Microsoft

Database migrations - Benchmarks and Steps to Import to Azure SQL DB Single Database from BACPAC

Prepared by

Data SQL Ninja Team (askdmjfordmtools@microsoft.com)

Disclaimer

The High-Level Architecture, Migration Dispositions and guidelines in this document is developed in consultation and collaboration with Microsoft Corporation technical architects. Because Microsoft must respond to changing market conditions, this document should not be interpreted as an invitation to contract or a commitment on the part of Microsoft.

Microsoft has provided generic high-level guidance in this document with the understanding that MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THE INFORMATION CONTAINED HEREIN.

This document is provided "as-is". Information and views expressed in this document, including URL and other Internet Web site references, may change without notice.

Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

Note: The detail provided in this document has been harvested as part of a customer engagement sponsored through the Azure Data Services Jumpstart Program.

Table of Contents

1 Introduction

This Whitepaper has been developed to help accelerate migrations from on-premises, IaaS and/or relevant SQL Server implementations that need to be migrated to an Azure SQL DB Single database, using BACPAC backup files, and are not able to leverage Data Migration Service (DMS).

For online migrations, when source compatibility permits, it is strongly recommended to use Data Migration Service instead.

2 SQL Server Migration Scenarios and Requirements

2.1 Possible scenarios

This whitepaper covers these possible scenarios:

- ✓ Source Database is not in Bulk Logged or Full recovery model, and cannot be turned to either option before the migration.
- ✓ There is no possibility to use Azure Data Migration Service
- ✓ Source is SQL Server 2008 or later
- ✓ Target is Azure SQL DB Single Database

2.2 Applications and tools required

- ✓ Access to source database as backup operator or administrator, to execute full BACPAC creation from source
- ✓ Access to the Azure Portal to deploy an Azure SQL DB Single Database
- ✓ Access to the Azure Portal to deploy an Azure Virtual Machine
- ✓ Access to the Azure Blob Storage container to create a SAS key to:
  - Write the BACPAC backup file
- ✓ Access to download and Install Data Migration Assistant (DMA)
- ✓ Access to download and Install sqlpackage
- ✓ Access to download and Install azcopy:
  - Windows 64-bit (zip)
  - Windows 32-bit (zip)
  - Linux (tar)
  - MacOS (zip)

3 Benchmark of BACPAC Restore operations

3.1 Benchmark reference

- 20 GB Database
- 10 GB BACPAC file
- 11 tables, 1 large with 140 million rows, and no PK (there is a higher overhead with tables with no PK as it does **not** take advantage of parallelism).

3.2 Benchmark table

Parallel	Target	Time	Savings	Source	BACPAC Location	Service/App
N/A	Standard 3000 DTU	02:30:00		Azure VM 16 cores (lowest priced)	Blob Storage	New-AzSqlDatabaseImport (Azure CLI)
N/A	Standard 3000 DTU	01:35:00	37%*	N/A	Blob Storage	Azure Portal (Import)
32	Standard 3000 DTU	01:18:27	48%	Laptop	Local Drive	SqlPackage
32	Standard 3000 DTU	00:54:40	64%	Azure VM 16 cores (lowest priced)	Local Drive	SqlPackage
32	Standard Gen 5 40 vCores	00:47:33	68%	Azure VM 16 cores (lowest priced)	Local Drive	SqlPackage
32	Business Critical Gen 5 16 vCores	00:39:47	73%	Azure VM 16 cores (lowest priced)	Local Drive	SqlPackage

* Time of the day can be relevant to the tier behind database import.

4 Planning the migration

The recommendation is to plan the migration, to ensure that little to no changes (recommended) are happening to the source database, to ensure data consistency.

It is also recommended to proceed when the database is not been intensively used for writes.

4.1 High Level Process

1. Ensure that Azure SQL DB Single Database is compatible with your database version and features, by running Data Migration Assistant (DMA)
2. Follow best practices guidance before creation of the BACPAC file
3. Perform the creation of BACPAC file
4. Transfer the BACPAC file to Azure Blob storage
5. Create an Azure Virtual Machine
6. Transfer the BACPAC file from Azure Blob storage to the Azure Virtual Machine
7. Create an Azure SQL DB Single Instance Business Critical Gen 5 16 vCores
8. Initiate the Restore with sqlpackage
9. Recreate any dropped indexes, and perform post migration maintenance
10. Migrate Azure SQL DB Single Instance Business Critical to the tier best suited for your workload

5 Initiate the Migration

5.1 Compatibility and best practices

1. Install and execute the Database Migration Assistant against the source database to be migrated. Ensure that there are no incompatibilities.
2. Follow best practices guidance before the creation of the BACPAC file:
    - Disable autostatistics during migration
    - For large datasets that contain historical data, consider adding partitions to tables and indexes.
    - Consider dropping indexed views, and recreate them once finished
3. Leverage sqlpackage to create the BACPAC file whenever possible, or use SSMS to create one.

5.2 Data Transfer and Resource creation

1. Create a SAS key for writing the BACPAC file to the BLOB Storage
2. It is recommended to transfer your file using azcopy to maximize bandwidth utilization whenever possible

Sample command:

```
azcopy cp "/path/src.bacpac" "https://account.blob.core.windows.net/container?SASKey"
```

3. Create an Azure VM with accelerated networking and enough vCores (tests have been performed with 16 vCores, however less could be leveraged)
4. Install sqlpackage into the new Azure VM
5. Copy the BACPAC file from Azure Blob storage to a local disk in the Virtual Machine just created
6. Create an Azure SQL DB Single Database
    a. Business Critical Tier
    b. Gen 5
    c. 16 vCores

5.3 Restore and post operations

1. Initiate the restore process with sqlpackage

Sample command:

```
sqlpackage /Action:Import /MaxParallelism:32 /SourceFile:D:\SQLBackup\file.bacpac
/TargetServerName:databasename.database.windows.net /TargetDatabaseName:TargetDatabase
/TargetPassword:****************** /TargetUser:username /TargetTimeout:60000
```

2. Monitor the log creation rate within the target database.
    a. If the rate is too low (e.g. under 30%), it is possible that higher parallelism can be indicated
    b. If the rate is too high (e.g. over 90%), it is possible that lower parallelism can be indicated

```
SELECT * FROM sys.dm_db_resource_stats ORDER BY end_time DESC;
```

3. Run statistics update operations
4. Lower the Azure SQL DB Single Database to a lower tier that aligns better to the target application (if applicable).
5. Database is ready for usage.
6. Azure VM can be paused or destroyed post migration.