Error 40613, State 10

Last updated by | Subbu Kandhaswamy | Sep 13, 2021 at 2:27 PM PDT

Contents

Issue

This is the error when login comes to xdbhost, and xdbhost cannot find the target SQL Instance based on the incoming login info. This can happen if a gateway directs logins to a node that does not host the user database - either temporarily during a failover, or permanently due to some other issue.

Troubleshoot

Using ASC

- If you're new to ASC, follow the instructions to generate SQL-troubleshooter results from here -> How to generate or use SQL troubleshooter results

- As part of startup, sql server registers itself with xdbhost. Similarly, as part of process finish, it unregisters from xdbhost. ASC troubleshooter results (XDBHost 40613 failures tab) we'll be able to review the the amount of time this registration / unregistration happened as seen from XDBHost.

- A socket duplication failures could be explained by frequent sql server restarts. Every unregister_socket_dup_instance indicates that the sql server de-registered from xdbhost and the process died (on purpose or unexpectedly). register_endpoint traces accompany a sql server registration event with xdbhost (happens as part of sql startup). Note that sql startup doesn't mean it would register with xdbhost successfully, but if it registered successfully it does mean sql server has started up.

- Confirm the occurrence of the issue from the results in Data Explorer

- Check if the issue is still occurring (if customer reported as ongoing)

○ Attempt to login to the failing server and database *with any random credentials* from a NON-SAW machine. (SAW machine firewalls cause pre-login errors if database is forced to redirection).

○ You should *NOT* get a Database Unavailable nor Login Timeout errors.

○ Firewall error or Bad Password (login failed to user 'blah') mean SQL Server is healthy. Examples:

Examples of unhealthy/unavailable responses

```
1) c:\test> sqlcmd -S <_Customer Servername_>.database.windows.net -U test -P test -d <_Database_Name_>
Sqlcmd: Error: Microsoft SQL Server Native Client 11.0 : Database 'Content_274449' on server 'servernmame' is

2) c:\test>sqlcmd -S <_Customer Servername_>.database.windows.net -U test -P test -d <_Database_Name_>
Sqlcmd: Error: Microsoft SQL Server Native Client 11.0 : TCP Provider: Timeout error [258]. .
Sqlcmd: Error: Microsoft SQL Server Native Client 11.0 : Login timeout expired.
Sqlcmd: Error: Microsoft SQL Server Native Client 11.0 : Unable to complete login process due to delay in logi

3) c:\test>sqlcmd -S <_Customer Servername_>.database.windows.net -U test -P test -d <_Database_Name_>
Sqlcmd: Error: Microsoft SQL Server Native Client 11.0 : TCP Provider: Timeout error [258]. .
Sqlcmd: Error: Microsoft SQL Server Native Client 11.0 : Login timeout expired.
Sqlcmd: Error: Microsoft SQL Server Native Client 11.0 : Unable to complete login process due to delay in logi
```

◀ ▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬   ▶

Examples of Available/Recovered responses (either firewall error or bad password)

```
1) c:\test>sqlcmd -S <_Customer Servername_>.database.windows.net -U test -P test -d <_Database_Name_>
Sqlcmd: Error: Microsoft SQL Server Native Client 11.0 : Cannot open server 'yjaij20em8' requested by the logi

2) c:\test>sqlcmd -S ea-uk-pm-dti-1.database.windows.net -U test -P test -d
Sqlcmd: Error: Microsoft ODBC Driver 13 for SQL Server : Login to read-secondary failed due to long wait on 'H
```

◀ ▬▬▬▬▬▬▬▬▬   ▶

If availability issue is confirmed, then engage engineering by following ICM escalation process.

Kusto

To identify the resources

```
MonLogin
| where originalEventTimestamp >= ago(20m)
| where error == 40613 and state == 10
| where NodeName == <node_name>
| project logical_server_name , database_name , instance_name , driver_name, ClusterName, NodeName
```

Note: It is possible that the MonLogin query will return an empty server and database name, which is expected. If that is the case, then follow step 2

Find the DB replica, using xts - > DB lookup view based on server/db name, or instance name This view will also return the server/db name, if not obtained in first step

If DB is under reconfiguration or has continuously failover for the past short time , follow Availability TSGs, and involve Availability on-call if needed.

Follow the remaining steps below *only* if there is no Availability issue and the database is healthy.

If the Primary replica is on a different node than the one shown in the incident, check if any GW redirects the login to the wrong DB node. e.g.

```
MonLogin
| where originalEventTimestamp >= ago(20m)
| where package == "xdbgateway"
| where logical_server_name == <logical server name> and database_name == <db>
| where error == 0
| summarize count() by NodeName, ClusterName, instance_name, fabric_node_name
```

If the logical server name and database name are not known, filter by instance or app name instead:
```
| where instance_name startswith <instance_name or app_name>
```

Note: Please use a short time interval of 20 minutes in the query. Using a larger interval can capture telemetry before and after failover, leading to false positives.

Mitigation

If the issue is active and confirmed availability issue, engage HA team to investigate further to mitigate.

Additional info : At times, the issue could be related to client side driver (ODBC/.net clients) .

- Try login (with ODBC ControlPanel or SQLcmd) with fake user/pwd.
- if you hit login failure/firewall error, this means the DB is available, this probably might be a client side driver issue, especially when driver is Tedious, OLEDB or blank._
- We seen this issue occur with very old ODBC / .NET clients too. And to confirm this as client issue: Verify (using query above) that no gateways are directing logins to the node where xdbhost reports the error._

RCA Template

Summary of Impact : Between **<Starttime>** and **<EndTime>** Database **<Database Name>** on Server **<Server name>** was not reachable, and this unavailability errors (40613) you reported were due to - Connection unable to reach the target SQL instance based on the incoming login information.

Cause: This mostly occurs when update SLO initiated and failover occurs (during failover process, DB switches to new primary node). To be specific, it happen if a gateway directs logins to a node that does not host the user database - either temporarily during a failover. The logins were still landing on the wrong old node and causing unavailability. Therefore there was a brief period of time wherein the logins were failing with 40613.

Recommendations - In most cases these are transient and apply retry logic to the application should help handle these effectively.

Classification

Root Cause: Azure SQL DB v2\Availability (for availability issues) Root Cause: Azure SQL DB v2\Availability (for non-availability issues)

How good have you found this content?