DB Inaccessible while running Alter Commands

Last updated by | Holger Linke | Feb 28, 2023 at 1:53 AM PST

Contents

Issue

The customer encountered an issue where the Azure SQL Database became seemingly unavailable after applying changes to the permission assignments. Specifically, three consecutive `ALTER ROLE` commands were executed against the database and made the database inaccessible for over 30 minutes. After the database became available/accessible again, the changes had not been saved (were rolled back).

Here are the commands that were executed on the database:

```
ALTER ROLE db_datareader DROP MEMBER [mymembername]
ALTER ROLE db_datawriter DROP MEMBER [mymembername]
ALTER ROLE db_standardrole ADD MEMBER [mymembername]
```

Error

The application logs reported the following errors:

> Error Message: SqlException (-2): Connection Timeout Expired.
> **The timeout period elapsed during *the post-login phase*.**
> The connection could have timed out while waiting for server to complete the login process and respond; Or it could have timed out while attempting to create multiple active connections.
> **This failure occurred while attempting to connect to the routing destination.**
> The duration spent while attempting to connect to the original server was - [Pre-Login] initialization=5; handshake=8; [Login]initialization=0; authentication=0; [Post-Login] complete=0; The duration spent while attempting to connect to this server was - [Pre-Login] initialization=26; handshake=2; [Login] initialization=0; authentication=0; [Post-Login] complete=14014;

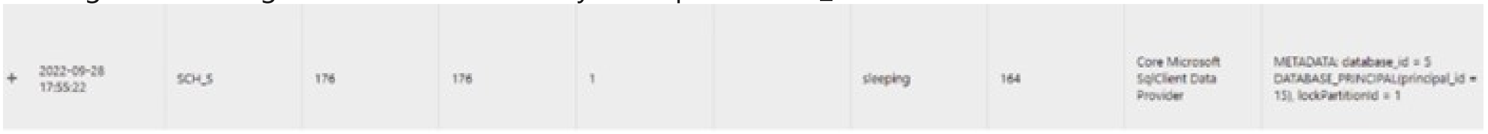
> Sqlcmd: Error: Microsoft ODBC Driver 17 for SQL Server : Cannot open server 'servername' requested by the login. Client with IP address <ipv4 address> is not allowed to access the server. To enable access, use the Windows Azure Management Portal or run sp_set_firewall_rule on the master database to create a firewall rule for this IP address or address range. It may take up to five minutes for this change to take effect..

Investigation

The symptom for the customer is that the database is unavailable and inaccessible. This is not true though; the database is still healthy and there are no indication of a performance issue or downtime when you check in ASC.

The reason rather is that a system process is blocking other sessions and prevents any users from making new connections to the database. The `ALTER ROLE` command is the blocking session. When the customer executed the ALTER command, it holds the lock for a long time, and while that happens, no other login succeeds, giving the impression that the database is down.

The logins are failing because of the inability to acquire a SCH_S lock:



+	2022-09-28 17:55:22	SCH_S	176	176	1		sleeping	164	Core Microsoft SqlClient Data Provider	METADATA: database_id = 5 DATABASE_PRINCIPAL(principal_id = 15), lockPartitionId = 1

Analysis

Further discussions with the customer brings more clarity to the cause:

- they are using an application that has `IMPLICIT_TRANSACTIONS` set to "ON" by default (JDBC-based apps seem to be key offenders)
- they are using an application that has implicit transactions set to "ON" through the `SET ANSI_DEFAULTS` options

Implicit transactions begin a transaction implicitly and require an explicit commit for it to complete. Thus a transaction has been left open when running the `ALTER` commands, and their changes had been rolled back when the open transaction/session was finally killed.

The issue is similar to the symptoms and cause described in Contained user login timeout due to blocking; note the slightly different timeout message, caused by a different type of blocking.

Mitigation

There are several options for mitigation:

- Run a `COMMIT TRAN` after each `ALTER` statement.
- Set `IMPLICIT_TRANSACTIONS` to "OFF" in the session, either through the application's connection options or through `SET ANSI_DEFAULTS OFF` . This option is often hidden behind the SQL driver's "auto_commit" option, with "IMPLICIT_TRANSACTIONS OFF" = "auto_commit ON".

Reference documents

- SET ANSI Defaults
- SET IMPLICIT_TRANSACTIONS

How good have you found this content?