Scaling - In-memory tables or objects exists

Last updated by | Charlene Wang | Dec 5, 2022 at 1:55 AM PST

Contents

Issue

Scale a Premium P1 125 DTU, 500 GB to Standard S2 50 DTU, 250 GB and it failed with below error:

The database cannot proceed with pricing-tier update as it has memory-optimized objects. Please drop such objects and try again.

Investigation/Analysis

Error is expected as In-memory tables or procedures not supported in lower tiers and it is only supported for premier tiers.

Work around either to drop the objects or scaling up to premium tier.

Mitigation

In-Memory OLTP isn't supported in the General Purpose, Standard or Basic tier. Therefore, it isn't possible to move a database that has any In-Memory OLTP objects to one of these tiers.

Before you downgrade the database to General Purpose, Standard, or Basic, remove all memory-optimized tables and table types, as well as all natively compiled T-SQL modules. Scaling-down resources in Business Critical tier: Data in memory-optimized tables must fit within the In-Memory OLTP storage that is associated with the tier of the database or the managed instance, or it is available in the elastic pool. If you try to scale-down the tier or move the database into a pool that doesn't have enough available In-Memory OLTP storage, the operation fails.

There is a programmatic way to understand whether a given database supports In-Memory OLTP. You can execute the following Transact-SQL query:

```
SELECT DatabasePropertyEx(DB_NAME(), 'IsXTPSupported');
```

If the query returns 1, In-Memory OLTP is supported in this database. The following queries identify all objects that need to be removed before a database can be downgraded to General Purpose, Standard, or Basic:

```
SELECT * FROM sys.tables WHERE is_memory_optimized=1
SELECT * FROM sys.table_types WHERE is_memory_optimized=1
SELECT * FROM sys.sql_modules WHERE uses_native_compilation=1
```

If customer unable to find memory objects in the above way, they can use below process to display and drop:

```
--Display all Memory Optimized objects
        select
                OBJECT_SCHEMA_NAME(object_id) as [schema],
                OBJECT_NAME(object_id) as [name],
                uses_native_compilation as is_memory_optimized,
                [type] = 'Natively compiled stored procedure'
        from sys.sql_modules
        where uses_native_compilation = 1
union
        select
                SCHEMA_NAME([schema_id]) AS [schema],
                [name],
                is_memory_optimized,
                [type] = 'Table'
        from sys.tables
        where is_memory_optimized = 1
union
        select
                SCHEMA_NAME([schema_id]) AS [schema],
                [name],
                is_memory_optimized,
                [type] = 'User Defined Table Type'
        from sys.table_types
        where is_memory_optimized = 1


--Create Drop statements for memory optimized objects
        select
                CONCAT('DROP PROCEDURE [', OBJECT_SCHEMA_NAME(object_id), '].[', OBJECT_NAME(object_id), '];')
        from sys.sql_modules
        where uses_native_compilation = 1
union
        select
                CONCAT('DROP TABLE [', SCHEMA_NAME([schema_id]), '].[', [name], '];')
        from sys.tables
        where is_memory_optimized = 1
union
        select
                CONCAT('DROP TYPE [', SCHEMA_NAME([schema_id]), '].[', [name], '];')
        from sys.table_types
        where is_memory_optimized = 1
```

Public Doc Reference

https://docs.microsoft.com/en-us/azure/sql-database/sql-database-in-memory

How good have you found this content?