MigrateMyDatabase
AN RKPC SERVICE

MIGRATION REPORT

MmdbShop → Azure SQL Managed Instance

Prepared for Example Company

Source	**MmdbShop** – SQL Server 2022 · compatibility level 160
Target	**Azure SQL Managed Instance** (Microsoft Azure)
Prepared for	Example Company
Prepared by	Ralph Klos Projects & Consultancy BV
Date	24/11/2026
Reference	MMDB-20261124-7V49

CONFIDENTIAL

Ralph Klos Projects & Consultancy BV · Gaspeldoornstraat 20, 2040 Zandvliet (Antwerpen), België · VAT BE1044614873 · migratemydatabase.eu

Contents

1	Executive summary
2	Recommended route & migration plan
3	Findings overview
4	Findings and solutions
5	About this report
Appendix A	Scripts
Appendix B	Inventory

1 Executive summary

Possible after resolving 1 important point(s); no blocking points.

KEY FIGURES

89/100	0	1	4	2	5–30
SCORE	BLOCKER	HIGH	MEDIUM	LOW	DOWNTIME (MIN)

KEY FACTS

Source	SQL Server 2022 · Data size 8 MB 7 tables, 1 views, 3 procedures, 1 functions, 2 triggers
Target	Azure SQL Managed Instance (Microsoft Azure) Maximum database size 32 TB · Version support always the newest engine
Recommended route	Log Replay Service
Accepted downtime	Minutes
Expected downtime for the switch-over	Typically 5 to 30 minutes at the switch. The initial copy (about 0.1 hours) runs beforehand while the source stays in use.

MAIN POINTS OF ATTENTION

1	HIGH	OLE Automation (sp_OACreate) is used (1 object(s))
2	MEDIUM	Distributed transactions (MSDTC) (1 object(s))
3	MEDIUM	Code changes server settings (sp_configure, trace flags) (1 object(s))
4	MEDIUM	Custom error messages (sp_addmessage) (1 object(s))
5	MEDIUM	Table without a primary key (1 object(s))

2 Recommended route & migration plan

Accepted downtime	Minutes (copy while the old database stays in use, then a short switch)
Recommended route	**Log Replay Service** · Minutes The managed instance keeps restoring full, differential and log backups from Blob Storage; the switch takes minutes.
Expected downtime for the switch-over	Typically 5 to 30 minutes at the switch. The initial copy (about 0.1 hours) runs beforehand while the source stays in use.
Other possible routes	**Azure Database Migration Service (online)** · Minutes Restores backups to the managed instance and keeps applying log backups until you start the switch. **Transactional replication** · Minutes The target is a subscriber of a publication on the old server and stays in sync; the switch takes minutes. Every replicated table needs a primary key. **Managed Instance link** · Practically none A distributed availability group to the managed instance keeps it in sync; the switch is a failover. Needs a supported SQL Server version and update.
What in your database makes the online routes harder	1 table(s) without a primary key: replication and change data capture cannot apply updates and deletes to them reliably dbo.AuditLog
Other routes for this target	Native backup to URL and restore · A maintenance window

STEP BY STEP: LOG REPLAY SERVICE

The commands of the recommended route, filled in for this database. Replace the values in <angle brackets>, run every step on a test copy first and time it: that time is your real downtime. Also in the script package (00-runbook.txt).

```
# Runbook: MmdbShop → Azure SQL Managed Instance
# Rehearse every step on a copy first and write down how long each one takes: that is your real downtime.

# ---------- 1. Backups to Azure Blob Storage (one folder per database, WITH CHECKSUM) ----------
BACKUP DATABASE [MmdbShop] TO URL =
N'https://<account>.blob.core.windows.net/<container>/MmdbShop/MmdbShop_full.bak' WITH CHECKSUM,
COMPRESSION;
BACKUP LOG [MmdbShop] TO URL =
N'https://<account>.blob.core.windows.net/<container>/MmdbShop/MmdbShop_log_<n>.trn' WITH CHECKSUM;   --
keep doing this until the switch

# ---------- 2. Start Log Replay Service in continuous mode ----------
az sql midb log-replay start -g <resource group> --mi <managed instance> -n MmdbShop \
    --storage-uri https://<account>.blob.core.windows.net/<container>/MmdbShop \
    --storage-sas "<SAS token>"
az sql midb log-replay show -g <resource group> --mi <managed instance> -n MmdbShop

# ---------- 3. The switch ----------
# Stop the application, take and upload the last log backup, then complete the replay with the name of
that file:
az sql midb log-replay complete -g <resource group> --mi <managed instance> -n MmdbShop \
    --last-backup-name MmdbShop_log_<last>.trn
```

```
# ---------- Validation and switch ----------
# Run the row-count validation (appendix) on the source just before the switch and on the target right
after the load: every table must match.
# Keep the source unchanged and read-only until the go-live is accepted (see the fallback step of the
plan).
```

MIGRATION PLAN

1	**Preparation**	Resolve the blocking and important findings, script server objects (logins, SQL Agent jobs, linked servers), agree on the downtime window. • finding 1: OLE Automation (sp_OACreate) is used • finding 2: Distributed transactions (MSDTC) • finding 3: Code changes server settings (sp_configure, trace flags) • finding 5: Table without a primary key – script A2 • finding 6: Deprecated data type (text, ntext, image) – script A3
2	**Test migration**	Migrate a recent copy to the target, run the upgrade test and the data check there, test the application and the heaviest reports.
3	**Switch-over**	Copy the bulk of the data in advance and keep applying the changes (log backups, replication or change data capture); at the switch stop writes, let the last changes arrive, finish the restore or stop the replication, sync sequences and identity values, switch the connection strings.
4	**Fallback**	Keep the source unchanged and read-only until the go-live is accepted; changes made on the target after the switch are lost on fallback unless replication is set up in the reverse direction. Agree on the point after which falling back is no longer possible.
5	**After the migration**	Check integrity (DBCC CHECKDB), update statistics, compare row counts with the source, monitor performance for two weeks. • finding 4: Custom error messages (sp_addmessage) – script A1 • finding 7: Compatibility level stays at the old value after the restore – script A4

VALIDATION: ROW COUNTS PER TABLE

Run the first query on the source just before the switch and the second on the target right after the load: every table must have the same number of rows. Also in the script package.

On the source

```
SELECT 'dbo.Customer' AS table_name, COUNT(*) AS row_count FROM [dbo].[Customer]
UNION ALL SELECT 'sales.Order' AS table_name, COUNT(*) AS row_count FROM [sales].[Order]
UNION ALL SELECT 'sales.OrderLine' AS table_name, COUNT(*) AS row_count FROM [sales].[OrderLine]
UNION ALL SELECT 'hr.Employee' AS table_name, COUNT(*) AS row_count FROM [hr].[Employee]
UNION ALL SELECT 'hr.EmployeeHistory' AS table_name, COUNT(*) AS row_count FROM [hr].[EmployeeHistory]
UNION ALL SELECT 'dbo.AuditLog' AS table_name, COUNT(*) AS row_count FROM [dbo].[AuditLog]
UNION ALL SELECT 'dbo.ThisTableNameIsDeliberatelyMuchLongerThanSixtyThreeCharactersForPostgres' AS
table_name, COUNT(*) AS row_count FROM [dbo].
[ThisTableNameIsDeliberatelyMuchLongerThanSixtyThreeCharactersForPostgres]
ORDER BY 1;
```

On the target

```
SELECT 'dbo.Customer' AS table_name, COUNT(*) AS row_count FROM [dbo].[Customer]
UNION ALL SELECT 'sales.Order' AS table_name, COUNT(*) AS row_count FROM [sales].[Order]
UNION ALL SELECT 'sales.OrderLine' AS table_name, COUNT(*) AS row_count FROM [sales].[OrderLine]
UNION ALL SELECT 'hr.Employee' AS table_name, COUNT(*) AS row_count FROM [hr].[Employee]
UNION ALL SELECT 'hr.EmployeeHistory' AS table_name, COUNT(*) AS row_count FROM [hr].[EmployeeHistory]
UNION ALL SELECT 'dbo.AuditLog' AS table_name, COUNT(*) AS row_count FROM [dbo].[AuditLog]
UNION ALL SELECT 'dbo.ThisTableNameIsDeliberatelyMuchLongerThanSixtyThreeCharactersForPostgres' AS
table_name, COUNT(*) AS row_count FROM [dbo].
[ThisTableNameIsDeliberatelyMuchLongerThanSixtyThreeCharactersForPostgres]
ORDER BY 1;
```

3 Findings overview

NO.	SEVERITY	FINDING	OBJECTS	PHASE	SCRIPT
1	HIGH	OLE Automation (sp_OACreate) is used	1	Preparation	–
2	MEDIUM	Distributed transactions (MSDTC)	1	Preparation	–
3	MEDIUM	Code changes server settings (sp_configure, trace flags)	1	Preparation	–
4	MEDIUM	Custom error messages (sp_addmessage)	1	After the migration	A1
5	MEDIUM	Table without a primary key	1	Preparation	A2
6	LOW	Deprecated data type (text, ntext, image)	2	Preparation	A3
7	LOW	Compatibility level stays at the old value after the restore	160	After the migration	A4

4 Findings and solutions

1 OLE Automation (sp_OACreate) is used

HIGH 1 object(s)

WHAT IT MEANS

Code uses OLE Automation (sp_OACreate / sp_OAMethod) to start COM objects on the database server – for files, HTTP calls or Office.

IMPACT

Not available on managed platforms, disabled by default on new installs, and a security risk.

HOW TO RESOLVE

1. Find out per procedure what the COM object does.
2. Move that work to the application, SSIS, or a scheduled script/function outside the database (see the alternative on this platform).

AFFECTED OBJECTS

- `dbo.usp_OnPremOnly`

2 Distributed transactions (MSDTC)

MEDIUM 1 object(s)

WHAT IT MEANS

Code starts distributed transactions (BEGIN DISTRIBUTED TRANSACTION), usually across linked servers.

IMPACT

Needs MSDTC on both sides; managed platforms support it partly or not at all.

HOW TO RESOLVE

1. Check whether the transaction really has to span servers; often one server suffices after the migration.
2. Otherwise use the platform's option (see the alternative) or make the steps idempotent and run them separately.

Alternative on this platform: DTC for Managed Instance: switch it on and open the network to the participants; no T-SQL distributed transactions with other database products

AFFECTED OBJECTS

- `dbo.usp_OnPremOnly`

3 Code changes server settings (sp_configure, trace flags)

MEDIUM 1 object(s)

WHAT IT MEANS

Code changes server settings (sp_configure, RECONFIGURE, trace flags).

IMPACT

Managed platforms do not allow this; the code fails, or the setting it relies on is different there.

HOW TO RESOLVE

1. List which settings the code changes and why.
2. Set the needed values once on the target (parameter groups, database flags, database-scoped configuration) and remove the calls from the code.

AFFECTED OBJECTS

- `dbo.usp_OnPremOnly`

4 Custom error messages (sp_addmessage)

MEDIUM 1 object(s) · Script A1

WHAT IT MEANS

Custom error messages (sp_addmessage, message numbers above 50000) are used. They live in the master database, not in the application database.

IMPACT

After a backup/restore RAISERROR with these numbers fails; on platforms without sp_addmessage they cannot be created at all.

HOW TO RESOLVE

1. Export the messages from the source (script below) and run it on the target.
2. On platforms without sp_addmessage: replace RAISERROR(number) by THROW with the message text.

AFFECTED OBJECTS

- 1 on the server

SCRIPT A1

```
-- run on the SOURCE server; it generates the sp_addmessage calls for the target
SELECT 'EXEC sp_addmessage @msgnum = ' + CAST(message_id AS varchar(10)) + ', @severity = ' +
CAST(severity AS varchar(5))
  + ', @msgtext = N''' + REPLACE(text, '''', '''''') + ''', @lang = ''us_english'', @replace =
''replace'';'
FROM sys.messages WHERE message_id > 50000 AND language_id = 1033;
```

5 Table without a primary key

MEDIUM 1 object(s) · Script A2

WHAT IT MEANS

These tables have no primary key.

IMPACT

Replication-based migration tools (AWS DMS, Azure DMS, transactional replication, change data capture) need a key to apply updates and deletes; without one they copy the table only as a full load or not at all. Duplicate rows are also possible.

HOW TO RESOLVE

1. Check per table whether a column (combination) is unique, or add a surrogate key: an IDENTITY column with a primary key.
2. Add the key before the migration and test the application with it.

AFFECTED OBJECTS

- `dbo.AuditLog`

SCRIPT A2

```
-- dbo.AuditLog: prefer an existing unique column (combination) if there is one
IF NOT EXISTS (SELECT 1 FROM sys.key_constraints WHERE parent_object_id = OBJECT_ID(N'[dbo].
[AuditLog]') AND type = 'PK')
  ALTER TABLE [dbo].[AuditLog] ADD [AuditLogId] int IDENTITY(1,1) NOT NULL CONSTRAINT [PK_AuditLog]
PRIMARY KEY;
```

6 Deprecated data type (text, ntext, image)

LOW 2 object(s) · Script A3

WHAT IT MEANS

text, ntext and image are deprecated since SQL Server 2005 and will be removed in a future version.

IMPACT

They still work, but many functions do not accept them, and tools and drivers handle them poorly.

HOW TO RESOLVE

1. Change them to varchar(max), nvarchar(max) and varbinary(max) (script below). The conversion rewrites the table; plan it outside office hours.
2. Code that uses READTEXT, WRITETEXT, UPDATETEXT or TEXTPTR must be rewritten to normal SELECT/UPDATE (see the upgrade test).

AFFECTED OBJECTS

- `dbo.Customer.Notes` – ntext
- `dbo.Customer.Photo` – image

SCRIPT A3

```
IF EXISTS (SELECT 1 FROM sys.columns WHERE object_id = OBJECT_ID(N'[dbo].[Customer]') AND name =
N'Notes' AND TYPE_NAME(system_type_id) = N'ntext')
  ALTER TABLE [dbo].[Customer] ALTER COLUMN [Notes] nvarchar(max) NULL;
IF EXISTS (SELECT 1 FROM sys.columns WHERE object_id = OBJECT_ID(N'[dbo].[Customer]') AND name =
N'Photo' AND TYPE_NAME(system_type_id) = N'image')
  ALTER TABLE [dbo].[Customer] ALTER COLUMN [Photo] varbinary(max) NULL;
```

7 Compatibility level stays at the old value after the restore

LOW 160 object(s) · Script A4

WHAT IT MEANS

A restored database keeps its compatibility level. Newer optimizer improvements and T-SQL features only become available after raising it.

IMPACT

Without raising it the database runs on the new version with the behaviour of the old one; raising it changes query plans, so it is a planned step with a test.

HOW TO RESOLVE

1. Enable Query Store first, so query performance before and after can be compared.
2. Raise the compatibility level (script below) after the application test passed at the old level.
3. Watch the heaviest queries for two weeks; fix regressions per query (Query Store plan forcing) instead of lowering the level again.

AFFECTED OBJECTS

- 160 → 170

SCRIPT A4

```
IF NOT EXISTS (SELECT 1 FROM sys.databases WHERE name = N'MmdbShop' AND is_query_store_on = 1)
  ALTER DATABASE [MmdbShop] SET QUERY_STORE = ON;
IF (SELECT compatibility_level FROM sys.databases WHERE name = N'MmdbShop') < 170
  ALTER DATABASE [MmdbShop] SET COMPATIBILITY_LEVEL = 170;
```

5 About this report

This report combines the MigrateMyDatabase scan of the database structure with, where available, the data check and the upgrade test on a copy of the database. It is based on the database as it was at the moment of collection.

Effort estimates are indicative. Application code outside the database (queries in the application, reports, ETL) is not part of this analysis unless stated otherwise.

Scripts are generated from the analysed structure: review and test them on a copy before running them in production.

PREPARED BY

Ralph Klos Projects & Consultancy BV
Gaspeldoornstraat 20, 2040 Zandvliet (Antwerpen), België
VAT BE1044614873

CONTACT

migratemydatabase.eu
rkpc.be

Appendix A Scripts

A1 Custom error messages (sp_addmessage)

```
-- run on the SOURCE server; it generates the sp_addmessage calls for the target
SELECT 'EXEC sp_addmessage @msgnum = ' + CAST(message_id AS varchar(10)) + ', @severity = ' +
CAST(severity AS varchar(5))
  + ', @msgtext = N''' + REPLACE(text, '''', '''''') + ''', @lang = ''us_english'', @replace =
''replace'';'
FROM sys.messages WHERE message_id > 50000 AND language_id = 1033;
```

A2 Table without a primary key

```
-- dbo.AuditLog: prefer an existing unique column (combination) if there is one
IF NOT EXISTS (SELECT 1 FROM sys.key_constraints WHERE parent_object_id = OBJECT_ID(N'[dbo].[AuditLog]')
AND type = 'PK')
  ALTER TABLE [dbo].[AuditLog] ADD [AuditLogId] int IDENTITY(1,1) NOT NULL CONSTRAINT [PK_AuditLog]
PRIMARY KEY;
```

A3 Deprecated data type (text, ntext, image)

```
IF EXISTS (SELECT 1 FROM sys.columns WHERE object_id = OBJECT_ID(N'[dbo].[Customer]') AND name = N'Notes'
AND TYPE_NAME(system_type_id) = N'ntext')
  ALTER TABLE [dbo].[Customer] ALTER COLUMN [Notes] nvarchar(max) NULL;
IF EXISTS (SELECT 1 FROM sys.columns WHERE object_id = OBJECT_ID(N'[dbo].[Customer]') AND name = N'Photo'
AND TYPE_NAME(system_type_id) = N'image')
  ALTER TABLE [dbo].[Customer] ALTER COLUMN [Photo] varbinary(max) NULL;
```

A4 Compatibility level stays at the old value after the restore

```
IF NOT EXISTS (SELECT 1 FROM sys.databases WHERE name = N'MmdbShop' AND is_query_store_on = 1)
  ALTER DATABASE [MmdbShop] SET QUERY_STORE = ON;
IF (SELECT compatibility_level FROM sys.databases WHERE name = N'MmdbShop') < 170
  ALTER DATABASE [MmdbShop] SET COMPATIBILITY_LEVEL = 170;
```

Appendix B Inventory

TABLES	ROWS	MB	COLUMNS
dbo.Customer	2	0.2	16
sales.OrderLine	0	0.1	7
sales.Order	0	0	7
hr.Employee	0	0	5
hr.EmployeeHistory	0	0	5
dbo.AuditLog	0	0	3
dbo.ThisTableNameIsDeliberatelyMuchLongerThanSixtyThreeCharactersForPostgres	0	0	1