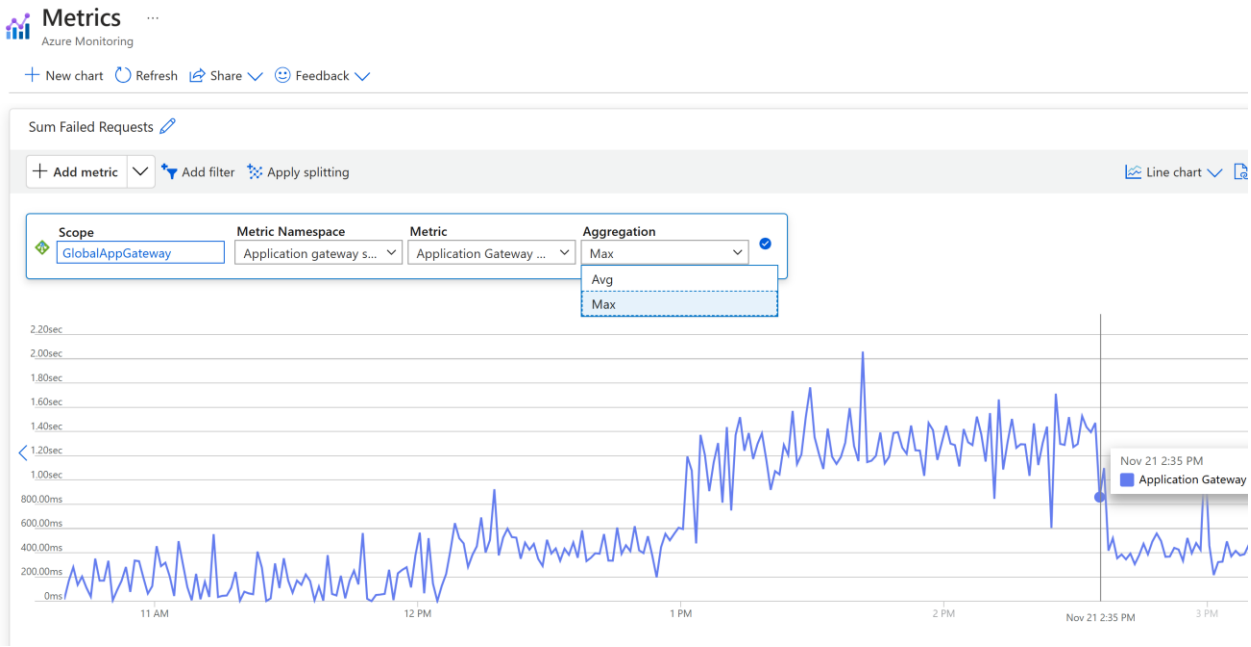
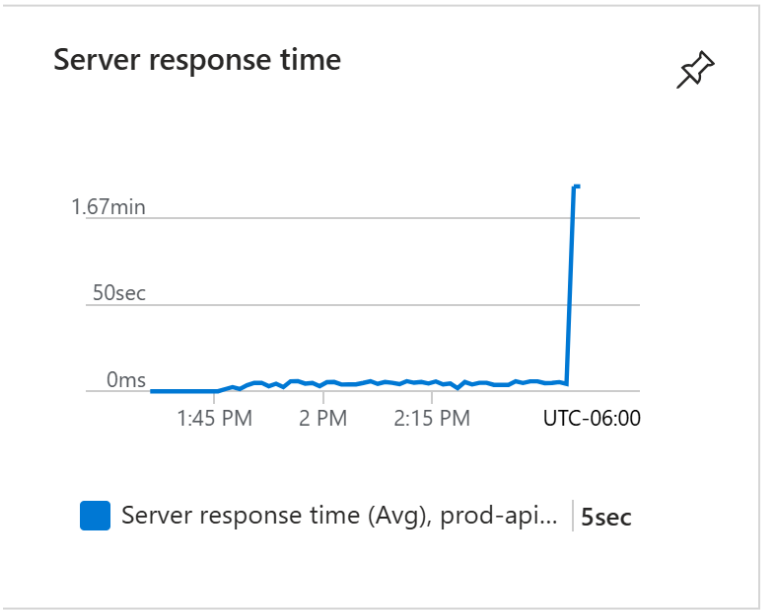
Connection pool exhaustion issue.

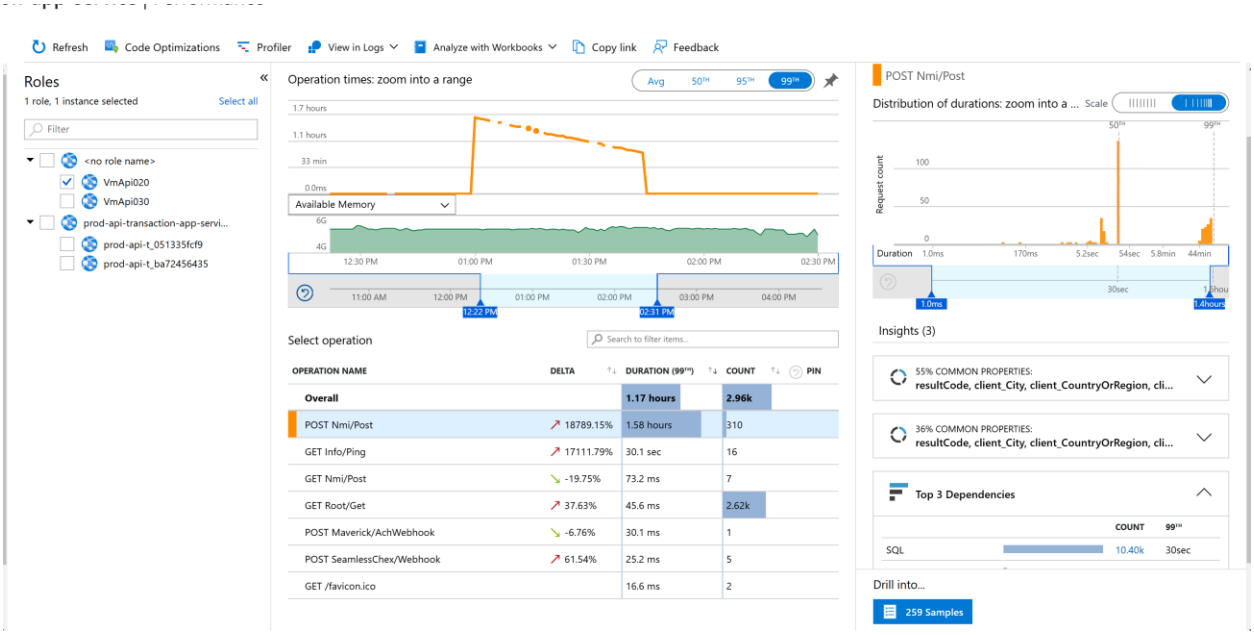
Time Frame: (1 PM to 2:35PM UTC-06)



Application Gateway / Time that it takes a request to be processed.



Same behavior from Transactions API Application Insights monitoring.



Timeout expired: The application couldn't obtain a database connection within the allocated timeout.

The error was observed across all the instances.

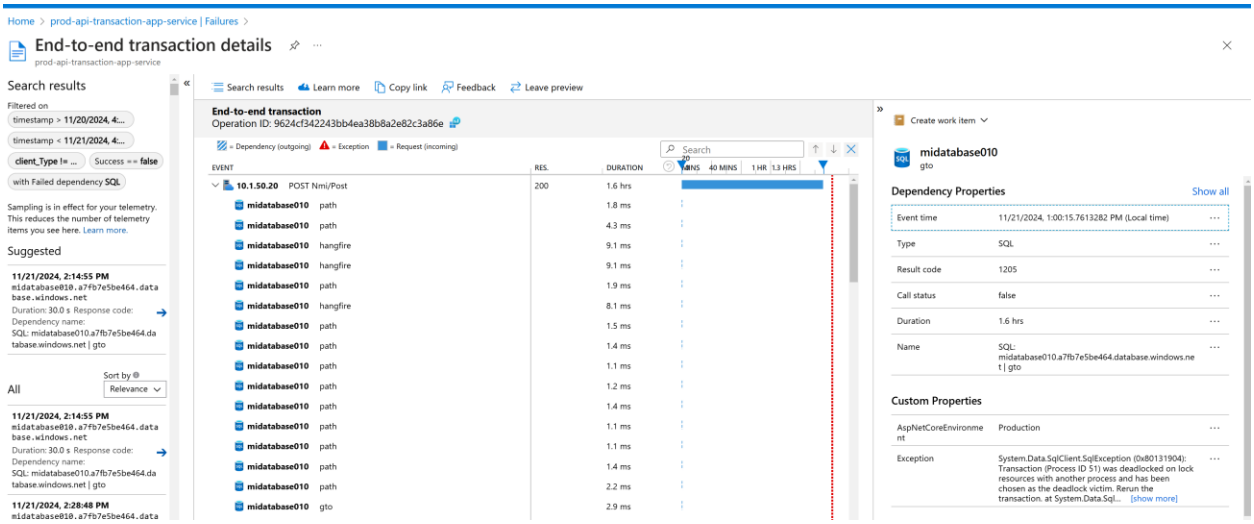
10.1.50.20 (VM)

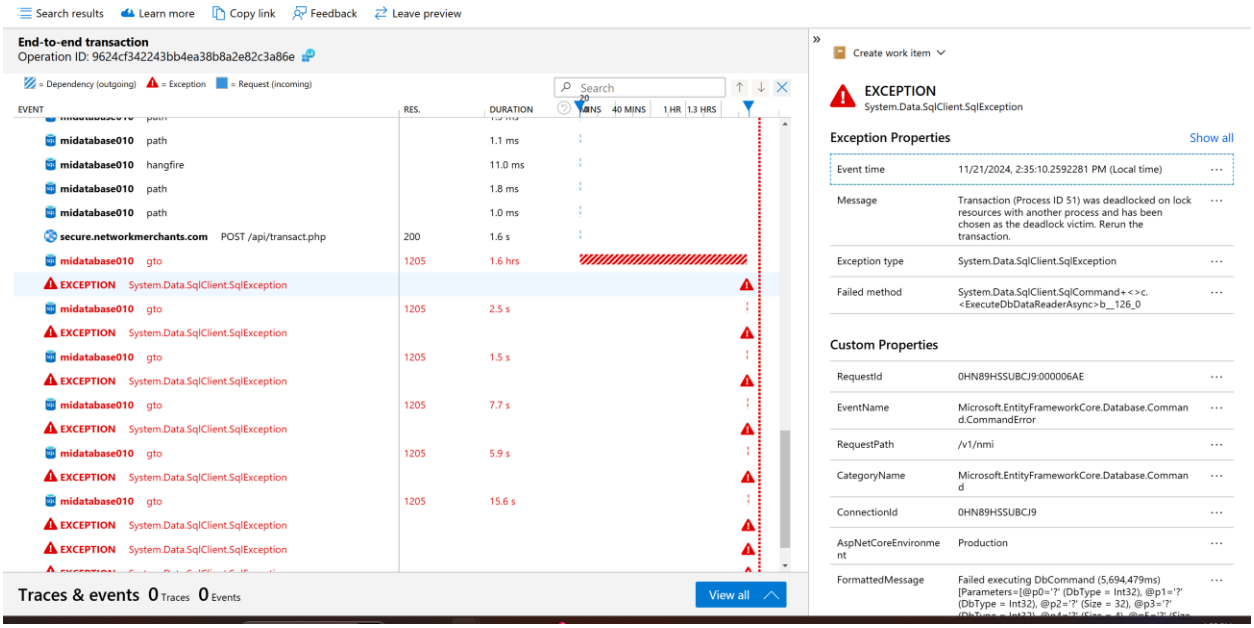
10.1.50.30 (VM)

10.1.50.40 (App Service)



Here’s an example of a query that took 1.6 hrs





Details:

Transaction (Process ID 51) was deadlocked on lock resources with another process and has been chosen as the deadlock victim. Rerun the transaction.

Failed executing DbCommand ({elapsed}ms) [Parameters=[{parameters}], CommandType='{commandType}', CommandTimeout='{commandTimeout}']{newLine}{commandText}

Failed executing DbCommand (5,694,479ms) [Parameters=[@p0='?' (DbType = Int32), @p1='?' (DbType = Int32), @p2='?' (Size = 32), @p3='?' (DbType = Int32), @p4='?' (Size = 4), @p5='?' (Size = 2048), @p6='?' (DbType = Int32), @p7='?' (Size = 32), @p8='?' (Size = 64), @p9='?' (Size = 16), @p10='?' (DbType = Int32), @p11='?' (DbType = Int64), @p12='?' (DbType = Int32), @p13='?' (DbType = Int32), @p14='?' (Size = 32), @p15='?' (Size = 32), @p16='?' (Size = 64), @p17='?' (DbType = Decimal), @p18='?' (DbType = Boolean), @p19='?' (DbType = Decimal), @p20='?' (DbType = Int32)], CommandType='Text', CommandTimeout='900000']

SET NOCOUNT ON;

INSERT INTO [Transactions] ([AccountId], [Approved], [AuthorizationCode], [CreditCardId], [Currency], [Description], [DeviceType], [EmployeeNumber], [InvoiceNumber], [IpAddress], [MarketType], [ParentTransactionId], [PaymentId], [PaymentMethodType], [PaymentTransactionId], [PoNumber], [ProcessingSystem], [Surcharge], [TaxExempt], [Total], [Type])

VALUES (@p0, @p1, @p2, @p3, @p4, @p5, @p6, @p7, @p8, @p9, @p10, @p11, @p12, @p13, @p14, @p15, @p16, @p17, @p18, @p19, @p20);

SELECT [TransactionId]

FROM [Transactions]

WHERE @@ROWCOUNT = 1 AND [TransactionId] = scope_identity();

Create work item

EXCEPTION

System.Data.SqlClient.SqlException

commandType	Text	···
commandTimeout	900000	···
commandText	SET NOCOUNT ON; INSERT INTO [Transactions] ([AccountId], [Approved], [AuthorizationCode], [CreditCardId], [Currency], [Description], [DeviceType], [EmployeeNumber], [InvoiceNumber], [IpAddress], [MarketType], [ParentTra... [show more]	···
elapsed	5,694,479	···
parameters	@p0='?' (DbType = Int32), @p1='?' (DbType = Int32), @p2='?' (Size = 32), @p3='?' (DbType = Int32), @p4='?' (Size = 4), @p5='?' (Size = 2048), @p6='?' (DbType = Int32), @p7='?' (Size = 32), @p8='?' (Size = 64), @p9='?' (S... [show more]	···

Action Name

Open.Path.Api.Transaction.Controllers.NmiController.Post (Open.Path.Api.Transaction)

Call Stack:

System.Data.SqlClient.SqlException:

at System.Data.SqlClient.SqlCommand+<>c.<ExecuteDbDataReaderAsync>b__126_0 (System.Data.SqlClient, Version=4.6.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a)

at System.Threading.Tasks.ContinuationResultTaskFromResultTask`2.InnerInvoke (System.Private.CoreLib, Version=4.0.0.0, Culture=neutral, PublicKeyToken=7cec85d7bea7798e)

at System.Threading.Tasks.Task+<>c.<.cctor>b__275_1 (System.Private.CoreLib, Version=4.0.0.0, Culture=neutral, PublicKeyToken=7cec85d7bea7798e)

at System.Threading.ExecutionContext.RunInternal (System.Private.CoreLib, Version=4.0.0.0, Culture=neutral, PublicKeyToken=7cec85d7bea7798e)

at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw (System.Private.CoreLib, Version=4.0.0.0, Culture=neutral, PublicKeyToken=7cec85d7bea7798e)

at System.Threading.ExecutionContext.RunInternal (System.Private.CoreLib, Version=4.0.0.0, Culture=neutral, PublicKeyToken=7cec85d7bea7798e)

at System.Threading.Tasks.Task.ExecuteWithThreadLocal (System.Private.CoreLib, Version=4.0.0.0, Culture=neutral, PublicKeyToken=7cec85d7bea7798e)

at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw (System.Private.CoreLib, Version=4.0.0.0, Culture=neutral, PublicKeyToken=7cec85d7bea7798e)

at System.Runtime.CompilerServices.TaskAwaiter.ThrowForNonSuccess (System.Private.CoreLib, Version=4.0.0.0, Culture=neutral, PublicKeyToken=7cec85d7bea7798e)

at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification (System.Private.CoreLib, Version=4.0.0.0, Culture=neutral, PublicKeyToken=7cec85d7bea7798e)

at System.Runtime.CompilerServices.TaskAwaiter`1.GetResult (System.Private.CoreLib, Version=4.0.0.0, Culture=neutral, PublicKeyToken=7cec85d7bea7798e)

at Microsoft.EntityFrameworkCore.Storage.Internal.RelationalCommand+<ExecuteAsync>d__17.MoveNext (Microsoft.EntityFrameworkCore.Relational, Version=2.2.6.0, Culture=neutral, PublicKeyToken=adb9793829ddae60)

Collateral Errors:

Maximum number of retries (5) exceeded while executing database operations with 'SqlServerRetryingExecutionStrategy'. See inner exception for the most recent failure. An error occurred while updating the entries. See the inner exception for details. Transaction (Process ID 100) was deadlocked on lock resources with another process and has been chosen as the deadlock victim. Rerun the transaction.

An exception occurred in the database while saving changes for context type '{contextType}'.{newline}{error}

Operation ID: 9624cf342243bb4ea38b8a2e82c3a86e

Other Common Errors during this timeframe

System.InvalidOperationException: Timeout expired. The timeout period elapsed prior to obtaining a connection from the pool. This may have occurred because all pooled connections were in use and max pool size was reached.

at System.Data.ProviderBase.DbConnectionFactory.TryGetConnection(DbConnection owningConnection, TaskCompletionSource`1 retry, DbConnectionOptions userOptions, DbConnectionInternal oldConnection, DbConnectionInternal& connection)

at System.Data.ProviderBase.DbConnectionInternal.TryOpenConnectionInternal(DbConnection outerConnection, DbConnectionFactory connectionFactory, TaskCompletionSource`1 retry, DbConnectionOptions userOptions)

at System.Data.ProviderBase.DbConnectionClosed.TryOpenConnection(DbConnection outerConnection, DbConnectionFactory connectionFactory, TaskCompletionSource`1 retry, DbConnectionOptions userOptions)

at System.Data.SqlClient.SqlConnection.TryOpen(TaskCompletionSource`1 retry)

at System.Data.SqlClient.SqlConnection.Open()

at Microsoft.EntityFrameworkCore.Storage.RelationalConnection.OpenDbConnection(Boolean errorsExpected)

Cannot insert explicit value for identity column in table 'CreditCardAddressPhones' when IDENTITY_INSERT is set to OFF.

Transaction (Process ID 388) was deadlocked on lock resources with another process and has been chosen as the deadlock victim. Rerun the transaction.

System.Data.SqlClient.SqlException (0x80131904): Cannot insert explicit value for identity column in table 'CreditCardAddressPhones' when IDENTITY_INSERT is set to OFF.

Transaction (Process ID 419) was deadlocked on lock resources with another process and has been chosen as the deadlock victim. Rerun the transaction.

at System.Data.SqlClient.SqlConnection.OnError(SqlException exception, Boolean breakConnection, Action`1 wrapCloseInAction)

at System.Data.SqlClient.SqlInternalConnection.OnError(SqlException exception, Boolean breakConnection, Action`1 wrapCloseInAction)

at System.Data.SqlClient.TdsParser.ThrowExceptionAndWarning(TdsParserStateObject stateObj, Boolean callerHasConnectionLock, Boolean asyncClose)

at System.Data.SqlClient.TdsParser.TryRun(RunBehavior runBehavior, SqlCommand cmdHandler, SqlDataReader dataStream, BulkCopySimpleResultSet bulkCopyHandler, TdsParserStateObject stateObj, Boolean& dataReady)

at System.Data.SqlClient.SqlDataReader.TryConsumeMetaData()

at System.Data.SqlClient.SqlDataReader.get_MetaData()

at System.Data.SqlClient.SqlCommand.FinishExecuteReader(SqlDataReader ds, RunBehavior runBehavior, String resetOptionsString)

at System.Data.SqlClient.SqlCommand.CompleteAsyncExecuteReader()

at System.Data.SqlClient.SqlCommand.InternalEndExecuteReader(IAsyncResult asyncResult, String endMethod)

at System.Data.SqlClient.SqlCommand.EndExecuteReaderInternal(IAsyncResult asyncResult)

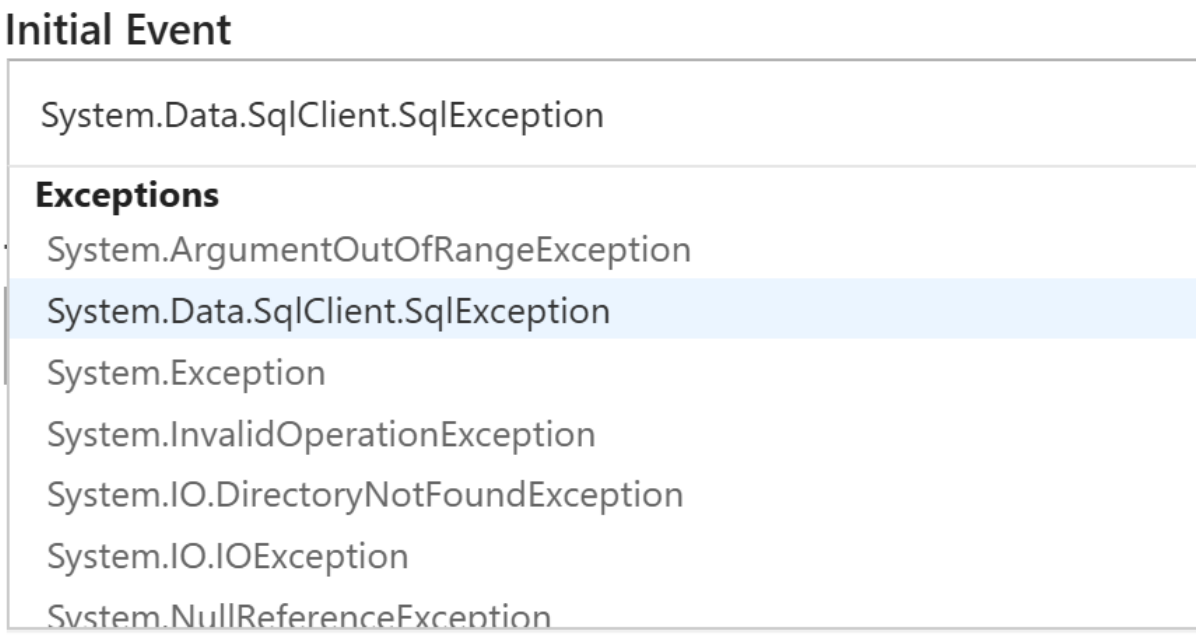
at System.Data.SqlClient.SqlCommand.EndExecuteReader(IAsyncResult asyncResult)

at System.Threading.Tasks.TaskFactory`1.FromAsyncCoreLogic(IAsyncResult iar, Func`2 endFunction, Action`1 endAction, Task`1 promise, Boolean requiresSynchronization)

ClientConnectionId:e632409f-3d88-4f6f-bb93-df300e172d06

Error Number:544,State:1,Class:16

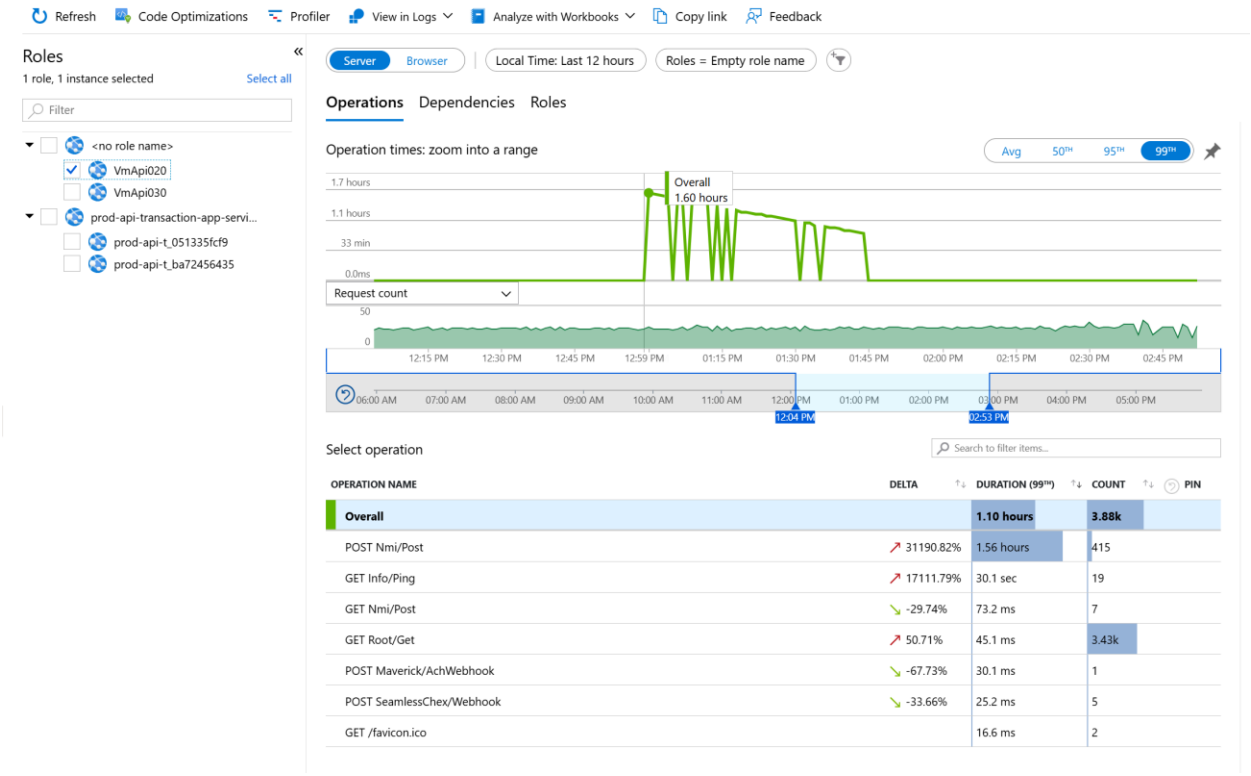
List of exceptions captured.



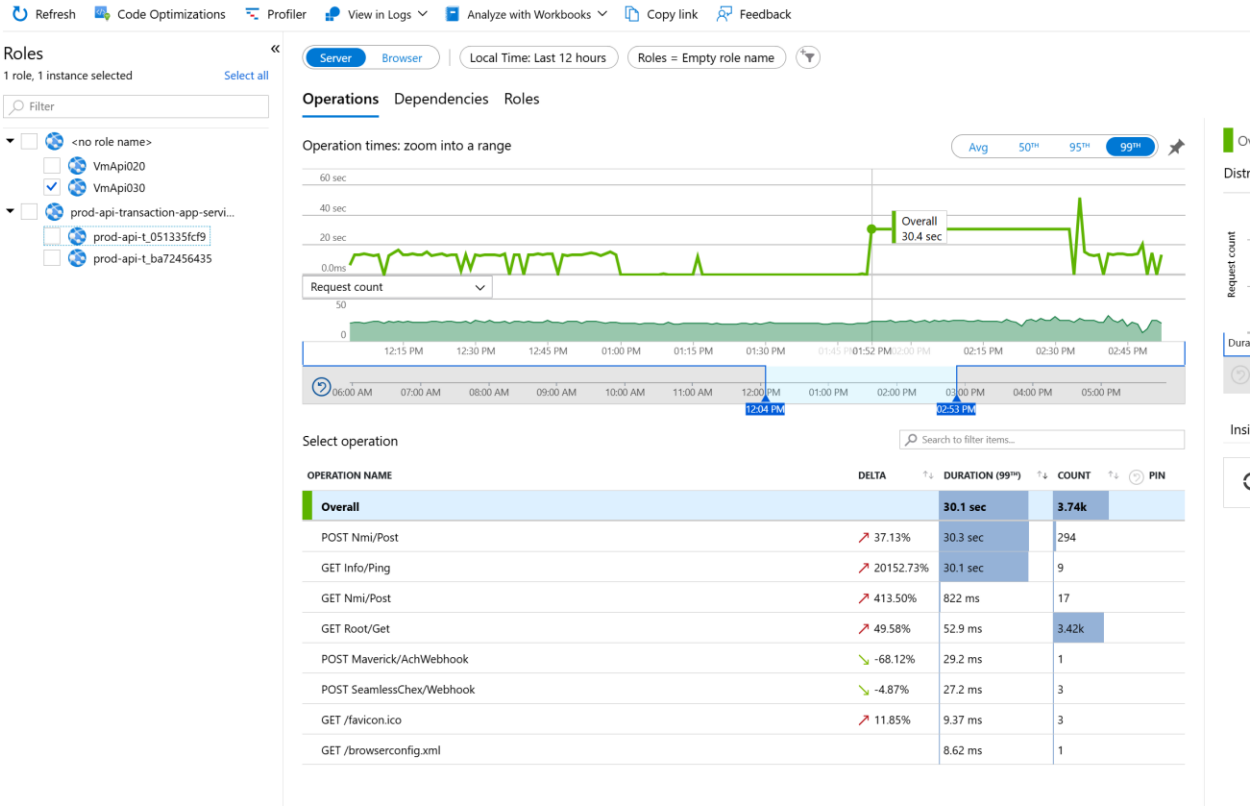
Last finding:

The first slow query, which might have triggered the issue, occurred at 12:59 UTC-06. This query, was executed from **VmApi020**, took approximately 1.6 hours to execute, leading to a deadlock. Then, as a collateral effect, the VmApi030 and the AppService were unable to access the GTO database. Based on this, I believe the issue is likely related to the slow query from VmApi020.

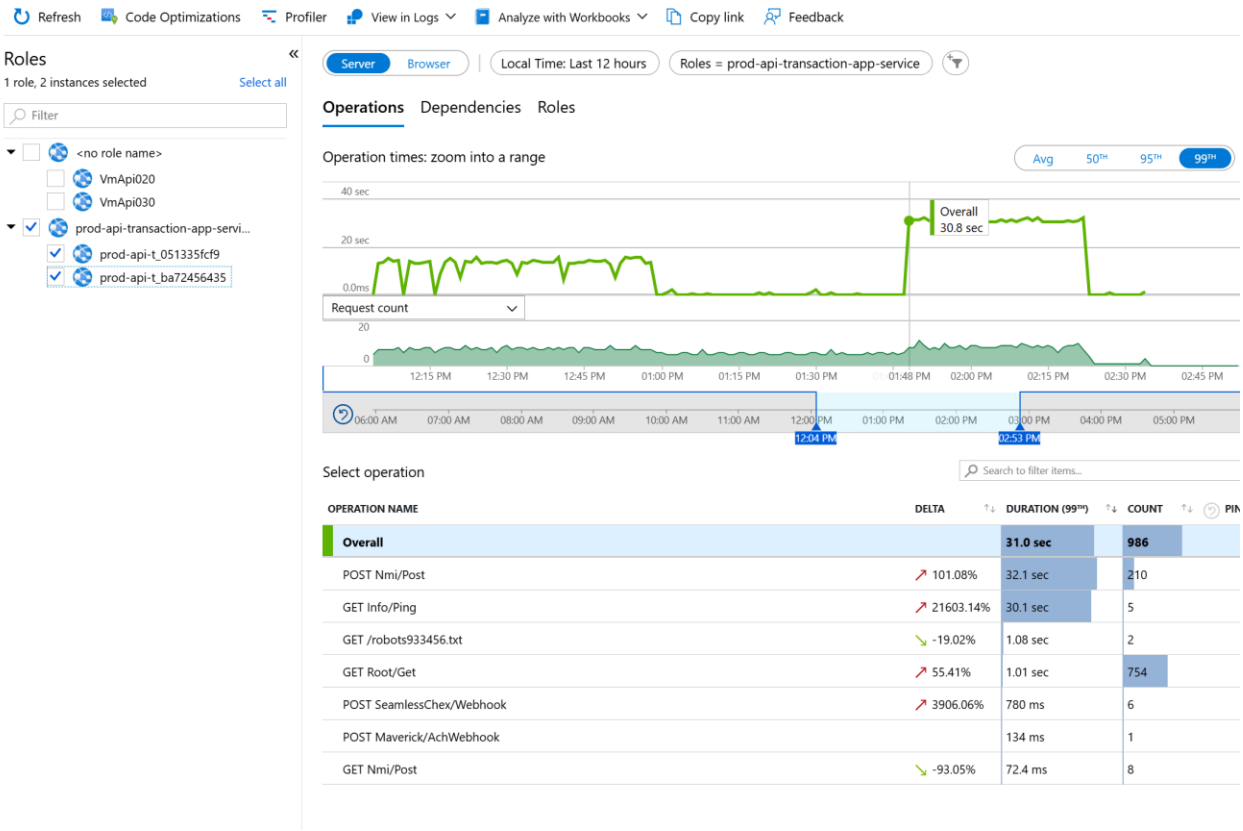
First Slow request 12:59 1.6 hours from **VmApi020**

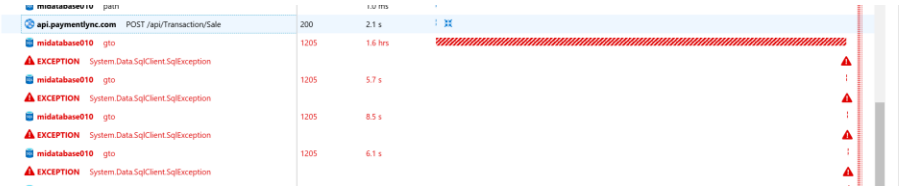


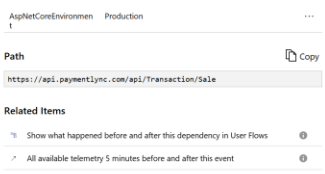
First slow request in **VmApi030** 1:52 PM (UTC-6) 30 seconds.,

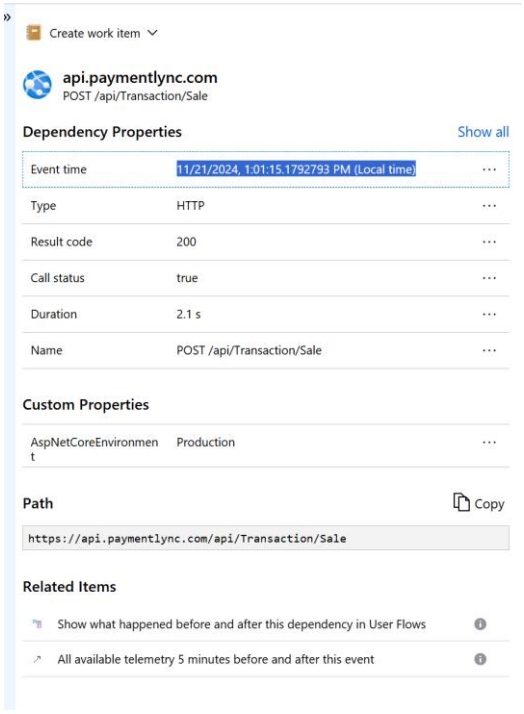


Same for the **App Service**. First Slow query 01:48 PM 30 seconds.




api.paymentlync.com POST /api/Transaction/Sale 200 2.1 s
midatabase010 gto 1205 1.6 hrs
EXCEPTION System.Data.SqlClient.SqlException
midatabase010 gto 1205 5.7 s
EXCEPTION System.Data.SqlClient.SqlException
midatabase010 gto 1205 8.5 s
EXCEPTION System.Data.SqlClient.SqlException
midatabase010 gto 1205 6.1 s
EXCEPTION System.Data.SqlClient.SqlException


AspNetCoreEnvironmen t Production
Path
Copy
https://api.paymentlync.com/api/Transaction/Sale
Related Items
Show what happened before and after this dependency in User Flows
All available telemetry 5 minutes before and after this event


Create work item
api.paymentlync.com
POST /api/Transaction/Sale
Dependency Properties
Show all
Event time 11/21/2024, 1:01:15.1792793 PM (Local time)
Type HTTP
Result code 200
Call status true
Duration 2.1 s
Name POST /api/Transaction/Sale
Custom Properties
AspNetCoreEnvironmen t Production
Path
Copy
https://api.paymentlync.com/api/Transaction/Sale
Related Items
Show what happened before and after this dependency in User Flows
All available telemetry 5 minutes before and after this event

Sentry Exception Details regarding the same transaction that got stuck for ~1 hour.

https://openpath-inc.sentry.io/issues/5552424077/?project=5762069&query=is%3Aunresolved%20issue.priority%3A%5Bhigh%2C%20medium%5D&referrer=issue-stream&statsPeriod=24h&stream_index=5

Error:

System.Data.SqlClient.SqlException: Transaction (Process ID 366) was deadlocked on lock resources with another process and has been chosen as the deadlock victim. Rerun the transaction.

This query here took ~60 minutes

Executed DbCommand (3,684,281ms) [Parameters=[@p0='?' (Size = 512), @p1='?' (Size = 512), @p2='?' (Size = 128), @p3='?' (Size = 128), @p4='?' (Size = 128), @p5='?' (DbType = DateTime2), @p6='?' (DbType = Int32), @p7='?' (DbType = Decimal), @p8='?' (DbType = Decimal), @p9='?' (Size = 128), @p10='?' (Size = 128), @p11='?' (Size = 32), @p12='?' (Size = 128), @p13='?' (DbType = DateTime2), @p15='?' (DbType = Int32), @p14='?' (Size = 256), @p17='?' (DbType = Int32), @p16='?' (DbType = DateTime2), @p19='?' (DbType = Int32), @p18='?' (DbType = DateTime2), @p20='?' (DbType = Int32), @p21='?' (DbType = Int32), @p22='?' (Size = 32), @p23='?' (DbType = Int32), @p24='?' (Size = 4), @p25='?' (Size = 2048), @p26='?' (DbType = Int32), @p27='?' (Size = 32), @p28='?' (Size = 64), @p29='?' (Size = 16), @p30='?' (DbType = Int32), @p31='?' (DbType = Int64), @p32='?' (DbType = Int32), @p33='?' (DbType = Int32), @p34='?' (Size = 32), @p35='?' (Size = 32), @p36='?' (Size = 64), @p37='?' (DbType = Decimal), @p38='?' (DbType = Boolean), @p39='?' (DbType = Decimal), @p40='?' (DbType = Int32)], CommandType='Text', CommandTimeout='900000']

SET NOCOUNT ON;

INSERT INTO [CreditCardAddresses] ([Address1], [Address2], [City], [Country], [CountyRegion], [Created], [CreditCardId], [Latitude], [Longitude], [NormalizedAddress], [Organization], [PostalZipCode], [ProvenceState], [Updated])

VALUES (@p0, @p1, @p2, @p3, @p4, @p5, @p6, @p7, @p8, @p9, @p10, @p11, @p12, @p13);

SELECT [CreditCardAddressId]

```
FROM [CreditCardAddresses]

WHERE @@ROWCOUNT = 1 AND [CreditCardAddressId] = scope_identity();


UPDATE [CreditCards] SET [CardCodeCore] = @p14

WHERE [CreditCardId] = @p15;

SELECT @@ROWCOUNT;


UPDATE [CustomerAddresses] SET [Updated] = @p16

WHERE [CustomerAddressId] = @p17;

SELECT @@ROWCOUNT;


UPDATE [Customers] SET [Updated] = @p18

WHERE [CustomerId] = @p19;

SELECT @@ROWCOUNT;


INSERT INTO [Transactions] ([AccountId], [Approved], [AuthorizationCode], [CreditCardId],
[Currency], [Description], [DeviceType], [EmployeeNumber], [InvoiceNumber], [IpAddress],
[MarketType], [ParentTransactionId], [PaymentId], [PaymentMethodType],
[PaymentTransactionId], [PoNumber], [ProcessingSystem], [Surcharge], [TaxExempt],
[Total], [Type])

VALUES (@p20, @p21, @p22, @p23, @p24, @p25, @p26, @p27, @p28, @p29, @p30,
@p31, @p32, @p33, @p34, @p35, @p36, @p37, @p38, @p39, @p40);

SELECT [TransactionId]

FROM [Transactions]

WHERE @@ROWCOUNT = 1 AND [TransactionId] = scope_identity();

{
```

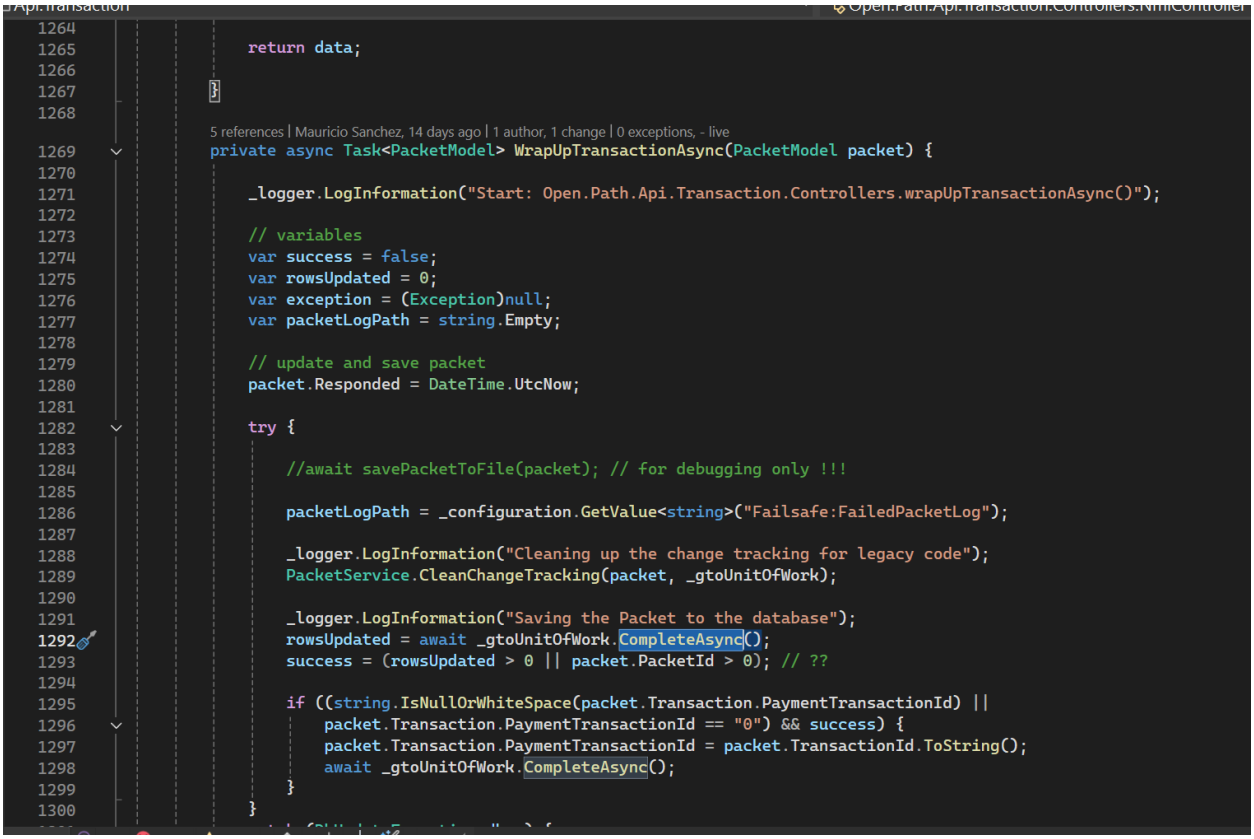
Steps before the failure:

Open.Path.Api.Transaction.Controllers.NmiController info 07:33:50.709 PM

Saving the Packet to the database

Sentry log.

Code Reference:



```
1264
1265            return data;
1266
1267        }
1268
                5 references | Mauricio Sanchez, 14 days ago | 1 author, 1 change | 0 exceptions, - live
1269        private async Task<PacketModel> WrapUpTransactionAsync(PacketModel packet) {
1270
1271            _logger.LogInformation("Start: Open.Path.Api.Transaction.Controllers.wrapUpTransactionAsync()");
1272
1273            // variables
1274            var success = false;
1275            var rowsUpdated = 0;
1276            var exception = (Exception)null;
1277            var packetLogPath = string.Empty;
1278
1279            // update and save packet
1280            packet.Responded = DateTime.UtcNow;
1281
1282            try {
1283
1284                //await savePacketToFile(packet); // for debugging only !!!
1285
1286                packetLogPath = _configuration.GetValue<string>("Failsafe:FailedPacketLog");
1287
1288                _logger.LogInformation("Cleaning up the change tracking for legacy code");
1289                PacketService.CleanChangeTracking(packet, _gtoUnitOfWork);
1290
1291                _logger.LogInformation("Saving the Packet to the database");
1292                rowsUpdated = await _gtoUnitOfWork.CompleteAsync();
1293                success = (rowsUpdated > 0 || packet.PacketId > 0); // ??
1294
1295                if ((string.IsNullOrWhiteSpace(packet.Transaction.PaymentTransactionId) ||
1296                    packet.Transaction.PaymentTransactionId == "0") && success) {
1297                    packet.Transaction.PaymentTransactionId = packet.TransactionId.ToString();
1298                    await _gtoUnitOfWork.CompleteAsync();
1299                }
1300            }
```



Saved 29 to the GTO database.

Code Reference:

```
    }

    8 references | 0 changes | 0 authors, 0 changes | 0 exceptions, - live
    public async Task<int> CompleteAsync() {

        var rowsUpdates = await _context.SaveChangesAsync();

        _logger.LogInformation($"Saved {rowsUpdates} to the GTO database");

        return rowsUpdates;

    }
```

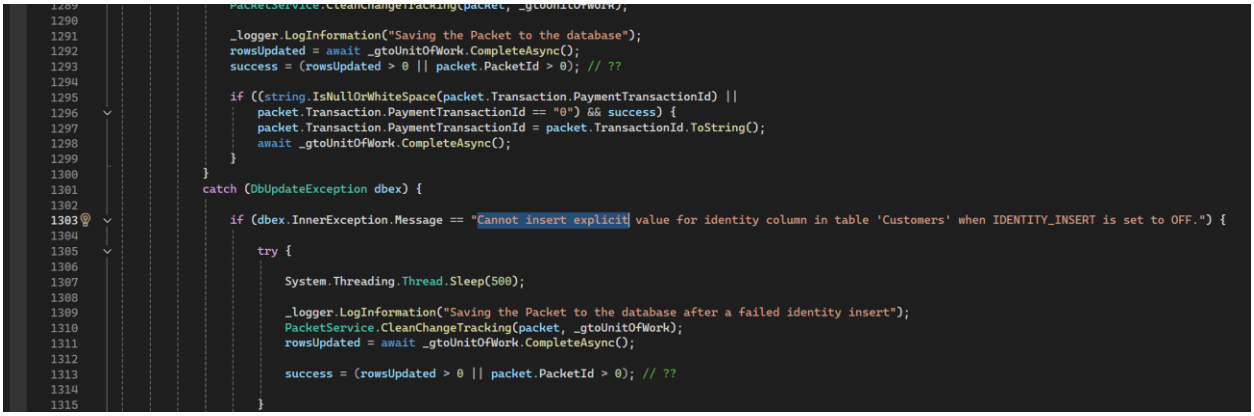
GtoDynamicUnitOfWork.cs saved 29 records successfully.

Then it got stuck for ~60 minutes and failed.

```
1289                PacketService.CleanChangeTracking(packet, _gtoUnitOfWork);
1290
1291                _logger.LogInformation("Saving the Packet to the database");
1292                rowsUpdated = await _gtoUnitOfWork.CompleteAsync();
1293                success = (rowsUpdated > 0 || packet.PacketId > 0); // ??
1294
1295                if ((string.IsNullOrWhiteSpace(packet.Transaction.PaymentTransactionId) ||
1296                    packet.Transaction.PaymentTransactionId == "0") && success) {
1297                    packet.Transaction.PaymentTransactionId = packet.TransactionId.ToString();
1298                    await _gtoUnitOfWork.CompleteAsync();
1299                }
1300            }
            catch (DbUpdateException dbex) {
```

Based on the errors captured from Application Insights, and Sentry.

We can assume it failed in line 1298 of method WrapUpTransactionAsync, NmiController.cs



```
1289                PacketService.CleanChangeTracking(packet, _gtoUnitOfWork);
1290
1291                _logger.LogInformation("Saving the Packet to the database");
1292                rowsUpdated = await _gtoUnitOfWork.CompleteAsync();
1293                success = (rowsUpdated > 0 || packet.PacketId > 0); // ??
1294
1295                if ((string.IsNullOrWhiteSpace(packet.Transaction.PaymentTransactionId) ||
1296                    packet.Transaction.PaymentTransactionId == "0") && success) {
1297                    packet.Transaction.PaymentTransactionId = packet.TransactionId.ToString();
1298                    await _gtoUnitOfWork.CompleteAsync();
1299                }
1300            }
1301            catch (DbUpdateException dbex) {
1302
1303                if (dbex.InnerException.Message == "Cannot insert explicit value for identity column in table 'Customers' when IDENTITY_INSERT is set to OFF.") {
1304
1305                    try {
1306
1307                        System.Threading.Thread.Sleep(500);
1308
1309                        _logger.LogInformation("Saving the Packet to the database after a failed identity insert");
1310                        PacketService.CleanChangeTracking(packet, _gtoUnitOfWork);
1311                        rowsUpdated = await _gtoUnitOfWork.CompleteAsync();
1312
1313                        success = (rowsUpdated > 0 || packet.PacketId > 0); // ??
1314
1315                    }
```

The Catch area captured thar error message in different scenarios.

Cannot insert explicit value for identity column in table 'CreditCardAddressPhones' when IDENTITY_INSERT is set to OFF.

Transaction (Process ID 388) was deadlocked on lock resources with another process and has been chosen as the deadlock victim. Rerun the transaction.

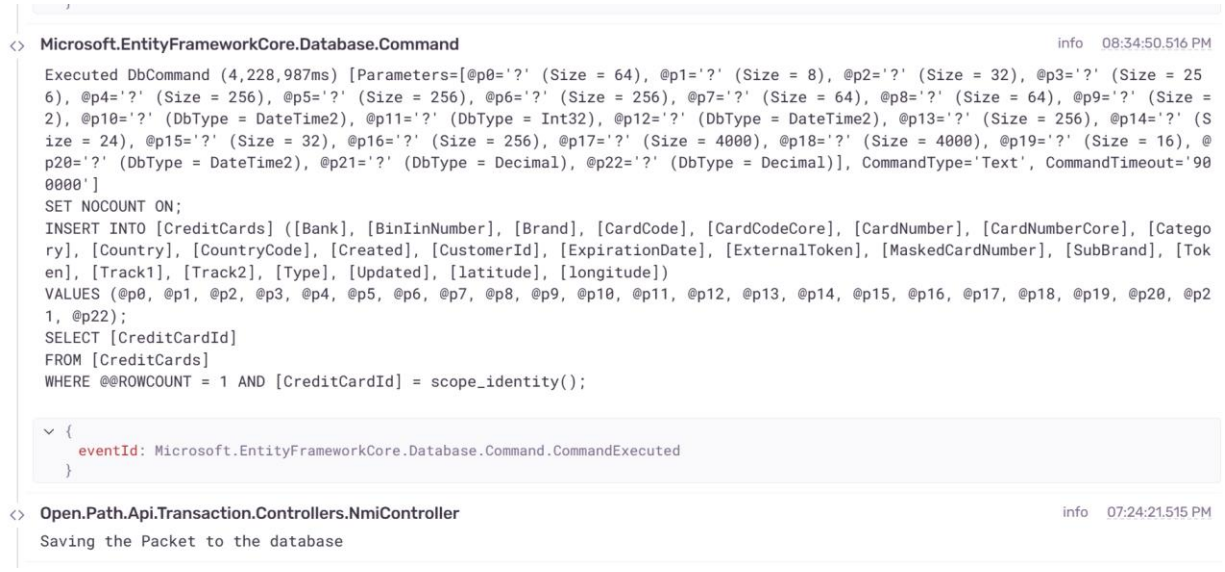
System.Data.SqlClient.SqlException (0x80131904): Cannot insert explicit value for identity column in table 'CreditCardAddressPhones' when IDENTITY_INSERT is set to OFF.

Transaction (Process ID 419) was deadlocked on lock resources with another process and has been chosen as the deadlock victim. Rerun the transaction.

There was an error saving the Packet to the database

This other log, has more details about the particular package:

https://sentry.io/organizations/openpath-inc/issues/6066344671/?project=5762069&query=is%3Aunresolved%20issue.priority%3A%5Bhigh%2C%20medium%5D&referrer=issue-stream&statsPeriod=14d&stream_index=7



Microsoft.EntityFrameworkCore.Database.Command info 08:34:50.516 PM

```
Executed DbCommand (4,228,987ms) [Parameters=[@p0='?' (Size = 64), @p1='?' (Size = 8), @p2='?' (Size = 32), @p3='?' (Size = 25
6), @p4='?' (Size = 256), @p5='?' (Size = 256), @p6='?' (Size = 256), @p7='?' (Size = 64), @p8='?' (Size = 64), @p9='?' (Size =
2), @p10='?' (DbType = DateTime2), @p11='?' (DbType = Int32), @p12='?' (DbType = DateTime2), @p13='?' (Size = 256), @p14='?' (S
ize = 24), @p15='?' (Size = 32), @p16='?' (Size = 256), @p17='?' (Size = 4000), @p18='?' (Size = 4000), @p19='?' (Size = 16), @
p20='?' (DbType = DateTime2), @p21='?' (DbType = Decimal), @p22='?' (DbType = Decimal)], CommandType='Text', CommandTimeout='90
0000']
SET NOCOUNT ON;
INSERT INTO [CreditCards] ([Bank], [BinIinNumber], [Brand], [CardCode], [CardCodeCore], [CardNumber], [CardNumberCore], [Catego
ry], [Country], [CountryCode], [Created], [CustomerId], [ExpirationDate], [ExternalToken], [MaskedCardNumber], [SubBrand], [Tok
en], [Track1], [Track2], [Type], [Updated], [latitude], [longitude])
VALUES (@p0, @p1, @p2, @p3, @p4, @p5, @p6, @p7, @p8, @p9, @p10, @p11, @p12, @p13, @p14, @p15, @p16, @p17, @p18, @p19, @p20, @p2
1, @p22);
SELECT [CreditCardId]
FROM [CreditCards]
WHERE @@ROWCOUNT = 1 AND [CreditCardId] = scope_identity();
```

```
{
  eventId: Microsoft.EntityFrameworkCore.Database.Command.CommandExecuted
}
```

Open.Path.Api.Transaction.Controllers.NmiController info 07:24:21.515 PM

```
Saving the Packet to the database
```

requestPayload.json="action=SALE&amount=3774&cardCVV=666&cardExpiryMonth=6&cardExpiryYear=26&cardNumber=4921817949564657&currencyCode=GBP&customerAddress=511+Ridgacre+Road+West&customerCountryCode=GB&customerEmail=lyndascarlett%40live.co.uk&customerName=Lynda+Shellis&customerPhone=&customerPostcode=B32+1AT&customerTown=Birmingham&deviceAcceptCharset=utf-8&deviceAcceptContent=%2A%2F%2A&deviceAcceptEncoding=gzip%2C+deflate%2C+br&deviceAcceptLanguage=en-GB%2Cen%3Bq%3D0.9&deviceCapabilities=javascript&deviceChannel=browser&deviceIdentity=Mozilla%2F5.0+%28iPhone%3B+CPU+iPhone+OS+15_8_3+like+Mac+OS+X%29+AppleWebKit%2F605.1.15+%28KHTML%2C+like+Gecko%29+Version%2F15.6.6+Mobile%2F15E148+Safari%2F604.1&deviceScreenResolution=414x736x32&deviceTimeZone=0&merchantID=235173&orderRef=r5wK4PZdxSYKtZ1pPQc2RTLbr&remoteAddress=2.216.231.112&threeDSRedirectURL=https%3A%2F%2Fhempelf.com%2Fa%2Fopenpath%2Fthreeds%2Fcallback%2Fr5wK4PZdxSYKtZ1pPQc2RTLbr%2F2355%2Fhempelf.com%3Facs%3D1&transactionUnique=19c56536-1a8c-4d98-b449-3a4ada911936&type=1&signature=012addaed80f1c66f9b6a6dce70b399832b82016a5cab83f65bdd5961f46c055c1ae2db6d6da89c9e9b85532d0f96da78bdc62b66631914483bf0f09805559e2"

responsePayload.json="responseCode=65802&responseMessage=3DS+authentication+required&responseStatus=2&merchantWebsite=https%3A%2F%2Fhempelf.com&merchantCategoryCode=5499&merchantID=235173&cardCVVMandatory=Y&threeDSEnabled=Y&threeDSCheckPref=attempted+authentication%2Cauthenticated&threeDSPolicy=4&riskCheckEnabled=N&avscv2CheckEnabled=N&surchargeEnabled=N&notifyEmail=farzad%40winchestermd.com&notifyEmailRequired=Y&customerReceiptsRequired=N&eReceiptsEnabled=N&__wafRequestID=2024-11-21T19%3A24%3A20Z%7C8b33914958%7C52.157.238.129%7CwRjeE7dPur&action=SALE&amount=3774&cardExpiryMonth=06&cardExpiryYear=26&currencyCode=826&customerAddress=511+Ridgacre+Road+West&customerCountryCode=GB&customerEmail=lyndascarlett%40live.co.uk&customerName=Lynda+Shellis&customerPostcode=B32+1AT&customerTown=Birmingham&deviceAcceptCharset=utf-8&deviceAcceptContent=%2A%2F%2A&deviceAcceptEncoding=gzip%2C+deflate%2C+br&deviceAcceptLanguage=en-GB%2Cen%3Bq%3D0.9&deviceCapabilities=javascript&deviceChannel=browser&deviceIdentity=Mozilla%2F5.0+%28iPhone%3B+CPU+iPhone+OS+15_8_3+like+Mac+OS+X%29+AppleWebKit%2F605.1.15+%28KHTML%2C+like+Gecko%29+Version%2F15.6.6+Mobile%2F15E148+Safari%2F604.1&deviceScreenResolution=414x736x32&deviceTimeZone=0&orderRef=r5wK4PZdxSYKtZ1pPQc2RTLbr&remoteAddress=2.216.231.112&threeDSRedirectURL=https%3A%2F%2Fhempelf.com%2Fa%2Fopenpath%2Fthreeds%2Fcallback%2Fr5wK4PZdxSYKtZ1pPQc2RTLbr%2F2355%2Fhempelf.com%3Facs%3D1&transactionUnique=19c56536-1a8c-4d98-b449-3a4ada911936&type=1&merchantCountryCode=GB&merchantCounty=ENG&merchantAddress=9+Chester+Place&merchantPhone=447970227204&merchantPostcode=NW1+4NB&subMerchantID=262960000008146&countryCode=826&requestorChallengeIndicator=04&facilitatorName=PIX&facilitatorID=10084515&merchantTown=London++447970227204&deviceIpAddress=2.216.231.112&requestID=673f88e4e614b&initiator=consumer&state=received&requestMerchantID=235173&processMerchantID=235173&paymentMethod=card&cardType=Visa+Debit&cardTypeCode=VD&cardScheme=Visa&cardSchemeCode=VC&cardIssuer=LLOYDS+BANK+PLC&cardIssuerCountry=United+Kingdom&cardIssuerCountryCode=GBR&cardFlags=272564225&cardNumberMask=492181%2A%2A%2A%2A%2A%2A4657&cardNumberValid=Y&xref=24112119TN24CQ20TY98CPN&cardExpiryDate=0626&threeDSVersion=2.2.0&threeDSEnrolled=Y&threeDSXID=26f20ebc-dbe6-42a6-b26e-9fc15bcbb4e1&threeDSRef=UDNLRVk6dHJhbnNhY3Rpb25JRD0zNjI4NDE2NTEmbWVyY2hhbnRJRD0yMzUxNzMmX19saWZlX189MTczMjIxODg2MQ%3D%3D&transactionID=362841651&threeDSResponseCode=65802&threeDSResponseMessage=3DS+authentication+required&threeDSVETimestamp=2024-11-21+19%3A24%3A20&threeDSCheck=not+checked&threeDSURL=https%3A%2F%2Fwww.securesuite.co.uk%2Flloyds%2FthreeDSMethod%2F3ds2%2FRetail&threeDSRequest%5BthreeDSMethodData%5D=eyJ0aHJlZURTTWV0aG9kTm90aWZpY2F0aW9uVVJMIjoiaHR0cHM6Ly9oZW1wZWxmLmNvbS9hL29wZW5wYXRoL3RocmVlZHMvY2FsbGJhY2svcjV3SzRQWmR4U1lLdFoxcFBRYzJSVExici8yMzU1L2hlbXBlbGYuY29tP2Fjcz0xJnRocmVlRFNBY3NSZXNwb25zZT1tZXRob2QiLCJ0aHJlZURTU2VydmVyVHJhbnNJRCI6IjI2ZjIwZWJjLWRiZTYtNDJhNi1iMjZlLTlmYzE1YmNiYjRlMSJ9&threeDSDetails%5BtransID%5D=26f20ebc-dbe6-42a6-b26e-9fc15bcbb4e1&threeDSDetails%5Bversion%5D=2.2.0&threeDSDetails%5Bversions%5D=2.2.0%2C2.1.0&threeDSDetails%5Bfallback%5D=N&threeDSDetails%5BissuerCountryCode%5D=826&threeDSDetails%5BacquirerCountryCode%5D=826&threeDSDetails%5Bpsd2Region%5D=Y&threeDSDetails%5BrequestorChallengeIndicator%5D=01&vcsResponseCode=0&vcsResponseMessage=Success+-+no+velocity+check+rules+applied&currencyExponent=2&amountRetained=0&timestamp=2024-11-21+19%3A24%3A21&merchantName=Hempelf&signature=0728a85331b5efa636ca9c8612239f729aa9d1edf40459452393c2502b12448a95ed9df63153191c721af6ff62eee476481920cec5cce72d320c2d648fafcd3c"

Some details about the transaction

***TransactionID**:*

- 362841651

merchantName

- Hempelf

Order Id:

- r5wK4PZdxSYKtZ1pPQc2RTLbr

threeDSResponseMessage

- *3DS authentication required threeDSVETimestamp = 2024-11-21 19:24:20*

3DS Redirect URL:

- https://hempelf.com/a/openpath/threeds/callback/r5wK4PZdxSYKtZ1pPQc2RTLbr/2355/hempelf.com?acs=1

Server Transaction ID:

- 26f20ebc-dbe6-42a6-b26e-9fc15bcbb4e1

Amount:

- 3774 (Currency Code: 826 for GBP)

Merchant ID:

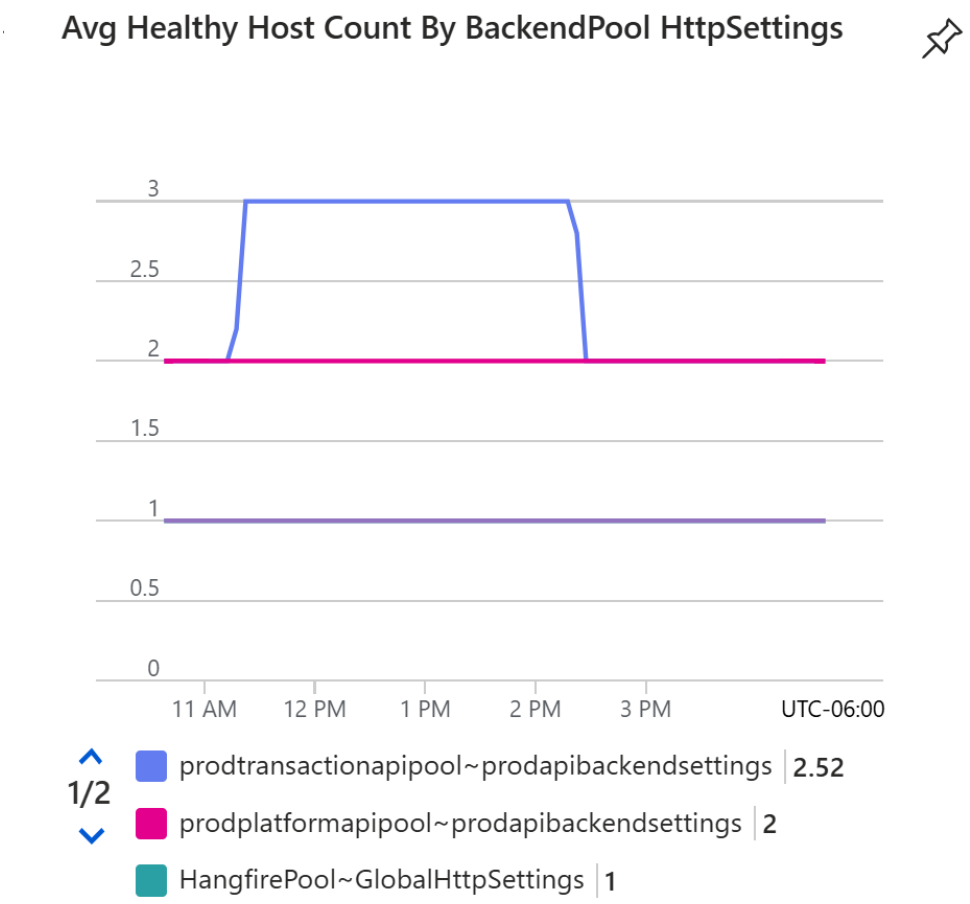
- 235173

Actions taken.

This morning, the new version of the App Service was found to be stopped since approximately 7 AM UTC-06 (or 5 AM PST). To resolve this, the App Service was restarted and re-added to the APIs backend pool.

Investigating - Monitor System Down Checking, on Collection System Down Checking, did not run successfully. 5 request(s) ran and 2 test(s) failed.
Nov 21, 05:25 PST

This issue did not impact operations. Around 11:00 AM (UTC-06), the App Service was restarted and re-added to the APIs backend pool with a new Health Check configuration applied.



The only actions taken were removing the App Service from the Backend Pool in the Application Gateway and disabling the Health Check configuration in the App Service.

A few minutes after making this change, all transactions resumed functioning normally.

The issue observed in the logs, in the other hand, is pointing to the Database, Max pool size reached, i.e. all available connections in the connection pool are in use, and no new connections can be allocated.

This typically occurs when:

1. **Connections are not being properly closed**: The application might be leaking connections by not disposing of or closing them correctly.
2. **High concurrency**: The number of simultaneous database requests exceeds the maximum connection pool size.
3. **Slow queries**: Long-running queries can monopolize connections, causing the pool to fill up.
4. **Inefficient pooling settings**: The connection pool size may be too small for the application's workload.