Network-related or instance-specific errors

Last updated by | Subbu Kandhaswamy | Jul 18, 2022 at 9:28 PM PDT

Contents

Error in Context

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 - An operation on a socket could not be performed because the system lacked sufficient buffer space or because a queue was full.)

The full error message from TCP Provider will tell you more about what the actual problem is. Note that this will be visible only if protocol is explicitly defined in the server_name field

How to understand various types of this error?

Scenario 1
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: Named Pipes Provider: Could not open a connection to SQL Server).

What this means

This hides the underlying error (that comes from tcp provider or SNI provider). This error message is not helpful. Ask customer to get the right error message by forcing tcp in their protocol (sqlcmd -S tcp:servername.database.windows.net -U username -P password -d database)

Scenario 2

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: -1 No such host is known.)

What this means

Most likely the DNS Record does not exist.

- If it exists, it's also possible that tcp endpoint is unresponsive. TCP Timeout is 21s and client did not receive SYN-ACK in response to its SYN packet within that time.
- Do tcping to the endpoint to capture statistics.

Scenario 3

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: SQL Network Interfaces, error: 26 - Error Locating Server/Instance Specified)

What this means

This happens because tcp protocol is not forced. Prepend servername with tcp: to get more helpful error message to understand the issue.

Scenario 4

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 - An operation on a socket could not be performed because the system lacked sufficient buffer space or because a queue was full.)

What this means

Client side issue typically. Local machine having buffer exhaustion or port exhaustion. Look for processes that are using up tcp (netstat.exe will tell you more), or if network output buffer is backed up (PerfMon will have counters for this)

Recommended Checks

Is this happening across all clients?

- Attempt connection to database from your dev box after replacing these values . If you can connect (you might get login failed or firewall error), this limits the error to the client side

```
sqlcmd -S tcp:servername.database.windows.net -U dummyValue -P dummyValue -d databasename
```

- If you cannot connect from your machine, it is clear that this limits error to server side

Client side

- Client side error message from customer
- do PSPing (*download here*) to trace servername.database.windows.net on port 1433

```
(psping -t <server>:<port>
```

- If this shows "Socket not available* errors every now and then or takes anything more than 1000ms to connect, this potentially indicates networking issue. Engage CloudNet team thru Collab task.
- If psping above doesn't show anything, getting wireshark traces while attempting to connect tcp:servername.database.windows.net via client will be useful.

Server side

- Check Is the issue transient?
- Are they connecting to Secure Proxy? (This is a TDS Proxy, much like Gateway, but audits all actions going through it. This is owned by the Data Security Proxy team in ILDC).
- If they are using Secure Proxy, engage Data Security Proxy team to help debug issues on their side.
- Is login being redirected? check Monlogin.

```
MonLogin | where event == "process login finish" and logical server name == "servername" and database_name == "dbname" | where originalEventTimestamp > ago(1h) | project originalEventTimestamp, result, peer_address, host_name
```

- If result = e_crContinue, then it is being redirected to backend (or security proxy).
- If it is e_crContinueSameState then it is being proxied

Confirming this will help isolate the problem further.

Recommended Mitigation Steps

- Isolate whether problem is with listening process or network. We have a network monitoring service that listens to a specific TCP port on every node in SQL Azure clusters. The port it listens to is different from the TDS port listened to by Gateway and XDBHost.
- To tcping to Gateway, connect to "servername.database.windows.net" on tcp port 1433.\ To tcping to backend xdbhost, find out where the database is currently hosted (primary) from "Database Availability.xts". Port number you want is 11000 + number of the node. For example, to connect to xdbhost in DB.111 of tr21 westus, Connect tr21.westus1-a.worker.database.windows.net,11111
- Use the instruction below to ping network monitoring service (netmonsvc). If connectivity to netmonsvc is clean, but to gateway/xdbhost is not clean, this implies the problem is with the process and not the networking layer.
    - "Netmonsvc" is a separate application running on GW and DB nodes for TCP probing, you can use it to check network issue, e.g. TCP connection, packet loss, etc.,
    - This service, on gateway node listens port range 16000 to 16000 + <GW Node number> -1 (*for example - if GW Node is 50, the listening port ranges between 16000 to 16049*)
    - This Service on DB Node listens from 16500 t0 16500 + <DB Node number> -1 (*example - if DB Node is 30, the listening port ranges between 16500 to 16529*)
- To ping the Port using PSPing or TCPPing you follow the examples
    - Gateway Node

ClusterName	.	NodeName
cr5.centralus1-a.control.database.windows.net	.	GW.10

```
C:\psping>psping cr5.centralus1-a.control.database.windows.net:16009

PsPing v2.10 - PsPing - ping, latency, bandwidth measurement utility
Copyright (C) 2012-2016 Mark Russinovich
Sysinternals - www.sysinternals.com

TCP connect to 104.208.21.1:16009:
5 iterations (warmup 1) ping test:
Connecting to 104.208.21.1:16009 (warmup): from 192.168.254.30:49952: 44.82ms
Connecting to 104.208.21.1:16009: from 192.168.254.30:49956: 45.10ms
...
Connecting to 104.208.21.1:16009: from 192.168.254.30:49959: 44.65ms
```

- DB Node

ClusterName	NodeName	...	LogicalServerI
tr19.centralus1-a.worker.database.windows.net	DB.127	..	seahawks

```
C:\psping>psping tr19.centralus1-a.worker.database.windows.net:16626

PsPing v2.10 - PsPing - ping, latency, bandwidth measurement utility
Copyright (C) 2012-2016 Mark Russinovich
Sysinternals - www.sysinternals.com
 TCP connect to 40.77.30.201:16626:
  5 iterations (warmup 1) ping test:
  Connecting to 40.77.30.201:16626 (warmup): from 192.168.254.30:49211: 44.84ms
  Connecting to 40.77.30.201:16626: from 192.168.254.30:49212: 45.33ms
  ...
  Connecting to 40.77.30.201:16626: from 192.168.254.30:49216: 45.37ms
```

- If connectivity to netmonsvc is clean, but to gateway/xdbhost is not clean, this implies the problem is with the process and not the networking layer.
- If the server connection to the backend is failing (**Servier Side** section above), and netmonsvc test to backend is clean, engage Gateway Engineering to help resolve issue sharing your findings.

How good have you found this content?