Geo Replication Chaining (Secondary of Secondary)

Last updated by | Vitor Tomaz | Aug 5, 2020 at 12:41 PM PDT

Contents

Scenario

From the Azure Portal, it can appear only replicas from Primary are possible. Although you can select the secondary > Geo-replica, but it doesn't go over 4 replicas, which is the limit of normal replica.

"If geo-replication is enabled, the application can initiate failover to a secondary database in a different Azure region. Up to four secondaries are supported in the same or different regions, and the secondaries can also be used for read-only access queries."
https://docs.microsoft.com/en-us/azure/sql-database/sql-database-geo-replication-overview

If a customer wants to go beyond 4, and according to the doc, the only way to do it is secondary of secondary

If tried from the portal by creating a secondary after selecting a secondary, but it gives the error:

Replication limit reached. The database 'dbname' cannot have more than 4 replication relationships

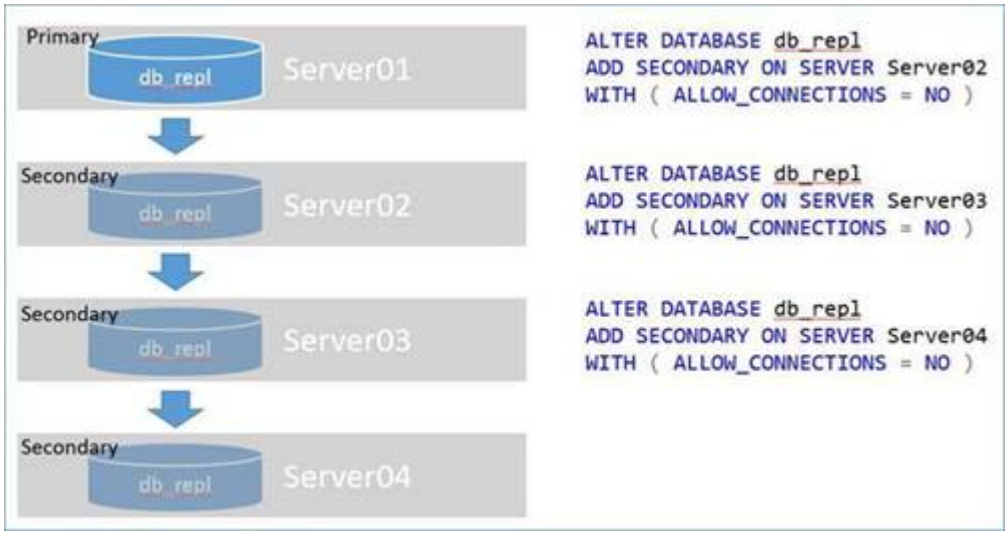
Solution

The solution is to use chaining replication (secondary of secondary), which is currently not supported (*as of 2018-10-11*) to be configured on Azure portal, but it is easy to be configured by using the following **T-SQL** statement.

```
ALTER DATABASE [database_name]   -- current secondary database

ADD SECONDARY ON SERVER [server_name] -- chained secondary server

WITH ( ALLOW_CONNECTIONS = NO )
```

To create a chained replication, just execute the above T-SQL statement on a server where you want to establish a secondary replication. (Make sure you modify the database name to be replicated, and the server name of the server that will host the secondary databases.)

E.g., the following illustration demonstrate a geo-replication chain, and the T-SQL statement used to create the replication chain.



You

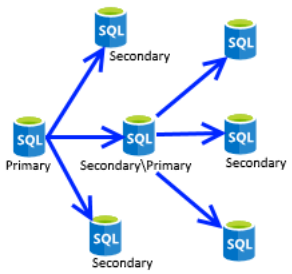
can also use **Powershell**

```
$databasename = "SampleDB"

$originalprimaryserver = "azuresupportacademy"


$primaryresourcegroupname = "FranceCentral"

$primaryservername = "drserver"


$secondaryresourcegroupname = "vtpombei"

$secondaryservername = "weserver"

$secondarylocation = "westeurope"


$adminlogin = "<adminlogin>"

$password = "<password>"


#If there is a need to create an empty server

New-AzureRmSqlServer -ResourceGroupName $secondaryresourcegroupname `

    -ServerName $secondaryservername `
```

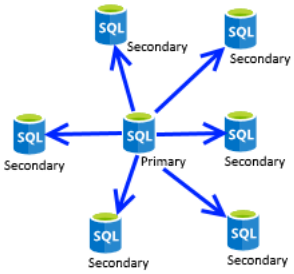
```
    -Location $secondarylocation `

    -SqlAdministratorCredentials $(New-Object -TypeName System.Management.Automation.PSCredential -ArgumentList $adminlogin, $(ConvertTo-SecureString -String $password -AsPlainText -Force))


#Create the geo-replication

$database = Get-AzureRmSqlDatabase -DatabaseName $databasename -ResourceGroupName $primaryresourcegroupname -ServerName $primaryservername

$database | New-AzureRmSqlDatabaseSecondary -PartnerResourceGroupName $secondaryresourcegroupname -PartnerServerName $secondaryservername -AllowConnections "All"


#Get the geo-replication info from original primary to secondary

$database = Get-AzureRmSqlDatabase -DatabaseName $databasename -ResourceGroupName $primaryresourcegroupname -ServerName $primaryservername

$database | Get-AzureRmSqlDatabaseReplicationLink -PartnerResourceGroupName $primaryresourcegroupname -PartnerServerName $originalprimaryserver


#Get the geo-replication info from secondary to secondary of secondary

$database = Get-AzureRmSqlDatabase -DatabaseName $databasename -ResourceGroupName $secondaryresourcegroupname -ServerName $secondaryservername

$database | Get-AzureRmSqlDatabaseReplicationLink -PartnerResourceGroupName $primaryresourcegroupname -PartnerServerName $primaryservername
```

On the portal you can see the chaining replication if you go to the “SQL Database”.

You can go to higher number of secondaries if you failover to the secondary that is also the primary on the other replication, but attention that is not supported.

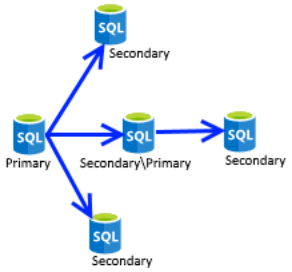


If we have this configuration and we failover to the Secondary\Primary

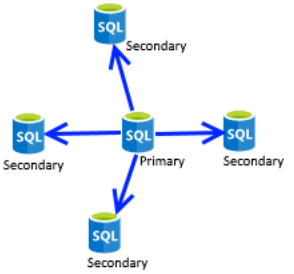


He get more than 4 secondaries but this is not supported

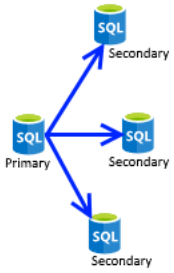
If we failover to the "Secondary\Primary" replica we can't easily go back to the original configuration.



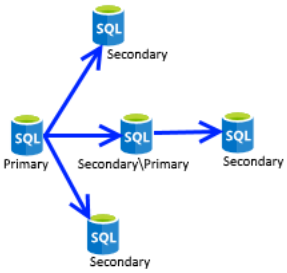
Chained replication



If you failover to the Secondary\Primary you can't go back to the configuration on the left unless you redo some of the replications links



Failover back to the original Primary, break the replication and drop the database



Recreate again the replication from the secondary to get the original configuration

Key Words

Geo Replication

Geo DR

Chaining

Chain

Secondary of Secondary

Portal

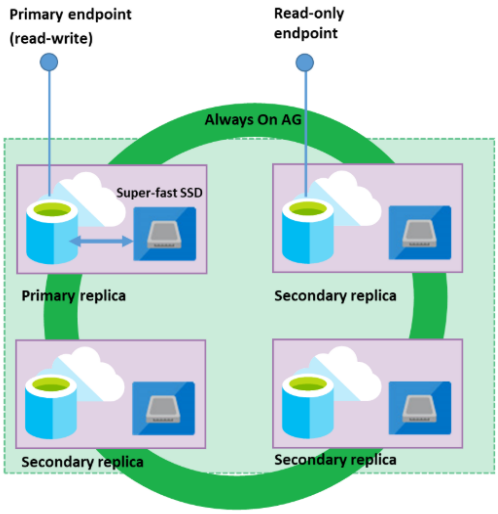
TSQL

Powershell

Workaround use Read Scale endpoints

With 1 primary + 4 replicas = 1 read/write endpoint + 9 readonly endpoint = **More cost effective**

https://docs.microsoft.com/en-us/azure/sql-database/sql-database-read-scale-out



Need to be Premium Tire

Need to set manually for each DB

$databasename = "db"

$resourcegroupname = "ResGroup"

$servername = "Server"

$database = Get-AzureRmSqlDatabase `

-DatabaseName $databasename `

-ResourceGroupName $resourcegroupname `

-ServerName $servername

$database | Set-AzureRmSqlDatabase -ReadScale Enabled

Use different connection strings

- Server=tcp:<server>.database.windows.net;Database=<mydatabase>;ApplicationIntent=ReadOnly;User ID=<myLogin>;Password=<myPassword>;Trusted_Connection=False; Encrypt=True;
- Server=tcp:<server>.database.windows.net;Database=<mydatabase>;ApplicationIntent=ReadWrite;User ID=<myLogin>;Password=<myPassword>;Trusted_Connection=False; Encrypt=True;

Classification

Root Cause: Azure SQL DB v2\GeoDR/AutoDR

How good have you found this content?