

# Drop server timeout when CMK is used

Last updated by | Vitor Tomaz | Feb 24, 2023 at 3:24 AM PST

## Contents

- Issue
- Investigation/Analysis
- Mitigation
- RCA (optional)
- Internal Reference (optional)
- Root Cause Classification

## Issue

Customer tried to drop the SQL Server with CMK enabled using "Remove-AzResource" cmdlet which returned timeout, but finally completed in the backend.

## Investigation/Analysis

1. Check the operations from MonManagement kusto table and you should see the first request\_id failed with "GatewayTimeout" and retry finally succeeded.

PreciseTimeStamp	operationName	correlationId	status	subStatus	properties
2022-02-10 17:35:56.9764862	Microsoft.Sql/servers/delete	a561370a-2dc5-4d40-b64d-0cc2d31ff02c	Started		{"eventCategory": "Administrative", "entity": "/si
2022-02-10 17:36:57.0986468	Microsoft.Sql/servers/delete	a561370a-2dc5-4d40-b64d-0cc2d31ff02c	Failed	GatewayTimeout	{"statusCode": "GatewayTimeout", "serviceRequest:
2022-02-10 17:37:01.7286374	Microsoft.Sql/servers/delete	2e6285b5-3337-4fd7-b9dd-6746230b884a	Started		{"eventCategory": "Administrative", "entity": "/si
2022-02-10 17:37:01.9932818	Microsoft.Sql/servers/delete	2e6285b5-3337-4fd7-b9dd-6746230b884a	Failed	BadRequest	{"statusCode": "BadRequest", "serviceRequestId": "
2022-02-10 17:47:05.1434167	Microsoft.Sql/servers/delete	2e6285b5-3337-4fd7-b9dd-6746230b884a	Succeeded		{"eventCategory": "Administrative", "entity": "/si

2. From first timeout request, you can see the deadlock exception was thrown.

MonManagement

```
| where originalEventTimestamp >= datetime(2022-02-09 10:00:00) and originalEventTimestamp <= datetime(2022-0
| where request_id == "first_timeout_request_id"
| where message != "" or error_message != ""
| sort by originalEventTimestamp asc
| limit 1000
```

Exception looks like this: System.Data.SqlClient.SqlException (0x80131904): Transaction (Process ID 341) was deadlocked on lock resources with another process and has been chosen as the deadlock victim. Rerun the transaction. at System.Data.SqlClient.SqlDataReader.TryHasMoreRows(Boolean& moreRows) at System.Data.SqlClient.SqlDataReader.Read() at Microsoft.Xdb.Common.FiniteStateMachineBaseSqlContext.

```
<LoadAll>d__66 1.MoveNext() in
d:\dbs\sh\5ui5\0117_234354\cmd\4\Sql\xdb\common\fsm\Contexts\Sql\FiniteStateMachineBaseSqlContext.cs:line 1811
at Microsoft.Xdb.Common.FiniteStateMachineBaseSqlContext.LoadAllFromUdf(Type fsmType, FiniteStateMachineMap
metadata, String functionName, SqlParameter[] parameters, Object transaction) in
d:\dbs\sh\5ui5\0117_234354\cmd\4\Sql\xdb\common\fsm\Contexts\Sql\FiniteStateMachineBaseSqlContext.cs:line 1134
at Microsoft.Xdb.Common.FiniteStateMachineContext.<GetInstancesFromUdf>d__141 1.MoveNext() in
```

```
d:\dbs\sh\5uj5\0117_234354\cmd\4\Sql\xdb\common\fsm\FiniteStateMachineContext.cs:line 3334 at
Microsoft.SqlServer.Management.Service.SqlLogicalDatabaseStateMachine.GetNextSqlLogicalDatabaseBatchFor
LogicalServerDropUnderLock(IFiniteStateMachineContext ctx, String logicalServerName, Int32
batchSizeDatabaseCount) in
d:\dbs\sh\5uj5\0117_234354\cmd\w\sql\xdb\manifest\svc\mgmt\fsm\SqlLogicalDatabaseStateMachine.cs:line
4699 at
Microsoft.SqlServer.Management.Service.LogicalServerStateMachine.TransitionDropUserDatabases(ActionOutco
me& actionOutcome) in
d:\dbs\sh\5uj5\0117_234354\cmd\w\sql\xdb\manifest\svc\mgmt\fsm\Server\LogicalServerStateMachine.cs:line
8458 ClientConnectionId:e6b50d5d-a23a-4405-bbd3-e0617eacefe0 Error Number:1205,State:51,Class:13
```

## Mitigation

Nothing to mitigate.

## RCA (optional)

Remove-AzResource retrying the operation after a failure, in this case a GatewayTimeout (504) after 60 seconds is intended behavior, as per [Delete resource group and resources - Azure Resource Manager | Microsoft Docs](#) . Regarding why the request timed out, we see exceptions being thrown when the server is switch the state. Some of the transactions needed to drop the next batch of dbs with CMK enabled were deadlocked against other processes, which unfortunately led to delays. The Remove-AzResource call will continue to retry for up to 15 minutes, and if the underlying drop goes through it will eventually return a succeeded response.

We are working on improving the batch drop db experiences which will help reduce the delay in such scenario. We do appreciate you as a customer and we understand that you've tied success of your product to success of SQL Azure platform.

## Internal Reference (optional)

- [Incident 288361842](#) 

Repair Item: [1688874](#) 

## Root Cause Classification

Root Cause: /Azure SQL v3/CRUD/Server/Other

## How good have you found this content?

