

 100 XP 

Choose the right Azure SQL Database feature


7 minutes

In our bicycle manufacturing scenario, you've already identified and profiled the databases that you want to migrate to Azure SQL Database. Now, you want to plan the migration, considering data recoverability, disaster recovery, security, and other implementation details.

You'd like to know the tools and features available to support with the migration process to Azure SQL Database.

Benefits of Azure SQL Database

The following summarize the benefits of deploying single and elastic pool databases:

 Expand table

Category	Feature
Backup and recovery	Automatic backup
	Point-in-time restore
	Backup retention 7 days+
	Long-term backup retention stores backups for up to 10 years
High availability	99.99% availability guarantee
	Built-in availability with three secondary replicas


Category	Feature
Disaster recovery	Zone redundancy via Azure availability zones
	Geo-restore of database backups
	Active-geo replication between Azure regions
Service scalability	Dynamic scale-up and scale-down
	Scale out with multiple shards
	Share compute resources between databases using elastic pools
Security	Support for Microsoft Entra authentication
	Cloud-only security features such as Advanced Threat Protection
	Transparent data encryption (TDE) enabled by default
	Support for dynamic and static data masking, row-level security, and Always Encrypted
	Firewall allowlist
Licensing	DTU purchasing model for predictive costing
	vCore purchasing model, enabling storage to be scaled independently of compute
	Combine the vCore purchasing model with Azure Hybrid Benefit for SQL Server to realize cost savings of up to 30 percent

**Tip**

To review the benefits of migrating to Azure SQL Database and the features available, please refer to [Deploy PaaS solutions with Azure SQL](#) module.

Exclusive features of Azure SQL Database

Some features are supported in Azure SQL Database that aren't available in other Azure SQL offerings:

 Expand table

Feature	Definition
Hyperscale	Cloud-native architecture that allows for independently scalable compute and storage, providing greater flexibility and resources than other tiers.
Auto-scale	With serverless compute tier
Automatic tuning (indexes)	This built-in feature automatically identifies and creates indexes that can improve the performance of your workload. It also verifies that query performance has improved and removes unused or duplicate indexes.
Elastic query	Allows you to run T-SQL queries that bridge multiple databases in SQL Database. This feature is useful for applications using three- and four-part names that can't be changed.
Elastic jobs	The elastic job feature is the SQL Server Agent replacement for Azure SQL Database. To some extent, elastic job is equivalent to the Multi Server Administration feature available on SQL Server instance.
Query Performance Insights (QPI)	This tool helps find the queries to optimize to improve overall workload performance and efficiently use the resource that you're paying for.

Important

To understand additional feature differences between SQL Database, SQL Server, and Azure SQL Managed Instance, as well as the differences among different Azure SQL Database

options, see [SQL Database features](#).

Migration options supported

There are two modes of migration to Azure SQL Database: **Online** and **Offline**. The online mode has minimal or no downtime, while the offline mode experiences downtime during the migration process.

 Expand table

Tool	Migration mode
Azure Database Migration Service	Offline
Transactional replication	Online
Azure Migrate	Offline
Import Export Wizard/BACPAC	Offline
Bulk copy (bcp utility)	Offline
Azure Data Factory	Offline
Data Migration Assistant (DMA)	Offline

* Can have a higher performance impact, depending on the workload.

Note

While the Database Migration Assistant is a useful tool available, we recommend that you use the [Azure Database Migration Service](#) for large migrations and enhanced overall experience.

Migration performance

Consider the following recommendations when migrating to Azure SQL Database:

- Monitor data file I/O and latency on the source, and mitigate any bottlenecks.
- Scale up the target Azure SQL database to Business Critical Gen5 8 vCore or use the Hyperscale service tier to minimize latency for log files.
- Ensure that your network bandwidth can accommodate the maximum log ingestion rate.
- Choose the highest service tier and compute size for maximum transfer performance, and scale down after migration.
- Minimize the distance between BACPAC files and the destination data center.
- Disable auto update and auto create statistics during migration.
- Partition tables and indexes, drop indexed views, and recreate them after migration.
- Consider migrating rarely queried historical data to a separate database in Azure SQL Database, and query it using elastic queries.

Retry application connections

When migrating to Azure SQL Database, it's important to anticipate occasional transient failures when connecting to the database resource, and implement a proper retry logic method. Setting a maximum number of retries before the program terminates is also important.

We recommend waiting for 5 seconds at a minimum on your first retry. Each subsequential retry should increase the delay exponentially, up to a maximum of 60 seconds.

ⓘ Note

If a SELECT statement fails with a transient error for SQL Database, don't directly retry it. Instead, retry the SELECT statement in a new connection.

To learn more about the connection retry principals, see [Troubleshoot transient connection errors in SQL Database and SQL Managed Instance](#).

Next unit: Use Azure SQL migration extension to migrate to Azure SQL Database

< Previous

Next >