Transaction Replication on a Geo Replicated Environment failed

Last updated by | Vitor Tomaz | Feb 18, 2021 at 2:30 AM PST

Contents

Transaction Replication on a Geo Replicated Environment failed

Scenario

(Applies only to SQL DB)

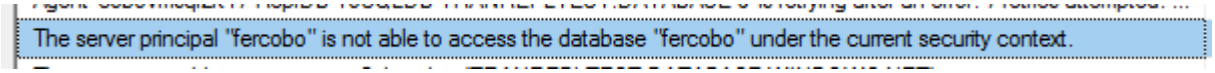
Geo-replication enabled (PRIMARY and SECONDARY server), and user was able to configure one-way transaction replication , but experience issue when a reconfiguration /failover occurs.

Cause

What happens here is that, since the Subscriber is tied to a specific server FQDN in a Geo-replication relationship, once the PRIMARY becomes the new SECONDARY, therefore Read-only only, it will fail to replicate changes from on-prem to Azure SQL DB.

What is the workaround? Well, instead of using Geo-replication you will need to ask the customer to use Failover Groups instead. Why? Well, since the Failover Group has a listener endpoint and it abstracts the two servers to PRIMARY and SECODARY, the Failover Groups will manage itself to switch between those two without having to change it's FQDN, therefore Transactional Replication won't break, BUT it won't work.

The difference here is that if you have Geo-replication, it will fail, since you're pointing to a server that is now Read-only, therefore writes will fail for all the uncommitted commands from the on-prem database. In the other hand, since the Failover Group has it's own FQDN and manages switching PRIMARY and SECONDARY behind the scenes, it will just switch those, and the Transactional Replication won't break, but it won't replicate the data... it will instead just keep retrying... but if you check the Replication Monitor, you'll notice this:



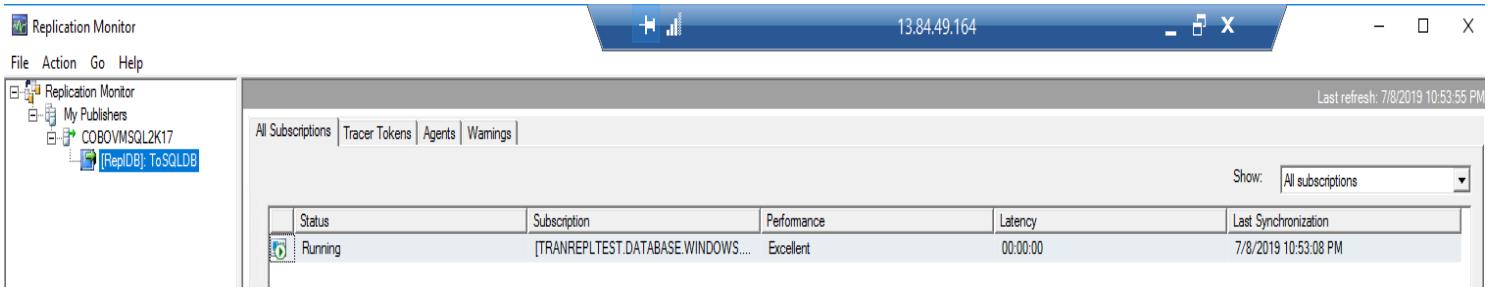
Even tough server A and server B have the same exact logins, it will fail due to the difference in Security Context.

The advantage with Failover Groups opposed to Geo-replication is that with Failover Groups the Distributor will keep trying to replicate and will never error out, it will just keep on retrying, opposed to Geo-rep that will instantly fail. Once you execute another manual failover, it will work and replication will resume.

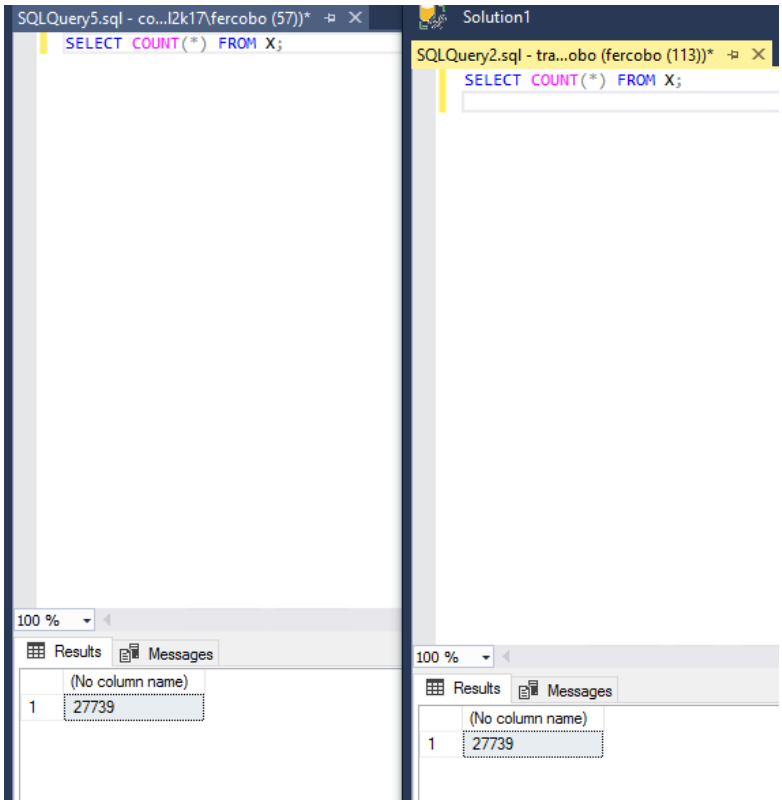
Mitigation /Workaround

The workaround is to go to the Failover Group again, start a manual Failover and then Transactional Replication will resume until it is up to date and all uncommitted commands are sent to the Azure SQL DB.

It will show like this once it is done:



And doing a simple SELECT COUNT will show that all commands have been replicated correctly now:



Classification

Root Cause: Azure SQL DB v2\GeoDR/AutoDR

How good have you found this content?