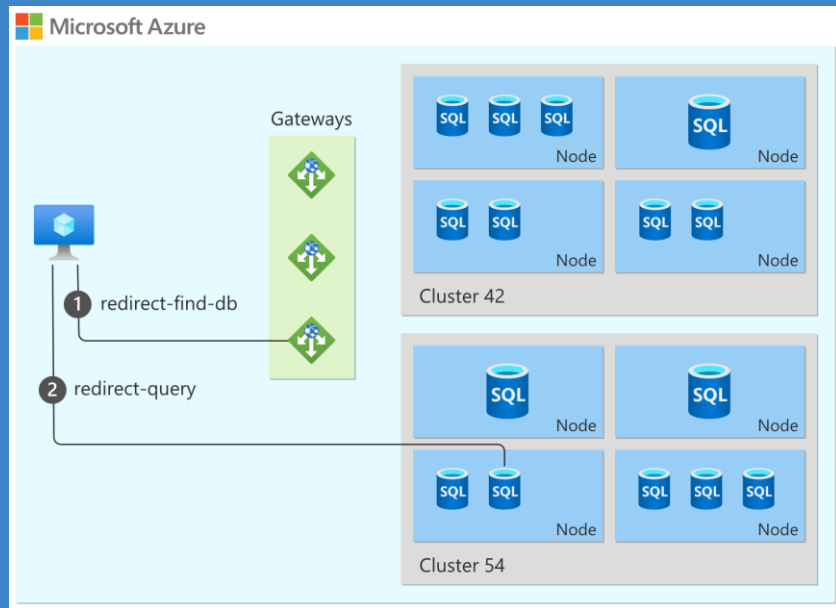
- Clients connect to the public gateway on port 1433
- Gateway redirects or proxies traffic to the correct database cluster
- Cluster routes traffic to the target database





Connectivity from within Azure

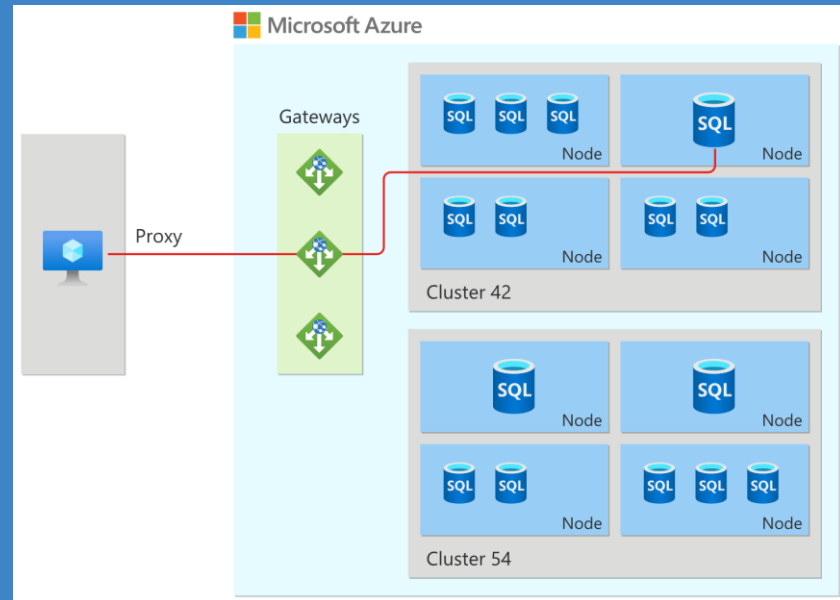
- Inside Azure, redirect is the default
- After TCP connection, the client is redirected from the gateway to the correct database cluster, and traffic then bypasses the gateway





Connectivity from outside of Azure

- Outside Azure, proxy is the default
- The TCP session is established through the Azure SQL Database gateway, and all traffic continues through the gateway





Demo



Backups

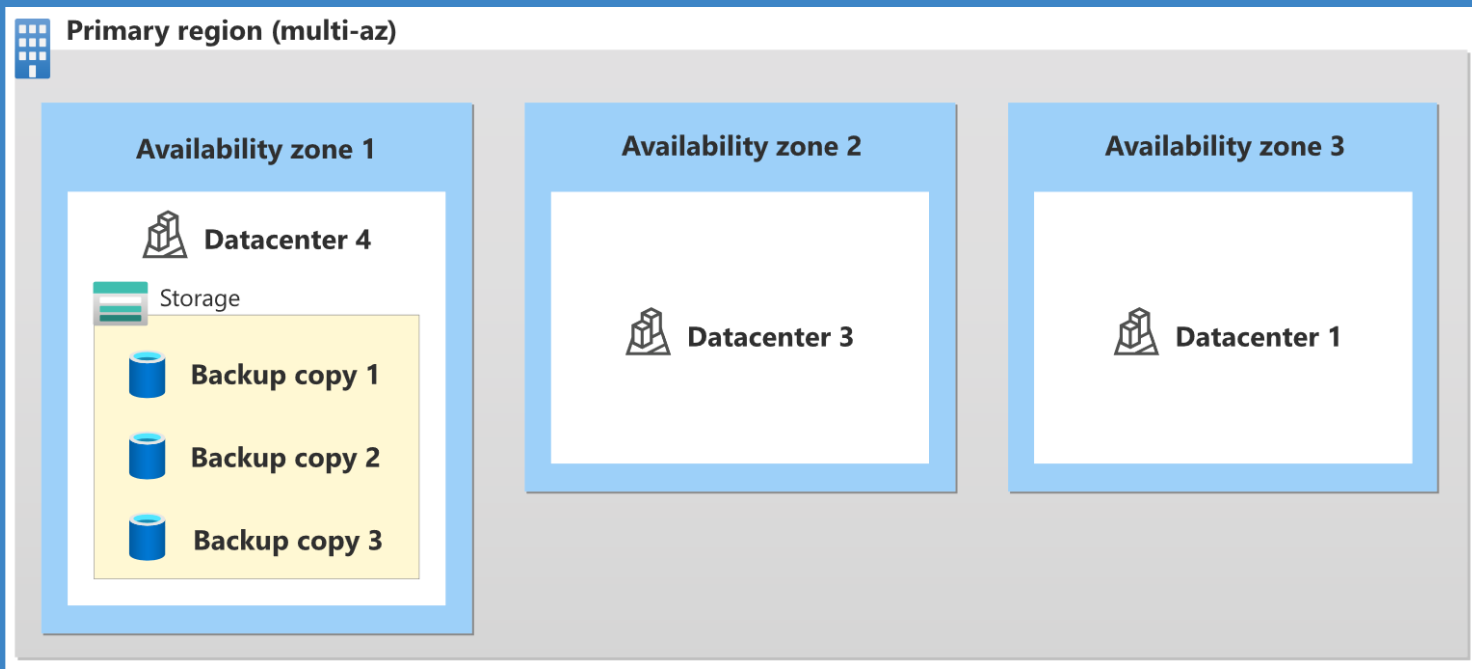


Backups

- Azure SQL Database automatically creates database backups
 - Full backups every week
 - Differential backups every 12 hours
 - Transaction log backups every 5-10 minutes
- Backups are stored in read-access geo-redundant storage replicated to a paired data center

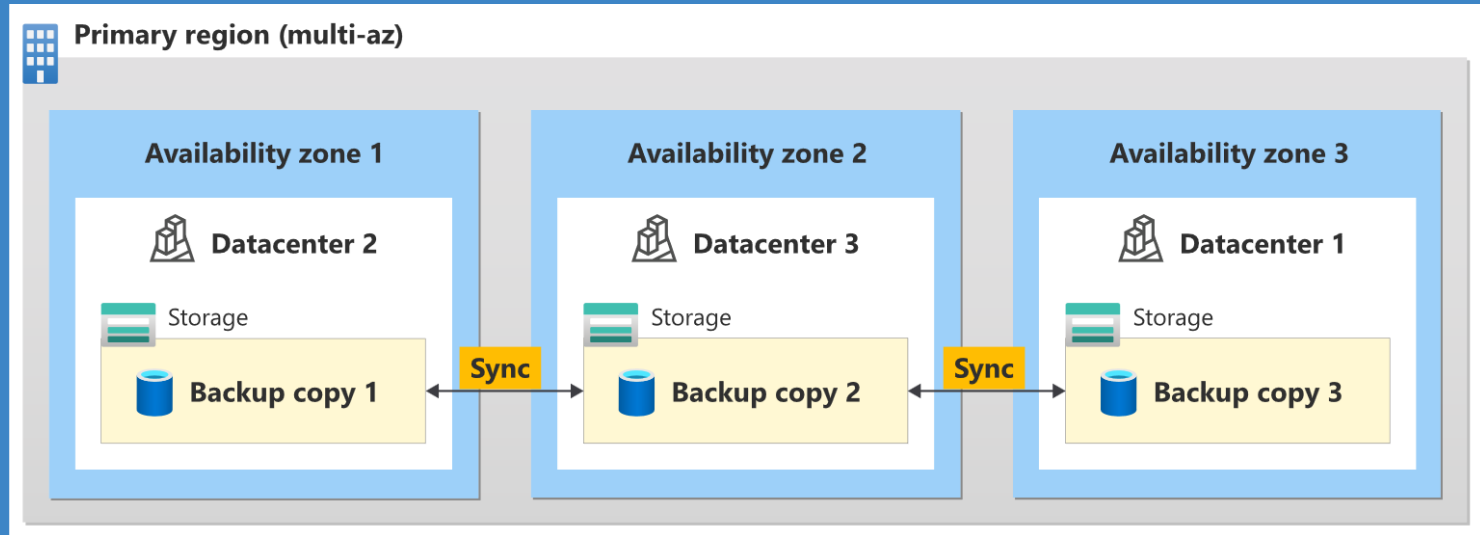


Backups: locally redundant storage (LRS)



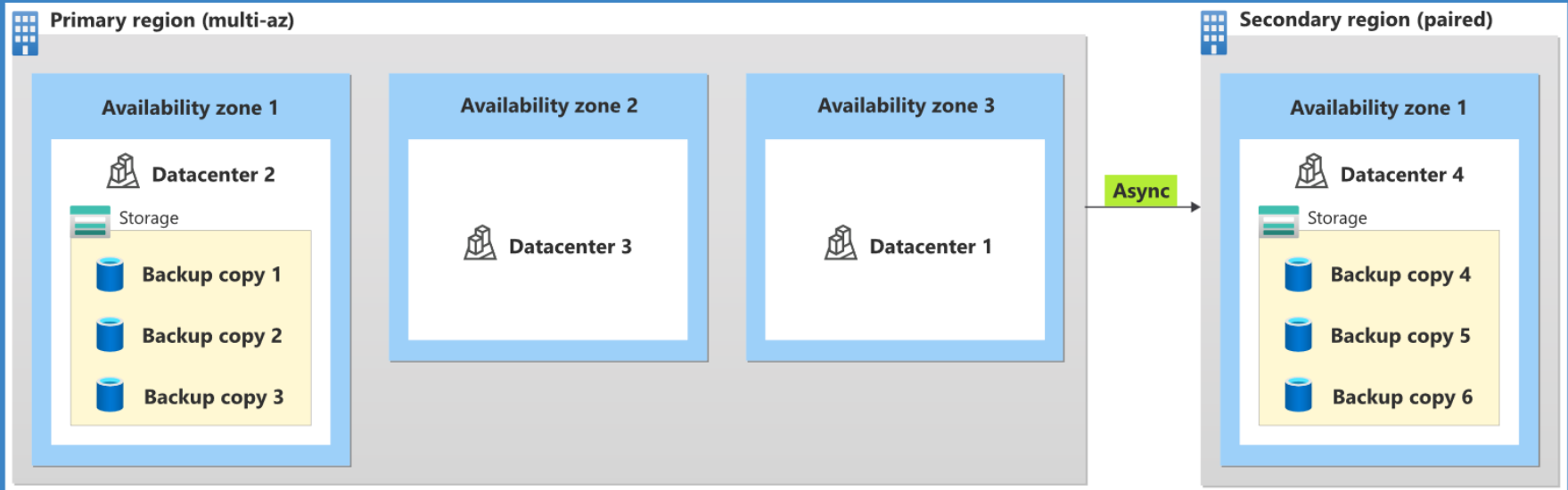


Backups: zone-redundant storage (ZRS)



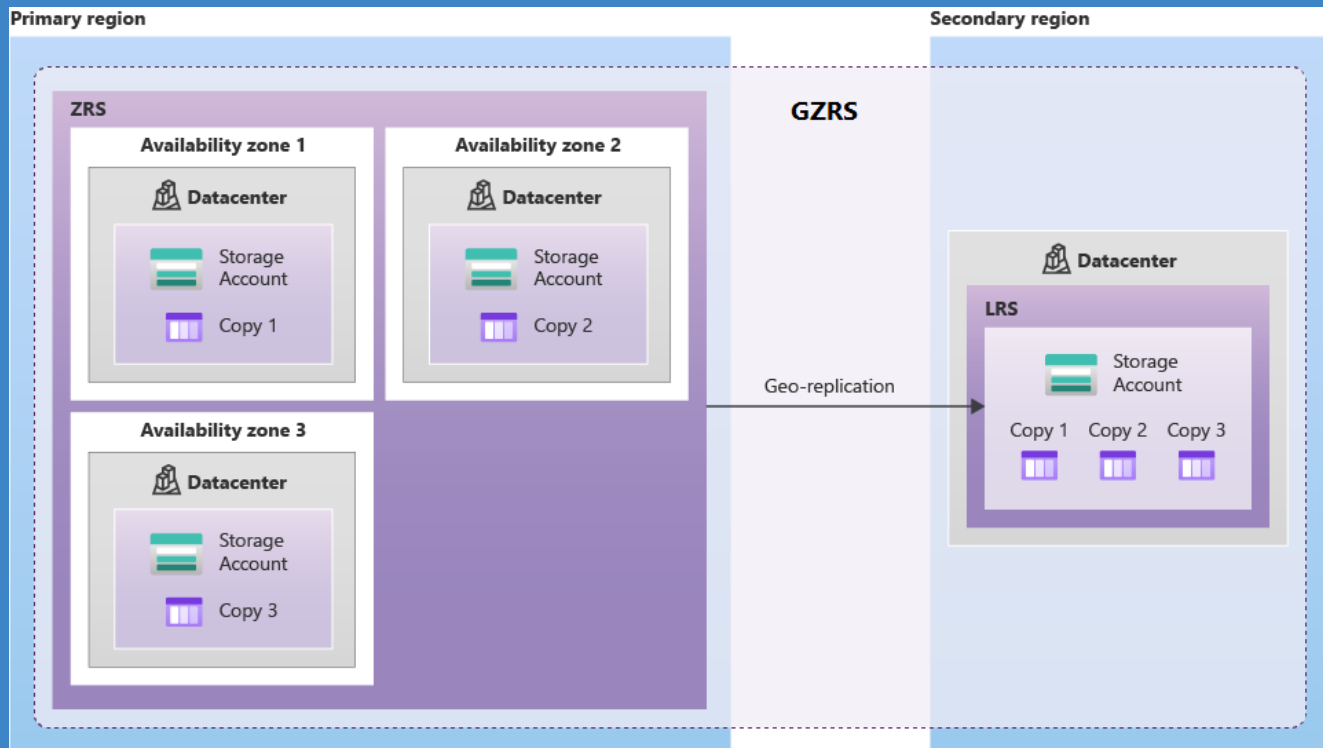


Backups: geo-redundant storage (GRS)





Backups: geo-zone redundant storage (GZRS)





Backups can be used for...

→ Restore

- Existing database to a point-in-time
- Deleted database to the time it was deleted
- Database to another geographical region
- Database from a specific long-term backup
- <https://bit.ly/34Y1IY2>



Backup retention

- All Azure SQL databases have a default backup retention period of 7 days
- Backup retention period can be changed up to 35 days (at 7-days chunks)
- Backups of deleted databases will be kept according to the retention period fixed
- <https://bit.ly/2SpEOq0>



Backup long-term retention (LTR)

- If you want to keep your backups for a period of time longer than the maximum allowed, you can evaluate the long-term backup retention service
- By using long-term retention, you can store specific backups up to 10 years
- <https://bit.ly/2E0PVkr>



Maintenance activities



Maintenance activities

- The key maintenance activities for database solutions implemented on Azure SQL are:
 - Query tuning
 - Index maintenance
 - Statistics update



Methods for maintenance activities

- The typical methods to do maintenance activities are:
 - Database Maintenance Plans
 - Ola Hallengren's Maintenance Solution Suite
 - Use of DBATools with PowerShell
- On-premise maintenance activities are scheduled with:
 - SQL Agent
 - Windows Scheduled Task
 - Cron in Linux environment
 - Third parties' tools



Methods for maintenance activities

- In Azure SQL Database we have several methods to do maintenance operations:
 - Linked Server
 - Database Maintenance Plan
 - Azure Automation Services
 - Azure SQL Database Elastic Jobs
- It is advised to use the stored procedures of the suite SQL Server Maintenance Solution done by Ola Hallengren
 - <https://ola.hallengren.com>



Automatic index compaction (preview)

- It helps you to reduce the consumption of:
 - Storage space
 - Disk I/O
 - CPU
 - Memory
- It improves workload performance without investing time and effort into index maintenance jobs
- It is disabled by default

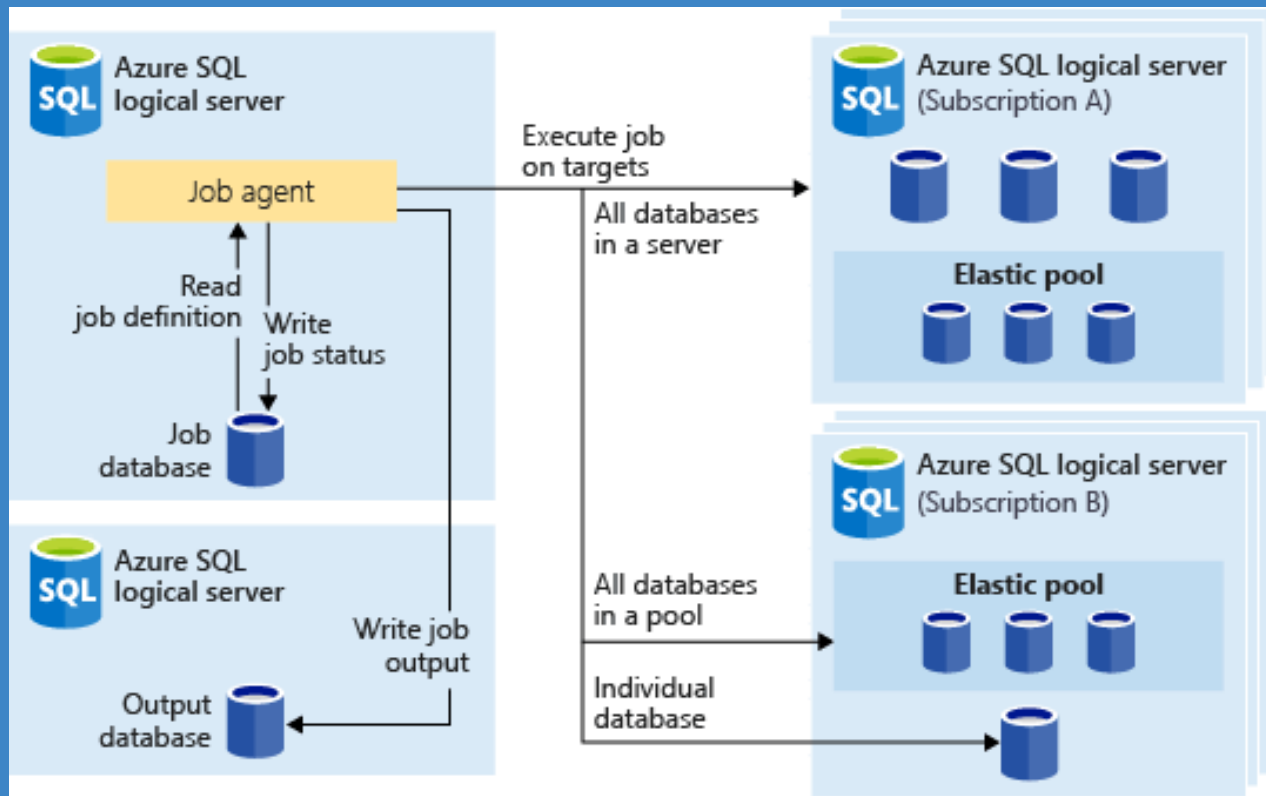


Automatic index compaction (preview)

- Benefits and considerations:
 - You don't need to set up and maintain index maintenance jobs
 - It avoids high resource consumption by index maintenance jobs
 - It reduces the space growth that can occur when data in the database is modified
 - It improves query performance:
 - Reading a compact index reads fewer pages
 - A compact index is more likely to be selected to improve a query plan
- Similar to index reorganization, the compaction process acquires short-term exclusive (X) page locks



Azure SQL Database Elastic Jobs





Azure SQL Database Elastic Jobs

- Azure SQL Database Elastic Jobs is the most complete solution for automation of scheduled activities to be performed on Azure SQL Database
- It's in general availability
- It runs jobs whose definitions are stored in a “job database”
- Jobs are T-SQL scripts scheduled to act for a group of Azure SQL databases



When to use Elastic Jobs

- Automate management tasks and schedule them
- Configure jobs to execute across a collection of databases on a recurring basis, such as during off-peak hours:
 - This is a significant differentiation from SQL Agent, which cannot dynamically enumerate the target databases, especially in SaaS customer scenarios where databases are added/deleted dynamically
- Collect data for reporting
- Data movement



The job database

- Azure SQL Database Elastic Jobs requires an empty database with service level of “S1” or higher hosted in an Azure SQL Database:
 - This DB is also known as “job database” and it’s like the MSDB for on-premises instance
 - The “job database” is used for defining jobs and tracking the status and history of job executions

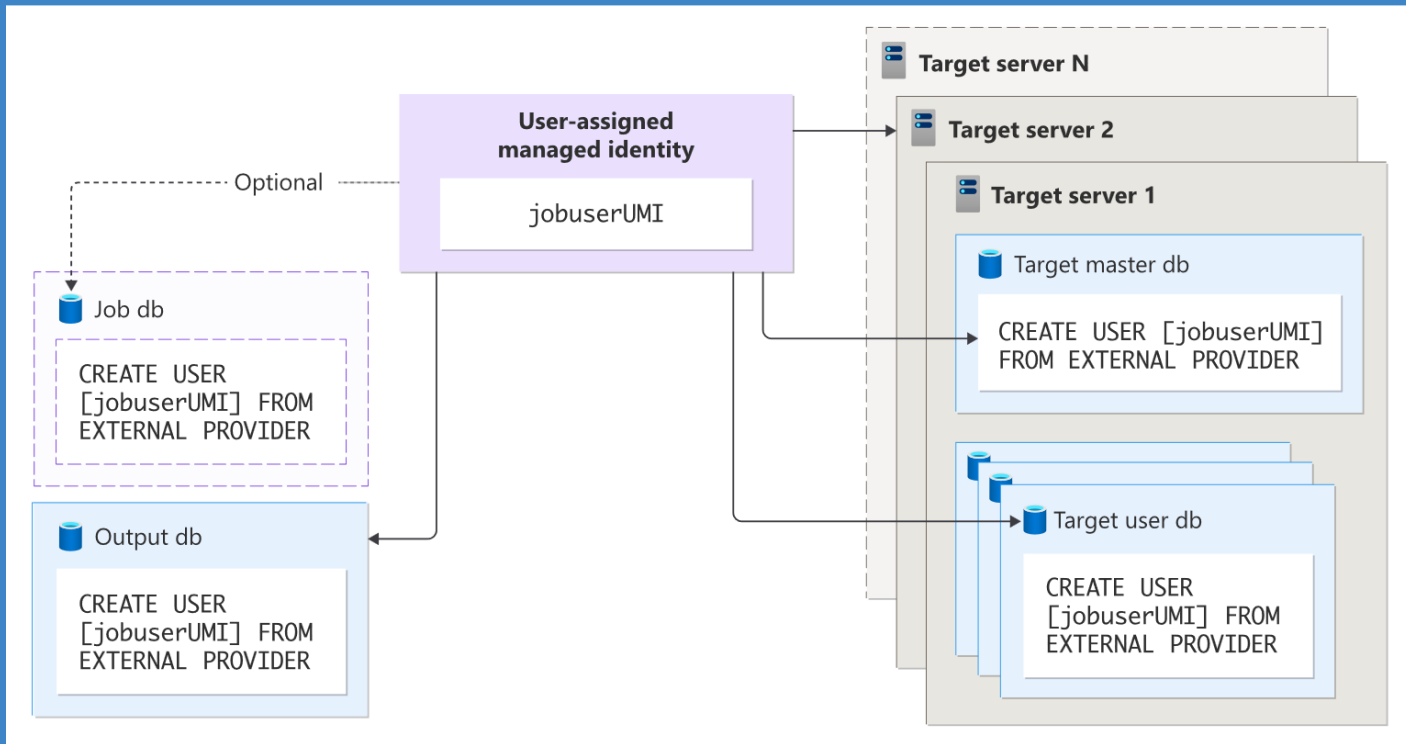


Elastic jobs authentication

- The elastic job agent can connect to the server(s)/database(s) specified by the target group via two authentication options:
 - Microsoft Entra authentication with a user-assigned managed identity (Recommended option)
 - Database-scoped credentials (previously, this was the only method of authentication)



Microsoft Entra authentication with UMI





Elastic jobs limitations and availability

- Azure SQL Team is actively working to remove as many limitations as possible
- Anyway, you can refer to the documentation for a list of known limitations, here: <https://aka.ms/EJ-limitations>
- Elastic Jobs is available in all the regions that Azure SQL Database is available



Demo

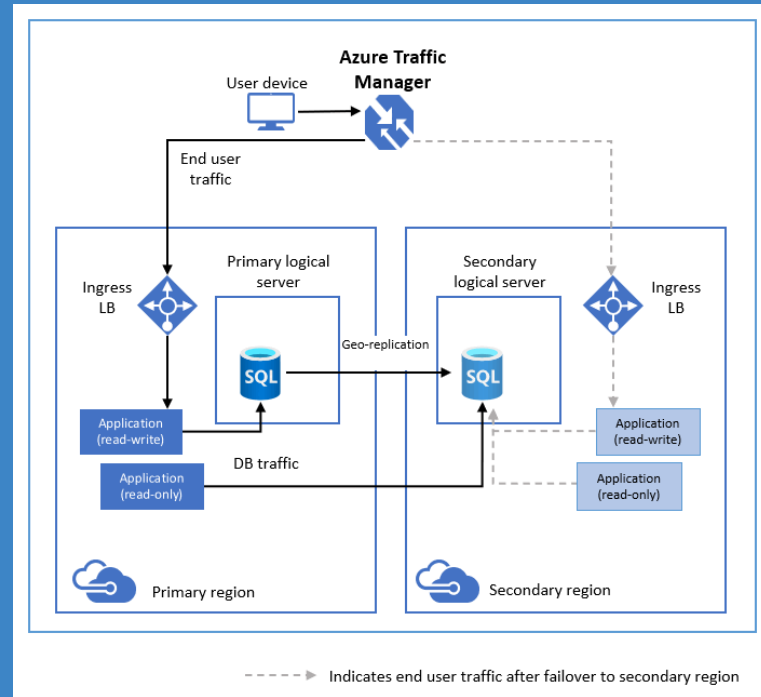


High availability



Active geo-replication

→ Active geo-replication is a business continuity solution for Azure SQL Database that continuously and asynchronously replicates data from a primary database to one or more readable secondary databases





Active geo-replication: Model

- Asynchronous replication based on Always On technology
- Transactions are committed on the primary first
- Changes are then replicated to geo-secondary databases
- Up to 4 readable geo-secondaries per primary database
- Geo-secondaries can offload read-only workloads
- Each geo-secondary can be placed in a different Azure region



Active geo-replication: Failover

- Failover is manual or application-controlled
- Can be initiated through Azure portal, API, PowerShell, or T-SQL
- Failover with no data loss waits for full synchronization
- Forced failover promotes the secondary immediately
- Forced failover can cause data loss
- After failover, the new primary is on a different logical server
- The connection endpoint changes after failover



Active geo-replication: Requirements and BP

- Primary and secondary databases must use the same service tier
- Keep compute size aligned when possible
- Keep credentials, firewall rules, and backup policies in sync
- Use contained database users to simplify failover
- For critical transactions, call [sp_wait_for_database_copy_sync](#) after commit

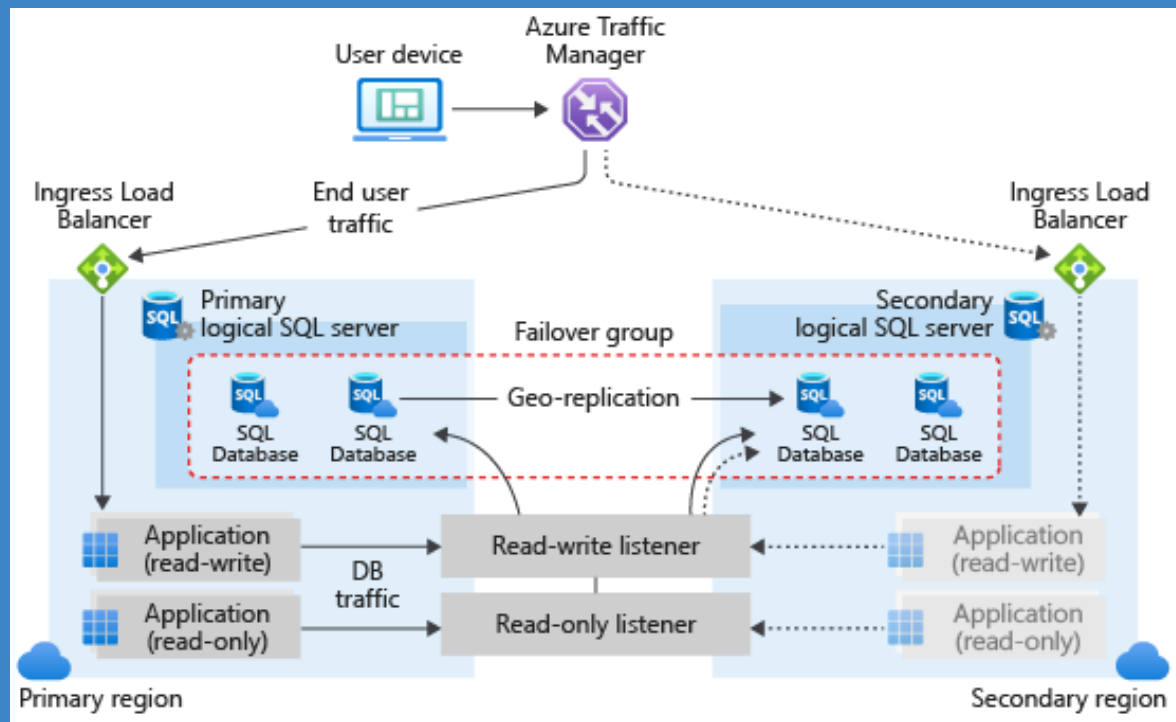


Failover Groups

- A Failover Group lets you manage geo-replication and automatic or coordinated failover for one or more databases, either single databases or databases in an elastic pool, across logical servers in different Azure regions
- It is a business continuity solution that simplifies management compared to active geo-replication, thanks to stable endpoints and group-level failover



Failover Groups





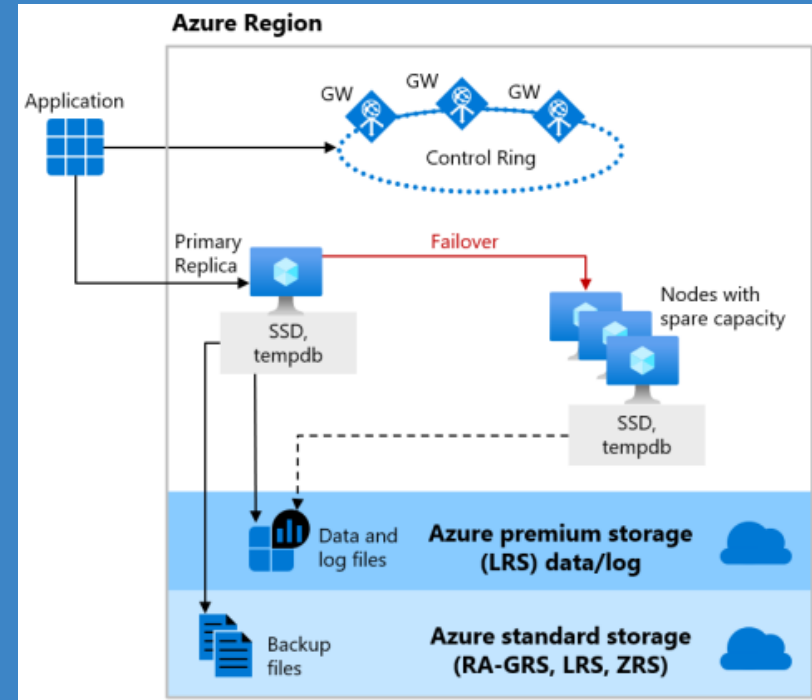
Azure SQL Database availability models

- Built-in availability is part of the platform
- Azure SQL handles patching, upgrades, and failover events
- Local redundancy protects against local failures
- Zone redundancy protects against availability zone outages
- Applications should use retry logic



HA: Basic, Standard, and General Purpose

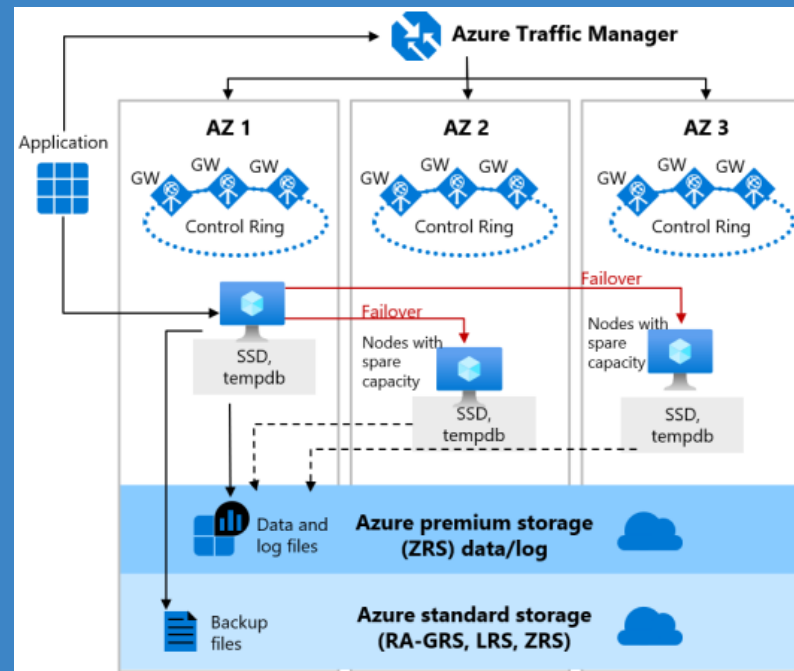
- Remote storage availability model
- Compute and storage are separated
- Database files are stored in Azure Blob Storage
- Compute layer is stateless and can fail over
- Failover can cause cold cache and temporary performance impact
- Zone redundancy is supported only for General Purpose





HA: General Purpose zone-redundant

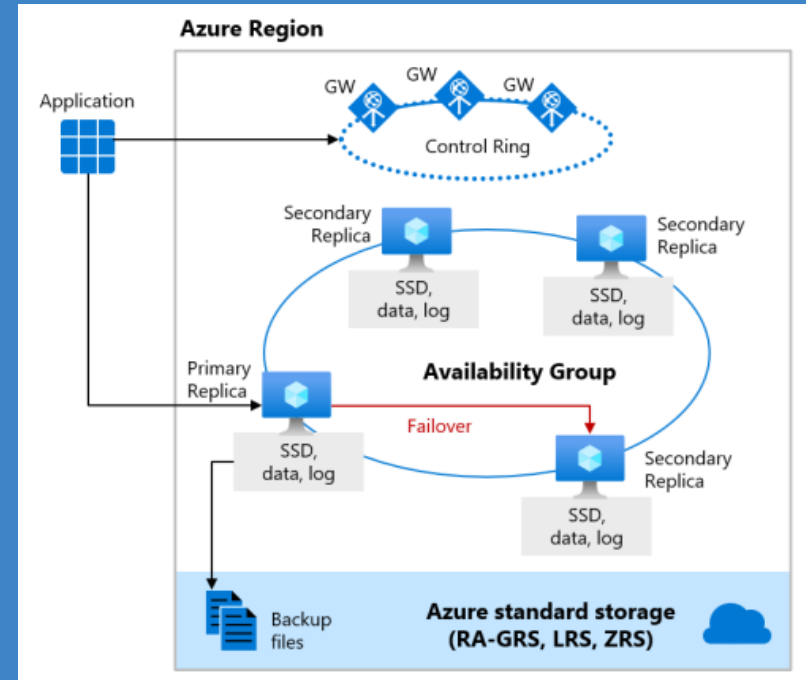
- Like the Basic, Standard model, except:
 - Compute nodes across multiple AZs
 - Azure Storage with ZRS





HA: Premium, Business Critical, Hyperscale

- Local storage availability model
- Compute and storage are placed on the same node
- Uses multiple database engine replicas
 - One primary replica serves read-write workload
 - Secondary replicas are kept synchronized
- Failover redirects connections to a synchronized replica



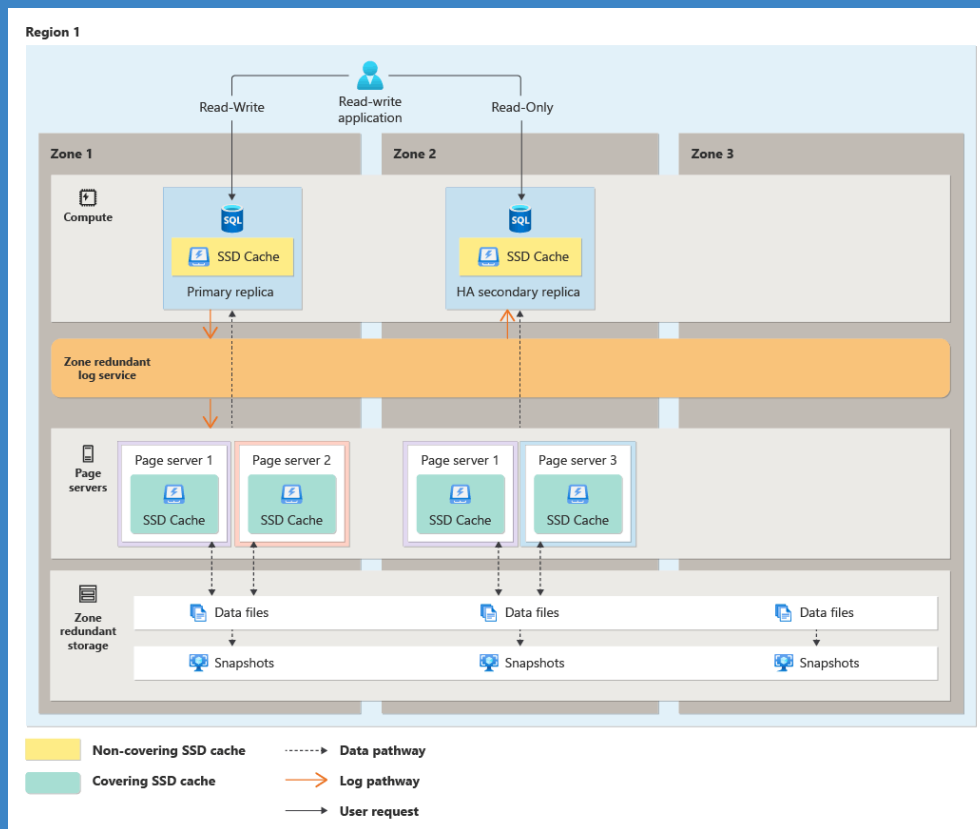


HA: Hyperscale and zone redundancy

- Hyperscale uses a distributed architecture
- Compute, page servers, log service, and storage are separated
- Each layer provides its own redundancy
- Zone redundancy replicates components across availability zones
- Supported for Hyperscale databases and elastic pools
- Zone redundancy must be planned carefully for existing databases



HA: Hyperscale and zone redundancy





Summary (1/2)

- Azure SQL Database is a fully managed PaaS database service in Microsoft Azure, based on the SQL Server engine
- It reduces infrastructure management by handling patching, upgrades, backups, high availability, and platform operations, while still requiring good database design, security, performance tuning, and maintenance practices



Summary (2/2)

- It supports different deployment and purchasing models, from single databases and elastic pools to DTU and vCore models, with service tiers such as General Purpose, Business Critical, and Hyperscale
- It is designed for modern cloud applications that need scalability, availability, security, and operational simplicity



Resources

- Start for free
 - aka.ms/getfreeazuresql
- Azure SQL fundamentals learning path
 - aka.ms/azuresqlfundamentals
- Azure SQL Workshop
 - aka.ms/sqlworkshops
- How to choose tool
 - aka.ms/chooseazuresql



Resources

- Azure SQL documentation
 - aka.ms/azuresqldocs
- Maintenance
 - [Automating Azure SQL Database maintenance tasks \(1° part\)](#)
 - [Automating Azure SQL Database maintenance tasks \(2° part\)](#)
 - [Automation of maintenance activities in Azure SQL Database \(3° Part\)](#)
- [Automatic index compaction \(preview\)](#)
- Azure Database Migration Guides
 - aka.ms/dmguide
- aka.ms/azuresqlbook

An aerial photograph of a historic Italian city, likely Florence, showing a dense cluster of buildings with terracotta roofs, a winding river, and a large square with a prominent building featuring arches.

DELPHIDAY

italian conference

THANK YOU