Connectivity errors - on Geodr secondary

Last updated by | Holger Linke | Feb 28, 2023 at 1:54 AM PST

Contents

Issue

The customer is trying to connect to a geo-replication secondary server and database, but it is failing with a login error and a hint to database unavailability.

The error message from SSMS includes the following details:

> Cannot connect to secondaryservername.database.windows.net.
>
> **Additional information:**
> The server principal "testuser" is not able to access the database "databasename" under the current security context.
> Database 'databasename' on server 'ab41d134d624.tr35.westus1-a.worker.database.windows.net' is not currentiy available. Please retry the connecton later. If the problem persists, contact customer support, and provide them the session tracing ID of '34EE8A7-EC55-472E-goc5-D77364022F56'
> Login failed for user 'testuser' (Microsoft SQL Server, Error: 916)

Investigation / Analysis

The explanation for error 916 indicates that the user connecting to the database needs to have at least `CONNECT` permissions. If a server login cannot be matched to a database user, it will be mapped to the "guest" user, which is disabled by default on Azure SQL Database. For contained users, the user still needs to have `CONNECT` permissions either by itself or through role membership; this permission might have been revoked.

In the context of a geo-replication secondary server and database, the error may occur when logins on the secondary are not mapped correctly from the primary server. This may happen if you executed `CREATE LOGIN` on both servers with the same login name, but have not used the `SID` parameter to provide the same SID on both sides. Geo-replication will replicate the user SID from primary to secondary but not the logins, which will then be mismatched to the secondary's server login SID.

It is also possible that the customer has not provided proper user permissions to the database at the primary server/database, and that the failed login at the secondary exposes an issue that already exists at the primary

database.

Use the following troubleshooting steps to narrow down the issue.

Step 1 - Verify proper login mapping between primary and secondary

Run the following query in the "master" database on both Primary and Secondary server. It will return the login name name and its SID. Compare the SIDs for matching login names to identify mismatched SIDs:

```
SELECT [name], [sid]
FROM [sys].[sql_logins]
WHERE [type_desc] = 'SQL_Login'

Sample output - PRIMARY:

name       sid
---------- ------------------------------------------------------------------
testuser   0x0106000000000064000000000000000956D9DDCE3BBD8469B04F4AE0277AA18
otheruser  0x0106000000000064000000000000000553800694B934244A1485150F2253722

Sample output - SECONDARY:

name       sid
---------- ------------------------------------------------------------------
testuser   0x0106000000000064000000000000004F1C2D39AE999147B9DA9792FDF4DE07
otheruser  0x0106000000000064000000000000000F2DC33FD1B1A6947A9FFE4C7E29B6743
```

Note how the SID of login "testuser" is different on primary and secondary server.

Step 2 - Compare the database user SID with the login SID

Run the following query in the affected user database on both Primary and Secondary server:

```
SELECT [name], [sid]
FROM [sys].[database_principals]
WHERE [type_desc] = 'SQL_USER'

Sample output - PRIMARY:

name       sid
---------- ------------------------------------------------------------------
dbo        0x0106000000000164000000000000000F26AA....
testuser   0x0106000000000064000000000000000956D9DDCE3BBD8469B04F4AE0277AA18

Sample output - SECONDARY:

name       sid
---------- ------------------------------------------------------------------
dbo        0x0106000000000164000000000000000F26AA....
testuser   0x0106000000000064000000000000000956D9DDCE3BBD8469B04F4AE0277AA18

(login SID:0x0106000000000064000000000000004F1C2D39AE999147B9DA9792FDF4DE07 from above)
```

Note how the SID of database user "testuser" is the same on primary and secondary database. Also note how login and user SIDs are different on the secondary.

If you you try to login as "testuser" to the secondary, it will find the login, take the login SID, go into the user database, search for the SID, and fail to find it.

Mitigation

Replace the login SID on the secondary server to use the same SID as the primary.

You need to first drop the existing login at the secondary, then recreate it by providing the correct SID:

```
-- in 'master' of the secondary
-- the example SID is taken from the sample output above - replace with your own

DROP LOGIN testuser;

CREATE LOGIN testuser
WITH PASSWORD ='$uperPw§1'
, SID = 0x0106000000000006400000000000000000956D9DDCE3BBD8469B04F4AE0277AA18
```

If this doesn't solve the issue, you should try to connect to the primary database through the same login/user and see if you can perform the same intended steps at the primary. Check the user permissions to verify proper access.

Public Doc Reference

- Set up user access to a secondary or recovered database
- Configure and manage Azure SQL Database security for geo-restore or failover

Classification

Root Cause: Azure SQL DB v2\Connectivity\Login Errors\Other

How good have you found this content?