Migrate SQL Server databases to an Azure SQL Managed Instance (online) using Azure Database Migration Service (DMS) with Azure Data Studio

Abstract

In this document we migrate MSSQL databases from on-premises instance of SQL server 2016 to Azure SQL Managed Instance with Minimal Downtime(online) using Azure Database Migration Service (DMS) with Azure Data Studio

Jayaprakash, Jagath

Introduction

We can use Azure Database Migration Service to migrate the databases from a SQL Server instance to an Azure SQL Managed Instance with minimal downtime(online migrations).

We can use the Azure SQL Migration extension in Azure Data Studio to migrate SQL Server database(s) to either Azure SQL Managed Instance (Platform-as-a-Service) or to SQL Server on Azure Virtual Machines (Infrastructure-as-a-Service). The Azure SQL Migration extension for Azure Data Studio provides a wizard to assess SQL Server database(s) for migration to Azure SQL Managed Instance or to SQL Server on Azure Virtual Machines and then migrate them by choosing between the online or offline migration modes.

Azure Database Migration service orchestrates data movement activities and provides monitoring of migration activities.

Prerequisites

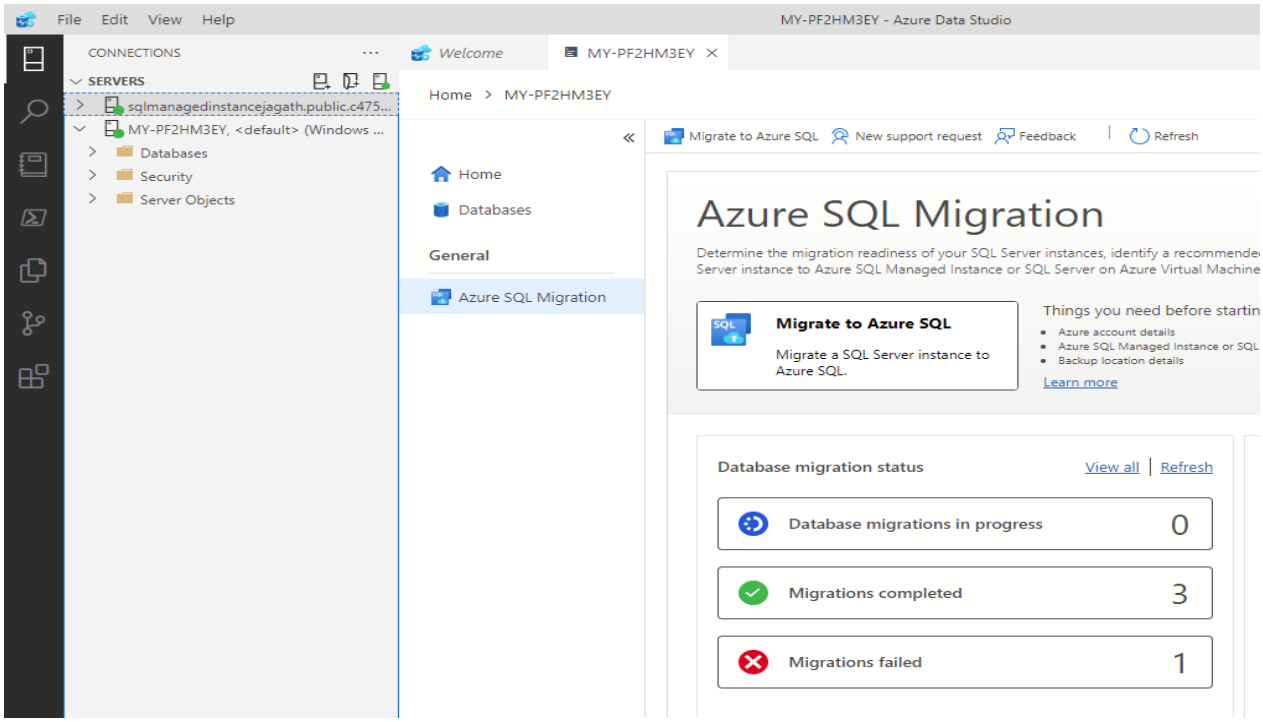
1. Download and install latest Azure Data Studio https://docs.microsoft.com/en-us/sql/azure-data-studio/download-azure-data-studio?view=sql-server-ver155
2. Install the Azure SQL Migration extension from the Azure Data Studio marketplace.
3. Have and Azure account with that is assigned to one of the built-in roles listed below:
    - Contributor for the target Azure SQL Managed Instance (and Storage Account to upload your database backup files from SMB network share).
    - Owner or Contributor role for the Azure Resource Groups containing the target Azure SQL Managed Instance or the Azure storage account.
    - Owner or Contributor role for the Azure subscription.
4. Create a target Azure SQL Managed Instance, I have created a test MI: sqlmanagedinstancejagath.public.c475f3d7c552.database.windows.net,3342
5. Create a source SQL instance which I have setup in my laptop.
6. Ensure that the logins used to connect the source SQL Server are members of the sysadmin server role or have CONTROL SERVER permission.
7. Use one of the following storage options for the full database and transaction log backup files: We use Azure storage account blob container in our case.
    - SMB network share
    - Azure storage account file share or blob container

      Important
        - ✓ If database backup files are provided in an SMB network share, Create an Azure storage account that allows the DMS service to upload the database backup files. Make sure to create the Azure Storage Account in the same region as the Azure Database Migration Service instance is created.
        - ✓ Azure Database Migration Service does not initiate any backups, and instead uses existing backups, which may already have as part of our disaster recovery plan, for the migration.
        - ✓ We should take backups using the WITH CHECKSUM option.

- ✓ Each backup can be written to either a separate backup file or multiple backup files. However, appending multiple backups (i.e. full and t-log) into a single backup media is not supported.
- ✓ Use compressed backups to reduce the likelihood of experiencing potential issues associated with migrating large backups.
- ✓ Ensure that the storage account created is in the same resource group and region that of target SQL Managed instance

Launch the Migrate to Azure SQL wizard in Azure Data Studio

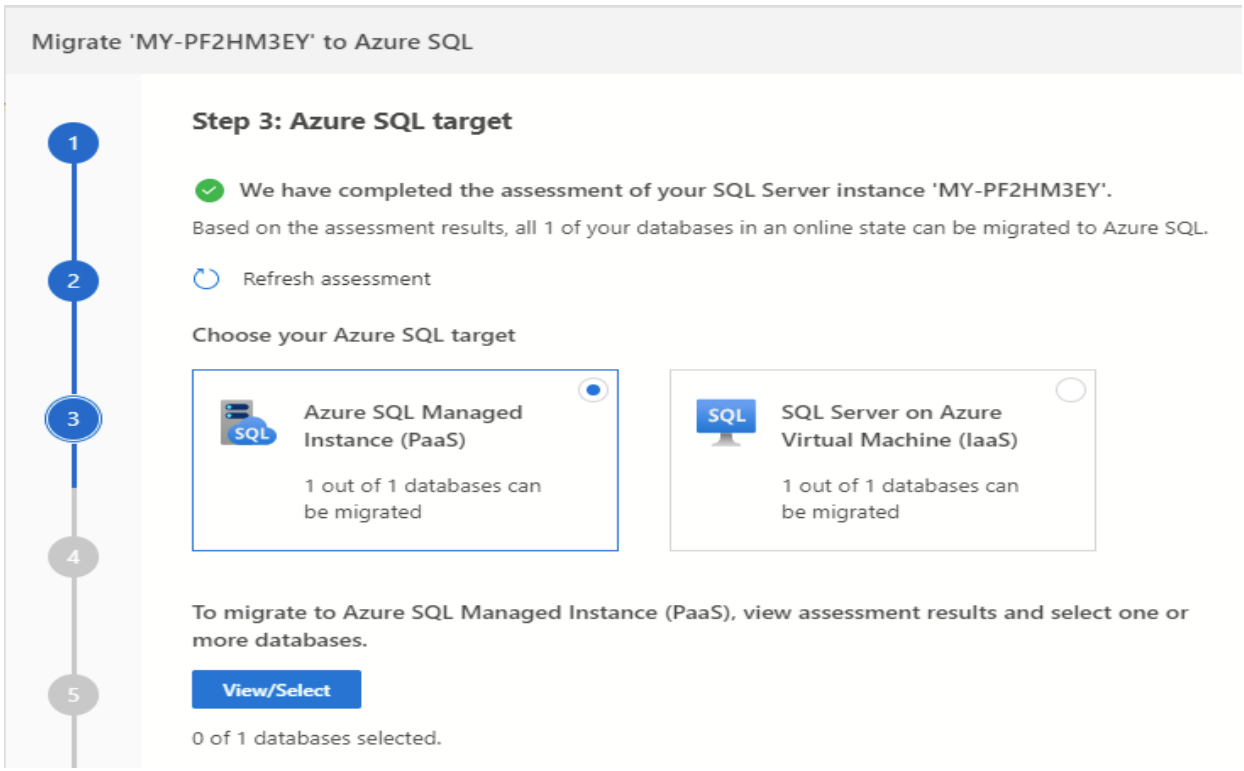
1. Open Azure Data Studio and select the server icon to connect to your on-premises SQL Server (or SQL Server on Azure Virtual Machine).
2. On the server connection, right-click and select Manage.
3. On the server's home page, Select Azure SQL Migration extension.
4. On the Azure SQL Migration dashboard, select Migrate to Azure SQL to launch the migration wizard.



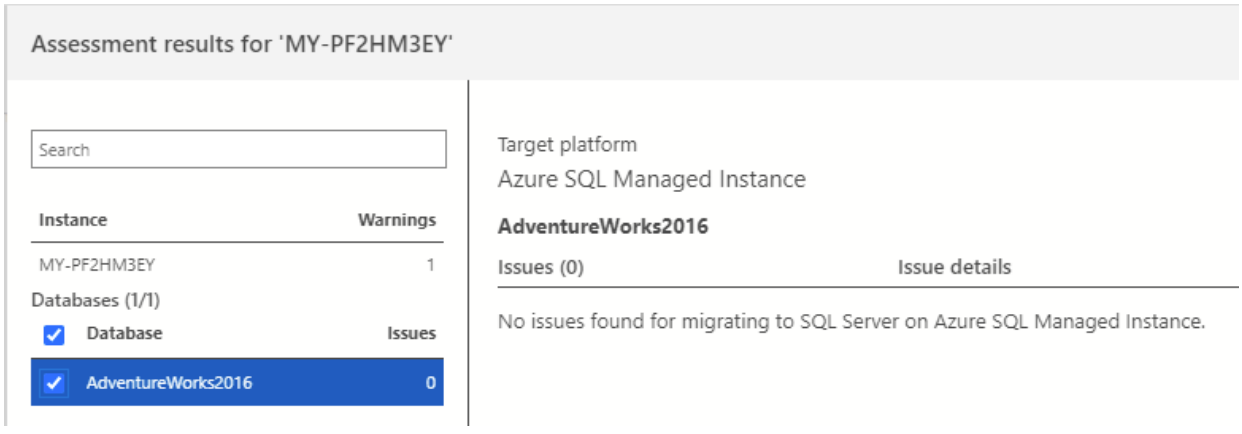
****Please ignore the Migration status that you see above as they show my previous attempts.*

Run database assessment and select target

1. Select the database(s) to run assessment and select Next. Now we use AdventureWorks2016.
2. Select Azure SQL Managed Instance as the target.



3. Select on the View/Select button to view details of the assessment results for your database(s), select the database(s) to migrate, and select OK. If any issues are displayed in the assessment results, they need to be remediated before proceeding with the next steps.

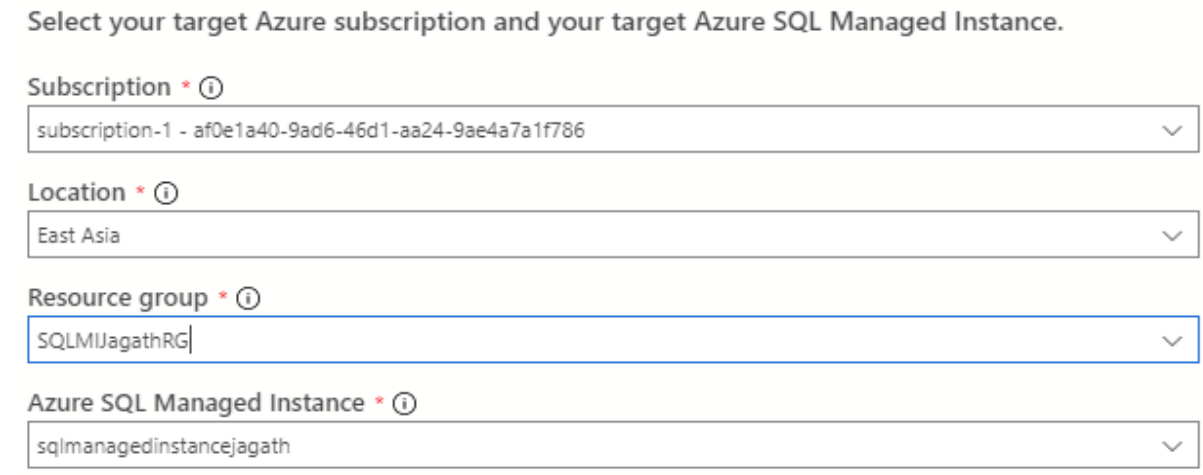


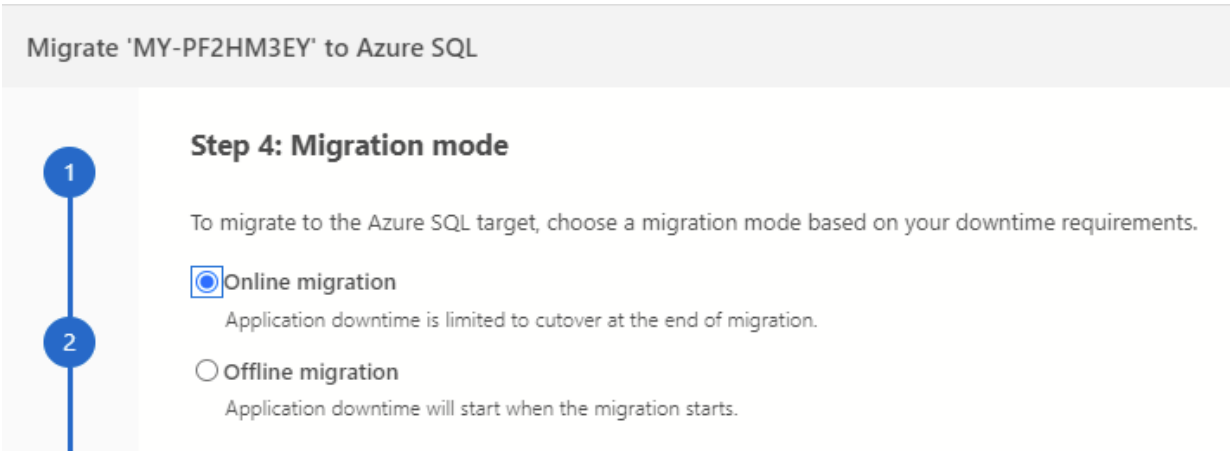
4. Specify our target Azure SQL Managed Instance by selecting our subscription, location, resource group from the corresponding drop-down lists and select Next.

Configure migration settings

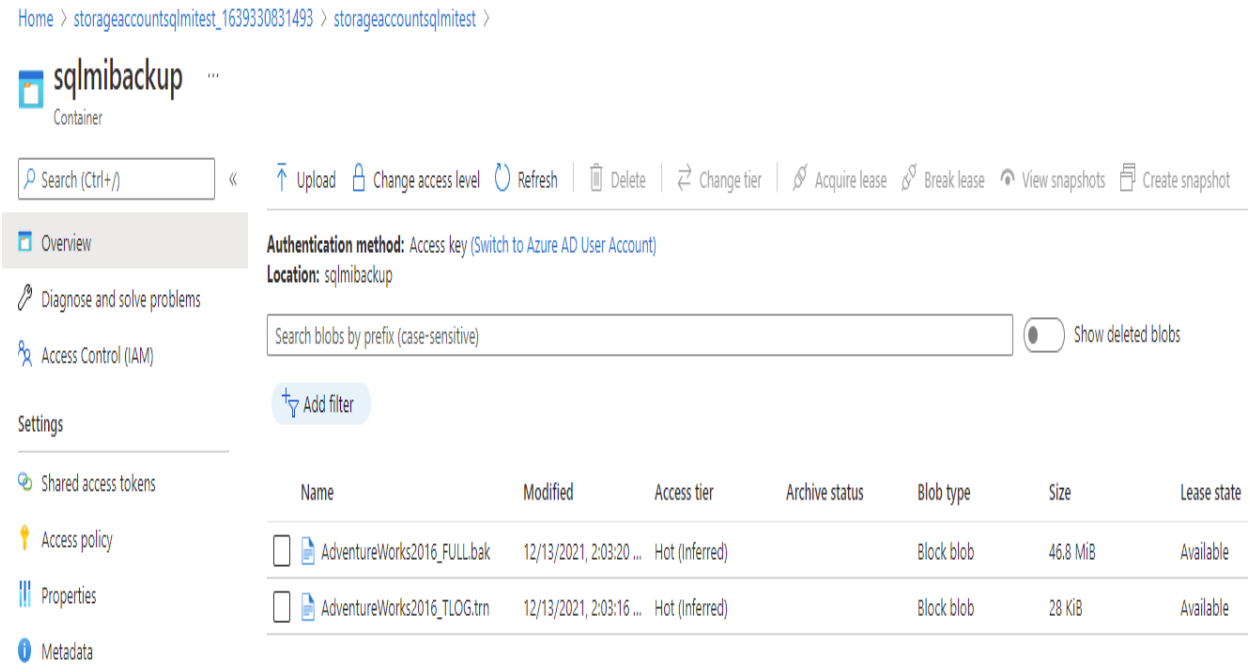
1. Select Online migration as the migration mode.
   Note: In the online migration mode, the source SQL Server database is available for read and write activity while database backups are continuously restored on target Azure SQL Managed Instance. Application downtime is limited to duration for the cutover at the end of migration.

2. Select the location of our database backups. Our database backups can either be located on an on-premises network share or in an Azure storage blob container.
   In this case we use Azure Storage blob container: storageaccountsqlmitest\sqlmibackup.

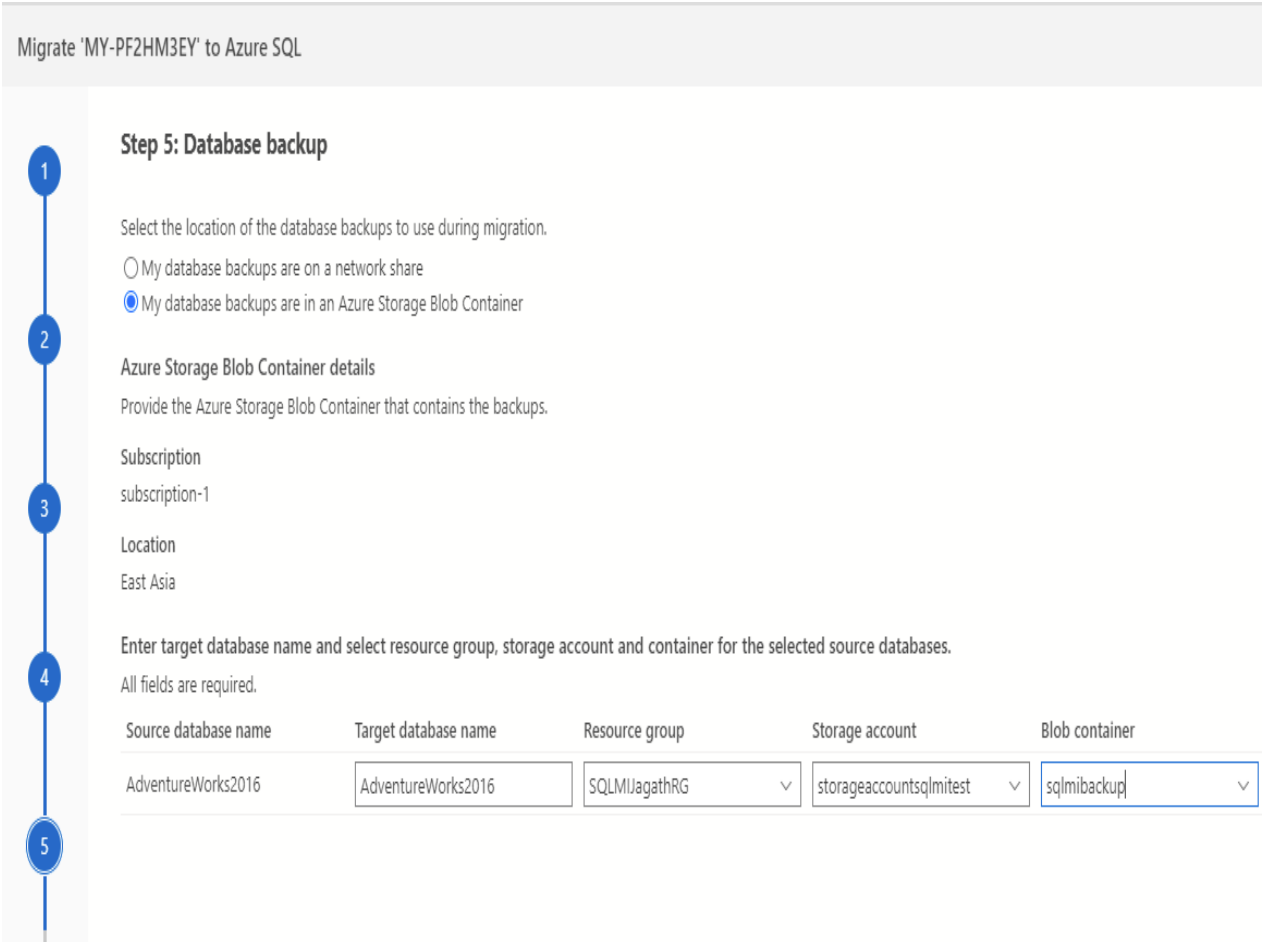




3. After selecting the backup location, provide details of source SQL Server and source backup location. Ensure that we move the database FULL and Tlog backup to the storage account\container which is storageaccountsqlmitest\sqlmibackup. We can use either azcopy\azure storage explorer or upload directly from azure portal.



4. Specify the Azure storage account by selecting the Subscription, Location, and Resource Group from the corresponding drop-down lists. This Azure storage account will be used by DMS to upload the database backups from network share.

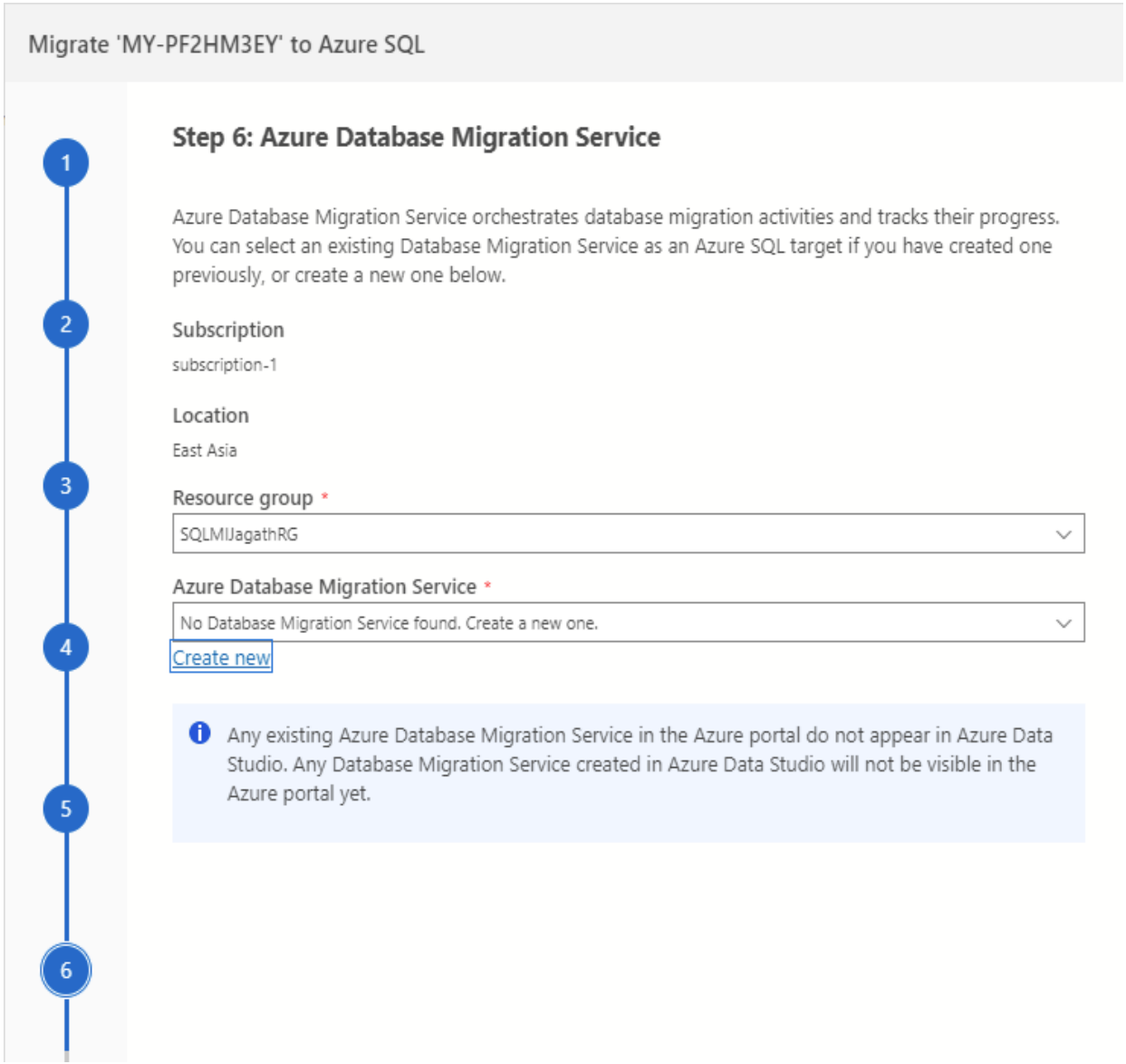


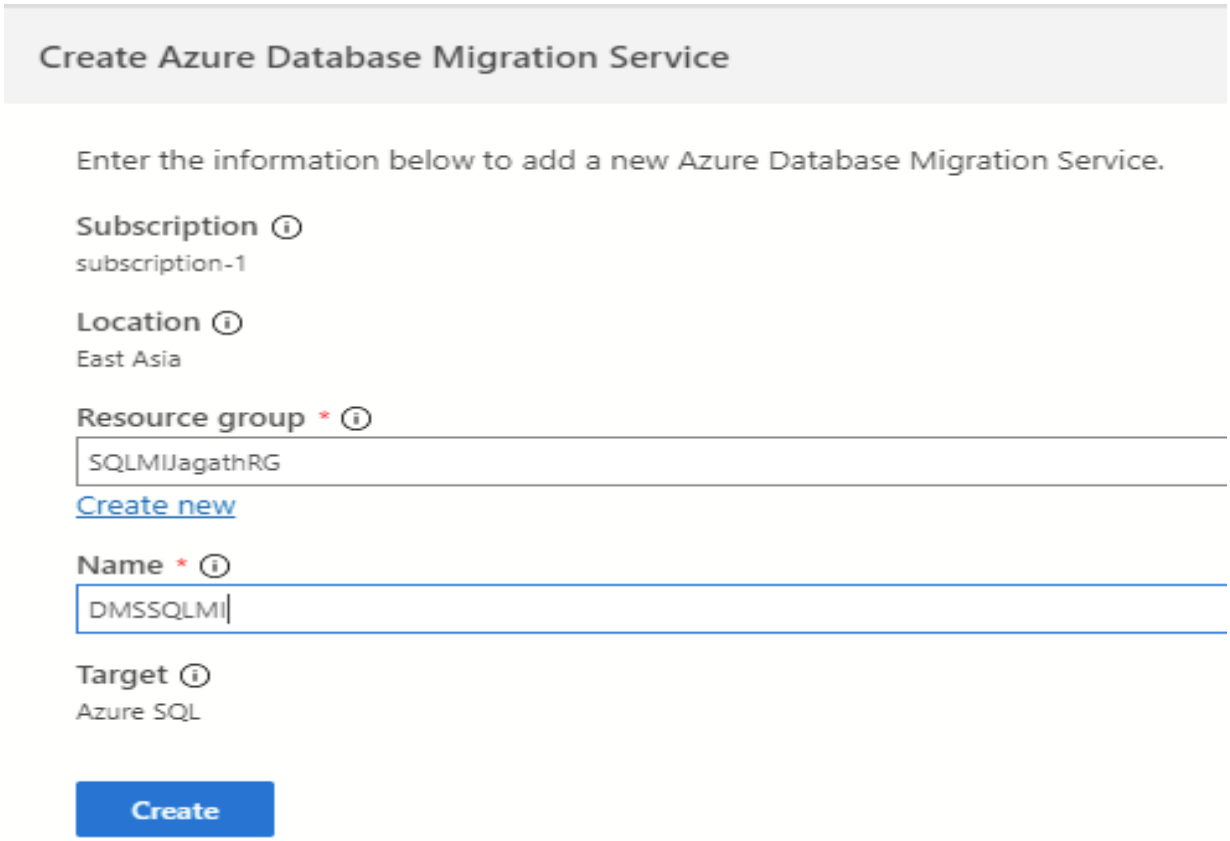
Create Azure Database Migration Service

1. Create a new Azure Database Migration Service.

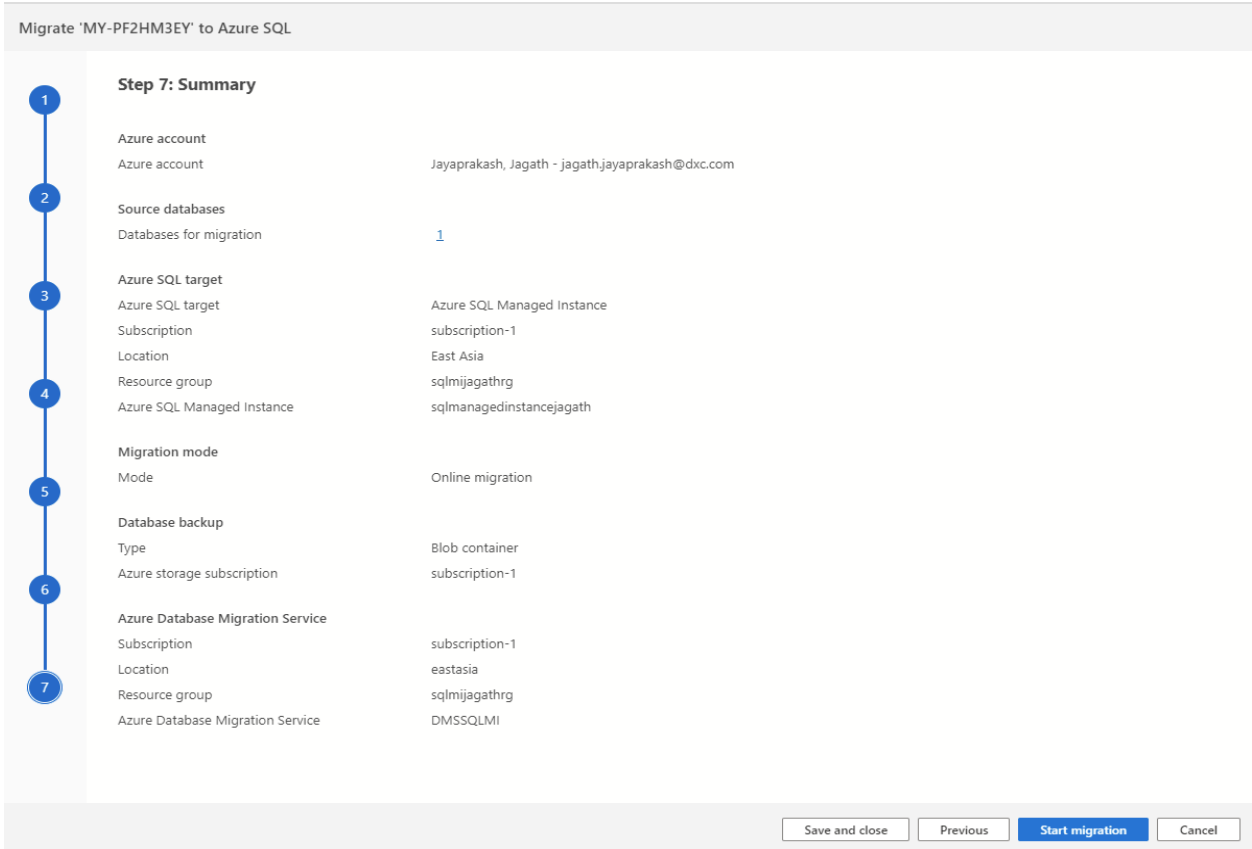
2. Select the Resource group where you have an existing DMS or need to create a new one. The Azure Database Migration Service dropdown will list any existing DMS in the selected resource group.
3. To reuse an existing DMS, select it from the dropdown list and the status of the self-hosted integration runtime will be displayed at the bottom of the page.

4. To create a new DMS, select Create new. On the Create Azure Database Migration Service, screen provide the name for DMS and select Create.



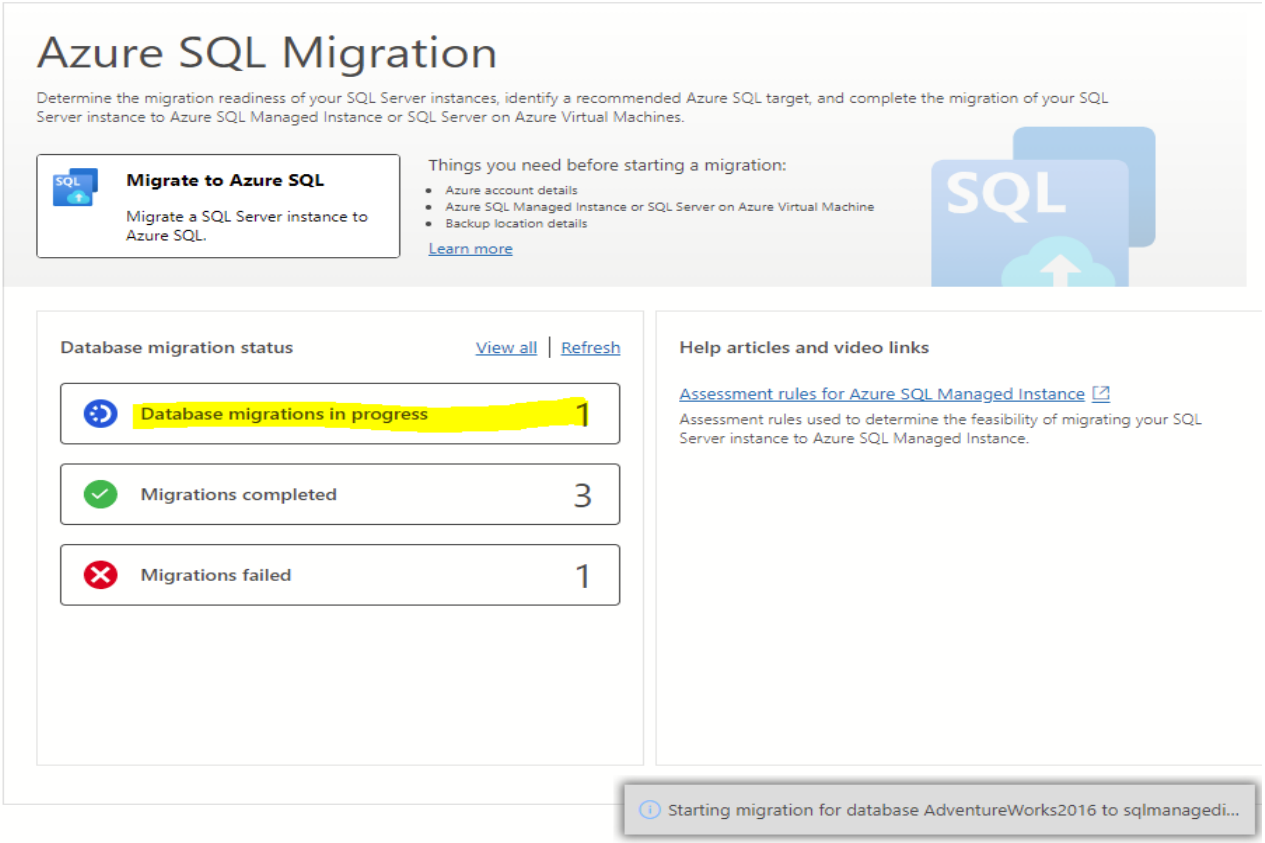


5. Review the migration summary and select Done to start the database migration.



Monitor your migration

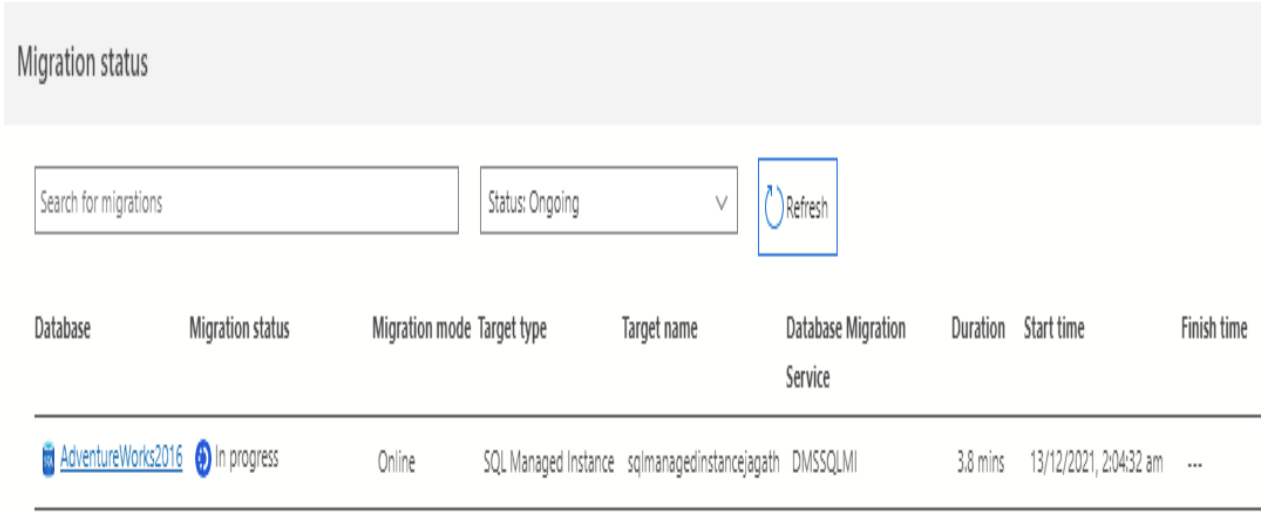
1. On the Database Migration Status, we can track the migrations in progress, migrations completed, and migrations failed (if any).



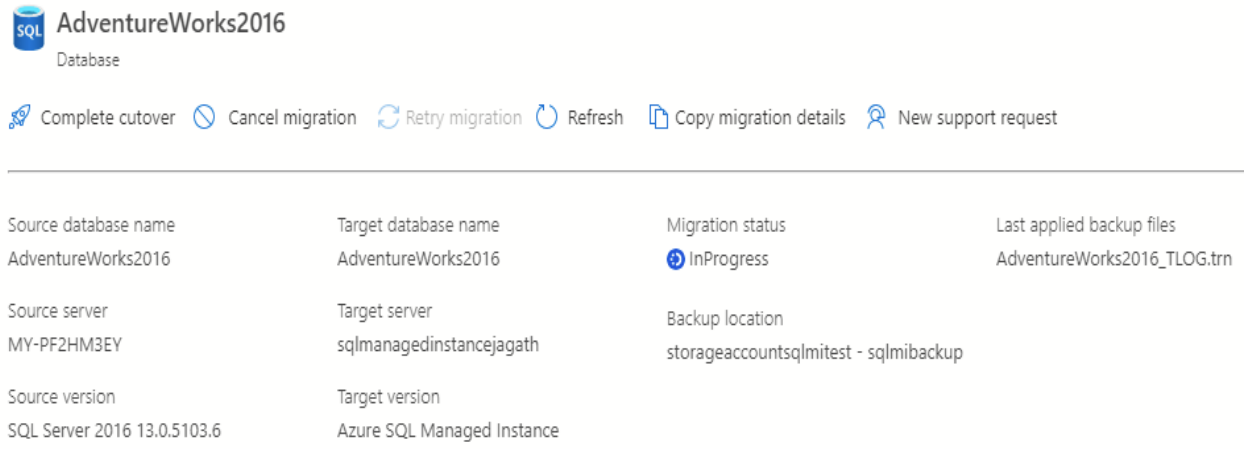
2. Select Database migrations in progress to view ongoing migrations and get further details by selecting the database name.

3. The migration details page displays the backup files and the corresponding status:

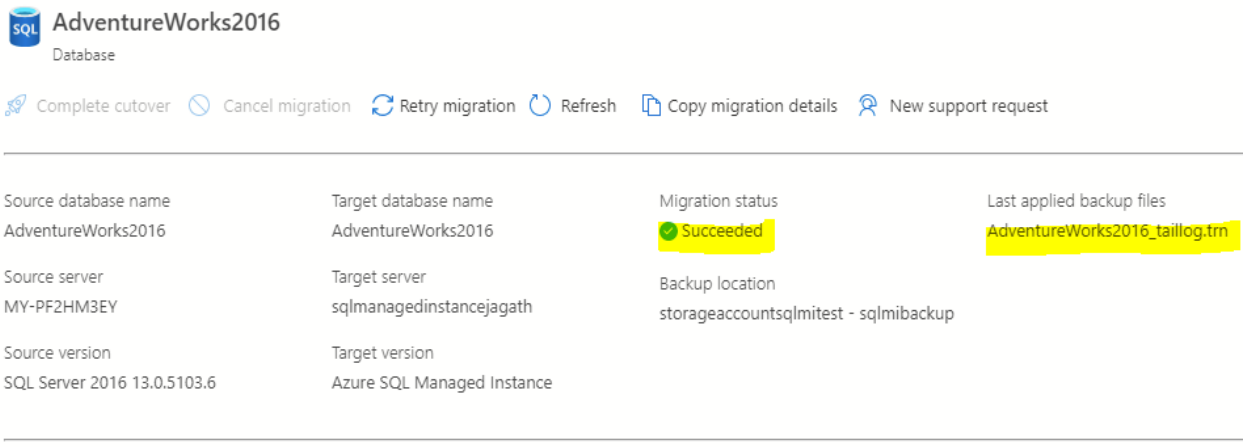
Status	Description
Arrived	Backup file arrived in the source backup location and validated
Uploading	Integration runtime is currently uploading the backup file to Azure storage
Uploaded	Backup file is uploaded to Azure storage
Restoring	Azure Database Migration Service is currently restoring the backup file to Azure SQL Managed Instance
Restored	Backup file is successfully restored on Azure SQL Managed Instance
Canceled	Migration process was canceled
Ignored	Backup file was ignored as it doesn't belong to a valid database backup chain



Click on database: AdventureWorks2016 to see more details on the migration progress.



We can see that the Tlog backup has been restored\applied successfully as shown above. Now its ready for completing migration cutover.

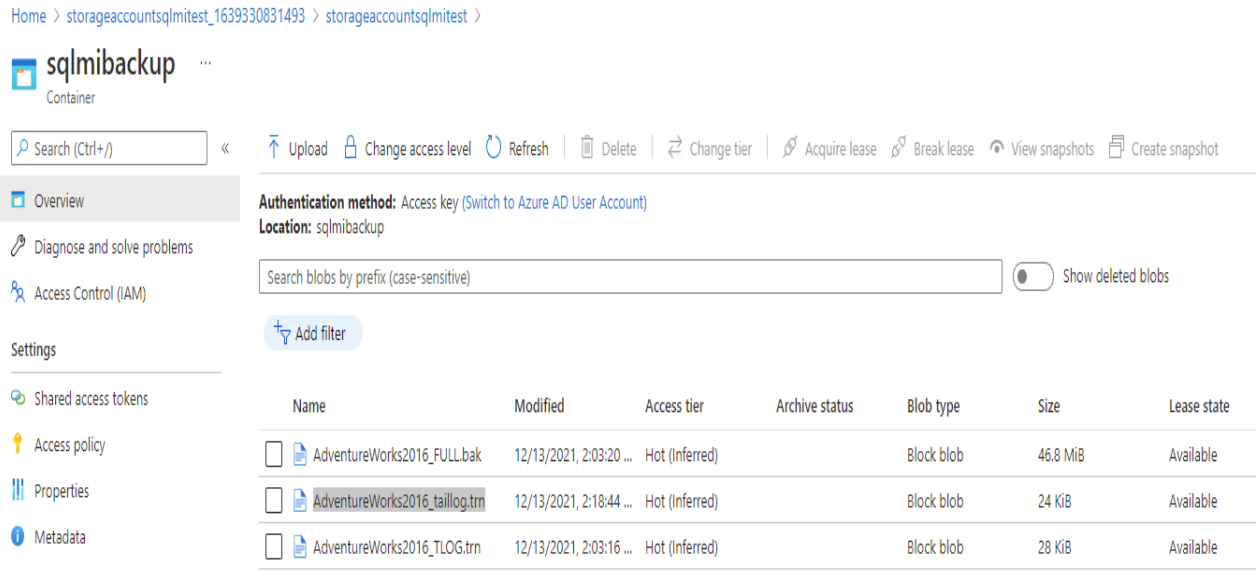


Complete migration cutover

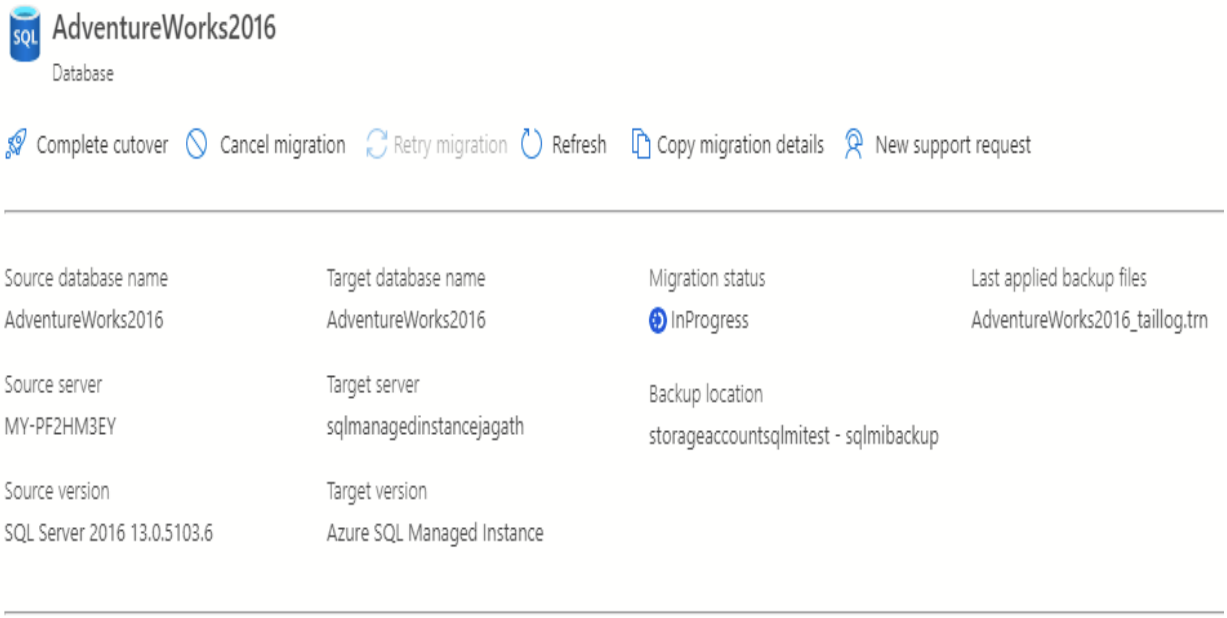
The final step of the tutorial is to complete the migration cutover to ensure the migrated database in Azure SQL Managed Instance is ready for use. This is the only part in the process that requires downtime for applications that connect to the database and hence the timing of the cutover needs to be carefully planned with business or application stakeholders.

To complete the cutover,

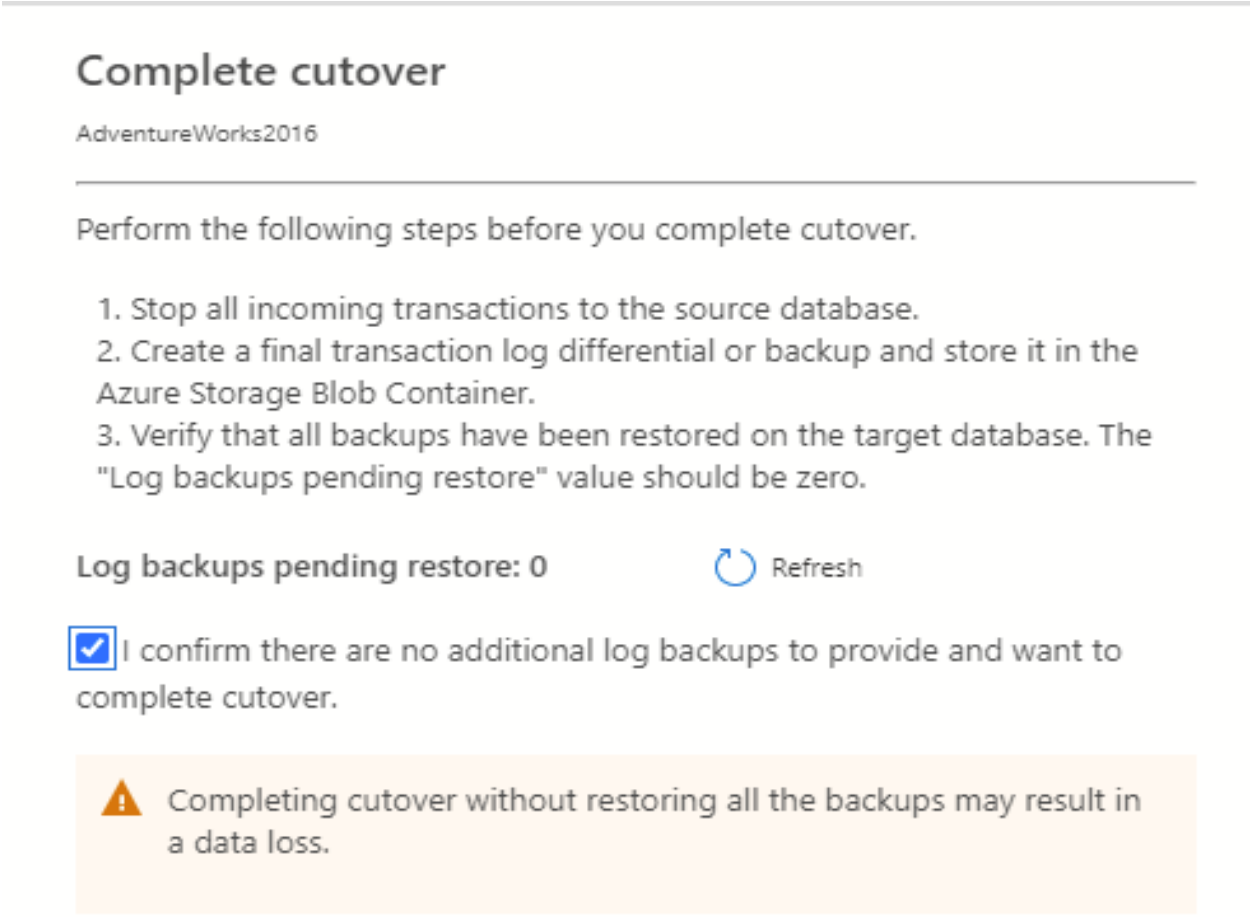
1. Stop all incoming transactions to the source database and prepare to make any application configuration changes to point to the target database in Azure SQL Managed Instance.
2. Take any tail log backups for the source database in the backup location specified, here my tail log backup is AdventureWorks2016_taillog.trn



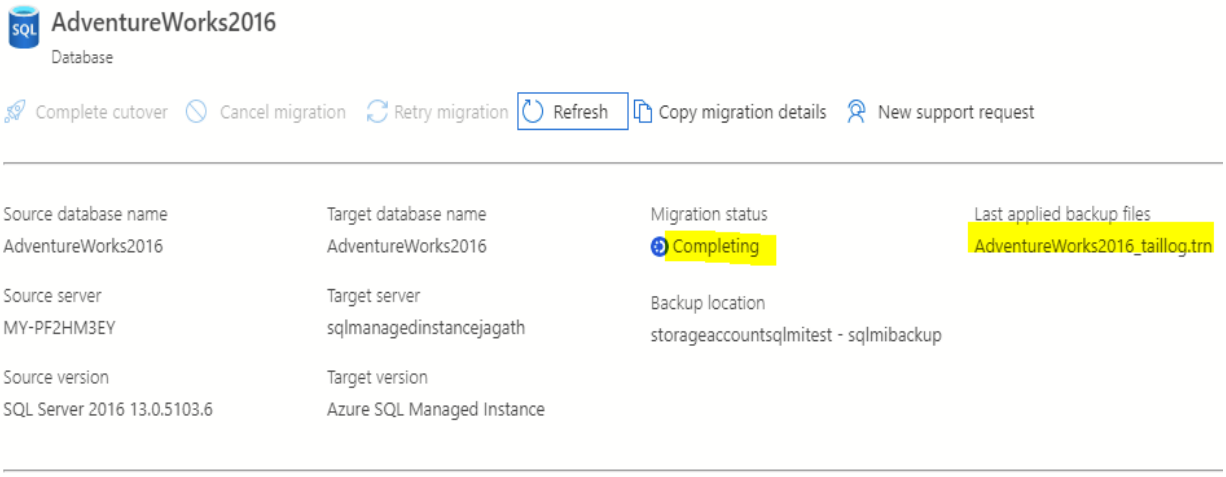
3. Ensure all database backups have the status Restored in the monitoring details page.



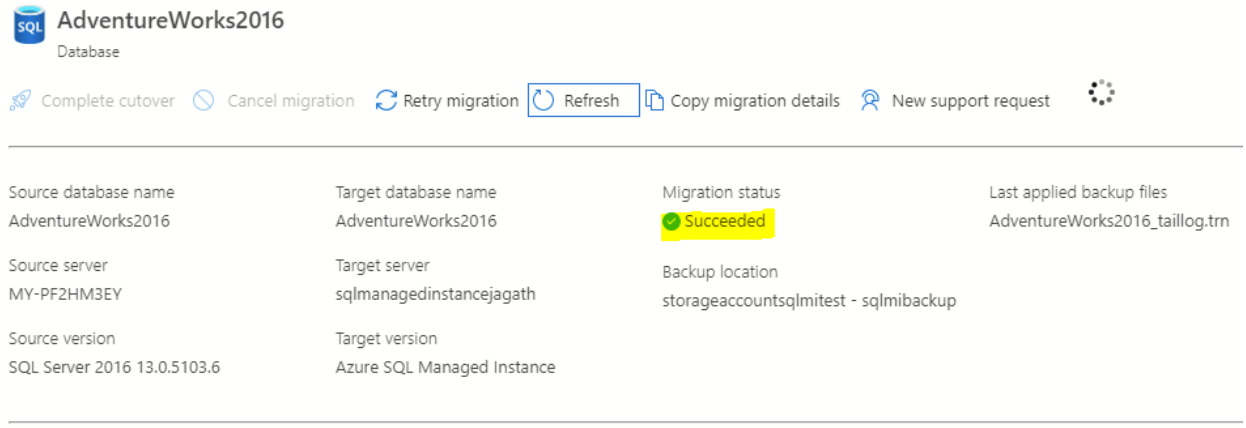
4. select Complete cutover in the monitoring details page



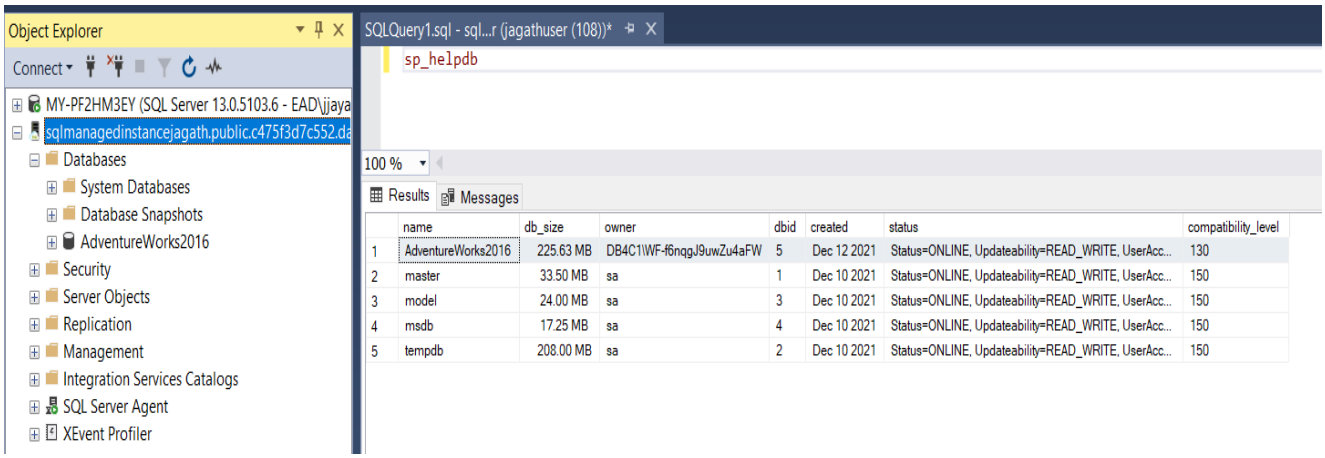
During the cutover process, the migration status changes from in progress to completing. When the cutover process is completed, the migration status changes to succeeded to indicate that the database migration is successful and that the migrated database is ready for use.



After refreshing in a few moments,



Upon checking the SSMS, we can see the AdventureWorks2016 database has been restored successfully.



Migrating multiple databases

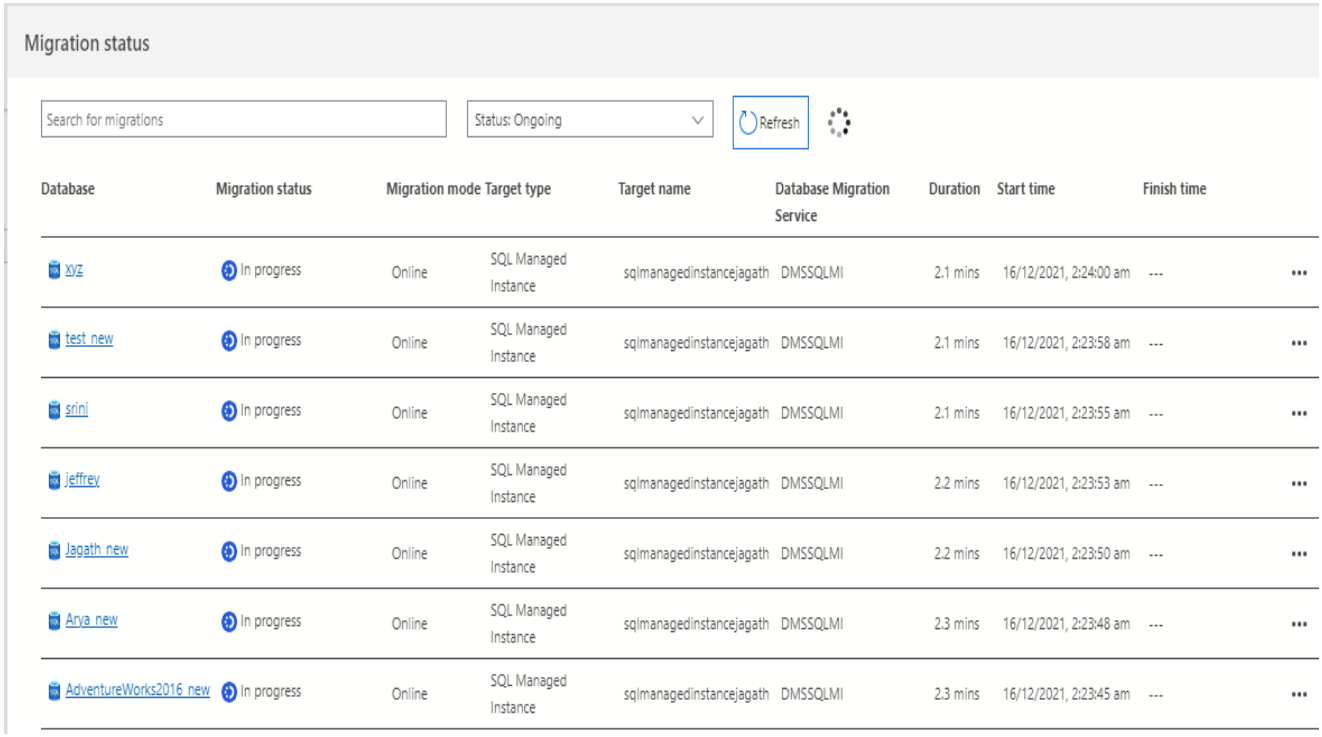
We can migrate more than one database at time using the latest version of Azure Data Studio which is 1.34.0

Status of restoration can be identified from the below scripts to be rain in SSMS or Azure Data Studio.

```
SELECT * FROM sys.dm_operation_status
   WHERE major_resource_id = 'testdb1000'
   ORDER BY start_time DESC;

SELECT session_id as SPID, command, a.text AS Query, start_time, percent_complete
   , dateadd(second,estimated_completion_time/1000, getdate()) as
estimated_completion_time
FROM sys.dm_exec_requests r
CROSS APPLY sys.dm_exec_sql_text(r.sql_handle) a
WHERE r.command in ('BACKUP DATABASE','RESTORE DATABASE')
```

Migration Status = InProgress

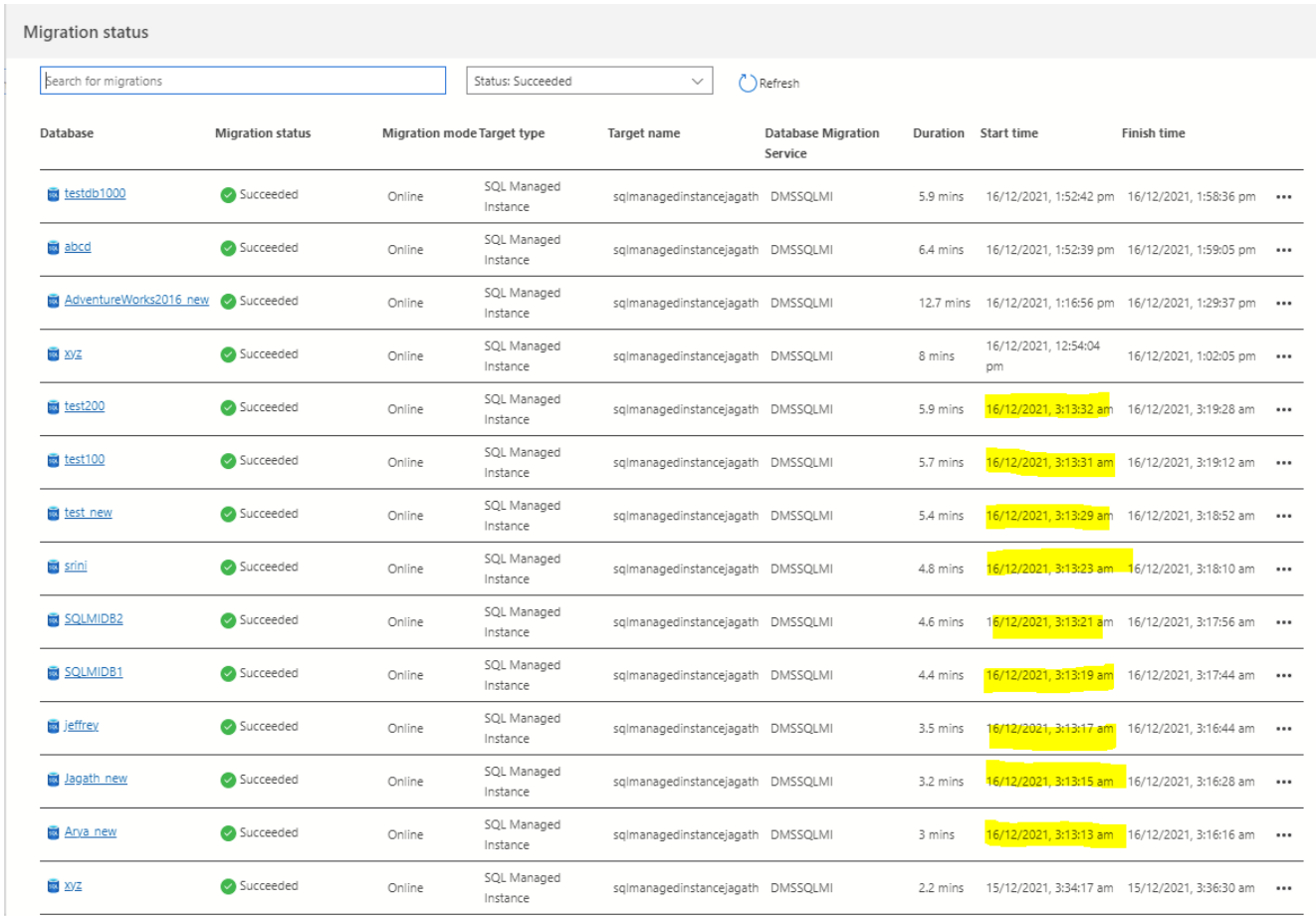


Migration status

Search for migrations | Status: Ongoing | Refresh

Database	Migration status	Migration mode	Target type	Target name	Database Migration Service	Duration	Start time	Finish time
xyz	In progress	Online	SQL Managed Instance	sqlmanagedinstancejagath	DMSSQLMI	2.1 mins	16/12/2021, 2:24:00 am	---
test new	In progress	Online	SQL Managed Instance	sqlmanagedinstancejagath	DMSSQLMI	2.1 mins	16/12/2021, 2:23:58 am	---
srini	In progress	Online	SQL Managed Instance	sqlmanagedinstancejagath	DMSSQLMI	2.1 mins	16/12/2021, 2:23:55 am	---
jeffrey	In progress	Online	SQL Managed Instance	sqlmanagedinstancejagath	DMSSQLMI	2.2 mins	16/12/2021, 2:23:53 am	---
Jagath new	In progress	Online	SQL Managed Instance	sqlmanagedinstancejagath	DMSSQLMI	2.2 mins	16/12/2021, 2:23:50 am	---
Arya new	In progress	Online	SQL Managed Instance	sqlmanagedinstancejagath	DMSSQLMI	2.3 mins	16/12/2021, 2:23:48 am	---
AdventureWorks2016 new	In progress	Online	SQL Managed Instance	sqlmanagedinstancejagath	DMSSQLMI	2.3 mins	16/12/2021, 2:23:45 am	---

Migration Status = Completed



Migration status

Search for migrations | Status: Succeeded | Refresh

Database	Migration status	Migration mode	Target type	Target name	Database Migration Service	Duration	Start time	Finish time
testdb1000	Succeeded	Online	SQL Managed Instance	sqlmanagedinstancejagath	DMSSQLMI	5.9 mins	16/12/2021, 1:52:42 pm	16/12/2021, 1:58:36 pm
abcd	Succeeded	Online	SQL Managed Instance	sqlmanagedinstancejagath	DMSSQLMI	6.4 mins	16/12/2021, 1:52:39 pm	16/12/2021, 1:59:05 pm
AdventureWorks2016 new	Succeeded	Online	SQL Managed Instance	sqlmanagedinstancejagath	DMSSQLMI	12.7 mins	16/12/2021, 1:16:56 pm	16/12/2021, 1:29:37 pm
xyz	Succeeded	Online	SQL Managed Instance	sqlmanagedinstancejagath	DMSSQLMI	8 mins	16/12/2021, 12:54:04 pm	16/12/2021, 1:02:05 pm
test200	Succeeded	Online	SQL Managed Instance	sqlmanagedinstancejagath	DMSSQLMI	5.9 mins	16/12/2021, 3:13:32 am	16/12/2021, 3:19:28 am
test100	Succeeded	Online	SQL Managed Instance	sqlmanagedinstancejagath	DMSSQLMI	5.7 mins	16/12/2021, 3:13:31 am	16/12/2021, 3:19:12 am
test new	Succeeded	Online	SQL Managed Instance	sqlmanagedinstancejagath	DMSSQLMI	5.4 mins	16/12/2021, 3:13:29 am	16/12/2021, 3:18:52 am
srini	Succeeded	Online	SQL Managed Instance	sqlmanagedinstancejagath	DMSSQLMI	4.8 mins	16/12/2021, 3:13:23 am	16/12/2021, 3:18:10 am
SQLMIDB2	Succeeded	Online	SQL Managed Instance	sqlmanagedinstancejagath	DMSSQLMI	4.6 mins	16/12/2021, 3:13:21 am	16/12/2021, 3:17:56 am
SQLMIDB1	Succeeded	Online	SQL Managed Instance	sqlmanagedinstancejagath	DMSSQLMI	4.4 mins	16/12/2021, 3:13:19 am	16/12/2021, 3:17:44 am
jeffrey	Succeeded	Online	SQL Managed Instance	sqlmanagedinstancejagath	DMSSQLMI	3.5 mins	16/12/2021, 3:13:17 am	16/12/2021, 3:16:44 am
Jagath new	Succeeded	Online	SQL Managed Instance	sqlmanagedinstancejagath	DMSSQLMI	3.2 mins	16/12/2021, 3:13:15 am	16/12/2021, 3:16:28 am
Arya new	Succeeded	Online	SQL Managed Instance	sqlmanagedinstancejagath	DMSSQLMI	3 mins	16/12/2021, 3:13:13 am	16/12/2021, 3:16:16 am
xyz	Succeeded	Online	SQL Managed Instance	sqlmanagedinstancejagath	DMSSQLMI	2.2 mins	15/12/2021, 3:34:17 am	15/12/2021, 3:36:30 am

Limitations\Known Issues with online migrations to Azure SQL Managed Instance

Backup requirements

- Backup with checksum - Azure Database Migration Service uses the backup and restore method to migrate on-premises databases to SQL Managed Instance. Azure Database Migration Service only supports backups created using checksum.
- Backup media - Make sure to take every backup on a separate backup media (backup files). Azure Database Migration Service doesn't support backups that are appended to a single backup file. Take full backup and log backups to separate backup files.

Data and log file layout

- Number of log files - Azure Database Migration Service doesn’t support databases with multiple log files. If we have multiple log files, shrink, and reorganize them into a single transaction log file. Because we can't remote to log files that aren't empty, need to back up the log file first.

SQL Server features

- FileStream/FileTables - SQL Managed Instance currently doesn’t support FileStream and FileTables. For workloads dependent on these features, we recommend to opt for SQL Servers running on Azure VMs as our Azure target.
- In-memory tables - In-memory OLTP is available in the Premium and Business Critical tiers for SQL Managed Instance; the General-Purpose tier doesn't support In-memory OLTP.

Migration resets

- Deployments - SQL Managed Instance is a PaaS service with automatic patching and version updates. During migration of your SQL Managed Instance, non-critical updates are held for up to 36 hours. Afterwards (and for critical updates), if the migration is disrupted, the process resets to a full restore state.

  Migration cutover can only be called after the full backup is restored and catches up with all log backups. If production migration cutovers are affected, contact the dmsfeedback@microsoft.com.