Error: A network-related or instance-specific error - Troubleshooting steps

Last updated by | Vitor Tomaz | Feb 24, 2023 at 3:26 AM PST

Contents

Issue

The customer is using App Services to connect from their application code to Azure SQL Database. Intermittently, a certain percentage of their connections is failing with the following symptoms reported to the application:

> Microsoft.Data.SqlClient.SqlException (0x80131904): **A network-related or instance-specific error occurred while establishing a connection to SQL Server. The server was not found or was not accessible.** Verify that the instance name is correct and that SQL Server is configured to allow remote connections. (provider: TCP Provider, error: 0 - A connection attempt failed because the connected party did not properly respond after a period of time, or established connection failed because connected host has failed to respond.)
> ---> System.ComponentModel.**Win32Exception (10060): A connection attempt failed because the connected party did not properly respond after a period of time, or established connection failed because connected host has failed to respond.**
> at Microsoft.Data.ProviderBase.DbConnectionPool.CheckPoolBlockingPeriod(Exception e)
> at Microsoft.Data.ProviderBase.DbConnectionPool.CreateObject(DbConnection owningObject, DbConnectionOptions userOptions, DbConnectionInternal oldConnection)
> (...)
> at Microsoft.Data.SqlClient.SqlConnection.Open()
> at Microsoft.EntityFrameworkCore.Storage.RelationalConnection.OpenDbConnection(Boolean errorsExpected)
> **ClientConnectionId:34caa8cc-f6fe-4ca1-a785-785e21711553**
> Error Number:10060,State:0,Class:20
> Routing Destination:b09a65f342ce.HS1.tr169.norwayeast1-a.worker.database.windows.net,11001

A variation of the error may be reported as:

> A network-related or instance-specific error occurred while establishing a connection to SQL Server. The server was not found or was not accessible. Verify that the instance name is correct and that SQL Server is configured to allow remote connections. (provider: TCP provider, error: 0 - A connection attempt failed because the connected party did not properly respond after a period of time, or established connection failed because connected host has failed to respond.) (Microsoft SQL Server, Error: 10060)

Application Insights for the App Service show that requests are timing out with error 10060 after about 20 seconds. Retry logic is in place and does not help avoiding the issue.

Investigation / Analysis

Error code 10060 means that the client cannot connect to the server, either because it is taking too long to get the response from the server, or the response is being lost or blocked. It usually indicates a client-side networking issue, which you need to pursue from the client side. The most likely causes are the client platform being too slow to respond to the login traffic, or a problem with the network that misroutes or drops part of the network traffic.

It may also indicate an issue with the Azure SQL gateway service, e.g. that one of the gateway nodes for the region has a routing or network issue.

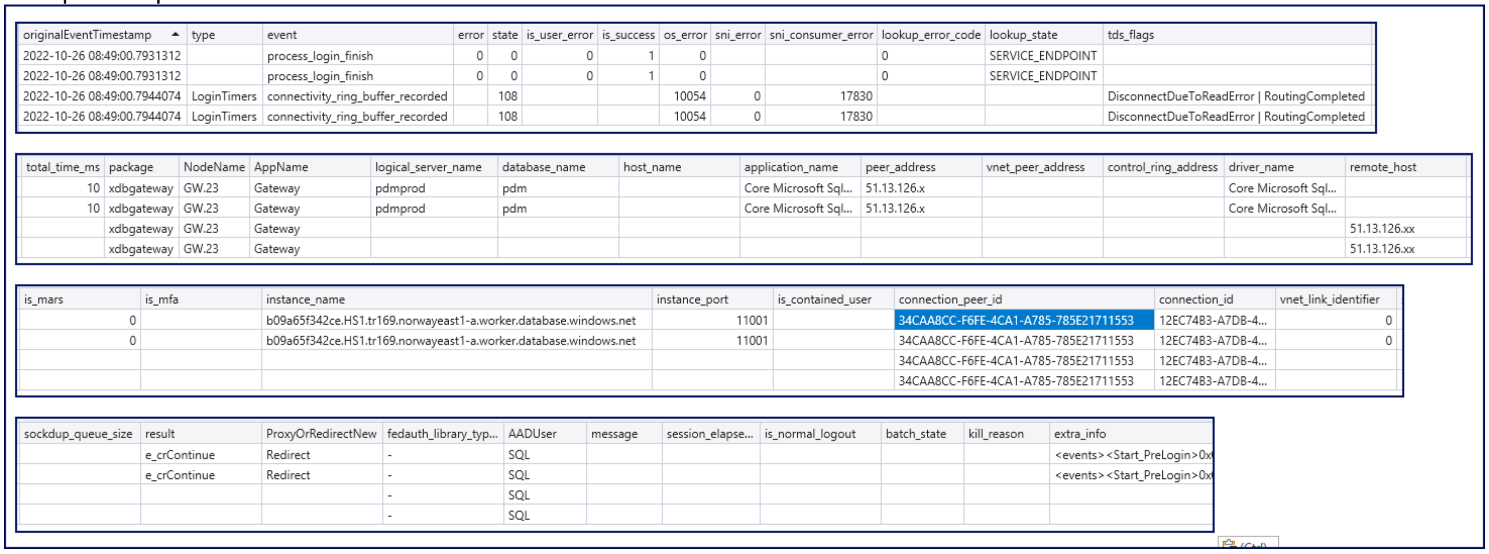
Step 1: Check MonLogin in Kusto

The graceful detail with the first error message is that it contains the client connection ID. This can be associated with the `connection_peer_id` on MonLogin so that it is possible to retrieve further details from the telemetry.

Run the following Kusto query to get further details:

```
let startTime = datetime(2022-10-26 08:30:00);
let endTime = datetime(2022-10-26 09:00:00);
MonLogin
| where TIMESTAMP >= startTime
| where TIMESTAMP <= endTime
| where connection_peer_id =~ "34caa8cc-f6fe-4ca1-a785-785e21711553"
//| where * has"34caa8cc-f6fe-4ca1-a785-785e21711553"
| extend ProxyOrRedirectNew = iif( result =~ "e_crContinue", "Redirect", iif( result =~ "e_crContinueSameState
| extend fedauth_library_type_desc =
    case(
        fedauth_library_type == 0, "SQL Server Authentication",
        fedauth_library_type == 2, "Token Base Authentication",
        fedauth_library_type == 3 and fedauth_adal_workflow == 1, "Azure Active Directory - Password Authentic
        fedauth_library_type == 3 and fedauth_adal_workflow == 2, "Azure Active Directory - Integrated Authent
        fedauth_library_type == 3 and fedauth_adal_workflow == 3, "Azure Active Directory - Universal Authenti
        strcat(tostring(fedauth_library_type) , "-" , tostring(fedauth_adal_workflow))
    )
| extend AADUser = iif( fedauth_adal_workflow > 0 or fedauth_library_type > 0, "AAD" , "SQL")
| project originalEventTimestamp, type, event, error, state, is_user_error, is_success, os_error, sni_error, s
```

Sample output:



originalEventTimestamp	type	event	error	state	is_user_error	is_success	os_error	sni_error	sni_consumer_error	lookup_error_code	lookup_state	tds_flags
2022-10-26 08:49:00.7931312		process_login_finish	0	0	0	1	0			0	SERVICE_ENDPOINT	
2022-10-26 08:49:00.7931312		process_login_finish	0	0	0	1	0			0	SERVICE_ENDPOINT	
2022-10-26 08:49:00.7944074	LoginTimers	connectivity_ring_buffer_recorded		108			10054	0	17830			DisconnectDueToReadError | RoutingCompleted
2022-10-26 08:49:00.7944074	LoginTimers	connectivity_ring_buffer_recorded		108			10054	0	17830			DisconnectDueToReadError | RoutingCompleted

total_time_ms	package	NodeName	AppName	logical_server_name	database_name	host_name	application_name	peer_address	vnet_peer_address	control_ring_address	driver_name	remote_host
10	xdbgateway	GW.23	Gateway	pdmprod	pdm		Core Microsoft Sql...	51.13.126.x			Core Microsoft Sql...	
10	xdbgateway	GW.23	Gateway	pdmprod	pdm		Core Microsoft Sql...	51.13.126.x			Core Microsoft Sql...	
	xdbgateway	GW.23	Gateway									51.13.126.xx
	xdbgateway	GW.23	Gateway									51.13.126.xx

is_mars	is_mfa	instance_name	instance_port	is_contained_user	connection_peer_id	connection_id	vnet_link_identifier
0		b09a65f342ce.HS1.tr169.norwayeast1-a.worker.database.windows.net	11001		34CAA8CC-F6FE-4CA1-A785-785E21711553	12EC74B3-A7DB-4...	0
0		b09a65f342ce.HS1.tr169.norwayeast1-a.worker.database.windows.net	11001		34CAA8CC-F6FE-4CA1-A785-785E21711553	12EC74B3-A7DB-4...	0
					34CAA8CC-F6FE-4CA1-A785-785E21711553	12EC74B3-A7DB-4...	
					34CAA8CC-F6FE-4CA1-A785-785E21711553	12EC74B3-A7DB-4...	

sockdup_queue_size	result	ProxyOrRedirectNew	fedauth_library_typ...	AADUser	message	session_elapse...	is_normal_logout	batch_state	kill_reason	extra_info
	e_crContinue	Redirect	-	SQL						<events><Start_PreLogin>0x
	e_crContinue	Redirect	-	SQL						<events><Start_PreLogin>0x
			-	SQL						
			-	SQL						

This confirms some details from the error message, like routing destination = `instance_name` and `instance_port`, server and database name, and ClientConnectionId = `connection_peer_id`.

The interesting detail is that the connection only reached the gateway level. The `process_login_finish` step was successful, but then within a millisecond, the login was disconnected again with OS error 10054, SNI error 17830, due to a read error. So this is definitely a client-side issue, with the SQL gateway being unable to read from the initial connection.

Step 2: Determine the scope of the issue

The customer reported that the issues occurred intermittently and that it stopped for some time, before picking up again later. The suspicion is that there are more such errors as shown in Step 1 above, and that there might be a pattern.

The problem is that you cannot directly relate these 10054 errors to the target server and database - note how the `logical_server_name` and `database_name` columns are empty on the disconnect rows in the output above. Therefore you need to take an indirect approach: first filter all connections that end with error 10054, then use this result to filter for the customer's server and database. If you filter for error 10054 only, you will get all errors 10054 for all databases that are hosted on the region. You also need to filter for the `remote_host` IP address that you were seeing on Step 1.

Use the following Kusto query to achieve this - although it might be rather slow if the time range is too long:

```
let startTime = datetime(2022-10-27 09:00:00);
let endTime = datetime(2022-10-27 09:10:00);
let srv = "servername";
let db = "databasename";
MonLogin
| where TIMESTAMP >= startTime
| where TIMESTAMP <= endTime
| where os_error == "10054"
| where remote_host == "51.13.126.xx"  // collected from Step 1
| distinct connection_peer_id
| join kind=inner
(
    MonLogin
    | where TIMESTAMP >= startTime
    | where TIMESTAMP <= endTime
    | where LogicalServerName contains srv or logical_server_name contains srv or server_name contains srv //o
    | where database_name =~ db
    | distinct connection_peer_id
    | join kind=inner
    (
        MonLogin
        | where TIMESTAMP >= startTime
        | where TIMESTAMP <= endTime
        //| where client_pid in (4184,3980)
        //| where event =~ "process_login_finish"
        //| where is_success == 0 or total_time_ms > 14000
        //| where ((error > 0 and isnotempty(error)) or (sni_consumer_error <> 0 and isnotempty(sni_consumer_e
        //| where package =~ "sqlserver"
        //| where package =~ "xdbgateway"
        | extend ProxyOrRedirect = iif( result =~ "e_crContinue", "Redirect", iif( result =~ "e_crContinueSame
        | extend fedauth_library_type_desc =
            case(
                fedauth_library_type == 0, "SQL Server Authentication",
                fedauth_library_type == 2, "Token Base Authentication",
                fedauth_library_type == 3 and fedauth_adal_workflow == 1, "Azure Active Directory - Password A
                fedauth_library_type == 3 and fedauth_adal_workflow == 2, "Azure Active Directory - Integrated
                fedauth_library_type == 3 and fedauth_adal_workflow == 3, "Azure Active Directory - Universal
                strcat(tostring(fedauth_library_type) , "-" , tostring(fedauth_adal_workflow))
            )
        | extend AADUser = iif( fedauth_adal_workflow > 0 or fedauth_library_type > 0, "AAD" , "SQL")
    ) on connection_peer_id
) on connection_peer_id
| project originalEventTimestamp, type, event, error, state, is_user_error, is_success, os_error, sni_error, s
| order by originalEventTimestamp asc
```

Step 3: Capture a network trace at the client application side

A network trace will help with identifying if the communication is working as expected or has issues. If you are not familiar with doing this, then involve the network support team through a collab - this will also help with the analysis. Depending on where the application is hosted, you might have to involve additional teams, e.g. Azure VM or App Services.

Mitigation

The mitigation steps depend on the results you get returned in the investigation steps.

In the scenario described above, the network trace revealed that each initial SYN request from the client was followed by two `SynReTransmit` packets a bit later; neither of them got answered by the SQL gateway. In a 10 minute trace there were hundreds of such retransmit packages, interleaved with successful connections and query result traffic returned from the database. This required further investigation from the App Service and Network support teams.

If you however see an issue pointing to an error at the SQL gateway level, then it depends on the current status. If the issue has already mitigated itself and no detailed RCA is needed, then you can explain it as a temporary gateway issue to the customer. If a detailed RCA is needed or if the issue is ongoing, you should open an IcM with the gateway team to investigate further. If unsure, involve your TA.

How good have you found this content?