



Azure Data Academy

Azure Database for MySQL – Flexible Server

<https://aka.ms/ada>

Azure Data Academy

<https://aka.ms/ada>

- Data-centric presentations and hands-on material designed to educate and accelerate Microsoft Partner solution and practice development
 - SQL Server, MySQL, PostgreSQL, Cosmos DB
- Similar to our Modern Analytics Academy series: <https://aka.ms/maa>
- Find other great content in the repo: <https://aka.ms/PartnerReadiness>
- Feedback & Contributions encouraged:
 - Through the main content page: <https://aka.ms/ada>
 - Through our feedback form: <https://aka.ms/ada-feedback>



Migration Options

Azure Database for MySQL – Flexible Server

June 2022

Azure Database for MySQL deployment options



Flexible Server

Maximum control for your databases with more cost optimization controls

Example use cases

- Application developments requiring more control and customizations
- Mission-critical apps needing high availability and fine-grained maintenance scheduling
- Applications requiring variable compute capacity to optimize costs

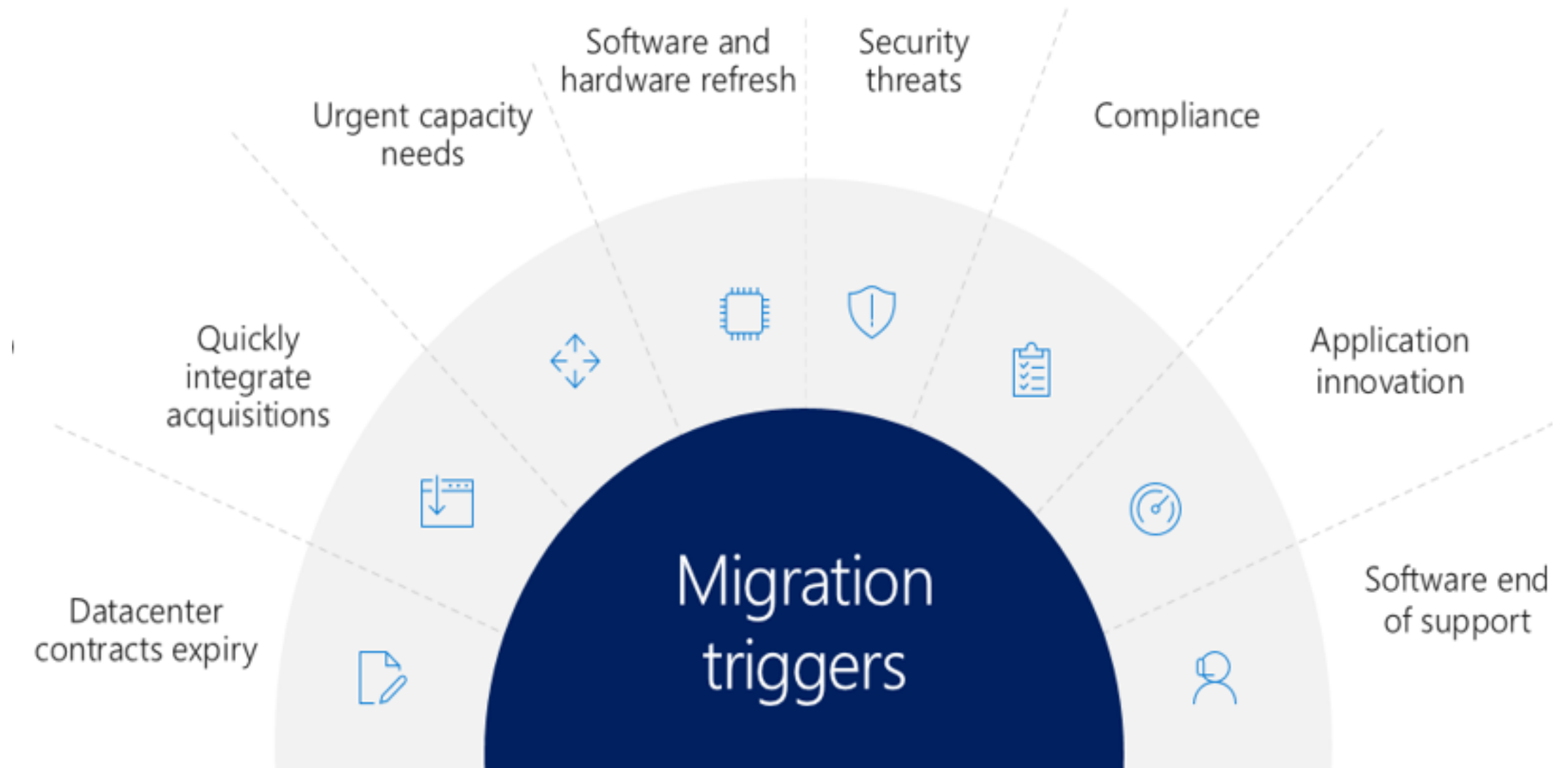
Single Server

Fully-managed, MySQL database service with built-in HA

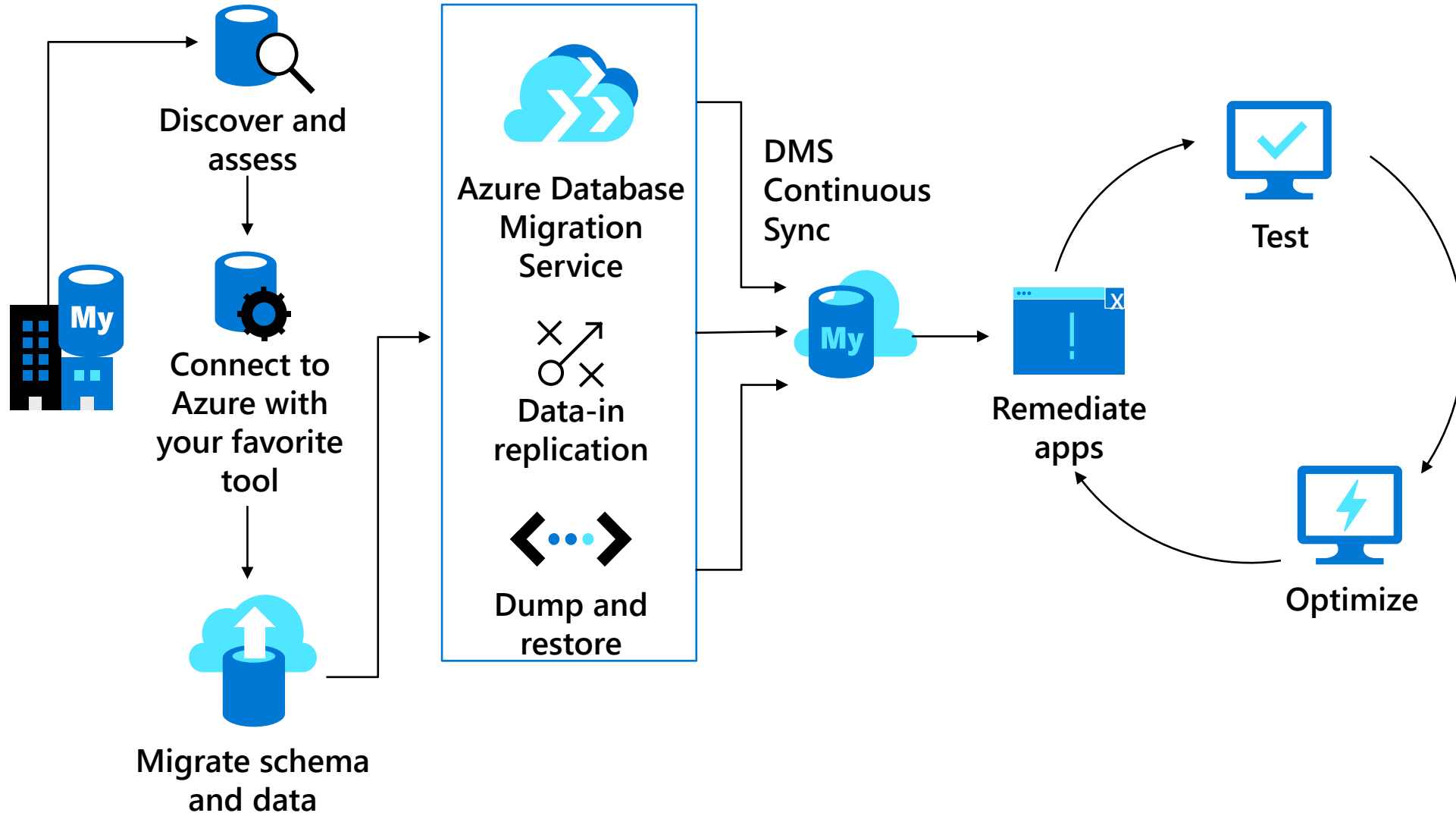
Example use cases

- Cloud native applications designed to handle automated patching
- Online web applications with minimal requirements for database customizations
- Applications without zonal redundancy requirements

What's driving migrations?



MySQL to MySQL migration options



Migration matrix for various flavors

Migration target	On-premises to Azure	Single Server to Flexible Server	In-place upgrades (Single Server & Flexible Server)
Azure DB for MySQL	<ol style="list-style-type: none">1. Dump/restore2. MyDumper/MyLoader3. Data-in replication4. Azure DMS	<ol style="list-style-type: none">1. Dump/restore2. MyDumper/MyLoader	<ol style="list-style-type: none">1. Dump/restore2. MyDumper/MyLoader3. In-place upgrades within the Azure MySQL platform (Single Server: private preview for 5.6 to 5.7)

Azure Database for MySQL: Migration

MySQL Dump and Restore

- Use mysqldump and mysqlpump to dump and load databases into Azure Database for MySQL.
- Requires downtime for the duration of the migration
- Create and set up MySQL with the recommended pricing tier and server parameters for migrations
- For more details, see:
 - [MySQL dump and restore](#)
 - [Preparing target Azure MySQL for fast data loads](#)

MyDumper and MyLoader

- MyDumper and MyLoader support multi-threaded data export and import
- Requires downtime for the duration of the migration
- You can configure the MyDumper parameters to speed up the export process
- Create and set up MySQL with recommended pricing tier and server parameters for migrations
- For more details, see:
 - [MyDumper and MyLoader](#)
 - [Preparing target Azure MySQL for fast data loads](#)

Data-in replication

- This requires a short downtime at the start when creating a dump and then another short downtime to stop writes to the source server, cutover to new target, and then resume writes.
- Create and set up MySQL with the recommended pricing tier and server parameters for migrations
- For more details, see:
 - [Data-in Replication](#)
 - [Preparing target Azure MySQL for fast data loads](#)
 - [Configure Data-in replication - Azure Database for MySQL Flexible Server](#)

[Select the right tools for migration to Azure Database for MySQL
Mysqldump vs Mysqlpump vs Mydumper \(wordpress.com\)](#)

Preparing the target Azure Database for MySQL instance fast data loads

- **max_allowed_packet** – set to 1073741824 (i.e. 1GB) to prevent any overflow issue due to long rows.
- **slow_query_log** – set to OFF to turn off the slow query log. This will eliminate the overhead caused by slow query logging during data loads.
- **query_store_capture_mode** – set to NONE to turn off the Query Store. This will eliminate the overhead caused by sampling activities by Query Store.
- **innodb_buffer_pool_size** – Scale up the server to 32 vCore Memory Optimized SKU from the Pricing tier of the portal during migration to increase the innodb_buffer_pool_size. Innodb_buffer_pool_size can only be increased by scaling up compute for Azure Database for MySQL server.
- **innodb_io_capacity & innodb_io_capacity_max** - Change to 9000 from the Server parameters in Azure portal to improve the IO utilization to optimize for migration speed.
- **innodb_write_io_threads & innodb_write_io_threads** - Change to 4 from the Server parameters in Azure portal to improve the speed of migration.
- **Scale up Storage tier** – The IOPs for Azure Database for MySQL server increases progressively with the increase in storage tier. For faster loads, you may want to increase the storage tier to increase the IOPs provisioned. Please do remember the storage can only be scaled up, not down.

MyDumper/MyLoader and Data-in replication

MyDumper/MyLoader

-- Dump all the databases except sys, mysql, performance schema and information schema

```
sudo mydumper --host=sourcemysql.mysql.database.azure.com --user=<username> --password=<yourpassword> --  
outputdir=/home/azure --rows=10000 --compress --build-empty-files --compress-protocol --regex  
'^(?!(mysql\.|information_schema\.|performance_schema\.|sys\.))' -L mydumpersrr-logs.txt
```

-- Restore

```
sudo myloader --host=azuretarget.mysql.database.azure.com --user=<username> --password=<yourpassword> --  
directory=/home/azure --queries-per-transaction=10 --overwrite-tables --compress-protocol --threads=4 --verbose=3 -e  
2>ssrloader-logs.txt
```

Data-in replication

```
CALL mysql.az_replication_change_master('sourcemysql.mysql.database.azure.com','syncssuser', 'yourpassword', 3306,  
'mysql-bin.000004', 154, '');
```

```
CALL mysql.az_replication_start
```

AWS to Azure MySQL Migration

From MySQL RDS on VPC to MySQL Flexible Server in a virtual network.

Steps for offline migrations to spin up a flexible server.

1. Backup MySQL RDS on AWS VM.
2. Copy backups to an Azure VM.
3. Restore to Flexible Server.

Steps for setting up active replication

1. Establish Site-to-Site connectivity.
2. Configure Data-in replication from MySQL flex.

