Bidirectional replication

Last updated by | Holger Linke | Aug 18, 2022 at 4:50 AM PDT

Contents

Update 2022-08-18

The article below is a basic script for setting up bidirectional Transactional Replication for SQLVM-to-MI, MI-to-SQLVM, and MI-to-MI scenarios.

A detailed, commented, and customer-ready sample script for the MI-to-MI scenario is available at Bidirectional Transactional Replication. Its content can be shared freely with customer, either directly or through our public blog at Bidirectional Transactional Replication in SQL Managed Instance ⧉.

Also see the article Bidirectional Transactional Replication Considerations regarding advantages and disadvantages of bidirectional replication.

Original content

It works fine to replicate the same table from MI to SQL VM as well as from SQL VM to MI. Same script works when setup bidirectional replication between two MI.

On SQL MI side:

```
USE [master]

GO

EXEC sp_adddistributor @distributor = @@ServerName;

EXEC sp_adddistributiondb @database = N'distribution';

GO

:setvar username xxxx

:setvar password xxxxxxxxx

:setvar file_storage "\\xxxxx.file.core.windows.net\xxxxx"

:setvar file_storage_key "DefaultEndpointsProtocol=https;AccountName=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx=

USE [master]

EXEC sp_adddistpublisher

    @publisher = @@ServerName,

    @distribution_db = N'distribution',

    @security_mode = 0,

    @login = N'$(username)',

    @password = N'$(password)',

    @working_directory = N'$(file_storage)',

    @storage_connection_string = N'$(file_storage_key)';

        -- Set variables

:setvar username xxxxx

:setvar password xxxxxxxxxxxxxxxxxx

:setvar source_db danzhang_repl

:setvar publication_name PublishData

:setvar object XXXXXX

:setvar schema dbo

:setvar target_server "VM IP"

:setvar target_username XXXX(I used sql account here)

:setvar target_password xxxxxx

:setvar target_db xxxxxx

-- Enable replication for your source database

USE [$(source_db)]

EXEC sp_replicationdboption

    @dbname = N'$(source_db)',

    @optname = N'publish',
```

```sql
    @value = N'true';
  -- Create your publication
  EXEC sp_addpublication
    @publication = N'$(publication_name)',
    @status = N'active';
  -- Configure your log reaer agent
  EXEC sp_changelogreader_agent
    @publisher_security_mode = 0,
    @publisher_login = N'$(username)',
    @publisher_password = N'$(password)',
    @job_login = N'$(username)',
    @job_password = N'$(password)';
  -- Add the publication snapshot
  EXEC sp_addpublication_snapshot
    @publication = N'$(publication_name)',
    @frequency_type = 1,
    @publisher_security_mode = 0,
    @publisher_login = N'$(username)',
    @publisher_password = N'$(password)',
    @job_login = N'$(username)',
    @job_password = N'$(password)';
  -- Add the ReplTest table to the publication
  EXEC sp_addarticle
    @publication = N'$(publication_name)',
    @type = N'logbased',
    @article = N'$(object)',
    @source_object = N'$(object)',
    @source_owner = N'$(schema)',
    @schema_option = 0x80030F3;//I need to use this schema option otherwise the primary key got lost upon replic
  -- Add the subscriber
  EXEC sp_addsubscription
    @publication = N'$(publication_name)',
    @subscriber = N'$(target_server)',
    @destination_db = N'$(target_db)',
    @subscription_type = N'Push',
```

```
    @sync_type = N'replication support only', //This option is needed thus we don't need to run snapshot. Snapsh

    @loopback_detection ='TRUE'; //This option is also needed for bidirectional replication

-- Create the push subscription agent

EXEC sp_addpushsubscription_agent

    @publication = N'$(publication_name)',

    @subscriber = N'$(target_server)',

    @subscriber_db = N'$(target_db)',

    @subscriber_security_mode = 0,

    @subscriber_login = N'$(target_username)',

    @subscriber_password = N'$(target_password)',

    @job_login = N'$(target_username)',

    @job_password = N'$(target_password)';
```

On SQL VM side,

1. I used the GUI to create the publication as I got login failed error for some reason when using the TSQL command.
2. I used following script to add subscription

```
use [dbname]

exec sp_addsubscription @publication = N'publicationname', @subscriber = N'mi.xxxxx.database.windows.net', @de

exec sp_addpushsubscription_agent @publication = N'publicationname', @subscriber = N'mi.xxxxxxx.database.windo

GO
```

How good have you found this content?

🙂 🙁