Restore or Migrate database containing Memory Optimized objects to Azure SQL Managed Instance

Last updated by | Radhika Shah | Jul 29, 2022 at 5:23 PM PDT

Contents

Issue

Customer restoring database (or migrating database) containing memory optimized objects to General Purpose Azure SQL Managed Instance fails with error:

```
Restore of database '<abc>' failed. (Microsoft.SqlServer.Management.RelationalEngineTasks)
------------------------------
Program Location:
at Microsoft.SqlServer.Management.RelationalEngineTasks.RestoreDatabaseTaskFormComponent.PerformTask(ITaskExec
at Microsoft.SqlServer.Management.RelationalEngineTasks.RestoreDatabaseTaskFormComponent.Perform(ITaskExecutio
at Microsoft.SqlServer.Management.TaskForms.TaskExecutionManager.ExecuteTaskSequence(ISfcScriptCollector colle
==================================
System.Data.SqlClient.SqlError: Memory-optimized filegroup must be empty in order to be restored on General Pu
```

This could occur in different scenarios such as:

- Customer migrating from on-prem to Azure SQL Managed Instance General Purpose tier.
- Customer generated .bak from on-prem database and restoring to Azure SQL Managed Instance General Purpose tier.
- Customer scaling from Azure SQL Managed Instance Business Critical tier (that supports memory optimized objects) to Azure SQL Managed Instance General Purpose tier (that doesn't support memory optimized objects).

Investigation/Analysis

By design, it is not possible to have in-memory optimized objects in General Purpose instance. Check to see if customer has memory optimized tables or objects on the source database.

Investigation from CSS end

Check the failed restore attempt from ASC or via below kusto:

```
MonRestoreEvents
| where restore_request_id == <restore_id guid>
| where originalEventTimestamp >= {datetime}
| project process_id, originalEventTimestamp, restore_database_progress, message, details, LogicalServerName,
```

Sample Output:

process_id	originalEventTimestamp	restore_database_progress	message	details	LogicalServ
15192	2021-02-01 21:05:43.2287627		Memory-optimized filegroup must be empty in order to be restored on General Purpose tier of SQL Database Managed Instance.		<servernar

In the scenario where customer is scaling from Azure SQL Managed Instance Business Critical tier to Azure SQL Managed Instance General Purpose tier, check to see if the source MI DB (on BC) contains memory optimized objects.

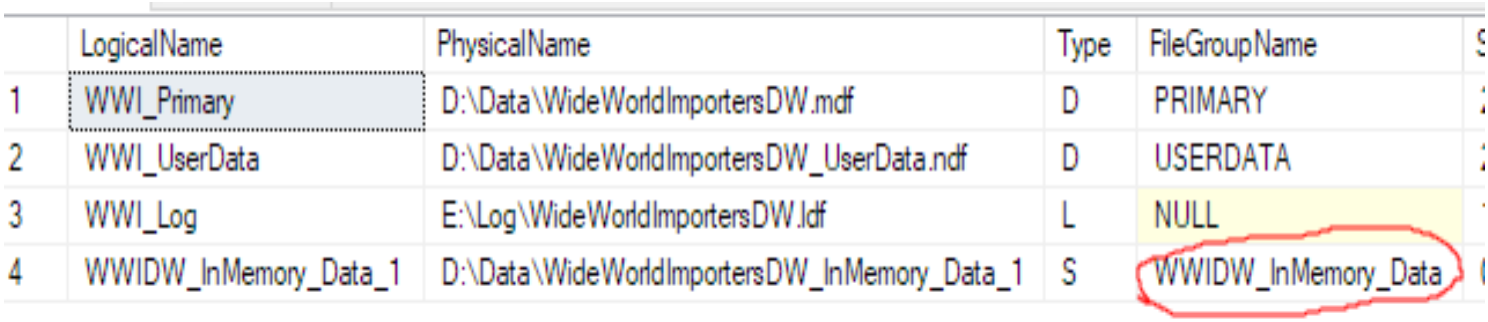
```
let database_metadata =
  MonDatabaseMetadata
  | where AppName == {AppName}
  | where TIMESTAMP > ago(2d)
  | extend
      is_b = AppName startswith "b-"
      , logical_database_name = logical_db_name
  | extend db_id = strcat(LogicalServerName, ";", logical_database_name, ";", is_b);
  database_metadata
  | where table_name == "sysschobjs" and type in ("U ", "IT", "ET")
  | where binary_and(status2, 8) != 0
  | project logical_database_name, table_name = name
  | distinct logical_database_name, table_name


MonDmRealTimeResourceStats
| where LogicalServerName == {MIServerName}
| where * contains "bd40df2e-11eb-41d3-a98e-8c30270c8f2f"
| take 1
| distinct database_name
```

Investigation from customer end

To check if the backup contains in-memory objects, customer can execute this:

```
RESTORE FILELISTONLY
FROM URL = 'https://<storage>.blob.core.windows.net/<container>/<backupfile>.bak'
```

Sample output:



	LogicalName	PhysicalName	Type	FileGroupName	S
1	WWI_Primary	D:\Data\WideWorldImportersDW.mdf	D	PRIMARY	2
2	WWI_UserData	D:\Data\WideWorldImportersDW_UserData.ndf	D	USERDATA	2
3	WWI_Log	E:\Log\WideWorldImportersDW.ldf	L	NULL	1
4	WWIDW_InMemory_Data_1	D:\Data\WideWorldImportersDW_InMemory_Data_1	S	WWIDW_InMemory_Data	0

Customer can then run below queries on their source database to check if there are any memory optimized tables or objects present.

```
--check for memory optimized tables
SELECT *
FROM sys.tables
WHERE is_memory_optimized=1
GO

--check for memory optimized table types that are considered memory optimized objects
SELECT *
FROM sys.table_types
WHERE is_memory_optimized=1
GO

SELECT *
FROM sys.sql_modules
WHERE uses_native_compilation=1
GO

SELECT object_name(object_id) ASName, *
FROM sys.dm_db_xtp_table_memory_stats
GO
```

Mitigation/Workaround

This behavior is by design. Customer cannot restore memory-optimized filegroups on General Purpose Edition. There are 2 options, either use Business Critical, or empty data from memory-optimized filegroups, take a new backup and that backup should be restorable to Azure SQL Managed Instance General Purpose tier.

Public Doc Reference

Restore limitations - SQL Managed Instance

Internal Reference

ICM 224786258

ICM 300111954

How good have you found this content?